

VARCHART XGantt

ActiveX Edition 5.2
Benutzer- und
Referenzhandbuch



VARCHART XGantt ActiveX Edition

Version 5.2

Benutzerhandbuch

NETRONIC Software GmbH
Pascalstraße 15
52076 Aachen
Deutschland
Tel: +49 (0) 2408 141-0
Fax: +49 (0) 2408 141-33
E-Mail sales@netronic.de
www.netronic.de

© Copyright 2020 NETRONIC Software GmbH
Alle Rechte vorbehalten.

Die Informationen im vorliegenden Benutzerhandbuch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Das illegale Kopieren und Vertreiben dieses Produktes stellt einen Diebstahl geistigen Eigentums dar und wird von NETRONIC Software GmbH strafrechtlich verfolgt.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Buch gezeigten Abbildungen ist nicht zulässig.

Microsoft Windows, Microsoft Explorer, Microsoft Visual Basic und Microsoft Visual Studio sind Warenzeichen der MICROSOFT Corp.

Bearbeitungsstand: 27: April 2020

Inhaltsverzeichnis

1	Einleitung	11
1.1	VARCHART XGantt auf einen Blick	11
1.2	Technische Voraussetzungen	13
1.3	Installation	14
1.4	Lizenzierung	15
1.5	Auslieferung	16
1.6	VARCHART ActiveX in Visual Studio 6.0 oder 7.0 mit Visual C++/MFC	17
1.7	VARCHART ActiveX in HTML-Seiten	20
1.8	Unterstützung und Beratung	26
2	Tutorium	27
2.1	Überblick	27
2.2	Steuerelement im Formular platzieren	29
2.3	Daten bereitstellen	30
2.4	Endtermine berechnen	35
2.5	Arbeitsfreie Zeiten in den Vorgängen kennzeichnen	38
2.6	Interaktionen im Tabellen- und Diagrammbereich	40
2.7	Interaktionen mit Vorgängen	42
2.8	Layer verwenden	44
2.9	Filter verwenden	47
2.10	Histogramme erstellen	51
2.11	Diagramm drucken	65
2.12	Diagramm exportieren	66
2.13	Konfigurationseinstellungen speichern	67
3	Wichtige Konzepte	69
3.1	Boxen	69
3.2	Datentabellen	73

4 Inhaltsverzeichnis

3.3	Datumsangaben und Zeitumstellung	83
3.4	Der Einsatz des Kalenders	85
3.5	Ereignisse	100
3.6	Filter	101
3.7	Grafikformate	103
3.8	Gruppierung	108
3.9	Hierarchische Anordnung	114
3.10	Histogramme	117
3.11	In-Interaction Events	125
3.12	Interaction Events	126
3.13	Knoten (Vorgänge)	136
3.14	Komplettansicht (World View)	139
3.15	Layer	141
3.16	Legendenansicht (Legend View)	145
3.17	Live Update	147
3.18	MultiState-Felder	153
3.19	OLE Drag & Drop	155
3.20	Ressourcenplanung	158
3.21	Schreiben von PDF-Dateien	163
3.22	Sortierung	165
3.23	Sprachanpassung von Textausgaben	172
3.24	Stichtaglinien	173
3.25	Tabelle	178
3.26	Tooltips zur Laufzeit	180
3.27	Unicode-Zeichen	181
3.28	Verbindungen	182
3.29	Verbindungsausssehen	186
3.30	Verschiebehilfen	187
3.31	Zeitrechnung	199
3.32	Zeitskala	202
3.33	Zuordnungstabellen	208
3.34	Verschiebehilfen	213

4	Eigenschaftenseiten und Dialogfelder	215
4.1	Allgemeines	215
4.2	Eigenschaftenseite "Allgemeines"	217
4.3	Eigenschaftenseite "Außenbereich"	230
4.4	Eigenschaftenseite "Knoten"	232
4.5	Eigenschaftenseite "Zusätzliche Ansichten"	241
4.6	Eigenschaftenseite "Layout"	245
4.7	Eigenschaftenseite "Objekte"	250
4.8	Eigenschaftenseite "Verbindungen"	252
4.9	Eigenschaftenseite "Zeitrechnung"	254
4.10	Dialogfeld "Aktualisierungsverhalten verwalten"	256
4.11	Dialogfeld "Aktualisierungsverhalten bearbeiten"	257
4.12	Dialogfeld "Datentabellen verwalten"	259
4.13	Dialogfeld "Balkenaussehen festlegen"	262
4.14	Dialogfeld "Layer bearbeiten"	267
4.15	Dialogfeld "Layerformat bearbeiten"	273
4.16	Dialogfeld "Filter verwalten"	277
4.17	Dialogfeld "Filter bearbeiten"	279
4.18	Dialogfeld "Linienformate verwalten"	283
4.19	Dialogfeld "Linienformat bearbeiten"	285
4.20	Dialogfeld "Gruppierung"	290
4.21	Dialogfeld "Kalendergitter verwalten"	301
4.22	Dialogfeld "Liniengitter verwalten"	304
4.23	Dialogfeld "Zuordnungstabellen verwalten"	307
4.24	Dialogfeld "Zuordnungstabelle bearbeiten"	309
4.25	Dialogfeld "Zuordnung einstellen"	311
4.26	Dialogfeld "Boxen verwalten"	312
4.27	Dialogfeld "Box bearbeiten"	316
4.28	Dialogfeld "Boxformate verwalten"	318
4.29	Dialogfeld "Boxformat bearbeiten"	320
4.30	Dialogfeld "Verbindungsaussehen verwalten"	323
4.31	Dialogfeld "Tabelle festlegen"	327
4.32	Dialogfeld "Tabelle bearbeiten"	329

6 Inhaltsverzeichnis

4.33	Dialogfeld "Tabellenformat bearbeiten"	332
4.34	Dialogfeld "Linienattribute bearbeiten"	337
4.35	Dialogfeld "Musterattribute bearbeiten"	338
4.36	Dialogfeld "Kalender festlegen"	339
4.37	Dialogfeld "Intervalle verwalten" (Kalender)	341
4.38	Dialogfeld "Kalenderprofile verwalten"	343
4.39	Dialogfeld "Intervalle verwalten" für Tagesprofil	345
4.40	Dialogfeld "Intervalle verwalten" für Wochenprofil	347
4.41	Dialogfeld "Intervalle verwalten" für Variables Profil	348
4.42	Dialogfeld "Intervalle verwalten" für Jahresprofil	350
4.43	Dialogfeld "Zeitskala festlegen"	351
4.44	Dialogfeld "Zeitskalenabschnitt bearbeiten"	354
4.45	Dialogfeld "Histogramme verwalten"	361
4.46	Dialogfeld "Histogramm bearbeiten"	363
4.47	Dialogfeld "Datenquelle der Kurve einstellen"	368
4.48	Dialogfeld "Typ des Skalenstreifens einstellen"	369
4.49	Dialogfeld "Stichtaglinien festlegen"	371
4.50	Dialogfeld "Stichtaglinie bearbeiten"	374
4.51	Dialogfeld "Texte, Grafiken und Legende festlegen"	376
4.52	Dialogfeld "Legendenattribute"	380
4.53	Dialogfeld "Lizenzierung"	382
4.54	Dialogfeld "Lizenzinformationen anfordern"	384

5 Benutzerschnittstelle 385

5.1	Übersicht	385
5.2	Navigation in Diagramm und Tabelle	387
5.3	Zoomen	388
5.4	Knoten bzw. Layer markieren	389
5.5	Knoten erzeugen	390
5.6	Knoten verschieben mit der Maus	391
5.7	Knoten verschieben und Dauer ändern mit der Tastatur	393
5.8	Layer verschieben	395
5.9	Anfangs-/Enddatum verändern	396

5.10	Knoten löschen, ausschneiden, kopieren und einfügen	397
5.11	Knotendaten bearbeiten	398
5.12	Verbindungen bearbeiten	400
5.13	Box an Knoten verankern	401
5.14	Gruppendaten bearbeiten	403
5.15	Gruppen kollabieren bzw. expandieren	404
5.16	Gruppen verschieben	405
5.17	Bearbeiten von Feldinhalten in der Tabelle	406
5.18	Breitenverhältnis Tabelle/ Diagramm bearbeiten	407
5.19	Breite der Tabellenspalte bearbeiten	408
5.20	Einfügen von neuen Tabellenzeilen	409
5.21	Zeitskala bearbeiten	410
5.22	Skalierung und Grenzen von Zeitskalenabschnitten bearbeiten	412
5.23	Stichtaglinie verschieben	414
5.24	Seite einrichten	415
5.25	Druckvorschau	421
5.26	Kontextmenü für die Kurve	424
5.27	Kontextmenü für das Diagramm	426
5.28	Kontextmenü für Knoten	431
5.29	Kontextmenü für Verbindungen	433
5.30	Kontextmenü für Gruppen	434
5.31	Kontextmenü für die Zeitskala	436
5.32	Kontextmenü für die Legende	437
5.33	Kontextmenü für Boxen	438

6 Häufig gestellte Fragen 439

6.1	Wie kann das Steuerelement neu lizenziert werden? Was ist bei Problemen mit der Lizenzierung zu tun?	439
6.2	Wie kann erreicht werden, dass eine veränderte ini-Datei des VARCHART ActiveX-Steuerelementes verwendet wird?	440
6.3	Was müssen Benutzer von Borland Delphi beim Upgrade auf eine neue Version von VARCHART XGantt tun?	441
6.4	Was tun, wenn das Steuerelement unerwartet nicht in allen Benutzerkonten eines Rechners funktioniert?	442

8 Inhaltsverzeichnis

6.5	Wie lässt sich in VARCHART XGantt der XP Visual Style aktivieren?	443
6.6	Wie kann die Spreizung der Zeitskala begrenzt werden?	444
6.7	Wie kann beim Klick in die Tabelle der zugehörige Balken in den sichtbaren Bereich gebracht werden?	445
6.8	Wie werden Überschneidungen von Vorgängen einer Gruppe sichtbar?	446
6.9	Wie lässt sich die Reihenfolge der Vorgänge sichern und laden?	447
6.10	Wieso können Knoten u. U. nicht interaktiv erzeugt werden?	448
6.11	Wie lassen sich die Standard-Kontextmenüs abschalten?	449
6.12	Was ist bei Problemen mit dem Drucken zu tun?	450
6.13	Wie lässt sich die Performance verbessern?	451
6.14	Fehlermeldungen	453
6.15	Sind alle Fonts verwendbar?	455

7 API Referenz 457

7.1	Objekttypen	457
7.2	DataObject	460
7.3	DataObjectFiles	467
7.4	VcBorderArea	471
7.5	VcBorderBox	472
7.6	VcBox	480
7.7	VcBoxCollection	496
7.8	VcBoxFormat	503
7.9	VcBoxFormatCollection	508
7.10	VcBoxFormatField	514
7.11	VcCalendar	525
7.12	VcCalendarCollection	533
7.13	VcCalendarGrid	539
7.14	VcCalendarGridCollection	558
7.15	VcCalendarProfile	565
7.16	VcCalendarProfileCollection	568
7.17	VcCurve	574
7.18	VcCurveCollection	606
7.19	VcDataDefinition	613

7.20	VcDataDefinitionTable	614
7.21	VcDataDefinitionTable	619
7.22	VcDataRecord	624
7.23	VcDataRecordCollection	630
7.24	VcDataTable	636
7.25	VcDataTableCollection	639
7.26	VcDataTableField	645
7.27	VcDataTableFieldCollection	652
7.28	VcDateLine	657
7.29	VcDateLineCollection	669
7.30	VcDateLineGrid	676
7.31	VcDateLineGridCollection	689
7.32	VcDefinitionField	696
7.33	VcField	700
7.34	VcFilter	701
7.35	VcFilterCollection	708
7.36	VcFilterSubCondition	714
7.37	VcGantt	719
7.38	VcGroup	964
7.39	VcGroupCollection	974
7.40	VcGroupLevelLayout	978
7.41	VcGroupLevelLayoutCollection	1006
7.42	VcHierarchyLevelLayout	1013
7.43	VcHistogram	1023
7.44	VcHistogramCollection	1032
7.45	VcInfoWindow	1037
7.46	VcInterval	1045
7.47	VcIntervalCollection	1062
7.48	VcLayer	1068
7.49	VcLayerCollection	1109
7.50	VcLayerFormat	1115
7.51	VcLayerFormatField	1118
7.52	VcLegendView	1131
7.53	VcLineFormat	1140

10 Inhaltsverzeichnis

7.54	VcLineFormatCollection	1144
7.55	VcLineFormatField	1150
7.56	VcLink	1164
7.57	VcLinkAppearance	1169
7.58	VcLinkAppearanceCollection	1179
7.59	VcLinkCollection	1185
7.60	VcMap	1189
7.61	VcMapCollection	1195
7.62	VcMapEntry	1203
7.63	VcNode	1213
7.64	VcNodeCollection	1225
7.65	VcNodeLevelLayout	1229
7.66	VcNumericScale	1242
7.67	VcNumericScaleCollection	1256
7.68	VcPrinter	1261
7.69	VcRect	1286
7.70	VcResourceScheduler2	1289
7.71	VcRibbon	1353
7.72	VcScheduler	1367
7.73	VcSection	1375
7.74	VcTable	1381
7.75	VcTableCollection	1386
7.76	VcTableFormat	1390
7.77	VcTableFormatCollection	1396
7.78	VcTableFormatField	1400
7.79	VcTimeScale	1418
7.80	VcTimeScaleCollection	1424
7.81	VcUpdateBehavior	1428
7.82	VcUpdateBehaviorCollection	1431
7.83	VcUpdateBehaviorContext	1438
7.84	VcWorldView	1441

8 Index

1451

1 Einleitung

1.1 VARCHART XGantt auf einen Blick

Gantt-Diagramme sind das geeignete Instrument, um die zeitliche Abfolge von Aufgaben oder Auslastung von Ressourcen zu planen und zu visualisieren. Durch die grafische Darstellung werden Zusammenhänge und Veränderungen im Zeitablauf für alle Beteiligten leichter und schneller erfassbar. Neben dem Einsatz im Projektmanagementbereich haben sich Gantt-Diagramme vor allem in Leitständen der Fertigung und in Systemen der Ressourcenverwaltung und Disposition etabliert.

VARCHART XGantt ist eine interaktive Grafikkomponente, die sich innerhalb kurzer Zeit sehr leicht in eigene Anwendungen integrieren lässt, da die aufwändige Grafikprogrammierung entfällt. Durch die vielfältigen Gestaltungsmöglichkeiten genügt VARCHART XGantt auch sehr individuellen grafischen Anforderungen. Die Druckausgabe überzeugt durch ihre erstklassige Qualität.

> Die Leistungsmerkmale von VARCHART XGantt sind:

- Anlegen, Löschen und Verschieben von Knoten
- Anlegen und Löschen von Verbindungen (vernetztes Gantt-Diagramm)
- Visualisierung der Termine durch Symbole und Balken
- Datengesteuerte Attributierung der Darstellungselemente
- Gruppierung und Sortierung nach beliebigen Kriterien
- Kollabieren und Expandieren von Vorgangsgruppen
- Variabler Aufbau der Zeitskala
- Flexible Gestaltung des Tabellenteils
- Hinzufügen von Stichtaglinien und Rasterlinien
- Stufenloses Zoomen im Diagrammbereich
- Bildschirmfüllende Darstellung von Diagrammausschnitten
- Integrierte Seitenvorschau und Druckausgabe mit Seitenaufteilung
- Austausch der Anwendungsdaten über Dateien oder die Programmierschnittstelle
- Vielfältige Gestaltungsoptionen für Histogramme

12 Einleitung

- Bequemes Einstellen der Eigenschaften zur Entwurfszeit über die Eigenschaftenseiten
- Abwandlung des Standardverhaltens durch Eingriff in Ereignisse
- Umfangreiche Programmierschnittstelle
- Das Ressourcenplanungsmodul von VARCHART XGantt unterstützt das Konzept der interaktiven Entscheidungsfindung. Er vereint die Möglichkeit, Zeitpläne automatisch nach vorgegebenen Strategien zu erstellen und zugleich individuelle Änderungen zu berücksichtigen.

Hinweis: Alle Beispiele in dieser Dokumentation beziehen sich auf Microsoft Visual Basic 6.0.

1.2 Technische Voraussetzungen

Zur Nutzung der VARCHART-ActiveX-Komponente müssen folgende Systemvoraussetzungen erfüllt sein:

- Betriebssystem Microsoft Windows Server 2003, Vista, Windows 7 oder Windows 8.
- Entwicklungsumgebung, die die Einbindung von ActiveX-Steuer-elementen unterstützt wie Visual C++, Visual Basic, Visual Fox Pro, Delphi, Centura, Oracle Forms, Progress, HTML (über Visual Basic Script)
- ca. 50 MB Festplattenspeicher.

1.3 Installation

Starten Sie das Setup-Programm und folgen Sie den Anweisungen.

Während des Installationsvorgangs wird ein Verweis auf die VARCHART ActiveX-Komponente in die Windows Registry eingetragen. Die Registrierung kann auch manuell mit dem Windows-Systemprogramm *regsvr32.exe* durchgeführt werden:

- `c:\windows\system32\regsvr32 "c:\program files\varchart\xgantt\vcgantt.ocx"`

Die angegebenen Pfade sind selbstverständlich abhängig von den Einstellungen auf Ihrem Rechner.

Der Installationsvorgang wird in der Datei *install.log* protokolliert. Damit haben Sie die Kontrolle, welche Dateien wohin kopiert wurden.

Die Deinstallation können Sie über das Windows-Startmenü (**Einstellungen** -> **Systemsteuerung** -> **Software**) vornehmen.

Den Registrierungseintrag können Sie manuell mit dem folgenden Befehl wieder rückgängig machen:

- `c:\windows\system32\regsvr32 -u "c:\program files\varchart\xgantt\vcgantt.ocx"`

Sie können VARCHART XGantt auch bedienerlos installieren. Dazu geben Sie ein:

`start/wait (NameDerSetupDatei).exe /L1031 /s /V"/qn ADDLOCAL=ALL"`

Damit läuft die Installation ohne jegliche Benutzereingabe und ohne Fortschrittmeldungen auf dem Bildschirm ab. Bitte beachten Sie Folgendes:

1. Der aufrufende Prozess, z.B. eine DOS-Box, muss mit Administratorrechten gestartet werden; andernfalls erscheint eine UAC-Abfrage, die eine Benutzereingabe notwendig macht.
2. Sprachparameter: /L1031: Installation in deutsch; /L1033: Installation in englisch; /L2052: Installation in chinesisch
3. Fortschrittmeldungen: /qb: Es erscheinen Fortschrittmeldungen; /qn: Es erscheinen keine Fortschrittmeldungen, d.h. Sie sehen am Bildschirm nichts.
4. Start/wait sollte man benutzen, wenn die Installation über eine Batch-Datei abläuft; andernfalls liefere die Batch-Datei während der Installation parallel weiter.

1.4 Lizenzierung

Zum Lizenzieren von VARCHART XGantt ziehen Sie nun das Steuerelement  von der Toolbox auf die Form.

Öffnen Sie dann die **Eigenschaftenseiten**, indem Sie mit der rechten Maustaste auf das Steuerelement klicken.

Auf der Registerkarte **Allgemeines** gelangen Sie über die Schaltfläche **Lizenzierung...** in den Lizenzierungsdialog.

Die Schaltfläche **Lizenzinformation von NETRONIC anfordern...** führt zu einem weiteren Dialogfeld, das den Eintrag der Benutzerdaten für die Lizenzierung ermöglicht.

Für die Registrierung benötigen wir vier Informationen:

- die Hardware-Identifikation
- die Lizenznummer
- den Mitarbeiternamen
- den Firmennamen

Tragen Sie die erforderlichen Daten in die vorgesehenen Felder ein. Die Lizenznummer "BXnnnn" finden Sie auf dem Lieferschein zu Ihrer Bestellung.

Durch Klick auf die Schaltfläche **E-Mail an NETRONIC senden...** wird eine E-Mail erstellt, die Sie uns nur noch schicken müssen. Sie können aber auch selber eine E-Mail verfassen, in der Sie uns die oben genannten Informationen mitteilen. Richten Sie bitte alle Lizenzierungsanfragen an license@netronic.com.

Danach erhalten Sie von uns umgehend eine passende Lizenzdatei, die Sie bitte im Installationsordner (Ordner mit der Datei vcgantt.ocx) ablegen. Der Lizenzierungsvorgang ist nun abgeschlossen.

1.5 Auslieferung

Wenn Sie Ihre Applikation ausliefern, müssen Sie folgende Dateien mit ausliefern bzw. prüfen, ob diese bereits auf dem Rechner des Kunden vorhanden sind:

VARCHART-XGantt-Dateien:

- *vcgantt.ocx* (Version 5.0)
- *vcgantt.dll*
- *vcpane32u.dll* (Version 5.5)
- *vcprct32u.dll* (Version 5.5)
- *vcwin32u.dll* (Version 5.5)
- *vxcsv32u.dll* (Version 1.320)
- *opsaps.dll* (Version 7.3)

Microsoft Bibliotheken:

- *gdiplus.dll*
- *mfc100u.dll*
- *msvcp100.dll*
- *msvcr100.dll*

Die Datei *vcgantt.ocx* muss mit dem Befehl `regsvr32 vcgantt.ocx` registriert werden.

Die Installation der Bibliotheken *mfc100u.dll*, *msvcp100.dll*, *mfc100u.dll* und *msvcr100.dll* kann entweder direkt durch Kopieren in das Windows-Systemverzeichnis oder über die im Unterverzeichnis **redist** mitgelieferte Setup-Datei *vc_redist_vs2010_x86.exe* erfolgen.

Folgende Dateien dürfen **nicht** an den Endanwender ausgeliefert werden:

- *vcgantt.lic* (enthält Ihre Entwicklerlizenz)
- *vcgantt.chm* (Online-Hilfe für Entwickler)

1.6 VARCHART ActiveX in Visual Studio 6.0 oder 7.0 mit Visual C++/MFC

Um das VARCHART-ActiveX-Steuerelement in Ihr MFC-Projekt einzubinden, gehen Sie folgendermaßen vor:

Visual Studio 6.0:

Im Menü **Projekt** wählen Sie den Menüpunkt **Dem Projekt hinzufügen...** und darin den Untermenüpunkt **Komponenten und Steuerelemente**. In dem nun erscheinenden Dialogfeld wählen Sie unter den registrierten Steuerelementen das NETRONIC VARCHART ActiveX aus und klicken auf die Schaltfläche **Einfügen**. Nach der Sicherheitsabfrage erscheint ein Dialogfeld, in dem Sie alle durch den Wizard erzeugbaren MFC-Hüllklassen in der Listbox deselektieren sollten mit Ausnahme der ersten (dies ist nicht möglich). Klicken Sie dann auf **OK** und verlassen Sie den übergeordneten Dialog mit **Schließen**.

Visual Studio 7.0:

Im Kontextmenü einer Dialogressource wählen Sie den Menüpunkt **Insert ActiveX Control...** und übernehmen das gewünschte ActiveX-Steuerelement in den Dialog. Anschließend erzeugen Sie eine Instanzvariable und einen DDX_CONTROL-Eintrag in der DoDataExchange-Methode entweder manuell oder durch den Wizard per Kontextmenü (Menüpunkt **Insert Variable...**), wobei dann automatisch auch eine MFC-Hüllklasse erzeugt wird. Alternativ kann man auch im ClassView MFC-Hüllklassen (inklusive der für die Unterobjekte) erzeugen lassen, aber dann fehlen die Enum-Definitionen.

Beide Entwicklungsumgebungen bieten also die automatische Erzeugung von MFC-Hüllklassen an. Mittels dieser Klassen ist es möglich, die Methoden und Eigenschaften des ActiveX-Steuerelements zu benutzen wie bei normalen MFC-Objekten. Ohne die Hüllklassen müßten Sie sich intensiver mit OLE-Konventionen auseinandersetzen. Leider sind die erzeugten Klassen nicht zufriedenstellend, denn:

- Die automatisch generierten Dateien enthalten keine Enum-Definitionen (nur Visual Studio 6.0).
- Alle Unterklassen kommen in eigene Dateien, was die gleichzeitige Verwendung mehrerer VARCHART-ActiveX-Steuerelemente unmöglich macht (Visual Studio 6.0), oder es werden gar keine Unterklassen generiert, was deren Benutzung unmöglich macht (Visual Studio 7.0).

18 Einleitung

- Bei API-Updates der Steuerelemente ist das Updaten der Hüllklassendateien nur auf Umwegen zu erreichen. Außerdem verwendet Visual Studio 7.0 nun andere Namenskonventionen, was Änderungen an älteren Projekten notwendig machen würde (Namenspräfixe **get_** und **set_** bei Eigenschaften statt wie bisher **Get** und **Set**).
- Wenn man mehrere VARCHART-ActiveX-Steuerelemente in einem Projekt verwenden will, gibt es Namenskonflikte bei den Unterobjekten.

Daher bietet NETRONIC Software GmbH im Unterverzeichnis MFC des Installationsverzeichnisses des VARCHART-ActiveX-Steuerelements ein eigenes Paar von MFC-Hüllklassendateien namens *xgantt.h* und *xgantt.cpp* an, die alle Hüllklassen inklusive der hilfreichen Enum-Definitionen zusammenfassen.

Dabei wurden alle Definitionen in einen Namespace gestellt, so dass Sie mehrere VARCHART-ActiveX-Steuerelemente in ein Projekt einfügen können, ohne dass es zu Namenskonflikten bei mehrfach vorhandenen Unterobjekten kommt.

Entfernen Sie also die automatisch erzeugten Hüllklassendateien wieder aus Ihrem Projekt, fügen Sie die *cpp*-Datei dem Projekt hinzu und importieren Sie die Headerdatei in die Dialogklasse.

Wenn Sie in einer Klasse nur ein Steuerelement benutzen, reichen folgende zwei Zeilen Code aus:

Code-Beispiel

```
#include "xgantt.h"
using namespace XGantt;
```

Wenn Sie allerdings mehrere VARCHART-ActiveX-Steuerelemente in einer Klasse verwenden, dann müssen Sie zusätzlich zu den jeweiligen zwei Zeilen von oben jedem Unterobjekt, das in mindestens zwei Steuerelementen vorkommt (wie z.B. *CVcNode* oder *CVcTitle*), den Namespace voranstellen. Ein Beispiel für die Deklaration einer Variable für ein Titelobjekt:

Code-Beispiel

```
XGantt::CVcTitle title = VcGantt1.GetTitle();
```

In Ereignisprozeduren bekommen Sie statt Objekten nur *LPDISPATCH*-Zeiger übergeben, die Sie aber über die jeweilige **Attach**-Methode eines Objekts an jenes anbinden können. Dabei ist das **Detach()** am Ende der Bearbeitung des Objekts nicht zu vergessen.

Sollten Sie bereits Projekte mit den generierten Dateien begonnen haben, sollte ein Umstieg nicht schwerfallen, weil die NETRONIC Software GmbH die vom Visual Studio 6.0 generierten Dateien als Grundlage verwendet, die

daher kompatibel sind. Der einzige Unterschied ist die Verwendung von Namespaces, um die Namen von Unterobjekten eindeutig zu machen.

1.7 VARCHART ActiveX in HTML-Seiten

Das VARCHART-ActiveX-Steuerelement kann auch in eine HTML-Seite eingebettet und mittels Skript gesteuert werden. Dabei sind zwei Arten der Einbettung zu unterscheiden: das direkte Einbetten und das Einbetten eines selbst entwickelten ActiveX-Steuerelements, das ein VARCHART-ActiveX-Steuerelement enthält. Ersteres ist für kleine Webapplikationen machbar, für größere Anwendungen sollte man ein eigenes ActiveX-Steuerelement entwickeln, was mit vielen Entwicklungsumgebungen möglich ist.

1.7.1 Einschränkungen

Dabei ergeben sich jedoch ein paar Einschränkungen gegenüber der Programmierung „normaler“ Applikationen:

- Der verwendete Client muss ein Windows-Rechner sein, weil nur dort die ActiveX-Steuerelemente laufen können. Der Server muss diese Voraussetzung nicht erfüllen.
- Bei der direkten Einbettung ist Javascript/JScript (ECMAScript) als Skriptsprache nicht geeignet, weil es keine Parameter by-reference bietet, so dass es unmöglich ist, in Funktionen Werte außer dem Rückgabewert zurückzugeben. Nennenswert wären hier die Methoden **IdentifyObjectAt** und die meisten Ereignisse wie z.B. **OnNodeCreate**. Das nur vom Microsoft Internet Explorer gebotene VBScript dagegen ist geeignet.
- Mozilla-Browser (also auch Firefox und Netscape) und Opera sind allenfalls bei indirekter Einbettung geeignet, und dann auch nur, wenn man ein ActiveX-Plugin verwendet. Dazu gibt es die Lösung vom Mozilla ActiveX Project und das browserunabhängige Plug-In MeadCo Neptune. Beim ersteren gibt es auch übrigens keine stille Installation per CAB-Datei, die beim Internet Explorer der Standard ist.

Man sollte außerdem bedenken, dass die direkte Einbettung und die daraus folgende Steuerung des VARCHART-ActiveX-Steuerelements über ein Skript kein Ersatz für eine "echte" Applikation sein kann. Skripte sind nur für kleine Applikationen geeignet. Sollten Sie eine größere Applikation planen, so empfiehlt sich der Weg über die Entwicklung eines eigenen ActiveX-Steuerelements z.B. mittels Visual Basic 6.0, in dem dann ein oder mehrere VARCHART-ActiveX-Steuerelemente enthalten sein können. In einem Skript kann beispielsweise normalerweise nicht auf die Massenspeicher des Zielrechners zugegriffen werden, was ein ActiveX-Steuerelement durchaus kann (aber eigentlich nicht darf).

1.7.2 Implementierung mit direkter Einbettung

Im Folgenden wird beschrieben, wie man VARCHART-ActiveX-Steuerelemente auf HTML-Seiten im Microsoft Internet Explorer direkt einbetten kann, wobei als Skript-Sprache VBScript verwendet wird.

Das Einbetten in die HTML-Seite geschieht über ein OBJECT-Tag:

Code-Beispiel

```
<OBJECT ID="VcGantt1" WIDTH=700 HEIGHT=350
  CLASSID="CLSID:A4E79A20-C9E1-11CF-BDD7-02608C4302A9"
  CODEBASE="vcgantt.cab#version=4,000,0,0">
</OBJECT>
```

In diesem Befehl werden die Größe und die Class-ID des VARCHART-ActiveX-Steuerelements angegeben. Alle VARCHART-ActiveX-Steuerelemente haben eine eindeutige Class-ID und werden über diese identifiziert, wenn sie sich in der Registry eingetragen haben. Soll ein ActiveX-Steuerelement ohne explizite Installation angezeigt werden, kommt der Codebase-Parameter zum Tragen. Hier wird angegeben, wo eine zugehörige Installationsdatei auf dem Server steht. Die dort anzugebende CAB-Datei wird von NETRONIC Software GmbH bereitgestellt. Als zusätzliche Angabe ist die Versionsnummer anzufügen, damit das Steuerelement immer dann neu heruntergeladen und installiert wird, wenn auf dem Zielrechner keine oder nur eine ältere Version vorhanden ist.

Die CAB-Datei wurde von NETRONIC Software GmbH signiert, so dass der Benutzer beim Internet Explorer eine Meldung über das Zertifikat erhält, wenn der Browser im Begriff ist, das Steuerelement zu installieren. Das VARCHART-ActiveX-Steuerelement ist absichtlich nicht als sicher für die Benutzung in Skriptsprachen ("Safe for Scripting") gekennzeichnet, da mit dem Grafikexport und mit dem Befehl **SaveAs** schreibender Zugriff auf das Dateisystems des Rechners möglich ist. Wenn Sie ein eigenes ActiveX-Steuerelement entwickeln, sollten sie dieses auf jeden Fall als sicher für die Installation und für die Benutzung in Skriptsprachen kennzeichnen (z.B. über den **Package and Deployment Wizard** von Visual Basic 6.0), damit die Verwendung im Internet Explorer reibungslos möglich ist.

Nachdem Sie nun das VARCHART-ActiveX-Steuerelement auf der HTML-Seite eingebaut haben, müssen Sie es noch mit einer eigenen Konfigurationsdatei bestücken, damit es das von Ihnen gewünschte Aussehen erhält. Dies geschieht über ein Stück Skript, in dem die Eigenschaft **ConfigurationName** des VARCHART-ActiveX-Steuerelements auf eine URL gesetzt wird (muss mit **http://** beginnen), die vorzugsweise wieder eine Datei beschreibt, die neben den anderen auf dem Server liegt:

Code-Beispiel

```
VcGantt1.ConfigurationName =  
"http://www.netronic_test.com/xgantt_sample.ini"
```

Dabei müssen Sie beachten, dass vom VARCHART-ActiveX-Steuerelement nicht nur die INI-Datei, sondern auch eine gleichnamige IFD-Datei eingelesen wird, die ebenfalls auf dem Server zu finden sein muss. Die Dateien können Sie erzeugen, indem Sie das VARCHART-ActiveX-Steuerelement in eine Entwicklungsumgebung ziehen und dort über die Eigenschaftenseiten konfigurieren. Anschließend speichern Sie die Konfigurationsdateien über die Eigenschaftenseite **Allgemeines**. Dabei wird auch Ihre Lizenz in der Konfigurationsdatei abgespeichert, die für das Benutzen des ActiveX-Steuerelements wichtig ist.

Eine kleine Webapplikation wird mit den Programmbeispielen ausgeliefert.

Wenn die URL der INI-Datei beim Verfassen der HTML-Seite bekannt ist (d.h. nicht per Skript ermittelt werden muss), dann kann man die Konfigurationsdatei auch per <PARAM>-Tag innerhalb des <OBJECT>-Tags zuweisen. Das hat den Vorteil, dass das ActiveX-Steuerelement direkt beim ersten Anzeigen mit den richtigen Einstellungen wie Farben und Proportionen gezeichnet wird und somit nicht kurzzeitig die Standardeinstellungen auftauchen.

Code-Beispiel

```
<OBJECT CLASSID=...>  
<PARAM NAME="ConfigurationName"  
      VALUE="http://www.netronic.de/mysample.ini">  
</OBJECT>
```

Anmerkung: Frühere Releases der VARCHART-ActiveX-Steuerelemente waren als „Licensed“ gekennzeichnet, so dass in der HTML-Seite der LicenseManager angesprochen werden musste. Dies wurde ersatzlos gestrichen, stört in altem Code aber auch nicht.

1.7.3 Implementierung mit indirekter Einbettung

Wenn Sie ein eigenes ActiveX-Steuerelement schreiben, das ein VARCHART-Steuerelement enthält, gehen Sie für die Einbettung dieses Steuerelements analog zur Anleitung oben vor.

Außerdem müssen Sie für die stille, automatische Installation im Internet Explorer auch eine eigene CAB-Datei erzeugen. Dies ist z.B. mit dem oben schon erwähnten **Package and Deployment Wizard** von Visual Basic 6.0 und mit dem kostenlosen Kommandozeilentool **cabarc** aus dem Microsoft Cabinet SDK möglich. Die CAB-Datei sollte die gleichen Dateien, die auch bei VARCHART-ActiveX-Steuerelementen in der mitgelieferten CAB-Datei

enthalten sind, wieder enthalten. Dazu können Sie den Inhalt der CAB-Datei mittels eines handelsüblichen ZIP-Tools oder auch mittels **cabarc** extrahieren. Für die Steuerung der Installation ist eine INF-Datei zuständig, die man selbst anpassen kann oder die auch vom **Package and Deployment Wizard** erzeugt wird. Sie können zum Erzeugen einer CAB-Datei alternativ das Tool **Iexpress** benutzen, das bei neueren Windows-Versionen mit an Bord ist und aus dem IEAK (Internet Explorer Administration Kit) stammt.

Zudem müssen Sie eigene Steuerelemente und CAB-Dateien signieren, weil aus Sicherheitsgründen nur dann eine Benutzung im Internet Explorer erlaubt ist (dies kann in den Internet-Optionen zwar zonenweise geändert werden, ist aber oft nicht erwünscht). Dies geht nur, indem Sie selbst eine Code-Signatur bei einer CA (Certification Authority, Listen s.u.) erwerben und dann Ihre DLL-, OCX- und schließlich die CAB-Dateien signieren. Dazu ist die Verwendung der kostenlosen Kommandozeilentools **signcode** aus dem Microsoft Platform SDK oder **signtool** aus neueren Microsoft .NET Framework SDKs notwendig.

1.7.4 Problembehandlung

Bei Problemen mit der Ausführung von ActiveX-Steuerelementen im Internet Explorer hilft das kostenlose Tool **Code Download Log Viewer** von Microsoft weiter. Hiermit können Sie sehen, was beim Download nicht funktioniert hat. Ansonsten wird die Verwendung eines Skript-Debuggers empfohlen (z.B. der kostenlose **Microsoft Script Debugger**).

Beim Download von INI- und IFD-Dateien von einem IIS-Webserver ist zu beachten, dass diese Dateitypen dem Webserver bekannt gemacht werden müssen, indem man in den Eigenschaften der „Websites“ im TreeView der Internet-Informationdienste (engl. „Internet Information Services“ = IIS) unter dem Tab **HTTP-Header** den Dialog **Dateitypen** aufruft und dort die Dateitypen INI und IFD dem MIME-Typ **text/plain** zuordnet.

Man sollte nicht unterschätzen, dass meistens auch die Skripte auf der Serverseite debuggt werden müssen, was mit Web-Applikations-Entwicklungsumgebungen möglich ist (z.B. für ASP mit Microsoft FrontPage). Bei serverseitigen Skripten ergibt sich die Schwierigkeit, dass Dinge wie Messageboxen und Log-Dateien zur einfachen Kenntlichmachung von Fehlern im Skript nicht möglich sind.

> **Hilfreiche Dokumente bei Problemen und für technisches Hintergrundwissen:**

OBJECT Tag which specifies component FileVersion and #Version

24 Einleitung

<http://support.microsoft.com/kb/167597>

How To Implement IObjectSafety in Visual Basic 6.0 Controls

<http://support.microsoft.com/kb/182598>

Mozilla ActiveX Project

<http://www.adamlock.com/mozilla/>

MeadCo Neptune

www.meadroid.com/neptune

Microsoft Cabinet SDK

<http://support.microsoft.com/kb/310618>

Microsoft IExpress

www.microsoft.com/technet/prodtechnol/ie/ieak/techinfo/deploy/60/en/iexpress.msp?mfr=true

Code Download Log Viewer (CDLLOGVW)

<http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/samples/internet/browsertools/cdllogvw/default.asp>

Microsoft Script Debugger

www.microsoft.com/downloads/details.aspx?FamilyID=2f465be0-94fd-4569-b3c4-dffdf19ccd99&DisplayLang=en

Code signing

http://msdn.microsoft.com/library/default.asp?url=/workshop/security/authcode/intro_authenticode.asp

Certification authorities

VeriSign: www.verisign.com/developer

VARCHART XGantt ActiveX Edition 5.2

Thawte: www.thawte.com

GeoTrust: www.trustcenter.de

GlobalSign: www.globalsign.net

Signcode tool

<http://msdn.microsoft.com/library/default.asp?url=/workshop/security/authcode/signing.asp>

Signtool tool

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/seccrypto/security/signtool.asp>

1.8 Unterstützung und Beratung

Sie wissen nicht, ob VARCHART XGantt die speziellen Anforderungen Ihres Balkenplans erfüllt?

Sie möchten sich einen Überblick darüber verschaffen, mit welchem Aufwand die eine oder andere Anforderung ihres Gantt-Diagramms zu realisieren ist?

Sie haben gerade damit begonnen, VARCHART XGantt zu testen und möchten eine ganz spezifische Anforderung Ihres Balkenplans programmieren?

Bei der Beantwortung dieser und anderer Fragen helfen wir Ihnen gerne weiter:

NETRONIC Software GmbH

Pascalstraße 15

52076 Aachen

Deutschland

Tel.: +49-2408-141-0

Fax: +49-2408-141-33

E-Mail: support@netronic.de

www.netronic.de

... übrigens, Sie können mit uns einen Wartungs- und Supportservice vereinbaren. Dann kommen Sie in den Genuss sofortiger Hilfe auch über den kostenfreien Support während der 30-tägigen Testphase hinaus. Der Wartungs- und Supportservice umfasst folgende Leistungen:

- Support-Hotline
- Qualifizierte und ausführliche Beratung in allen Anwendungsfragen
- Beseitigung von möglichen Fehlern in der gelieferten Software
- Upgrade auf ein neues Release von VARCHART XGantt für Entwicklungs- und Laufzeitversionen

Auf Wunsch können wir Ihnen auch Schulungskurse und Workshops (in unserem Hause oder bei Ihnen vor Ort) anbieten.

2 Tutorium

2.1 Überblick

In diesem Kapitel machen wir Sie mit den Grundlagen und Grundbegriffen von VARCHART XGantt vertraut, die notwendig sind, um die Balkenplan-Komponente in eigene Anwendungen integrieren zu können.

Wir werden Schritt für Schritt die bedeutsamen Aspekte von VARCHART XGantt für die Anwendungsentwicklung erläutern und auf die vielfältigen Gestaltungsmöglichkeiten näher eingehen. Aus diesem Grund empfiehlt es sich, dieses Kapitel sequenziell durchzuarbeiten. Die anderen Kapitel des Handbuchs sind eher zum Nachschlagen gedacht:

- **Eigenschaftenseiten und Dialogfelder**

Hier finden Sie umfassende Informationen zu den Eigenschaftenseiten und Dialogen, die Ihnen zur Entwurfszeit erlauben, VARCHART XGantt nach Wunsch zu konfigurieren, ohne dass Sie dafür Programm-Code schreiben müssen.

- **Elemente der Benutzerschnittstelle**

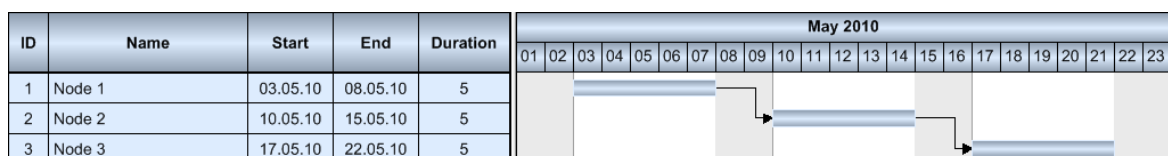
In diesem Kapitel werden die bereits im Diagramm vorhandenen Interaktionen beschrieben. Details der Benutzerschnittstelle können individuell angepasst bzw. abgewandelt werden.

- **API-Referenz**

In diesem Kapitel finden Sie ausführliche Informationen zu allen Objekten, Eigenschaften, Methoden und Ereignissen, die Ihnen VARCHART XGantt bietet.

Als Entwicklungsumgebung für die Beispiele verwenden wir Visual Basic 6.0.

Unser erstes Programmierbeispiel wird Sie zu folgendem Ergebnis führen:



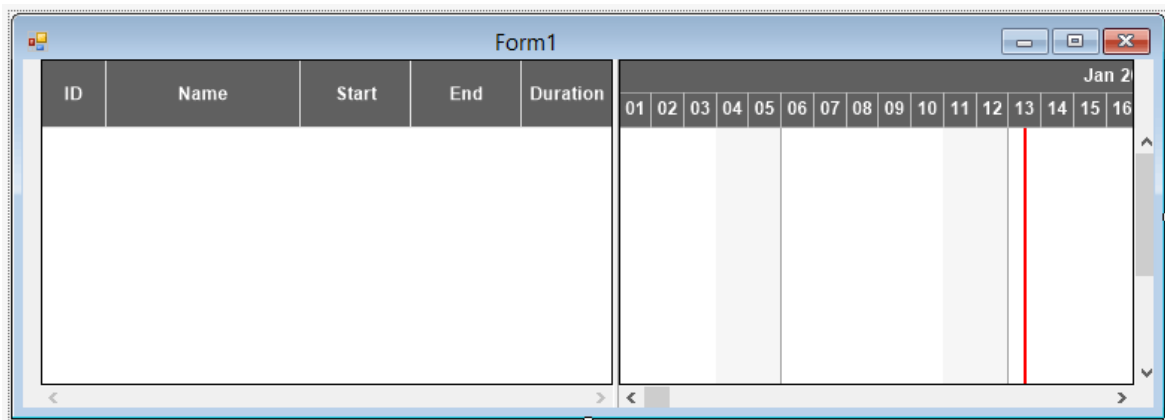
Sie finden dieses Einstiegsbeispiel im Ordner **Programs\UserGuide-Samples\XGantt_Tutorial01_App**

Anhand dieser Anwendung werden Sie die in VARCHART XGantt bereits eingebauten Interaktionen näher kennenlernen.

2.2 Steuerelement im Formular platzieren

Um VARCHART XGantt in ein Formular einzufügen, wählen Sie bitte das Steuerelement  in der Werkzeugleiste aus und ziehen im Formular an der gewünschten Stelle mit der Maus einen Rahmen auf. Um das Beispiel-Programm in der Entwicklungsumgebung nachvollziehen zu können, laden Sie bitte nun die von uns im Verzeichnis **UserGuideSamples\VB6** bereitgestellte Konfigurationsdatei "ActiveX-Beispiel.ini". In dieser Datei sind die gleichen Bezeichnungen, Farben und Maße eingestellt, die auch in den folgenden Abschnitten verwendet werden. Wie Sie eine Konfigurationsdatei importieren können, ist am Ende des Tutoriums im Kapitel "Konfigurationsdatei speichern und laden" beschrieben.

Nachdem Sie das Programm gestartet haben, sollten Sie folgendes Ergebnis sehen:



Wenn sich das Steuerelement zur Laufzeit mit seinem rechten und unteren Rand stets an die Fenstergröße anpassen soll, müssen sowohl im Load- als auch im Resize-Ereignis des Formulars die folgenden Programmzeilen enthalten sein:

Code-Beispiel

```
Private Sub Form_Load()
    VcGantt1.Width = ScaleWidth - VcGantt1.Left
    VcGantt1.Height = ScaleHeight - VcGantt1.Top
End Sub

Private Sub Form_Resize()
    VcGantt1.Width = ScaleWidth - VcGantt1.Left
    VcGantt1.Height = ScaleHeight - VcGantt1.Top
End Sub
```

Hinweis: Das eingefügte Steuerelement heißt hier und in allen Code-Beispielen **VcGantt1**. Der Name wird von der Entwicklungsumgebung vorgegeben, kann jedoch bei Bedarf geändert werden.

2.3 Daten bereitstellen

Damit Vorgänge und Verbindungen dargestellt werden können, muss VARCHART XGantt mit Daten versorgt werden. Die dazu notwendige Kommunikation erfolgt standardmäßig über zwei Tabellen:

1. Maindata
2. Relations

Durch das Laden einer Datendatei mit dem Namen **Beispiel.ini** werden diese Tabellen schon mit folgenden Daten gefüllt:

Felder der Maindata-Tabelle:

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat
0	ID	True	Integer	
1	Name	False	Alphanumerisch	
2	Start	False	Datum/Zeit	DD.MM.YYYY
3	Ende	False	Datum/Zeit	DD.MM.YYYY
4	Duration	False	Integer	

Felder der Relations:

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat
0	Link ID	True	Alphanumerisch	
1	Predecessor Node ID	Alphanumerisch		
2	Successor Node ID	Alphanumerisch		

Alle weiteren Felder müssen Sie selbst definieren. Dies kann zur Designzeit über den Dialog **Datentabelle bearbeiten** geschehen oder zur Laufzeit über die Methode **Add(...)** des Objekts **VcDataTableFieldCollection** erfolgen.

Sollten Sie mehr Tabellen benötigen als die beiden standardmäßig vorhandenen, können Sie diese über die Eigenschaftenseite **Datentabellen verwalten** anlegen, nachdem Sie auf der Eigenschaftenseite **Allgemeines** die Option **Erweiterte Datentabellen zulassen** aktiviert haben. Die Felder für diese Tabellen können Sie ebenfalls im Dialog **Datentabelle bearbeiten** anlegen bzw. verändern.

Datentabellen verwalten

Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata		☐	
Relations		☐	

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt
0	ID	☒	Integer		☐	☐
1	Name	☐	String		☒	☐
2	Start	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
3	End	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
4	Duration	☐	Integer		☒	☐
5	Completion	☐	Integer		☒	☐
6	Group Level 1	☐	String		☒	☐
7	Group Level 2	☐	String		☒	☐
8	Release Date	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
9	Due Date	☐	Datum/Zeit	DD.MM.YYYY	☒	☐

OK Abbrechen Übernehmen Hilfe

Mit der Methode **DataRecordByID()** von **VcDataRecordCollection** lässt sich ein gesuchtes Objekt anhand des Primärschlüssels schnell finden.

Damit in unserem Einstiegsbeispiel Vorgänge und Verbindungen im Gantt-Diagramm zu sehen sind, müssen zunächst einige Datensätze in die Datentabellen eingefügt werden.

Dies geschieht mit der Methode **Add(...)** des Objekttyps **VcDataRecordCollection**. Mit **EndLoading** wird die Dateneingabe abgeschlossen und die entsprechende Grafik kann aufgebaut werden. Deshalb ergänzen Sie bitte die nachfolgenden Programmzeilen im **Load**-Ereignis des Formulars.

Code-Beispiel

```
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
dataRecCltn.Add "1;Node 1;07.05.2007;;5"
dataRecCltn.Add "2;Node 2;14.05.2007;;5"
```

32 Daten bereitstellen

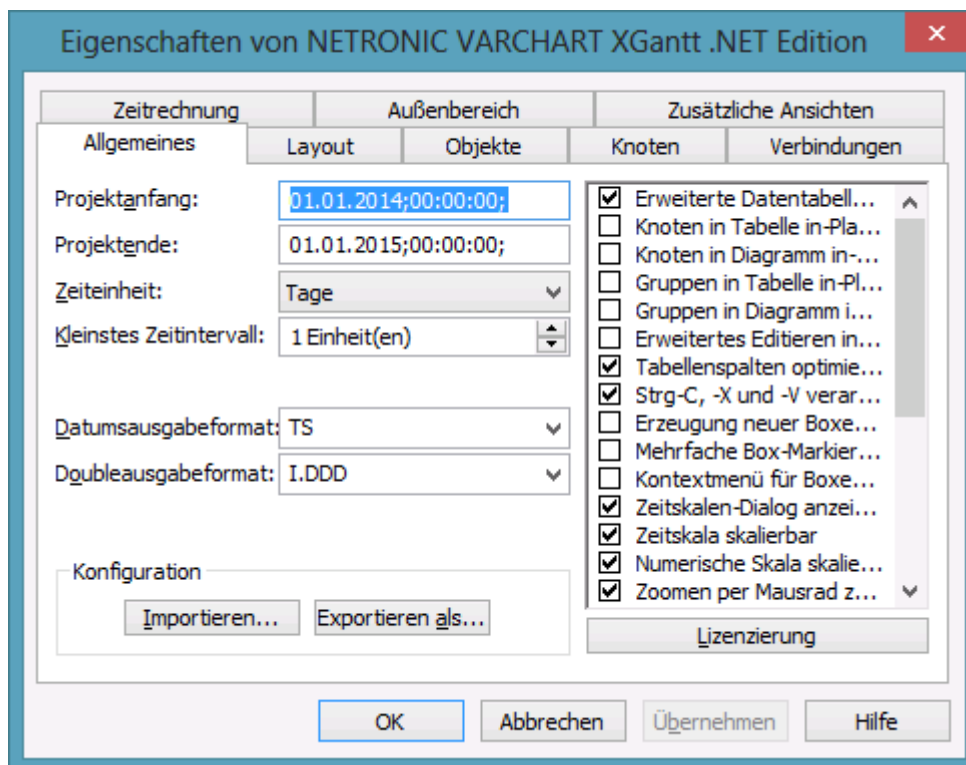
```
dataRecCltn.Add "3;Node 3;21.05.2007;;5"

Set dataTable =
VcGantt1.DataTableCollection.DataTableByName("Relations")
Set dataRecCltn = dataTable.DataRecordCollection
dataRecCltn.Add "1;1;2"
dataRecCltn.Add "2;2;3"

VcGantt1.EndLoading
```

Die Werte innerhalb eines Datensatzes werden durch Semikolon getrennt. Die Reihenfolge der Felder muss der Reihenfolge der Felder in der Datendefinition entsprechen. Jeder neue Datensatz erfordert eine eindeutige, nicht leere Identifikation. Die Datumsangabe im Datensatz muss mit der Datumsformat-Vereinbarung in der Datendefinitionstabelle übereinstimmen. Die Interpretation der Dauer hängt von der Einstellung **Zeiteinheit** auf der Eigenschaftenseite **Allgemeines** ab und ist auf **Tage** voreingestellt.

Das Format der Datumsausgabe wird einheitlich für die Tabelle und alle Dialoge auf der Eigenschaftenseite **Allgemeines** durch das **Datumsausgabeformat** eingestellt.



> Daten aus einer CSV-Datei importieren

Alternativ können Sie die Daten auch aus einer so genannten CSV-Datei laden. Der Aufbau der Datei muss folgendem Schema entsprechen:

Code-Beispiel

```
1;Node 1;07.05.2007;;5
```

```
2;Node 2;14.05.2007;;5
3;Node 3;21.05.2007;;5
****
1;1;2
2;2;3
```

Jeder Datensatz steht in einer eigenen Zeile. Die Zeileninhalte entsprechen dem Übergabeparameter der Methode **Add(...)** des Objekttyps **VcDataRecordCollection**.

Zuerst werden alle Datensätze der Maindata aufgeführt und anschließend die Datensätze der Relations. Die Datensatzgruppen werden mit

**** Tabellenname **** eingeleitet.

Wenn Sie eine solche Datei beispielsweise unter dem Namen **intro.csv** abgespeichert haben, können Sie die Daten mit folgendem Befehl in VARCHART XGantt importieren:

Code-Beispiel

```
VcGantt1.Open("c:\intro.csv")
```

> Darstellungszeitraum festlegen

Bislang würden Sie noch keine Vorgänge sehen, weil die Zeitskala noch nicht auf den Zeitraum eingestellt ist, in dem die Vorgänge liegen. Der darstellbare Zeitskalenbereich kann entweder durch die Eigenschaften **TimeScaleStart** und **TimeScaleEnd** vorgegeben werden oder durch die Methode **OptimizeTimeScaleStartEnd(...)** des Objekts **VcGantt** aus den Daten ermittelt werden.

Code-Beispiel

```
VcGantt1.TimeScaleEnd = DateSerial(2008, 1, 1)
VcGantt1.TimeScaleStart = DateSerial(2007, 5, 4)
```

Hier sind im Überblick noch einmal alle Programmzeilen zusammengefasst, die wir für unser Einstiegsbeispiel benötigen.

Code-Beispiel

```
Private Sub Form_Load()
    VcGantt1.Width = ScaleWidth - VcGantt1.Left
    VcGantt1.Height = ScaleHeight - VcGantt1.Top

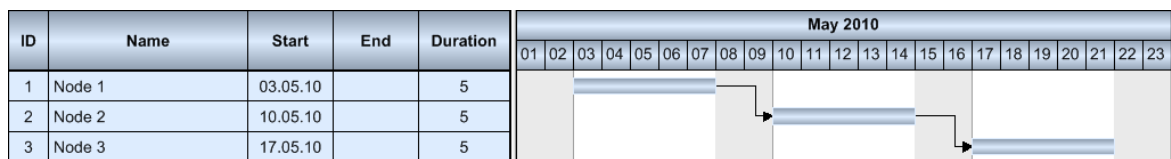
    Set dataTable =
VcGantt1.DataTableCollection.DataTableByName("Maindata")
    Set dataRecCltn = dataTable.DataRecordCollection
    dataRecCltn.Add "1;Node 1;07.05.2007;;5"
    dataRecCltn.Add "2;Node 2;14.05.2007;;5"
    dataRecCltn.Add "3;Node 3;21.05.2007;;5"

    Set dataTable =
VcGantt1.DataTableCollection.DataTableByName("Relations")
    Set dataRecCltn = dataTable.DataRecordCollection
    dataRecCltn.Add "1;1;2"
```

34 Daten bereitstellen

```
dataRecCltn.Add "2;2;3"  
  
    VcGantt1.EndLoading  
  
    VcGantt1.OptimizeTimeScaleStartEnd (3)  
End Sub  
  
Private Sub Form_Resize()  
    VcGantt1.Width = ScaleWidth - VcGantt1.Left  
    VcGantt1.Height = ScaleHeight - VcGantt1.Top  
End Sub
```

Wenn Sie nun das Programm ausführen, sollte das Ergebnis der Abbildung entsprechen.



2.4 Endtermine berechnen

Die Tabellenspalte mit den Endterminen ist noch leer. Das Ende eines Vorgangs kann aus **Start** und **Dauer** mit Hilfe des in VARCHART XGantt enthaltenen Kalenders berechnet werden.

Der Kalender ist so eingestellt, dass die Wochentage (Montag bis Freitag) als aktive Zeiten, die Wochenenden (Samstag und Sonntag) als nicht aktive Zeiten zählen.

Im Diagramm sind die nicht aktiven Zeiten an den grau hinterlegten Bereichen zu erkennen. Abgeschaltet werden kann der Kalender auf der Eigenschaftenseite **Knoten**, indem die Option **den Knoten Kalender zuweisen** deaktiviert wird.

Der Unterschied in der Berechnung mit und ohne Kalender ist folgender:

Ein Vorgang, der am Freitag beginnt und 3 Tage dauert, wird bei aktiviertem Kalender am Dienstag enden. Ohne Kalender wird dieser Vorgang bereits am Sonntag enden.

Die Berechnung der Endtermine wird über die Methode **AddDuration(...)** des Objektes **VcCalendar** ausgeführt. Dazu werden von jedem Vorgang **Start** und **Dauer** benötigt. Der Zugriff auf die entsprechenden Datenfelder erfolgt über den Index. Nachdem der Endtermin mit der Methode **DataField(..)** gesetzt wurde, muss noch die Methode **UpdateNode** von **VcNode** aufgerufen werden, damit die Datenänderung in der Darstellung sichtbar wird.

Code-Beispiel

```
Dim tmpCal As VcCalendar
Dim tmpDate As Date
Set tmpCal = VcGantt1.CalendarCollection.Active
tmpDate = tmpCal.AddDuration(node.DataField(2), node.DataField(4))
node.DataField(3) = tmpDate
node.UpdateNode
```

Durch Mausinteraktionen festgelegte oder veränderte Vorgänge starten und enden immer in einer aktiven Zeit.

ID	Name	Start	End	Duration									
					01	02	03	04	05	06	07	08	09
1	Node 1	03.05.10	08.05.10	5									

Die über die API oder über die Editierdialoge gesetzten Termine können jedoch auch in nicht-aktiven Zeiten liegen.

ID	Name	Start	End	Duration									
					01	02	03	04	05	06	07	08	09
1	Node 1	03.05.10	08.05.10	5									

Durch Berechnung ermittelte Termine liegen dagegen immer in einer aktiven Zeit. Wenn auch für die über die API eingefügten Vorgänge sichergestellt werden soll, dass alle Starttermine in einer aktiven Zeit liegen, dann muss der Starttermin aus dem Endtermin und der Dauer berechnet werden.

Code-Beispiel

```
tmpDate = tmpCal.AddDuration(node.DataField(3), _
                             (-1) * node.DataField(4))
node.DataField(2) = tmpDate
```

Zur Wahrung der Konsistenz sollten fehlende oder negative Dauern als unzulässige Werte betrachtet und auf 0 zurückgesetzt werden. Bei einem fehlendem Starttermin kann keine Berechnung des Endtermins erfolgen. Die notwendigen Codezeilen wurde in eine eigene Methode mit dem Namen **SetNodeEndDate(...)** zusammengefasst.

Code-Beispiel

```
Private Sub SetNodeEndDate(ByVal node As VcNode)
    'Leere oder negative Dauer vermeiden
    If node.DataField(4) = "" Or node.DataField(4) < 0 Then
        node.DataField(4) = "0"
    End If
    'Wenn das Startdatum leer ist, sollte das Enddatum
    'ebenfalls leer sein
    If node.DataField(2) = "31.12.1899 00:00:00" Then
        node.DataField(3) = ""
    Else
        'Voraussetzung: Property Page Knoten
        '"Den Knoten Kalender zuweisen" muss true sein
        Dim tmpCal As VcCalendar
        Dim tmpDate As Date
        Set tmpCal = VcGantt1.CalendarCollection.Active
        tmpDate = tmpCal.AddDuration(node.DataField(2), _
                                     node.DataField(4))
        node.DataField(3) = tmpDate
        'Startdatum nur in aktiven Zeiten
        tmpDate = tmpCal.AddDuration(node.DataField(3), _
                                     (-1) * node.DataField(4))
        node.DataField(2) = tmpDate
        node.UpdateNode
    End If
End Sub
```

Die Terminberechnung muss in folgenden Situationen ausgeführt werden:

1. Nach dem Laden der Vorgänge

2. Nach Ändern der Termine oder Dauern über den Dateneditierdialog oder Inplace-Editor
3. Nach dem Verändern der Datenwerte eines Vorgangs per API

Die Terminberechnung muss jedoch nicht bei Vorgangsänderungen über Mausinteraktionen angestoßen werden, denn in diesem Fall erfolgt intern eine automatische Berechnung.

Über die Eigenschaft **NodeCollection** von **VcGantt** kann eine Berechnungsschleife über alle Knoten gebildet werden. Dieser Code wird nach dem Laden der Vorgänge am Ende des Ereignisses **Form1_Load(...)** hinzugefügt.

Code-Beispiel

```
'Enddatum aller Knoten berechnen
Dim node As VcNode
For Each node In VcGantt1.NodeCollection
    SetNodeEndDate node
Next
```

Durch den Benutzer verursachte Änderungen der Daten können über das Ereignis **OnNodeModifyComplete** erfasst werden. Der eingefügte Methodenaufruf führt die Berechnung des Endtermins aus.

Code-Beispiel

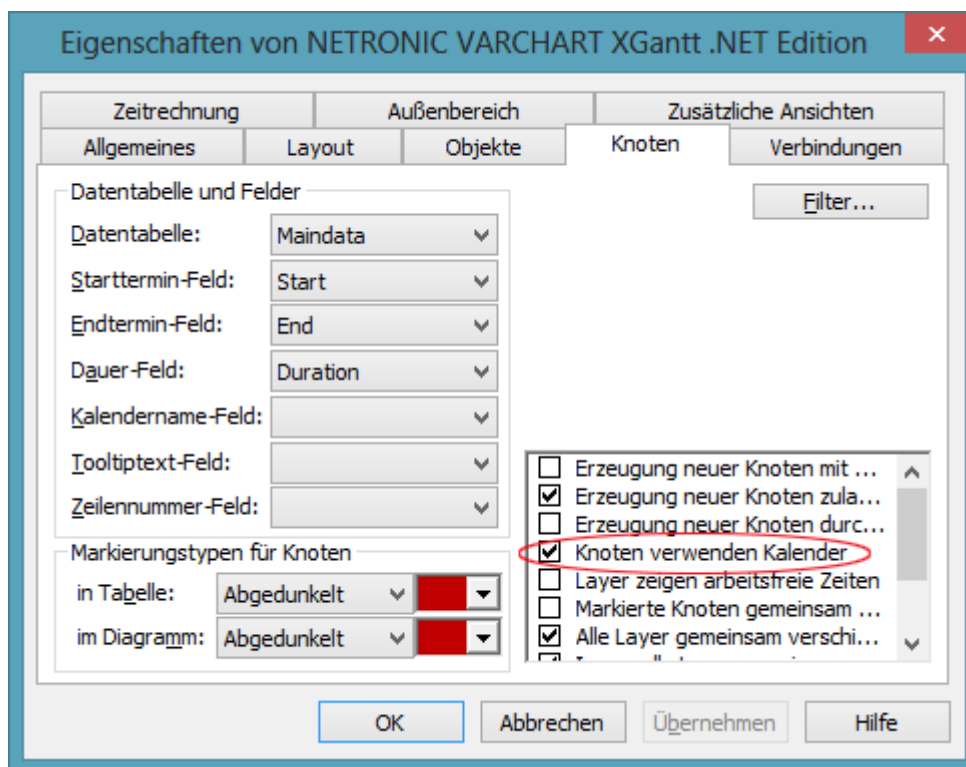
```
Private Sub VcGantt1_OnNodeModifyComplete _
    (ByVal node As VcGanttLib.VcNode, _
    ByVal isLastNodeInSeries As Boolean)
    SetNodeEndDate node
End Sub
```

Bei Datenwertänderungen über die API muss explizit ein **SetNodeEndDate(...)** aufgerufen werden.

2.5 Arbeitsfreie Zeiten in den Vorgängen kennzeichnen

Die Unterbrechung der Vorgänge durch arbeitsfreie Zeiten kann mit der Darstellungsoption **Arbeitsfreie Zeiten anzeigen** sichtbar gemacht werden. Die Option wird nur wirksam, wenn die Vorgänge von einem Kalender abhängen. Dies wird durch die Option **Den Knoten Kalender zuweisen** bestimmt.

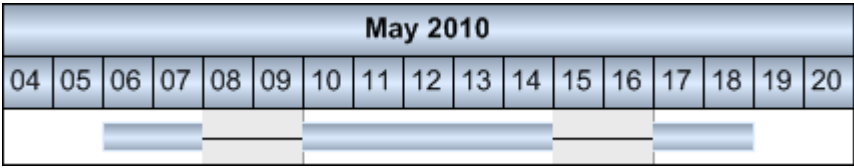
Die Einstellungen können zur Laufzeit oder Designzeit erfolgen. Auf den Eigenschaftenseiten ist die Einstellung auf der Eigenschaftenseite **Knoten** unten rechts zu finden.



Zur Laufzeit erfolgt die Setzung über die Eigenschaft **ShowNonWorkInterval** des Objekts VcGantt.



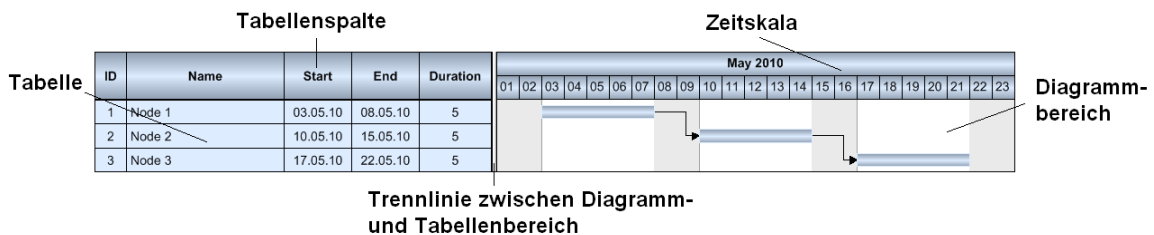
ShowNonWorkInterval = false



ShowNonWorkInterval = true

2.6 Interaktionen im Tabellen- und Diagrammbereich

In diesem und dem nachfolgenden Unterkapitel werden Sie in einem Kurzüberblick die wichtigsten Interaktionen im Gantt-Diagramm kennenlernen. Weiterführende Informationen finden Sie in Kapitel 5, **Benutzerschnittstelle**.



> Breitenverhältnis von Tabellen- zu Diagrammbereich verändern

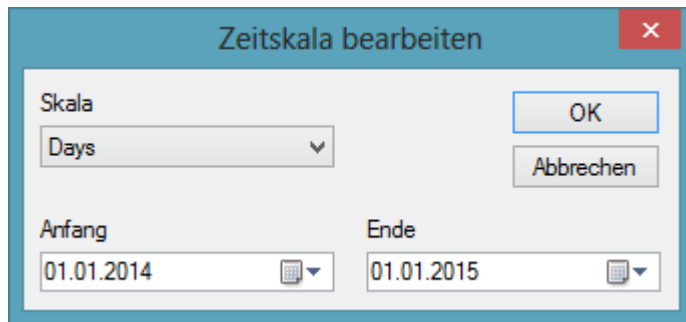
Die Aufteilung zwischen Tabellen- und Diagrammbereich lässt sich durch Ziehen der grauen, vertikalen Trennlinie (Sash bzw. Splitterbar) verändern. Auf den Eigenschaftenseiten unter **Layout** lässt sich das Verhältnis über **Breitenverhältnis Tabelle/Diagramm** voreinstellen.

> Breite der Tabellenspalten ändern

Durch Ziehen der rechten vertikalen Trennlinie im Bereich der Tabellenüberschriften lässt sich die Breite einer Spalte verändern. Mit einem Doppelklick auf diese Linie wird die Breite der Spalte automatisch an die Länge des darin enthaltenen Textes angepasst. Die automatische Spaltenbreitenanpassung kann auf der Eigenschaftenseite **Allgemeines** über die Option **Tabellenspalten optimierbar** in der Optionsliste rechts deaktiviert werden.

> Anfang und Ende der Zeitskala festlegen

Durch einen Doppelklick auf die Zeitskala öffnet sich das Dialogfeld zum Verändern von Anfang und Ende der Zeitskala. Auf der Eigenschaftenseite **Allgemeines** kann dieses Verhalten durch die Option **Zeitskalen-Dialog anzeigen** in der Optionsliste rechts deaktiviert werden.



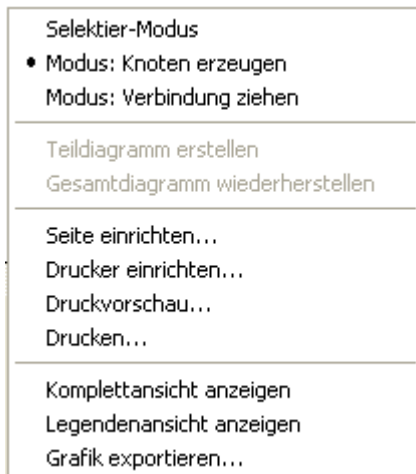
> Zeitskala skalieren

Durch Ziehen der Maus im Zeitskalenbereich nach rechts oder links lässt sich die Einheitenbreite der jeweiligen Zeitskala vergrößern bzw. verkleinern. Auf der Eigenschaftenseite **Allgemeines** unter **Zeitskala skalierbar** kann dieses Verhalten deaktiviert werden.

2.7 Interaktionen mit Vorgängen

> Neuen Vorgang anlegen

Um einen neuen Vorgang anlegen zu können, müssen Sie zunächst über das Kontextmenü des Diagrammbereichs (rechte Maustaste) in den Modus **Knoten erzeugen** wechseln.



Nach der Wahl der Menü-Option nimmt der Mauszeiger die Form eines kleinen Fadenkreuzes an. Ziehen Sie nun mit gedrückter linker Maustaste an der gewünschten Position einen neuen Vorgang auf. Schalten Sie nach dem Anlegen des Vorgangs wieder über das Kontextmenü zurück in den **Selektier-Modus**.

In den Erzeugemodus kann vom Programm über das Ereignis **OnNodeCreateCompleteEx()** eingegriffen werden. Dies ist zum Beispiel sinnvoll, wenn Datenwerte vorzubsetzen sind.

Code-Beispiel

```
Private Sub VcGantt1_OnNodeCreateCompleteEx _
    (ByVal node As VcGanttLib.VcNode, _
    ByVal creationType As VcGanttLib.CreationTypeEnum, _
    ByVal isLastNodeInSeries As Boolean)

    node.DataField(1) = "Node " + node.DataField(0)
    node.UpdateNode
End Sub
```

Der hier gezeigte Code verändert den Inhalt des Datenfeldes **Name**; er hängt an den Begriff **Node** den aktuellen Inhalt des Feldes **ID** an.

> Dauer eines Vorgangs ändern

Fahren Sie im Selektiermodus mit dem Mauszeiger von innen an den rechten oder linken Rand des Vorgangs heran. Der Mauszeiger ändert seine Form zu

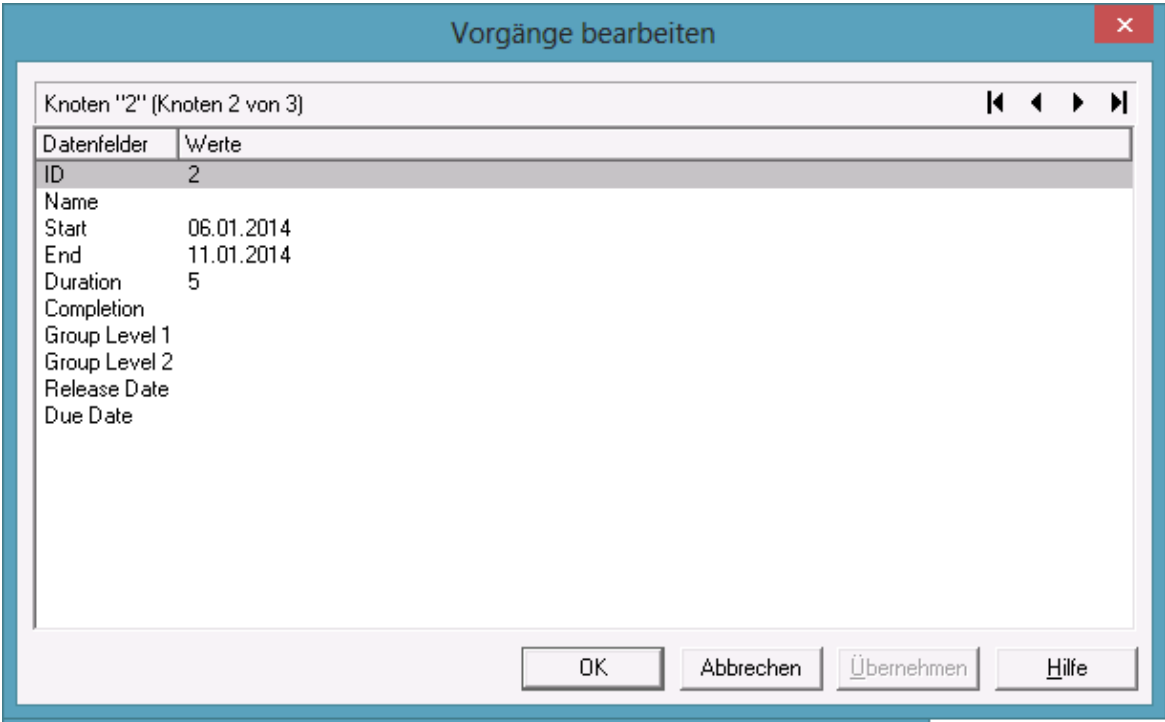
einer senkrechten Linie mit Pfeil. Durch Ziehen nach links oder rechts verlängern bzw. verkürzen Sie den Vorgang.

> Vorgang verschieben

Zum Verschieben eines Vorganges fahren Sie im Selektiermodus mit dem Mauszeiger auf den zentralen Bereich des Vorgangs. Der Mauszeiger wird zu einem kleinen Quadrat mit vier Pfeilen. Nun lässt sich der Vorgang durch Ziehen mit der Maus an eine beliebige Position verschieben. Mehrere Vorgänge können gleichzeitig verschoben werden, wenn auf der Eigenschaftenseite **Knoten** die Option **Markierte Knoten gemeinsam verschieben** in der Optionsliste rechts aktiviert ist.

> Daten eines Vorgangs bearbeiten

Durch einen Doppelklick auf einen Vorgang oder auf den zugehörigen Tabelleneintrag gelangen Sie in das Dialogfeld **Vorgänge bearbeiten**.



Datenfelder	Werte
ID	2
Name	
Start	06.01.2014
End	11.01.2014
Duration	5
Completion	
Group Level 1	
Group Level 2	
Release Date	
Due Date	

> Vorgang löschen

Mit der **Entf**-Taste oder über **Knoten löschen** im Kontextmenü des Vorgangs können markierte Vorgänge gelöscht werden.

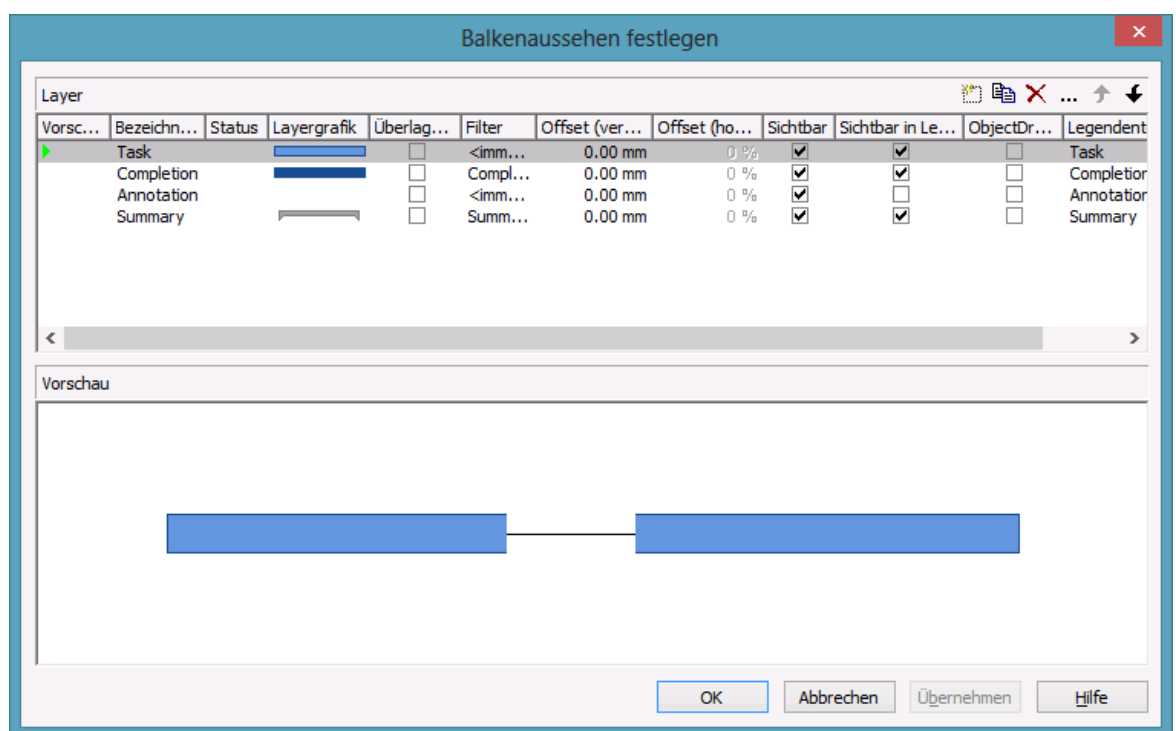
Ein Vorgang wird markiert, in dem Sie ihn direkt oder die entsprechende Tabellenzeile anklicken. Bei gedrückter Strg-Taste wird der Vorgang zu der bestehenden Auswahl hinzugefügt.

2.8 Layer verwenden

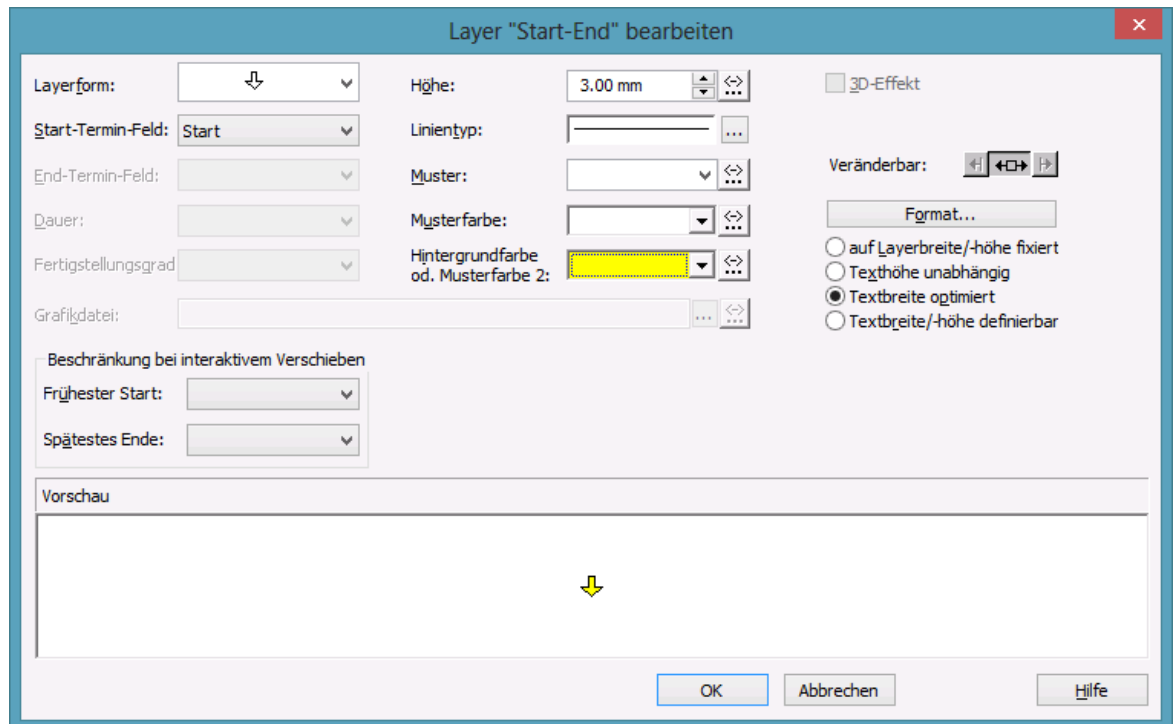
Ein Layer beschreibt die grafische Umsetzung eines Terminpaares. Jedes Terminpaar kann durch mehrere Layer abgebildet werden, die schichtweise der Reihe nach übereinander liegen.

Wir möchten in unserem Beispiel einen zweiten Layer mit anderem Aussehen vereinbaren.

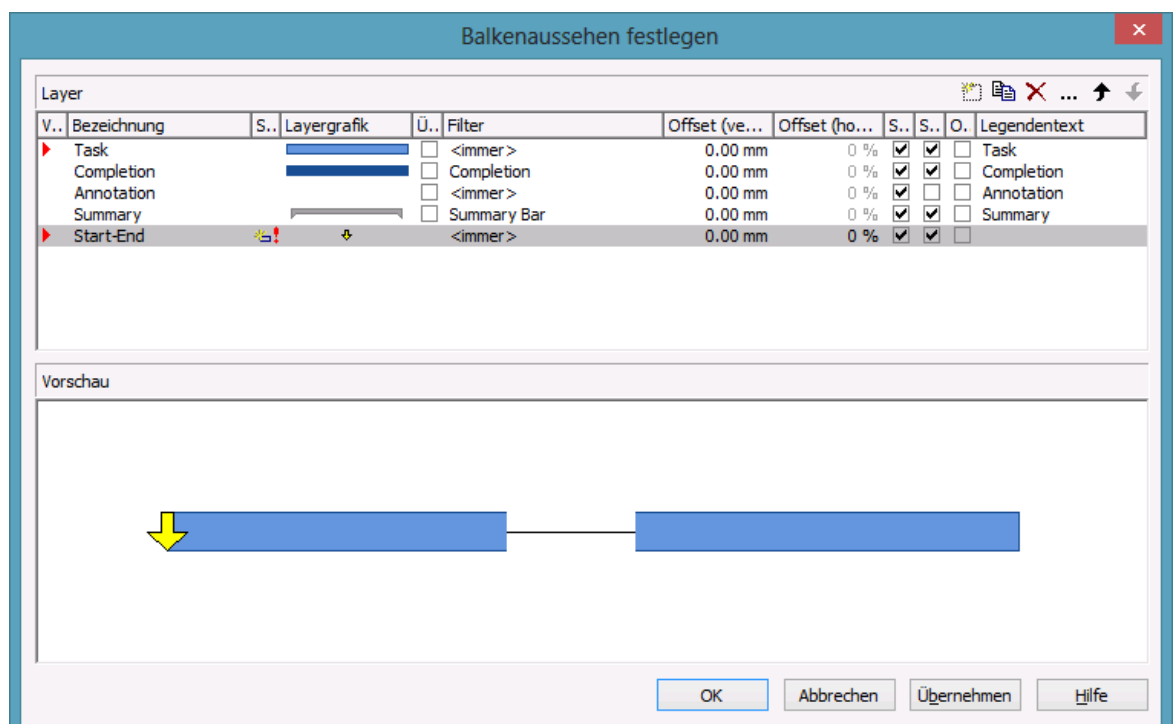
1. Wählen Sie **Layer...** auf der Eigenschaftenseite **Objekte**. Sie gelangen in den Dialog **Balkenaussehen festlegen**. Der Layer mit dem Namen **Task** ist bereits vordefiniert.



2. Kopieren Sie die Layerdefinition "Task" mit Hilfe der Schaltfläche .
3. Bearbeiten Sie nun den neuen Layer **NeuerLayer**. Nennen Sie ihn **Start-End** und öffnen dann den Dialog **Edit Layer** durch Klick auf .
4. Ändern Sie die **Hintergrundfarbe** auf gelb und die **Form** auf Pfeilspitze nach unten.

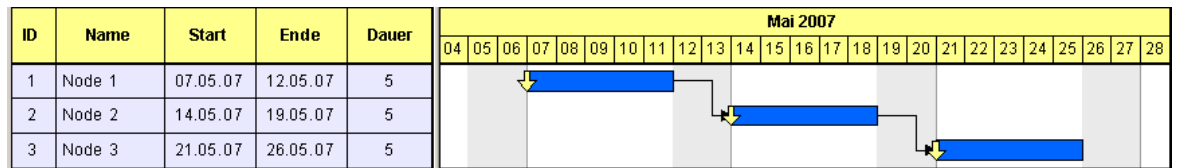


5. Mit der Schaltfläche OK gelangen Sie zurück in den Dialog **Balkenaussehen festlegen**.
6. Die Überlagerung der Schichten können Sie bereits in der Vorschau sehen, wenn Sie in der Spalte **Vorschau** in jeder Zeile einmal klicken, so dass jeweils ein rotes Dreieck erscheint.



7. Die geänderte Definition führt im Beispielpogramm zu folgendem Ergebnis:

46 Layer verwenden



2.9 Filter verwenden

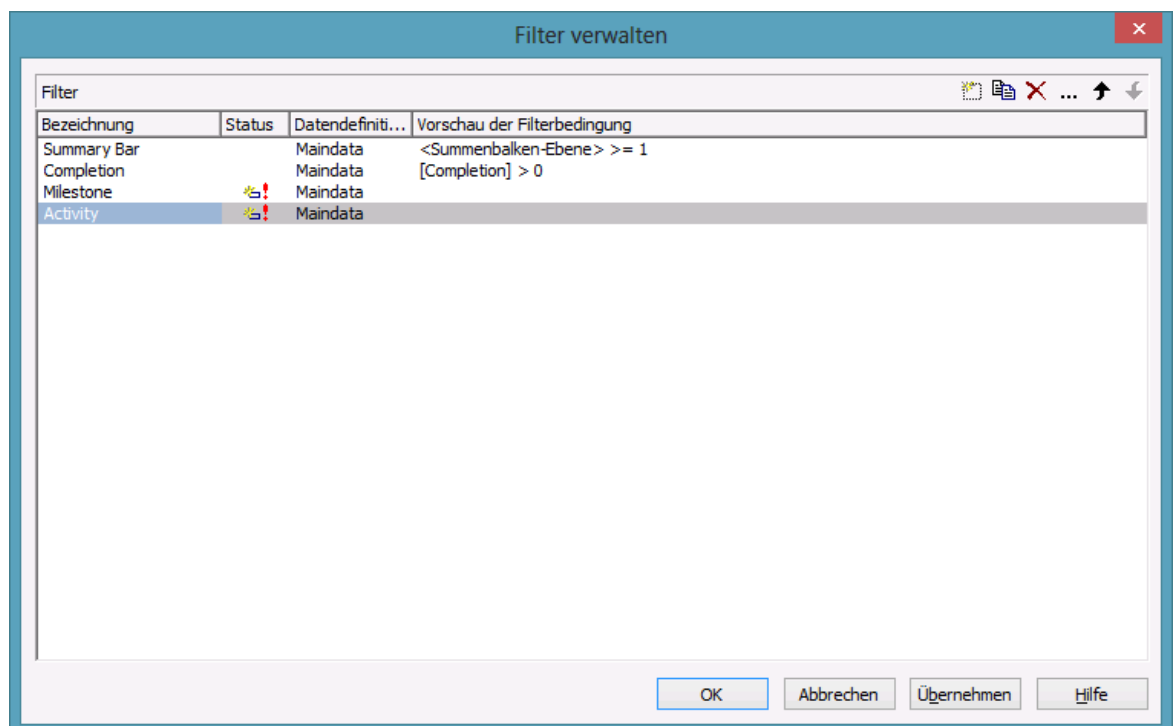
Als Nächstes möchten wir erreichen, dass der gelbe Pfeil nur dann erscheint, wenn ein Meilenstein vorliegt, das heist, die Dauer des Vorgangs ist gleich 0.

Mit Hilfe von Filtern können wir dieses Problem sehr einfach lösen. Ein Filter besteht aus einer Folge von verknüpften Bedingungen, die logisch zu einer Ja/Nein-Aussage führen.

Layer sind immer an Filter gebunden. Der jeweilige Layer wird nur sichtbar, wenn die Auswertung der Filterbedingung zu einem Ja-Wert führt. Der eingebaute Filter <always>, der bereits jedem Layer standardmäßig zugewiesen ist, liefert immer einen Ja-Wert zurück.

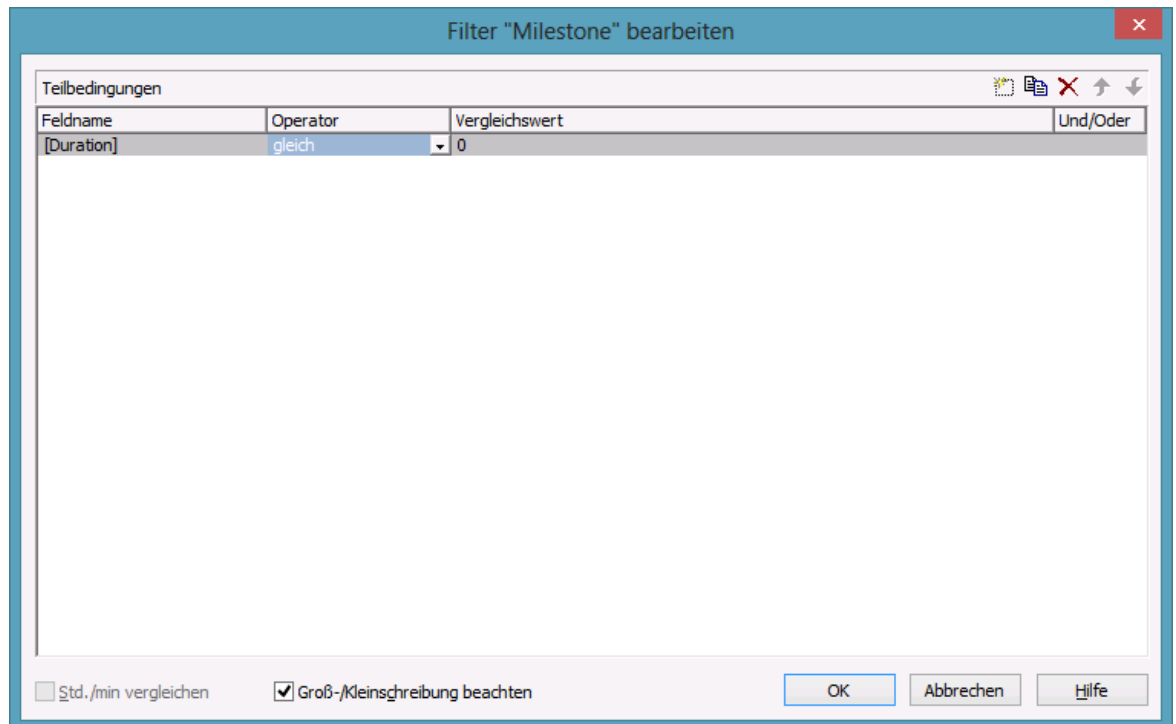
Für unser Beispiel benötigen wir zwei Filter mit je einer Bedingung:

- Der gelbe Pfeil soll erscheinen, wenn die Dauer = 0 ist.
 - Der blaue Balken soll erscheinen, wenn die Dauer > 0 ist.
1. Auf der Eigenschaftenseite **Objekte** gelangen Sie über **Filter...** in den Dialog **Filter verwalten**.
 2. Legen Sie über die Schaltfläche  zwei neue Filter an.

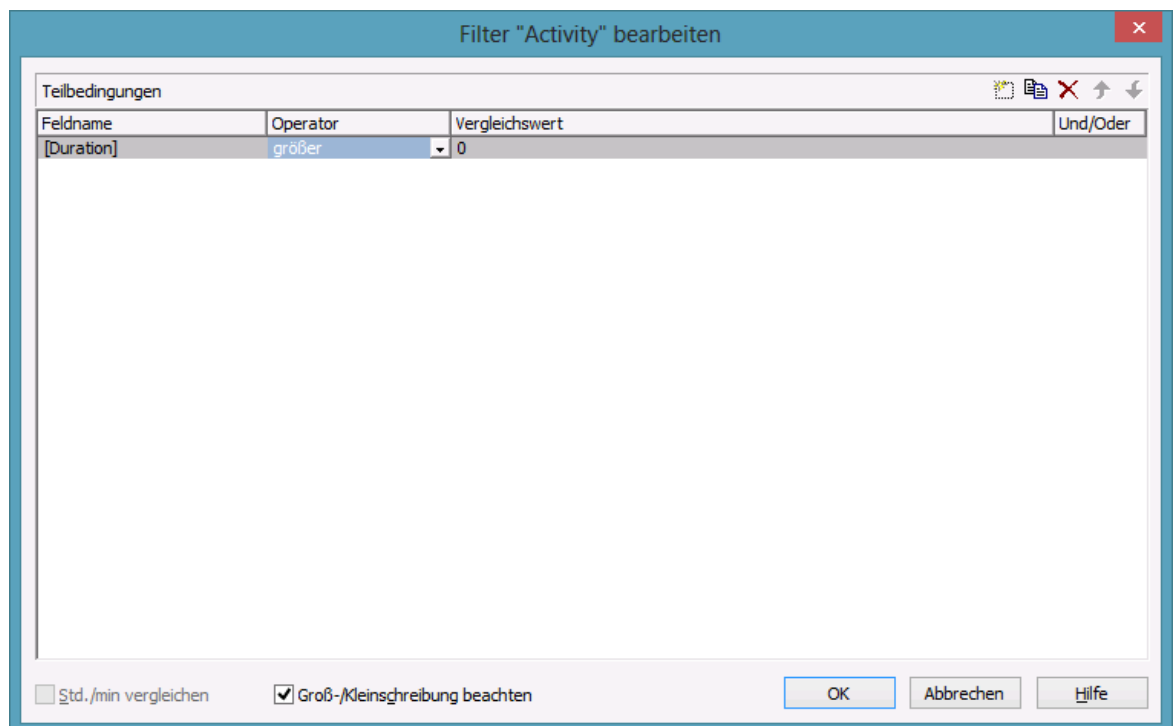


3. Wählen Sie den Filter "Milestone" aus und verzweigen durch Klick auf die Schaltfläche  in den Dialog **Filter bearbeiten**.
4. Wählen Sie als **Feldname** "Dauer" aus. Für den **Operator** setzen Sie gleich und für **Vergleichswert** 0 ein.

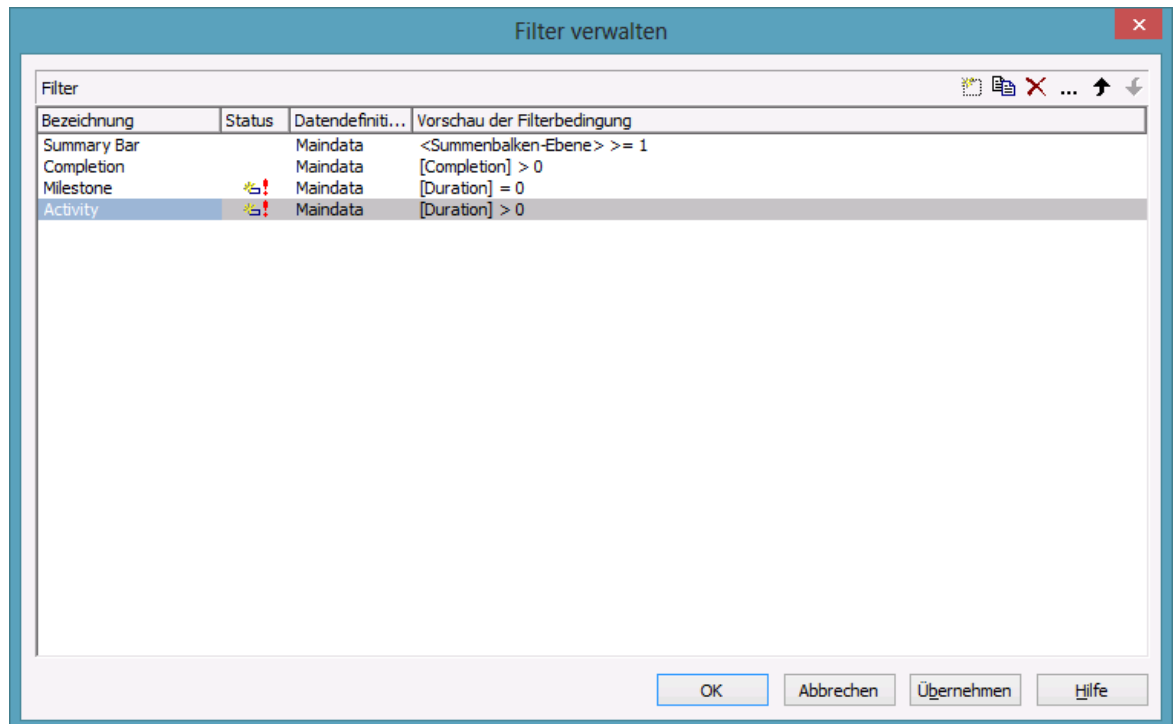
48 Filter verwenden



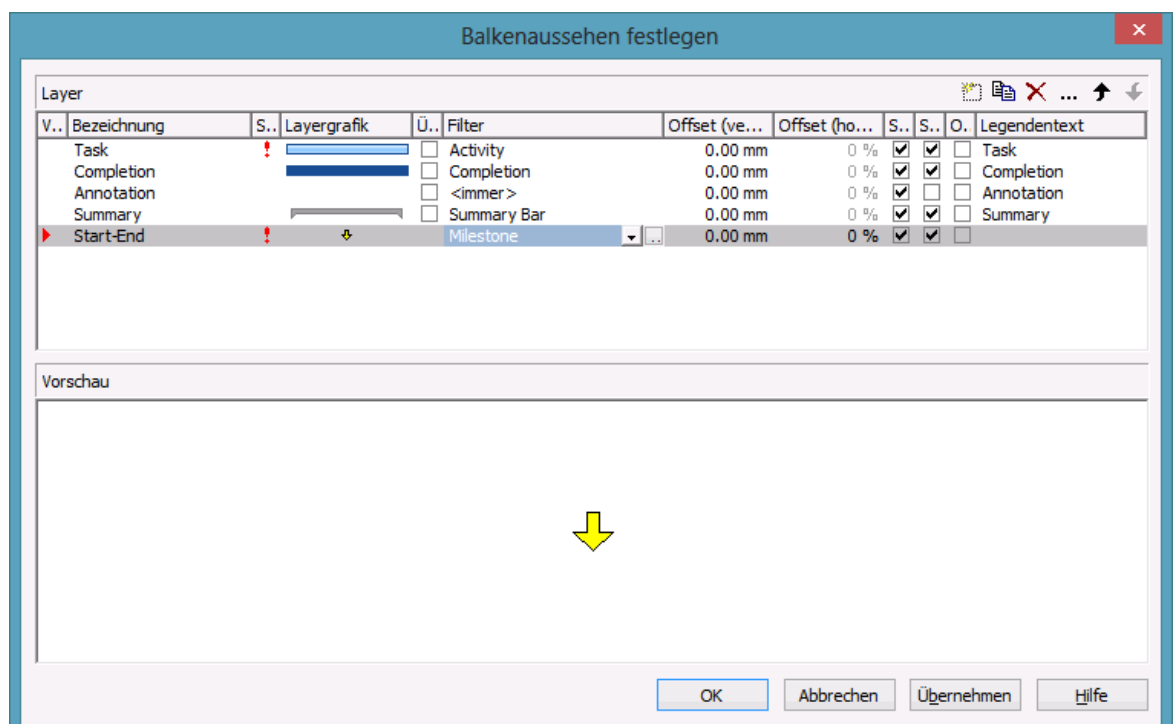
5. Mit **OK** verlassen Sie den Dialog.
6. Wählen Sie **Activity** aus und verzweigen Sie erneut durch die Schaltfläche **...** in den **Filter bearbeiten** Dialog.
7. Wählen Sie als **Feldnamen** "Dauer" aus. Für den **Operator** setzen Sie "größer als" und für **Vergleichswert** 0 ein.



8. Mit **OK** verlassen Sie den Dialog.

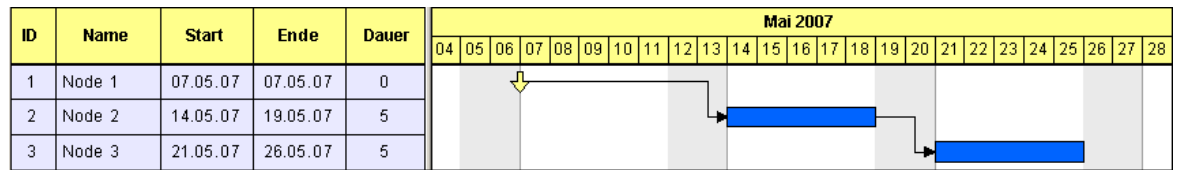


9. Klicken Sie erneut auf **OK** und kehren zu den Eigenschaftenseiten zurück.
10. Damit die Filter wirksam werden, weisen Sie nun bitte die Filter den Layern zu. Bitte rufen Sie über die Schaltfläche **Layer...** den Dialog **Balkenaussehen festlegen** auf.



11. Wenn jetzt das Programm ausgeführt und die Dauer des ersten Vorgangs auf 0 gesetzt wird, ergibt sich folgendes Bild:

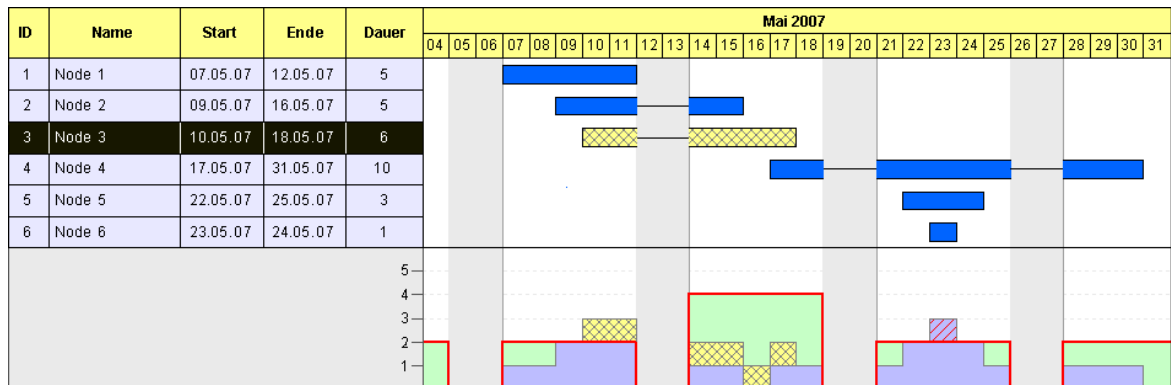
50 Filter verwenden



Das Einstiegsbeispiel finden Sie im Ordner
UserGuideSamples\VB6\XGantt_Tutorial01_App.

2.10 Histogramme erstellen

In diesem Beispiel lernen Sie Histogramme näher kennen. Wir zeigen Ihnen, wie Sie eine Verfügbarkeitskurve definieren, wie Sie die Kapazitätskurve aus den Vorgängen bilden und wie Sie den Anteil der markierten Vorgänge an der Kapazitätsauslastung sichtbar machen.



Jeder Vorgang verursacht in diesem Beispiel während seiner Dauer eine Ressourcenbelegung von einer Einheit. In den Zeiträumen, wo sich Vorgänge zeitlich überlappen, werden die Einzelbelegungen zur Gesamtkapazitätsbelegung aufsummiert.

In den nächsten Schritten werden wir unser bisheriges Beispiel entsprechend erweitern. Zur Verdeutlichung der Histogramm-Funktionen haben wir andere Datensätze verwendet und die Links weggelassen. Das Form1_Load-Beispiel hat sich wie folgt verändert:

Code-Beispiel

```
Private Sub Form_Load()
    VcGantt1.Width = ScaleWidth - VcGantt1.Left
    VcGantt1.Height = ScaleHeight - VcGantt1.Top

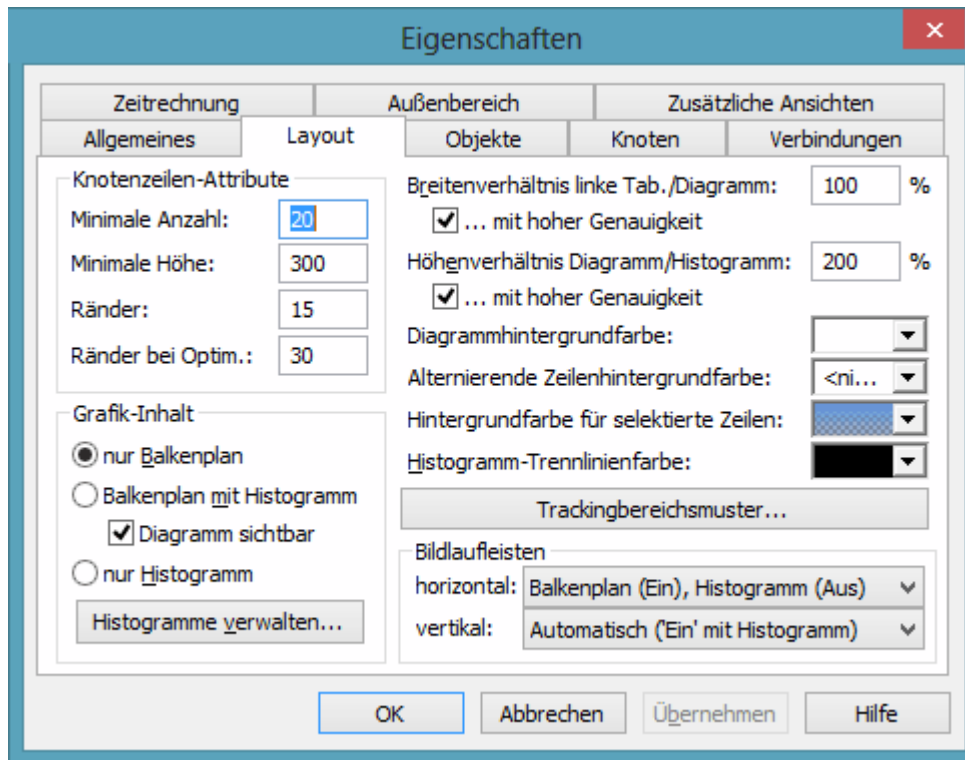
    VcGantt1.InsertNodeRecord ("1;Node 1;07.05.07;;5")
    VcGantt1.InsertNodeRecord ("2;Node 2;09.05.07;;5")
    VcGantt1.InsertNodeRecord ("3;Node 3;10.05.07;;6")
    VcGantt1.InsertNodeRecord ("4;Node 4;17.05.07;;10")
    VcGantt1.InsertNodeRecord ("5;Node 5;22.05.07;;3")
    VcGantt1.InsertNodeRecord ("6;Node 6;23.05.07;;1")
    VcGantt1.EndLoading

    VcGantt1.OptimizeTimeScaleStartEnd (3)

    'Enddatum aller Knoten berechnen
    Dim node As VcNode
    VcGantt.SuspendUpdate True
    For Each node In VcGantt1.NodeCollection
        SetNodeEndDate node
    Next
    VcGantt.SuspendUpdate False
End Sub
```


Außerdem haben wir die Filter ("Milestone", "Aktivität") und den zusätzlichen Layer ("Milestone") aus dem obigen Beispiel wieder herausgenommen. Das fertige Programm finden Sie im Verzeichnis **UserGuideSamples\VB6\XGantt_Tutorial02**.

Auf der Eigenschaftenseite **Layout** müssen Sie zunächst die Histogramm-Darstellung unter **Grafik-Inhalt** durch die Option **Balkenplan mit Histogramm** aktivieren.



Da markierte Knoten durch eine eigene Kreuzschraffur gekennzeichnet werden sollen, setzen Sie auf der Eigenschaftenseite **Knoten** die **Markierungstypen für Knoten im Diagramm** auf **Ohne**.



Für unser Beispiel benötigen wir im weiteren Verlauf ein zusätzliches Datenfeld. Im Dialogfeld **Datentabelle bearbeiten** fügen Sie der Tabelle **Maindata** ein Feld mit dem Namen **Selected** vom Typ **Integer** hinzu. Dieses Feld wird benötigt, um die Darstellung des Vorgangs davon abhängig zu machen, ob er markiert ist oder nicht.

54 Histogramme erstellen

Datentabellen verwalten

Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata	!	☐	
Relations		☐	

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt
0	ID	☒	Integer		☐	☐
1	Name	☐	String		☒	☐
2	Start	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
3	End	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
4	Duration	☐	Integer		☒	☐
5	Completion	☐	Integer		☒	☐
6	Group Level 1	☐	String		☒	☐
7	Group Level 2	☐	String		☒	☐
8	Release Date	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
9	Due Date	☐	Datum/Zeit	DD.MM.YYYY	☒	☐
10	Selected	☐	String		☒	☐

OK Abbrechen Übernehmen Hilfe

Der Wert für das Feld **Selected** muss jedesmal aktualisiert werden, wenn das Ereignis **OnNodesMarkComplete** ausgelöst wird.

Code-Beispiel

```
Private Sub VcGantt1_OnNodesMarkComplete ()
    Dim node As VcNode
    For Each node In VcGantt1.NodeCollection
        If node.MarkNode = True Then
            node.DataField(5) = 1
        Else
            node.DataField(5) = 0
        End If
        node.UpdateNode
    Next
End Sub
```

Im Ereignis **OnNodeCreateCompleteEx** verhindert der folgende Programmcode, dass ein neu erzeugter Knoten direkt markiert ist. Da gleichzeitig beim Erzeugen eines neuen Knotens die Markierung aller vorher ausgewählten Knoten aufgehoben wird, muss der Feldinhalt von **Selected** aktualisiert werden.

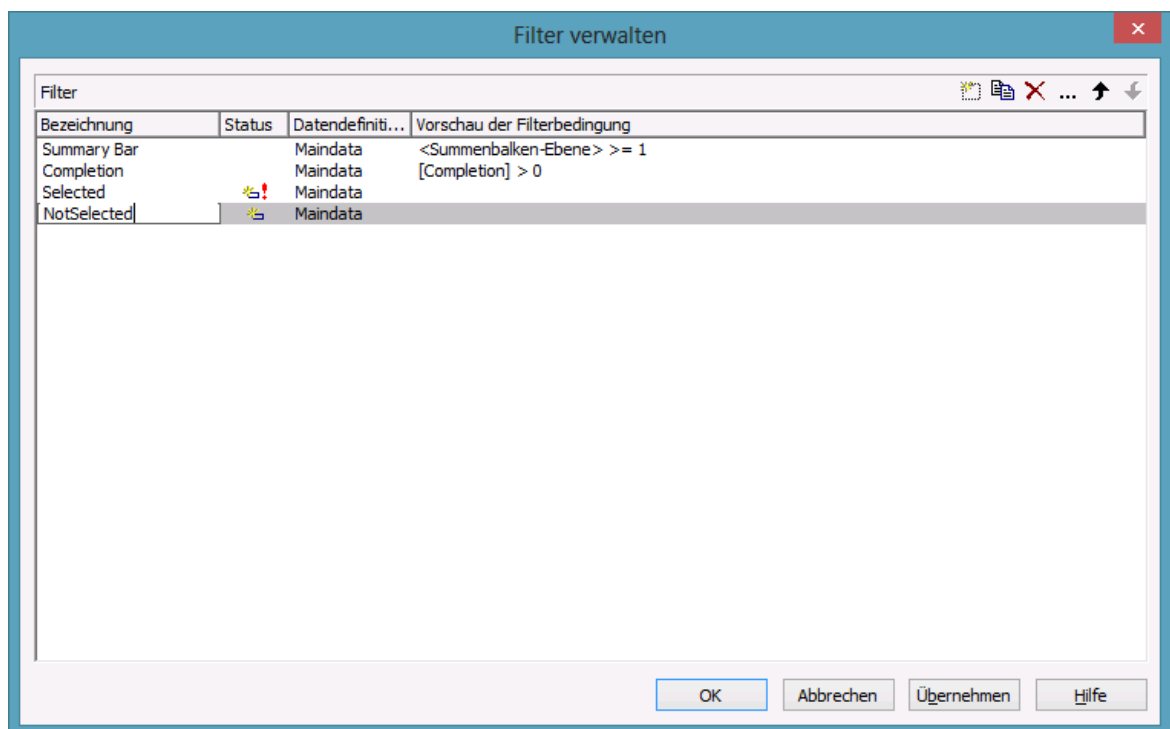
Code-Beispiel

```
Private Sub VcGantt1_OnNodeCreateCompleteEx _
    (ByVal node As VcGanttLib.VcNode, _
    ByVal creationType As VcGanttLib.CreationTypeEnum, _
    ByVal isLastNodeInSeries As Boolean)

    node.DataField(1) = "Node " + node.DataField(0)
    node.MarkNode = False
    node.UpdateNode

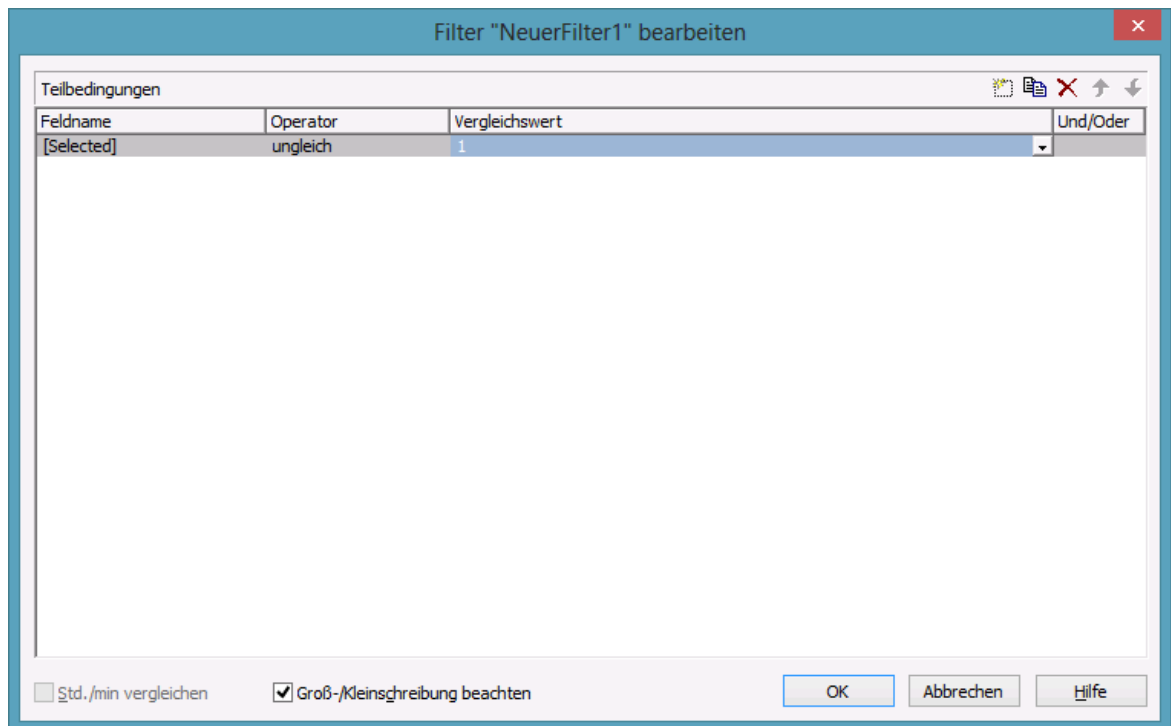
    Dim node As VcNode
    For Each node In VcGantt1.NodeCollection
        node.DataField(5) = 0
        node.UpdateNode
    Next
End Sub
```

Im nächsten Schritt definieren Sie Filter, die markierte und nicht markierte Vorgänge unterscheiden. Auf der Eigenschaftenseite **Objekte** gelangen Sie über die Schaltfläche **Filter...** in den Dialog **Filter verwalten**. Legen Sie dort über die Schaltfläche  zwei neue Filter an und nennen Sie den einen **NotSelected** und den anderen **Markiert**.

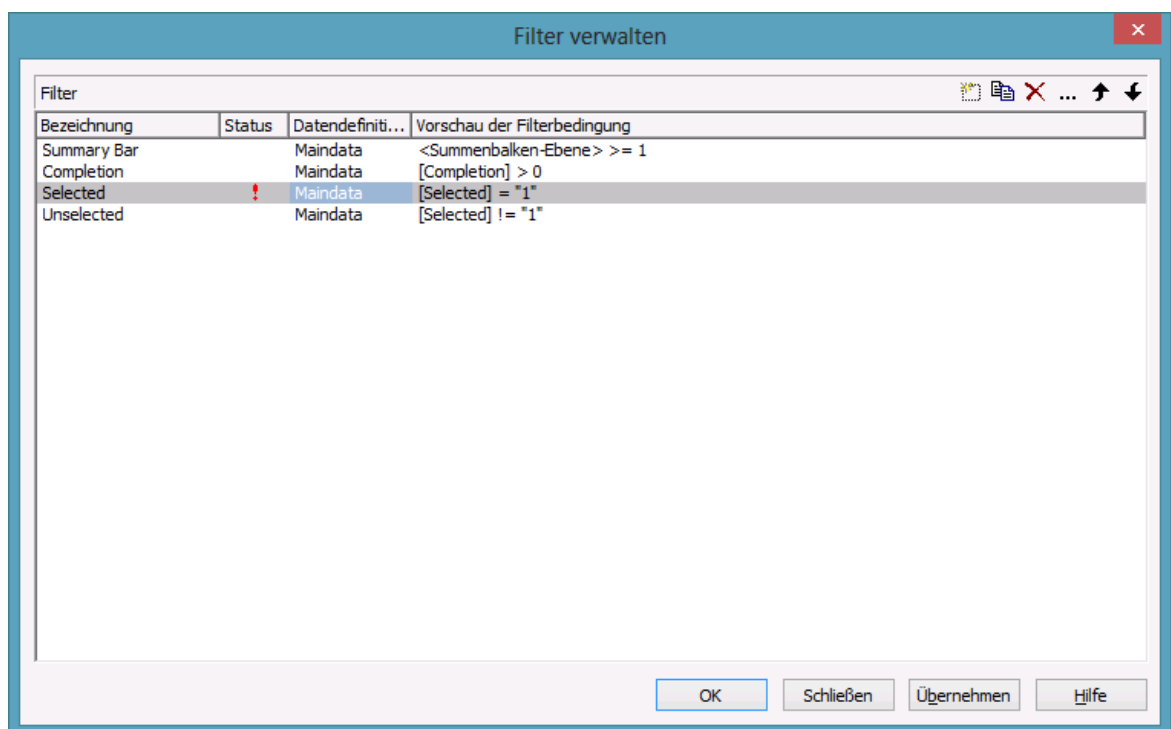


56 Histogramme erstellen

Dem Filter mit dem Namen **NotSelected** setzen Sie die Bedingung **Selected ungleich 1**.

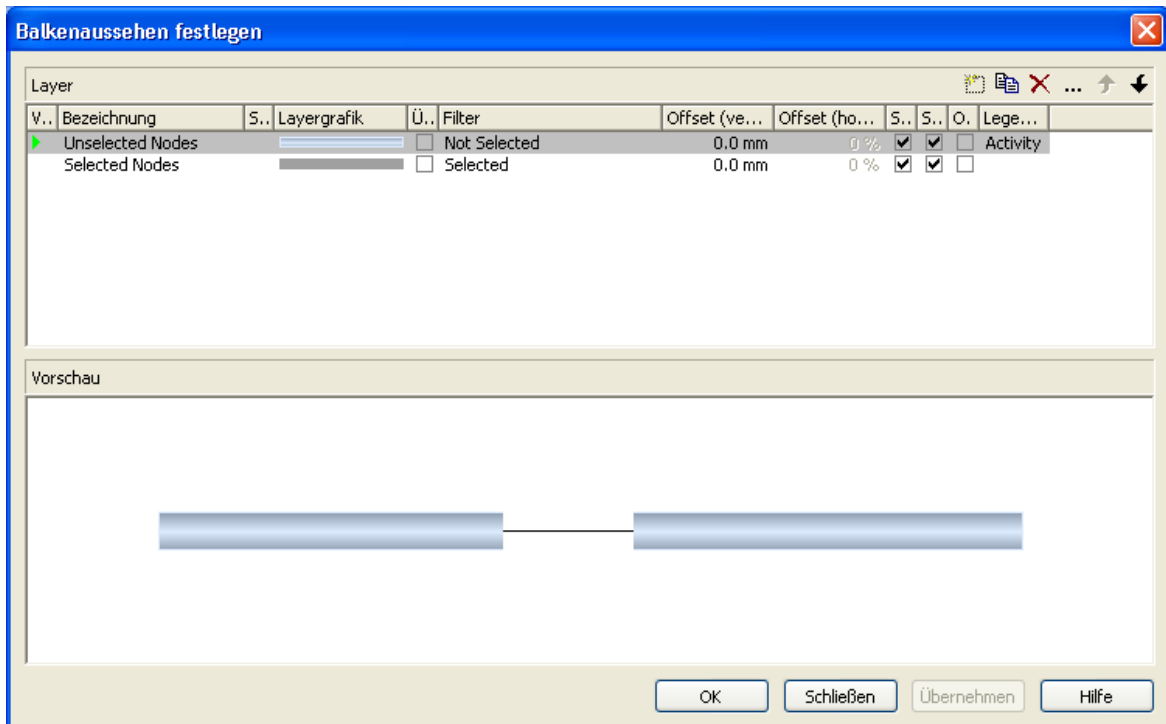


Der Filter mit dem Namen **Selected** erhält die Bedingung **Selected gleich 1**.



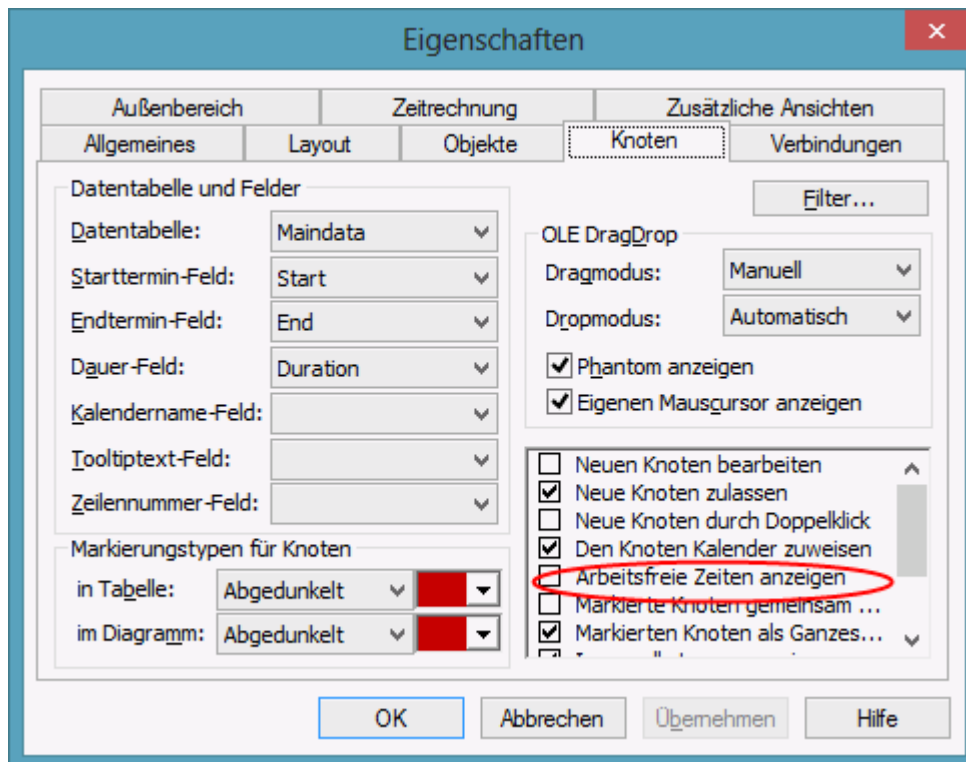
Das Aussehen der Vorgänge soll nun von den Filtern abhängig gemacht werden.

Auf der Eigenschaftenseite **Objekte** gelangen Sie über **Layer...** in den Dialog **Balkenaussehen festlegen**. Den vorhandenen Layer **Start-Ende** benennen Sie in **Unmarkierte Knoten** um und weisen ihm den Filter **Unmarkiert** zu. Dann fertigen Sie über die Schaltfläche  eine Kopie des Layers an und nennen ihn **Markierte Knoten**. Weisen Sie dem Layer den Filter **Selected** zu.

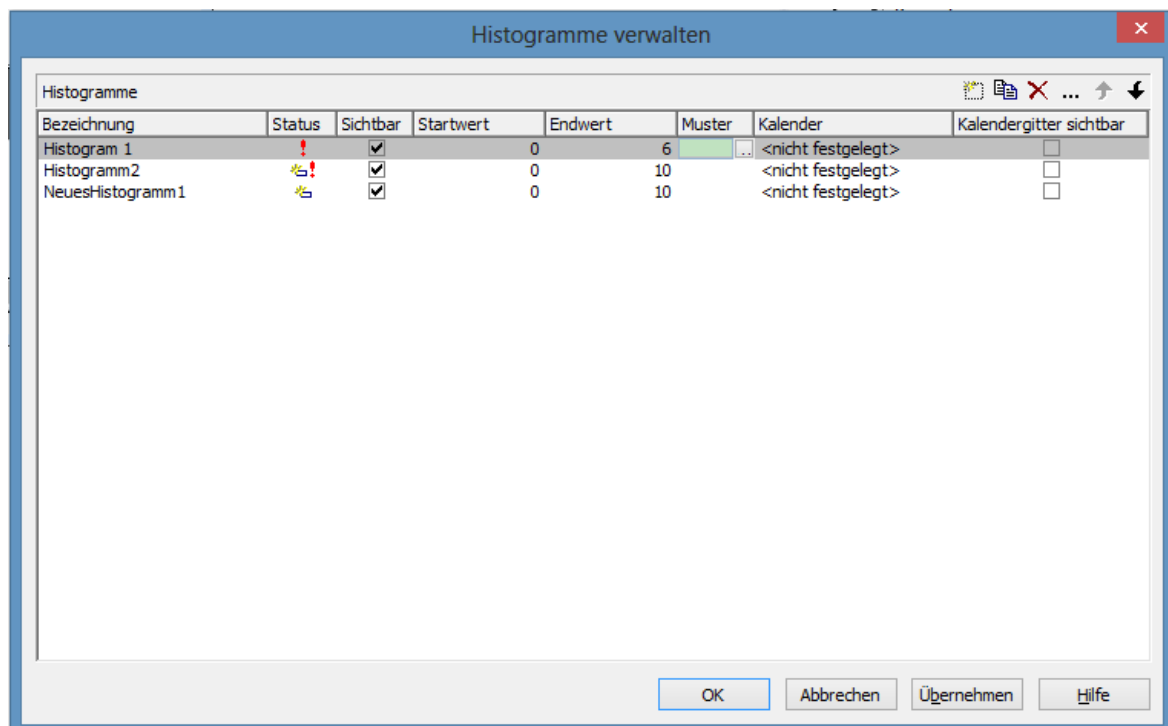


Noch besitzen beide Layer das gleiche Aussehen. Für den Layer **SelectedNodes** ändern Sie das Design, indem Sie im Dialog **Layer bearbeiten** das Muster auf Kreuzschraffur und die Hintergrundfarbe auf gelb umstellen.

Hinweis: Auf der Eigenschaftenseite **Knoten** sollte die Option **Arbeitsfreie Zeiten anzeigen** aktiviert sein, damit innerhalb der arbeitsfreien Zeit (z.B. am Wochenende) nur eine Linie statt eines Balkens erscheint.



Im nächsten Schritt müssen wir die Kurven für das Histogramm definieren. Auf der Eigenschaftenseite **Layout** gelangen Sie über **Histogramme verwalten...** in den entsprechenden Dialog.



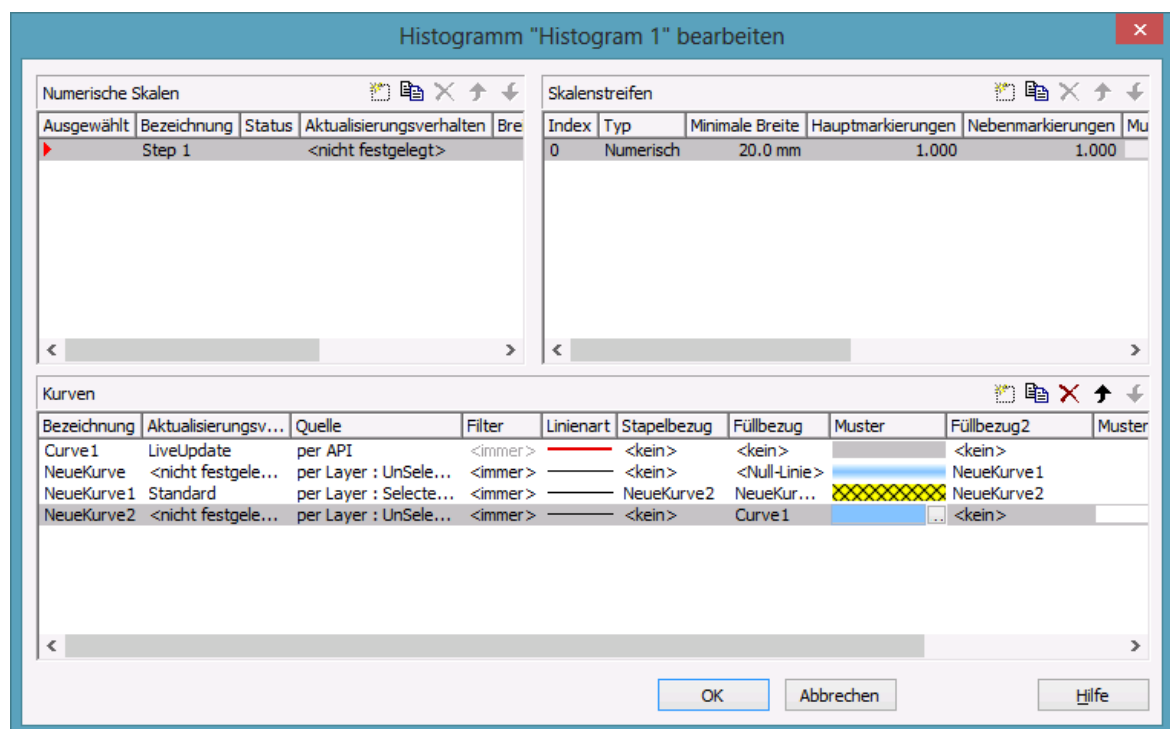
In einem Gantt-Diagramm können gleichzeitig mehrere Histogramme existieren. Jedes Histogramm besteht aus einer numerischen Skala und einer beliebigen Anzahl von Kurven.

Legen Sie nun den Start- und Endwert für die numerische Skala des Histogramms fest. Dazu stellen Sie im **Histogramm 1** den Endwert auf 6.

Um nun das bereits vordefinierte Histogramm im Detail weiter zu bearbeiten, klicken Sie auf die Schaltfläche **Histogramme bearbeiten** ...

"Curve 1" legt die rote Verfügbarkeitskurve fest. "Curve 2" übernimmt die Darstellung der markierten Knoten und "Curve 3" die der unmarkierten Knoten. "Curve 4" sorgt für den grünen Hintergrund zur Darstellung der noch freien Kapazitäten.

Eine Kurve ist bereits vorhanden. Erstellen Sie drei weitere Kurven und legen Sie die Eigenschaften, wie in der Abbildung zu sehen, entsprechend fest.

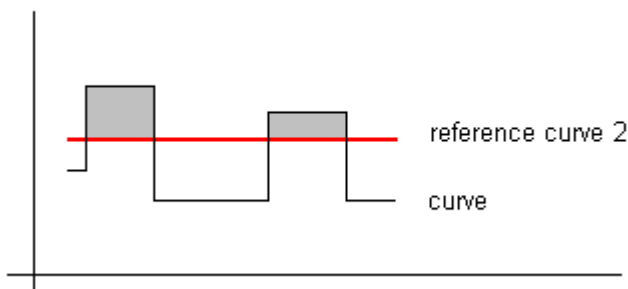
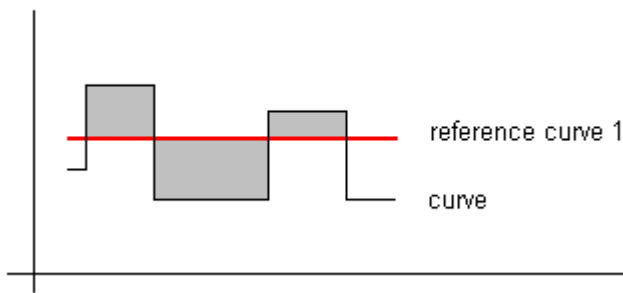


Kurven können gestapelt werden, so dass ihre Werte sich addieren. Eine Kurve, deren Werte zu denen einer anderen hinzuaddiert werden sollen, benötigt die entsprechende Angabe einer Bezugskurve. Diese wählen Sie unter **Stapelbezug** aus.

Zwischen den Kurven eines Histogramms entstehen Flächen, denen Sie Farbe und Muster zuweisen können. Eine Kurve, die eine Fläche begrenzen soll, benötigt eine Bezugskurve.

Zwei Arten von Flächen werden unterschieden:

1. Flächen, oberhalb und/oder unterhalb einer Kurve liegen können
2. Flächen, die nur oberhalb einer Kurve liegen und damit Überschreitungen kennzeichnen



Für den Fall 1 geben Sie bitte die Bezugskurve bei **Füllbezug** an; für den Fall 2 bei **Füllbezug2**. Der Bezug zur X-Achse wird durch die Option "<Null-Linie>" angegeben.

Zuletzt muss noch der Programmcode im **Load-Ereignis** verändert werden. Benötigt werden die Werte für die Vorgänge und die Verfügbarkeitskurve.

Code-Beispiel

```
Private Sub Form_Load()
    VcGantt1.Width = ScaleWidth - VcGantt1.Left
    VcGantt1.Height = ScaleHeight - VcGantt1.Top

    VcGantt1.InsertNodeRecord ("1;Node 1;07.05.09;;5")
    VcGantt1.InsertNodeRecord ("2;Node 2;09.05.09;;5")
    VcGantt1.InsertNodeRecord ("3;Node 3;10.05.09;;6")
    VcGantt1.InsertNodeRecord ("4;Node 4;17.05.09;;10")
    VcGantt1.InsertNodeRecord ("5;Node 5;22.05.09;;3")
    VcGantt1.InsertNodeRecord ("6;Node 6;23.05.09;;1")
    VcGantt1.EndLoading

    VcGantt1.OptimizeTimeScaleStartEnd (3)

    'Enddatum aller Knoten berechnen
    Dim node As VcNode
    For Each node In VcGantt1.NodeCollection
        SetNodeEndDate node
    Next

    Dim histogram As VcHistogram
    Dim curve As VcCurve

    Set histogram = VcGantt1.HistogramCollection.FirstHistogram
    Set curve = histogram.CurveCollection.CurveByName("Curve1")
    curve.PointsEquidistant = False
    curve.SetValues "01.05.09", "2"
```

```

curve.SetValues "05.05.09", "0"
curve.SetValues "07.05.09", "2"
curve.SetValues "12.05.09", "0"
curve.SetValues "14.05.09", "4"
curve.SetValues "19.05.09", "0"
curve.SetValues "21.05.09", "2"
curve.SetValues "26.05.09", "0"
curve.SetValues "28.05.09", "2"
End Sub

```

Starten Sie das Programm und klicken Sie auf einen Vorgang. Im Histogramm können Sie durch die gelb hinterlegte Schraffur sofort erkennen, welchen Anteil der Vorgang an der gesamten Ressourcenbelegung hat.

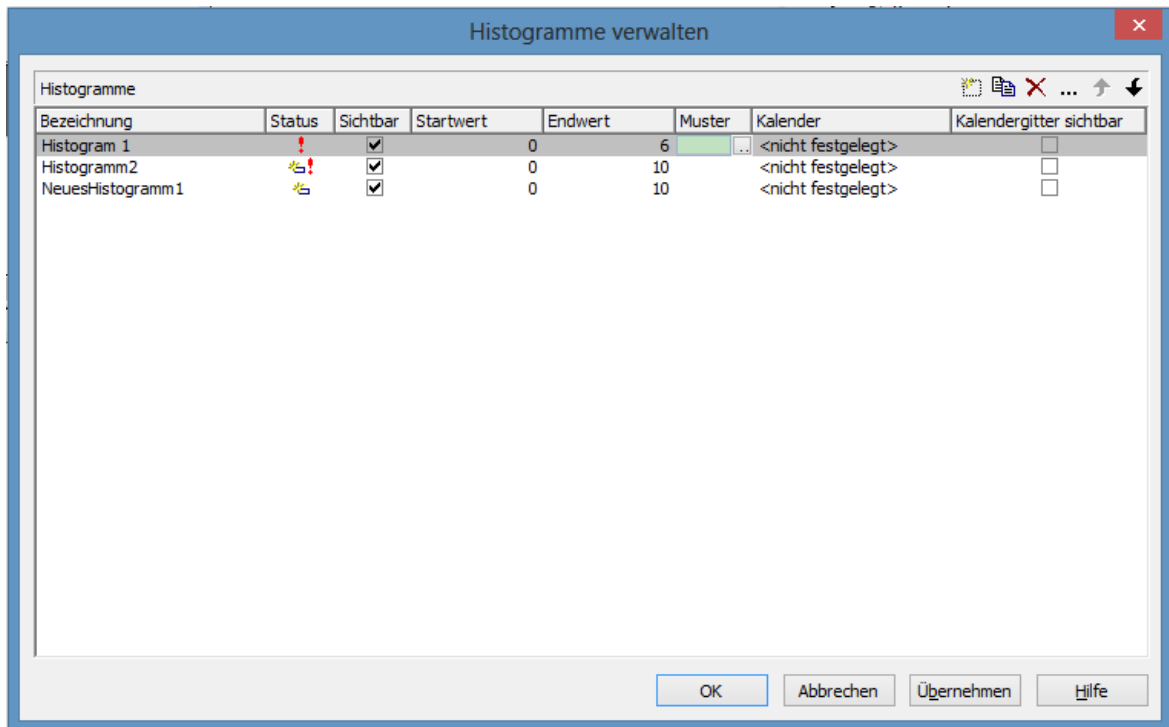
Wenn Sie Vorgänge verschieben, ändert sich das Kapazitätsgebirge und Sie erkennen Über- bzw. Unterlastungen, die sich aus dieser Interaktion ergeben.

Kalendergitter im Histogramm

Sie können einem Histogramm ein oder mehrere Kalendergitter zuweisen, so dass Sie dort z.B. die unterschiedlichen Kalendergitter des Gantt-Graphen auch im Histogramm abbilden können.

Um einem Histogramm ein eigenes Kalendergitter zuweisen zu können, müssen drei Bedingungen erfüllt sein:

1. Dem Histogramm muss ein Kalender zugewiesen sein
2. Das Kalendergitter muss eingeschaltet sein
3. Für die Darstellung des Kalendergitters muss ein Aussehen definiert sein



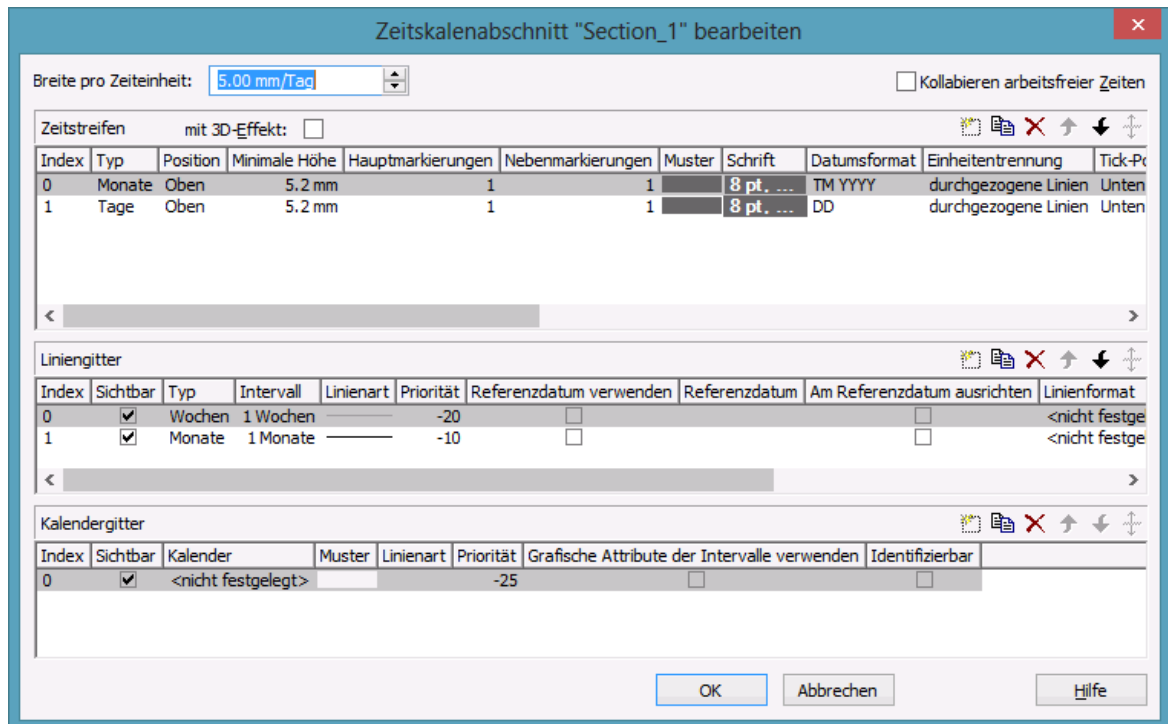
Kalender zugewiesen, Kalendergitter eingeschaltet

Die entsprechenden API-Aufrufe heißen:

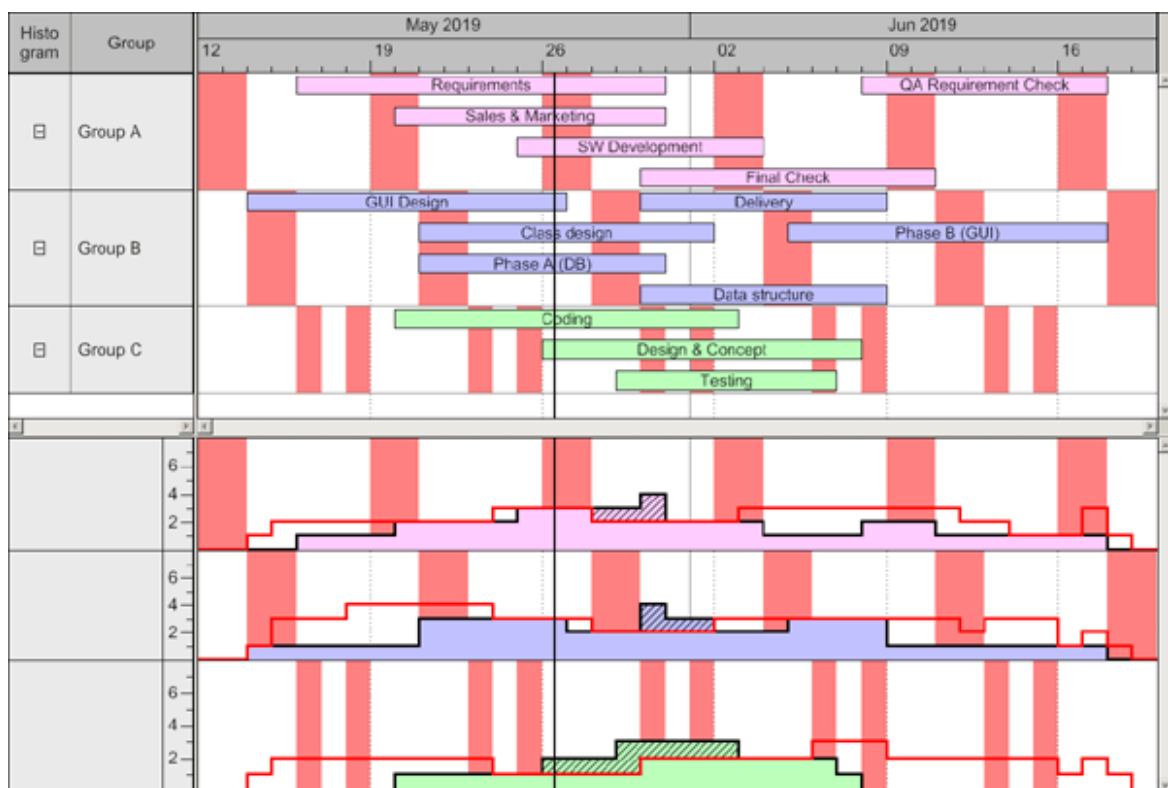
Code-Beispiel

```
// Zuweisung des Kalendernamens an das Histogramm:
histogram.calendarName = group.DataField(14)
// Einschalten des Kalendergitters
histogram.ShowCalendarGrids = True
// Einschalten des Histogramms
histogram.Visible = True
```

Als Kalendergitter für das Histogramm verwendet VARCHART XGantt das erste nicht sichtbare Kalendergitter des ersten Zeitabschnitts (Section) in der Zeitskala, sofern es dort kein weiteres gibt. Dieses ist auch das Kalendergitter, das vom Gantt-Graphen für die Gruppen verwendet wird (groupwise).



Das Kalendergitter zeigt dadurch dasselbe Aussehen im Gantt-Graphen wie im Histogramm. Im Beispiel unten ist dies ein Kalendergitter, das für jede Gruppe ein unterschiedliches Muster zeigt (groupwise calendar grid):



Setzen Sie dem Zeitskalenabschnitt ein weiteres unsichtbares Kalendergitter, verwendet VARCHART XGantt dieses als Kalendergitter des Histogramms:

64 Histogramme erstellen

Zeitskalenabschnitt "Section_1" bearbeiten

Breite pro Zeiteinheit: 5.00 mm/Tag ☐ Kollabieren arbeitsfreier Zeiten

Zeitschreibe mit 3D-Effekt: ☐

Index	Typ	Position	Minimale Höhe	Hauptmarkierungen	Nebenmarkierungen	Muster	Schrift	Datumsformat	Einheitentrennung	Tick-Pr
0	Monate	Oben	5.2 mm	1	1		8 pt, ...	TM YYYY	durchgezogene Linien	Unten
1	Tage	Oben	5.2 mm	1	1		8 pt, ...	DD	durchgezogene Linien	Unten

Liniengitter

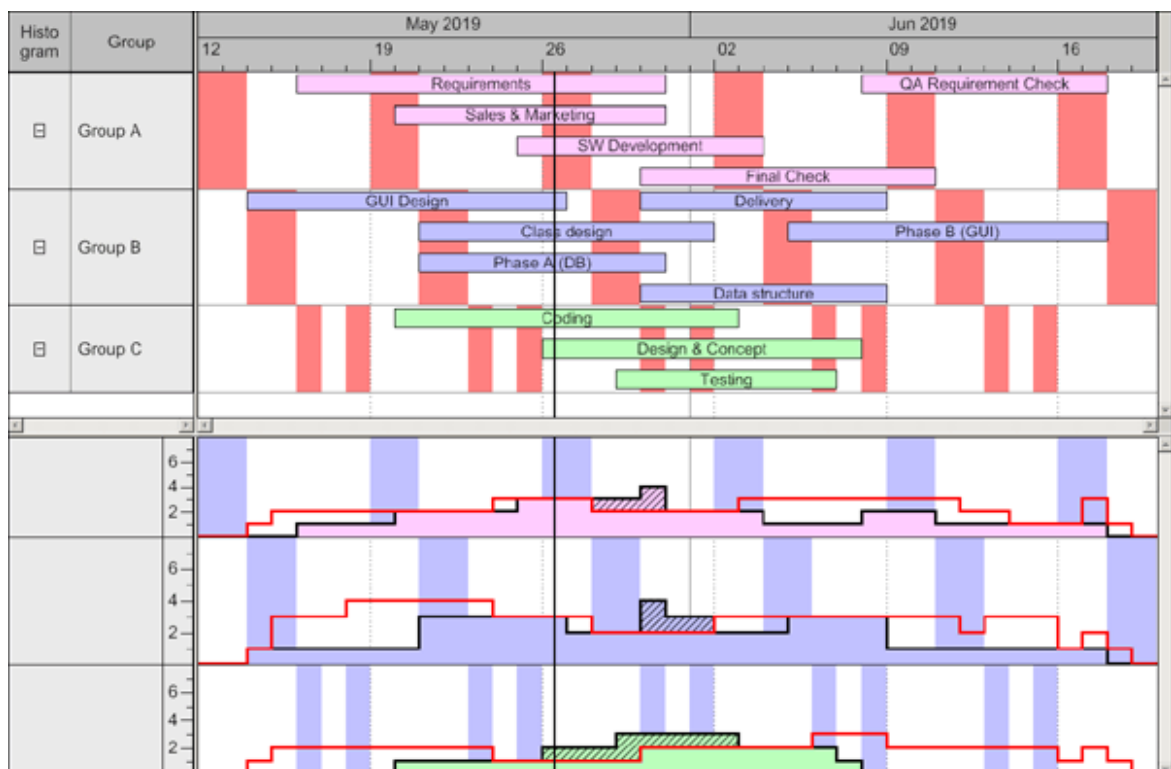
Index	Sichtbar	Typ	Intervall	Linienart	Priorität	Referenzdatum verwenden	Referenzdatum	Am Referenzdatum ausrichten	Linienformat
0	☒	Wochen	1 Wochen		-20	☐		☐	<nicht festge
1	☒	Monate	1 Monate		-10	☐		☐	<nicht festge

Kalendergitter

Index	Sichtbar	Kalender	Muster	Linienart	Priorität	Grafische Attribute der Intervalle verwenden	Identifizierbar
0	☐	<nicht festgelegt>			-25	☐	☐
1	☐	<nicht festgelegt>			-100	☐	☐

OK Abbrechen Hilfe

Dieses Kalendergitter kann ein anderes Aussehen haben als das Kalendergitter des Gantt-Graphen. In unserem Fall hat es eine andere Farbe:



2.11 Diagramm drucken

Wenn Sie Ihr Diagramm nach Ihren Vorstellungen gestaltet haben, können Sie es schließlich ausdrucken. Wählen Sie dazu zur Laufzeit den Befehl **Drucken** des Kontextmenüs (rechter Mausklick im freien Diagrammbereich).

Alternativ können Sie auch die Methode **PrintIt** des Objektes VcGantt verwenden.

Sie gelangen dann in das Windows-Dialogfeld **Drucken**. Gegebenenfalls können Sie zur Laufzeit auch die Druckereinstellungen bearbeiten, indem Sie den Befehl **Drucker einrichten** des Kontextmenüs aufrufen und damit das entsprechende Windows-Dialogfeld aufrufen.

Mit der Methode **PrintDirect** des Objektes VcGantt können Sie das Diagramm direkt ausdrucken, ohne dass zuvor ein Dialogfeld erscheint.

Wenn Sie zur Laufzeit die Seiteneinstellungen verändern möchten, können Sie aus dem Kontextmenü den Befehl **Seite einrichten** auswählen, oder auf **Seitenansicht** klicken und dort auf die Schaltfläche **Seite einrichten**.

Alternativ können Sie auch die Methode **PageLayout** des Objektes VcGantt verwenden, um das entsprechende Dialogfeld aufzurufen.

Im **Seite einrichten**-Dialogfeld können Sie Einstellungen zur Skalierung, zur Seitenaufteilung, zur Seitennummerierung, zu den Seitenrändern etc. vornehmen. Weitere Informationen hierzu finden Sie im Kapitel 5.23, "Seite einrichten".

2.12 Diagramm exportieren

Sie können Ihr Diagramm auch als Grafikdatei exportieren. Dazu gibt es folgende Möglichkeiten:

- Wählen Sie den Befehl **Grafik exportieren** des Standard-Kontextmenüs. Dann gelangen Sie in das Windows-Dialogfeld **Speichern unter**, in dem Sie das dargestellte Diagramm als Grafikdatei speichern können.
- Verwenden Sie die API-Methode **ShowExportGraphicsDialog** bzw. **ExportGraphicsToFile**

Detaillierte Erläuterungen zu den Grafikformaten finden Sie im Kapitel: **Wichtige Konzepte: Grafikformate**.

2.13 Konfigurationseinstellungen speichern

Alle Einstellungen der Eigenschaftenseiten können Sie jederzeit in Form einer Konfiguration außerhalb Ihres Projektes speichern und nach Bedarf wieder einlesen. Dies ist sehr praktisch, wenn Sie zu einem früheren Stand der Einstellungen zurückkehren oder die gleichen Einstellungen für andere Projekte verwenden möchten.

Eine gespeicherte Konfiguration besteht aus zwei Dateien mit gleichem Namen aber unterschiedlichen Dateiendungen. Zu einer Konfiguration gehört jeweils eine INI- und eine IFD-Datei, die beide zwingend benötigt werden.

> So speichern Sie Ihre aktuelle Konfiguration:

Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, in der die aktuellen Einstellungen gespeichert werden sollen. Wenn Sie einen nicht vorhandenen Dateinamen angeben und auf die **Übernehmen**-Schaltfläche klicken, wird diese INI-Datei erzeugt und mit der VARCHART ActiveX-Instanz verknüpft. Jede Änderung auf einer Eigenschaftenseite wird nun in dieser Datei gespeichert.

> So lesen Sie eine bereits gespeicherte Konfiguration wieder ein:

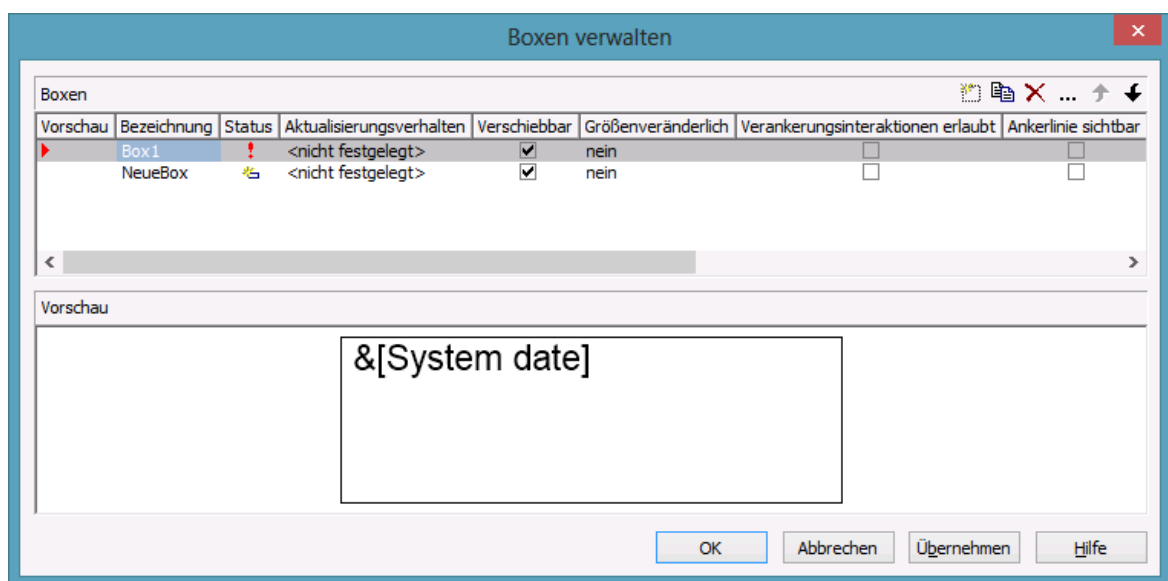
Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, aus der die Einstellungen geladen werden sollen. Wenn Sie einen vorhandenen Dateinamen angeben und auf die **Übernehmen**-Schaltfläche klicken, wird die gewählte Konfiguration geladen und von nun an mit der VARCHART XGantt ActiveX-Instanz verknüpft. Alle aktuellen Einstellungen verfallen dabei unwiderruflich.

Hinweis: Die Einstellungen der Konfigurationsdatei werden nur einmal geladen, d. h. VARCHART XGantt liest sie nicht ein zweites Mal aus derselben Datei. Stattdessen werden die Einstellungen aus dem internen Speicher der Entwicklungsoberfläche geladen, die mit denen der Konfigurationsdatei identisch sind. Daher hätte es keine Auswirkung, wenn Sie die Konfigurationsdatei mit Hilfe eines Editors bearbeiten würden. Falls Sie möchten, dass das VARCHART XGantt eine veränderte Konfigurationsdatei verwendet, müssen Sie die veränderte ini-Datei und die zugehörige ifd-Datei umbenennen und auf der Eigenschaftenseite **Allgemeines** unter **Konfigurationsdatei** den Dateinamen der veränderten ini-Datei angeben.

3 Wichtige Konzepte

3.1 Boxen

Im Diagrammbereich können beliebig viele Boxen, die Text oder Grafiken enthalten können, dargestellt werden. Über die Eigenschaftenseite **Objekte**, Schaltfläche **Boxen...** gelangen Sie zum Dialog **Boxen verwalten**, in dem Sie Boxen neu anlegen, kopieren, bearbeiten und löschen können.



Mithilfe der Eigenschaften **Ursprung**, **Referenzpunkt**, **X-Offset** und **Y-Offset** können Sie jede einzelne Box im Diagrammbereich positionieren, wobei die relative Position der Box zum Diagramm unabhängig von der Diagrammgröße ist.

Für jede Box können Sie hier folgendes festlegen:

- ihre Bezeichnung
- ob sie zur Laufzeit frei im Diagrammbereich verschiebbar sein soll
- ob und wie ihre Größe interaktiv geändert werden kann
- ob Verankerungsinteraktionen per Maus oder Kontextmenü erlaubt sind
- ob bei Verankerung die eingestellten Referenzpunkte beim Knoten und bei der Box durch eine Linie verbunden werden sollen
- eine Knoten-ID, um den Knoten zu identifizieren, an dem die jeweilige Box verankert werden soll

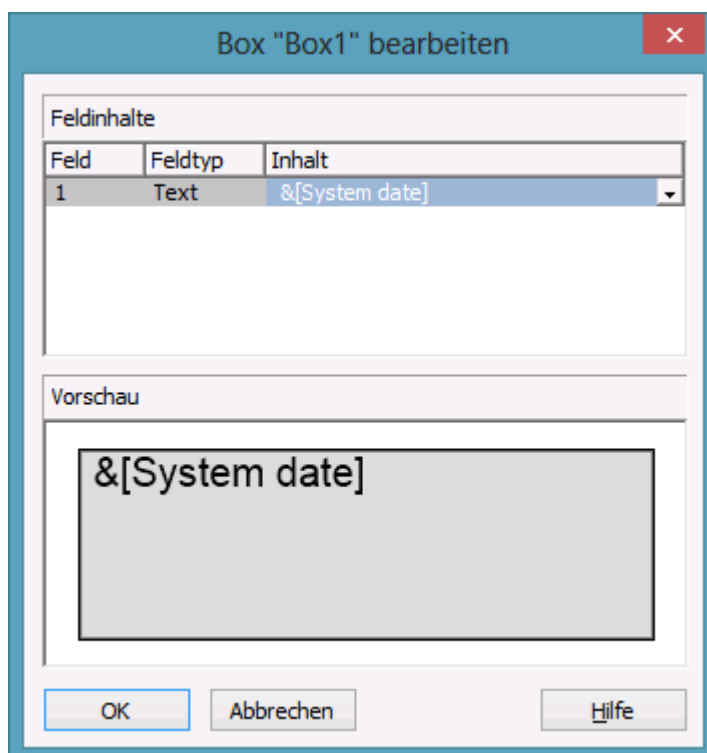
70 Wichtige Konzepte: Boxen

- ihren Ursprung (den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird)
- ihren Referenzpunkt (Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird)
- ihren X- bzw. Y-Offset (Abstand zwischen Ursprung und Referenzpunkt in x- bzw. y-Richtung)
- Typ, Dicke und Farbe der Umrandungslinie der Box
- ihre relative Priorität gegenüber anderen Objekten im Diagramm
- ob die Box sichtbar ist
- das Boxformat

> Boxen bearbeiten

Im Dialogfeld **Box bearbeiten** können Sie den Inhalt der Felder festlegen.

Zur Designzeit erreichen Sie dieses Dialogfeld, indem Sie im Dialogfeld **Boxen verwalten** auf die **Box bearbeiten**-Schaltfläche klicken. Zur Laufzeit kann ein Benutzer durch Doppelklick mit der linken Maustaste auf die jeweilige Box in das Dialogfeld gelangen. Wenn die Option **In-Place-Editieren zulassen** auf der Eigenschaftsseite **Allgemeines** aktiviert wird, können Texte zur Laufzeit auch direkt bearbeitet werden.



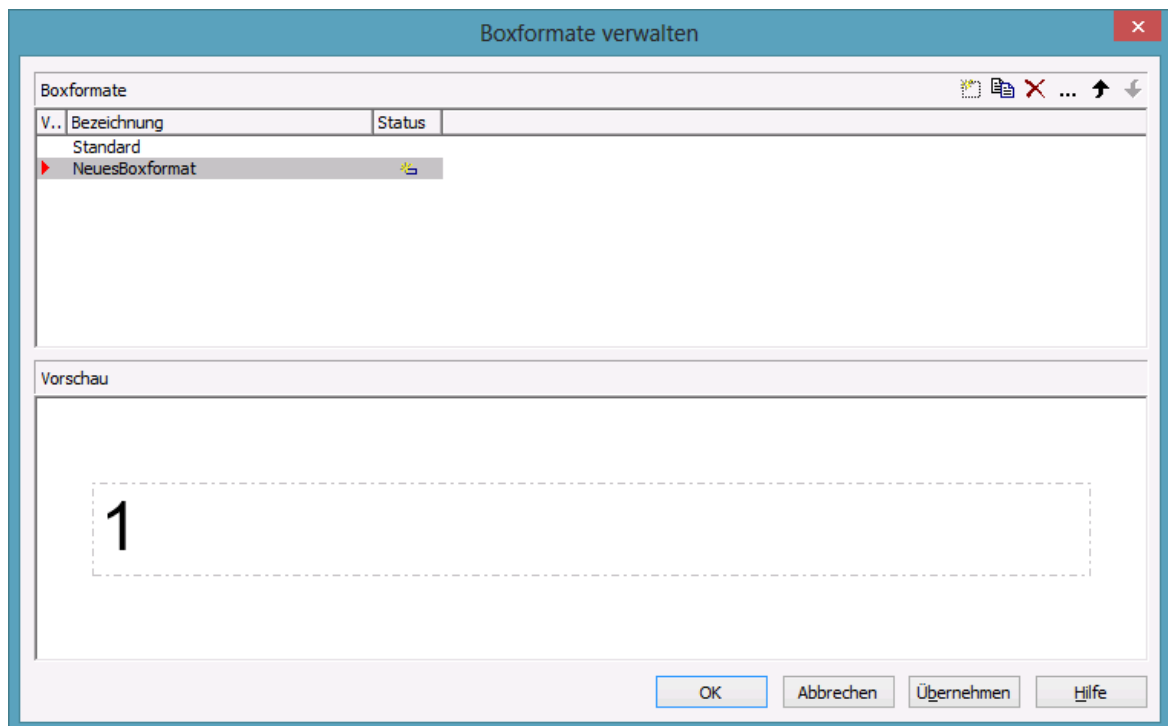
In der Spalte **Feld** werden die Nummern aller Felder der Box aufgeführt. Die Anzahl der Felder hängt vom gewählten Boxformat ab (s. weiter unten).

In der Spalte **Typ** wird der Feldtyp jedes Feldes angezeigt (Text oder Grafik). In der Spalte **Inhalt** können Sie den Inhalt des Feldes bzw. den Namen einer Grafikdatei eingeben. Bei mehrzeiligen Textfeldern müssen für einen erzwungenen Umbruch die einzelnen Zeilen des Textfelds mit "\n" im String getrennt sein (Beispiel: "Zeile1\nZeile2"). Ohne erzwungenen Umbruch wird automatisch an Leerzeichen umgebrochen.

> Boxformate

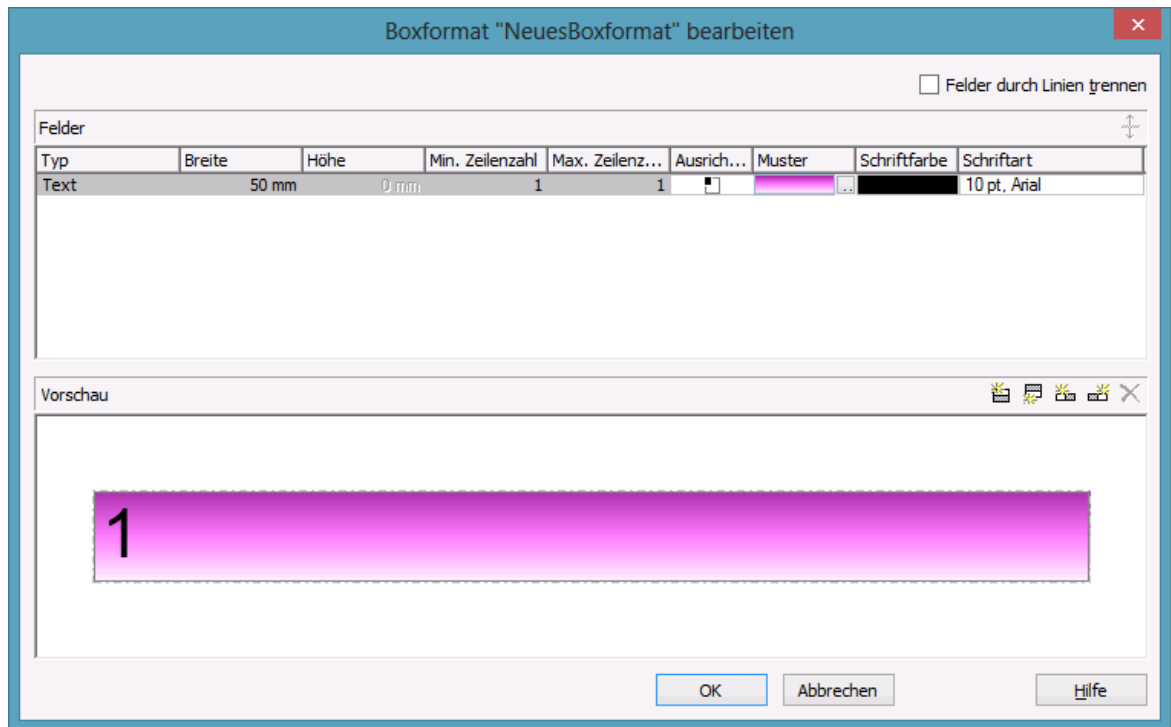
Für jede Box können Sie ein Boxformat wählen. Die Boxformate können Sie selbst festlegen.

Im Dialog **Boxformate verwalten** können Sie Boxformate hinzufügen, kopieren, bearbeiten und löschen. Sie erreichen dieses Dialogfeld, indem Sie im Dialogfeld **Boxen verwalten** im Feld **Boxformat** auf die **Bearbeiten**-Schaltfläche klicken.



Im Dialog **Boxformat bearbeiten** können Sie das Boxformat festlegen. Sie erreichen dieses Dialogfeld, indem Sie im Dialogfeld **Boxformate verwalten** auf die Schaltfläche **Boxformat bearbeiten** klicken.

72 Wichtige Konzepte: Boxen



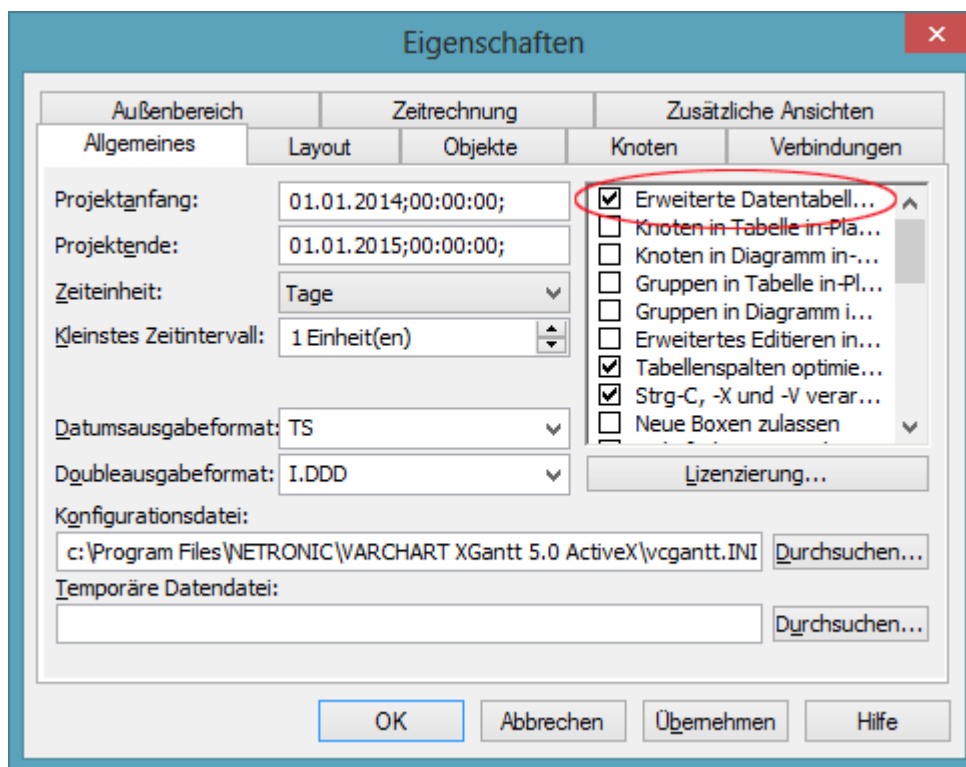
Sie können hier festlegen, ob die Felder der Box durch Linien getrennt werden sollen. Außerdem können Sie für jedes Feld der Box folgendes festlegen:

- Feldtyp (Text oder Grafik)
- Breite und Höhe
- Zeilenzahl
- Ausrichtung
- Füllfarbe und -muster
- Schriftattribute

3.2 Datentabellen

Zur Darstellung von Gantt-Diagrammen verwendet VARCHART XGantt zwei Standard-Datentabellen für Knoten und Verbindungen, deren Felder individuell festgelegt werden können. In VARCHART XGantt 4.0 wurde dieses Konzept erweitert. Es können jetzt bis zu 90 Datentabellen vereinbart und zusätzlich Beziehungen in Form von 1:n Relationen zwischen diesen Tabellen definiert werden. Indem - wie in Datenbanken üblich - die Daten in voneinander abhängigen Datensätzen organisiert werden, werden Redundanzen vermieden und die integrierte Ressourcenplanung mit den notwendigen Daten versorgt.

Aus Kompatibilitätsgründen mit bestehenden Anwendungen arbeitet VARCHART XGantt standardmäßig im bisherigen Modus. Erst durch Aktivieren der entsprechenden Option zur Design- oder Laufzeit können die neuen Möglichkeiten genutzt werden. Auf der Eigenschaftenseite **Allgemeines** finden Sie dazu die Option **Erweiterte Datentabellen**:

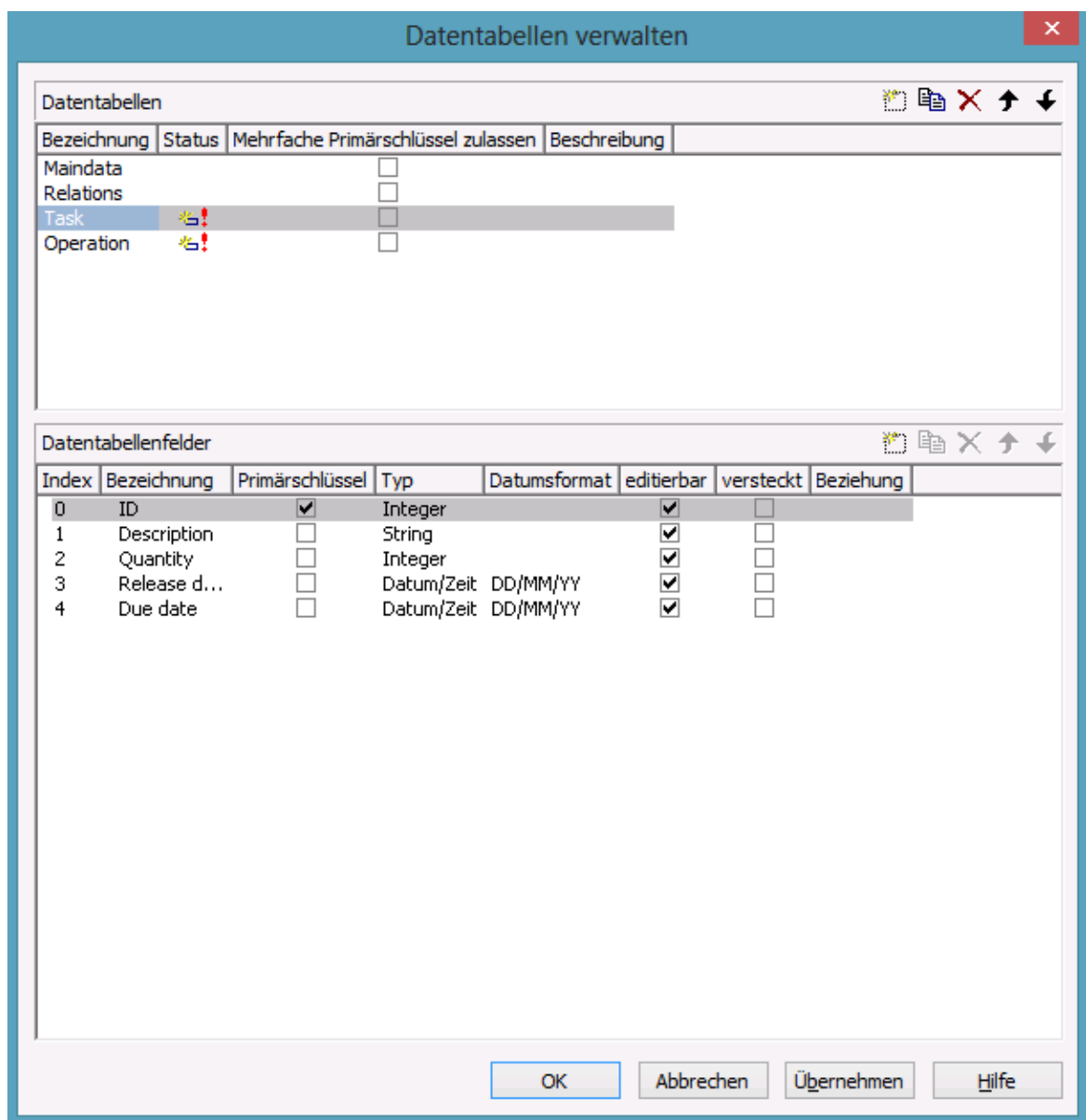


Alternativ können die erweiterten Datentabellen statt zur Designzeit auch zur Laufzeit über die API eingeschaltet werden, indem die Eigenschaft **ExtendedDataTables** des Objekts **VcGantt** auf **True** gesetzt wird.

> Datentabellen verwalten

Standardmäßig sind die beiden Datentabellen **Maindata** und **Relations** vorhanden. Auf der Eigenschaftenseite **Objekte** gelangen Sie über die Schaltfläche **Datentabellen...** in den Dialog **Datentabellen verwalten**. Nur im erweiterten Modus ist es möglich, neue Datentabellen anzulegen. Im Bild unten wurden die Datentabellen **Task** und **Operation** durch Klick auf  im Bereich **Datentabellen** neu angelegt.

Im Bereich **Datentabellenfelder** können Sie neue Felder anlegen , bestehende Felder löschen  oder Felder kopieren .



Die Spalte **Index** besitzt eine zentrale Bedeutung, denn der Zugriff auf die Inhalte der Datenfelder innerhalb eines Datensatzes geschieht ausschließlich über den Index. Wenn Sie die Reihenfolge der Felder ändern, nachdem

bereits Programmcode existiert, dann müssen Sie den Programmcode für den Datenfeldzugriff ebenso anpassen.

Die Änderung des Datentyps eines Feldes kann bewirken, dass bereits definierte Formate und Layer geändert werden müssen, um den Zugriff mit dem richtigen Datentyp sicherzustellen.

Mit der Primärschlüssel-Eigenschaft kennzeichnen Sie das Feld, dessen Werte die Datensätze einer Datentabelle eindeutig unterscheidbar machen. Der Primärschlüssel kann auch aus mehreren Feldern - *jedoch nicht mehr als 3* - bestehen. Eine ausführliche Beschreibung zur Handhabung von zusammengesetzten Primärschlüsseln finden sie im Kapitel **Datentabellen verwalten**. Für jede Datentabelle, auf die in einer Verknüpfung verwiesen wird, ist die Festlegung eines Primärschlüsselfeldes zwingend erforderlich.

Die Verknüpfung zweier Tabellen ist sinnvoll, wenn eine inhaltliche 1:n Beziehung zwischen den Tabellen besteht und von jedem Detaildatensatz direkt auf ein Datenfeld im Hauptdatensatz Bezug genommen werden soll.

Zwischen zwei Tabellen A und B ist derzeit nur eine einzige 1:n-Beziehung möglich; es kann sich also kein zweites Feld der Tabelle B auf den Primärschlüssel in A beziehen. Allerdings kann sich ein Feld aus einer weiteren Tabelle C auf den Primärschlüssel in A beziehen.

Hinweis: Wird eine Datentabelle mit einem zusammengesetzten Primärschlüssel in einer Beziehung verwendet, muss auch die Beziehung entsprechend definiert werden. Andernfalls ist eine eindeutige Anbindung nicht möglich. Ist die Beziehung nicht korrekt definiert - was weder an der API noch im Dialog **Datentabellen verwalten** überprüft wird, findet **keine** Datensatzanbindung statt, was zum Ereignis **OnDataRecordNotFound** führt.

In dem Beispiel wird eine Verknüpfung zwischen den Tabellen **Operation** und **Task** durch die Angabe von **Task:ID** in der Spalte **Beziehung** vereinbart.

76 Wichtige Konzepte: Datentabellen

Datentabellen verwalten

Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata		☐	
Relations		☐	
Task		☐	
Operation		☐	

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt	Beziehung
0	ID	☒	String		☒	☐	
1	TaskID	☐	Integer		☒	☐	Task.ID

OK Abbrechen Übernehmen Hilfe

Tabelle Task:

ID	Description	Quantity	Release date	Due date
1	Task 1	10	12.05.07	20.05.07
2	Task 2	20	01.06.07	15.06.07

Tabelle Operation:

ID	TaskID	Description	Start	End
1	1	Operation 1	12.05.07	14.05.07
2	1	Operation 2	15.05.07	19.05.07

ID	TaskID	Description	Start	End
3	2	Operation 3	01.06.07	05.06.07
4	2	Operation 4	05.06.07	11.06.07
5	2	Operation 5	11.06.07	15.06.07

Code-Beispiel

```

Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByName("Task")

dataTable.DataRecordCollection.Add ("1;Task 1;10;12.05.2007;20.05.2007")
dataTable.DataRecordCollection.Add ("2;Task 2;10;01.06.2007;15.06.2007")

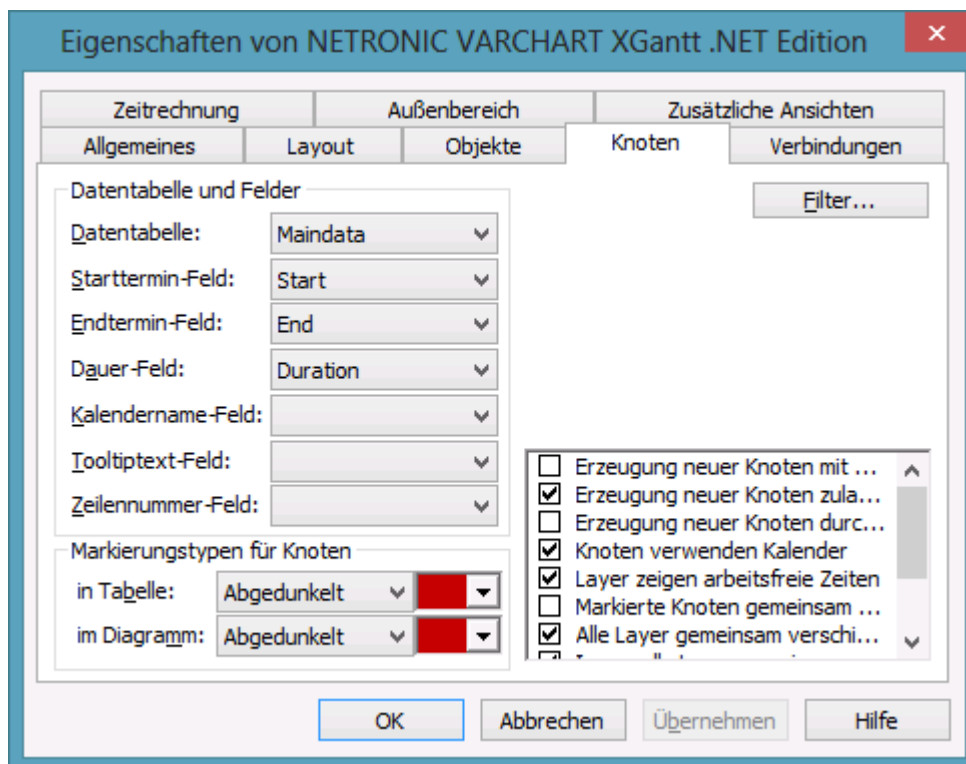
Set dataTable = dataTableCltn.DataTableByName("Operation")
dataTable.DataRecordCollection.Add ("1;1;Operation
1;12.05.2007;14.05.2007")
dataTable.DataRecordCollection.Add ("2;1;Operation
2;15.05.2007;19.05.2007")
dataTable.DataRecordCollection.Add ("3;2;Operation
3;01.06.2007;05.06.2007")
dataTable.DataRecordCollection.Add ("4;2;Operation
4;05.06.2007;11.06.2007")
dataTable.DataRecordCollection.Add ("5;2;Operation
5;11.06.2007;15.06.2007")

VcGantt1.EndLoading

```

In Abhängigkeit davon, welche Datentabelle unter **Datentabelle** bei der Eigenschaftenseite **Knoten** angegeben wird, ergibt sich eine andere Basis, aus der die grafische Darstellung der Knoten hergeleitet wird. Beim interaktiven Anlegen von Knoten ist es diejenige Tabelle, der automatisch neue Datensätze hinzugefügt werden. Die in der Visualisierung sichtbaren Zeilen werden von dem aktiven Knotenfilter, der Gruppierung und von Darstellungsoptionen beeinflusst.

78 Wichtige Konzepte: Datentabellen



Dies ist das Ergebnis im Tabellenteil des Gantt-Diagramms, wenn die Tabelle **Operation** als Basis genommen wird. Die Einträge für Description, Quantity und Due date stammen aus der Haupttabelle **Task**.

Description	Quantity	Due date	Operation
Task1	10	20.05.07	Operation1
Task1	10	20.05.07	Operation2
Task2	20	15.06.07	Operation3
Task2	20	15.06.07	Operation4
Task2	20	15.06.07	Operation5

Die sichtbare Tabelle in XGantt besteht nur aus zwei Einträgen, wenn statt der Tabelle **Operation** die Tabelle **Task** herangezogen wird.

ID	Description	Quantity	Due date	Operation
1	Task 1	10	20.05.07	
2	Task 2	20	15.06.07	

In der VARCHART XGantt Version 4.0 sind aufgrund der erweiterten Datentabellen eine Reihe neuer Objekttypen entstanden, die bestehende Objekttypen ablösen werden. Aus Gründen der Kompatibilität sind in dieser Version die bisherigen Objekttypen noch enthalten. Für neue Anwendungen

und beim Aktualisieren von bestehenden Anwendungen sollten nur noch die neuen Objekttypen eingesetzt werden.

Bisher	Ab Version 4.0
VcDataDefinition	VcDataTable
VcDefinitionTable	VcDataTableFieldCollection
VcDefinitionField	VcDataTableField
	VcDataRecord

> Datensätze anlegen und ändern

Nach der Definition der Datentabellenfelder können Sie einer Tabelle über die API Datensätze hinzufügen. Es gibt zwei Wege, auf denen die Datensätze mit Daten gefüllt werden können. Der normale und empfehlenswerte Weg besteht in der Vereinbarung eines Arrays vom Datentyp Variant, wobei dessen Elementanzahl auf die Anzahl der Datentabellenfelder gesetzt wird.

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection

Dim dataRecVal() As Variant
Dim dataRec1 As VcDataRecord
Dim dataRec2 As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection

ReDim dataRecVal(dataTable.DataTableFieldCollection.Count)

dataRecVal(0) = 1
dataRecVal(1) = "Node 1"
dataRecVal(2) = DateSerial(2007, 1, 8)
dataRecVal(4) = 8
```

Ein neuer Datensatz wird durch die Methode **Add()** des Objekts **DataRecordCollection** erzeugt, der als Parameter das Variant-Array mitgegeben wird.

Code-Beispiel

```
Set dataRec1 = dataRecCltn.Add(dataRecVal)
```

Der zweite Weg besteht in der Verwendung einer Zeichenkette, bei der die Datenwerte durch Semikolon getrennt aneinander gereiht werden.

Code-Beispiel

```
Set dataRec2 = dataRecCltn.Add("2;Node 2;15.01.07;;9")
```

80 Wichtige Konzepte: Datentabellen

Wenn ein Semikolon selber im Datenwert enthalten ist, dann muss die Zeichenkette in doppelte Hochkommata eingeschlossen werden.

Code-Beispiel

```
Set dataRec2 = dataRecCltn.Add("2; ""Node 2;"";15.01.07;;9")
```

Der Verweis auf ein Datensatzobjekt kann mit Hilfe des Primärschlüssels über die Methode **DataRecordByID ()** schnell gefunden werden.

Code-Beispiel

```
Set dataRec1 = dataRecCltn.DataRecordByID(1)
Set dataRec2 = dataRecCltn.DataRecordByID(2)
```

Einzelne Datenfeldinhalte eines Datensatzes können Sie leicht über die indizierte Eigenschaft **DataField()** ändern. Mit der Eigenschaft **AllData** lassen sich alle Datenfeldinhalte eines Datensatzes auf einmal ersetzen.

Code-Beispiel

```
dataRec1.DataField(Main_ID) = 1
dataRec1.DataField(Main_Name) = "Activity X"
dataRec1.DataField(Main_Start) = DateSerial(2007, 1, 4)
dataRec1.DataField(Main_Duration) = 12
dataRec1.UpdateDataRecord
```

```
dataRec2.AllData = "2;Activity Y;18.01.07;;5"
dataRec2.UpdateDataRecord
```

Jede Änderung in einem Datensatz wird erst darstellungswirksam, wenn die Methode **Update()** des Objekts **DataRecord** aufgerufen wird.

Das Auslesen der Werte mit **Alldata** eignet sich für die schnelle Anzeige aller Datenwerte während der Programmentwicklung und zum leichten Übertragen des Datensatzinhalts in den Datensatz einer anderen Tabelle.

Code-Beispiel

```
Dim content As String
content = dataRec1.AllData & vbCr & dataRec2.AllData & vbCr &
dataRec1.DataField(Main_Name)
MsgBox (content)
```

Tipp:

Um die Lesbarkeit bei den Datenfeldzugriffen zu erhöhen, können Sie globale Konstanten definieren, deren Namen sprechender sind als die Indexzahlen.

Hier noch einmal das gesamte Programmfragment im Zusammenhang:

Code-Beispiel

```
Const Main_ID = 0
Const Main_Name = 1
Const Main_Start = 2
Const Main_Duration = 4
```

```

'...

Dim dataRec1 As VcDataRecord
Dim dataRec2 As VcDataRecord

Dim content As String

VcGantt1.TimeScaleEnd = DateSerial(2008, 1, 1)
VcGantt1.TimeScaleStart = DateSerial(2007, 1, 1)

VcGantt1.ExtendedDataTablesEnabled = True
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection

ReDim dataRecVal(dataTable.DataTableFieldCollection.Count)

dataRecVal(Main_ID) = 1
dataRecVal(Main_Name) = "Node 1"
dataRecVal(Main_Start) = DateSerial(2007, 1, 8)
dataRecVal(Main_Duration) = 8
Set dataRec1 = dataRecCltn.Add(dataRecVal)

dataRecCltn.Add("2;Node 2;15.01.07;;9")

VcGantt1.EndLoading

'...

Set dataRec1 = dataRecCltn.DataRecordByID(1)
Set dataRec2 = dataRecCltn.DataRecordByID(2)

dataRec1.DataField(Main_ID) = 1
dataRec1.DataField(Main_Name) = "Activity X"
dataRec1.DataField(Main_Start) = DateSerial(2007, 1, 4)
dataRec1.DataField(Main_Duration) = 12
dataRec1.UpdateDataRecord

dataRec2.AllData = "2;Activity Y;18.01.07;;5"
dataRec2.UpdateDataRecord

content = dataRec1.AllData & vbCr & dataRec2.AllData & vbCr &
dataRec1.DataField(Main_Name)
MsgBox (content)

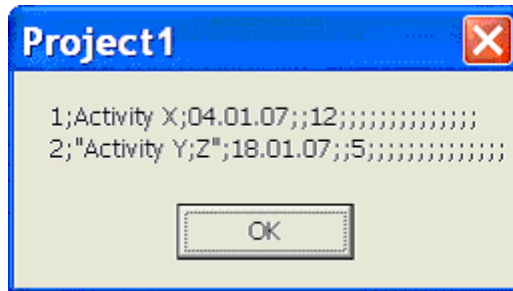
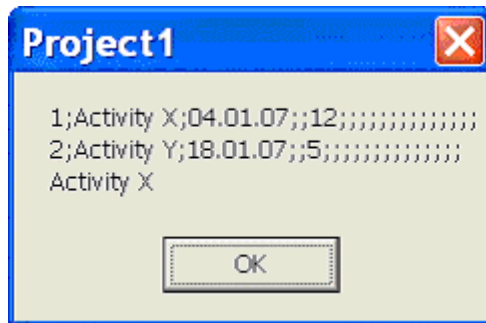
'...

dataRec2.AllData = "2;""Activity Y;Z"";18.01.07;;5"
dataRec2.UpdateDataRecord
content = dataRec1.AllData & vbCr & dataRec2.AllData
MsgBox (content)

```

Erzeugte Ausgabe:

82 Wichtige Konzepte: Datentabellen



3.3 Datumsangaben und Zeitumstellung

Datumsangaben in VARCHART Komponenten sind immer bezogen auf die im System eingestellte Zeitzone. Es ist nicht möglich, Datumsangaben aus anderen Zeitzonen anzugeben, d.h. diese müssen vor der Übergabe an eine VARCHART Komponente für die eingestellte Zeitzone umgerechnet werden. Dabei beachtet die VARCHART Komponente automatisch die im System vorhandenen Informationen zu Beginn und Ende der Sommerzeit.

Damit eine VARCHART-Komponente die Umschaltzeitpunkte vom System mitgeteilt bekommt, ist es wichtig, dass die Option **Uhr automatisch auf Sommer-/Winterzeit umstellen** gesetzt ist, wie im Bild gezeigt. Der Dialog ist im Windows-Betriebssystem unter "Start" > "Einstellungen" > "Systemsteuerung" > "Datum und Uhrzeit" erreichbar oder über einen Doppelklick auf die Uhrzeit in der Taskleiste.



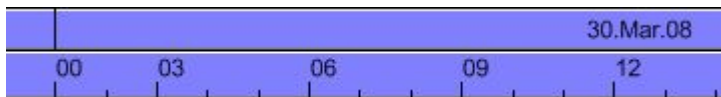
Eine VARCHART-Komponente nimmt bei der Umschaltung den Startzeitpunkt, den Endzeitpunkt inkl. Stunde, Monat und Tag, den das System als Regel mitteilt. Das heißt aber auch, dass die Umschaltzeiten für die Jahre vor und nach dem aktuellen Jahr extrapoliert werden, sodass etwaige Abweichungen, die für vorherige Jahre galten bzw. für kommende Jahre gelten werden, nicht berücksichtigt werden können, weil sie dem Betriebssystem ebenfalls nicht bekannt sind. Beispielsweise wurde die Sommerzeit in den USA vor wenigen Jahren um Wochen am Beginn und Ende verlängert. Da dem System hier nur die aktuelle Regelung bekannt ist, werden

Datumsangaben in der betroffenen Vergangenheit hier systembedingt falsch interpretiert.

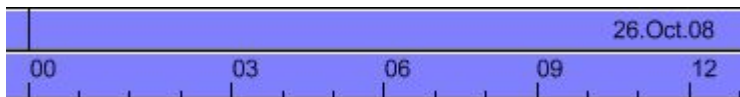
Augenblicklich können VARCHART Komponenten bei der Sommerzeit nur eine Zeitdifferenz zur Winterzeit von exakt einer Stunde berücksichtigen. Außerdem darf die Umschaltung nur zur vollen Stunde erfolgen.

Da eine VARCHART-Komponente die Datumsangaben immer in lokaler Zeit entgegennimmt und ausgibt, gibt es am Beginn der Sommerzeit eine nicht erlaubte Stunde und am Ende der Sommerzeit zwei Stunden mit derselben Zahl, die derzeit bei Übergabe, bei Rückgabe und bei Ausgabe nicht unterschieden werden kann.

In der Zeitskala wird die Umschaltzeit sichtbar, wenn diese stundengenau anzeigt.



Umschaltzeit im Bereich zwischen 0 und 3 Uhr im Frühling (1 Stunde fehlt)



Umschaltzeit im Bereich zwischen 0 und 3 Uhr im Herbst (1 Stunde doppelt)

> Neues Standarddatum ab Version 4.3

Wenn in einer VARCHART-Komponente ein Datum erfragt wird, das nicht existiert, wurde bis zur Version 4.3 das Datum **31.12.1899 00:00:00** zurückgegeben. Ab Version 4.3 wird das Datum **01.01.0001 00:00:00** zurückgeliefert.

Das kann unter bestimmten Konstellationen die Ausnahme **Parameter außerhalb des zulässigen Bereichs** (ArgumentOutOfRangeException exception) zur Folge haben, was durch eine Ausnahmebehandlung abgefangen werden kann.

Wenn Sie ein Datum beispielsweise mit DateTimePicker-Controls von .NET innerhalb Ihrer Anwendung verwalten und wenn Sie versuchen ein "leeres" Datum darin anzuzeigen, wurde vor Version 4.3 der Zeitpunkt **31.12.1899 00:00:00** angezeigt. Der Default ab Version 4.3, also das Datum **01.01.0001 00:00:00**, kann aber mit den Standard-Einstellungen des DateTimePickers nicht angezeigt werden und es wird die Ausnahme **ArgumentOutOfRangeException** ausgelöst.

Ihr Programm sollte also in jedem Fall darauf reagieren, d.h. Ausnahmebehandlungen müssen programmiert werden, andernfalls kann es zu einer unbehandelten Ausnahme und damit zu einem unerwarteten Programmende kommen.

3.4 Der Einsatz des Kalenders

Ein Kalender besteht aus der lückenlosen Abfolge von Arbeits- und Nichtarbeitszeiten. Bei einem Kalender mit variablem Profil (Schichtkalender) reißen sich unterschiedliche Arbeitszeitabschnitte wie zum Beispiel Früh-, Spät- und Nachtschichten wiederholt aneinander. Der Kalender selbst besitzt keine visuelle Darstellung, er besteht lediglich aus der logischen Unterscheidung zwischen Arbeits- und Nichtarbeitszeiten. Sichtbar werden kann ein Kalender nur mithilfe des Objekts **CalendarGrid**.

In VARCHART XGantt wird ein Kalender dazu benutzt, um über die Dauer die Anfangs- oder Endtermine der Vorgänge zu berechnen. Der vordefinierte Basiskalender mit dem Namen **BaseCalendar** wird für alle Berechnungen herangezogen, solange keine anderweitigen Festlegungen getroffen werden. Im Basiskalender sind die Wochentage von Montag bis Freitag als Arbeitszeit und die Wochenenden Samstag und Sonntag als arbeitsfrei vereinbart. Bei Bedarf kann der Basiskalender verändert werden.

Einen Kalender definieren

Ein Kalender kann zur Entwurfszeit über die Eigenschaftenseiten oder zur Laufzeit über die Programmierschnittstelle spezifiziert werden. In diesem Kapitel werden die grundlegenden Konzepte im Umgang mit Kalendern aus Programmiersicht erläutert und anhand von Programmbeispielen in C# verdeutlicht. Die Festlegung von Kalendern über die Eigenschaftenseiten wird im Kapitel **Eigenschaftenseiten und Dialogfelder** näher beschrieben.

Das Steuerelement **VcGantt** besitzt ein Objekt vom Typ **VcCalendarCollection**, das für die Verwaltung der Kalender verantwortlich ist. Es besitzt die gleichen Verwaltungsfunktionen wie alle Auflistungen (Collections) in VARCHART XGantt. Der vordefinierte Kalender mit dem Namen **BaseCalendar** sowie alle anderen zur Entwurfszeit erstellten Kalender werden automatisch in die Auflistung aufgenommen.

Ein neuer Kalender wird über die Methode **Add** des Objekts **CalendarCollection** angelegt. Als Parameter benötigt die Methode einen eindeutigen Namen zur späteren Identifizierung des Objekts. Der neue Kalender besteht zunächst nur aus Arbeitszeit.

Hinweis: Ein Kalender muss zu jedem Zeitpunkt mindestens eine Arbeitszeitspanne umfassen, da ein Kalender, der nur aus Nichtarbeitszeit besteht, nicht möglich ist.

Damit die Ergebnisse in den Abbildungen der Gantt-Diagramme nachvollziehbar sind, wird für die Zeitskala in den Beispielen ein fester Darstellungs-

86 Wichtige Konzepte: Der Einsatz des Kalenders

Zeitraum vom 1.1.2011 bis zum 31.12.2011 vereinbart. Ein Kalender wird im Hintergrund des Gantt-Diagramms erst sichtbar, wenn er zum aktiven Kalender der Kollektion gemacht wird:

Code-Beispiel

```
'Darstellung des neuen Kalenders
Dim calendar As VcCalendar
VcGantt1.TimeScaleEnd = "01.01.2012"
VcGantt1.TimeScaleStart = "01.01.2011"
Set calendar = VcGantt1.CalendarCollection.Add("CompanyCalendar1")
VcGantt1.CalendarCollection.Active = calendar
```

January 2011																														
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	

Falls der bereits standardmäßig vorhandene Kalender wieder wirksam werden soll, geschieht dies durch folgende Zuweisung:

Code-Beispiel

```
' Darstellung des Standardkalenders
Dim calendar As VcCalendar
Set calendar =
VcGantt1.CalendarCollection.CalendarByName("BaseCalendar")
VcGantt1.CalendarCollection.Active = calendar
```

January 2011																															
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Im nächsten Beispiel wird gezeigt, wie für einen neuen Kalender das Arbeitszeitprofil durch Angabe von **Intervallen** festgelegt wird. Der 1. Januar 2011 sowie die Tage im Zeitraum vom 6.01.2011 bis 20.01.2011 sollen mit Ausnahme der beiden Tage 10.01.2011 und 11.01.2011 als Nichtarbeitszeit definiert werden:

Code-Beispiel

```
'Nichtarbeitszeiten definieren
VcGantt1.TimeScaleEnd = "01.01.2012"
VcGantt1.TimeScaleStart = "01.01.2011"

Dim calendar As VcCalendar
Set calendar = VcGantt1.CalendarCollection.Add("CompanyCalendar1")
```

```
VcGantt1.CalendarCollection.Active = calendar

Dim interval As VcInterval
Set interval = calendar.IntervalCollection.Add("NewYear")
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = "01.01.2011"
interval.EndDateTime = "02.01.2011"

Set interval = calendar.IntervalCollection.Add("NonworkPeriod")
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = "06.01.2011"
interval.EndDateTime = "21.01.2011"

Set interval = calendar.IntervalCollection.Add("WorkPeriod")
interval.CalendarProfileName = "<WORK>"
interval.StartDateTime = "11.01.2011"
interval.EndDateTime = "13.01.2011"
VcGantt1.CalendarCollection.Update
```

January 2011																															
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Optisch sind die Nichtarbeitszeiten an den Flächen mit der hellgrauen Farbe zu erkennen. Da Arbeitszeiten keine standardmäßige Einfärbung besitzen, bleibt der weiße Hintergrund des Diagramms sichtbar. In nächsten Schritt sollen die Arbeitszeiten hellgelb und die arbeitsfreien Zeiten hellblau eingefärbt werden. Dies bewirken grafische Attribute, die bei den Intervallen angegeben werden können.

Code-Beispiel

```
'Einfärben von Intervallen
VcGantt1.TimeScaleEnd = "01.01.2012"
VcGantt1.TimeScaleStart = "01.01.2011"

Dim calendar As VcCalendar
Set calendar = VcGantt1.CalendarCollection.Add("CompanyCalendar1")
VcGantt1.CalendarCollection.Active = calendar

VcGantt1.TimeScaleCollection.FirstTimeScale.Section(0).CalendarGridEx(0)
.UseGraphicalAttributesOfIntervals = True

Dim interval As VcInterval

Set interval = calendar.IntervalCollection.Add("Work")
interval.CalendarProfileName = "<WORK>"
interval.BackColorAsARGB = &HFFFFFFE0
interval.UseGraphicalAttributes = True

Set interval = calendar.IntervalCollection.Add("NewYear")
interval.CalendarProfileName = "<NONWORK>"
```

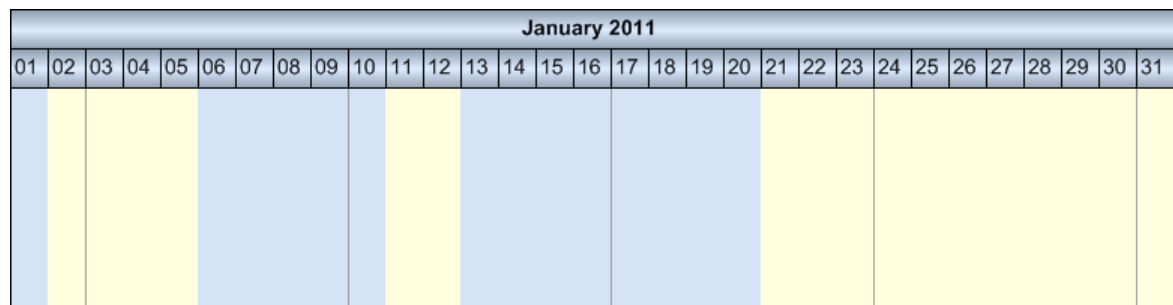
88 Wichtige Konzepte: Der Einsatz des Kalenders

```
interval.StartDateTime = "01.01.2011"  
interval.EndDateTime = "02.01.2011"  
interval.BackColorAsARGB = &HFFD4E3F5  
interval.UseGraphicalAttributes = True
```

```
Set interval = calendar.IntervalCollection.Add("NonworkPeriod")  
interval.CalendarProfileName = "<NONWORK>"  
interval.StartDateTime = "06.01.2011"  
interval.EndDateTime = "21.01.2011"  
interval.BackColorAsARGB = &HFFD4E3F5  
interval.UseGraphicalAttributes = True
```

```
Set interval = calendar.IntervalCollection.Add("WorkPeriod")  
interval.CalendarProfileName = "<WORK>"  
interval.StartDateTime = "11.01.2011"  
interval.EndDateTime = "13.01.2011"  
interval.BackColorAsARGB = &HFFFFFFE0  
interval.UseGraphicalAttributes = True
```

```
VcGantt1.CalendarCollection.Update
```



Das folgende Beispiel zeigt, wie eine Woche mit Arbeitszeit von Montag bis Freitag und freien Wochenenden vereinbart wird. Die bisher vorgestellten Möglichkeiten reichen dazu noch nicht aus; es muss ein Objekt vom Typ **VcCalendarProfile** zur Hilfe genommen werden.

Hinweis: Objekte vom Typ **VcCalendarProfile** können in **VARCHART XGantt** auf globaler oder lokaler Ebene angelegt werden. Lokale Calendar-Profile-Objekte sind nur in dem Kalender verwendbar, in dem sie definiert worden sind, während die globalen Objekte gleichzeitig in mehreren Kalendern eingesetzt werden können. In den Beispielen werden ausschließlich lokale CalendarProfile-Objekte verwendet. Funktionell bestehen keine Unterschiede zu den globalen Objekten. Wenn unter dem gleichen Namen sowohl ein globales als auch lokales Profil angelegt wurde, kann in dem betreffenden Kalender nur auf das lokale Profile zugegriffen werden; das globale Profil ist unzugänglich.

Mit einem **CalendarProfile** vom Typ **vcWeekProfile** werden die Arbeitszeiten und Nichtarbeitszeiten der Wochentage beschrieben. Das Wochenprofil wird erst wirksam, wenn es der Intervallauflistung (**IntervalCollection**) des Kalenders hinzugefügt wird. Die Angabe von **StartDateTime** und **EndDateTime** beim Kalenderintervall kann entfallen, weil die Setzung ohne

zeitliche Einschränkung für den gesamten Kalenderzeitraum gelten soll. Die Kalenderprofile mit den vordefinierten Namen **<WORK>** und **<NON-WORK>** haben eine feste Bedeutung. Sie werden dazu benutzt, um Arbeitszeit und Nichtarbeitszeit zuordnen zu können.

Code-Beispiel

```
'Definition eines Wochenprofils
Dim calendar As VcCalendar
Dim interval As VcInterval
Dim calendarProfile As VcCalendarProfile

Set calendar = VcGantt1.CalendarCollection.Add("CompanyCalendar1")
VcGantt1.CalendarCollection.Active = calendar
Set calendarProfile =
calendar.CalendarProfileCollection.Add("WeekProfile")
calendarProfile.Type = vcWeekProfile

VcGantt1.TimeScaleCollection.FirstTimeScale.Section(0).CalendarGridEx(0)
.UseGraphicalAttributesOfIntervals = True

Set interval = calendarProfile.IntervalCollection.Add("Mo-Fr")
interval.CalendarProfileName = "<WORK>"
interval.StartWeekday = vcMonday
interval.EndWeekday = vcFriday

Set interval = calendarProfile.IntervalCollection.Add("Sa")
interval.CalendarProfileName = "<NONWORK>"
interval.BackColorAsARGB = &HFFFFFF69F
interval.StartWeekday = vcSaturday
interval.EndWeekday = vcSaturday

Set interval = calendarProfile.IntervalCollection.Add("So")
interval.CalendarProfileName = "<NONWORK>"
interval.BackColorAsARGB = &HFFFB3AA
interval.StartWeekday = vcSunday
interval.EndWeekday = vcSunday

Set interval = calendar.IntervalCollection.Add("StandardWeek")
interval.CalendarProfileName = "WeekProfile"
```

Wenn Arbeitszeiten von Nichtarbeitszeiten innerhalb eines Tages unterschieden werden sollen, dann wird zusätzlich ein Tagesprofil benötigt, in dem die exakten Arbeitszeitintervalle angegeben werden: z.B. von 8.00 Uhr bis 12.00 Uhr und von 13.00 Uhr bis 17.00 Uhr. Da ein neu angelegtes Tagesprofil nur aus Arbeitszeit besteht, müssen die Abweichungen davon als Nichtarbeitszeitintervalle angegeben werden.

Code-Beispiel

```
'Definition eines Tagesprofils
Dim interval As VcInterval
Dim calendarProfile As VcCalendarProfile

Set calendarProfile =
calendar.CalendarProfileCollection.Add("DayProfile")
calendarProfile.Type = vcDayProfile
```

90 Wichtige Konzepte: Der Einsatz des Kalenders

```
Set interval = calendarProfile.IntervalCollection.Add("Interval_1")
' 00:00-8:00
interval.CalendarProfileName = "<NONWORK>"
interval.StartTime = "1.1.2011 0:00"
interval.EndTime = "1.1.2011 8:00"

Set interval = calendarProfile.IntervalCollection.Add("Interval_2")
' 12:00-13:00
interval.CalendarProfileName = "<NONWORK>"
interval.StartTime = "1.1.2011 12:00"
interval.EndTime = "1.1.2011 13:00"

Set interval = calendarProfile.IntervalCollection.Add("Interval_3")
' 17:00-24:00
interval.CalendarProfileName = "<NONWORK>"
interval.StartTime = "1.1.2011 17:00"
interval.EndTime = "1.1.2011 00:00"
```

Die Angabe der Uhrzeit erfolgt durch das Objekt **DateTime**, bei dem der Datumsanteil ignoriert wird, weil er in diesem Zusammenhang ohne Bedeutung ist. Das Datum muss nur mit angegeben werden, damit alle notwendigen Parameter im Aufruf des Konstruktors belegt sind. Im **Interval_3** ist zu beachten, dass statt 24 Uhr, was im Objekt **DateTime** als Stundenangabe nicht akzeptiert wird, 0 Uhr zu spezifizieren ist.

Jährlich wiederkehrende Tage wie zum Beispiel **Neujahr** am 1.1 oder **Weihnachten** am 25.12. und 26.12. werden über ein Jahreskalenderprofil festgelegt.

Code-Beispiel

```
'Profil der festen Feiertage vereinbaren
Dim calendarProfile As VcCalendarProfile
Dim interval As VcInterval

Set calendarProfile =
calendar.CalendarProfileCollection.Add("YearProfile")
calendarProfile.Type = vcYearProfile

Set interval = calendarProfile.IntervalCollection.Add("New Year")
interval.CalendarProfileName = "<NONWORK>"
interval.DayInStartMonth = 1
interval.StartMonth = vcJanuary
interval.DayInEndMonth = 1
interval.EndMonth = vcJanuary
Call SetAppearanceForHolidays(interval)

Set interval = calendarProfile.IntervalCollection.Add("Christmas")
interval.CalendarProfileName = "<NONWORK>"
interval.DayInStartMonth = 25
interval.StartMonth = vcDecember
interval.DayInEndMonth = 26
interval.EndMonth = vcDecember
Call SetAppearanceForHolidays(interval)
```

Um zu vermeiden, dass die für das gleichartige Aussehen der Feiertage notwendigen Setzungen wiederholt im Programmcode aufgeführt werden,

wurden die Anweisungen in einer Methode mit dem Namen **SetAppearanceForHolidays** gebündelt:

Code-Beispiel

```
'Methode zur Festlegung der visuellen Darstellung der Feiertage

Private Sub SetAppearanceForHolidays(ByVal interval As VcInterval)
    interval.BackColorAsARGB = &HFFFFFFA4A4
    interval.Pattern = vcWeavePattern
    interval.PatternColorAsARGB = &HFF404040
    interval.LineColor = &HFF808080
    interval.LineThickness = 1
    interval.LineType = vcSolid
    interval.UseGraphicalAttributes = True
End Sub
```

Hinweis: Die Eigenschaften zur Farbgestaltung werden nur bei den Intervallen wirksam, deren CalendarProfileName auf eine der beiden vordefinierten Begriffe **<WORK>** und **<NONWORK>** gesetzt ist. Zudem muss beim Intervall die Eigenschaft **UseGraphicalAttribute** auf **true** und beim Objekt **CalendarGrid** die Eigenschaft **UseGraphicalAttributesOfIntervals** auf **true** gesetzt sein.

Bewegliche Feiertage wie beispielsweise Ostern und davon abhängige Feiertage müssen für jedes Jahr einzeln berechnet werden und dem Kalender als feste Termine zugewiesen werden. Dazu eignet sich folgende Unterstützungsmethode:

Code-Beispiel

```
'Methode zur Berechnung der beweglichen Feiertage
Const AshWednesday = 0
Const GoodFriday = 1
Const EasterSunday = 2
Const EasterMonday = 3
Const FeastOfCorpusChristi = 4
Const AscensionOfChrist = 5
Const WhitSunday = 6
Const WhitMonday = 7
Const CentralEuropeanSummerTimeStart = 8
Const CentralEuropeanSummerTimeEnd = 9

Private Function calculateAnniversaryForYear(ByVal year As Integer,
ByVal specialDay As Integer) As Date
    Dim g As Integer
    Dim c As Integer
    Dim h As Integer
    Dim i As Integer
    Dim j As Integer
    Dim month As Integer
    Dim day As Integer
    Dim dayOffset As Integer

    g = year Mod 19
    c = year Mod 100
    h = (c - c / 4 - (8 * c + 13) / 25 + 19 * g + 15) Mod 30
```


92 Wichtige Konzepte: Der Einsatz des Kalenders

```
i = h - (h / 28) * (1 - (29 / (h + 1)) * ((21 - g) / 11))
j = (year + year / 4 + i + 2 - c + c / 4) Mod 7
month = 3 + (i - j + 40) / 44
day = i - j + 28 - 31 * (month / 4)
dayOffset = 0

Select Case specialDay
    Case AshWednesday
        dayOffset = -40
    Case GoodFriday
        dayOffset = -2
    Case EasterSunday
        dayOffset = 0
    Case EasterMonday
        dayOffset = 1
    Case AscensionOfChrist
        dayOffset = 39
    Case WhitSunday
        dayOffset = 49
    Case WhitMonday
        dayOffset = 50
    Case FeastOfCorpusChristi
        dayOffset = 60
    Case CentralEuropeanSummerTimeStart
        month = 3
        day = 31 - Weekday("31.3" + yearConvert + 1)
    Case CentralEuropeanSummerTimeEnd
        month = 10
        day = 31 - Weekday("31.10" + yearConvert + 1)
End Select

Dim tmpDate As Date
tmpDate = day & "." & month & "." & year
calculateAnniversaryForYear = tmpDate + dayOffset
End Function
```

In nächsten Schritt werden das Wochenprofil und das Jahrestageprofil dem Kalender als Intervalle hinzugefügt. Danach werden die beweglichen Feiertage berechnet und ebenso dem Kalender als Intervalle zugewiesen:

Code-Beispiel

```
'Wochenprofil, Jahrestageprofil und bewegliche Feiertage zusammenführen
Set interval = calendar.IntervalCollection.Add("Weekly_Pattern")
interval.CalendarProfileName = "WeekProfile"

Set interval = calendar.IntervalCollection.Add("Yearly_Pattern")
interval.CalendarProfileName = "YearProfile"

Dim startYear As Integer
Dim endYear As Integer

startYear = year(VcGantt1.TimeScaleStart)
endYear = year(VcGantt1.TimeScaleEnd)

Dim i As Integer
For i = startYear To endYear Step 1
    Set interval = calendar.IntervalCollection.Add("GoodFriday_" & i)
    interval.CalendarProfileName = "<NONWORK>"
    interval.StartDateTime = calculateAnniversaryForYear(i, GoodFriday)
    interval.EndDateTime = calculateAnniversaryForYear(i, EasterMonday)
```

```

interval.StartDateTime
Call SetAppearanceForHolidays(interval)

Set interval = calendar.IntervalCollection.Add("EasterMonday_" & i)
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = calculateAnniversaryForYear(i, EasterMonday)
interval.EndDateTime = interval.StartDateTime
Call SetAppearanceForHolidays(interval)

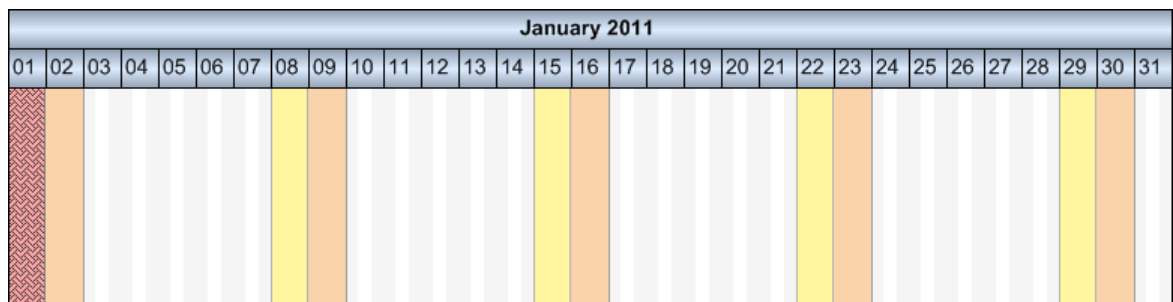
Set interval =
calendar.IntervalCollection.Add("FeastOfCorpusChristi_" & i)
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = calculateAnniversaryForYear(i,
FeastOfCorpusChristi)
interval.EndDateTime = interval.StartDateTime
Call SetAppearanceForHolidays(interval)

Set interval = calendar.IntervalCollection.Add("AscensionOfChrist_" &
i)
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = calculateAnniversaryForYear(i,
AscensionOfChrist)
interval.EndDateTime = interval.StartDateTime
Call SetAppearanceForHolidays(interval)

Set interval = calendar.IntervalCollection.Add("WhitMonday_" & i)
interval.CalendarProfileName = "<NONWORK>"
interval.StartDateTime = calculateAnniversaryForYear(i, WhitMonday)
interval.EndDateTime = interval.StartDateTime
Call SetAppearanceForHolidays(interval)
Next

VcGantt1.CalendarCollection.Update

```



Zusammenfassend noch einmal ein Überblick über die Schrittfolge, die nötig ist, um einen Kalender aufzubauen. Je nach Situation können gegebenenfalls einzelne Schritte entfallen:

1. Tagesprofile für unterschiedliche Arbeitstage anlegen
2. Wochenprofil unter Verwendung der Tagesprofile aufbauen
3. Jahrestageprofil festlegen
4. Wochenprofil und Jahrestageprofil der IntervalCollection des Kalenders zuordnen
5. Feste Termine der IntervalCollection des Kalenders zuweisen

Das Intervallobjekt ermöglicht es, Zeitintervalle zu definieren, die als Arbeitszeit oder Nichtarbeitszeit interpretiert werden. Die Unterscheidung zwischen den beiden Ausprägungen erfolgt durch die speziellen Setzungen **<WORK>** und **<NONWORK>** bei der Eigenschaft **CalendarProfileName**. Ein Intervall kann sich durch die Eigenschaft CalendarProfileName auch auf andere bereits definierte Profile beziehen und dessen Einstellungen übernehmen. Bei der Angabe der Eigenschaft ist zu beachten, dass abhängig vom Intervalltyp nur bestimmte Profiltypen zugeordnet werden können. Der Intervalltyp ist stets implizit durch den gewählten Profiltyp gegeben. Der vorgegebene Standardwert **vcDayProfile** für **VcCalendarProfile** kann zu Beginn, das heißt, bevor einzelne Intervalle definiert werden, durch eine entsprechende Zuweisung geändert werden.

Objekt	Gewählter Profiltyp	Zugeordneter Intervalltyp
VcCalendar		vcCalendarInterval
VcCalendarProfile	vcYearProfile	vcYearProfileInterval
	vcWeekProfile	vcWeekProfileInterval
	vcDayProfile	vcDayProfileInterval
	vcVariableProfile	vcVariableProfileInterval

Der Intervalltyp gibt vor, welche Art von Beschreibungsintervallen möglich sind. Zum Beispiel besitzt ein Tagesprofil immer Intervalle vom Typ **vcDayProfileInterval**.

Intervall-Typ	<WORK>	<NONWORK>	vcDayProfile	vcWeekProfile	vcYearProfile	vcVariableProfile
vcCalendarInterval	■	■	■	■	■	■
vcVariableProfileInterval	■	■	■	■	■	
vcYearProfileInterval	■	■	■	■		
vcWeekProfileInterval	■	■	■			
vcDayProfileInterval	■	■				

Bei den Kalenderprofilen wird unterschieden zwischen **Tagesprofilen**, **Wochenprofilen**, **Jahresprofilen** und **variablen Profilen**. In einem Tagesprofil können ausschließlich Zeitintervalle durch Uhrzeiten definiert werden, die innerhalb der Grenzen eines Tages liegen. In einem Wochenprofil wird

bestimmt, welche Tagesprofile an welchen Wochentagen zum Einsatz kommen. Ein Jahresprofil ordnet einzelnen oder mehreren zusammenhängenden Tagen ausgewählte Tagesprofile zu, die jährlich wiederkehrend zum gleichen Zeitpunkt (z.B. Neujahr) angewendet werden. Das variable Profil kann eine Abfolge von unterschiedlich gekennzeichneten, hintereinander liegenden Arbeitszeiten aufnehmen. In Abhängigkeit vom vorliegenden Intervalltyp **vcCalendarInterval**, **vcDayProfileInterval**, **vcWeekProfileInterval**, **vcYearProfileInterval**, **vcVariableProfileInterval** sind jeweils nur bestimmte Eigenschaften des Objektes relevant. Die nachfolgende Tabelle gibt Aufschluss über den Zusammenhang:

vcCalendar-Interval	vcYearProfile-Interval	vcWeekProfile-Interval	vcDayProfile-Interval	vcVariable-Interval
<u>StartDateTime</u>	<u>StartMonth</u>	<u>StartWeekday</u>	<u>StartTime</u>	Duration
<u>EndDateTime</u>	<u>EndMonth</u>	<u>EndWeekday</u>	<u>EndTime</u>	<u>TimeUnit</u>
	<u>DayInEndMonth</u>			
	<u>DayInStartMonth</u>			

Ein **CalendarInterval** beschreibt eine einmalige Zeitspanne innerhalb eines genau spezifizierten Zeitraums. Beispiel: 5.5.2010 11:30 Uhr bis 15.9.2010 17:00 Uhr.

Ein **YearProfileInterval** erlaubt die Vereinbarung eines jährlich wiederkehrenden Tages oder einer wiederkehrenden Zeitspanne. Beispiel: 1. 5. oder 24.12-26.12.

Ein **WeekProfileInterval** bezieht sich auf einzelne oder mehrere zusammenhängende Tage einer Woche. Beispiel: Samstag oder Montag bis Freitag.

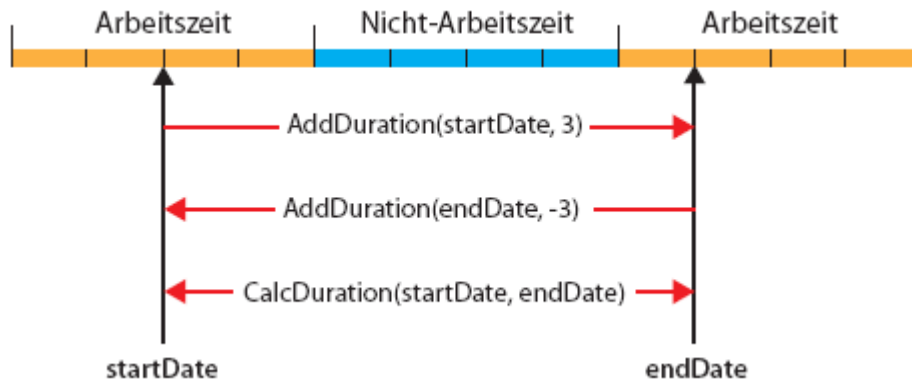
Ein **DayProfileInterval** bezieht sich auf die Angabe von Zeiten innerhalb eines Tages. Beispiel: 8.00 bis 17.00 Uhr.

Ein **VariableProfile** beschreibt eine Zeitspanne, ohne sich dabei auf einen bestimmten Termin zu beziehen. Die Einheit der Dauer, ob Tage, Stunden, Minuten oder Sekunden wird in der Eigenschaft **TimeUnit** des Intervall-Objekts angegeben. Beispiel: 4 Stunden.

Kalenderberechnungen durchführen

Die Berechnungen im Kalender sind nicht an den sichtbaren Bereich der Zeitskala gebunden. Die Methode **AddDuration** des Objekts **Calendar** berechnet aus einem Starttermin und der angegebenen Anzahl von Arbeitszeiteinheiten den Endtermin, der sich aus dem Überspringen der arbeitsfreien Zeiten ergibt. Negativ übergebene Zeiteinheiten bewirken, dass zu einem

spezifizierten Endtermin der passende Anfangstermin berechnet wird. Die zu **AddDuration** komplementäre Methode **CalcDuration** kann wiederum aus einer Zeitspanne, die durch Start- und Endtermin gegeben ist, die Dauer der dazwischen liegende Arbeitszeiteinheiten ermitteln.



> Wirkung der Berechnungsmethoden

Hinweis: Beachten Sie bitte, dass die Angabe der Arbeitszeiteinheiten in der Einheit Tag, Stunde, Minute oder Sekunde erfolgen muss, die aktuell in der Eigenschaft **TimeUnit** des Objekts **VcGantt** eingestellt ist.

Die Methode **AddDuration** gewährleistet, dass die berechneten Termine stets innerhalb eines Arbeitszeitintervalls liegen. Dies bedeutet gleichzeitig jedoch auch, dass eine Berechnung nicht eindeutig umkehrbar ist, wenn der Ausgangstermin sich ursprünglich innerhalb der arbeitsfreien Zeit befunden hat.

> Begrenzte Umkehrbarkeit der Berechnung

Beim interaktiven Anlegen und Ändern von Vorgängen sorgt **VARCHART XGantt** automatisch dafür, dass Vorgänge nicht innerhalb arbeitsfreier Zeitspannen beginnen oder enden können. Wenn Sie erreichen möchten, dass das Verhalten beim Erzeugen oder Ändern von Vorgängen über die API zur interaktiven Arbeitsweise konsistent ist, müssen Sie dies durch die Korrektur des Start- bzw. Enddatums selbst sicherstellen. Sie verlegen den Starttermin, der sich innerhalb einer arbeitsfreien Zeit befindet, auf den Beginn des nächsten Arbeitszeitintervalls und den Endtermin entsprechend auf das Ende des vorherigen Arbeitszeitintervalls. Die Methoden zur Bestimmung der Zeitintervallgrenzen ermöglichen dies auf einfache Weise und werden im nächsten Abschnitt im Detail besprochen.

Code-Beispiel

```
If calendar.IsWorktime(startDate) = False Then
    startDate = calendar.GetNextIntervalBorder(startDate)
End If
```

```

If calendar.IsWorktime(EndDate) = False Then
    EndDate = calendar.GetNextIntervalBorder(EndDate)
End If

```

> Sommerzeit

VARCHART XGantt unterstützt automatisch die Sommerzeit. Die Sommerzeit beginnt in Mitteleuropa am letzten Sonntag im März und endet am letzten Sonntag im Oktober. Zu Beginn der Sommerzeit werden die Uhren von 2.00 Uhr auf 3.00 Uhr vorgestellt und am Ende der Sommerzeit von 3.00 Uhr auf 2.00 Uhr zurückgestellt.

Sommerzeitbeginn:

00:00 Uhr	01:00 Uhr	03:00 Uhr	04:00 Uhr	05:00 Uhr	06:00 Uhr	...
-----------	-----------	-----------	-----------	-----------	-----------	-----

Sommerzeitende:

00:00 Uhr	01:00 Uhr	02:00 Uhr	02:00 Uhr	03:00 Uhr	04:00 Uhr	...
-----------	-----------	-----------	-----------	-----------	-----------	-----

Die Methode **calcDuration** ermittelt für den Tag des Sommerzeitbeginns eine Zeitspanne von 23 Stunden und für den Tag des Sommerzeitendes eine Zeitspanne von 25 Stunden zurück, wenn Stunden als Einheit verwendet werden. Wenn **TimeUnit** auf Tage eingestellt ist, beträgt die Zeitspanne in beiden Fällen exakt einen Tag.

Zeitintervallgrenzen ermitteln

Die Methoden des Objekts **Calendar** zum Bestimmen der Zeitintervallgrenzen **GetStartOfInterval**, **GetNextIntervalBorder** und **GetPreviousIntervalBorder** ermöglichen es, über Arbeitszeitintervalle und Nichtarbeitszeitintervalle zu iterieren. Die Ergebnisse der Methoden werden relativ auf ein Referenzdatum bezogen, welches den Methoden als Übergabeparameter mitgegeben wird.

Mit der Methode **IsWorkTime** des Objekts **Calendar** lässt sich für ein beliebiges Datum feststellen, ob es innerhalb eines Arbeitszeitintervalls oder Nichtarbeitszeitintervalls liegt. Obwohl der Beginn eines neuen Intervalls zugleich das Ende des vorherigen Intervalls darstellt, gehört das Startdatum selbst zum neuen Intervall (rechtsoffenes Intervall).

Die beiden Komfortmethoden **GetEndOfPreviousWorkTime** und **GetStartOfNextWorkTime** stellen keine neuen Möglichkeiten bereit, sondern vereinfachen nur den Umgang mit Arbeitszeitintervallen.

98 Wichtige Konzepte: Der Einsatz des Kalenders

Im folgenden Programmbeispiel werden die Zeitintervalle des Kalenders ausgelesen und in eine Datei geschrieben. Zudem wird die im Betrachtungszeitraum verfügbare Arbeitszeit ermittelt:

Code-Beispiel

```
Private Sub writeCalendarIntervalsToFile(ByVal filename As String, ByVal
calendar As VcCalendar, ByVal startDate As Date, ByVal endDate As Date,
ByVal listWorkIntervals As Boolean, ByVal listNonWorkIntervals As
Boolean)
Dim tmpStartDate As Date
Dim nextStartDate As Date
Dim totalWorkTime As Integer

Open filename For Output As #1
Print #1, "Time Intervals of " & calendar.Name & "between " & startDate
& " - " & endDate

    tmpStartDate = startDate
    Do While tmpStartDate < endDate

        nextStartDate = calendar.GetNextIntervalBorder(tmpStartDate)

        If tmpStartDate = nextStartDate Then
            nextStartDate = endDate
        End If

        If nextStartDate > endDate Then
            nextStartDate = endDate
        End If

        If calendar.IsWorktime(tmpStartDate) Then
            If listWorkIntervals Then
                Print #1, "WorkInterval" & " " & tmpStartDate & " " &
nextStartDate
            End If
        Else
            If listNonWorkIntervals Then
                Print #1, "NonWorkInterval" & " " & tmpStartDate & " " &
nextStartDate
            End If
        End If

        tmpStartDate = nextStartDate
    Loop

    totalWorkTime = calendar.CalcDuration(startDate, endDate)
    Print #1, "Total work time: " & totalWorkTime & " Units"
    Close #1
End Sub
```

Hinweis: Die Intervalle im Kalender können sekundengenau spezifiziert kann und einen Zeitbereich von maximal 137 Jahren (ulong in Sekunden) umfassen.

> Anweisung zum Schreiben der Intervalle in eine Datei

Code-Beispiel

```
Call writeCalendarIntervalsToFile("C:\text.txt", calendar,
VcGantt1.TimeScaleStart, VcGantt1.TimeScaleEnd, True, True)
```

```
Time Intervals of CompanyCalendar_1 between
01.01.2011 00:00:00 - 01.01.2012 00:00:00

01.01.2011 00:00:00 - 02.01.2011 00:00:00 non-work time
02.01.2011 00:00:00 - 03.01.2011 00:00:00 non-work time
03.01.2011 00:00:00 - 03.01.2011 08:00:00 non-work time
03.01.2011 08:00:00 - 03.01.2011 12:00:00 work time
03.01.2011 12:00:00 - 03.01.2011 13:00:00 non-work time
03.01.2011 13:00:00 - 03.01.2011 17:00:00 work time
03.01.2011 17:00:00 - 04.01.2011 00:00:00 non-work time
04.01.2011 00:00:00 - 04.01.2011 08:00:00 non-work time
04.01.2011 08:00:00 - 04.01.2011 12:00:00 work time
04.01.2011 12:00:00 - 04.01.2011 13:00:00 non-work time
04.01.2011 13:00:00 - 04.01.2011 17:00:00 work time
04.01.2011 17:00:00 - 05.01.2011 00:00:00 non-work time
...
30.12.2011 00:00:00 - 30.12.2011 08:00:00 non-work time
30.12.2011 08:00:00 - 30.12.2011 12:00:00 work time
30.12.2011 12:00:00 - 30.12.2011 13:00:00 non-work time
30.12.2011 13:00:00 - 30.12.2011 17:00:00 work time
30.12.2011 17:00:00 - 31.12.2011 00:00:00 non-work time
31.12.2011 00:00:00 - 01.01.2012 00:00:00 non-work time

Total work time: 2064 Units
```

3.5 Ereignisse

Über Ereignisse werden die Interaktionen des Anwenders vom VARCHART-ActiveX-Steuerelement an die Applikation weitergegeben. Immer wenn der Anwender mit dem VARCHART-ActiveX-Steuerelement interagiert, indem er beispielsweise Daten ändert oder auf irgendeine Stelle des Steuerelements klickt, wird ein entsprechendes Ereignis gemeldet. Sie können in Ihrem Programmcode die Ereignisse auffangen und mit Ihrer Anwendung beliebig darauf reagieren.

In jeder Entwicklungsumgebung werden für die Ereignisse entsprechende Funktionen bereitgestellt, in denen die vom VARCHART-ActiveX-Steuerelement zur Verfügung gestellten Parameter schon eingetragen sind. Die einzelnen Ereignisse sind in der API-Referenz ausführlich beschrieben. Hier sei nur kurz darauf hingewiesen, dass Sie unter Verwendung des Parameters **returnStatus** in den Ereignissen alle im VARCHART-ActiveX-Steuerelement verfügbaren Kontextmenüs abschalten (und ggf. durch eigene ersetzen) und alle Interaktionen kontrollieren und bei Bedarf unterbinden bzw. rückgängig machen können.

> Rückgabewerte

Die folgende Tabelle enthält die Rückgabewerte für VARCHART-ActiveX-Ereignisse:

Konstante	Wert	Beschreibung
vcRetStatDefault	2	standardmäßige Vorbesetzung
vcRetStatFalse	0	Abbrechen der Aktion
vcRetStatNoPopup	4	Kontextmenü erscheint nicht

3.6 Filter

Ein Filter enthält Bedingungen für Layer, Histogrammkurven, Verbindungen oder Tabellenformate. Mit Hilfe eines Filters können Sie alle Layer, Kurven, Verbindungen bzw. Tabellenformate, die die Filterbedingungen erfüllen, auswählen, um sie beispielsweise grafisch hervorzuheben.

Wenn Sie einen Filter anwenden, werden die Informationen des Datensatzes eines Vorgangs, einer Histogrammkurve, einer Verbindung bzw. eines Tabellenformats mit den Filterkriterien verglichen und die Objekte ausgewählt, die die Filterkriterien erfüllen. Beispielsweise können Sie den Filter "Alle Vorgänge mit Beginn ab Januar 2010" definieren.

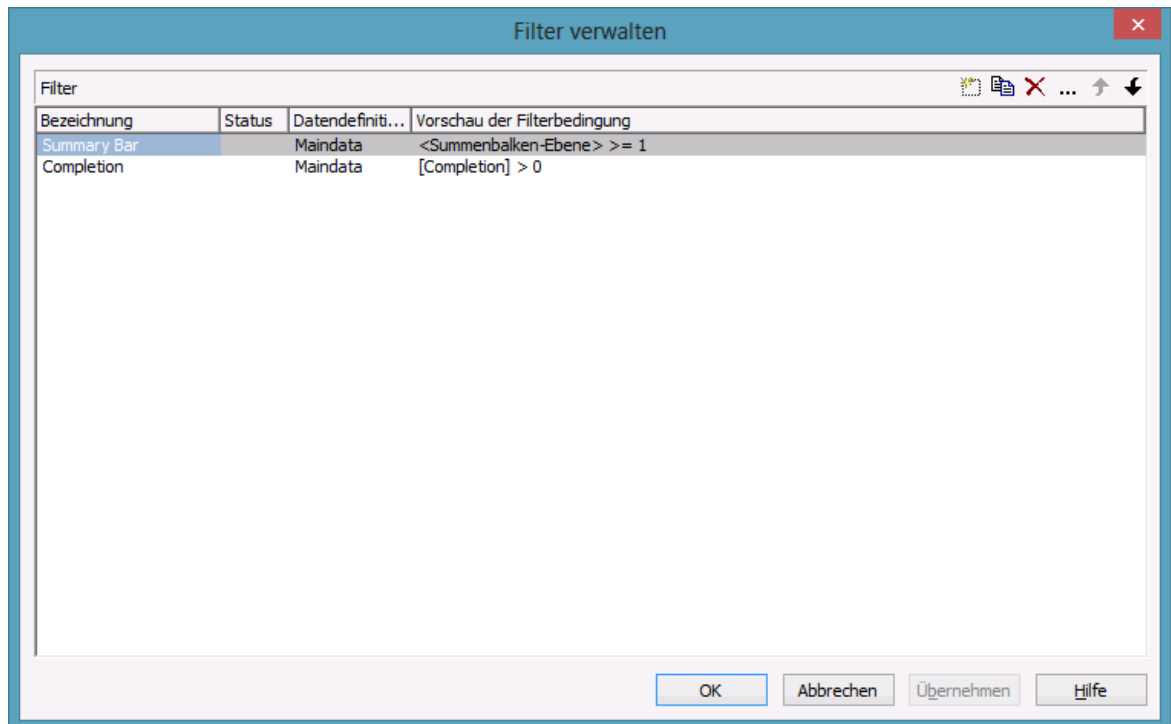
Filter können nur im Design-Modus erzeugt, bearbeitet und verwaltet werden.

Sie erreichen das Dialogfeld **Filter verwalten**

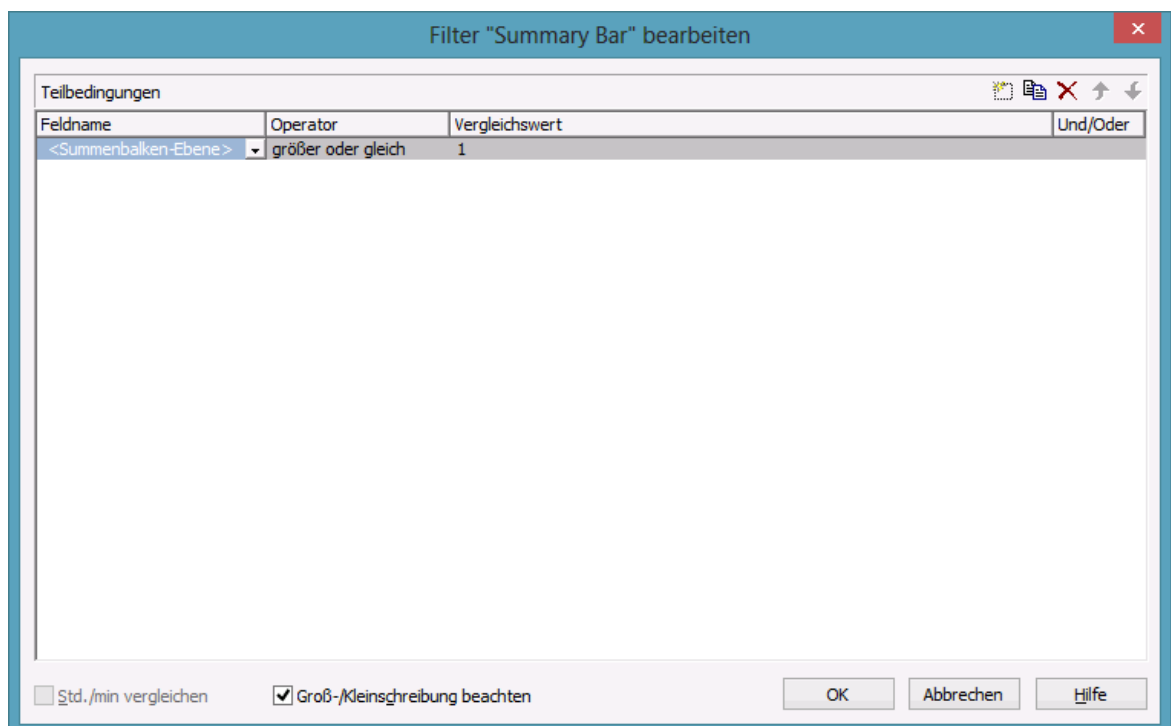
- von der Eigenschaftenseite **Objekte**
- für Layer: vom Dialogfeld **Balkenaussehen festlegen**
- für Tabellenformate: vom Dialog **Tabelle bearbeiten**
- für Verbindungen: von der Eigenschaftenseite **Verbindungen** über die **Filter**-Schaltfläche
- für Histogrammkurven: vom Dialog **Histogramm bearbeiten** über die **Filter**-Kombobox
- für Knoten: von der Eigenschaftenseite **Knoten** über die **Filter**-Schaltfläche

Im Dialogfeld **Filter verwalten** können Sie Filter umbenennen, kopieren, löschen, bearbeiten oder neue Filter definieren.

102 Wichtige Konzepte: Filter



Einen vorhandenen Filter bearbeiten können Sie im Dialogfeld **Filter bearbeiten**, das Sie vom Dialogfeld **Filter verwalten** aus erreichen.



3.7 Grafikformate

VARCHART unterstützt die folgenden Grafikformate, was für den Export von Grafiken für Bedeutung ist, vor allem für die Aufrufe **VcGantt1.ShowGraphicsExportDialog** und **VcGantt1.ExportGraphics**.

Das XGantt-Steuerelement unterstützt sowohl den Import von Grafikdateien z.B. für die Darstellung in Knoten oder in Boxen wie auch den Export ganzer Charts in Grafikdateien. Dabei spielen die verschiedenen unterstützten Grafikformate eine nicht ganz unwichtige Rolle in Bezug auf die Qualität der Darstellung der Grafik nach dem Import im Steuerelement bzw. nach dem Export in einem externen Anzeigeprogramm. Im folgenden werden die einzelnen Grafikformate mit ihren Vorzügen und Beschränkungen kurz vorgestellt. Dabei sind grundsätzlich zwei Typen zu unterscheiden:

Vektorgrafikformate speichern einzelne geometrische Figuren wie Linien, Ellipsen oder Rechtecke als Beschreibung der Figur mit darauf bezogenen Parametern wie Startkoordinaten, Ausdehnung und Farbe. Sie sind damit auflösungsunabhängig, d.h. bei stärkerem Hineinzoomen werden Linien weiterhin sauber dargestellt. Eine Einschränkung gibt es hier höchstens bei der Größe des zur Verfügung stehenden Koordinatenraums insbesondere beim WMF-Format. Allgemein haben Vektorgrafikformate also einen großen Vorteil durch die Auflösungsunabhängigkeit und oft auch bei der sich ergebenden Dateigröße. Leider hat sich kein plattformunabhängiges, genormtes Format durchgesetzt.

Bitmapgrafikformate speichern alle Bildpunkte mit ihrer Farbe in einer vorgegebenen Ausdehnung. Beim stärkeren Hineinzoomen werden die Grafiken dann automatisch "pixelig". Um einer ausufernden Dateigröße entgegenzuwirken, werden Bitmapgrafiken bei vielen Formaten verlustfrei oder gar verlustbehaftet komprimiert. Ein Verlust ist aber nur bei Fotos und nicht bei Diagrammen hinnehmbar. Ein Vorteil haben Bitmapgrafikformate nur dadurch, dass sie sich über Digitalkameras und das Internet durchgesetzt und plattformunabhängig weitverbreitet sind.

> WMF (Windows Metafile Format)

Dieses Vektorgrafik-Format existiert seit Windows 3.0 und besteht intern aus Befehlsdatensätzen, die den GDI-Befehlen der Windows-API entsprechen. Hiermit können GDI-Befehle sozusagen persistent gemacht werden. Dieses Format war aber schon zu Zeiten seiner Entwicklung nicht vollständig: So besaß und besitzt es bis heute nur einen eingeschränkten Koordinatenraum. Außerdem fehlt das Clipping, die Koordinatentransformation und das Füllen komplexer Polygone. Das Problem, nicht die tatsächliche Ausdehnung einer

WMF-Datei in "echten" Koordinaten wie cm oder Zoll festlegen zu können, wurde schon früh durch die Firma Aldus angegangen, die den sogenannten "Aldus Placeable Header" entwickelte, der seit langem in praktisch allen Programmen zur Anzeige von WMF-Dateien erkannt und genutzt wird mit Ausnahme der Windows-API selbst, die bis heute diesen Header nicht erzeugen oder verarbeiten kann, obwohl er in der Microsoft-Dokumentation erwähnt und erklärt wird.

Mit Windows NT und 95 wurde das Format von Microsoft eigentlich "in Rente" geschickt und sein Nachfolger namens EMF eingeführt. Trotzdem erfreut sich WMF bis heute einiger Beliebtheit vornehmlich im Bereich von ClipArt-Grafiken, wo die erweiterten Möglichkeiten des Nachfolgeformats nicht die Rolle spielen. Neuere Entwicklungen in Windows 95 und NT und deren Nachfolgern flossen nicht mehr in das Format ein; es ist seitdem unverändert geblieben.

WMF kennt auch einen Kommentardatensatz, der dazu genutzt werden kann, dort EMF-Befehle unterzubringen. Wenn ein Anzeigeprogramm solche Kommentare entdeckt, also auch EMF-Dateien anzeigen kann, dann verwirft es automatisch die WMF-Befehlsdatensätze und zeigt die EMF-Befehlsdatensätze. So kann eine einzige Datei WMF- wie auch EMF-Grafik enthalten. Dies ist wohl aus Kompatibilitätsgründen eingebaut worden, bläht aber die Dateigröße stark auf.

Formatbeschreibung siehe:

<http://msdn.microsoft.com/en-us/library/cc215212.aspx>

Beschränkungen des Formats siehe:

<http://support.microsoft.com/kb/81497/en-us>

> **EMF (Enhanced Metafile Format)**

Dieses Vektorgrafik-Format wurde mit den 32bit-Betriebssystemen Windows NT und 95 eingeführt und behebt die Einschränkungen des WMF-Formats und besteht intern aus Grafikbefehlen, die den GDI32-Befehlen Windows-API entsprechen. Der Koordinatenraum ist nun also 32bittig, Transformationen und Clipping werden unterstützt. Die später ins GDI32 hinzu genommenen Befehle zum maskierten oder mit AlphaBlending versehenem Blenden von Speicher-Bitmaps werden aber nicht unterstützt.

Das Format ist bis heute trotz der Vorteile, die es gegenüber WMF bietet, recht unbekannt geblieben. Alle Anzeigeprogramme und Office-Pakete können allerdings mit EMF-Dateien umgehen.

Ein Nachteil bei der Verwendung von GDI+ ergibt sich dergestalt, dass die von GDI+ neu eingeführten Möglichkeiten der grafischen Gestaltung wie

Farbverläufen und Transparenzen nicht voll unterstützt werden und außerdem beim Export in eine EMF-Datei unterbrochene Linien (gestrichelt u.ä.) in einzelne kleine nicht unterbrochene Linien gespeichert werden, wodurch zum einen der Speicherbedarf stark ansteigt und zum anderen eine Anzeige einer so geschriebenen Datei sehr lange Zeit benötigt.

EMF kennt auch einen Kommentardatensatz, der dazu genutzt werden kann, dort EMF+-Befehle unterzubringen. Wenn ein Anzeigeprogramm solche Kommentare entdeckt, also auch EMF+-Dateien anzeigen kann, dann verwirft es automatisch die EMF-Befehlsdatensätze und zeigt die EMF+-Befehlsdatensätze. So kann eine einzige Datei EMF- wie auch EMF+-Grafik enthalten. Dies ist wohl aus Kompatibilitätsgründen eingebaut worden, bläht aber die Dateigröße stark auf.

Druckjobs werden Windows-intern übrigens auch als EMF-Datenstrom ggf. zwischengespeichert und an den Druckertreiber übergeben.

Formatbeschreibung siehe:

<http://msdn.microsoft.com/en-us/library/cc204166.aspx>

> **EMF+ (Enhanced Metafile Format Plus)**

Obwohl der Name nahelegt, dass es sich um eine Erweiterung des EMF handelt, ist dies ein eigenes Vektorgrafikformat, das mit der Vorstellung der GDI+-Windows-API eingeführt wurde und intern aus Grafikbefehlsdatensätzen besteht, die den GDI+-Befehlen entspricht (GDI+ ist übrigens auch keine Erweiterung von der GDI-API, sondern eine eigene Grafikbibliothek). Zusätzlich zu EMF werden hier Transparenzen und Farbverläufe voll unterstützt.

Das Format ist bis heute recht unbekannt geblieben und wird auch von üblichen Anzeigeprogrammen oft nicht unterstützt, außer in Microsoft Office ab 2003. Microsoft hat erst im letzten Jahr den Aufbau des EMF+-Dateiformats veröffentlicht.

Formatbeschreibung siehe:

<http://msdn.microsoft.com/en-us/library/cc204376.aspx>

> **GIF (Graphics Interchange Format)**

Dieses Bitmap-Format wurde von CompuServe in Zeiten vor dem Entstehen des World Wide Web zur verlustfreien, komprimierten Speicherung von Grafiken entwickelt und kann nur gleichzeitig 256 Farben darstellen. Daher kann es die heutigen Grafiken nicht zufriedenstellend speichern. Es wird nur noch aus Kompatibilitätsgründen angeboten.

Die Unterart "Animated GIF" wird überhaupt nicht unterstützt.

> **JPEG (Joint Photographic Experts Group)**

Dieses Bitmap-Format wurde von der JPEG zur verlustbehafteten, komprimierten Speicherung von Fotos entwickelt. Daher ist eine Speicherung von Diagrammen, bei denen es z.B. auf die saubere Speicherung von Linien ankommt, nicht sinnvoll. Es wird nur noch aus Kompatibilitätsgründen angeboten.

> **BMP (Windows Bitmap)**

Dieses Bitmap-Format wurde von Microsoft zur verlustfreien, unkomprimierten Speicherung von Grafiken entwickelt. Das Format wird auch intern im Speicher von der Windows-API GDI direkt verwendet. Eine einzige Einschränkung gibt es dadurch, dass es keinen Alphakanal unterstützt, d.h. es können maximal 24 Bits pro Pixel gespeichert werden. Aufgrund seines hohen Speicherbedarfs sollte auf dieses Format verzichtet werden. Es wird nur noch aus Kompatibilitätsgründen angeboten.

> **TIFF (Tagged Image File Format)**

Dieses Bitmap-Format wurde von Aldus (in Adobe aufgegangen) zur Speicherung von Grafiken entwickelt. Die Grafik kann darin verlustbehaftet oder verlustfrei gespeichert werden. Das Format wird seit längerem nicht mehr weiterentwickelt. Es wird nur noch aus Kompatibilitätsgründen angeboten.

> **PNG (Portable Network Graphics)**

Dieses Bitmap-Format wurde vom World Wide Web Consortium (W3C) zur verlustfreien, komprimierten Speicherung von Grafiken entwickelt, um das vormals Copyright-behaftete und beschränkte GIF Format abzulösen. PNG ist hervorragend geeignet zur Speicherung von VARCHART-Grafiken und beim Einlesen werden transparente Anteile auch transparent gezeichnet. Es wird auch durchgängig von praktisch allen Anzeigeprogrammen und Internetbrowsern unterstützt. Das Format selbst ist frei von Copyrights und vollständig dokumentiert.

Seit der Version 4.2 wird für den Export die frei verfügbare Bibliothek **libpng** verwendet, um durch die Vorgabe der Auflösung beliebig große Bitmaps speichern zu können. Hierbei muss aber darauf geachtet werden, dass sehr große PNG-Dateien zu Schwierigkeiten beim Einlesen führen können, denn üblicherweise wird die PNG-Datei im Speicher erst komplett entpackt und dann dargestellt.

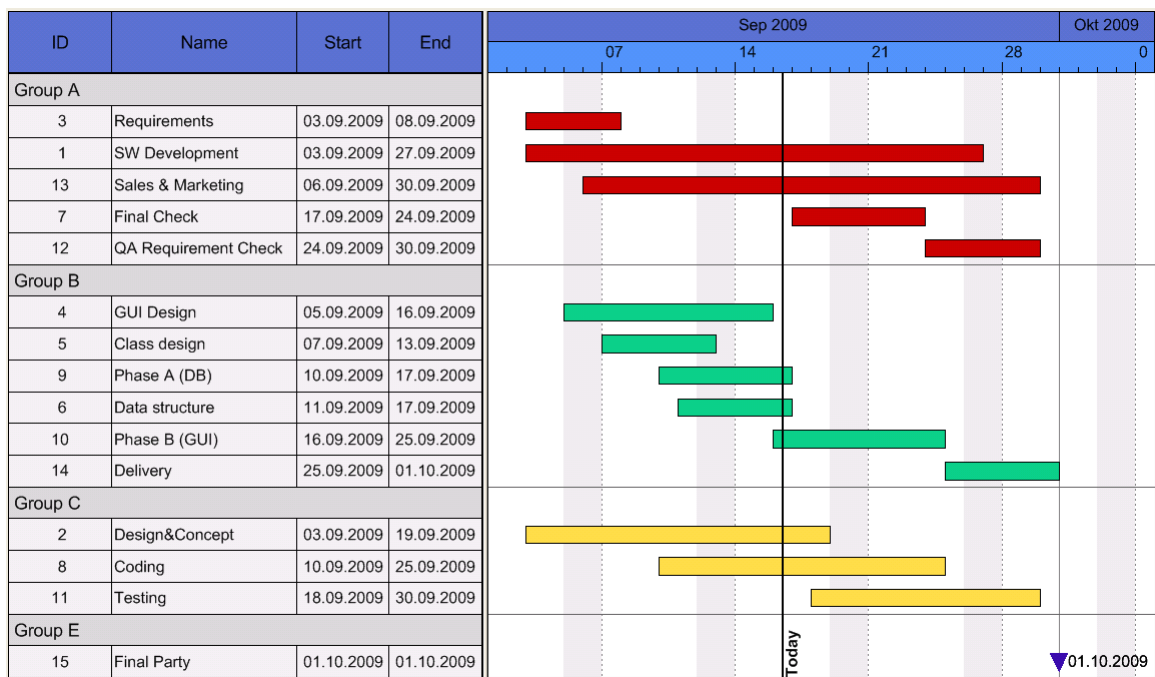
Formatbeschreibung siehe:

<http://www.libpng.org/pub/png/spec/1.1/PNG-Contents.html>

3.8 Gruppierung

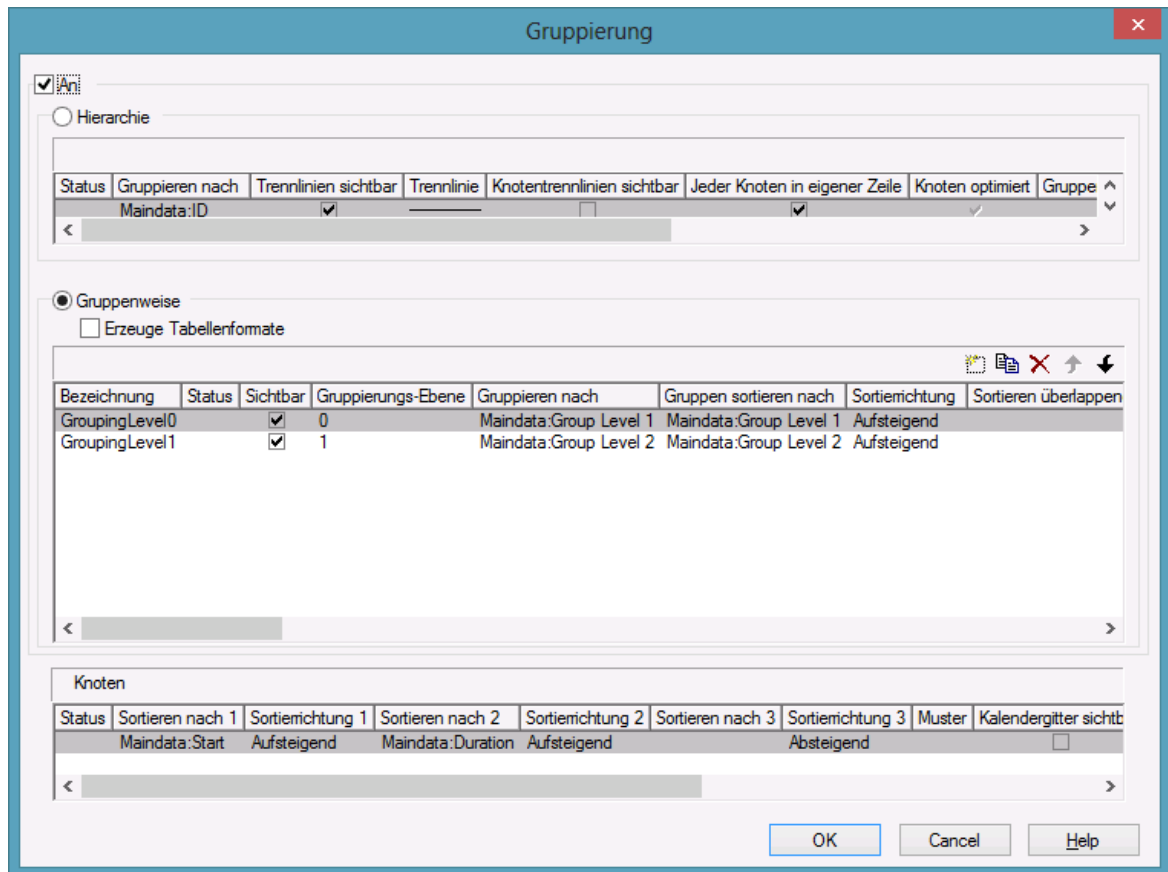
In vielen Anwendungsfällen ist es gefordert, Vorgänge in Gruppen einzuteilen und diese Gruppen in der Darstellung besonders hervorzuheben. Beispielsweise werden Vorgänge häufig nach bestimmten Projektphasen gruppiert (z. B. Planung, Konstruktion, Fertigung, etc.) oder nach bestimmten Abteilungen (Abteilung Konstruktion, Abteilung Rechnungswesen, etc.).

Ein gruppierter Plan sieht beispielsweise folgendermaßen aus:



Gruppierung bedeutet, dass alle Vorgänge ein gemeinsames Merkmal besitzen. Daher bilden alle Knoten, die in einem bestimmten Datenfeld denselben Wert haben, eine Gruppe.

Sämtliche Kriterien für die Gruppierung können Sie im gleichnamigen Dialog vornehmen. Sie erreichen diesen Dialog über die Eigenschaftenseite **Objekte**.



Alle Vorgänge, die denselben Wert für das unter **Gruppieren nach** gewählte Datenfeld besitzen, bilden eine Gruppe.

In der Darstellung wird für jede Gruppe eine zusätzliche Zeile ausgegeben, die nur den Gruppentitel enthält. Dessen Aussehen kann im Dialog **Tabellenformat bearbeiten** separat für den expandierten und für den kollabierten Zustand der Gruppe festgelegt werden (Tabellenformate **Subtitle** und **Collapsed**), beispielsweise mit verschiedenen Farben oder Datenfeldern.

Neben jedem Gruppentitel steht ein kleines Plus- bzw. Minus-Symbol, das den kollabierten bzw. expandierten Zustand der Gruppe anzeigt. Indem Sie auf dieses Symbol klicken, verkehrt sich der Zustand in sein Gegenteil. Voraussetzung ist, dass im Dialog **Gruppierung** die Option **Änderungen erlaubt** ausgewählt ist.

Um die Reihenfolge der Gruppen zu kontrollieren, können Sie die Felder **Gruppen sortieren nach** sowie **Sortierrichtung** verwenden.

Für die Gruppierung können Sie weiterhin folgende Kriterien festlegen:

- ob **Tabellenformate** angelegt werden sollen
- ein **Muster** für die Überschriftenzeile der Gruppe (nur im Diagramm)
- Anzeige, Art und Aussehen von **Kalender- und Liniengittern**

- ob alle Vorgänge einer Gruppe in einer eigenen Zeile ausgegeben werden sollen oder nicht (Aktivieren/Deaktivieren der Option **Jeder Knoten in eigener Zeile**) und wenn ja, ob das Layout automatisch optimiert werden soll (**Knoten optimiert**)
- ob die Gruppen beim Start kollabiert sein sollen (**Gruppen kollabiert**)
- Anzeige, Position und Aussehen von **Trennlinien** zwischen den Gruppen
- ob dem Anwender die Kollabieren/Expandieren-Funktionalität angeboten wird (**Änderungen erlaubt**)
- die Anzeige von **Summenbalken**
- ob **Gruppenknoten** sichtbar sein sollen
- ob Gruppen in Tabelle und Diagramm mit der Maus verschoben werden können um damit ihre **Reihenfolge** zu ändern
- ob nach jeder Gruppe **Seitenumbrüche** ausgeführt werden sollen

> Interaktives Neuanlegen von Gruppen

Wenn Sie in einem leeren Diagramm interaktiv den ersten Knoten anlegen, wird dafür automatisch eine neue Gruppe angelegt. Sie können im Dialog **Vorgänge bearbeiten** in das Datenfeld, das im Dialog **Gruppierung** unter **Gruppierung nach** gewählt wurde, einen Namen für die Gruppe des Knotens eintragen.

Falls Sie eine neue Gruppe anlegen möchten, gehen Sie folgendermaßen vor: Erzeugen Sie zunächst einen Knoten innerhalb einer bestehenden Gruppe. Klicken Sie dann doppelt auf diesen Knoten, um den Dialog **Vorgänge bearbeiten** aufzurufen. Tragen Sie nun in das Datenfeld, das im Dialog **Gruppierung** unter **Gruppierung nach** gewählt wurde, einen Namen für die Gruppe des Knotens ein. Die neue Gruppe wird dann automatisch erzeugt.

> Interaktives Umgruppieren von Knoten

Wenn der Anwender mit Hilfe der Maus einen Vorgang von einer Gruppe in eine andere schiebt, wird der Wert in dem Gruppierfeld automatisch angepasst.

> Leere Gruppen

Wenn Sie alle Knoten einer Gruppe löschen, bleibt die Gruppenüberschrift dieser Gruppe in der Tabelle zunächst stehen. Wenn Sie die Gruppierung aus- und dann wieder einschalten oder wenn Sie das Programm beenden und neu starten, werden die Überschriften aller leeren Gruppen nicht mehr angezeigt.

> Interaktives Verschieben von Untergruppen

Sie können interaktiv die Reihenfolge der Untergruppen einer Gruppe verändern. Markieren Sie dazu den Summenbalken der zu verschiebenden Untergruppe. Schieben Sie nun bei gedrückter linker Maustaste das Phantom des Summenbalkens nach oben bzw. unten. Sobald das Phantom auf einem anderen Summenbalken derselben Gruppierungsebene liegt, zeigt ein Pfeil am Cursor-Symbol, ob Sie den Summenbalken ober- bzw. unterhalb davon einfügen können. Wenn Sie die Maustaste loslassen, wird die Gruppe mit allen untergeordneten Knoten an der gewählten Stelle eingefügt.

> Alle Vorgänge aller Gruppen in einer Zeile/in getrennten Zeilen/expandieren/kollabieren

Sie können mit Hilfe weniger Code-Zeilen die Darstellung aller Vorgänge aller Gruppen festlegen. Im folgenden Beispiel werden über einen Menübefehl die Vorgänge von zwei Gruppierungsebenen in einer Zeile dargestellt.

Code-Beispiel

```
Private Sub mnuAllNodesOneRow_Click()

Set groupcollection = VcGantt1.groupcollection

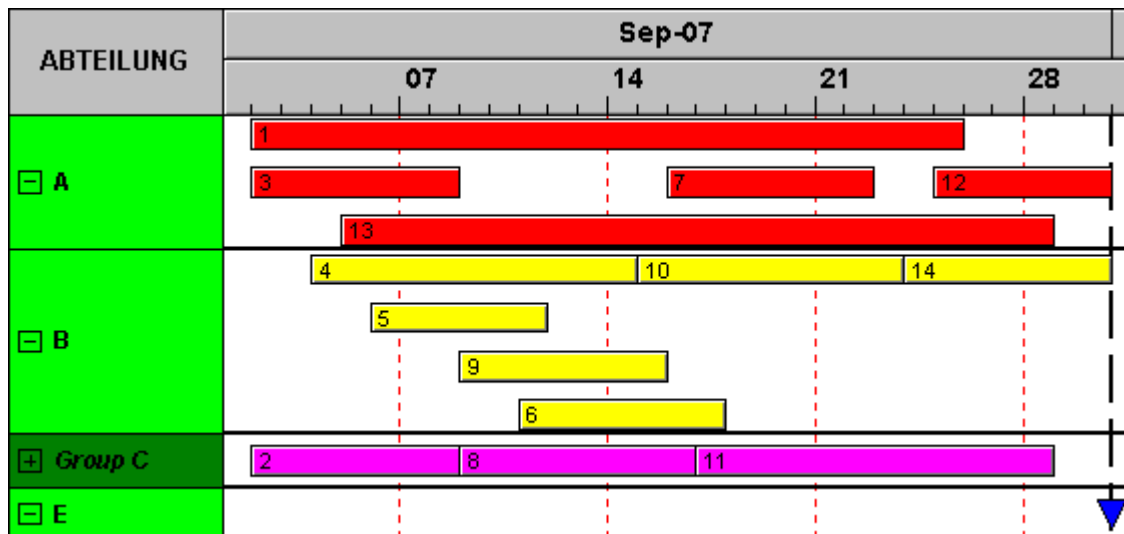
For Each group In groupcollection
    Set subgroupcollection = group.SubGroups
    group.AllNodesInOneRow = True
    For Each subgroup In subgroupcollection
        subgroup.AllNodesInOneRow = True
    Next
Next

End Sub
```

Analog können Sie alle Knoten aller Gruppen in getrennten Reihen darstellen (group.AllNodesInOneRow = False), expandieren (group.Collapsed = False) oder kollabieren (group.Collapsed = True).

> Darstellung jeder Gruppe in einer Zeile

In diesem Abschnitt wird kurz der Sonderfall **Jeder Knoten in eigener Zeile** für die gruppierte Anordnung der Vorgänge besprochen. Ein solcher Plan sieht etwa folgendermaßen aus:



Der Gruppierungsvorgang entspricht dem oben Beschriebenen. Dabei gab es für jeden Vorgang eine eigene Zeile. Wenn die Option **Jeder Knoten in eigener Zeile** im Dialog **Gruppierung** nicht aktiviert wurde, bildet jede Gruppe eine Zeile. Innerhalb dieser Zeile kann es natürlich zu Überlappungen kommen. Um diese sichtbar zu machen, kann die Gruppe expandiert werden, wodurch die Anordnung genau genommen "In möglichst wenigen Zeilen" heißen müsste. Durch entsprechendes Verschieben der Vorgänge kann man erreichen, dass keine Überlappungen mehr vorkommen. So hat man mit der expandierten Darstellung die Gewähr, dass eine Überlappung sofort ins Auge fällt, auch wenn sie nur eine Sekunde dauert.

Wird eine Gruppe kollabiert (wie im Beispiel Group C), so kann man zwar erkennen, dass mehrere Vorgänge in dieser Gruppe liegen, über zeitliche Überschneidungen kann man jedoch keine Aussage machen.

Bei dieser Art der Darstellung hat ein Tabellenformat für die Vorgänge natürlich keinen Sinn, und daher gibt es ein solches auch nicht. Es empfiehlt sich deshalb, den Layer zu beschriften oder einen Tooltip zur Identifizierung der Layer zu verwenden.

> Darstellung übereinander liegender Knoten

Wenn der Modus **Jeder Knoten in eigener Zeile** nicht gewählt wurde, können Sie über die Sortierreihenfolge der Knoten bestimmen, welche Knoten zuletzt gezeichnet werden und somit bei Überlappung vollständig sichtbar sind.

> Summenbalken

In den Gruppenzeilen können Summenbalken ausgegeben werden. Sie können festlegen, ob und ggf. auf welchen Gruppierungsebenen Summenbalken ausgegeben werden sollen.

Damit Summenbalken auf der unter **Gruppierungs-Ebene** eingestellten Ebene dargestellt werden, müssen Sie im Dialog **Gruppierung** das Kontrollkästchen **Summenbalken** für diese Ebene aktivieren.

Um Summenbalken zur Laufzeit an- bzw. abzuschalten, können Sie die VcGantt-Eigenschaft **SummaryBarsVisible** verwenden. Bei nicht-hierarchischer Gruppierung können Sie mit Hilfe des Parameters **GroupingLevel** ebenenweise die Summenbalken ein- bzw. ausschalten.

Das Aussehen von Summenbalken legen Sie auf der Eigenschaftenseite **Layer** fest, indem Sie für jede gewünschte Darstellung einen Layer erstellen. Dabei können Sie sowohl einen Layer für alle bzw. mehrere Ebenen als auch unterschiedliche Layer für jede einzelne Ebene definieren. Möglich wäre z.B. ein Layer "Summenbalken 1" für die 1. Ebene, ein Layer "Summenbalken 2" für die 2. Ebene etc.

Damit die Summenbalken dann auch tatsächlich in der gewünschten Form dargestellt werden, müssen ihnen entsprechende Filter zugewiesen werden, die Sie im Dialog **Filter verwalten** erstellen, z.B. "Summenbalken1" für die 1. Ebene. Um die passende Ebene für Ihren Filter einzustellen, wählen Sie im Dialog **Filter bearbeiten** unter **Feldname** den Eintrag "<Summenbalken-Ebene>", wählen den geeigneten **Operator** (gleich, ungleich größer als etc.) und tragen unter **Vergleichswert** die gewünschte Ebene ein.

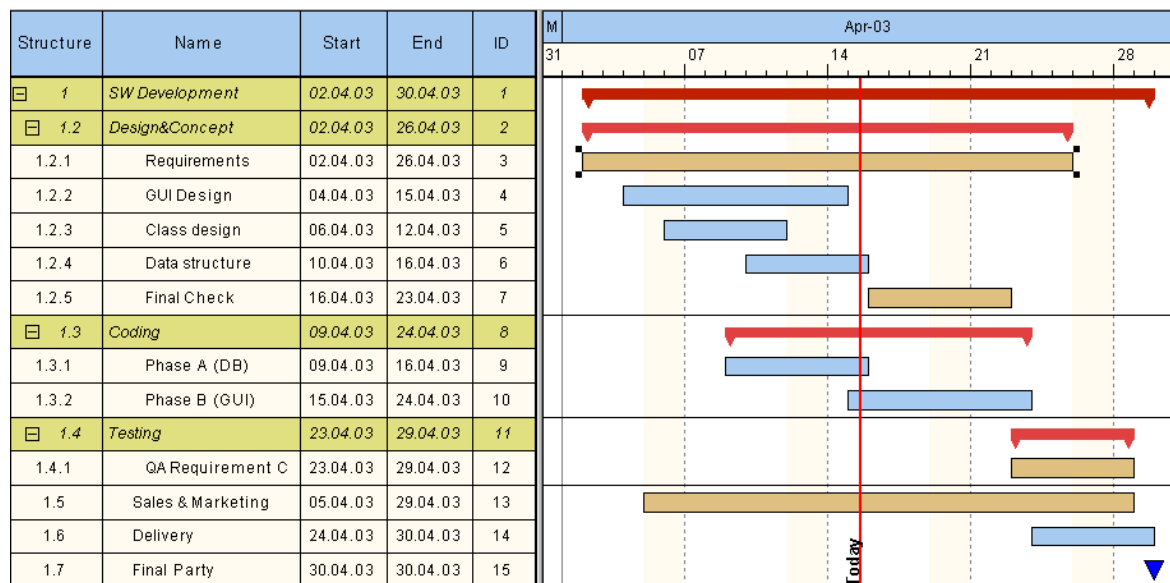
Wenn Sie nun das Programm starten, werden die so definierten Summenbalken dargestellt.

3.9 Hierarchische Anordnung

Neben der Gruppierung gibt es noch eine weitere Möglichkeit, Vorgänge in Ebenen anzuordnen: die Hierarchie. Dafür müssen die Projektdatensätze einen hierarchischen Code enthalten, der in folgender Weise aufgebaut sein muss:

1., 1.1, 1.1.1, 1.2, 1.2.1, ...

Eine hierarchische Anordnung sieht beispielsweise folgendermaßen aus:

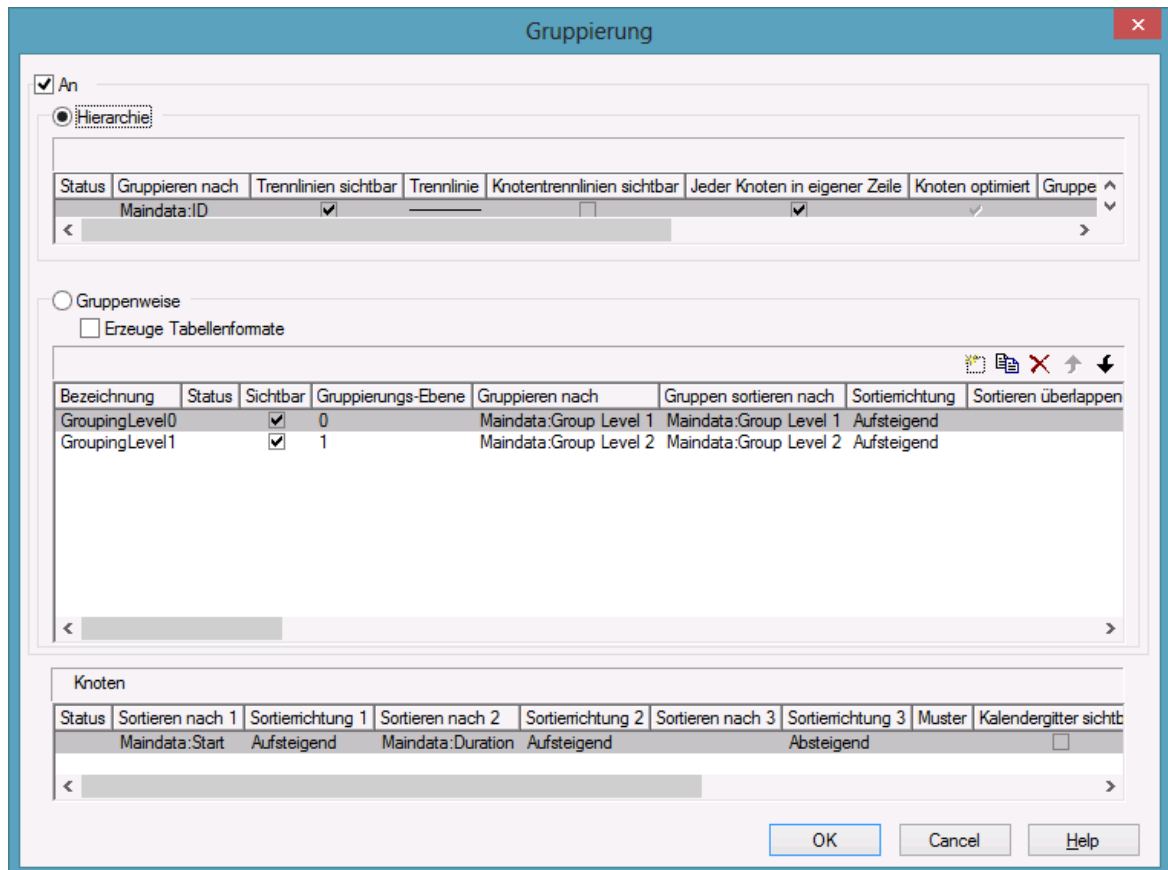


Die Symbole + und - werden automatisch den **Summenvorgängen** vorangestellt. Untergeordnete Ebenen werden automatisch eingerückt. Durch einen Klick auf das --Symbol werden jeweils die untergeordneten Vorgänge kollabiert (eingeklappt), durch einen Klick auf das +-Symbol expandiert (ausgeklappt).

Es wird nicht überprüft, ob die übergeordneten Vorgänge mit ihren Terminen die untergeordneten überdecken. Es wird also keine Prüfung oder Setzung der Dauer vorgenommen.

Bei der hierarchischen Anordnung ist keinerlei zusätzliche Gruppierung oder Sortierung möglich.

Die hierarchische Anordnung wird im Dialog **Gruppierung** eingestellt:



Hier muss das Kontrollkästchen **Hierarchie** aktiviert sein. Aus der Kombobox **Gruppieren nach** ist dann das Datenfeld auszuwählen, das den Code für die hierarchische Anordnung enthält. Die Anordnung der Vorgänge wird dadurch eindeutig bestimmt.

Außerdem lassen sich noch folgende Kriterien für die Hierarchie festlegen:

- Anzeige und Aussehen von **Trennlinien**
- ob die Vorgänge beim Start kollabiert sein sollen (**Gruppen kollabiert**)
- ob **Summenbalken** angezeigt werden sollen
- ob nach jeder Gruppe und bis zu welcher Ebene höchstens **Seitenumbrüche** ausgeführt werden sollen

Für die Ausgabe der Summenvorgänge werden die beiden Tabellenformate **Hierarchy** und **HierarchyCollapsed** verwendet. Sie können sie im Dialog **Tabellenformat bearbeiten** festlegen.

> Vorgänge interaktiv verschieben

Sie können Vorgänge interaktiv verschieben. Der verschobene Knoten wird dabei immer vor bzw. hinter dem Bezugsknoten eingefügt, auch bei kollabierten Gruppen.

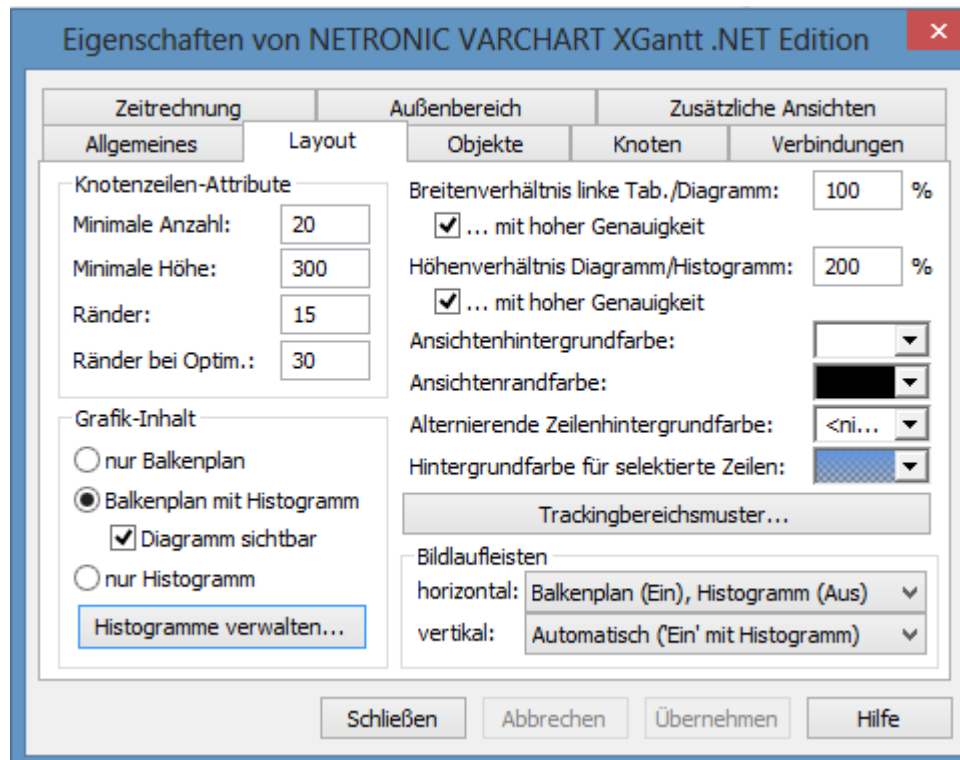
> **Summenbalken interaktiv verschieben**

Summenbalken können Sie genauso wie Knoten interaktiv verschieben. Dabei werden die untergeordneten Knoten mitverschoben.

3.10 Histogramme

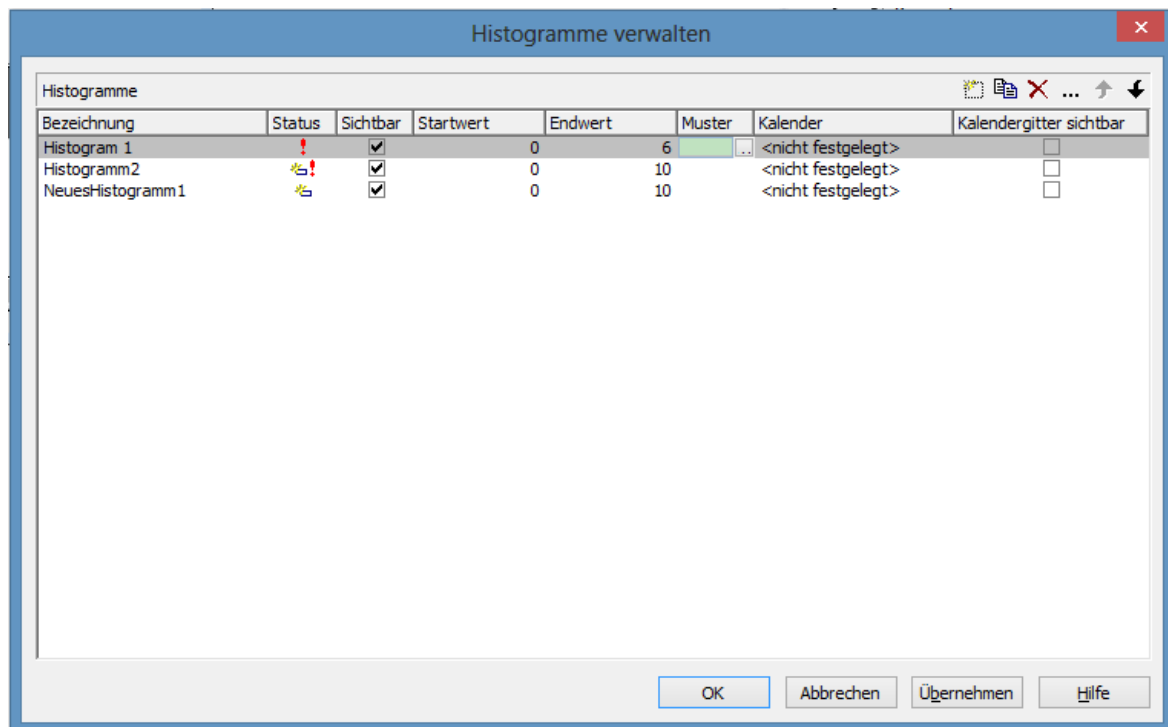
In den Histogrammen von VARCHART XGantt werden Vorgänge nach Kriterien, die Sie festlegen, über der Zeitachse zu Kurven aufsummiert.

Auf der Eigenschaftenseite **Layout** können Sie unter **Grafik-Inhalt** festlegen, ob nur der Balkenplan, nur das Histogramm oder beides ausgegeben werden soll.



Um auszuwählen, welche Histogramme dargestellt werden sollen, und um Histogramme zu bearbeiten, klicken Sie auf die Schaltfläche **Histogramme verwalten**. Dann erscheint der gleichnamige Dialog.

118 Wichtige Konzepte: Histogramme

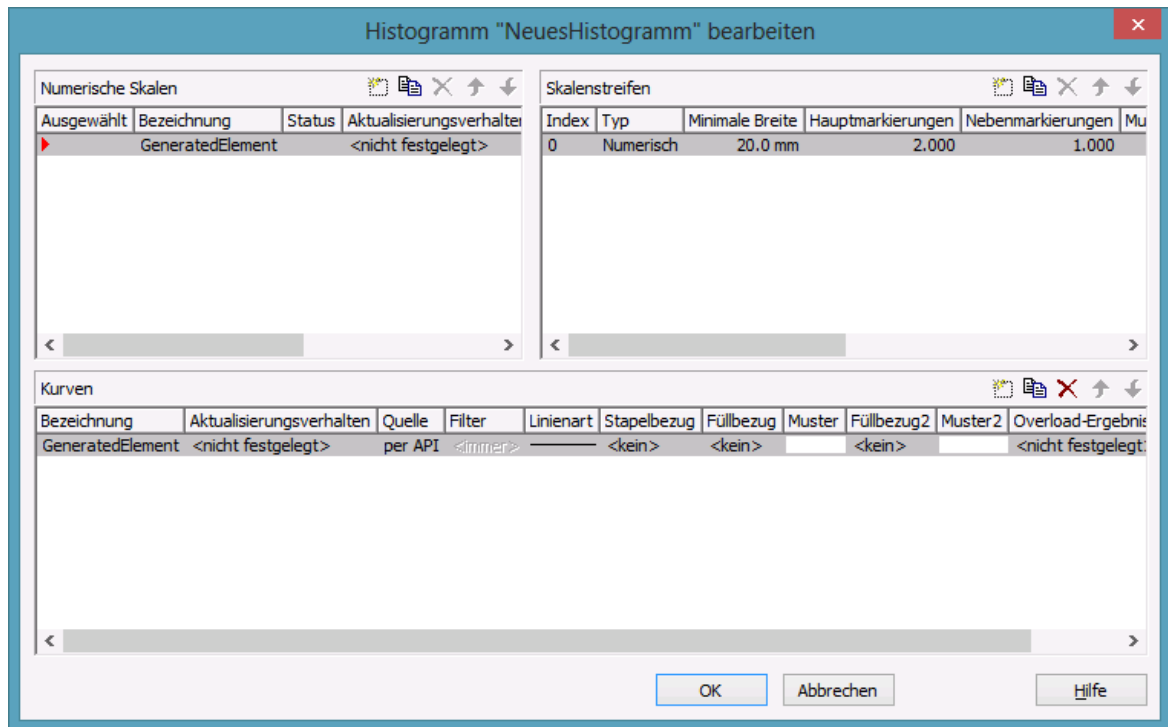


Hier können Sie ein oder mehrere Histogramme auswählen, die dargestellt werden sollen.

Jedes Histogramm besteht aus einer numerischen Skala (y-Achse) und Kurven. Für die x-Richtung wird die Zeitskala des Balkendiagramms verwendet.

Sie können hier für jedes Histogramm den Start- und den Endwert der numerischen Skala festlegen.

Um das Histogramm weiter zu bearbeiten, klicken Sie auf die Schaltfläche **Histogramme bearbeiten** (...). Dann erscheint der gleichnamige Dialog.

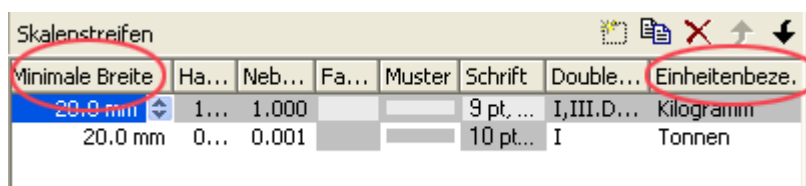


> Numerische Skalen

Sie können hier verschiedene numerische Skalen definieren und auswählen, welche davon für das zu bearbeitende Histogramm verwendet werden soll.

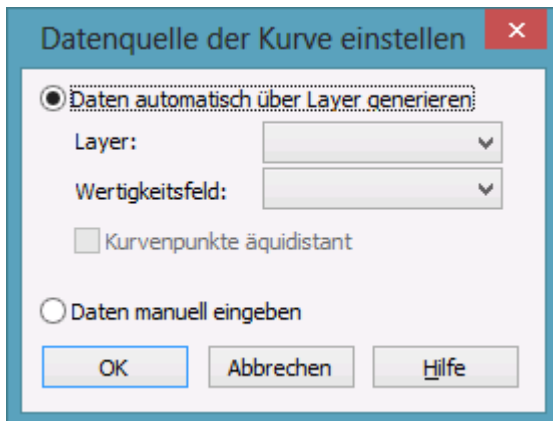
Für jede numerische Skala können Sie hier unter **Breite/Einheit** die Skalierung des Histogramms in y-Richtung festlegen. Außerdem können Sie hier entscheiden, ob ein Liniengitter dargestellt werden soll und ggf. dieses festlegen.

Im Bereich **Skalenstreifen** können Sie für die jeweils markierte numerische Skala einen oder mehrere Skalenstreifen festlegen und für jeden einen **Typ**, die **Minimale Breite**, die Anzahl der Einheiten, nach denen eine **Haupt-** bzw. eine **Nebenmarkierungen** erscheinen soll, eine Hintergrund-**Farbe**, verschiedene **Schriftattribute**, als auch ein **Doubleformat**. Darüberhinaus können Sie festlegen, ob Sie den Inhalt des Skalenstreifens über die **ObjectDraw-Ereignisse** selber gestalten möchten sowie eine **Bezeichnung** für die im Ribbon verwendeten Einheiten angeben. Für die Darstellung der Bezeichnung beachten Sie bitte, dass Sie über die Minimalbreite des Skalenstreifens genügend Platz dafür einrichten; andernfalls kann die Bezeichnung nicht angezeigt werden und bleibt unsichtbar.



> Histogrammkurven

Ein Histogramm kann mehrere Kapazitätskurven enthalten, für die Sie individuell einen **Namen**, die **Linienart** und das **Muster** unterhalb oder oberhalb der Kurvenlinie festlegen können. Für jede Kurve muss die **Quelle** angegeben werden. Dadurch legen Sie fest, woher die Daten für die Berechnung einer Kurve stammen sollen. Um die Quelle für eine Kurve auszuwählen, klicken Sie auf das Feld **Quelle** und dann auf die **Bearbeiten**-Schaltfläche (...). Es öffnet sich der folgende Dialog:



Es gibt zwei grundsätzliche Alternativen für die Datenquelle einer Kurve:

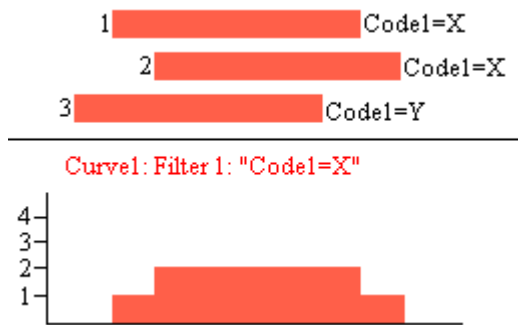
> 1. Daten automatisch über Layer generieren

Die Kurven werden aus den Vorgängen generiert. Beim Aufsummieren der Vorgänge zu einer Kurve werden für jeden Vorgang die Anfangs- und Enddaten des gewählten Layers (z.B. des Layers "Start-Ende") übernommen.

Falls die Kurve aus Layern erzeugt wird, können Sie festlegen, welche Vorgänge zu der Kurve beitragen sollen, indem Sie im Dialog **Histogramm bearbeiten** für jede Kurve einen **Filter** zur Selektion von Vorgängen auswählen.

Beispiel:

Zu Curve1 tragen nur die Vorgänge bei, die den Filter 1 erfüllen. Filter 1 enthält die Bedingung **Code1 = X**. Es tragen also nur die Vorgänge 1 und 2, für die **Code1 = X** gilt, zu Curve1 bei.



Für Kurven, die per Layer generiert worden sind, können Sie unter **Wertigkeitsfeld** das Datenfeld auswählen, aus dem pro Vorgang die Wertigkeit für die Addition der Kapazitäten entnommen wird.

> 2. per API (manuelle Eingabe)

Die Werte werden bei dieser Wahl über die Programmierschnittstelle (API) gesetzt. Darüber können Sie mit der VcCurve-Methode **SetValues** die Werte einer Histogrammkurve beliebig setzen.

Für per API generierte Kurven können Sie im Dialog **Datenquelle der Kurve einstellen** festlegen, ob die **Kurvenpunkte äquidistant** erzeugt werden sollen. Andernfalls werden Kurvenpunkte nur an den Stellen erzeugt, an denen sich der y-Wert ändert.

Kurvenpunkte äquidistant: Hier müssen ein Anfangswert (**startDate**) und die y-Werte der Histogrammkurve gesetzt werden. Aus dem Anfangswert werden zusammen mit den Werten für **Zeiteinheit** und **Kleinstes Zeitintervall** (Eigenschaftenseite **Allgemeines**) die Stützstellen der Histogrammkurve ermittelt.

Set Values X, Y1, Y2, Y3, ...

So gesetzte Kurven können nicht interaktiv verändert werden.

Kurvenpunkte nicht äquidistant: Hier müssen Paare von x- und y-Werten gesetzt werden:

Set Values X1, Y1

Set Values X2, Y2

Set Values X3, Y3...

Die **Zeiteinheit** und das **Kleinste Zeitintervall** sind hier irrelevant. Die Kurve kann hier interaktiv verändert werden.

> Referenzkurve

Eine mittels **SetValues** gesetzte Kurve kann als Referenzkurve verwendet werden, z. B. um die Verfügbarkeit darzustellen.

> Kurven stapeln

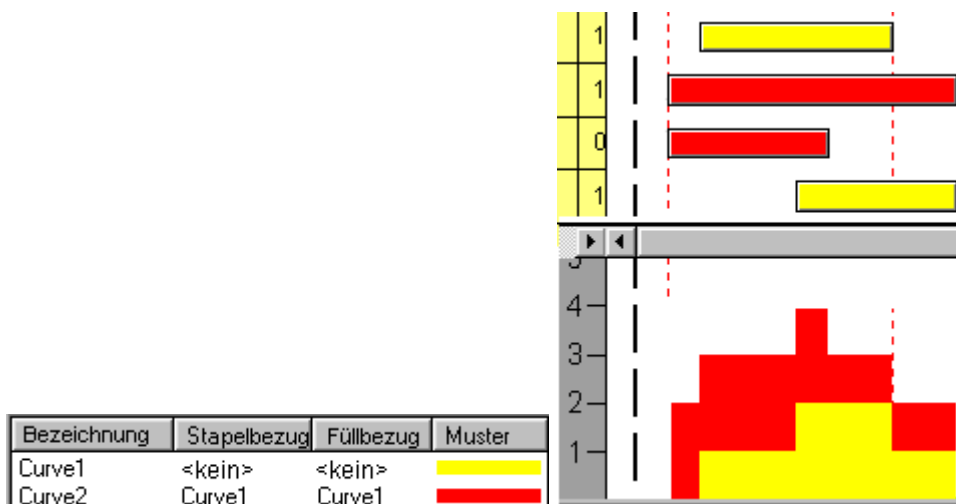
Das Stapeln von Kurven macht beispielsweise Sinn, um die Gesamtauslastung Ihrer Ressourcen darzustellen. Bei gestapelten Kurven werden die y-Werte eines x-Wertes zusammenaddiert. Dazu sind die Filter so zu definieren, dass sie die Vorgänge mit bestimmten Ressourcen selektieren.

Wenn Sie Histogrammkurven aufeinander stapeln möchten, geben Sie für jede Kurve im Feld **Stapelbezug** an, auf welche andere Kurve sie gestapelt werden soll. Soll eine Kurve nicht auf eine andere Kurve gestapelt werden, wählen Sie für diese Kurve im Feld **Stapelbezug** den Eintrag **kein**.

Wählen Sie für alle Kurven den Eintrag <kein>, überlagern die Kurven einander. Um die Kurven dennoch voneinander unterscheiden zu können, kennzeichnen Sie sie durch unterschiedliche Linienattribute.

Zwei Kurven können die Begrenzungen einer Fläche darstellen, die Sie mit Farbe und Muster versehen können. Im Feld **Füllbezug** weisen Sie der editierten Kurve eine Referenzkurve zu, die mit ihr eine Füllfläche bildet. Die Referenzkurve kann eine normale Kurve oder die X-Achse sein (**Null-Linie**).

Füllflächen können andere Kurve verdecken, daher müssen den Kurven die gewünschten Zeichnungsprioritäten zugewiesen werden.



Curve2 wird auf Curve1 gestapelt.

Soll bei gestapelten Kurven die Kurvenlinie nicht dargestellt werden, setzen Sie in der Programmierschnittstelle die VcCurve-Eigenschaft **LineType** auf den Wert **vcNone** oder wählen Sie im Dialogfeld **Linie bearbeiten** für **Typ** den Leereintrag.

Die Kurvenlinie und die Füllung unterhalb einer Kurvenlinie legen Sie in den Feldern **Linienart** und **Muster** fest. Wenn Sie auf den Eintrag des Feldes **Linienart** klicken, öffnet sich das Dialogfeld **Linie bearbeiten**, in dem Sie Farbe, Strichstärke und Linienart jeder Kurvenlinie festlegen können. Wenn

Sie auf das Feld **Muster** klicken, öffnet sich das Dialogfeld **Muster**, in dem Sie ein Muster sowie die Vorder- und Hintergrundfarbe für die Füllung unter einer Kurve festlegen können.

Sie können außerdem eine zweite Bezugskurve auswählen. Kreuzen Sie dazu das Kontrollkästchen **2. Bezug** an.

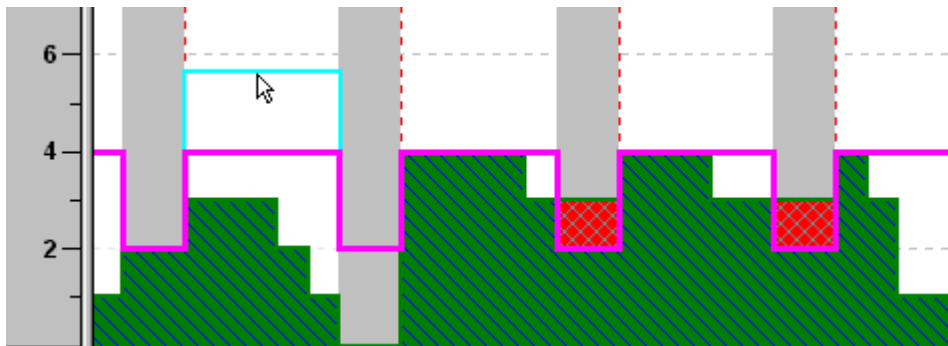
Wählen Sie unter **Füllbezug 2** eine 2. Bezugskurve aus. Die Füllung zur 2. Bezugskurve wird nur dargestellt, wenn die y-Werte der in dieser Zeile definierten Kurve größer sind als die der 2. Bezugskurve.

In dem 2. **Muster**-Feld können Sie das Muster und die Farbe der Füllung über der 2. Bezugskurve festlegen.

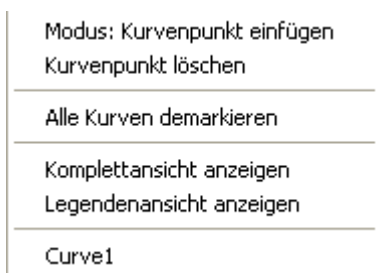
Beispiele zur Verwendung von Histogrammen finden Sie unter "Tutorium: Histogramme erstellen" sowie "Tutorium: Kapazitätsengpässe darstellen".

> Nicht-äquidistante Verfügbarkeitskurven interaktiv verändern

Sie können eine Veränderung der verfügbaren Kapazitäten interaktiv darstellen: Verändern Sie dazu zur Laufzeit die über die API gesetzten nicht-äquidistanten Kurven (Verfügbarkeitskurven), indem Sie mit der Maus ein horizontales Teilstück der Kurve nach oben oder unten ziehen. Ein Phantom zeigt Ihnen dabei die neue Position der Kurve an.



Sie können außerdem einzelne Kurvenpunkte hinzufügen oder löschen. Klicken Sie dazu mit der rechten Maustaste in den Histogrammbereich. Das folgende Kontextmenü erscheint:



Wenn mehrere Verfügbarkeitskurven definiert sind, werden deren Namen im Kontextmenü angezeigt. Wenn Sie auf den Namen einer Kurve klicken, wird diese markiert.

Im **Modus: Kurvenpunkt einfügen** können Sie mit jedem Linksklick auf die Verfügbarkeitskurve einen neuen Stützpunkt für diese definieren.

Um einen Stützpunkt zu löschen, klicken Sie diesen mit der rechten Maustaste an und wählen Sie im Kontextmenü den Befehl **Kurvenpunkt löschen**.

> **Stützstellen anzeigen**

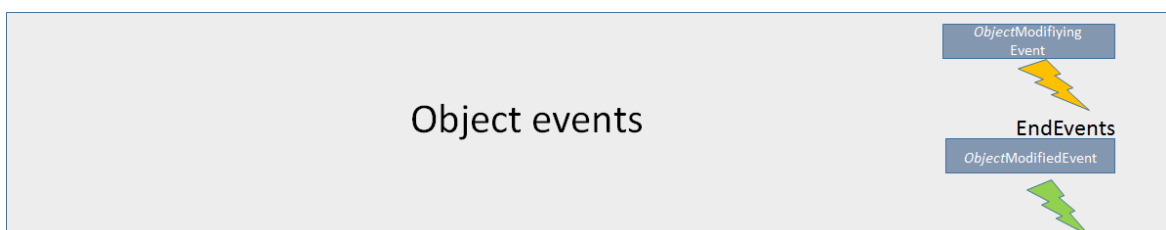
Sobald Sie auf eine nicht-äquidistante Verfügbarkeitskurve klicken, werden die per API definierten Kurvenpunkte durch kleine schwarze Quadrate angezeigt. Wenn Sie erneut auf die Verfügbarkeitskurve klicken, werden die Stützstellen nicht mehr angezeigt.

3.11 In-Interaction Events

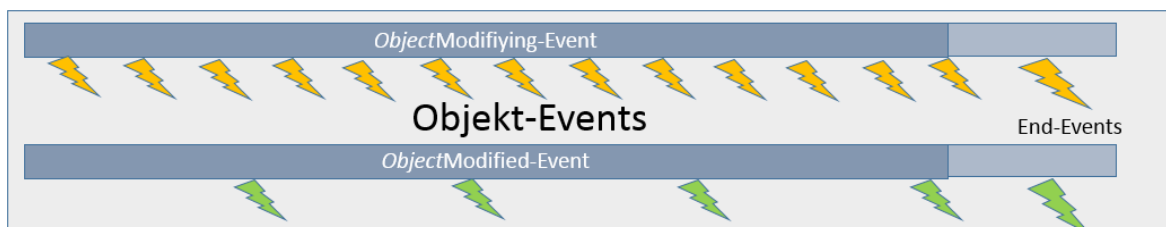
3.12 Interaction Events

Während einer Maus-Interaktion innerhalb des Live-Updates, d.h., während des Verschiebens eines Knotens mit der Maus, ist es hilfreich, Informationen über das betreffende Objekt zu erhalten und zu verarbeiten.

Im Standardverhalten von VARCHART XGantt wird keine Information über den Zustand des betroffenen Objekts geliefert. Erst beim Loslassen der Maustaste liefert ein **ObjectModifying Event** Informationen über den alten (bevor die Maustaste gedrückt wurde) und neuen (nachdem die Maustaste losgelassen wurde) Zustand. Zusätzlich zeigt ein **ObjectModified Event** an, dass der Vorgang intern abgeschlossen ist:



Um bereits während der Mausektion - und nicht erst nach deren Abschluss - Informationen über das sich verändernde Objekt zu erhalten, kann man einige die Interaktion begleitende und beschreibende Events, die Interaction Events, nutzen. Darüberhinaus wurden mit XGantt-Version 5.0 Änderungen bei Aufrufzeitpunkt und Häufigkeit der Objekt-Events vorgenommen.



Betroffene Interaktionen

Es werden Events erläutert, die den Vorgang einer Interaktion in VARCHART XGantt und der daran beteiligten Objekte näher beschreiben, also "Drag(Drop)" Events bei Interaktionen, die in der Regel:

- mit Drücken der linken Maustaste auf einem Objekt beginnen
- mit gehaltener Maustaste Bewegungen durchführen
- mit einem Loslassen der linken Maustaste enden
- im Rahmen des Aktualisierungsverhaltens "Live Update" behandelt werden.

Terminologie

Zum besseren Verständnis werden hier noch einmal kurz einige der nachfolgend verwendeten Begriffe erläutert.

> Objekt-Events

Objekt-Events sind alle Events, die auch schon bisher bei den angesprochenen Interaktionen am Ende einer Aktion geworfen wurden, z.B. **VcDateLineModifying**, **VcDateLineModified**, **VcNodeModifying**, **VcNodeModified** usw.

> Live Update

Beim Live Update wird während einer Drag/Drop-Aktion permanent ein "Was wäre, wenn hier das Objekt aktualisiert würde?"-Szenario gezeigt. Es werden also verschiedene Kontexte, z.B. direkte oder abhängige Funktionalitäten bei einer Interaktion, zu verschiedenen Zeitpunkten abgearbeitet. Verschiebt man z.B. einen Knoten, so werden verschiedene Daten und seine Position verändert und abhängig davon verändern sich z.B. Kurven des Histogramms oder Summenbalken. Durch entsprechende Konfigurierung des Live Updates können diese Veränderungen sofort durchgeführt werden, oder erst, wenn man während der Bewegung eine festzulegende Zeit mit der Maus verharret oder aber auch erst am Ende der Aktion beim Loslassen der Maus.

Aktualisierungsverhalten "LiveUpdate" bearbeiten		
Kontexte (Objekte und Funktionen)		
Bezeichnung	Aktualisierungsmodus	Verzögerungszeit
Tabellen		
Spaltenbreite ändern	Beim Verschieben der Maus	500 ms
Zeitskalen		
Einheitenbreite ändern	Beim Verschieben der Maus	500 ms
Anfangsdatum des Zeitskalenab...	Beim Verschieben der Maus	500 ms
Stichtaglinien		
Datum ändern	Beim Verschieben der Maus	500 ms
Boxen		
Größe ändern	Beim Verschieben der Maus	500 ms
Position ändern	Beim Verschieben der Maus	500 ms
Verankerungsknoten ändern	Beim Verschieben der Maus	500 ms
Knoten		
Termine/Dauer ändern	Beim Verschieben der Maus	500 ms
Gruppe ändern	Beim Verharren während des Verschiebens	500 ms
Filtern/Mappen	Beim Verharren während des Verschiebens	500 ms
Gruppieren	Beim Verharren während des Verschiebens	500 ms
Automatische Terminberechnung	Beim Verharren während des Verschiebens	500 ms

Beispiel: Wie erfolgen die Aktualisierungen beim Verschieben eines Knotens wenn als Aktualisierungsverhalten "Beim Verschieben der Maus" gewählt ist?

Zeitlich direkte Anpassungen am Knoten:

- alle Datumswerte des Knotens
- Filter werden ausgewertet, wodurch z.B. andere Farben im Tabellenbereich erscheinen können
- Summenbalken
- Histogrammkurven

Anpassungen nach einer Wartezeit(500ms)

- Positionierung des Knotens z.B. in einer Gruppe
- Optimierung bei entsprechendem Layout der Knotenanordnung

Hierbei sollte darauf geachtet werden, nur diejenigen Aktualisierungen durchzuführen, die im Gesamtkontext der Aktion notwendig und sinnvoll sind, da ansonsten während der Aktion ein sehr unruhiges Bild entstehen könnte.

InInteraction Events

Seit VARCHART XGantt 5.0 SR3 ist es möglich, Objekt-Events auch während der Interaktion zu verarbeiten. Diese Objekt-Events, die während einer Interaktion erscheinen, werden als InInteraction-Events bezeichnet.

Wichtig: Die Verwendung der InInteraction Events muss dazu explizit erlaubt sein. Dies erfolgt entweder über die Eigenschaft **VcGantt.InInteractionEventsEnabled = true** oder auf der Eigenschaften-seite **Allgemeines**.

(Im Folgenden wird bei Interaktionen mit Knoten im Realmodus das Anzeigeobjekt **Real(knoten)** und der echte Knoten, also das Datenelement im Chart, **Chartknoten** genannt. Der Chartknoten ist während der Liveinteraktion im Diagrammbereich nicht sichtbar, weil er dort temporär durch den Realknoten ersetzt wird, sein Vorhandensein wirkt sich aber im Diagramm aus (Bandhöhe, Optimierung, Farbgebung im Tabellenteil usw.)).

Auf diese Art werden auch während der Interaktion passend zum angezeigten Phantom oder Realknoten entsprechende Informationen über die normalen Objekt-Events geliefert.

Beim Verschieben eines Knotens wird bei jedem Einrasten des Knotens (abhängig von der gewählten Zeiteinheit und der Schrittweite) ein **VcNodeModifying Event** gefeuert (gelbe Blitze). Der Realknoten zeigt hierbei die mögliche Position und das mögliche Aussehen und beschreibt diesen Zustand über den **VcNodeModifying Event**.

Der in den Event-Args übergebene Knoten (e.Node) stellt dabei den Zustand des Realknotens dar.

Achtung: Aus diesem Grund sind Abfragen auf Eigenschaften des Chartknotens nicht sinnvoll oder möglich sondern es können nur die Eigenschaften get/setDataField, AllData, ID ausgefragt/gesetzt werden!

Wird nun, abhängig vom gewählten Aktualisierungskontext, z.B. beim Verharren während des Verschiebens, das echte Objekt aktualisiert, wird dies über den **Modified Event** (grüner Blitz) angezeigt. Das kann, muss aber nicht zeitgleich mit den **Modifying Events** geschehen.

Wenn ein Knoten verschoben wird und als Aktualisierungsverhalten für das Verschieben "Beim Verschieben der Maus" gewählt ist, werden beide Events immer gleichzeitig kommen.

Fazit bisher:

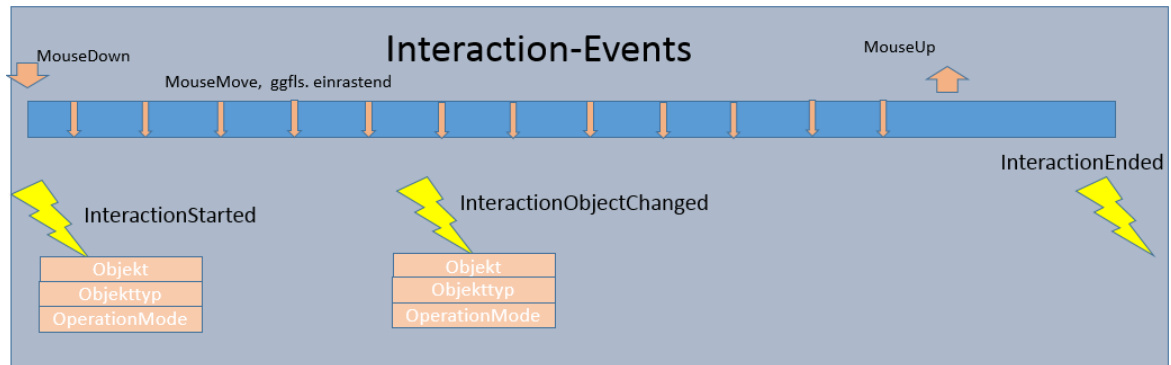
- Verschiebt man einen Knoten, wird seine Veränderung, angezeigt durch den Realknoten, permanent über den VcNodeModifying Event beschrieben.
- Wird der echte Knoten, also der Chartknoten, modifiziert, wird dies durch den VcNodeModified Event angezeigt.
- Am Ende der Interaktion, beim Loslassen der Maustaste, wird wie üblich das abschließende Eventpaar, bestehend aus VcNodeModifying und VcNodeModified Event, zur Verfügung gestellt.

Bei Events, die Realknoten verwenden, sind hierbei nun die betroffenen Objekte die echten Objekte.

Im letzten **VcNodeModifying Event** wird der Chartknoten (im Gegensatz zu den vorhergehenden **VcNodeModifying Events**) mit den im Verlauf der Interaktion zuletzt gesetzten Werten zur Verfügung gestellt, also der Zustand, der beim letzten kleinen grünen Blitz vorlag. **e.OldNode1** > **der EventArgs** beschreibt den Zustand zu Beginn der Aktion. Es ist also ein Vergleich zwischen Start- und Endzustand der Interaktion möglich.

Im letzten **VcNodeModified**-Event liegt nun wie immer der Chartknoten vor; alle internen Vorgänge sind abgeschlossen.

Interaction Events



Die Objekt-Events werden, wie beschrieben, nun während und am Ende einer Interaktion geworfen. Die Signatur eines Eventhandlers, z.B. die des **VcNodeModifying** Events, unterscheidet sich dabei nicht. Wie kann man nun unterscheiden, ob der Event während oder am Ende der Interaktion geworfen wurde?

Dies könnte wichtig sein, da ja z.B. nicht bei jeder Mausbewegung die Änderung eines Vorganges zeitaufwändig in eine Datenbank geschrieben werden soll; das soll selbstverständlich erst nach Abschluss der Aktion geschehen.

Hierfür gibt es einige Events, die die Interaktion begleiten und beschreiben und die während der Interaktion in den Objekt-Events ausgewertet werden können.

Sobald die linke Maustaste gedrückt wird, erfährt man über den **VcInteractionStarted** Event, auf welchem Objekt der Mauszeiger steht (Objekt und Objektyp) und was mit dem Objekt passiert. Hier kann man alles vorbereiten, was für die Interaktion notwendig ist.

Tipp: Hier kann auch noch objekt- und kontextspezifisch das Updateverhalten umgeschaltet werden! Man könnte also im Extremfall einen Knoten komplett dynamisch, einen anderen mit blauem Phantomrahmen reagieren lassen. Darüber hinaus kann noch durch entsprechende Setzung (`InInteractionEventsEnabled`) individuell entschieden werden, ob die ObjectEvents auch während der Interaktion kommen sollen oder nicht.

Beispiel Knoten:

Durch

- Objekt: KnotenObjekt
- Typ: vcObjTypeNodeInDiagram
- OperationMode: vcIIMMoveNode

wird beim Drücken der linken Maustaste im **VcInteractionStarted** Event angezeigt, dass damit begonnen wurde, einen Knoten im Diagramm zu verschieben.

Informationen oder Elemente, die die Interaktion begleiten sollen, können hier initialisiert werden.

> Erzeugen von Objekten

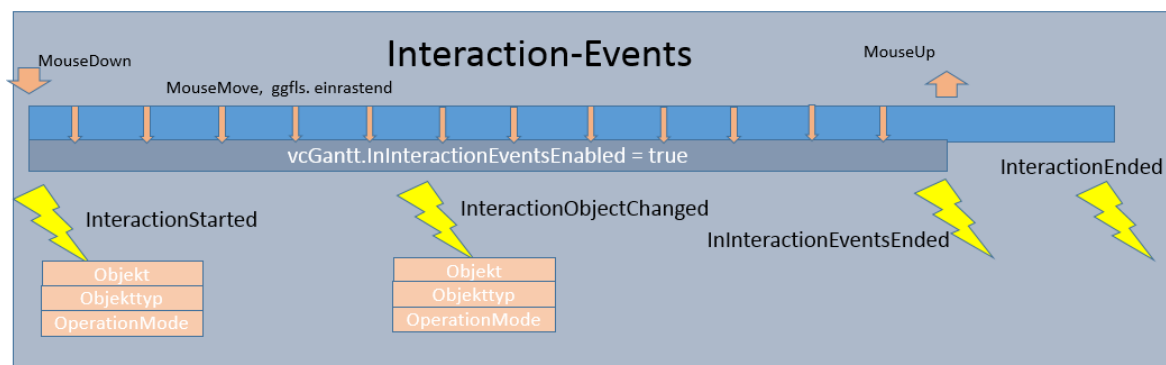
Bei manchen Interaktionen steht initial noch kein Objekt zur Verfügung, z.B. beim Erzeugen eines Knotens oder einer Box. In diesem Fall erscheint der Event **VcInteractionObjectChanged**, sobald intern das Objekt erzeugt wurde. Bei Knoten ist dies der echte Chartknoten.

Ist die Aktion abgeschlossen, wird dies durch den Event **VcInteractionEnded** angezeigt. Hier kann man alles, was man während der Interaktion an zusätzlichen Elementen verwendet hat, abbauen.

Bei Erzeugung von neuen Objekten mit Interaction-Events ist der Ablauf wie folgt:

- VcInteractionStarted
- VcInteractionObjectChanged
- Modifying/Modified Events, die Veränderungen bei der Erzeugung eines Elementes anzeigen
- Creating und Created Events
- VcInteractionEnded

> InInteraction Events während der Interaktion freigeschaltet



Wenn zusätzlich die Interaction Events während der Interaktion erlaubt sind (**vcGantt.InInteractionEventsEnabled = true**), dann gibt es einen weiteren Event, der beim Loslassen der Maustaste das Ende dieser Events anzeigt: **VcInInteractionEventsEnded**.

132 Wichtige Konzepte: Interaction Events

So können nun leicht die Objekt-Events während der Interaktion von denen am Ende der Interaktion unterschieden werden: Wenn dieser Event geworfen worden ist, so wird der nächste Objekt-Event der abschließende Event sein.

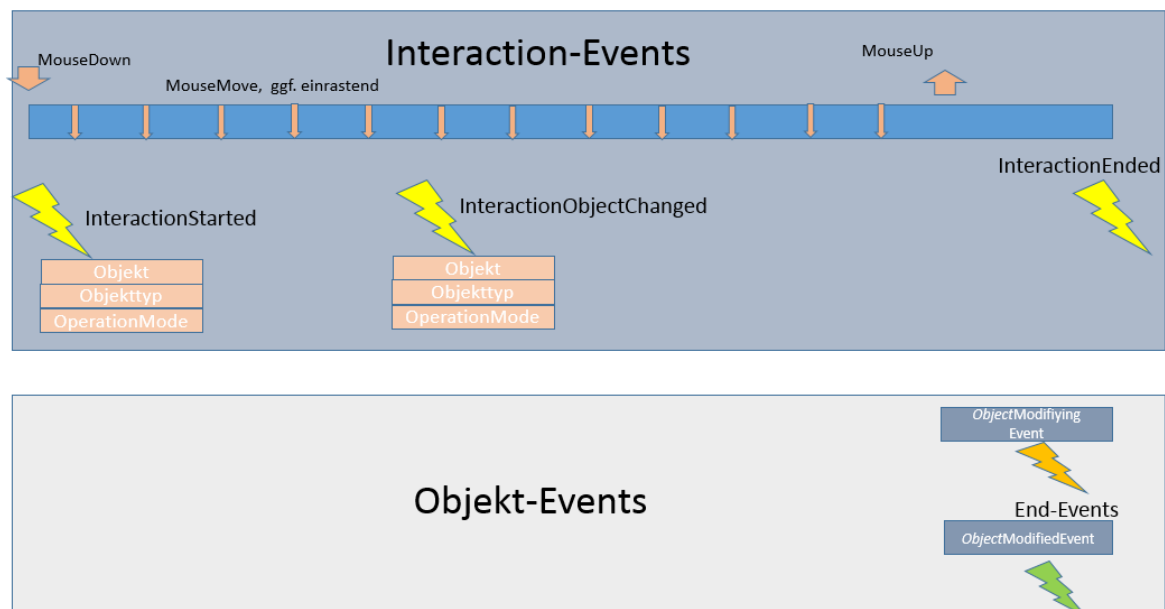
> Mögliche Szenarien

Bei der Verwendung der Interaction Events gibt es also zwei mögliche Zustände:

Steuerung einer Interaktion bei

- deaktivierten InInteraction Events
- aktivierten InInteraction Events

> Zusammenspiel mit den Events der beteiligten Objekte bei deaktivierten InInteraction-Events



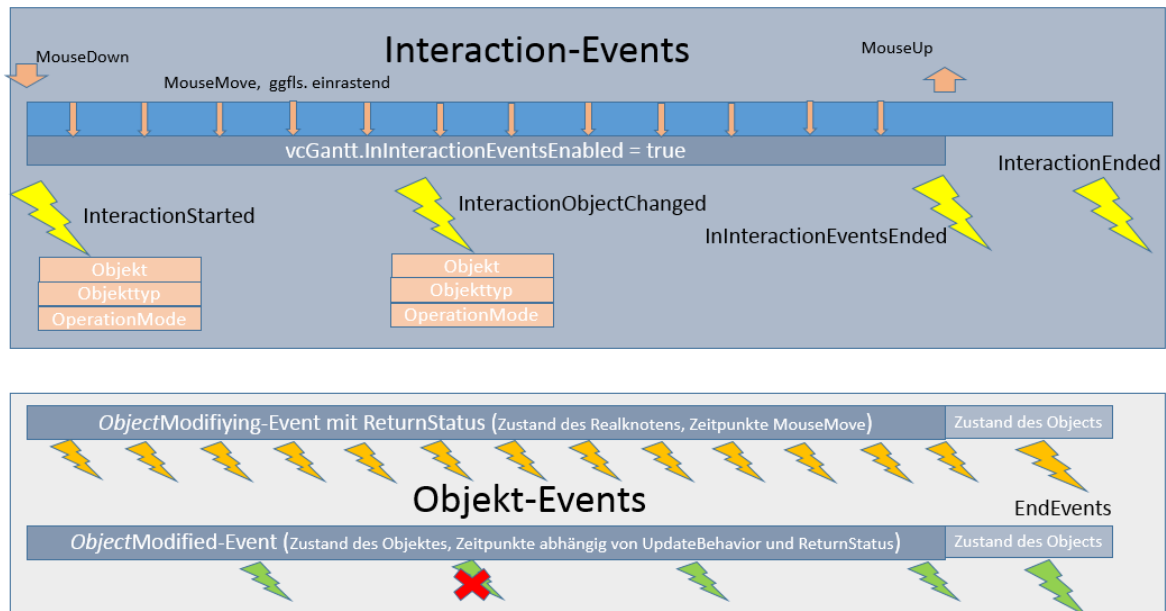
Hier sieht man das Zusammenspiel der Interaction- (gelbe Blitze) und Objekt-Events (ockerfarbene und grüne Blitze) bei deaktivierten InInteraction Events (**vcGantt.InInteractionEventsEnabled = false**):

Die Interaktion wird gestartet, angezeigt durch den Event **InteractionStarted**.

Beim Loslassen der Maustaste kommen zunächst die Objekt-Events, z.B. beim Knoten **VcNodeModifying** und **VcNodeModified1** - **hinsichtlich der ObjectEvents also das alte Verhalten, sodass der vorhandene Code in den ObjectEvents nicht geändert werden muss, wenn die InInteractionEvents nicht verwendet werden.**

Danach kommt der Event **VcInteractionEnded**, der das Ende der Interaktion anzeigt.

> **Zusammenspiel mit den Events der beteiligten Objekte bei aktivierten InInteraction-Events**



Sind die InInteraction Events in Verwendung, erscheinen folgende Events:

- **VcInteractionStarted** beim Drücken der linken Maustaste
- Modifying- und Modified Events bei der Mausbewegung
- **VcInInteractionEventsEnded** und dann die abschließenden Objekt-Events beim Loslassen der linken Maustaste
- **VcInteractionEnded** um das Ende der Interaktion anzuzeigen

Beispiel1: Verschieben eines Knotens:

Die Interaktion wird gestartet durch Drücken der linken Maustaste, wenn der Mauszeiger auf einem Knoten steht. Es erscheint der Event **VcInteractionStarted**

Die beim Bewegen des Mauszeigers auftretenden Events zeigen den Zustand des Realknotens (**VcNodeModifying**) und bei Updates (**VcNodeModified**) des Chartknotens an.

Wird die Maustaste losgelassen, so erscheint der Event **VcInInteractionEventsEnded1**>

Die danach auftretenden Objekt-Events **VcNodeModifying** und **VcNodeModified** zeigen den Zustand des Chartknotens am Ende der Interaktion an.

Als letztes erscheint der Event **VcInteractionEnded**.

> **Beispiel: Verhalten der Objekt-Events beim Knoten-Aktualisierungsverhalten "Verschieben mit der Maus"**



Da der **VcNodeModifying Event** die Modifizierung des EventReturnstatus (e.ReturnStatus) erlaubt, gilt dies nun auch während der Interaktion.

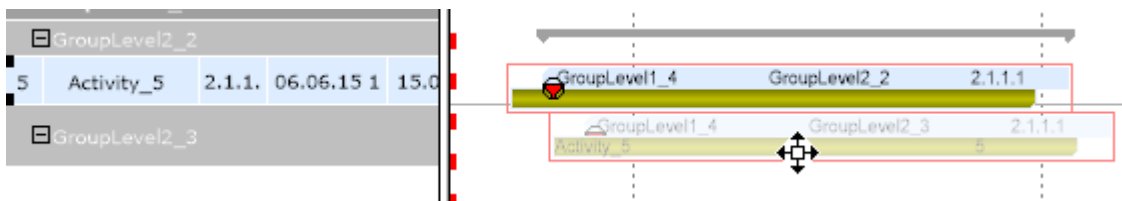
Wird also über e.ReturnStatus = ReturnStatusFalse angezeigt, dass die gelieferten Daten nicht „valide“ sind, so wird im Chart die Objektaktualisierung beim nächsten möglichen Update nicht durchgeführt, ein entsprechender VcNodeModified Event wird nicht geworfen.

Dies wird visualisiert, indem das das Objekt "stehen bleibt" und die aktuelle Position weiter durch das Phantom angezeigt wird.

Bei Objekten mit Realdarstellung (momentan nur Knoten und Knotenboxen) wird der Zustand folgendermaßen dargestellt:

Die aktuelle Position wird durch ein aufgehelltes Real dargestellt; dessen Werte sind auch die, die in den Events weiterhin geliefert werden.

Der letzte valide Zustand, also der, bei dem zum letzten Mal e.ReturnStatus nicht ReturnStatusFalse war, wird durch ein weiteres Real dargestellt, das quasi dort "hängenbleibt". So sind beide notwendigen Informationen visualisiert.



Beim Knoten entsprechen die Werte des letzten validen Zustandes, also des hängengebliebenen Reals, dann dem **e.OldNode** im **VcNodeModifying-Event**.

Falls der letzte **VcNodeModifying-Event** vor dem **VcInInteractionEventsEnded1> Event** mit **ReturnStatusFalse** abgeschlossen wurde, wird in den End-Events der letzte valide Zustand geliefert.

Dann kann dort noch mal entschieden werden, ob dieser übernommen werden soll oder nicht. Setzt man im End-Event `ReturnStatusFalse`, wird der originale Startzustand wiederhergestellt.

Tipp: In der Praxis empfiehlt es sich, ein "begleitendes `InteractionInfo`"-Objekt zu erstellen, das in den Events die notwendigen Informationen zur Interaktion liefert und entsprechend ausgewertet werden kann

3.13 Knoten (Vorgänge)

Ein Knoten (Vorgang) entspricht einem Datensatz aus der Maindata-Tabelle. Knoten können über die API geladen oder interaktiv vom Anwender erzeugt werden.

> Knoten erzeugen

Auf der Eigenschaftenseite **Knoten** können Sie festlegen, ob der Anwender

- neue Knoten durch Ziehen mit der Maus erzeugen kann (im **Modus: Knoten erzeugen**) (**Neue Knoten zulassen**)
- neue Knoten durch Doppelklick erzeugen kann (**Neue Knoten durch Doppelklick**)
- neue Knoten direkt im Dialogfeld **Vorgänge bearbeiten** bearbeiten kann (**Neuen Knoten bearbeiten**).

Wenn zur Laufzeit ein neuer Knoten durch Ziehen mit der Maus erzeugt wird, erscheint dabei die Infobox **Knoten neu anlegen**, die Ihnen Anfangs- und Endtermin sowie die Dauer des neuen Knotens anzeigt.

Knoten neu anlegen	
Anfang:	09.09.2007
Ende:	11.09.2007
Dauer:	2 Tage

Wenn auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Neuen Knoten bearbeiten** aktiviert ist, öffnet sich das Dialogfeld **Vorgänge bearbeiten**, sobald Sie einen neuen Knoten erzeugt haben. Hier werden die Daten des interaktiv erzeugten neuen Knotens angezeigt, und Sie können sie nun bearbeiten. Jedes interaktive Neuanlegen eines Knotens wird der Applikation mit dem Ereignis **OnNodeCreate** bzw. **OnNodeCreate-CompleteEx** mitgeteilt.

Sie können Knoten auch über die API mit der Methode **InsertNodeRecord** anlegen.

> Knoten löschen

Zur Laufzeit können Sie Knoten löschen, indem Sie den Cursor auf dem betreffenden Knoten positionieren und die rechte Maustaste drücken. Es erscheint folgendes Kontextmenü:

Daten bearbeiten...
Knoten löschen
Teildiagramm erstellen
Gesamtdiagramm wiederherstellen

Wählen Sie die Option **Knoten löschen**.

Das interaktive Löschen eines Knotens löst das Ereignis **OnNodeDelete** aus.

Sie können Knoten auch über die API mit der Methode **DeleteNodeRecord** löschen.

> Weitere Festlegungen für Knoten

Auf der Eigenschaftenseite **Knoten** können Sie außerdem Folgendes festlegen:

- die Datenfelder, in die die Daten für Start, Ende und Dauer von interaktiv angelegten Vorgängen geschrieben werden sollen.
- ob arbeitsfreie Zeiten hervorgehoben werden sollen (Bei Rechtecklayern werden arbeitsfreie Zeiten dann durch eine durchgezogene Linie angezeigt.)
- ob den Knoten Kalender zugewiesen werden sollen (Die Kalenderzuweisung wirkt sich zum einen beim Verschieben von Vorgängen aus: Anfang und Ende der Vorgänge werden nicht auf arbeitsfreie Tage gelegt. Zum anderen werden beim Berechnen der Dauer von Vorgängen die arbeitsfreien Zeiten berücksichtigt. Derzeit ist standardmäßig ein Fünf-Tage-Kalender definiert ("WeekCalendar").)
- das Datenfeld, das den Namen des für einen Knoten zu verwendenden Kalenders enthalten soll, wenn ein individueller Kalender gefordert wird.
- ob der Anwender mehrere markierte Knoten gemeinsam verschieben können soll
- ob ein markierter Knoten als Ganzes (d. h. mit all seinen Layern) verschoben werden können soll.

> Ereignisse

Sie können auf die folgenden Ereignisse reagieren:

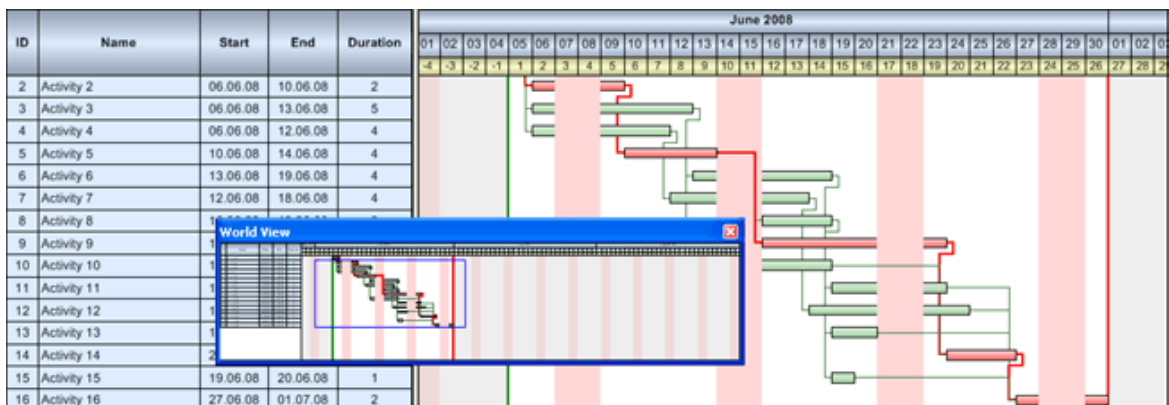
- OnNodeCreate
- OnNodeCreateCompleteEx
- OnNodeDelete
- OnNodeLClick
- OnNodeLDbClick
- OnNodeModify
- OnNodeModifyComplete

138 Wichtige Konzepte: Knoten (Vorgänge)

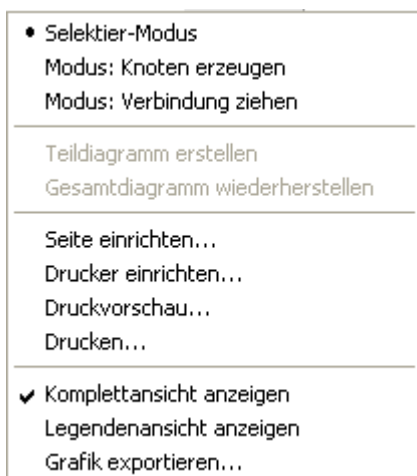
- OnNodeRClick
- OnNodesMarkEx

3.14 Komplettansicht (World View)

Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm und ggf. das Histogramm angezeigt wird. Jeweils ein Rahmen zeigt an, welchen Diagramm- bzw. Histogrammausschnitt das Hauptfenster gerade anzeigt. Wenn Sie mit der Maus einen dieser Rahmen verschieben, wird der angezeigte Ausschnitt des Hauptfensters beim Loslassen der Maustaste entsprechend verschoben. In ähnlicher Weise können Sie durch Größer- oder Kleinerziehen des Rahmens in der Komplettansicht den realen Bildausschnitt zoomen. Umgekehrt ändern sich Position bzw. Größe des entsprechenden Rahmens, wenn der Ausschnitt im Hauptfenster gescrollt oder gezoomt wird.



Zur Laufzeit können Sie über den Menüpunkt **Komplettansicht anzeigen** des Standard-Kontextmenüs die Komplettansicht ein- und ausschalten.



Auf der Eigenschaftenseite **Zusätzliche Ansichten** können Sie die Eigenschaften der Komplettansicht festlegen. Einzelheiten hierzu finden Sie im Kapitel **Eigenschaftenseiten und Dialogfelder**, Eigenschaftenseite **Zusätzliche Ansichten**.

140 Wichtige Konzepte: Komplettansicht (World View)

Alternativ können Sie die Optionen der Komplettansicht auch über die API-Eigenschaft **VcGantt.VcWorldView** festlegen.

3.15 Layer

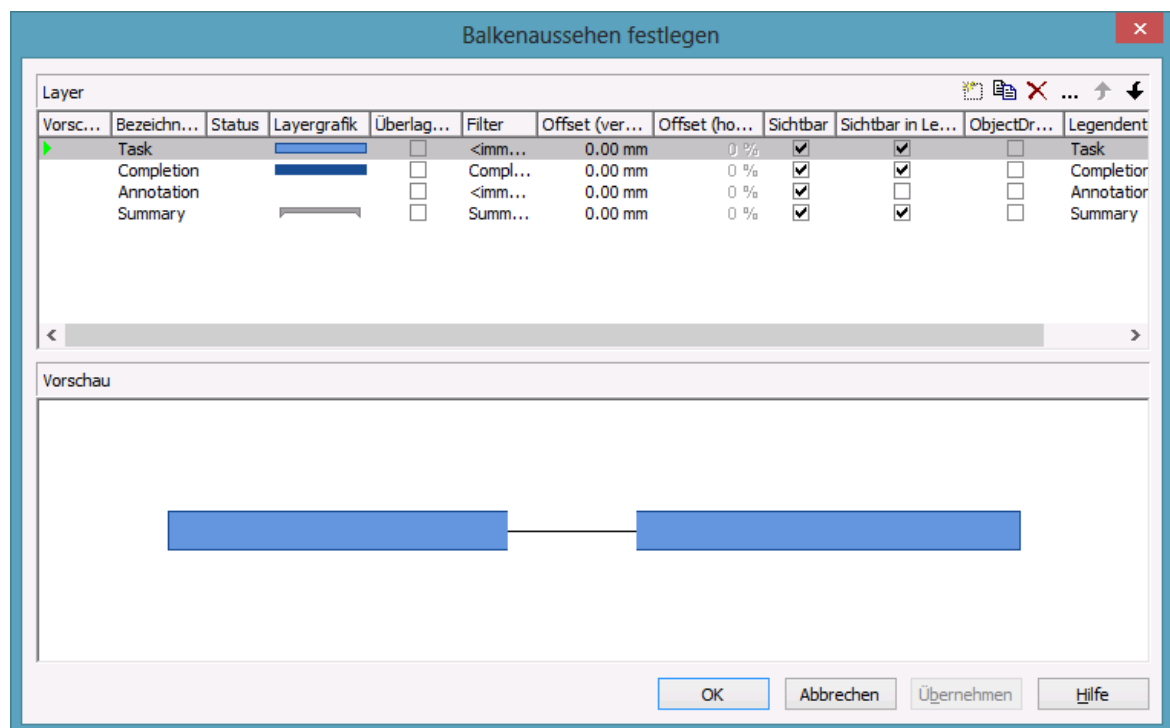
Ein Layer stellt einen Termin (Symbollayer oder Bitmaplayer) oder ein Terminpaar (Rechtecklayer, keilförmige Layer oder Linienlayer) grafisch dar.

Vorgänge (Knoten) werden durch einen oder mehrere Layer grafisch festgelegt. Wenn ein Vorgang aus mehreren Layern besteht, werden die einzelnen Layer dieses Vorgangs grafisch schichtweise übereinander gelegt. Dabei wird mit dem Layer der niedrigsten Priorität angefangen. Der Layer mit der höchsten Priorität liegt zuoberst.

Jedem Layer wird ein Filter zugewiesen, der angibt, unter welchen Bedingungen Vorgänge durch diesen Layer dargestellt werden.

Die Layer können sich hinsichtlich ihres Musters, ihrer Hintergrund- und Musterfarbe und/oder ihrer Beschriftung unterscheiden. Außerdem können sie unterschiedliche Höhen und einen horizontalen und/oder vertikalen Offset haben. Dadurch können Sie sicherstellen, dass alle Layer, die einem Knoten zugeordnet sind, auch sichtbar sind.

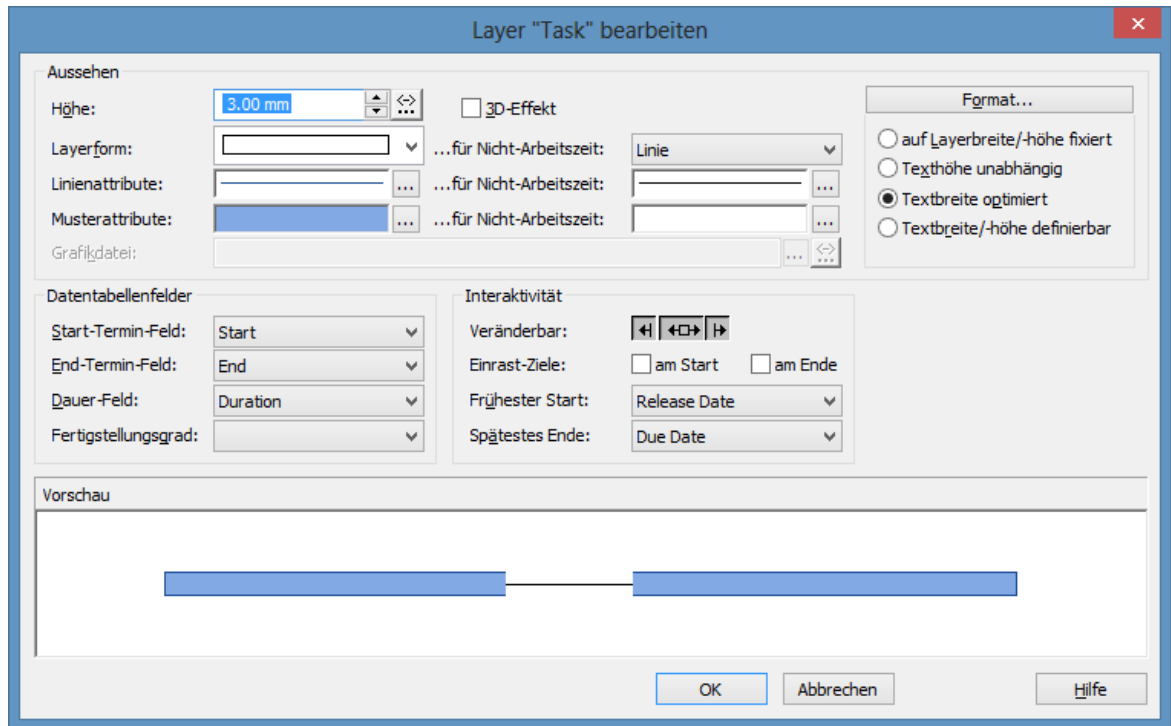
Sie können im Dialog **Balkenaussehen festlegen** beliebig viele Layer definieren. Alle vorhandenen Layer werden hier, nach Zeichnungsreihenfolge sortiert, angezeigt.



Mit Hilfe der Schaltflächen oberhalb der Layer-Tabelle können Sie Layer hinzufügen (, kopieren (, löschen () oder bearbeiten ().

142 Wichtige Konzepte: Layer

Um einen Layer zu bearbeiten, wählen Sie einen Layer aus der Liste aus und klicken Sie auf die Schaltfläche **Layer bearbeiten** (...) oder klicken doppelt auf die **Layergrafik**. Das Dialogfeld **Layer bearbeiten** öffnet sich. Hier können Sie die grafischen Attribute für den gewählten Layer einstellen.

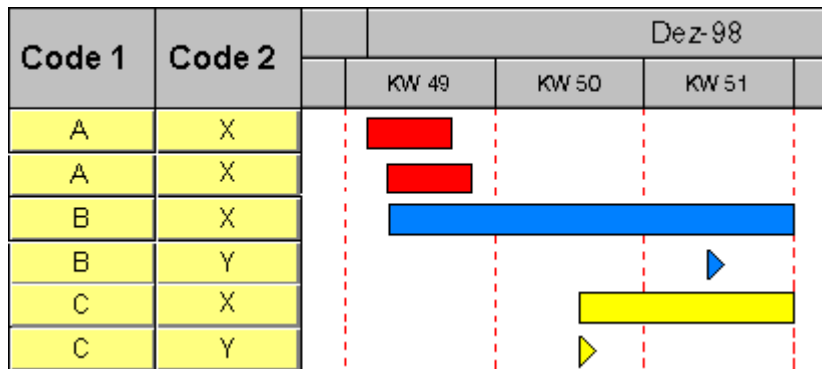


> Filter für Layer

Durch die Verwendung von Filtern erreicht man, dass ein Layer nur einer ausgewählten Menge von Knoten zugewiesen wird, in Abhängigkeit von den Daten.

Um einen Filter zu bearbeiten, klicken Sie im Dialog **Balkenaussehen festlegen** auf das **Filter**-Feld. Es erscheinen zwei Schaltflächen. Klicken Sie auf die **Bearbeiten**-Schaltfläche, um den **Filter verwalten**-Dialog zu öffnen. Von hier können Sie das Dialogfeld **Filter bearbeiten** öffnen, wo Sie die Filterkriterien bearbeiten können. (Siehe hierzu: "Wichtige Begriffe: Filter").

Im folgenden Beispiel ist für "Code2 = X" ein Rechtecklayer und für "Code2 = Y" ein Symbollayer definiert. Die Farbgebung erfolgt durch das Mapping bezüglich Code1.



> Layerformen

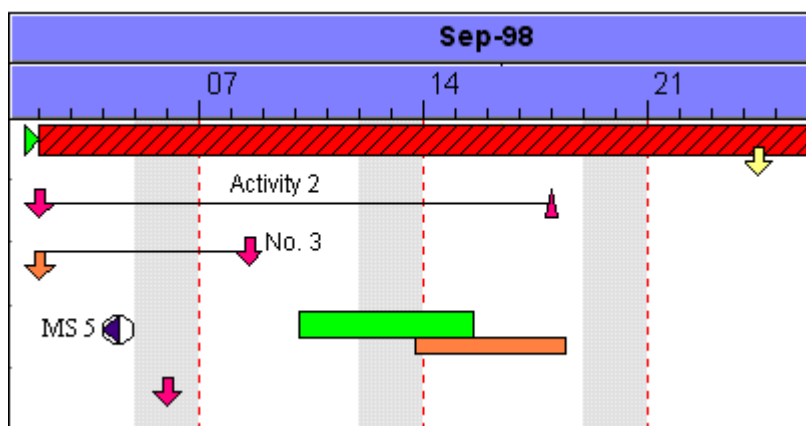
Es gibt verschiedene Layerformen: Rechteck-Layer, keilförmige Layer, Linien-Layer, Symbol-Layer, Bitmap-Layer und den "unsichtbares Symbol"-Layer. Die Layer-Form wird im Dialogfeld **Layer bearbeiten** unter **Layer-Form** ausgewählt.

Symbol-Layer werden zur Darstellung spezieller Termine verwendet. Eine Reihe von Symbol-Layern ist vordefiniert. Zusätzlich können Sie Bitmap-Layer selbst definieren, um Ihre eigenen Grafiken (z. B. Firmenlogos) als Layer zu verwenden. Unter **Grafikdatei** können Sie die zu verwendende Bitmap-Datei auswählen.

Terminpaare werden durch Rechteck-Layer, keilförmige Layer oder Linien-Layer grafisch dargestellt. Keilförmige Layer können verwendet werden, um ansteigende bzw. ausklingende Aktivitäten darzustellen, beispielsweise am Anfang bzw. am Ende von Vorgängen.

Der Layer **unsichtbares Symbol** ist mit Ausnahme seiner Beschriftung unsichtbar und wird nicht in der Legende dargestellt. Er kann zur zusätzlichen Beschriftung von Vorgängen verwendet werden.

Die Kombination von Layer-Formen, Hintergrundfarben, Mustern, Beschriftungen und Filtern ermöglicht eine Vielfalt möglicher Layer. Einige Beispiele zeigt die folgende Abbildung:



> Fertigstellungsgrad

Sie können sich den prozentualen Fertigstellungsgrad jedes Vorgangs auf einen Blick zeigen lassen.

Gehen Sie dazu folgendermaßen vor: Erstellen Sie einen Layer **Fertiggestellt** und bearbeiten Sie ihn im Dialog **Layer bearbeiten**. Hier können Sie für Rechtecklayer und keilförmige Layer das Datenfeld auswählen, das den prozentualen Fertigstellungsgrad eines Vorgangs enthält.

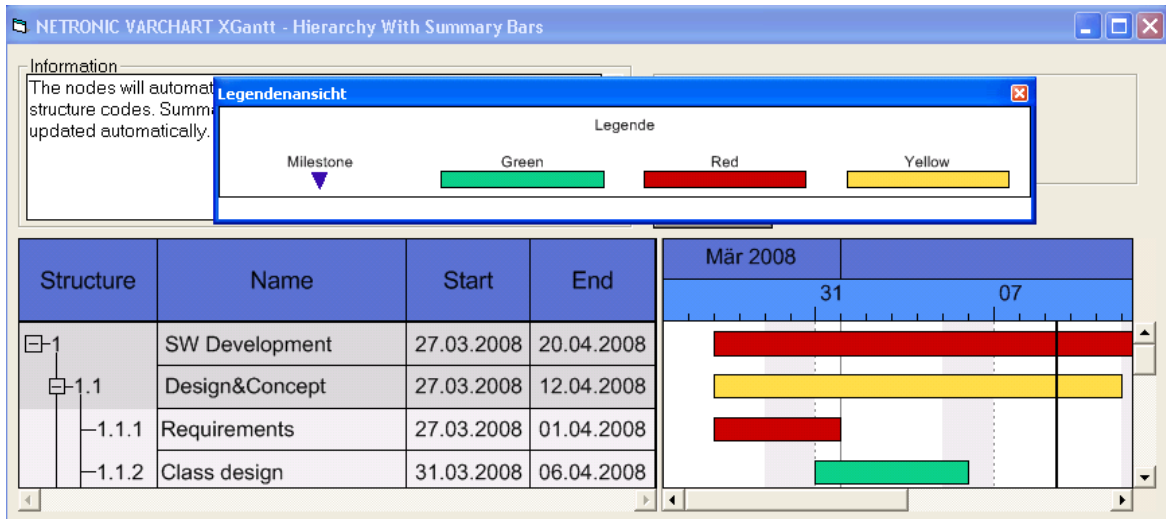
Wählen Sie für den Layer **Fertiggestellt** unter **Fertigstellungsgrad** beispielsweise das Datenfeld **abgeschlossen %** aus, das einen numerischen Wert zwischen 0 und 100 enthalten muss. Aus den Terminen und dem Fertigstellungsgrad wird nun ein neuer Endtermin berechnet. Vereinbaren Sie nun die grafischen Attribute des Layers so, dass der Layer "Fertiggestellt" deutlich von dem Standard-Layer zu unterscheiden ist, beispielsweise durch ein Schraffurmuster oder eine andere Farbe.



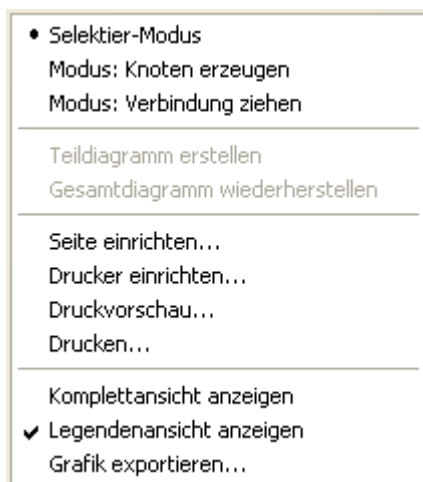
Vorgang, der zu 90 % fertiggestellt ist

3.16 Legendenansicht (Legend View)

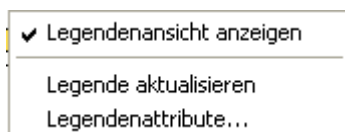
Die Legendenansicht ist ein zusätzliches Fenster zur Darstellung einer Legende auf dem Bildschirm. Das Aussehen der Legende wird festgelegt im Dialog **Legendenattribute**, der über die Eigenschaftenseite **Außenbereich** zu erreichen ist, oder über die Legendenattribute des Objektes **VcBorderBox**.



Zur Laufzeit können Sie über den Menüpunkt **Legendenansicht anzeigen** des Standard-Kontextmenüs die Legendenansicht ein- und ausschalten.



Die Legende verfügt über ein eigenes Kontextmenü, über das die Legendenansicht ebenfalls ein- und ausgeschaltet werden kann.



Außerdem können Sie über das Kontextmenü auch den Dialog **Legendenattribute** aufrufen sowie die Legende aktualisieren.

Eine Aktualisierung über das Menü kann notwendig sein, da nach Änderungen im Diagramm die Legende nicht automatisch aktualisiert wird. Werden also zum Beispiel Knoten hinzugefügt oder gelöscht, muss eine Aktualisierung entweder über das Kontextmenü oder durch Aus- und Einschalten der Legende durchgeführt werden. Dies gilt auch für das Laden von Knoten. Wenn für die Legendenansicht auf der Eigenschaftenseite **Zusätzliche Ansichten** die Option **Beim Start sichtbar** eingestellt wurde, aber zum Zeitpunkt des Aufbaus noch keine Knoten geladen waren, bleibt die Legende bis zur Aktualisierung leer.

Auf der Eigenschaftenseite **Zusätzliche Ansichten** können Sie die Eigenschaften der Legendenansicht festlegen. Einzelheiten hierzu finden Sie im Kapitel **Eigenschaftenseiten und Dialogfelder, Eigenschaftenseite Zusätzliche Ansichten**.

Alternativ können Sie die Optionen der Legendenansicht auch über die API mit der Eigenschaft **VcGantt.VcLegendView** festlegen.

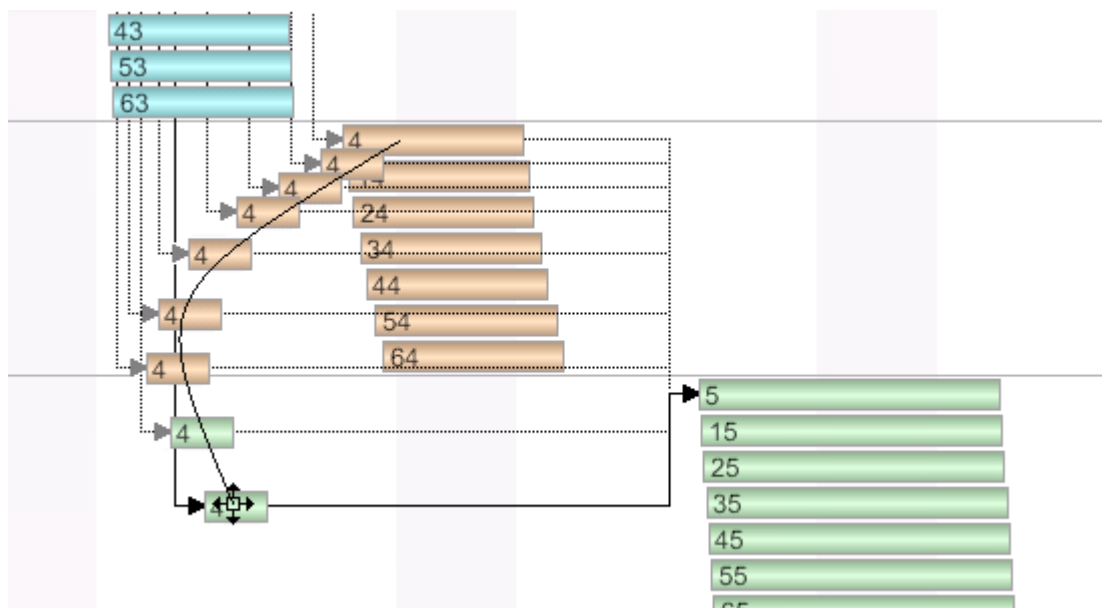
3.17 Live Update

Was ist Live Update?

Mit der Live Update Funktionalität, neu ab der Version XGantt 5, werden Auswirkungen von Mausinteraktionen bereits während der Interaktion visualisiert und nicht erst nachdem sie beendet sind.

Bislang wurde bei Interaktionen mit Phantomen gearbeitet und der Gantt-Graph gab Rückmeldung über die Konsequenzen der Aktion für die Gesamtplanung, sobald sie durch Loslassen der Maustaste an der gewünschten Stelle beendet war. Durch Live Update kann man jetzt schon während der Interaktion erkennen, welche Änderungen aus einer Mausaktion resultieren, da mit jeder Mausaktion die Ansicht aktualisiert wird, d.h. die Änderungen werden stetig wiederholt direkt auf dem Objekt ausgeführt und führen damit zu einem Live Update des Objektes und der Grafik.

Während der Verschiebeinteraktion sieht man also zu jedem Zeitpunkt eine zur jeweiligen Mausposition passende Visualisierung des Knotens mit den angehängten Links.



Arten von Datenänderungen

Bei der Änderung von Daten und ihrer anschließende Auswertung unterscheidet man zwischen zwei Arten:

- Änderungen, deren Auswirkungen sich nur auf das einzelne Objekt beziehen, z.B. einfache Datumsänderung, nachfolgend **individuelle**

Änderungen genannt. Individuelle Änderungen treten bei jeder Interaktion auf.

- Änderungen, deren Auswirkungen sich nicht nur auf das einzelne Objekt beziehen, sondern bei denen ganze Strukturen verändert werden können, z.B. Gruppierung oder Optimierung, nachfolgend **strukturelle Änderungen** genannt.

Strukturelle Änderungen können z.Zt. Nur beim Verschieben von Knoten und Gruppen auftreten, denn nur Knoten und Gruppen können in Strukturen zusammengefasst und angeordnet werden.

Strukturelle Änderungen werden timer-gesteuert vorgenommen (s. weiter unten **Timer-gesteuertes Live-Update**. Ein **OldNode** bzw. **PreviewNode** ist nicht vorgesehen.

Nach Durchführung einer strukturellen Änderung wird der verschobene Knoten automatisch wieder unter den Cursor gescrollt (Node-Tracking).

Betroffene Aktionen

Vom Live Update betroffen sind Verschiebe-Aktionen von Knoten und Gruppen sowie interaktives Anlegen von Knoten und Links.

> Verschieben von Knoten und Links im Diagramm

Knoten und Links können optisch frei verschoben werden. Dabei wird sowohl die horizontale als auch vertikale Lage des Knotens immer an der Mausposition ausgerichtet, sodass der verschobene Knoten zu jedem Zeitpunkt unter dem Mauscursor steht. Angehängte Links, die mit Linkrouting <orthogonal> oder <straight> gezeichnet werden, werden entsprechend mitgeführt. Das Linkrouting <distinguish> funktioniert hier nicht, es wird dann <orthogonal> verwendet. Bei der Positionsänderung wird auch die Visualisierung des Knotens und der Links ständig aktuell gehalten, d.h. Filterung (=>Layer) und Mapping werden auf das gesamte Konstrukt angewendet. An der bisherigen Knotenposition bleibt ein leerer Bereich, was den Verschiebeeffekt noch verstärkt. Man zieht optisch den Knoten von seinem bisherigen Platz weg. Dazu wird der Knoten mit seinen Links auf **VC_VISIBILITY=VC_NO** gesetzt. Kopien vom Knoten und den Links werden angelegt und beim Verschiebevorgang aktualisiert.

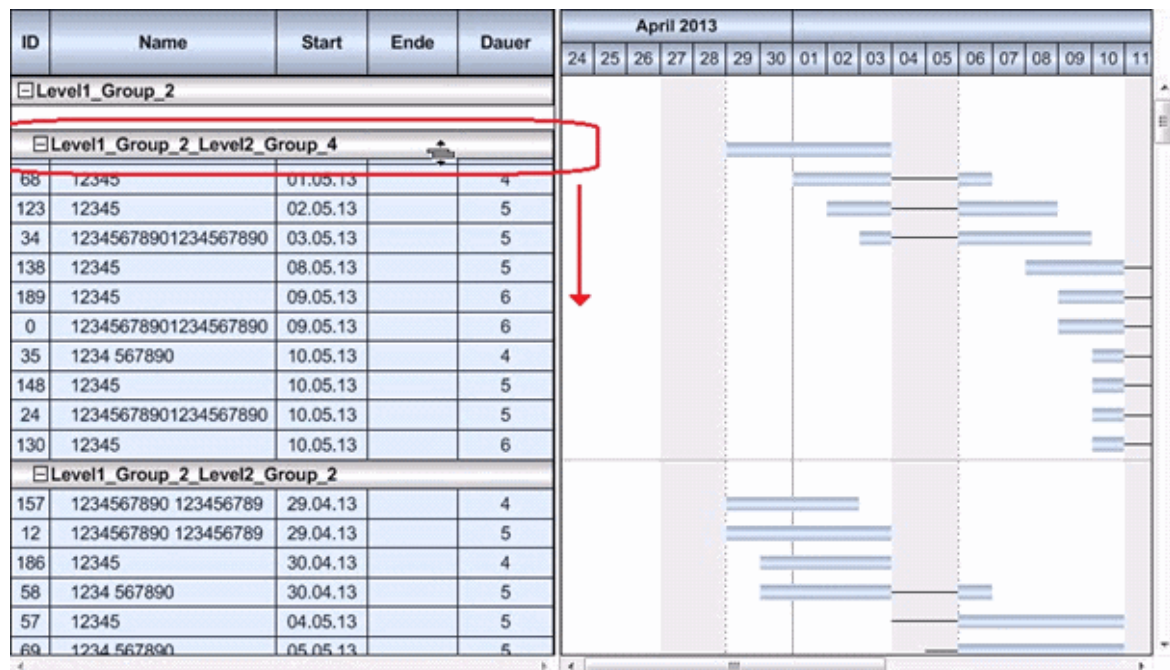
> Verschieben von Gruppen

VARCHART XGantt erlaubt dem Anwender, Gruppen innerhalb ihrer Ebene interaktiv zu verschieben. Dies geschieht, indem er im Diagramm entweder den Summenbalken oder Gruppenknoten vertikal verschiebt oder in

der Tabelle das jeweilige Tabellenformat vertikal verschiebt. Diese Strukturänderung entspricht einer manuellen Umsortierung, zu der es kein datenmäßiges Äquivalent gibt; es findet also keine Datenänderung statt.

Nach Durchführung der Änderung wird der verschobene Summenbalken/Gruppenknoten bzw. das verschobene Tabellenformat automatisch wieder unter den Cursor gescrollt. Dieses Scroll-Verhalten wird hier als Group-Tracking bezeichnet.

Im Diagrammbereich wird ein VARCHART-Knotenphantom mit Real-Darstellung des Summenbalkens/Gruppenknotens verwendet; im Tabellenbereich ein VARCHART-Knotenphantom mit Real-Darstellung der Tabellenbox. Da bei dem Verschiebevorgang keine Datenänderung stattfindet, bleibt die Real-Darstellung unverändert.



Timer-gesteuertes Live Update

Die gesamte Grafik wird durch die ständigen (ggf. umfangreichen) visuellen Änderungen sehr unruhig und der direkte Zustandswechsel ohne die Möglichkeit von Animationen kann verwirrend oder sogar störend wirken.

Daher ist eine Alternative zum direkten Zustandswechsel sinnvoll. Bei strukturellen Änderungen soll die Grafik nicht ständig aktualisiert werden, sondern gesteuert über einen Timer: Wenn der Anwender während der Interaktion mit der Maus einen Moment verharret, wird erst nach kurzer, aber signifikanter Wartezeit die strukturelle Änderung durchgeführt und die Grafik aktualisiert. Der Anwender sieht dann eine zur jeweiligen Mausposition passende Grafik. Nun kann er die Interaktion fortsetzen, denn

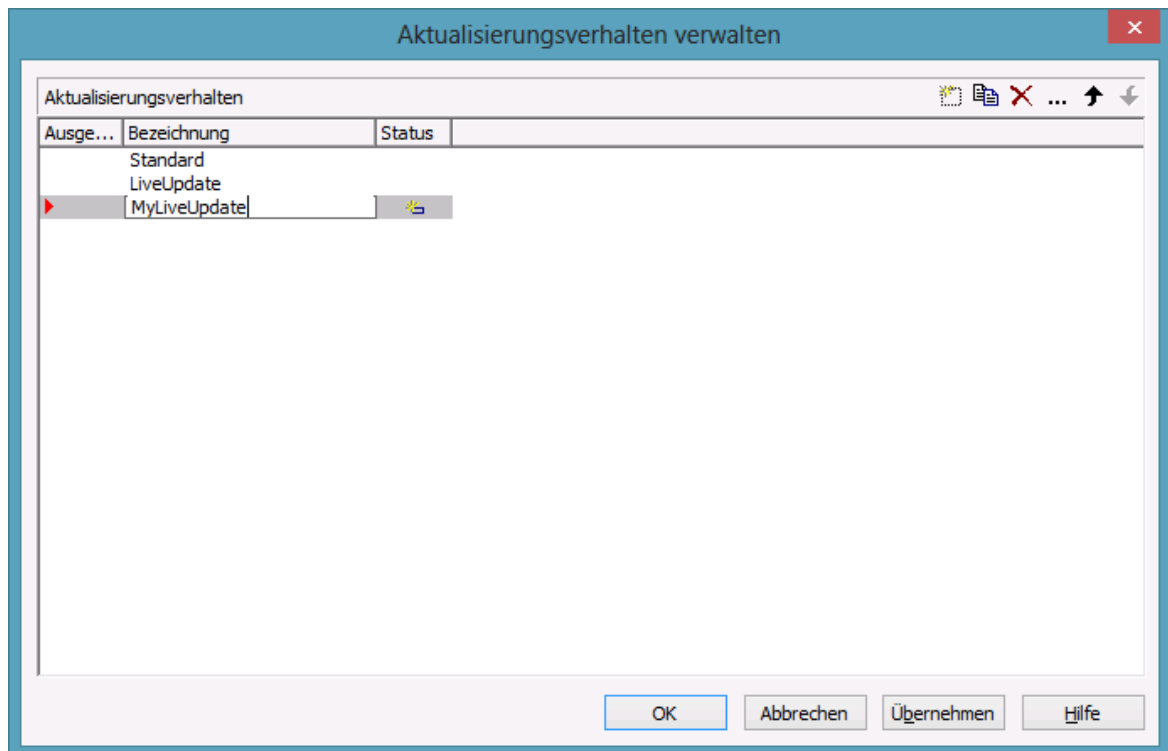
er bewegt die Maus noch immer mit gedrückter Taste. Dabei werden die strukturellen Änderungen wieder zurückgehalten bis er erneut verharret. Wiederum werden die strukturellen Änderungen erst nach kurzer, aber signifikanter Wartezeit durchgeführt und die Grafik aktualisiert. Der Prozess wiederholt sich bis zum Ende der Interaktion (Loslassen der Maustaste). Bei diesem Verfahren bleibt die Grafik während der Interaktion relativ ruhig.

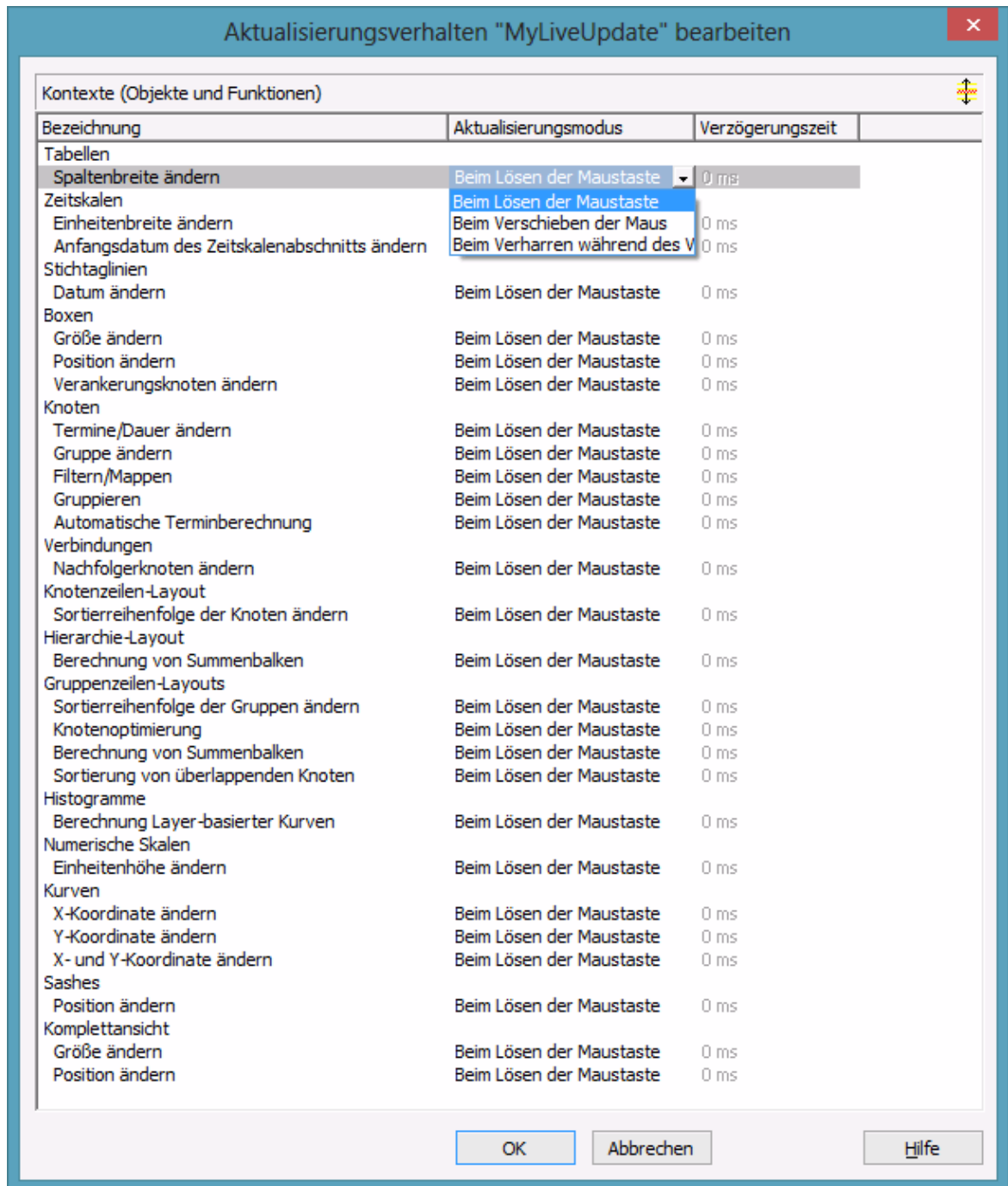
Live-Update in VARCHART XGantt einrichten

> Zur Designzeit

Die Einstellungen für das Live-Update werden zur Designzeit in den beiden Dialogen **Aktualisierungsverhalten verwalten** und **Aktualisierungsverhalten bearbeiten** vorgenommen. VARCHART XGantt verfügt bereits über die beiden Aktualisierungsverhalten **Standard** und **Live Update**, deren Einstellungen **nicht** vom Benutzer geändert werden können.

Es gibt jedoch die Möglichkeit, über die unten dargestellten Dialoge eigene Aktualisierungsverhalten zu erstellen, in denen dann individuelle Einstellungen vorgenommen werden können.





Hinweis: Bitte beachten Sie, dass die Zuweisung von individuellen Aktualisierungsverhalten für datengetriebene Objekte (Knoten, Links und Gruppen) **ausschließlich** über die API erfolgt.

> Zur Laufzeit

Die Einstellungen erfolgen über die Objekte:

- VcBox

152 Wichtige Konzepte: Live Update

- VcCurve
- VcDateLine
- VcGantt
- VcGroup
- VcLinks
- VcNode
- VcNumericScale
- VcTable
- VcTimeScale
- VcUpdateBehavior
- VcUpdateBehaviorCollection
- VcUpdateBehaviorContext
- VcWorldView

Weitere Informationen finden Sie in der API-Referenz dieses Handbuchs.

3.18 MultiState-Felder

Was sind MultiState-Felder?

Im Tabellenteil gibt es die Möglichkeit, mit grafischen Feldern und Zuordnungstabellen verschiedene Datenfeldinhalte als Grafiken darzustellen. MultiState-Felder stellen eine Erweiterung dieses Konzepts dar, bei denen jeder Klick auf eine Grafik zum Zustandswechsel des zugehörigen Datenfeldes führt. Es handelt sich also um eine bequeme Möglichkeit, Datenfelder, die eine endliche Anzahl von Zuständen annehmen können, zu editieren. Deshalb funktionieren MultiState-Felder auch nur, wenn das Modul **Data Editing** lizenziert ist.

> Wie sie funktionieren

Mit jedem Klick wird zyklisch in der Zuordnungstabelle nach der nächsten Grafik gesucht, welche sich von der aktuellen Grafik unterscheidet. Der entsprechende Wert (Schlüssel aus der Zuordnungstabelle) wird dann dem Datenfeld zugewiesen. Falls neben der Zuordnungstabelle auch noch eine weitere Grafik (Default-Grafik) angegeben ist, wird diese ebenfalls beim zyklischen Durchlauf durch die Grafiken berücksichtigt. Erscheint nach einem Klick die Default-Grafik, dann wird als Wert der leere String ins Datenfeld eingetragen. Ansonsten erscheint die Default-Grafik immer dann, wenn im Datenfeld ein Wert steht, der nicht in der Zuordnungstabelle als Schlüssel vorkommt.

Die einfachste denkbare Anwendung von Multistate-Feldern sind boolesche Datenfelder, bei denen die Werte **true** und **false** beispielsweise durch ein Checkbox-Symbol mit bzw. ohne gesetzten Häkchen visualisiert werden. Bei jedem Klick auf das gerade sichtbare Symbol wechselt die Ansicht zum anderen Symbol, und damit ändert sich auch der Wert des zugehörigen Datenfeldes von **true** nach **false** bzw. umgekehrt.

> Tipps für die Programmierung:

- Positionieren Sie in der Zuordnungstabelle Schlüssel, die auf ein und dieselbe Grafik verweisen, unmittelbar hintereinander. Nur so ist gewährleistet, dass dieselbe Grafik bei einem Zyklus durch die Zuordnungstabelle nur einmal angezeigt wird. Denn bei jedem Klick wird die nächste Grafik gewählt, die ungleich der aktuell gezeigten Grafik ist. So können Sie z.B. in der Map die Schlüssel **true**, **t** und **True** mit derselben Grafik verbinden. Wird diese Grafik angezeigt, dann erscheint beim

nächsten Klick in jedem Fall eine andere Grafik; es wird also nicht dreimal hintereinander dieselbe Grafik gezeigt.

- Aus dem gleichen Grund sollten Sie auch in der Zuordnungstabelle alle Schlüssel an den Anfang setzen, die auf dieselbe Grafik verweisen wie die Default-Grafik.
- Wenn eine Grafik mehrmals hintereinander in der Zuordnungstabelle vorkommt, dann wird beim Klicken als Wert in das Datenfeld immer der erste Schlüssel gesetzt. Stehen z.B. **true**, **t** und **True** hintereinander in der Map (mit gleicher Grafik), dann wird immer **true** und nicht **t** oder **True** als Wert ins Datenfeld eingetragen.
- MultiState-Felder ändern ihren Zustand nur, wenn das Editieren zugelassen ist (s. entsprechende VcGantt-Eigenschaften **InPlaceEditingOnGroupsInDiagramEnabled**, **InPlaceEditingOnGroupsInTableEnabled**, **InPlaceEditingOnNodesInDiagramEnabled**, **InPlaceEditingOnNodesInTableEnabled**).
- Damit die Grafiken bei verschieden hohen Tabellenzeilen nicht unterschiedlich groß dargestellt werden, sollte man die Höhe des Grafikfeldes auf einen festen Wert ungleich 0 mm setzen (s. Dialog **Tabellenformat bearbeiten** auf den Eigenschaftenseiten von XGantt).

Für weitere Informationen bzgl. der Grafikdateien und Zuordnungstabellen sei auf die Kapitel **Der Dialog "Tabellenformat bearbeiten"** und **Zuordnungstabellen** im Benutzerhandbuch sowie auf die Beschreibung der VcGantt-Eigenschaft **FilePath** im Referenzhandbuch verwiesen.

3.19 OLE Drag & Drop

Das OLE Drag & Drop-Verhalten des VARCHART ActiveX ist kompatibel zu dem in Visual Basic üblichen, d. h. die Methoden, Eigenschaften und Ereignisse tragen dieselben Namen und haben dieselbe Bedeutung wie bei den Standardobjekten aus Visual Basic.

Mit dem OLE Drag & Drop-Modus können Vorgänge und ganze Teildiagramme verschoben werden. Dieser Modus wird entweder manuell über die VcGantt-Methode **OLEDrag** oder automatisch gestartet.

> OLE-Dragmodus

Mit dem OLE-Dragmodus können Sie festlegen, ob das Ziehen eines Knotens über die Grenze des VARCHART-ActiveX-Steuerelements hinaus erlaubt sein soll. Mögliche Alternativen sind:

- **Manuell:** Bei diesem Modus müssen Sie die Methode **OLEDrag** aufrufen, um das Ziehen eines Knotens zu starten.
- **Automatisch:** Das Ziehen eines Knotens über die Grenzen des VARCHART-ActiveX-Steuerelements wird automatisch gestartet.

Beim Start des Vorgangs füllt die Quellkomponente das **DataObject** mit den Daten des gezogenen Knotens und setzt den Effekt-Parameter, um damit das **OLEStartDrag**-Ereignis sowie andere quellenseitige OLE Drag & Drop-Ereignisse auszulösen. Dies gibt Ihnen die Kontrolle über die Drag&Drop-Operation und erlaubt Ihnen einzugreifen, z. B. um andere Datenformate hinzuzufügen.

VARCHART ActiveX verpackt die Daten in die Zwischenablage-Formate CF_TEXT (Visual Basic: vbCFText) und CF_UNICODETEXT (bei Windows NT 4.0/2000/XP; Visual Basic: 13) die mühelos gelesen werden können. Das Datenformat ist identisch mit dem verwendeten CSV-Dateiablageformat.

Während des Ziehens kann der Benutzer mit Hilfe der Strg-Taste festlegen, ob das Objekt verschoben oder kopiert werden soll.

> OLE-Dropmodus

Mit dem OLE-Dropmodus können Sie festlegen, ob ein Knoten aus einer anderen VARCHART-ActiveX-Komponente in die aktuelle Komponente herein gezogen werden darf.

Mögliche Alternativen sind:

- **Kein:** Knoten aus einer anderen VARCHART-ActiveX-Komponente können nicht in die aktuelle Komponente hineingezogen werden.
- **Manuell:** Sie erhalten beim Dropping das Ereignis **OLEDragDrop**, so dass Sie die übertragenen Daten selbst weiterverarbeiten können, um z. B. einen Knoten zu erzeugen oder eine Datei einzulesen. Wenn Quell- und Zielkomponente identisch sind, erhalten Sie wie bei abgeschaltetem OLE Drag&Drop eins der Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**.
- **Automatisch:** Der Dropping-Vorgang wird von der Komponente selbst verarbeitet, d. h. es wird, falls möglich, ein Knoten an entsprechender Mausposition erzeugt.

> Auftretende Ereignisse

Falls Sie die Drag&Drop-Operation nicht automatisch durch die VARCHART-ActiveX-Komponenten durchführen lassen wollen, können Sie folgendermaßen in den Prozeß eingreifen:

Nach dem erfolgreichen Beginn einer OLE Drag & Drop-Operation wird das Ereignis **OLEStartDrag** auf dem Quellcontrol ausgelöst. Hiermit können Sie noch Änderungen am übergebenen **DataObject** vornehmen (d. h. andere Datenformate hinzufügen) und auch die erlaubten Drop-Effekte (Kopieren und/oder Verschieben) festlegen. Das **DataObject** ist sozusagen die programmtechnische Verpackung des grafisch gezogenen Knotens. Beim Verschieben des Objekts werden dann bei dem jeweiligen Control unter dem Mauscursor (Zielcontrol) **OLEDragOver**-Ereignisse ausgelöst. Hier kann der jeweils an der aktuellen Mausposition erlaubte Drop-Effekt auf Kopieren, Verschieben oder Verboten gesetzt werden.

Nach jedem Auftreten des **OLEDragOver**-Ereignisses auf dem Zielcontrol wird ein **OLEGiveFeedback**-Ereignis auf dem Quellcontrol ausgelöst, wo der Mauscursor gesetzt werden kann. lässt der Benutzer das gezogene Objekt fallen, dann wird auf dem Zielcontrol das **OLEDragDrop**-Ereignis ausgelöst, falls dort der **OLEDropMode** nicht auf automatisch gesetzt und die Ziel- nicht gleich der Quellkomponente ist. Ist er dagegen auf manuell gesetzt, müssen Sie für ein Ergebnis entsprechend des übergebenen Drop-Effekts sorgen. Nach Abschluß dieses Vorgangs wird auf dem Quellcontrol das **OLECompleteDrag**-Ereignis ausgelöst. Hier sollten Sie den Mauscursor wieder zurücksetzen, wenn Sie ihn im **OLEGiveFeedback**-Ereignis selbst gesetzt haben.

Bitte beachten Sie, dass Quell- und Zielcontrol identisch sein können und dass Quell- oder Zielcontrol natürlich auch andere Komponenten als VARCHART-ActiveX-Komponenten sein können, die evtl. sogar außerhalb

Ihrer Applikation liegen. Wenn Sie sichergehen wollen, dass Quell- und Zielcontrol nur festgelegte Controls Ihrer Applikation sind, dann können Sie beim **OLEStartDrag**-Ereignis ein eigenes Format mittels der **DataObject**-Methode **SetData** hinzufügen, das Sie über den Windows-API-Befehl **RegisterClipboardFormat** registriert haben, und dieses Format bei den **OLEDragOver**- und **OLEDragDrop**-Ereignissen auf dem Zielcontrol auf Vorhandensein mittels der **DataObject**-Methode **GetFormat** prüfen.

Wenn Sie Ihre Daten in mehreren Datenformaten gleichzeitig anbieten wollen, aber die Daten nicht gleich in allen Formaten in das **DataObject** übergeben wollen (weil die Daten zu aufwendig zu formulieren sind oder eine zu große Datenmenge darstellen würden), dann können Sie bei **SetData** das Schlüsselwort **Empty** für die Daten angeben:

dataObject.SetData Empty, myClipFormat

In der Zielanwendung wird eine Anfrage auf Vorhandensein des Formats mittels **dataObject.GetFormat** dann mit **True** beantwortet, und bei einem darauffolgenden **DataObject.GetData** in einem solchen Format wird auf der Quellseite das Ereignis **OLESetData** ausgelöst, in dem man die Daten in dem gewünschten Format nachreichen kann.

Wenn man Dateinamen aus einer Anwendung heraus- oder in sie hineinziehen will, dann ist das beim **DataObject** mit der Methode **Files** erreichbare Unterobjekt **DataObjectFiles** interessant. Für das Herausziehen von Dateinamen muss man beim Ereignis **OLEStartDrag** in der eigenen Anwendung zuerst das Format **vbCFFiles** (bzw. **CF_HDROP**) ohne Inhalt ablegen, also per **dataObject.SetData Empty, vbCFFiles**. Dann kann man Dateinamen mit der Methode **DataObject.Files.Add** hinzufügen. Beim Hereinziehen von Dateinamen (z.B. aus dem Windows Explorer) fragt man umgekehrt im Ereignis **OLEDragDrop** auf Vorhandensein des Formats **vbCFFiles** mit der Methode **DataObject.GetFormat** und liest die Dateinamen mittels **DataObject.Files(i)** o.ä. aus.

3.20 Ressourcenplanung

Der neue ResourceScheduler2 stellt eine wesentliche Erweiterung des ResourceScheduler1 aus Version 3.1 dar. Es werden die für eine Ressourcenplanung benötigten Objektarten nun in eigenen Datentabellen erwartet, was erst mit VARCHART XGantt 4.0 möglich wird. Im ResourceScheduler1 waren dagegen Aufgaben, Operationen, Zuweisungen und Ressourcen implizit in der Maindata-Tabelle zu definieren.

Zur Verfügung stehen die folgenden Objektarten, die jeweils in einer eigenen Datentabelle erwartet werden (bei Ressourcen können es bis zu 25 sein):

- **Aufgaben** (Tasks): Diese Objekte bilden eine Gruppierung von Operationen (s.u.) und halten elementare Eigenschaften wie das Freigabe- und das Fälligkeitsdatum sowie Priorität und Aufgabenmenge
- **Operationen**: Diese Objekte sind über Zuweisungen (siehe unten) auf Ressourcen (siehe unten) einplanbar und erhalten durch die Ressourcenplanung als Ergebnis Start- und Enddatum des Ausführungszeitraums. Operationen besitzen eine definierte Reihenfolge in der zugehörigen Aufgabe und können als bereits begonnen gekennzeichnet werden. Außerdem ist es möglich, mehrere einander ausschließende Operationsfolgen (sogenannte Routen) als Alternativen vorzugeben. Alle Operationen der von der Ressourcenplanung gewählten Route werden gemeinsam eingeplant.
- **Ressourcen**: Diese Objekte besitzen als Hauptmerkmal eine Kapazitätskurve und nach der Ausführung einer Ressourcenplanung auch eine Belegungskurve. Außerdem können Ressourcen für eine auf ihr eingeplante Operation zeitdauerbestimmend sein, wovon augenblicklich einer Operation immer eine zugewiesen sein muss, um die Operation einplanen zu können. Daneben können einer Operation beliebig viele weitere Arbeits- und Material-Ressourcen zugewiesen sein. Weiteres Wesensmerkmal einer zeitdauerbestimmenden Ressource ist die Fähigkeit zur ein- oder mehrstufigen Gruppierung, wobei eine solche Ressource auch mehreren Gruppen gleichzeitig angehören kann.
- **Zuweisungen** (Assignments): Diese Objekte sind die eigentliche Verknüpfung zwischen Operationen und Ressourcen, wobei hier ein Übersetzungsfaktor für die Aufgabenmenge angegeben werden kann. Im Falle von zeitdauerbestimmenden Ressourcengruppen werden Zuweisungen von der Ressourcenplanung entsprechend als solche gekennzeichnet und konkrete Ersatzzuweisungen zu der ausgewählten Einzelressource erzeugt, so dass die Einplanung nachvollziehbar und im XGantt darstellbar wird.

- **Verbindungen (Links):** Diese Objekte beschreiben die Reihenfolge von Aufgaben, das heißt, vorangehende Aufgaben müssen beendet sein, bevor nachfolgende Aufgaben begonnen werden dürfen.

Überblick über die Objekte und die zugehörigen Eigenschaften

Aufgabentabelle	
TaskDataTableName	Name der Aufgabentabelle
TaskDueDateFieldIndex	Datum, bis zu dem eine Aufgabe beendet sein muss
TaskPlanningStrategyFieldIndex	Planungsstrategie: ASAP oder JIT für einzelne Aufträge
TaskPriorityFieldIndex	Priorität, die die Gewichtung angibt, mit der die Aufgabe in die Planung mit einbezogen wird
TaskQuantityFieldIndex	Gewünschte Stückzahl, die durch den Arbeitsauftrag fertig gestellt werden soll.
TaskReleaseDateFieldIndex	Datum, ab dem ein Auftrag eingeplant werden darf.
TaskResultEndDateFieldIndex	Ermitteltes Enddatum der Aufgabe
TaskResultPostEndDateFieldIndex	Ermitteltes Enddatum der Nachlaufzeit
TaskResultPreparationsStartDateFieldIndex	Ermitteltes Startdatum der Vorlaufzeit
TaskResultProcessingStepFieldIndex	Ermittelte Abfolgenummer der Aufgabe
TaskResultProcessingTimeFieldIndex	Ermittelte Planungszeit der Aufgabe
TaskResultRouteFieldIndex	Ermittelte Route, bestehend aus verfügbaren Ressourcen, über die die Aufgabe abgearbeitet wird
TaskResultStartDateFieldIndex	Berechnetes Anfangsdatum der Aufgabe

Operationen-Tabelle	
OperationDataTableName	Name der Operationentabelle
OperationMaximumInterruptionTimeFieldIndex	Maximale Unterbrechungszeit der Operation (mit Belegung der Resource)
OperationLoadPerItemFieldIndex	Belegung der Resource pro Stück
OperationOverlapQuantityFieldIndex	Überlappung mit anderen Operationen
OperationPostLoadFieldIndex	Nachlaufbelegung der Operation
OperationPreparationLoadFieldIndex	Vorbereitungsbelegung der Operation

Operationen-Tabelle	
OperationResultPostEndDateFieldIndex	Ermitteltes Enddatum der Nachlaufphase
OperationResultProcessingTimeFieldIndex	Ermittelte Verarbeitungszeit der Operation
OperationResultPreparationStartDateFieldIndex	Ermitteltes Startdatum der Vorbereitungsphase
OperationResultSelectedTimingResourceIDFieldIndex	Ermittelte ID der Zeitdauer bestimmenden Ressource
OperationResultStatusFieldIndex	Fehler- oder Warnstatus
OperationRouteFieldIndex	Route, der die Operation angehört
OperationSequenceNumberFieldIndex	Reihenfolgewart der Operation innerhalb einer Route
OperationStartLockDateFieldIndex	Fixdatum
OperationTaskIDFieldIndex	Aufgabe, der die Operation angehört
OperationWorkInProgressFieldIndex	Fertigstellungsgrad der Operation

Ressourcen-Tabelle	
ResourceCalendarNameFieldIndex	Name des Ressourcen-Kalenders
ResourceCapacityType	Begrenzte oder unbegrenzte Kapazitäten für alle Ressourcen
ResourceCapacityTypeFieldIndex	Begrenzte oder unbegrenzte Kapazitäten für Einzelressourcen
ResourceConstraintTypeFieldIndex	Bedingung für Arbeits- und Materialressourcen
ResourceDataTableName	Name der Ressourcentabelle
ResourceEfficiencyFieldIndex	Effizienz in %
ResourceGroupDataTableName	Name der Tabelle für Gruppenressourcen
ResourceGroupIDFieldIndex	Gruppenzugehörigkeit der Ressource
ResourceNameFieldIndex	Name der Ressource
ResourceResultLoadCurveNamePrefix	Kurve, in die die ermittelte Auslastung von Arbeits- und Zeitressourcen eingetragen werden soll
ResourceResultStockCurveNamePrefix	Kurve, in die der ermittelte Lagerbestand von Materialressourcen eingetragen werden soll
ResourceSelectionStrategy	Auswahlstrategie für Ressourcen

Ressourcen-Tabelle	
ResourceSoftConstraintStartDateFieldIndex	Datum für den Statuswechsel einer Ressource von "hart" nach "weich"
ResourceType	Ressourcentyp
ResultProcessingStepCount	Ermittelte Anzahl von Aufgaben

Tabelle für Zuweisungen	
AssignmentDataTableName	Name der Zuweisungstabelle
AssignmentIsResultFieldIndex	Wurde der Datensatz als Ergebnis der Zeitplanung erzeugt?
AssignmentIsVisibleFieldIndex	Soll die Darstellung im Diagramm erfolgen?
AssignmentLoadOrConsumptionFieldIndex	Wert pro Stück
AssignmentMaximumLoadFieldIndex	Maximaler Auslastungsgrenzwert
AssignmentMinimumLoadFieldIndex	Minimaler Auslastungsgrenzwert
AssignmentOperationIDFieldIndex	Zugewiesene Operation
AssignmentResourceSelectionStrategyrFieldIndex	ASAP oder JIT für eine Ressource
AssignmentResourceIDFieldIndex	Zugewiesene Ressource

Tabelle für Verbindungen	
LinkDataTableName	Name der Tabelle
LinkDurationFieldIndex	Minimaler zeitlicher Abstand
LinkPredecessorTaskIDFieldIndex	Vorgängeraufgabe der Verbindung
LinkSuccessorTaskIDFieldIndex	Nachfolgeraufgabe der Verbindung

Allgemeine Eigenschaften	
BaseTimeUnit	Mögliche eigene Zeiteinheit für die Ressourcenplanung
BaseTimeUnitsPerStep	Grob- oder Feinberechnung?
DataRecordEventsEnabled	Werden DataRecord-Ereignisse zugelassen?
DefaultOperationMaximumInterruptionTime	Maximale Zeitdauer einer einmaligen Unterbrechung für Operationen
DefaultResourceCalendarName	Zuweisung eines Default-Kalenders für die Zeitplanung
FullUsageOfPlanningUnitsEnabled	Nutzung von Restkapazitäten von Ressourcen

162 Wichtige Konzepte: Ressourcenplanung

Allgemeine Eigenschaften	
PlanningEndDate	Ende des Planungszeitraums
PlanningStartDate	Anfang des Planungszeitraums
PlanningStrategy	Planungsstrategie: ASAP oder JIT für alle Aufträge
Process	Ressourcenplanung starten
ToleranceTimeOnASAPDueDates	Zeittoleranz beim Fälligkeitsdatum
ToleranceTimeOnJITReleaseDates	Zeittoleranz beim Freigabedatum
ToleranceTimeOnStartLockDates	Zeittoleranz für festes Startdatum
WorkInProcessType	Einheit des Fertigstellungsgrades
WritingDebugFilesEnabled	Verfügbarkeit von Debug-Dateien

Nachdem Sie die Eigenschaften der Tabellen gesetzt haben, können Sie die Ressourcenplanung über die Methode **Process** starten.

3.21 Schreiben von PDF-Dateien

Das Schreiben von PDF-Dateien ist nur möglich, wenn ein geeigneter PDF-Druckertreiber installiert ist. Die kostenlosen und kommerziell verfügbaren Treiber unterscheiden sich hinsichtlich der Funktionalität und Qualität der erzeugten PDF-Dateien.

Für die Ansteuerung der Treiber gibt es keinen einheitlichen Standard, sodass jeder Druckertreiber individuell konfiguriert werden muss. So ist beispielsweise bei vielen PDF-Druckertreibern der Zielpfad für die Ausgabedatei fest vorgegeben und kann nur durch Eingriffe in die Windows-Registry, durch Editieren von INI-Dateien oder durch Verwenden treiberspezifischer Funktions-APIs oder COM-Objekte geändert werden.

Ein PDF-Druckertreiber muss die folgenden Anforderungen hinsichtlich der Ansteuerung und Druckqualität erfüllen, damit er sich für den Einsatz eignet:

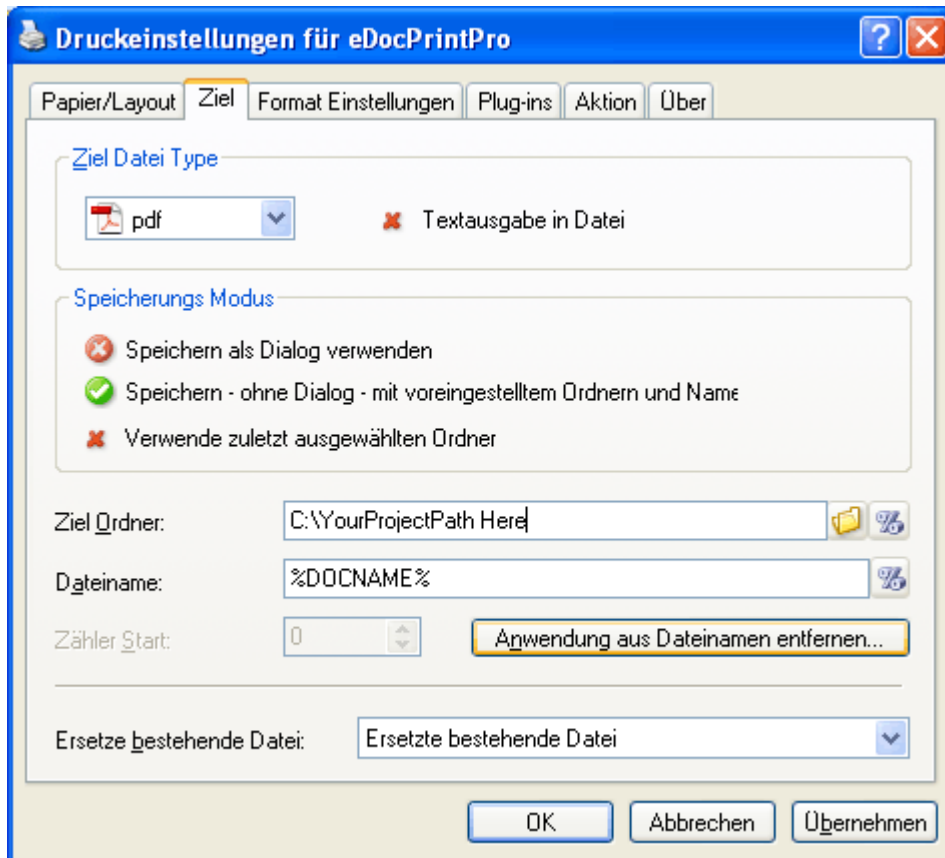
- Je nach Design der Anwendung kann es notwendig sein, den Treiber über die API so einzustellen, dass zur Laufzeit keine Dialoge und Messageboxen erscheinen. Dazu gehören insbesondere Dialoge zur Festlegung von Dateinamen und Pfaden.
- Soll das Setzen von Dateinamen und Pfad erst zur Laufzeit erfolgen, und ist dies nur über das Ändern von Windows-Registry-Einträgen möglich, dann müssen die Rechte des Benutzerkontos dies auch erlauben.
- Zur korrekten Ausgabe von Texten ist Unicode-Unterstützung erforderlich.
- Die Wiedergabe von Füllmustern muss in ausreichender Qualität erfolgen. Dabei ist zu beachten, dass Transparenzen mit Ausnahme bei Bitmaps grundsätzlich nicht dargestellt werden können. Dort können jedoch unerwünschte Artefakte auftreten.
- Der Treiber muss die Ausgabe vertikaler Texte unterstützen, sonst kann die vertikale Beschriftung von Datumslinien in VARCHART XGantt nicht genutzt werden.

Die vorgenannten Anforderungen erfüllen z.B. der in der **Adobe Acrobat Suite** ab Version 6 enthaltene Druckertreiber [www.adobe.com] sowie der kostenlose Treiber **eDocPrintPro** [www.pdfprinter.at].

Im Folgenden werden die Schritte skizziert, die zur Ansteuerung der Druckertreiber notwendig sind. Dies wird exemplarisch am Treiber **eDocPrintPro** verdeutlicht:

- In den **Druckeinstellungen** (erreichbar über die Einstellungen des Treibers in der Systemsteuerung oder über einen eigenen Eintrag des

Treibers unter Start/Programme oder über den normalen Druckdialog in einer Anwendung) kann gegebenenfalls spezifiziert werden, dass die PDF-Erzeugung ohne Dialog abläuft, und dass der Name der Zielfeile z.B. über den Dokumentennamen bestimmt wird. Bei **eDocPrintPro** sehen die notwendigen Einstellungen dann wie folgt aus:



- Im Programm wird dann das VcPrinter-Objekt von VARCHART XGantt folgendermaßen bestückt:

Code-Beispiel

```
VcGantt1.Printer.PrinterName = "eDocPrintPro"
VcGantt1.Printer.DocumentName = "abc.pdf"
VcGantt1.PrintEx
```

Ganz wenige Druckertreiber erfordern eine andere Code-Sequenz:

Code-Beispiel

```
VcGantt1.Printer.PrinterName = "Win2PDF"
VcGantt1.PrintToFile "abc.pdf"
```

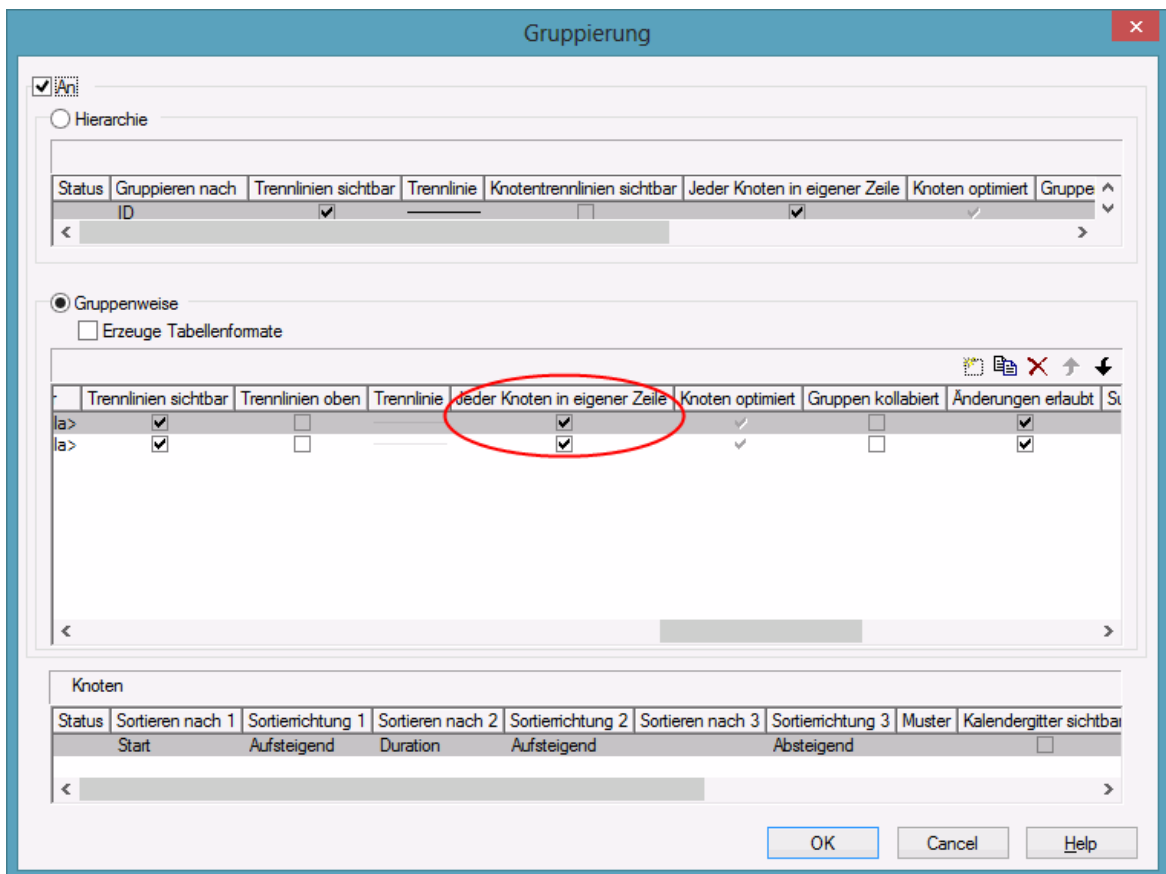
Für weitere Fragen zur Konfiguration und Benutzung von **eDocPrintPro** bitten wir Sie, sich mit dem Hersteller in Verbindung zu setzen.

3.22 Sortierung

Vorgänge werden in Anwendungen meist nach bestimmten Kriterien sortiert. Eine Sortierung kann nur bei Knoten stattfinden, die nicht in eine Hierarchie eingebunden sind, d.h. also nur bei Basisknoten oder solchen, die zu einer Gruppe gehören. Daher finden Sie Sortieroptionen nur dort, wo Sie Gruppen- und Basisknoten beeinflussen können. Bei einer Sortierung muss unterschieden werden, ob die Knoten in jeweils einer eigenen Zeile angeordnet sind oder ob sich mehrere Knoten in einer Zeile befinden.

Anordnung: Jeder Knoten in eigener Zeile

Wenn Sie die Knoten in einzelnen Zeilen anordnen wollen, rufen Sie den Dialog **Gruppierung** auf, den Sie bei den Eigenschaftenseiten auf der Seite **Objekte** unter **Gruppierung** finden:



Hier setzen Sie im mittleren Fenster die Einstellung **Jeder Knoten in eigener Zeile**. Sie können diese Einstellung alternativ über die Programmierschnittstelle mit der Eigenschaft **VcGroupLevelLayout.AllNodesInOneRow** setzen.

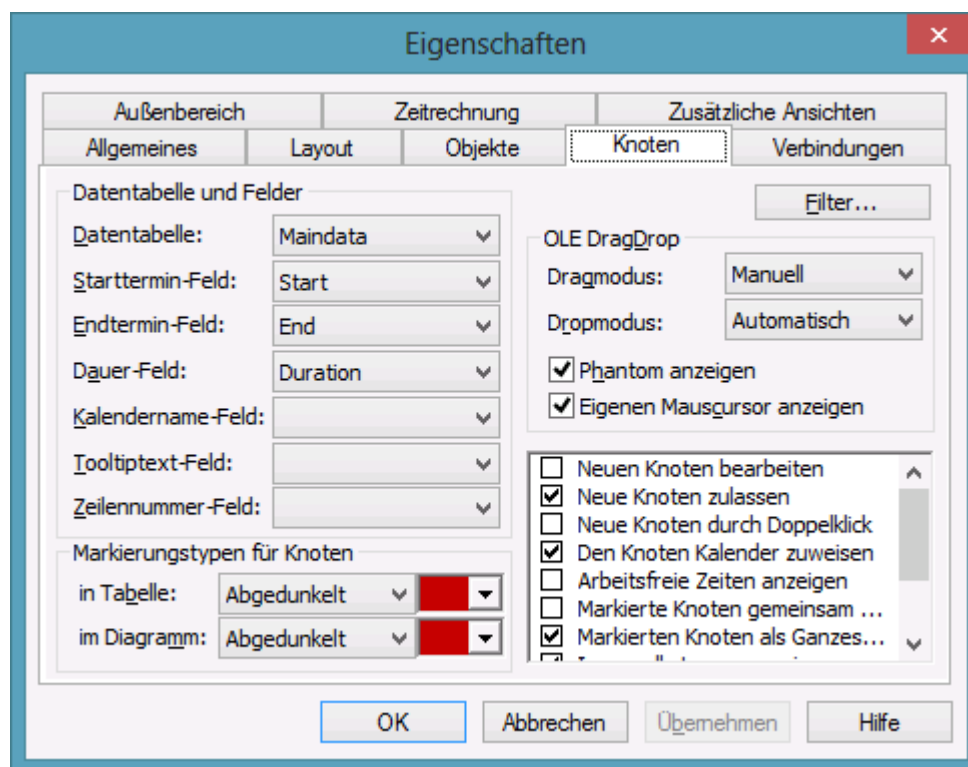
In dem darunterliegenden Fenster **Knoten** können Sie drei Datenfelder festlegen, nach denen die Vorgänge beim Öffnen des Diagramms sortiert werden sollen. Für jedes dieser Datenfelder können Sie außerdem bestimmen, ob aufsteigend oder absteigend sortiert werden soll.

Falls die Vorgänge gruppiert sind, gilt die Sortierung innerhalb jeder Gruppe.

Weiterhin stehen Ihnen folgende Möglichkeiten zur Verfügung, um das Aussehen der Knotenzeilen zu bestimmen:

- Auswahl eines **Musters**
- Anzeige, Position und Aussehen der **Trennlinien**
- über **Trennlinienschnittweite** können Sie festlegen, nach wie vielen Vorgängen jeweils eine Trennlinie gezogen werden soll. Falls die Vorgänge gruppiert sind, wird in jeder Gruppe neu zu zählen begonnen.

Weitere Optionen für die Sortierung finden Sie auf der Eigenschaftenseite **Knoten**:



- Sie können ein Datenfeld festlegen, in das die **Zeilennummer** jedes Vorgangs geschrieben werden soll. Das Zeilennummer-Feld wird immer erst aktualisiert, wenn die Daten mit der Methode **Save As** gesichert wurden.
- Über **Veränderung der Knotenreihenfolge über das Diagramm** bzw. **Veränderung der Knotenreihenfolge über die Tabelle** können Sie fest-

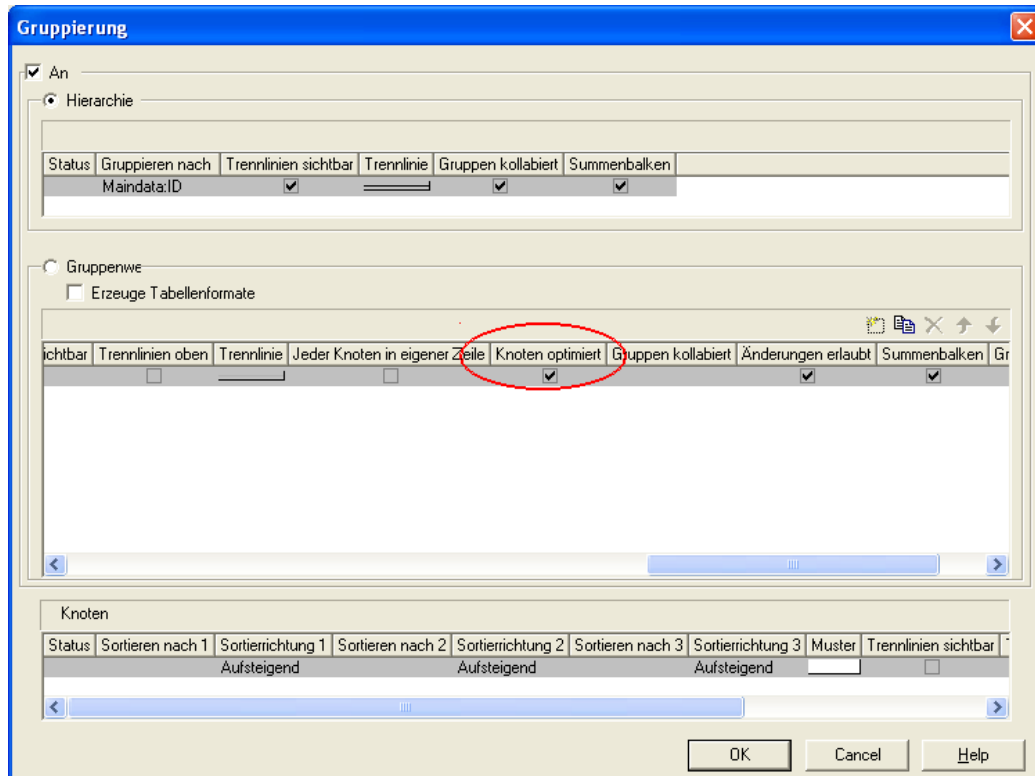
legen, ob dem Anwender Änderungen in der Vorgangsreihenfolge erlaubt sein sollen. Dann lassen sich Vorgänge per Drag & Drop in eine andere Zeile ziehen. Wird ein Vorgang in eine andere Gruppe verschoben, werden sein Gruppierfeld und seine Farbe aktualisiert. Wird der Vorgang durch mehrere Layer repräsentiert, muss zusätzlich die Umschalt-Taste gedrückt werden.

Hinweis: Beachten Sie bitte, dass die Einstellungen im Dialog **Gruppierung** und auf der Eigenschaftenseite **Knoten** nur für die Sortierung der Vorgänge beim Öffnen des Diagramms gelten. Wenn Sie die Vorgänge danach neu sortieren möchten, verwenden Sie dazu bitte die Methode **SortNodes**. Die Aktualisierung der Sortierung muss also über diesen Aufruf angestoßen werden.

Anordnung: Knoten einer ganzen Gruppe in einer Zeile

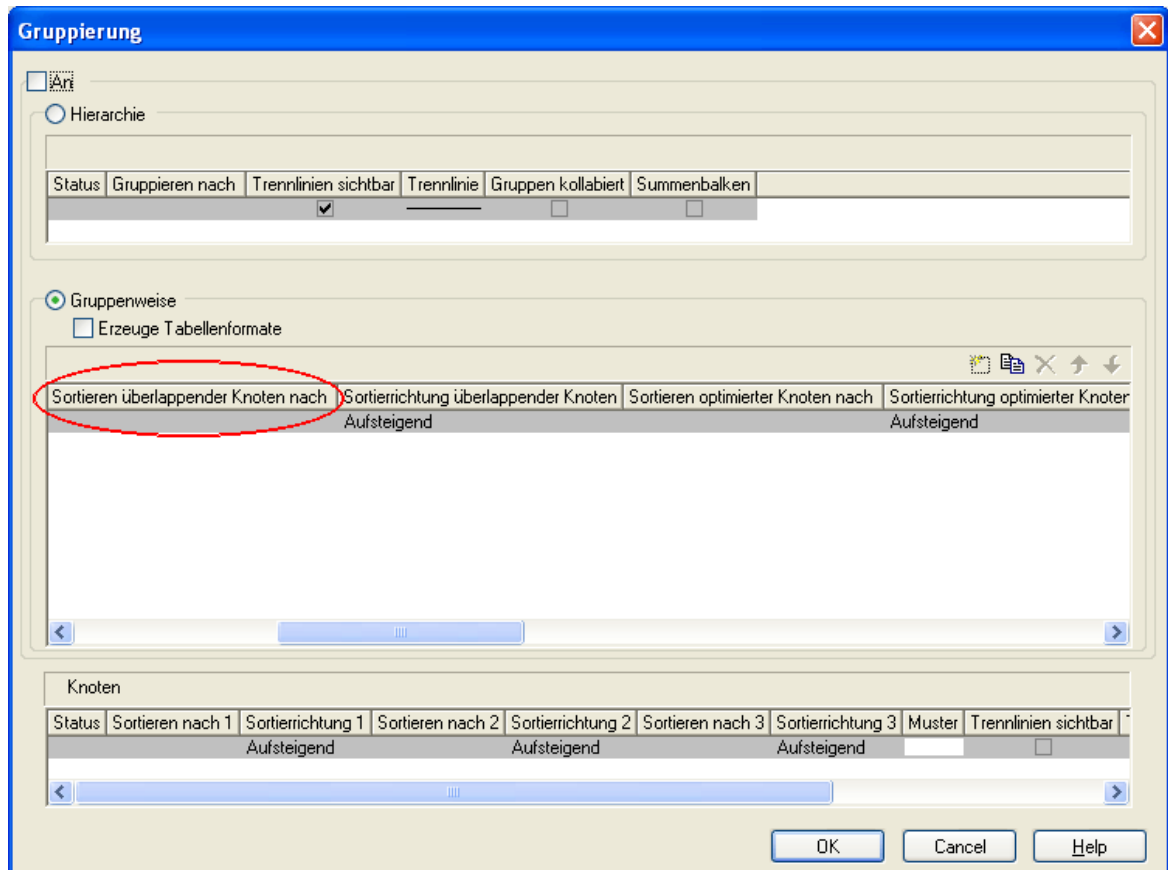
Wenn Sie mehrere Knoten (d.h. die Knoten einer Gruppe) in eine Zeile anordnen, können Sie auch für diese Knoten eine Sortierung einrichten (die einer Darstellungspriorität entspricht). Dabei ist zwischen der **überlappenden** und **optimierten** Darstellung zu unterscheiden, d.h. Knoten können sich entweder überlappen oder eine Überlappung durch Verbreiterung der Zeile vermeiden.

Mehrere Knoten in eine Zeile setzen Sie im Gruppierungsdialog im mittleren Fenster über das Feld **Knoten optimiert**; diese Einstellung wird standardmäßig angeschaltet, wenn Sie **Jeder Knoten in eigener Zeile** deaktivieren:



Sie können aber auch **Knoten optimiert** abschalten, dann erhalten Sie die Darstellung mit sich überlappenden Knoten. Sie können diese Einstellung alternativ über die API-Eigenschaft **VcGroup.NodesArrangedOptimized** setzen.

Die Darstellungspriorität können Sie über das Feld **Überlappende Knoten sortieren nach** im Dialog **Gruppierung** festlegen:



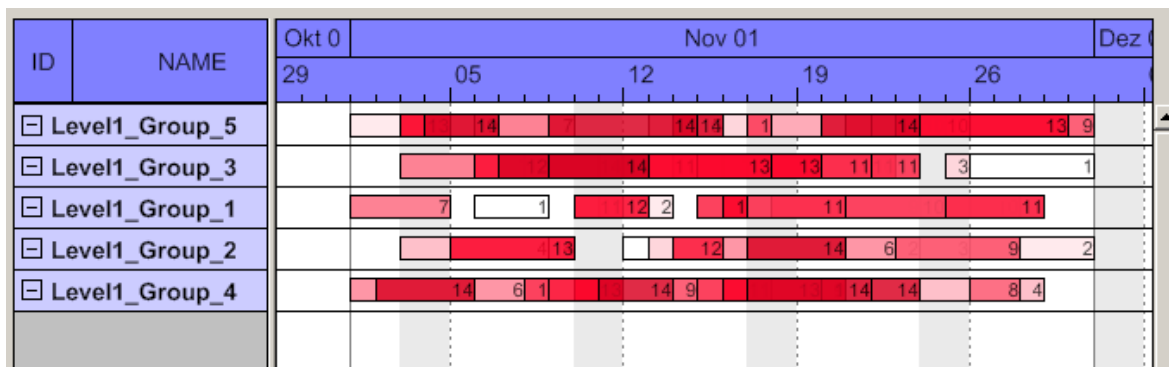
Analog zu den überlappenden Knoten können Sie über das Feld **Optimierte Knoten sortieren nach** auch optimierte Knoten sortieren.

Wenn Sie keine Darstellungspriorität setzen, werden die Knoten standardmäßig nach Anfangsdatum und Dauer abgebildet: die spätesten und kürzesten werden am weitesten oben liegend dargestellt. Sie können die Darstellungspriorität alternativ über die Eigenschaften der Programmierschnittstelle **VcLevelLayout.OverlaidNodesSortDataFieldIndex** bzw. **VcLevelLayout.OptimizedNodesSortDataFieldIndex** setzen.

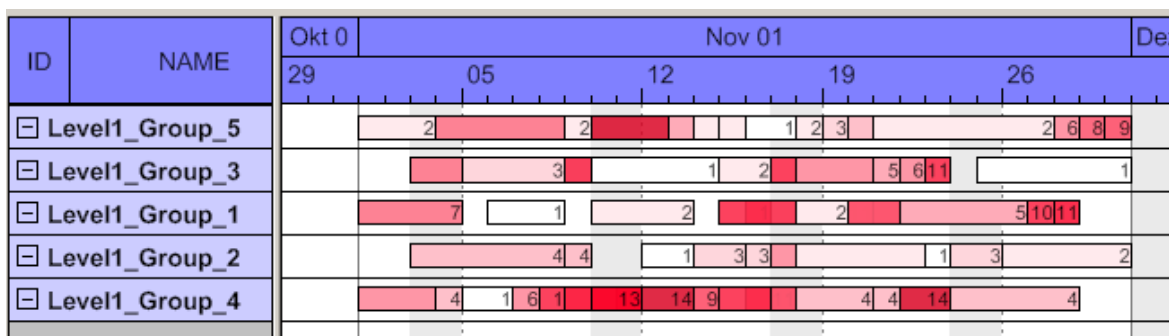
Die Aktualisierung der Sortierung erfolgt automatisch; d.h. Sie brauchen diese nicht durch einen eigenen Aufruf anzustoßen.

Außerdem haben Sie in im benachbarten Feld **Sortierrichtung überlappender Knoten** die Möglichkeit, die Reihenfolge aufsteigend oder absteigend zu wählen. Alternativ können Sie die Sortierrichtung über die Eigenschaften **OverlappingNodesSortOrder** bzw. **OptimizedNodesSortOrder** der Programmierschnittstelle setzen. Hier mögliche Ergebnisse:

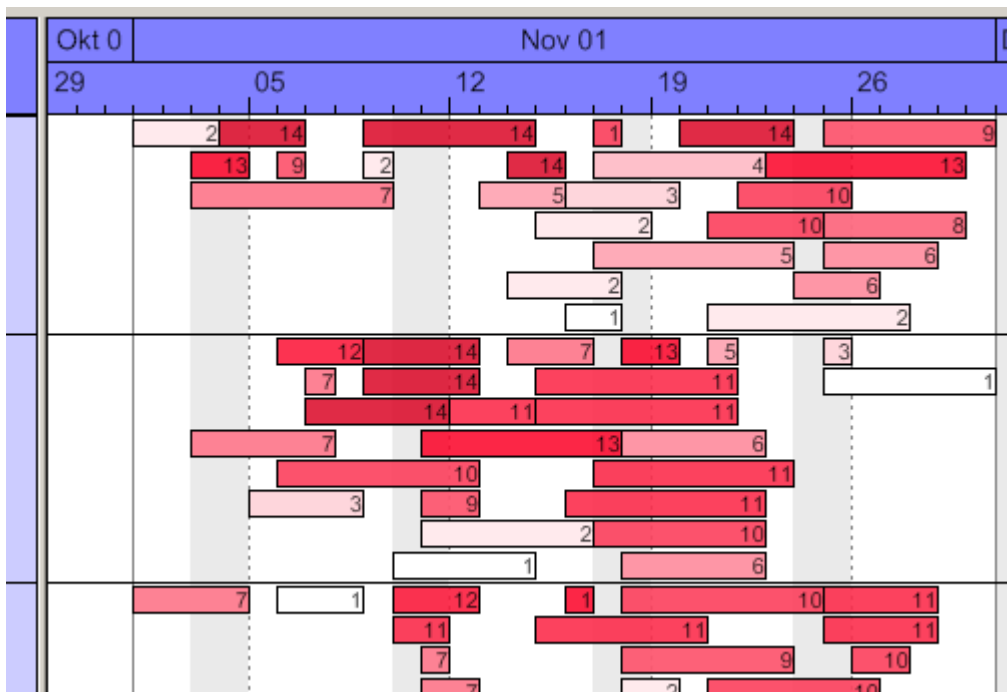
170 Wichtige Konzepte: Sortierung



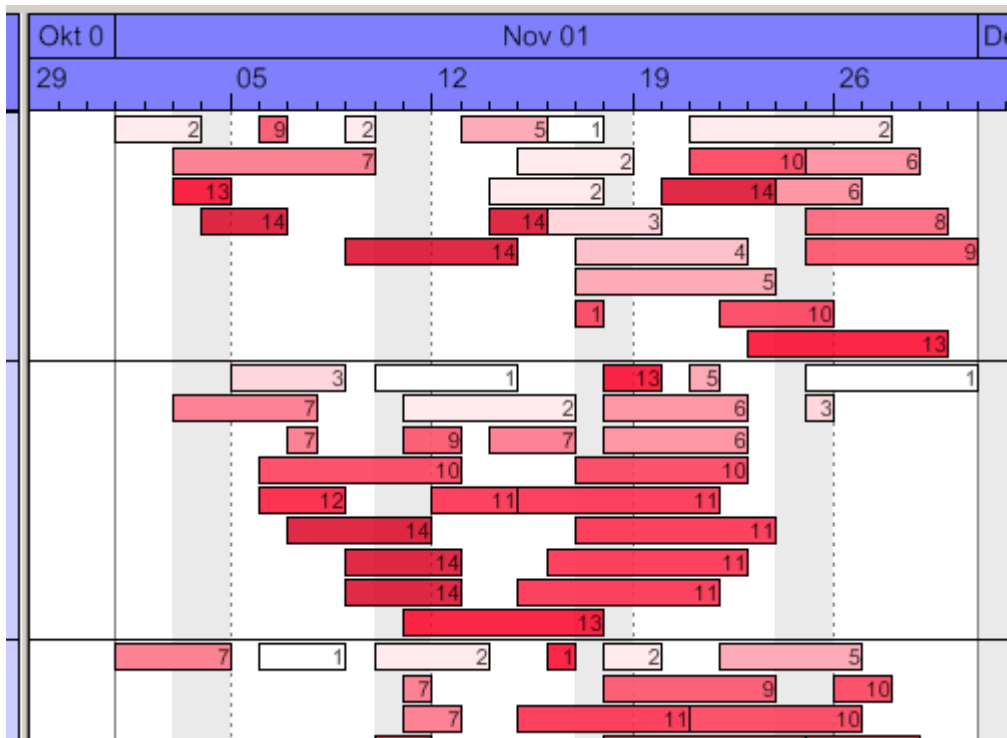
Überlappende Anordnung mit aufsteigender Darstellungspriorität von dunklen Knoten (dunkle Knoten im Vordergrund liegend)



Überlappende Anordnung mit absteigender Darstellungspriorität von dunklen Knoten (dunkle Knoten im Hintergrund liegend)



Optimierte Anordnung mit aufsteigender Darstellungspriorität von dunklen Knoten (dunkle Knoten im oberen Bereich der Zeile)



Optimierte Anordnung mit absteigender Darstellungspriorität von dunklen Knoten (dunkle Knoten im unteren Bereich der Zeile)

3.23 Sprachanpassung von Textausgaben

Sie können mit Hilfe des Ereignisses **OnSupplyTextEntry** die Texte aller zur Laufzeit erscheinenden Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen sowie Monats- und Tagesnamen bearbeiten, z. B. um sie in unterschiedliche Sprachen zu übersetzen.

Aktivieren Sie dazu das Kontrollkästchen **OnSupplyTextEntry-Ereignisse** auf der Eigenschaftenseite **Allgemeines**.

Oder setzen Sie dazu die Eigenschaft **EnableSupplyTextEntryEvent** auf den Wert **True**, um das Ereignis zu aktivieren.

Code-Beispiel

```
VcGantt1.EnableSupplyTextEntryEvent = True
```

Fangen Sie dann das Ereignis **OnSupplyTextEntryEvent** ab und legen Sie fest, welcher Text erscheinen soll.

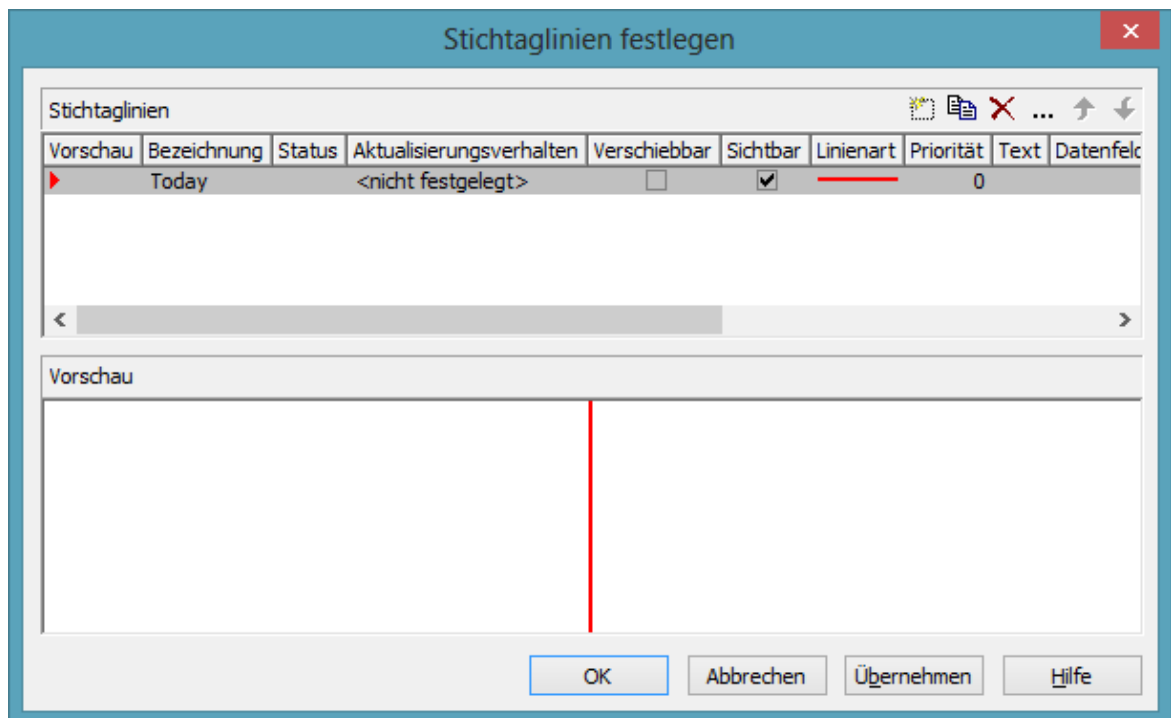
Code-Beispiel

```
Private Sub VcGantt1_OnSupplyTextEntry(ByVal controlIndex As _
                                         VcGanttLib.TextEntryIndexEnum, _
                                         TextEntry As String, _
                                         returnStatus As Variant)

    Select Case controlIndex
        Case vcTXERibCW
            TextEntry = "Kalenderwoche"
        Case vcTXERibDay0
            TextEntry = "MO"
        Case vcTXERibMon8
            TextEntry = "Sept."
        Case vcTXERibQuar2
            TextEntry = "2. Quart."
    End Select
End Sub
```

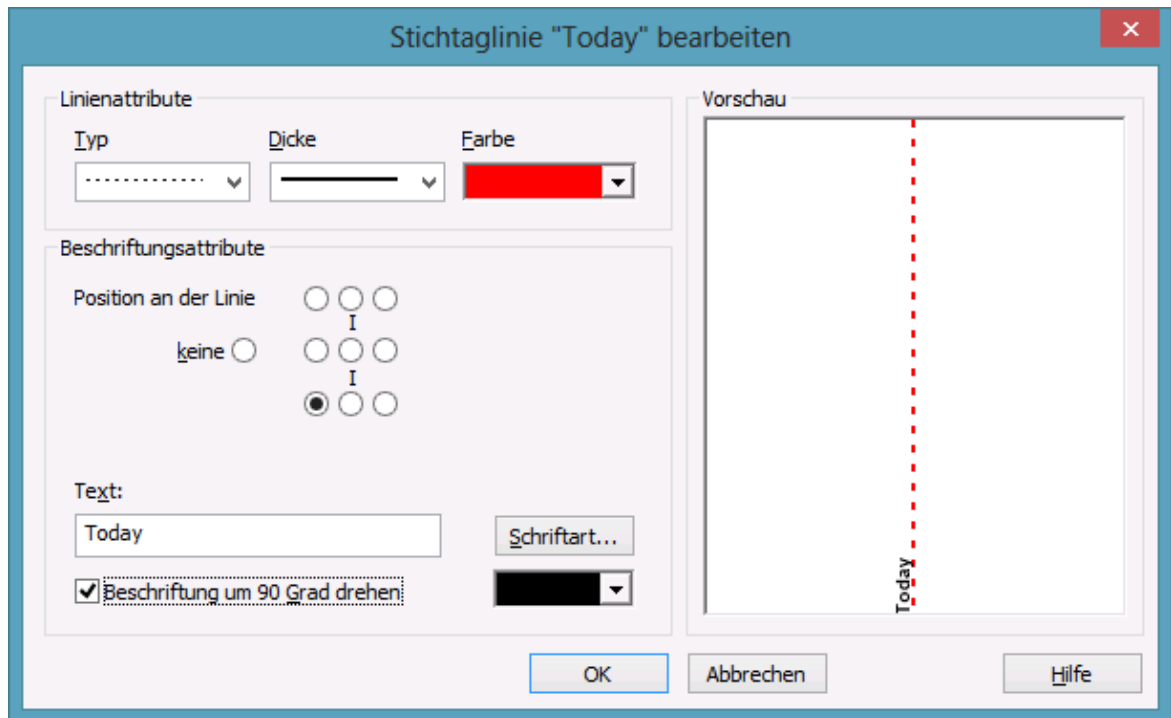
3.24 Stichtaglinien

Durch Stichtaglinien (vertikale Linien im Diagramm) können Sie bestimmte Daten besonders hervorheben. Die Eigenschaften der Stichtaglinien werden in den folgenden Dialogen bestimmt: die Beschriftung, das Datum, die Linienattribute, die Priorität (relativ zueinander und zu den Rastern und den Knoten) sowie ob sie verschiebbar und sichtbar sein sollen. Sie erreichen den ersten Dialog über die Eigenschaftenseite **Objekte**, den zweiten über die **Bearbeiten**-Schaltfläche:



Stichtaglinien festlegen

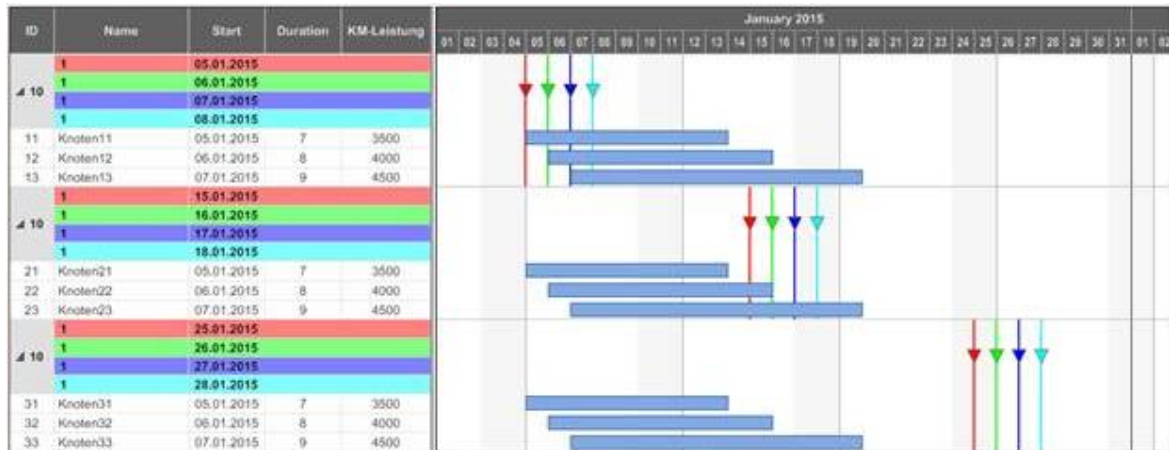
174 Wichtige Konzepte: Stichtaglinien



Stichtaglinie bearbeiten

Individuelle, datenbasierte Stichtaglinien

Stichtaglinien können auch alternativ zu dem fixen Datum auf ein Datum aus einem Datensatz eines Knotens oder einer Gruppe zugreifen. Es können dann automatisch zu jedem Datensatz eines Knotens oder einer Gruppe individuelle Stichtaglinien als grafische Kopien entstehen, d.h. es werden die Eigenschaften (Farbe etc.) aus der zugrunde liegenden Stichtaglinien der DateLineCollection außer Datum verwendet, wobei Datum und Position im Plan jedoch individuell sind. Solche Stichtaglinien werden unter Anwendung des **NodeLevelLayout** bzw. **GroupLevelLayout** nur innerhalb der Bänder der Knoten bzw. Gruppen gezeichnet (siehe Bild: je vier Stichtaglinien wurden individuell für drei Gruppen erzeugt und positioniert; vier Symbollayer des eingeschalteten Gruppenknotens benutzen dieselben Termine wie die Stichtaglinien).



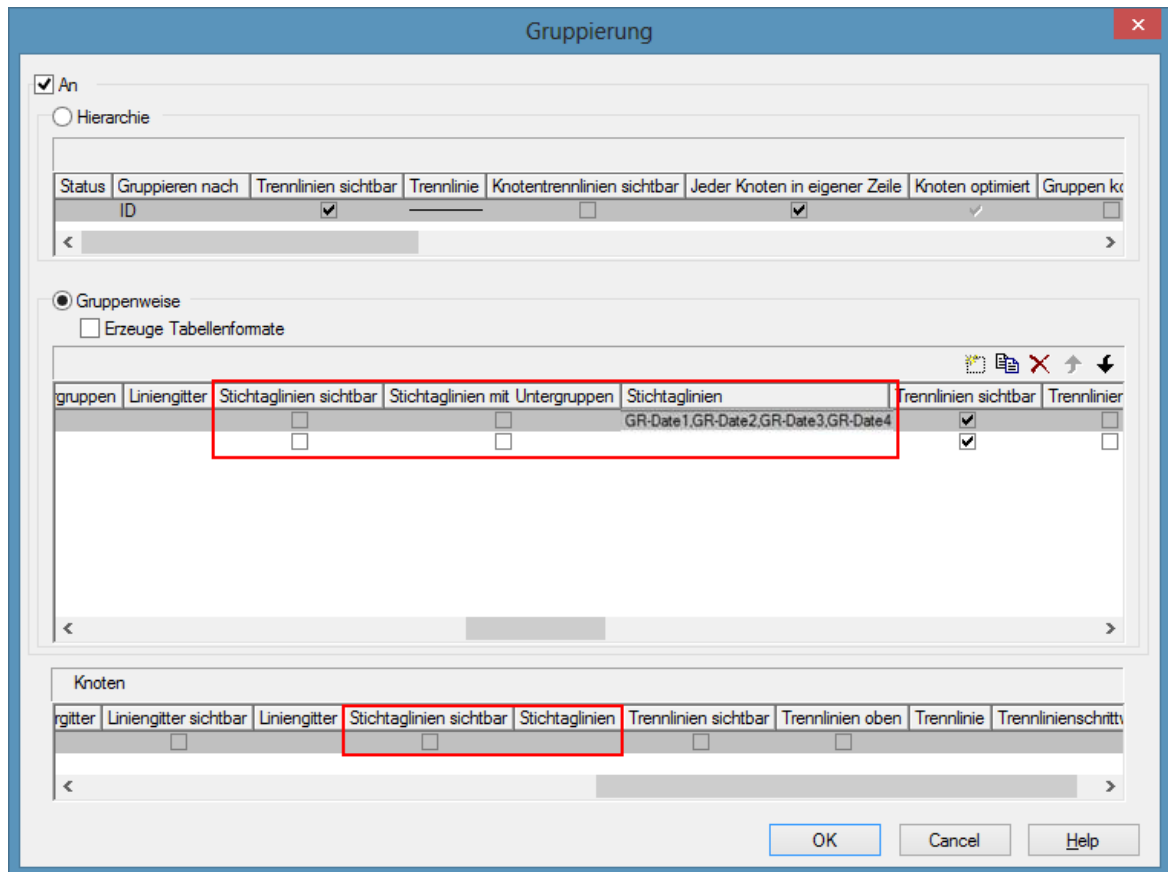
Um das zu erreichen, muss zunächst bei der Stichtaglinie ein Datenfeld angegeben werden, aus dem das Datum geholt werden soll.

status	Aktualisierungsverhalten	Verschiebbar	Sichtbar	Linienart	Priorität	Text	Datenfeld	Heute (dynamisch)
<nicht festgelegt>	☒	☒	☒	---	0			☒
<nicht festgelegt>	☐	☐	☐	---	0		Gruppen:Termin1	☐
<nicht festgelegt>	☒	☒	☒	---	0		Gruppen:Termin2	☐
<nicht festgelegt>	☒	☒	☒	---	0		Gruppen:Termin3	☐
<nicht festgelegt>	☒	☒	☒	---	0		Gruppen:Termin4	☐

Wichtig: Bei einem individuell festgelegten Datenfeld hat das Datum aus dem Datensatz Priorität vor dem fixen Datum (**VcDateLine.Date**). Wenn kein Datum ermittelt werden konnte, z.B. weil das angegebene Datenfeld im Datensatz leer ist, erscheint keine Stichtaglinie.

Nachdem ein Datenfeld festgelegt wurde, muss die Stichtaglinie noch mit einem Datensatz verknüpft werden. Dies geschieht durch entsprechende Angaben im Dialog **Gruppierung**:

176 Wichtige Konzepte: Stichtaglinien



An der API geschieht dies durch die entsprechenden Befehle:

VcGroupLevelLayout.ShowDateLines

VcGroupLevelLayout.DateLinesWithChildGroups

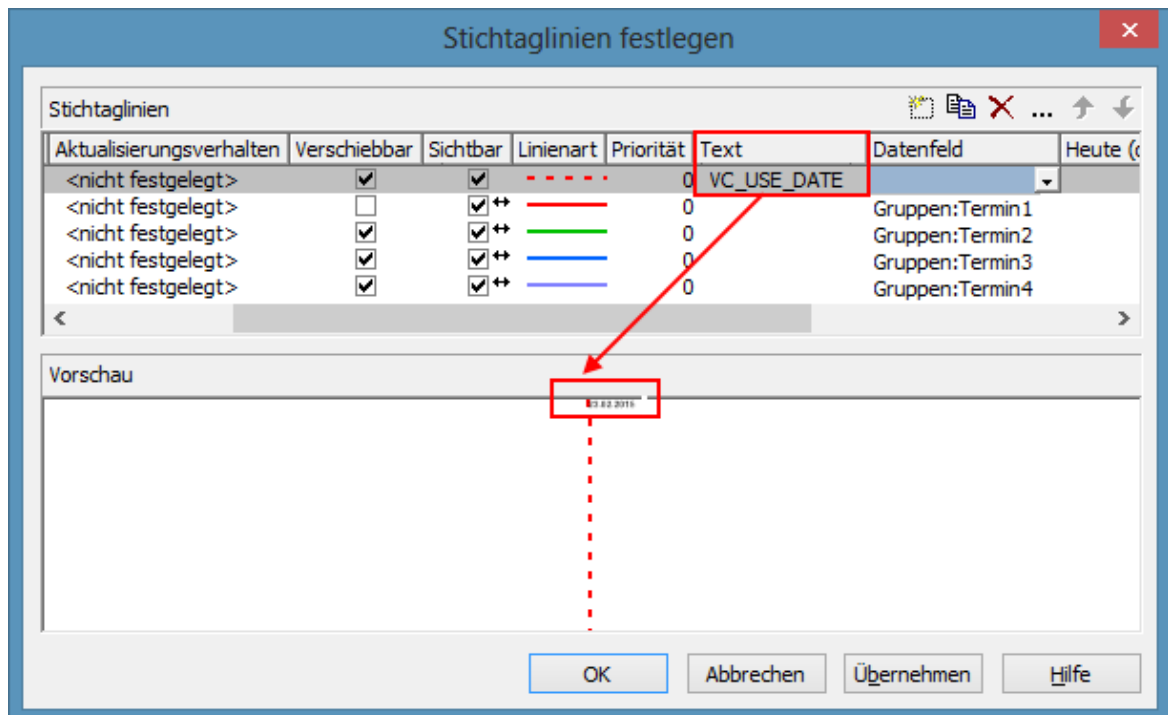
VcGroupLevelLayout.DateLineName

VcNodeLevelLayout::ShowDateLines

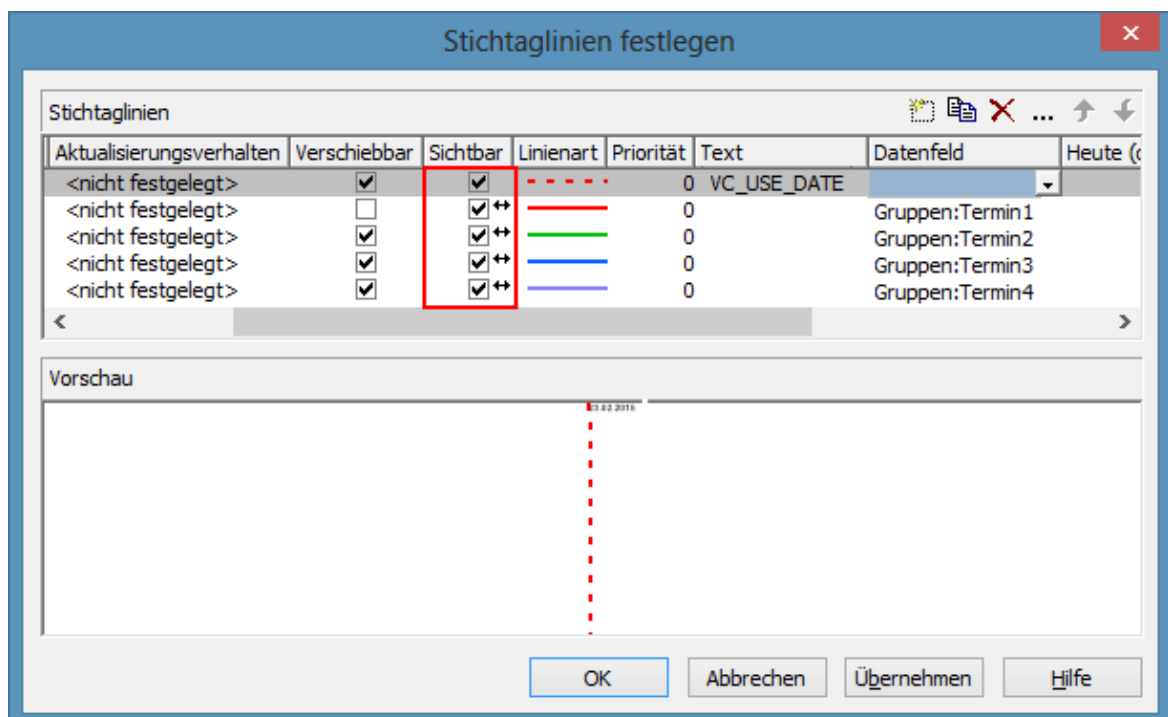
VcGroupLevelLayout.DateLineName

Beschriftung von Stichtaglinien

Stichtaglinien können mit einer Beschriftung versehen werden. Dies geschieht in der Regel durch einen festen Text. Generell bei allen, speziell aber bei individuellen Stichtaglinien ist ggf. die Ausgabe des individuellen Datums gewünscht. Durch das Schlüsselwort **VC_USE_DATE** wird das jeweilige Datum an der spezifizierten Stelle der Stichtaglinie (**VcDateLine.LabelPosition**) im eingestellten Datumsformat (**VcGantt.DateOutputFormat**) ausgegeben.



Damit die Stichtaglinien individuell sichtbar gemacht werden können, kann die Option **Sichtbar** datengetrieben und damit individuell festgelegt werden.



Die entsprechenden API-Eigenschaften:

VcDateLine.VisibleDataFieldIndex

VcDateLine.VisibleMapName.

3.25 Tabelle

Die Eigenschaften der Tabelle können Sie mit Hilfe von drei Dialogen festlegen, die Sie über die Eigenschaftenseite **Objekte** und die Auswahl taste **Tabelle** erreichen. Die dort aufrufbaren Dialoge heißen **Tabelle festlegen**, **Tabelle bearbeiten** und **Tabellenformat bearbeiten**. Sie können im ersten Dialog auch mehrere Tabellen anlegen.

Die Tabelle besteht aus standardmäßig sechs Spalten, die jeweils nur bei einer Breite größer 0 sichtbar sind. Die Zeilen der Tabelle werden durch sogenannte Tabellenformate beschrieben. Für jedes Tabellenformat können Sie Schriftart, Schriftfarbe, Hintergrundfarbe sowie Ausrichtung und Ränder festlegen. Die einzelnen Tabellenformate werden jeweils unter bestimmten Bedingungen verwendet:

- **StandardListCaption** für den Tabellenkopf
- **StandardList** für alle Vorgänge/Zeilen, sofern nichts anderes zutrifft.

Zusätzlich zu den Standardtabellenformaten können Sie weitere Formate anlegen, für die Sie Namen und Filter individuell festlegen können.

Tabellenformate für hierarchische Anordnung:

Die hierarchische Anordnung kann auf der Eigenschaftenseite **Objekte** gesetzt werden, indem Sie dort auf die Schaltfläche **Gruppierung** klicken.

- **Hierarchy**: Format für die Hierarchieebenen in expandiertem Zustand. Dabei wird das 2. Feld (normalerweise der Vorgangsname) ebenenweise eingerückt. Ein "-" zeigt an, dass kollabiert werden kann.
- **HierarchyCollapsed**: Format für die Hierarchieebenen in kollabiertem Zustand. Ein "+" zeigt an, dass expandiert werden kann.

ID	NAME	START
1	☐ SW Development	02.09.98
1.2	☒ Design&Concept	02.09.98
1.3	☐ Coding	09.09.98
1.3.1	Phase A (DB)	09.09.98
1.3.2	Phase B (GUI)	15.09.98
1.4	☒ Testing	17.09.98
1.5	Sales & Marketing	05.09.98
1.6	Delivery	24.09.98
1.7	Final Party	

Bild oben: Das Format **HierarchyCollapsed** in der Zeile **Design&Concept** mit kollabierter Hierarchieebene und das Format **Hierarchy** in der Zeile **Coding** mit expandierter Hierarchieebene

Tabellenformate für gruppierte Anordnung:

Eine Gruppierungs-Anordnung kann auf der Eigenschaftenseite **Objekte** gesetzt werden, indem Sie dort auf die Schaltfläche **Gruppierung** klicken.

- **Subtitle:** für die Gruppenüberschriften nicht kollabierter Gruppen. Die Gruppenüberschrift besteht aus nur einem Feld, das sich über die gesamte Tabellenbreite erstreckt. Ein "-" zeigt an, dass die Gruppe kollabiert werden kann.
- **Collapsed:** Format für die Gruppenüberschriften kollabierter Gruppen. Ein "+" zeigt an, dass die Gruppe expandiert werden kann.

ID	NAME	START
[-] A		
1	SW Development	02.09.98
3	Requirements	02.09.98
7	Final Check	16.09.98
12	QA Requirement Check	23.09.98
[+] Group C		
[+] Group B		
[-] E		
15	Final Party	30.09.98

Bild oben: Das Format **Subtitle** in den Zeile **GroupC** und **GroupB** mit kollabierter Gruppen-Ebene und das Format **Collapsed** in der Zeile **E** mit expandierter Gruppenebene

3.26 Tooltips zur Laufzeit

Sie können Tooltips verwenden, um Informationen über das mit der Maus berührte Objekt bereitzustellen. Mit Hilfe des Ereignisses **OnToolTipText** bzw. **OnToolTipTextAsVariant** können Sie die Texte aller zur Laufzeit erscheinenden Tooltips bearbeiten, z. B. um sie in unterschiedliche Sprachen zu übersetzen oder zu unterdrücken.

Das Ereignis **OnToolTipTextAsVariant** benötigen Sie, wenn Sie eine Skriptsprache benutzen, die keine Rückgabe von Strings erlaubt, z. B. VBScript.

Aktivieren Sie dazu das Kontrollkästchen **OnToolTipText-Ereignisse** auf der Eigenschaftenseite **Allgemeines**, oder setzen Sie die Eigenschaft **ShowToolTip** auf den Wert True, um das Ereignis zu aktivieren.

Code-Beispiel

```
VcGantt1.ShowToolTip = True
```

Fangen Sie dann das Ereignis **OnToolTipText** bzw. **OnToolTipTextAsVariant** ab und legen Sie fest, welcher Text erscheinen soll, oder ob an dieser Stelle kein Tooltip erscheinen soll.

3.27 Unicode-Zeichen

Damit zur Entwurfszeit in den Eigenschaftenseiten Unicode-Zeichen erscheinen, muss unter **Start / Systemsteuerung / Einstellungen / Anzeige / Darstellung** dem Bildelement **Fenster** ein geeigneter Schrifttyp zugewiesen werden.

Außerdem können trotz des Schrifttyps nur Zeichen angezeigt werden, deren zugehörige Sprache unter **Start / Systemsteuerung / Einstellungen / Region- und Spracheinstellungen** gewählt wurde.

Zur Laufzeit kann jedes Objekt in einer VARCHART-Komponente, das Texte enthält, Unicode-Zeichen darstellen, wenn ein geeigneter Schrifttyp bei der zugehörigen Eigenschaft **Font** gesetzt ist.

Den Kontextmenüs, Tooltips und Laufzeit-Dialogen kann über die Eigenschaft **DialogFont** des Objektes **DummyObject** ein Unicode-Schrifttyp zugewiesen werden.

Eine Übersicht über alle verfügbaren Fonts, die zumindest einen Teil aller Unicode-Zeichen enthalten, ist bei "Wazu Japan's Gallery of Unicode Fonts" zu finden ([http:// www.wazu.jp/index.html](http://www.wazu.jp/index.html)). Eine weitere gute Informationsquelle für den Unicode-Standard bietet die Homepage des Unicode-Konsortiums ([http:// www.unicode.org](http://www.unicode.org)) sowie die Einführung in Unicode auf der GlobalDev-Homepage von Microsoft (http://www.microsoft.com/globaldev/getwr/steps/wrg_unicode.mspx). Unter **Start / Programme / Zubehör / Systemprogramme / Zeichentabelle** kann man sich in Windows 2000 und XP darüber informieren, welcher installierte Schriftarttyp welche Zeichen enthält.

Beim Import von CSV-Dateien erkennt die Methode **VcGantt.Load** automatisch, ob die Datei im Unicode- oder ANSI-Format vorliegt.

Hinweis: Die Entwicklungsumgebungen von Visual Studio 6 können in Quellcodedateien keine Unicode-Zeichen verwenden. Die interne Zeichendarstellung in Strings von Visual Basic 6 ist aber Unicode. Bei Visual C++ in Verbindung mit MFC muss man die Defines `_UNICODE` und `UNICODE` setzen, um Strings in Unicode zu verwenden. Ab Visual Studio .NET 2002 ist das Editieren von Quellcodedateien in Unicode-Kodierung möglich, beim Speichern muss man als Kodierung "Unicode" auswählen.

3.28 Verbindungen

Eine Verbindung entspricht einem Datensatz aus der Datentabelle, die die Verbindungsdaten enthält. Verbindungsdaten werden zusammen mit den Knotendaten in einer Datei abgelegt. Verbindungen können über die API geladen oder interaktiv vom Anwender erzeugt werden.

> Verbindungen erzeugen

Zur Laufzeit können Sie mit der Maus Verbindungen zwischen zwei Vorgängen ziehen, wenn Sie den **Modus: Verbindung ziehen** gewählt haben.



Die Verbindungslinie geht dabei jeweils vom 1. Layer des Anfangsvorgangs zum 1. Layer des Endvorgangs. Jedes interaktive Neuanlegen einer Verbindung wird der Applikation mit dem Ereignis **OnLinkCreate** mitgeteilt. Sie können Verbindungen auch über die API mit **InsertLinkRecord** anlegen.

> Verbindungen löschen

Eine Verbindung können Sie löschen, indem Sie sie mit der rechten Maustaste anklicken und dann im Kontextmenü den Befehl **Löschen** wählen. Außerdem können Sie Verbindungen über die API mit der VARCHART-ActiveX-Methode **DeleteLinkRecord** oder mit der Methode **VcLink.DeleteLink** löschen.

> Ereignisse

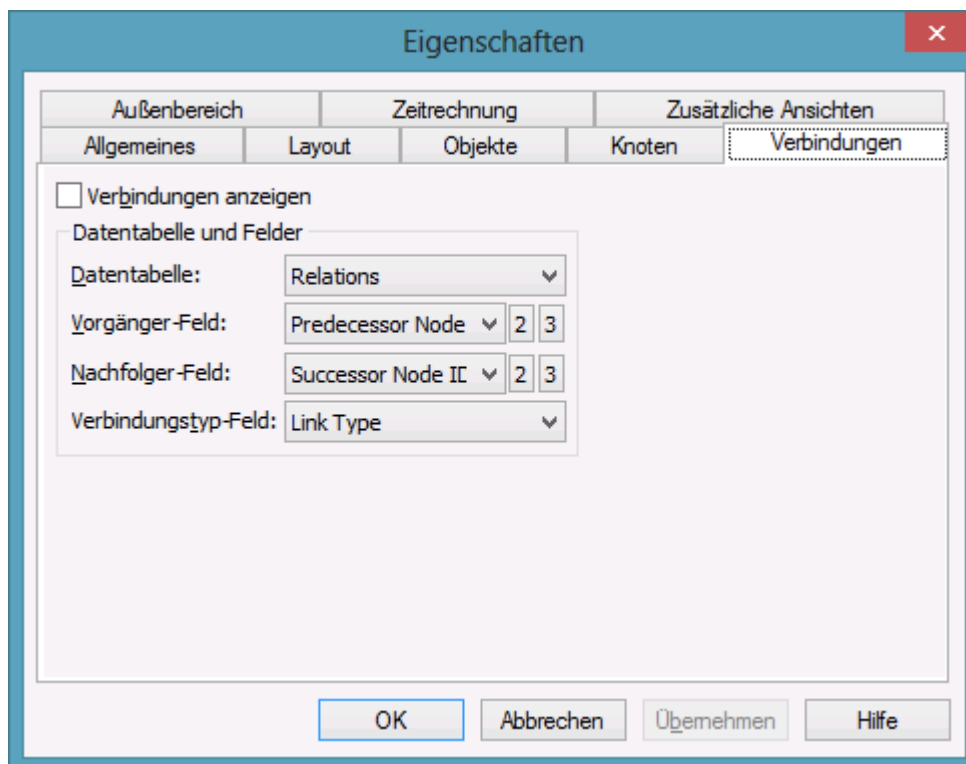
Auf folgende Ereignisse können Sie reagieren:

- **OnLinkCreate**
- **OnLinkCreateComplete**
- **OnLinkDelete**
- **OnLinkDeleteComplete**
- **OnLinkLClickCltn**
- **OnLinkLDbClickCltn**
- **OnLinkRClickCltn**

> Verbindungen festlegen

Auf der Eigenschaftenseite **Verbindungen** können Sie festlegen, ob die Verbindungen angezeigt werden sollen, und ggf. die Verbindungen spezifizieren.

Sie können hier die Datenfelder festlegen, in denen die Identifizierung des Vorgänger- und des Nachfolgerknotens sowie des Verbindungstyps festgelegt ist. Wenn die Identifikation eines Vorgänger - oder Nachfolgerknotens mehrteilig ist, muss auch die jeweilige Verbindung entsprechend aufgebaut sein d.h. dass bei **Vorgänger-Feld** bzw. **Nachfolger-Feld** ggf. ein zweites oder drittes Feld entsprechend der ID des jeweiligen Knotens angegeben werden muss. standardmäßig wird jeweils das erste Feld angezeigt. Zur Angabe des zweiten oder dritten Feldes klicken Sie auf die entsprechenden Schaltflächen und wählen dann aus der Drop-Down-Liste das gewünschte Feld aus.



Darüberhinaus können im Dialog **Verbindungsausssehen verwalten** verschiedene Verbindungsausssehen erstellt und bearbeitet werden, für die Filter, Vorgänger- und Nachfolger-Layer, Linienart, Vorgänger- und Nachfolger-Portsymbole sowie die Art der Verbindungsführung festgelegt werden können.

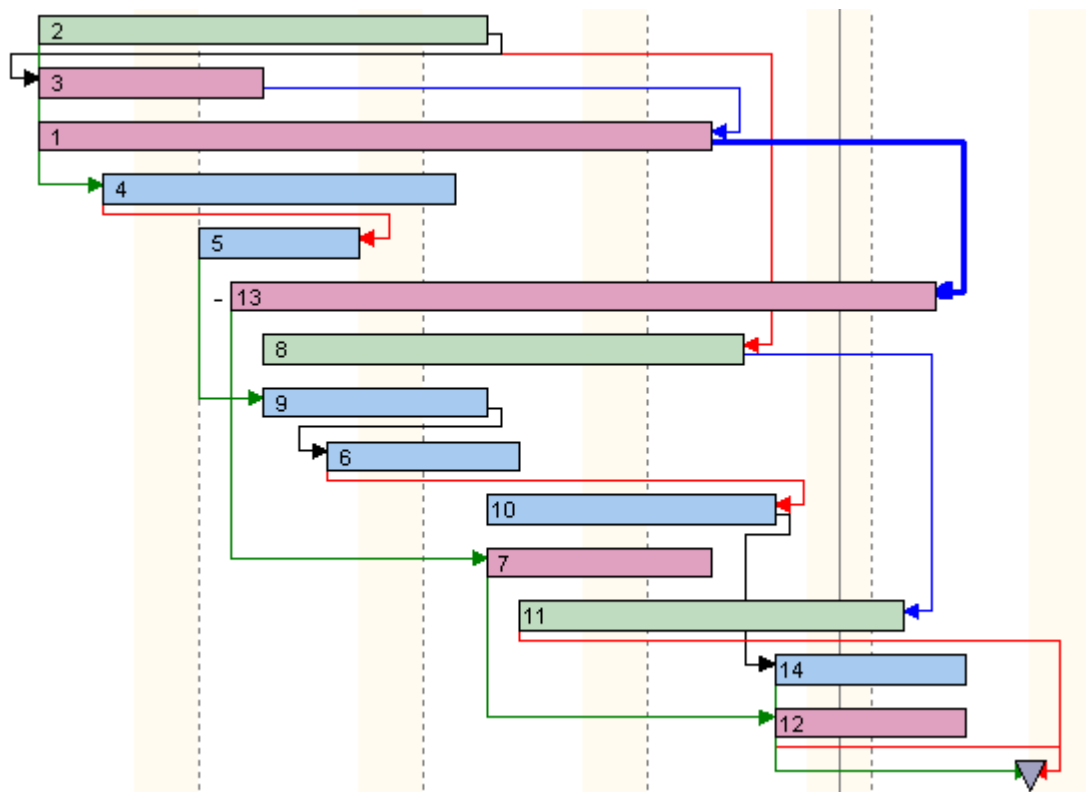
> Verbindungstypen

Auf der Eigenschaftenseite **Verbindungen** legen Sie unter **Verbindungstyp-Feld** fest, aus welchem Datenfeld der Verbindungstyp gelesen werden soll.

Verbindungstypen:

- FF: Ende-Ende
- FS: Ende-Anfang
- SF: Anfang-Ende
- SS: Anfang-Anfang

Sie können mit Hilfe dieses Datenfeldes den Verbindungstyp durch die entsprechende Linienführung darstellen.



Einige Beispiele für verschiedene Verbindungstypen

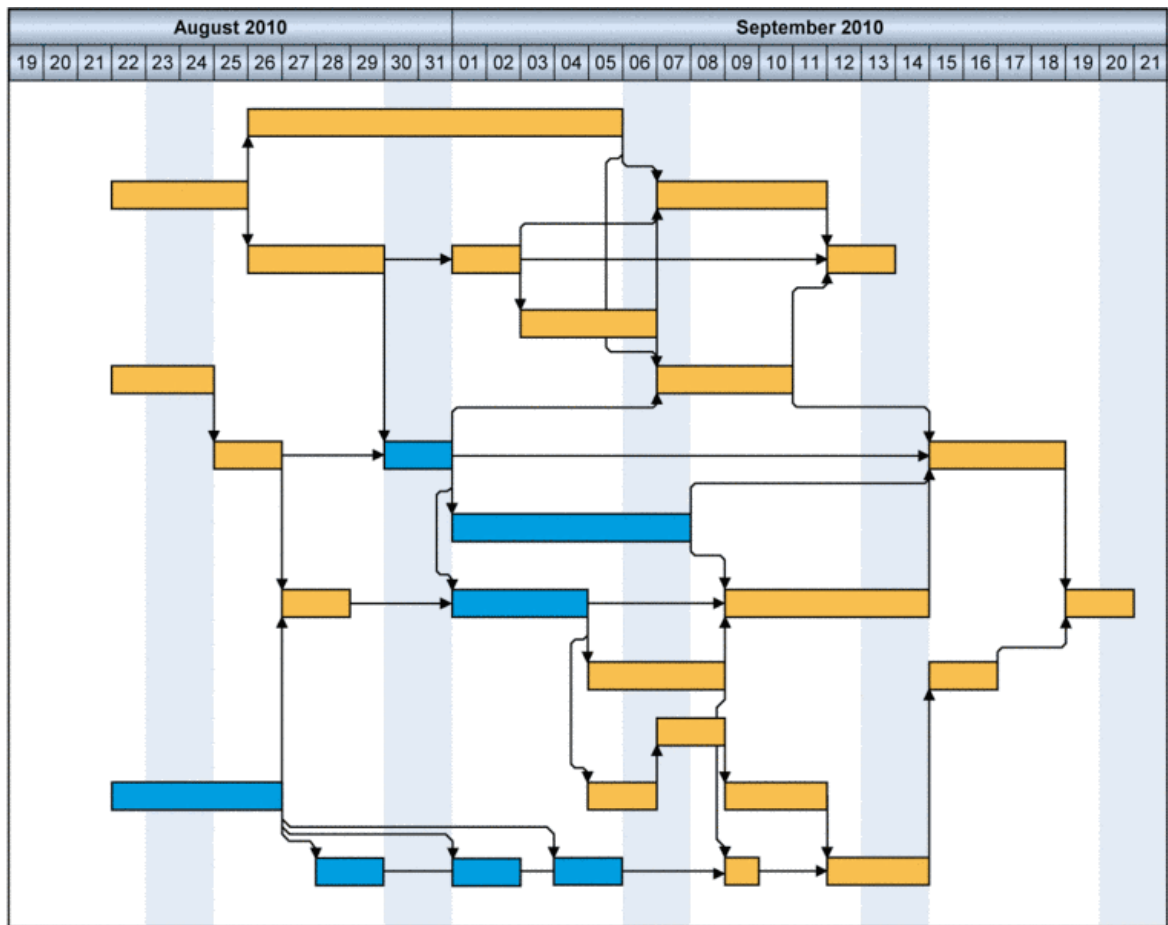
> Automatisches Layout

Für eine optimierte Linienführung der Verbindungen steht ein Layouter zur Verfügung. Durch diesen wird die Linienführung stets eindeutig und es ist immer klar erkennbar, woher die Verbindung kommt und wohin sie führt.

Zudem wird die Linienführung so optimiert, dass durch die Schachtelung von Krümmungen möglichst wenige Linienkreuzungen entstehen.

Die Höhe von Zeilen wird automatisch angepasst, um Platz für die parallelen horizontalen Verbindungsabschnitte zu schaffen.

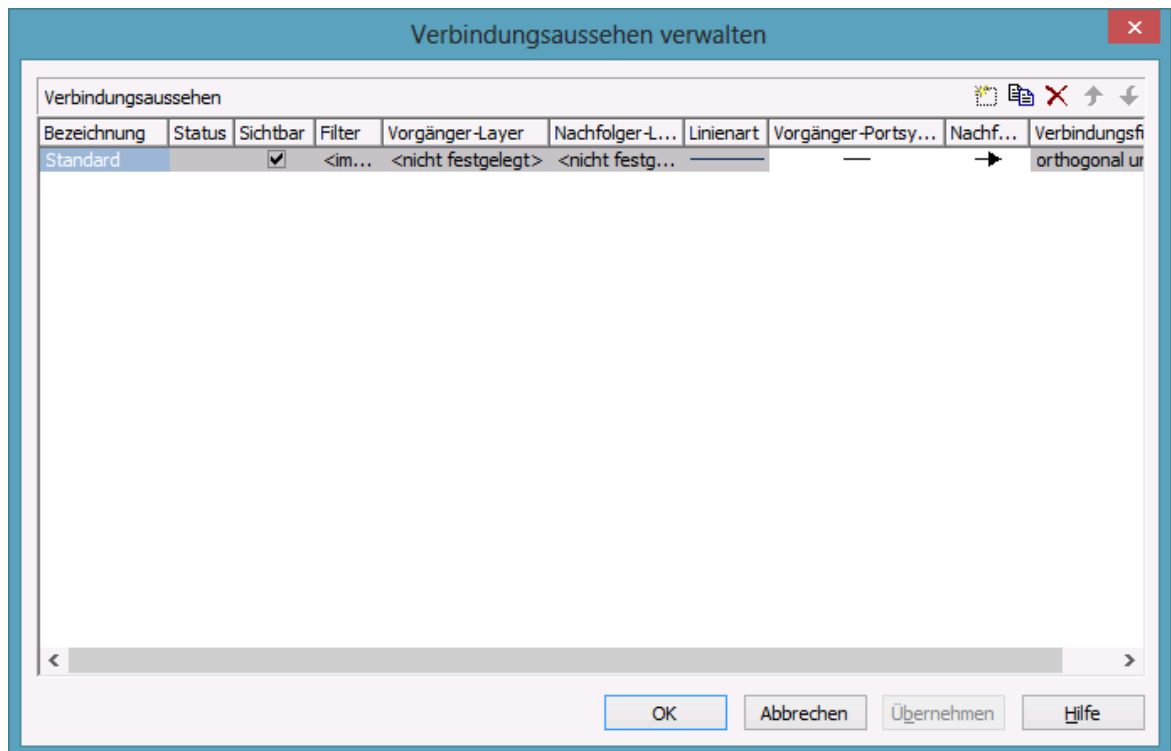
Kleine Schrägen in den Ecken zeigen die Änderung der Verlaufsrichtung einer Verbindung an.



3.29 Verbindungsaussehen

Für die Verbindungen lassen sich im Dialog **Linienaussehen verwalten** unterschiedliche Verbindungsaussehen definieren, die den Verbindungen dynamisch über Filter zugewiesen werden.

> Neues Verbindungsaussehen definieren



3.30 Verschiebehilfen

In Gantt-Charts kann einfach in die Planung eingegriffen werden, indem Aufträge, Vorgänge oder Ressourcen per Maus verschoben werden. Die exakte Positionierung ist jedoch nicht immer ganz einfach, da mit der Maus eine bestimmte Stelle im Gantt-Diagramm genau getroffen werden muss.

In vielen Gantt-Anwendungen gibt es zudem auch mehrstufige Gruppen. In großen Plänen und wenn Quell- und Zielgruppen weit voneinander entfernt liegen, kann so das Verschieben eines Knotens per Maus rasch unübersichtlich werden

Verschiebehilfen in horizontaler Richtung (Einrastwerkzeuge)

Aus Zeichenprogrammen oder Designer-Tools ist das sog. Einrastgitter bekannt, eine Hilfe zur exakten Positionierung von Objekten anhand eines vorgegebenen Gitters, meist in Pixel-Abständen gemessen. Eine ähnliche Funktionalität wurde in VARCHART XGantt implementiert. Dort werden die verschobenen Objekte aber nicht an einem festen Gitter ausgerichtet, sondern an anderen Objekten in der Grafik, die dann ein Einrastgitter mit unregelmäßigen Abständen festlegen.

> Objekte als Einrastziele

Bei Knoten (bzw. deren Layern), Stichtaglinien, Liniengittern und Kalendergittern ist die Definition von sog. Einrastzielen möglich, d.h. diese Objekte definieren bestimmte Stellen an sich selbst, die als Ziele für eine Einrastaktion anderer Objekte dienen. Beim horizontalen Verschieben oder Ändern der Größe eines Knotens oder Layers wird dessen Start- bzw. Endtermin zeitlich an den definierten Einrastzielen der anderen Objekte ausgerichtet. Der Start- oder Endtermin springt dabei innerhalb von 5 Pixeln neben einem Einrastziel an dieses heran und übernimmt damit dessen exakten Termin.

> Spezielles Verhalten für jedes Knotenlayout

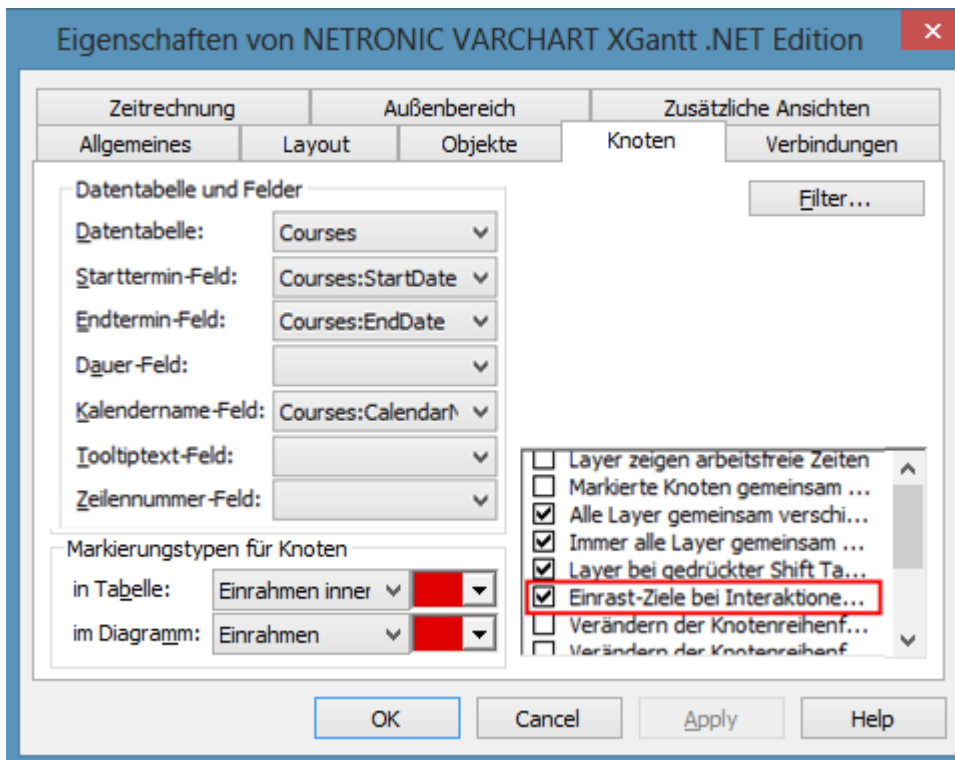
Für jedes Knotenlayout (ungruppiert, gruppiert, hierarchische Anordnung) wurde ein gesondertes Verhalten festgelegt (vorausgesetzt, die entsprechenden Objekte definieren Einrastziele):

- Für alle Knotenlayouts: Der zu verschiebende Layer wird an Stichtaglinien, Liniengittern und Kalendergittern ausgerichtet.

- Ungruppierte Anordnung: Der zu verschiebende Layer wird an den Layern aller Knoten ausgerichtet.
- Gruppierte Anordnung: Der zu verschiebende Layer wird an den Layern der Knoten derselben Gruppe (ohne Untergruppen) ausgerichtet. Bei einem eventuellen Wechsel der Gruppe während der Interaktion wird der Layer an den Objekten der neuen Gruppe ausgerichtet.
- Hierarchische Anordnung: Der zu verschiebende Layer wird an den Layern der Knoten desselben Astes (mit Unterästen) ausgerichtet. Bei einem evtl. Wechsel des Astes während der Interaktion wird der Layer an den Objekten des neuen Astes ausgerichtet.

> Neue Eigenschaften und API-Aufruf

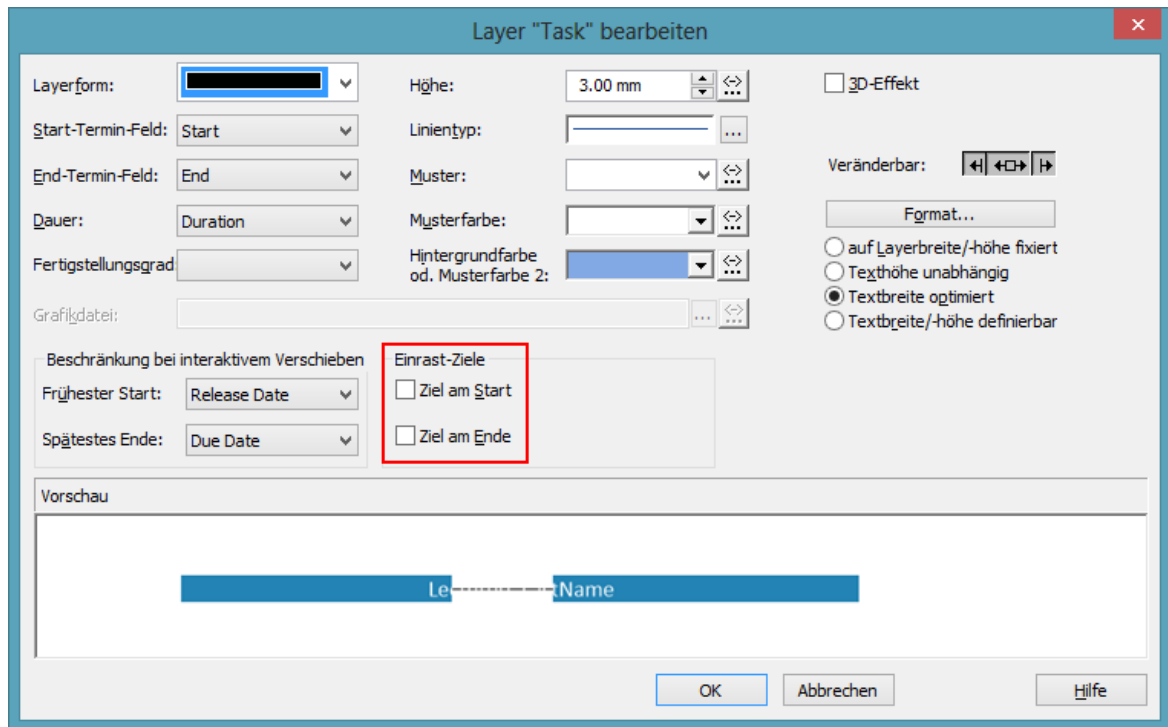
Damit die Einrastwerkzeuge wirksam werden, müssen sie auf der Eigenschaftenseite **Knoten** zugelassen werden:



API-Aufruf: **vcGantt.UseSnapTargetsInInteractions = true/false**

> Einrastziel LAYER

Im Dialog Layer bearbeiten kann durch Auswahl der Checkboxes **Ziel am Start-Termin** und **Ziel am End-Termin** bestimmt werden, dass der Layer seine Position (also seine Termine) als Einrastziele für das Verschieben eines Knotens bzw. Layers definiert.



API-Aufrufe:

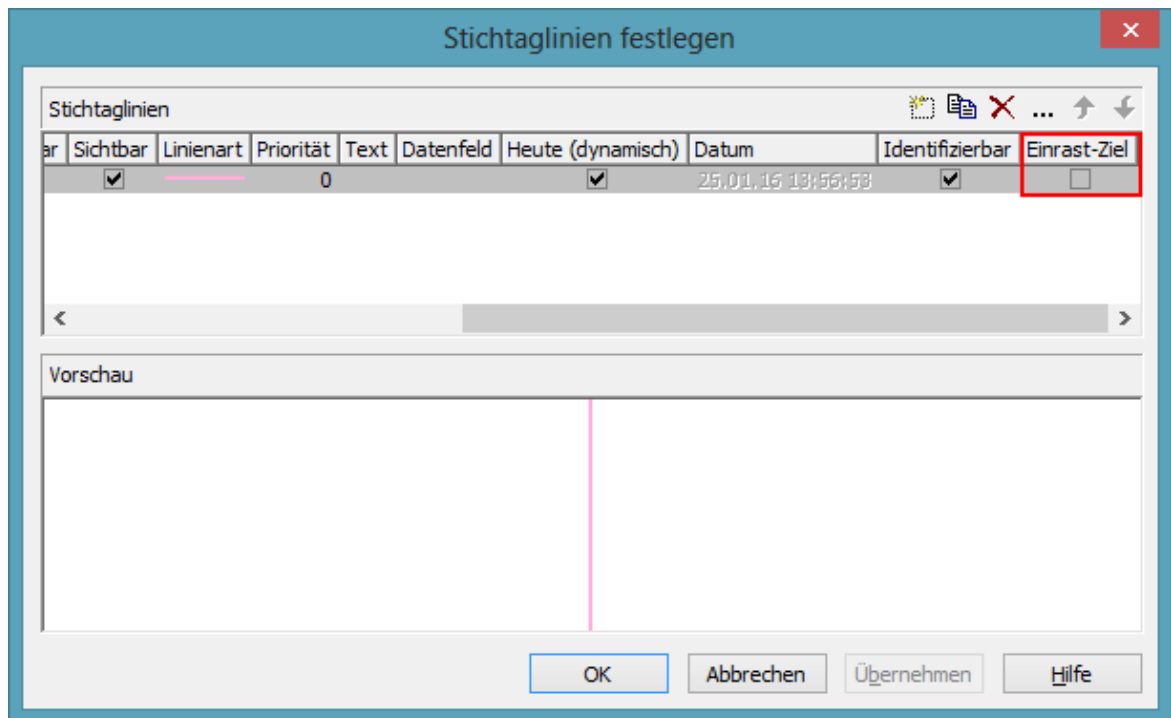
VcLayer.StartSnapTarget = true/false

VcLayer.EndSnapTarget = true/false

> Einrastziel STICHTAGLINIE

Im Dialog Stichtaglinien festlegen gibt es für jede Stichtaglinie eine entsprechende Checkbox Einrast-Ziel. Wenn sie ausgewählt ist, definiert die Stichtaglinie ihre Position (also ihr Datum) als Einrastziel für das Verschieben eines Knotens oder Layers.

190 Wichtige Konzepte: Verschiebehilfen



API-Aufruf: **VcDateLine.SnapTarget = true/false**

> **Einrastziel LINIEN-/KALENDERGITTER**

Für Liniengitter und Kalendergitter können die Einrastziele an zwei verschiedenen Stellen definiert werden:

- im Dialog Zeitskalenabschnitt bearbeiten für nicht individuelle Objekte
- unterhalb des Gruppierung-Dialogs für individuelle, auf Gruppen oder Knoten bezogene Objekte

Wenn die entsprechenden Checkboxes im Dialog Zeitskalenabschnitt bearbeiten ausgewählt sind, definieren die jeweiligen Objekte ihre Positionen (also ihre Termine) als Einrastziele für das Verschieben eines Knotens oder Layers.

Zeitskalenabschnitt "Section_1" bearbeiten

Breite pro Zeiteinheit: ☐ Kollabieren arbeitsfreier Zeiten

Zeitsstreifen mit 3D-Effekt: ☐

Index	Typ	Position	Minimale Höhe	Hauptmarkierungen	Nebenmarkierungen	Muster	Schrift	Datumsformat	Einheitentrennung
0	Monate	Oben	5.2 mm	1	1		8 pt, Arial	TM YYYY	durchgezogene Linien
1	Tage	Oben	5.2 mm	1	1		8 pt, Arial N...	DD	durchgezogene Linien

Liniengitter

Referenzdatum	Am Referenzdatum ausrichten	Sommerzeit beachten	Einrast-Ziel	Linienformat	Drehen	Ausrichtung	Oben	Mitte	Unten
	☐	nicht festgelegt	☐	<nicht festgelegt>	☐	☐	☐	☐	☐
	☐	nicht festgelegt	☐	<nicht festgelegt>	☐	☐	☐	☐	☐

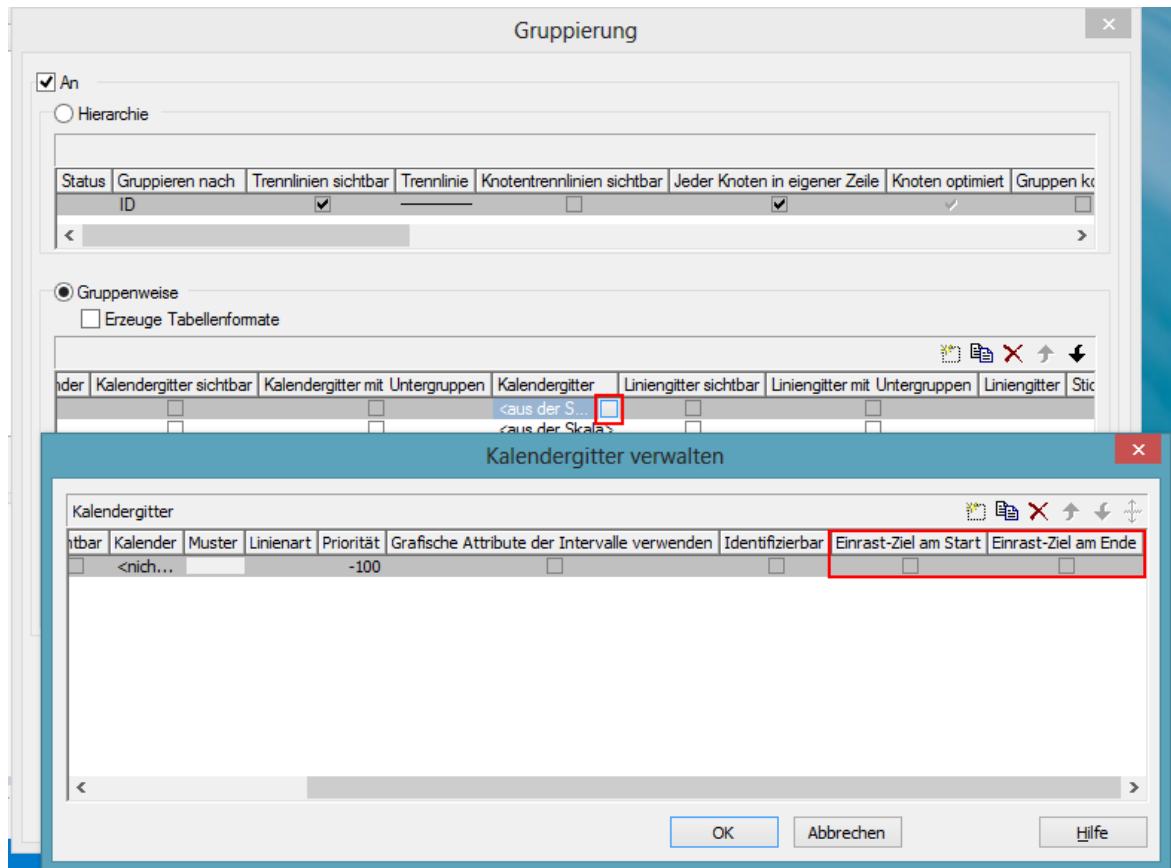
Kalendergitter

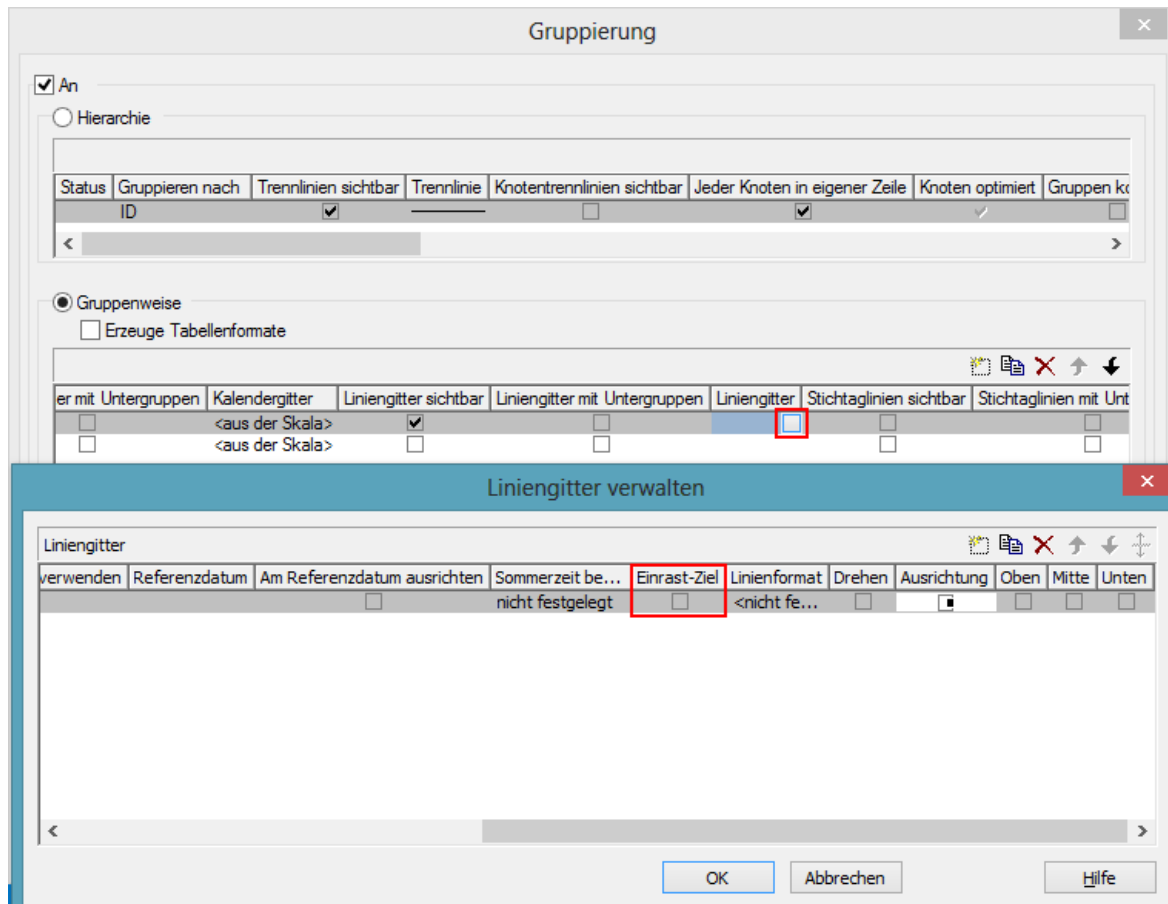
Index	Muster	Linienart	Priorität	Grafische Attribute der Intervalle verwenden	Identifizierbar	Einrast-Ziel am Start	Einrast-Ziel am Ende
<nicht festgelegt>			-25	☐	☐	☐	☐

OK Abbrechen Hilfe

Aus dem Dialog **Gruppierung** kann man jeweils in die Dialoge **Kalendergitter** **verwalten** bzw. **Liniengitter** **verwalten** verzweigen. Dort kann durch Auswahl der entsprechenden Checkboxes bestimmt werden, dass die jeweiligen Objekte ihre Positionen (also ihre Termine) als Einrastziele für das Verschieben eines Knotens oder Layers definieren.

192 Wichtige Konzepte: Verschiebehilfen





API-Aufrufe:

VcDateLineGrid.SnapTarget = true/false

VcCalendarGrid.StartSnapTarget = true/false

VcCalendarGrid.StartSnapTarget = true/false

Achtung: Einrastziele von individuellen Objekten werden nur dann berücksichtigt, wenn ein einzelner Knoten verschoben wird, da es nicht sinnvoll ist, beim gleichzeitigen Verschieben mehrerer Knoten die Einrastziele aller Objekte (d.h. der Objekte aus mehreren Bändern) zu mischen. Ein separates Einrasten eines Knotens an den Einrastzielen des Bandes, in dem er liegt, ist nicht vorgesehen.

> Verschieben eines Knotens mit den Pfeiltasten

Knoten können nicht nur interaktiv mit der Maus verschoben werden, sondern auch über die Pfeiltasten auf der Tastatur. Zuvor muss allerdings noch folgende Einstellung vorgenommen werden:

vcGantt1.ArrowKeyMode = VcArrowKeyMode.vcResizeOrMoveNode

Der Aufzählung **VcArrowKeyMode** wurde der Wert **vcNodeJumpToSnapTarget** hinzugefügt. Wenn dieser Wert gesetzt ist,

kann durch Drücken von STRG + Pfeiltaste links bzw. rechts mit einem zuvor markierten Knoten zum vorherigen bzw. nächsten Einrastziel gesprungen werden. Diese Aktion ist zyklisch, d.h. ist man an einem Ende angekommen, beginnt man wieder vorne.

Verschiebehilfen in vertikaler Richtung (Automatisches Kollabieren und Expandieren von Gruppen)

Im Windows-Explorer wird beim Verschieben einer Datei die Ordnerstruktur automatisch aufgeklappt. Dabei schiebt man die Datei auf einen kollabierten Ordner im Navigationsbereich, verharnt einen Moment, der Ordner öffnet sich und man kann weiter schieben, bis man den gewünschten Zielordner gefunden hat.

> Verhalten in älteren Versionen

In VARCHART XGantt musste man bisher beim vertikalen Verschieben eines Knotens in eine andere Gruppe recht lange nach der Zielgruppe suchen, wenn das Diagramm sehr viele Knoten in vielen expandierten Gruppen enthielt. Um die gesuchte Zielgruppe zu erreichen, war meist ein automatisches vertikales Scrollen notwendig, was mitunter langwierig und daher unkomfortabel war.

> Neu: Einfache Orientierung & schnelles vertikales Verschieben

Mit der neuen Funktionalität lässt sich die Suche nach der Zielgruppe deutlich abkürzen. Da hierbei die Kombinations- und Einstellmöglichkeiten ausgesprochen vielfältig sind, möchten wir uns an dieser Stelle darauf beschränken, eine mögliche Konfiguration vorzustellen.

Beispiel: Alle Gruppen außer der aktuellen kollabieren automatisch

Man kann VARCHART XGantt so konfigurieren, dass beim Verschieben eines Knotens alle Gruppen mit Ausnahme der gerade berührten Gruppe automatisch kollabiert werden. Der Zustand dieser Gruppe wird für den Fall beibehalten, dass man den Knoten nur innerhalb derselben Gruppe verschieben möchte. Durch das Kollabieren der anderen Gruppen reduziert sich die vertikale Ausdehnung des Plans auf einen Bruchteil der ursprünglichen Ausdehnung, wodurch wesentlich mehr Gruppen im Bild sichtbar sind als bisher. Im Idealfall sieht man schon jetzt die gesuchte Zielgruppe. Wenn nicht, kann VARCHART XGantt automatisch über die kollabierten Gruppen scrollen. Damit lässt sich die gewünschte Zielgruppe

wesentlich schneller finden als bisher. Ist die Zielgruppe erreicht, verharrt man einen Moment, die Zielgruppe wird expandiert und man kann weiterschieben. Dabei wird die zuvor berührte Gruppe kollabiert, so dass die Größe des Plans minimiert bleibt. Man schiebt weiter, ggf. noch einmal auf eine andere Gruppe. Diese wird expandiert, die zuvor expandierte Gruppe kollabiert usw., bis man sein Ziel erreicht hat. Mit dem Loslassen des Knotens in der Zielgruppe ist die Interaktion beendet und VARCHART XGantt kann, falls gewünscht, den alten Zustand der Gruppen wieder herstellen, wobei auf die neue Position des verschobenen Knotens gescrollt wird.

> **Viele Kombinationsmöglichkeiten**

Dies war nur ein Beispiel für die neue Funktionalität. Weitere Einstellmöglichkeiten gibt es für:

- automatisches Kollabieren von Gruppen
- automatisches Expandieren von Gruppen
- automatisches Restaurieren von automatisch kollabierten oder expandierten Gruppen, die sich über ein Aktualisierungsverhalten zeitlich genau steuern lassen.

Diese Einstellungen können pro Gruppenebene und auch für die hierarchische Anordnung der Knoten vorgenommen werden, wodurch man die Verschiebevorgänge sehr detailliert gestalten kann.

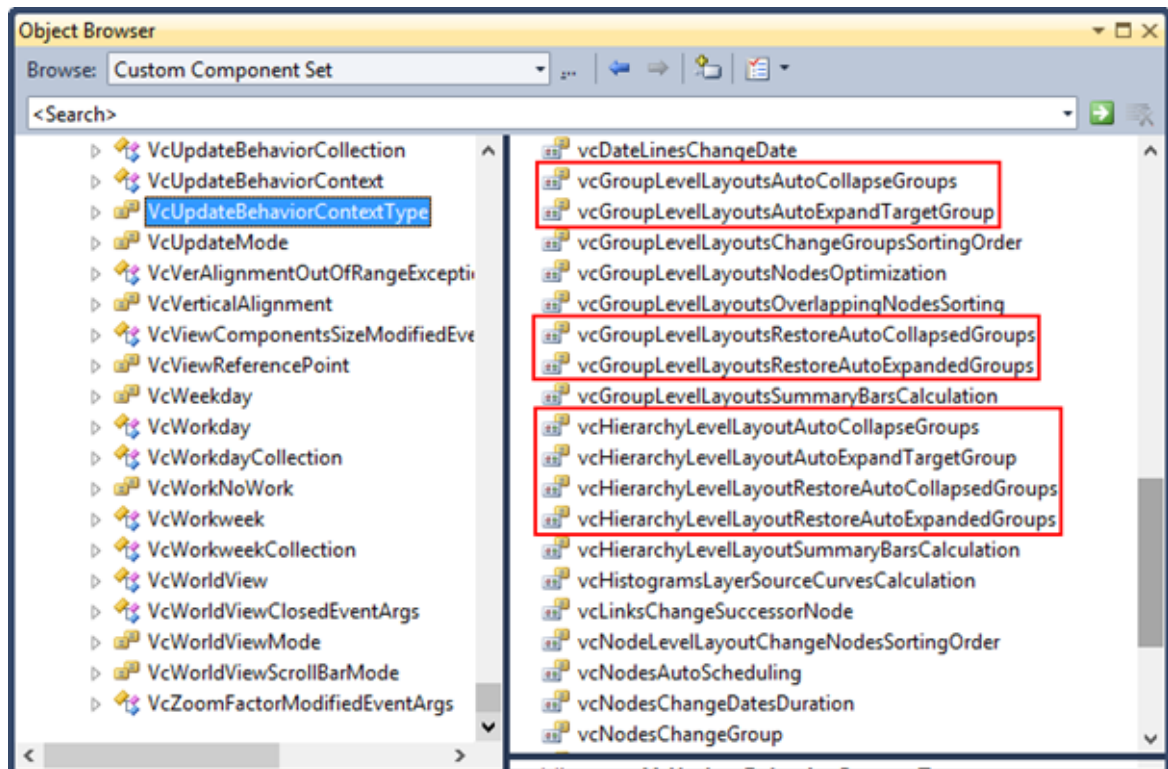
> **Neue Eigenschaften und API-Aufrufe**

Im Dialog **Aktualisierungsverhalten bearbeiten** gibt es acht entsprechende Kontexte, jeweils vier bei **Gruppenzeilen-Layouts** und bei **Hierarchie-Layout**:

196 Wichtige Konzepte: Verschiebehilfen

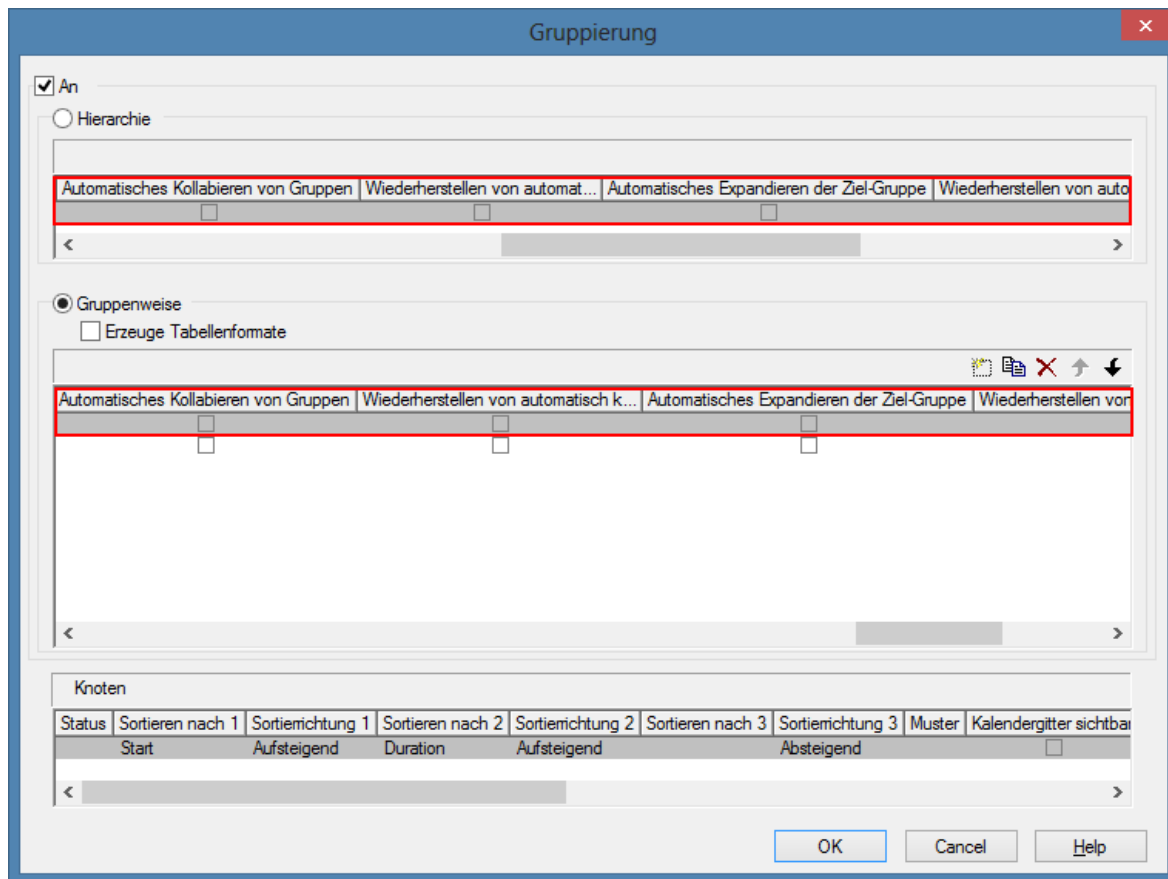
Aktualisierungsverhalten "LiveUpdate" bearbeiten			
Kontexte (Objekte und Funktionen)			
Bezeichnung	Aktualisierungsmodus	Verzögerungszeit	
Tabellen			
Spaltenbreite ändern	Beim Verschieben der Maus	500 ms	
Zeitskalen			
Einheitenbreite ändern	Beim Verschieben der Maus	500 ms	
Anfangsdatum des Zeitskalenabschnitts ändern	Beim Verschieben der Maus	500 ms	
Stichtaglinien			
Datum ändern	Beim Verschieben der Maus	500 ms	
Boxen			
Größe ändern	Beim Verschieben der Maus	500 ms	
Position ändern	Beim Verschieben der Maus	500 ms	
Verankerungsknoten ändern	Beim Verschieben der Maus	500 ms	
Knoten			
Termine/Dauer ändern	Beim Verschieben der Maus	500 ms	
Gruppe ändern	Beim Verharren während d...	500 ms	
Filtern/Mappen	Beim Verharren während d...	500 ms	
Gruppieren	Beim Verharren während d...	500 ms	
Automatische Terminberechnung	Beim Verharren während d...	500 ms	
Verbindungen			
Nachfolgerknoten ändern	Beim Verschieben der Maus	500 ms	
Knotenzeilen-Layout			
Sortierreihenfolge der Knoten ändern	Beim Verharren während d...	500 ms	
Hierarchie-Layout			
Berechnung von Summenbalken	Beim Verschieben der Maus	500 ms	
Automatisches Kollabieren von Gruppen	Beim Verharren während d...	500 ms	
Restaurieren von automatisch kollabierten Gru...	Beim Lösen der Maustaste	0 ms	
Automatisches Expandieren einer Ziel-Gruppe	Beim Verharren während d...	500 ms	
Restaurieren von automatisch expandierten G...	Beim Verharren während d...	500 ms	
Gruppenzeilen-Layouts			
Sortierreihenfolge der Gruppen ändern	Beim Verharren während d...	500 ms	
Knotenoptimierung	Beim Verharren während d...	500 ms	
Berechnung von Summenbalken	Beim Verschieben der Maus	500 ms	
Sortierung von überlappenden Knoten	Beim Verharren während d...	500 ms	
Automatisches Kollabieren von Gruppen	Beim Verharren während d...	500 ms	
Restaurieren von automatisch kollabierten Gru...	Beim Lösen der Maustaste	0 ms	
Automatisches Expandieren einer Ziel-Gruppe	Beim Verharren während d...	500 ms	
Restaurieren von automatisch expandierten G...	Beim Verharren während d...	500 ms	
Histogramme			
Berechnung Layer-basierter Kurven	Beim Verschieben der Maus	500 ms	
Numerische Skalen			
Einheitenhöhe ändern	Beim Verschieben der Maus	500 ms	
Kurven			
X-Koordinate ändern	Beim Verschieben der Maus	500 ms	
Y-Koordinate ändern	Beim Verschieben der Maus	500 ms	
X- und Y-Koordinate ändern	Beim Verschieben der Maus	500 ms	
Sashes			
Position ändern	Beim Verschieben der Maus	500 ms	
Komplettansicht			
Größe ändern	Beim Verschieben der Maus	500 ms	
Position ändern	Beim Verschieben der Maus	500 ms	

Um diese neuen Kontexte auch zur Laufzeit einstellen zu können, wurde die Aufzählung **VcUpdateBehaviorContextType** ebenfalls um 8 Werte ergänzt:



Die Funktionalitäten, die durch diese Kontexte durch Timer gesteuert aktiviert werden, können im Dialog Gruppierung ein- und ausgeschaltet werden. Dort wurden jeweils vier Spalten bei Hierarchie und Gruppenweise hinzugefügt.

198 Wichtige Konzepte: Verschiebehilfen



API-Aufrufe:

VcGroupLevelLayout.AutoCollapseGroups = true/false

VcGroupLevelLayout.AutoExpandTargetGroup = true/false

VcGroupLevelLayout.RestoreAutoCollapsedGroups = true/false

VcGroupLevelLayout.RestoreAutoExpandedGroups = true/false

VcHierarchyLevelLayout.AutoCollapseGroups = true/false

VcHierarchyLevelLayout.AutoExpandTargetGroup = true/false

VcHierarchyLevelLayout.RestoreAutoCollapsedGroups = true/false

VcHierarchyLevelLayout.RestoreAutoExpandedGroups = true/false

3.31 Zeitrechnung

Mit dem VARCHART-XGantt-Scheduler können Sie einfache Terminberechnungen durchführen. Die gewünschten Projektstart- und Projektende-Termine werden dabei als Parameter übergeben.

Mit Hilfe der Eigenschaftenseite **Zeitrechnung** können Sie die Zeitrechnung des VARCHART XGantt an Ihre Schnittstelle anpassen, indem Sie festlegen, welche Datenfelder für die Eingabe (**Zeitrechnungseingabe**) und die Ausgabe (**Zeitrechnungsergebnis**) des Schedulers verwendet werden sollen.

Als Ein- und Ausgaben für die Zeitrechnung verwendet der Scheduler Datenfelder der jeweils eingestellten Knoten- und Verbindungstabellen

Die Grundlage der Berechnung sind die Dauer der einzelnen Vorgänge, deren logische Abhängigkeiten und der Projektanfang. Daraus werden die frühesten bzw. spätesten Start- und Endtermine sowie der Gesamtpuffer und der freie Puffer berechnet. Die Felder **Vorgänger** und **Nachfolger** können in der **Zeitrechnungseingabe**-Tabelle nicht bearbeitet werden. Sie geben nur die auf der Eigenschaftenseite **Verbindungen** vorgenommenen Einstellungen wieder.

The screenshot shows the 'Eigenschaften' (Properties) dialog box for VARCHART XGantt, specifically the 'Zeitrechnung' (Time Calculation) tab. The dialog is divided into sections for 'Zeitrechnungseingabe' (Time Calculation Input) and 'Zeitrechnungsergebnis' (Time Calculation Output). The input section lists fields like 'Vorgänger (Teil 1)', 'Nachfolger (Teil 1)', 'Verbindungstyp', and 'Dauer'. The output section lists 'früh. Start', 'früh. Ende', 'spät. Start', 'spät. Ende', 'Freier Puffer', and 'Gesamtpuffer'. There are checkboxes for 'Nur Knoten mit Vorgängern berechnen' and 'Automatische Zeitrechnung'. Buttons at the bottom include 'OK', 'Abbrechen', 'Übernehmen', and 'Hilfe'.

Die Ergebnisse werden wiederum in Datenfelder der Schnittstelle geschrieben. Als Ergebnisse stehen zur Verfügung: **früh. Start**, **früh. Ende**, **spät. Start**, **spät. Ende**, **Freier Puffer** und **Gesamtpuffer**. Jedem dieser

200 Wichtige Konzepte: Zeitrechnung

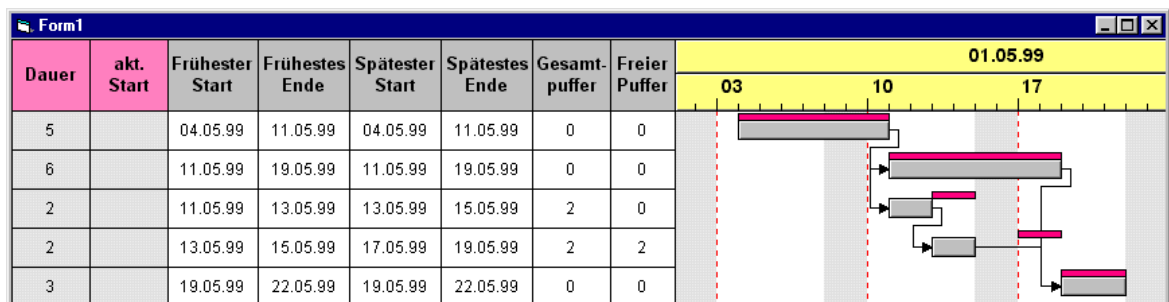
Ergebnisse können Sie ein Feld aus der in der Datendefinition vereinbarten Liste von Feldern zuweisen.

Alle folgenden Beispiele wurden für den Projektstart am 4.5.07 berechnet. Das erreichen Sie über die API mit der folgenden Codezeile:

Code-Beispiel

```
VcGantt1.ScheduleProject "04.05.07, 0"
```

Mit den oben dargestellten Einstellungen ergibt sich folgendes grafisches Ergebnis:



In diesem Beispiel sind die frühen und späten Termine jeweils als Layer dargestellt.

Es gibt noch weitere Möglichkeiten, die Zeitrechnung zu beeinflussen:

1. Sie können aktuelle Start- bzw. Endtermine angeben. Die Vorgänge sind dann unverrückbar.
2. Sie können für die Bedingungen **Start nicht früher als** und **Ende nicht später als** Referenztermine angeben. Dazu wird auf der Eigenschaften-seite **Zeitrechnung** in der linken Tabelle für die entsprechenden Werte auch jeweils ein Feld aus der Datendefinition festgelegt.

Für das folgende Beispiel wurden diese Einstellungen vorgenommen:

Eigenschaften

Allgemeines Layout **Objekte** Knoten Verbindungen
 Außenbereich **Zeitrechnung** Zusätzliche Ansichten

Zeitrechnungseingabe **Zeitrechnungsergebnis**

Eingabe	aus Feld	Ausgabe	in Feld
Vorgänger (Teil 1)	Relations:Prede...	früh. Start	Start
Vorgänger (Teil 2)		früh. Ende	End
Vorgänger (Teil 3)		spät. Start	LateStart
Nachfolger (Teil 1)	Relations:Succe...	spät. Ende	LateEnd
Nachfolger (Teil 2)		Freier Puffer	Free float
Nachfolger (Teil 3)		Gesamtpuffer	Total float
Verbindungstyp	Relations:Link T...		
Verbindungsdauer			
Dauer			
akt. Start			
akt. Ende			
Start nicht früher als			
Ende nicht später als			

☐ Nur Knoten mit Vorgängern berechnen
☐ Automatische Zeitrechnung

OK Abbrechen Übernehmen Hilfe

Wenn der aktuelle Start eines Vorgangs gesetzt ist, werden damit frühe und späte Termine fixiert. In dem folgenden Beispiel ist der gesetzte akt. Start durch ein grünes Dreieck markiert.



Die Verwendung der Bedingungen **Start nicht früher als** und **Ende nicht später als** kann oder kann keine Auswirkungen haben. Im folgenden Beispiel sind die einschränkenden Termine durch rote und grüne Dreiecke visualisiert. Einige haben keinen Einfluss auf die Berechnung. Durch die Ende-Beschränkung des zweiten Vorgangs ergibt sich jedoch ein negativer Puffer für die ersten beiden Vorgänge.



3.32 Zeitskala

Oberhalb des Gantt-Graphen wird die Zeitskala angezeigt. Ggf. können Sie einen oder mehrere beschriftete Zeitstreifen der Zeitskala auch unterhalb des Gantt-Graphen ausgeben (siehe Dialog **Zeitskalenabschnitt bearbeiten, Zeitstreifen, Position**). Sie können aus verschiedenen Zeitskalen die zum dargestellten Zeitbereich passende Zeitskala auswählen.

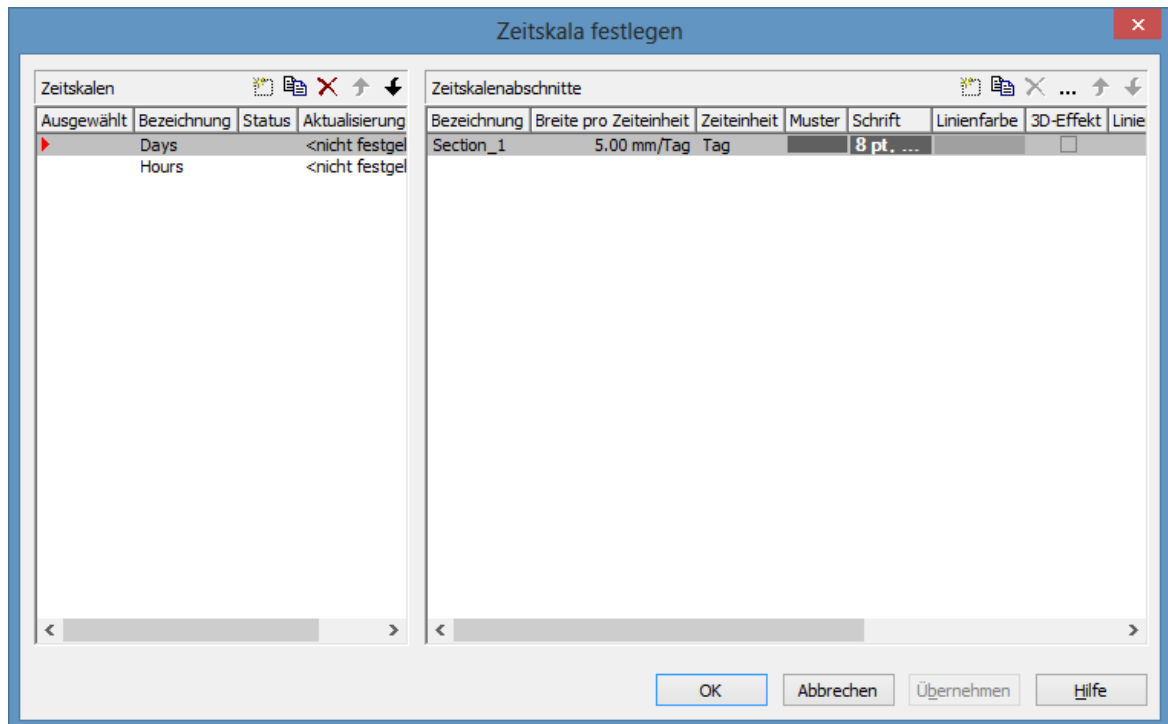
Sie können die Zeitskala in verschiedene Zeitskalenabschnitte unterteilen und deren Anzahl, Ausdehnung und Skalierung festlegen. Zeitabschnitte, die Sie besonders detailliert planen möchten, lassen sich dann gegenüber den übrigen gestreckt darstellen. So lässt sich beispielsweise die Planung für die nähere Zukunft detaillierter darstellen als die Planung für die fernere Zukunft oder der Vergangenheit. Auf diese Weise können Sie den Fokus auf die für Sie interessantesten Planungsabschnitte legen und diesen mit der Zeit verschieben. Sie können so ausgehend von einer Grobplanung sukzessive detaillierter planen.

June 2011					July 2011				
Cw 23	Cw 24	Cw 25	Cw 26	Cw 27	Cw 28	Cw 29	Cw 30	Cw 31	

Für die Darstellung der Zeitskala, Zeitskalenabschnitte und Gitter haben Sie vielfältige Gestaltungsmöglichkeiten. Sie können die Skalierungen, Notationen, Schriftattribute, die Ausrichtung der Beschriftung, die Farben, Strichdicken, Linientypen etc. festlegen, und zwar jeweils individuell für jedes Objekt.

Für die übersichtliche Gestaltung Ihrer Planung können Sie für jeden Zeitskalenabschnitt individuell Linien- und Kalendergitter einfügen.

Im Dialog **Zeitskala festlegen** können Sie verschiedene Zeitskalen definieren und auswählen, welche Sie für Ihre Darstellung verwenden möchten (**Ausgewählt**). Die Zeitskalen unterscheiden sich in der **Breite pro Zeiteinheit** und in den Zeitstreifen voneinander.



Die Auswahl der Zeitskala ist zur Laufzeit noch veränderbar.

> Anfang und Ende der Zeitskala festlegen

Auf der Eigenschaftenseite **Allgemeines** können Sie den standardmäßigen Anfang der Zeitskala festlegen (**Projektanfang** bzw. **Projektende**). Der Projektanfang wird zur Laufzeit mit der Eigenschaft **TimeScaleStart** oder der Methode **OptimizeTimeScaleStartEnd** an die aktuellen Daten angepasst. Das Datumsformat ist "DD.MM.YYYY;hh:mm:ss".

Hinweis: Das Enddatum ist nicht eingeschlossen. Geben Sie beispielsweise das Enddatum "31.12.2011" an, ist der letzte Tag, der dargestellt wird, der 30.12.2011.

> Zeitskalenabschnitte

Sie können die Zeitskala in verschiedene Zeitskalenabschnitte unterteilen, um bestimmte Planungsabschnitte deutlich voneinander abzuheben, und für jeden Zeitskalenabschnitt verschiedene Zeitstreifen festlegen. Im Dialog **Zeitskala festlegen** können Sie für jeden Zeitskalenabschnitt individuell die **Zeiteinheit** und die **Breite pro Zeiteinheit** festlegen. Außerdem können Sie hier für jeden Zeitskalenabschnitt individuell **Farbe**, **Schriftart** und **3D-Effekt** festlegen, ein **Liniengitter** und ein **Kalendergitter** vereinbaren und festlegen, ob arbeitsfreie Zeiten ausgeblendet werden sollen.

Wenn Sie ein **Liniengitter** vereinbaren, werden in dem entsprechenden Zeitskalenabschnitt einstellbare vertikale Gitternetzlinien dargestellt.

Haben Sie für einen Zeitskalenabschnitt ein **Kalendergitter** vereinbart, werden in diesem Zeitskalenabschnitt die arbeitsfreien Zeiten durch vertikale farbige Flächen besonders markiert.

Vom Dialog **Zeitskala festlegen** gelangen Sie in das Dialogfeld **Zeitskalenabschnitt bearbeiten**, in dem Sie die einzelnen Zeitstreifen und Gitter jedes Zeitskalenabschnitts bearbeiten können.

Zeitskalenabschnitt "Section_1" bearbeiten

Breite pro Zeiteinheit: ☐ Kollabieren arbeitsfreier Zeiten

Zeitstreifen mit 3D-Effekt: ☐

Index	Typ	Position	Minimale Höhe	Hauptmarkierungen	Nebenmarkierungen	Muster	Schrift	Datumsformat	Einheitentrennung
0	Monate	Oben	5.2 mm	1	1		8 pt., Arial	TM YYYY	durchgezogene Linien
1	Tage	Oben	5.2 mm	1	1		8 pt., Arial N...	DD	durchgezogene Linien

Liniengitter

Index	Sichtbar	Typ	Intervall	Linienart	Priorität	Referenzdatum verwenden	Referenzdatum	Am Referenzdatum ausrichten	Sommerzeit be
0	☒	Wochen	1 Wochen		-20	☐		☐	nicht festgele
1	☒	Monate	1 Monate		-10	☐		☐	nicht festgele

Kalendergitter

Index	Sichtbar	Kalender	Muster	Linienart	Priorität	Grafische Attribute der Intervalle verwenden	Identifizierbar	Einrast-Ziel am Start
0	☒	<nicht festgelegt>			-25	☐	☐	☐

OK Abbrechen Hilfe

> Breite pro Zeiteinheit

Die Zeiteinheit ist die kleinste Einheit, in die die Zeitskala unterteilt wird. Sie können Sie im Dialog **Zeitskala festlegen** in der Spalte **Zeiteinheit** auswählen. Möglich sind: Sekunde, Minute, Stunde und Tag.

Die **Breite pro Zeiteinheit** wird in Millimetern/Zeiteinheit angegeben und lässt sich in Schritten von Hundertstel Millimetern/Zeiteinheit verändern. Die minimale Breite beträgt 0,01 mm.

> Zeitstreifen

Zeitstreifen dienen zur Beschriftung der Zeitskala. Jeder Zeitskalenabschnitt kann mehrere Zeitstreifen besitzen (z. B. je einen mit einer Monats- und einer Tagesskala). Sie können für jeden Zeitstreifen die **Position** festlegen, d. h. ob er oberhalb oder unterhalb des Diagramms oder überhaupt nicht dargestellt werden soll. Außerdem können Sie für jeden Zeitstreifen Folgendes festlegen: Typ, Haupt- und Nebenmarkierungen, Mindesthöhe, Farbe, Muster, Schriftart, Datumsformat, Einheitentrennung, Trennstrichposition,

Trennstrichfarbe, Ausrichtung, serielle Beschriftung, Referenzdatum und Kalender.

Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM: Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- MM: zweistellige Monatsnummer: 01-12
- MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)
- YY: zweistellige Jahreszahl
- YYYY: vierstellige Jahreszahl
- WW: zweistellige Nummer der Kalenderwoche: 01-53
- TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- Q: einstellige Quartalsnummer: 1-4
- TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- hh: Stunde zweistellig im 24-Stunden-Format: 00-23
- HH: Stunde zweistellig im 12-Stunden-Format: 01-12
- Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- mm: Minute zweistellig: 00-59
- ss: Sekunde zweistellig: 00-59
- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über

die Regions- und Sprachoptionen definiert

TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

xC/XC: Sie können das Datum als maximal zehngliedrige, einfache Aufwärtszählung ab einem bestimmten Referenzdatum gestalten, z.B. "15:05:07:16:00". Dieses Beispiel datiert 15 Monate, 5 Tage, 7 Stunden, 16 Minuten und 0 Sekunden später als das gesetzte Referenzdatum. Die Notation dazu ist: **xC44:C33:C22:C11:C00**. Es sollen je 2 Stellen für die Glieder 4...0 vorgesehen werden, mit einem vorausgehenden "-" Symbol falls der Wert negativ ist. Die Trennzeichen sind variabel und könnten durch andere ersetzt werden. "x" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, aber kein "+"Symbol, falls er positiv ist. "X" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, und ebenso ein "+"Symbol, falls er positiv ist. Im Dialog **Zeitskalenabschnitt ... bearbeiten** sollten die Felder **Referenzdatum verwenden** und **Hauptmarkierung am Referenzdatum ausrichten** aktiviert sein, ebenso sollte **Serielle Beschriftungen** auf **Keine** gesetzt sein. Das Referenzdatum wird zur Laufzeit über die Methode **VcRibbon.set ReferenceDate** gesetzt und überschreibt die Werte des Dialogs.

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

> Beispiel für die Beschriftung von Zeitstreifen

1. Zeitstreifen: **TW WW-TM-TQ-YYYY**, 2. Zeitstreifen: **TD**

KW37 - September - 3. Quartal - 2003							
Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag	Montag

Die vordefinierten Langtexte können durch benutzerdefinierte Texte ersetzt werden, indem die Eigenschaft **TextEntrySupplyingEventEnabled** auf "True" gesetzt wird. Dann kann auf die folgenden Werte des ControlIndex reagiert werden:

- vcTXERibDay0 bis vcTXERibDay6 (2212 bis 2218)
- vcTXERibCW (2223)

- vcTXERibMon0 bis vcTXERibMon11 (2200 bis 2211)
- vcTXERibQuar0 bis vcTXERibQuar2 (2219 bis 2222)

Code-Beispiel

```
Private Sub VcGantt1_OnSupplyTextEntry(ByVal controlIndex As _
    VcGanttLib.TextEntryIndexEnum, TextEntry As String, _
    returnStatus As Variant)
    Select Case controlIndex
        Case vcTXERibCW
            TextEntry = "Kalenderwoche"
        Case vcTXERibDay0
            TextEntry = "MO"
        ....
        Case vcTXERibMon8
            TextEntry = "Sept."
        Case vcTXERibQuar2
            TextEntry = "3. Quart."
    End Select
End Sub
```

Kalenderwoche 37 - Sept. - 3. Quart. - 2003							
MO	DI	MI	DO	FR	SA	SO	MO

3.33 Zuordnungstabellen

Zuordnungstabellen dienen dazu, ohne den Einsatz von Filtern bestimmte Eigenschaften datenfeldabhängig festzulegen. So können Sie z.B. Hintergrundfarbe, Muster, Musterfarbe und weitere Merkmale von Layern über Zuordnungstabellen datenabhängig gestalten.

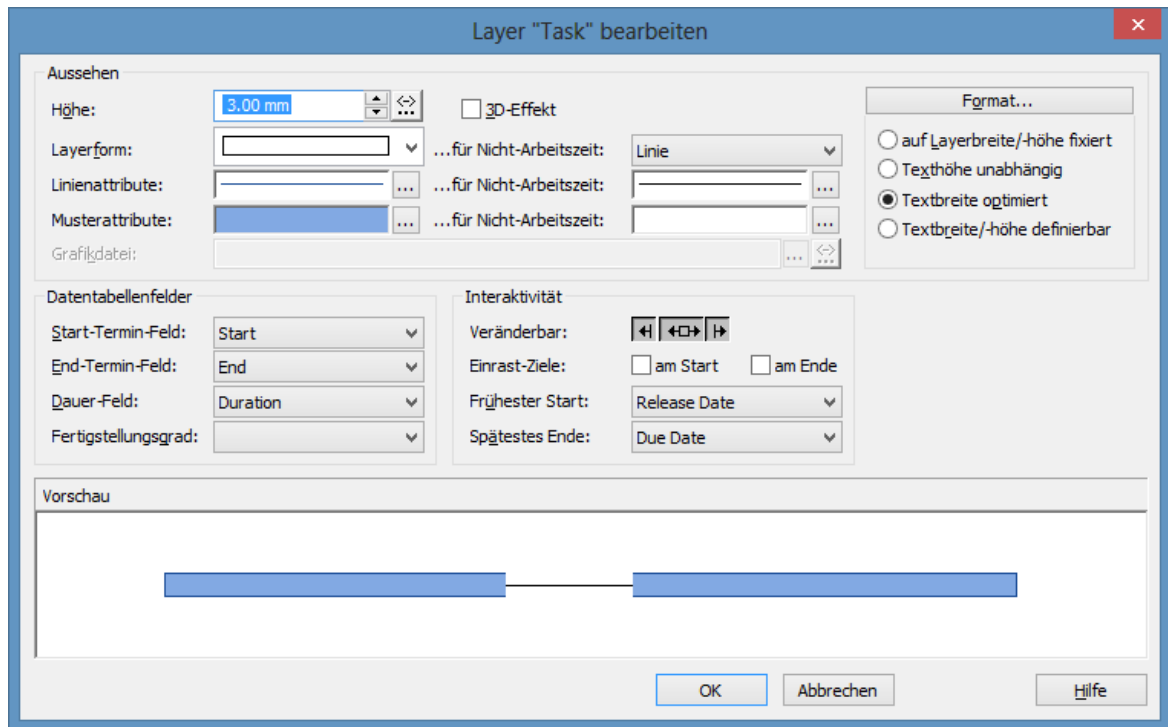
Zuordnungstabellen bestehen aus einer Liste von Zuordnungen. Jede Zuordnung besitzt einen Schlüssel und einen Wert. Der Wert ist je nach Typ der Zuordnungstabelle ein Grafikdateiname, ein Muster etc. Der Schlüssel ist jeweils ein möglicher Eintrag in einem Datenfeld. Zur Laufzeit werden die Schlüssel mit dem tatsächlichen Inhalt des angesprochenen Datenfelds verglichen und bei Gleichheit wird der Wert für die angesprochene grafische Eigenschaft eingesetzt.

Beispiel: dem Schlüssel "A" ist in der Zuordnungstabelle der Wert "grün" zugeordnet. Wenn nun die Zuordnungstabelle verwendet wird und irgendein Knoten den Feldwert "A" besitzt, wird die Farbe "grün" (z.B. als Hintergrundfarbe des Layers) zugewiesen. Als zweiter Wert könnte hier ein Legendentext "Ausführung termingerecht" zugeordnet sein. Grundsätzlich werden also die Inhalte der Datenfelder mit den Schlüsseln der Tabelle verglichen. Wenn sie sich entsprechen, werden die Werte verwendet, die den Schlüsseln zugeordnet sind.

Durch den Einsatz eines Filters statt eines Schlüssels lassen sich auch komplexere Zuordnungen angeben. Dabei werden generell bei der Auswertung einer Zuordnungstabelle zuerst die konkreten Schlüssel und dann bei Nichtzutreffen die Filter ausgewertet.

> Beispiel: Hintergrundfarbe von Layern

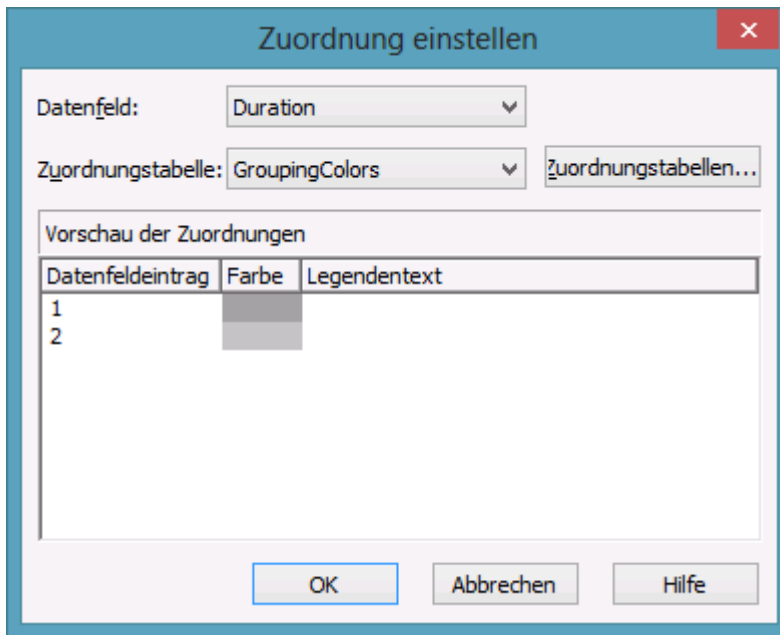
Im folgenden Beispiel wird die Hintergrundfarbe eines Layers über eine Zuordnungstabelle datenabhängig festgelegt werden. Klicken Sie dazu im Dialogfeld **Layer bearbeiten** auf die Schaltfläche neben dem Feld **Hintergrundfarbe** ()



Sie gelangen in den Dialog **Zuordnung einstellen**.

> Zuordnung einstellen

Im Dialogfeld **Zuordnung einstellen** verbinden Sie das Datenfeld eines Knotens mit einer Zuordnungstabelle, sodass die Werte des Datenfeldes und der Schlüssel der Tabelle miteinander verglichen werden können. So wird das gewünschte Attribut, im Beispiel die Hintergrundfarbe des Layers, datenabhängig festgelegt. Soll das Attribut nicht in Abhängigkeit nur eines einzelnen Wertes, sondern eines Wertebereiches oder noch komplexerer Kriterien festgelegt werden, so kann man einen Filter erstellen, den man im Dialog **Zuordnungstabelle bearbeiten** statt eines konkreten Wertes auswählt. Dieser Filter erscheint dann auch im Dialogfeld **Zuordnung einstellen** in der Auflistung der Datenfeldeinträge mit der Zuordnung des entsprechenden Attributes.

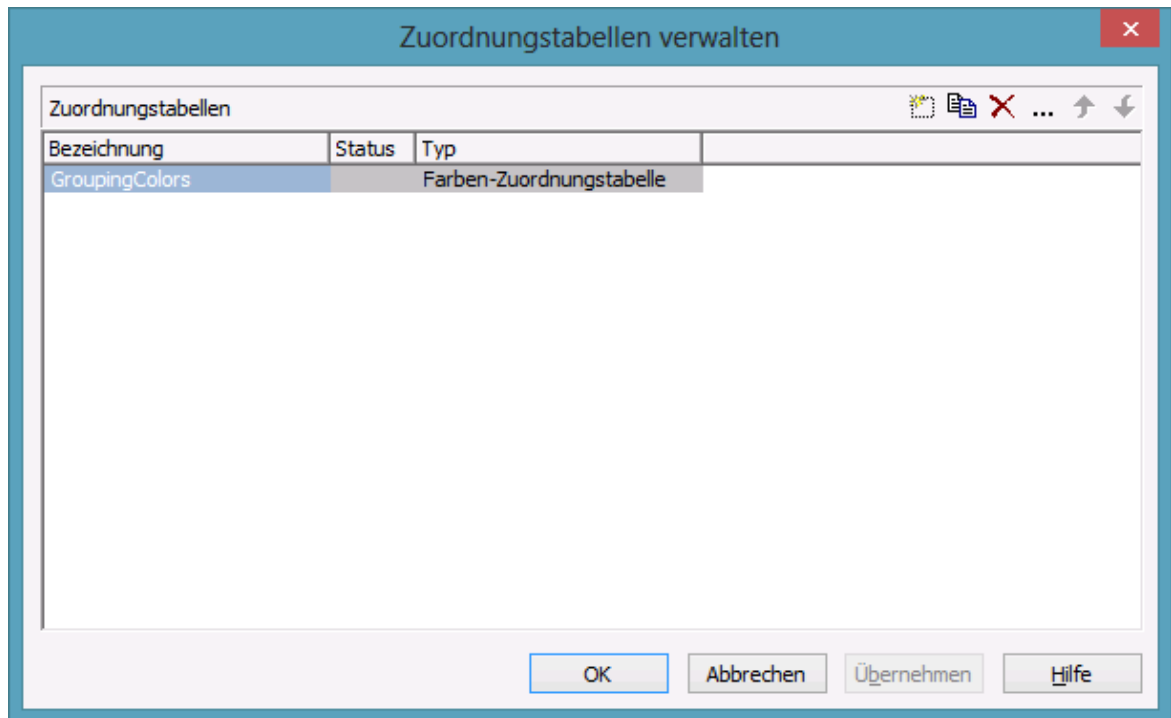


Um eine Zuordnung einzustellen, wählen Sie im obersten Feld des Dialogs das **Datenfeld** des Knotens, dessen Einträge mit den Schlüsseln der Tabelle verglichen werden sollen. Wählen Sie dann im Feld darunter die **Zuordnungstabelle**. (Es werden nur die Zuordnungstabellen angezeigt, die zu dem ausgewählten Attribut passen. Da Sie in unserem Beispiel im Dialog **Layer bearbeiten** die Hintergrundfarbe ausgewählt hatten, werden Ihnen hier auch nur Zuordnungstabellen vom Typ "Farben-Zuordnungstabelle" angeboten.) Nach der Auswahl werden die Werte der Zuordnungstabelle im Vorschauenfenster des Dialogs sichtbar. Falls in Ihrem Beispiel noch keine Zuordnungstabelle vorhanden ist, legen Sie eine nach der Beschreibung des folgenden Kapitels an.

> Zuordnungstabellen verwalten

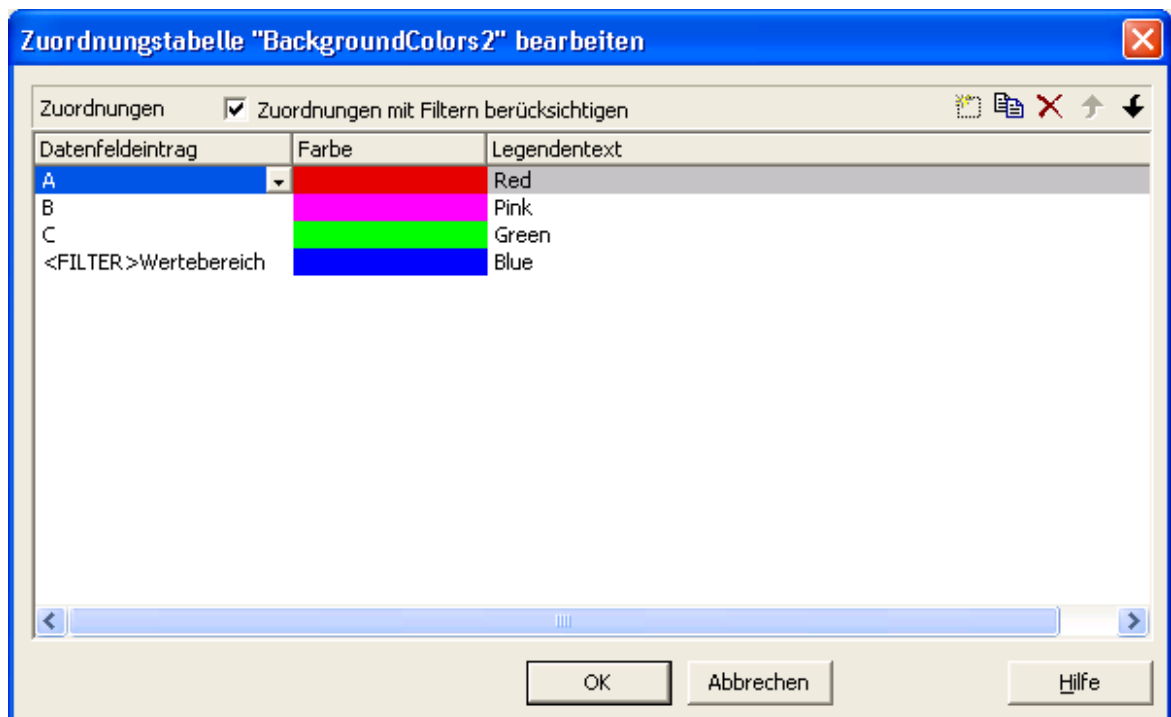
Im Dialogfeld **Zuordnungstabellen verwalten**, das Sie durch Klick auf die Schaltfläche **Zuordnungstabellen** oder über die Schaltfläche **Zuordnungstabellen** auf der Eigenschaftenseite **Objekte** erreichen, können Sie Namen und Typ einer Zuordnungstabelle durch direkte Eingabe verändern sowie über die entsprechenden Schaltflächen oben rechts im Fenster Zuordnungstabellen erstellen, kopieren, löschen oder bearbeiten.

Sie können aus verschiedenen Typen von Zuordnungstabellen auswählen, je nachdem, ob den Datenfeldinhalten Farben, Muster, Grafiken, Schrifttypen, Längen oder Nummern zugeordnet werden sollen.



> Zuordnungstabellen bearbeiten

Um eine Zuordnungstabelle zu bearbeiten, markieren Sie diese in der Tabelle und klicken Sie auf die Schaltfläche  oberhalb der Tabelle. Es erscheint das Dialogfeld **Zuordnungstabelle bearbeiten**.



In der **Zuordnungen**-Tabelle werden für jeden Schlüssel die entsprechenden Werte aufgelistet, in unserem Beispiel sind dies die Hintergrundfarbe und der Legendentext.

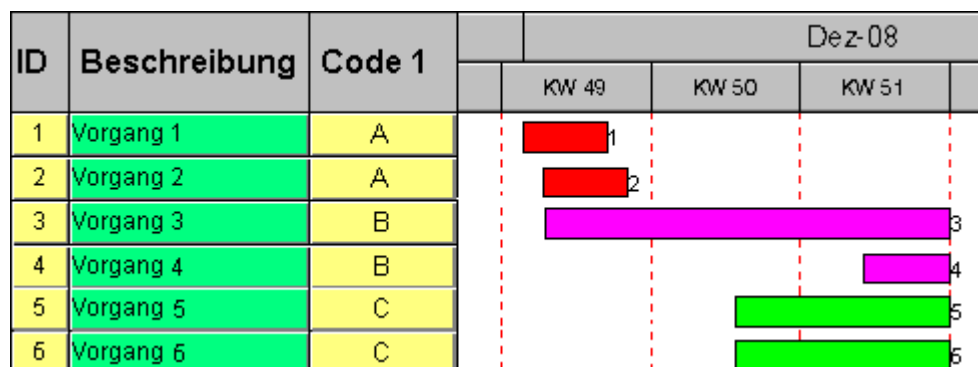
Über die Schaltflächen oben rechts können Sie Schlüssel (Zuordnungen) hinzufügen, kopieren oder löschen oder deren Reihenfolge verändern.

Wenn die Option **Zuordnungen mit Filtern berücksichtigen** ausgewählt ist, werden nicht nur die in der Liste der Datenfeldeinträge angegebenen festen Werte als Schlüssel berücksichtigt, sondern auch Filter, die aus der Dropdown-Liste ausgewählt werden können. Dadurch hängen die Ausführungswerte nicht mehr nur von einem konkreten Wert, sondern von komplexeren Kriterien ab.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie weitere Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

> Beispiel

Bei dem hier definierten Layer werden die Vorgänge mit Datenfeldeintrag "A" in rot dargestellt, die Vorgänge mit Datenfeldeintrag "B" in rosa, etc. Die Standard-Hintergrundfarbe ist grau. Diese wird für Layer ohne Datenfeldeintrag bzw. mit einem Datenfeldeintrag, der nicht in der Zuordnungstabelle aufgeführt ist, verwendet.



Einzelheiten zu den hier beschriebenen Dialogfeldern finden Sie im Kapitel "Eigenschaftenseiten und Dialogfelder".

> Anpassung der Zuordnungstabelle zur Laufzeit

Sie können die Zuordnungstabellen auch zur Laufzeit noch mit Hilfe der VcMap-Methoden anpassen. Damit geben Sie dem Anwender die Möglichkeit, Ihre Voreinstellungen über einen von Ihnen erstellten Dialog zu verändern.

3.34 Verschiebehilfen

4 Eigenschaftenseiten und Dialogfelder

4.1 Allgemeines

Durch die Eigenschaftenseiten kann VARCHART XGantt bereits zur Entwurfszeit konfiguriert werden. Es gibt zwei Möglichkeiten, zu den Eigenschaftenseiten zu gelangen:

- Drücken Sie die rechte Maustaste, wenn der Mauszeiger sich innerhalb des Steuerelements befindet, und wählen Sie im Kontextmenü den Befehl **Eigenschaften** aus.

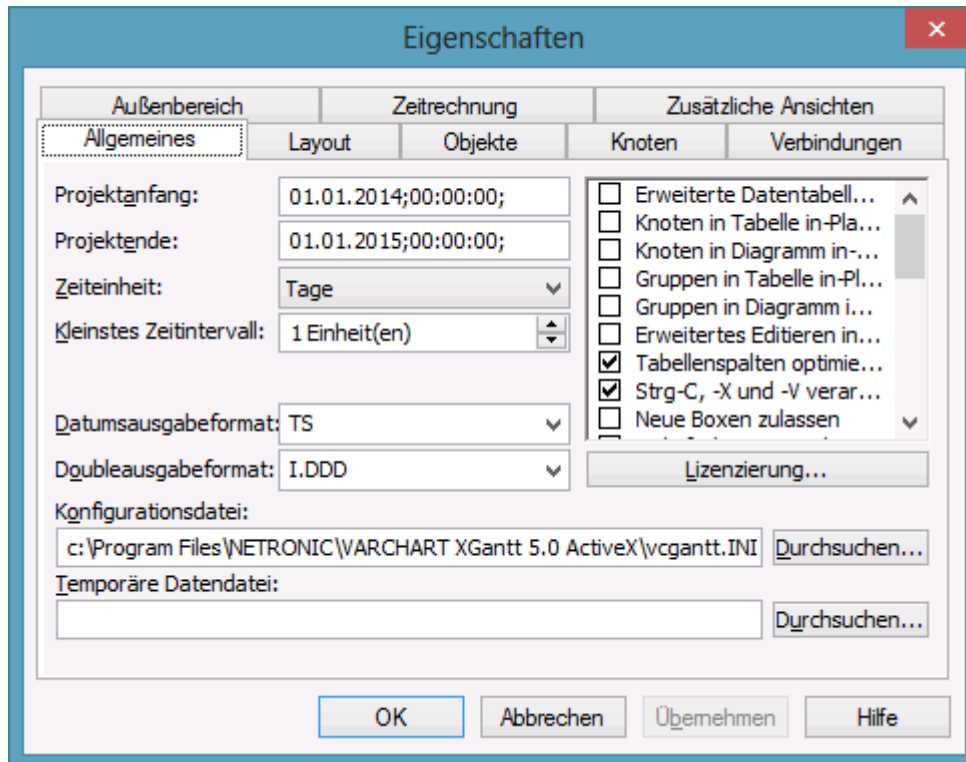
oder

- Im Eigenschaftfenster (kann mit F4 geöffnet werden) klicken Sie in der Symbolleiste auf das ganz rechts stehende Symbol .



Nähere Informationen zu jeder Eigenschaftenseite bzw. jedem Dialogfeld erhalten Sie, indem Sie auf die **Hilfe**-Schaltfläche klicken oder die F1-Taste drücken. Sie erhalten dann direkt die Online-Hilfe zu der Eigenschaftenseite bzw. dem Dialogfeld.

4.2 Eigenschaftenseite "Allgemeines"



Auf dieser Eigenschaftenseite können Sie allgemeine Einstellungen für VARCHART XGantt vornehmen.

Projektanfang

Legen Sie hier den standardmäßigen Anfang der Zeitskala fest. Der Anfang der Zeitskala kann zur Laufzeit mit der Eigenschaft **TimeScaleStart** oder der Methode **OptimizeTimeScaleStartEnd** an die aktuellen Daten angepasst werden. Das Datumsformat ist "DD.MM.YYYY;hh:mm:ss;".

Projektende

Legen Sie hier das standardmäßige Ende der Zeitskala fest. Das Ende der Zeitskala kann zur Laufzeit mit der Eigenschaft **TimeScaleEnd** oder der Methode **OptimizeTimeScaleStartEnd** an die aktuellen Daten angepasst werden. Das Datumsformat ist "DD.MM.YYYY;hh:mm:ss;".

Hinweis: Das angegebene Enddatum ist nicht eingeschlossen. Geben Sie beispielsweise das Enddatum "31.12.2011" an, ist der letzte Tag der Zeitskala der 30.12.2011.

Zeiteinheit

Der hier eingestellte Wert wird für die Berechnung der Dauer (siehe Kapitel "Wichtige Begriffe: Layer") und für das interaktive Verändern und Verschieben Ihrer Knoten in der Darstellung verwendet.

Beispiel: Haben Sie hier die Zeiteinheit "Tage" gewählt, lassen sich Knoten nur in Sprüngen von so vielen ganzen Tagen verschieben, wie unter **Kleinstes Zeitintervall** definiert wurde.

Diese Option kann auch über die Eigenschaft **VcGantt.timeUnit** gesetzt werden.

Kleinstes Zeitintervall

Legen Sie hier fest, wie viele Zeiteinheiten einem Schritt beim interaktiven Verändern eines Knotens entsprechen sollen.

Beispiel: Wenn Sie **Zeiteinheit** auf den Wert "Minuten" und **Kleinstes Zeitintervall** auf den Wert "30" setzen, beträgt das Zeitintervall für die Knotenpositionierung eine halbe Stunde. Der Balken bzw. Layer wird dann also immer zur vollen oder halben Stunde "einrasten".

Diese Option kann auch über die Eigenschaft **VcGantt.TimeUnitsPerStep** festgelegt werden.

Datumsausgabeformat

Wählen Sie aus der Kombobox das Datumsformat aus, in dem Ihre Daten ausgegeben werden sollen, oder definieren Sie Ihr eigenes Datumsformat. Dieses Format gilt auch für alle in VARCHART ActiveX integrierten Dialogfelder, die zur Laufzeit verfügbar sind.

Diese Option kann auch über die Eigenschaft **VcGantt.DateOutputFormat** festgelegt werden.

Für das Datum stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)

TM:	Monatsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
MM:	zweistellige Monatsnummer: 01-12
MMM:	erste drei Buchstaben des Monatsnamens (nicht anpassbar)
YY:	zweistellige Jahreszahl
YYYY:	vierstellige Jahreszahl
WW:	zweistellige Nummer der Kalenderwoche: 01-53
TW:	Text für "Kalenderwoche" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
Q:	einstellige Quartalsnummer: 1-4
TQ:	Quartalsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
hh:	Stunde zweistellig im 24-Stunden-Format: 00-23
HH:	Stunde zweistellig im 12-Stunden-Format: 01-12
Th:	Text für "o' clock" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
mm:	Minute zweistellig: 00-59
ss:	Sekunde zweistellig: 00-59
TS:	kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TL:	langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TT:	Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

Doubleausgabeformat

Markieren Sie in der Kombo-Box das Format, in dem Ihre Daten vom Typ **Double** ausgegeben werden sollen. Sie können wählen zwischen **I** (ganze Zahl), **I,DDD**, **I,DDDDDD** bzw. **I.DDD**, **I.DDDDD** (3 oder 6 Dezimalstellen) und **\$ I,III.DD** bzw. **I,III,DD €** (Währung mit 2 Dezimalstellen).

Diese Option kann auch über die Eigenschaft **VcGantt.DoubleOutputFormat** festgelegt werden.

Konfigurationsdatei

Hier wird der Name der verwendeten Konfigurationsdatei angezeigt. Sie können hier mit Hilfe der Schaltfläche **Durchsuchen** eine andere Konfigurationsdatei auswählen.

Die ini-Dateien können verwendet werden, um die aktuelle Konfiguration zu exportieren oder um eine gespeicherte Konfiguration zu importieren:

Export: Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, in die die aktuellen Einstellungen gespeichert werden sollen. Geben Sie einen nicht vorhandenen Dateinamen an und klicken Sie auf die **Übernehmen**-Schaltfläche. Dann wird diese ini-Datei erzeugt und mit der VARCHART-ActiveX-Instanz verknüpft. Jede Änderung auf einer Eigenschaftenseite wird nun in dieser Datei gespeichert.

Import: Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, aus der die Einstellungen geladen werden sollen. Wenn Sie einen vorhandenen Dateinamen angeben und auf die **Übernehmen**-Schaltfläche klicken, wird die gewählte Konfiguration geladen und von nun an mit der VARCHART-ActiveX-Instanz verknüpft. Die aktuellen Einstellungen gehen dabei verloren.

Die Einstellungen der Konfigurationsdatei werden nur einmal geladen, d. h. VARCHART ActiveX liest sie nicht ein zweites Mal aus derselben Datei. Stattdessen werden die Einstellungen aus dem internen Speicher der Entwicklungsoberfläche geladen, die mit denen der Konfigurationsdatei identisch sind. Daher hätte es keine Auswirkung, wenn Sie die Konfigurationsdatei mit Hilfe eines Editors bearbeiten würden. Falls Sie möchten, dass das VARCHART ActiveX-Steuerelement eine veränderte Konfigurationsdatei verwendet, müssen Sie sie zusätzlich umbenennen und dann diese Konfigurationsdatei importieren.

Temporäre Datendatei

Hier können Sie für die Design-Phase eine Datei mit Vorgangsdaten angeben oder auswählen, um die Einstellungen für die Tabelle und die Layer zu kontrollieren.

Über die **Bearbeiten**-Schaltfläche können Sie das Windows-Dialogfeld **Öffnen** mit der Voreinstellung "Vorgangsdaten (*.bar)" aufrufen.

Diese Einstellung gilt nur für die Design-Phase. Zur Laufzeit muss diese Datei mit der Methode **Open** geöffnet werden.

Erweiterte Datentabellen zulassen

Wenn Sie dieses Kontrollkästchen aktivieren, können Sie mehr (bis zu 90) Datentabellen anlegen und nutzen, als die standardmäßig vorhandenen **Main data** und **Relations**. Diese Option kann auch über die Eigenschaft **VcGantt.ExtendedDataTables** festgelegt werden.

Knoten in Tabelle in-Place-Editieren

Aktivieren Sie dieses Kontrollkästchen, wenn das direkte Editieren von Knotendaten (bei eingeschalteter Gruppierung: von Blattknotendaten) in der Tabelle möglich sein soll. Diese Option kann auch über die Eigenschaft **VcGantt.InPlaceEditingOnNodesInTableEnabled** gesetzt werden.

Wenn einzelne Datenfelder nicht editierbar sein sollen, so darf in der Datendefinition die Option **editierbar** nicht ausgewählt werden.

Knoten in Diagramm in-Place-Editieren

Aktivieren Sie dieses Kontrollkästchen, wenn das direkte Editieren von Knotenlayern (bei eingeschalteter Gruppierung: von Blattknotenlayern) im Diagramm möglich sein soll. Diese Option kann auch über die Eigenschaft **VcGantt.InPlaceEditingOnNodesInDiagramEnabled** gesetzt werden.

Wenn einzelne Datenfelder nicht editierbar sein sollen, so darf in der Datendefinition die Option **editierbar** nicht ausgewählt werden.

Gruppen in Tabelle in-Place-Editieren

Aktivieren Sie dieses Kontrollkästchen, wenn das direkte Editieren von Gruppenknotendaten in der Tabelle möglich sein soll. Voraussetzung ist die Verwendung von eigenen Datentabellen für Gruppendaten.

Diese Option kann auch über die Eigenschaft **VcGantt.InPlaceEditingOnGroupsInTableEnabled** gesetzt werden.

Wenn einzelne Datenfelder nicht editierbar sein sollen, so darf in der Datendefinition die Option **editierbar** nicht ausgewählt werden.

Gruppen in Diagramm in-Place-Editieren

Aktivieren Sie dieses Kontrollkästchen, wenn das direkte Editieren von Gruppenknotenlayern im Diagramm möglich sein soll. Diese Option kann auch über die Eigenschaft **VcGantt.InPlaceEditingOnGroupsInDiagramEnabled** gesetzt werden.

Wenn einzelne Datenfelder nicht editierbar sein sollen, so darf in der Datendefinition die Option **editierbar** nicht ausgewählt werden.

Erweitertes Editieren in Tabelle

Aktivieren Sie dieses Kontrollkästchen, um erweiterte Möglichkeiten zum Bearbeiten des Tabelleninhalts und zum Navigieren nutzen zu können. Diese Option kann auch über die Eigenschaft **VcGantt.ExtendedEditingBehavior** gesetzt werden.

- **Vorgänge markieren und neue Feldinhalte eingeben:**

Beim Klick auf einen Vorgang werden nicht nur die entsprechende Tabellenzeile sowie der zugehörige Knoten im Diagrammbereich markiert, sondern Sie können auch direkt neue Feldinhalte eingeben.

Dabei sollte man folgendes beachten:

Wenn Sie in den **Diagrammbereich** klicken, ist immer das jeweils **erste** Feld der zugehörigen Tabellenzeile markiert und damit im Editiermodus, egal, welches Feld zuvor in der Tabelle markiert war. Beim Klick auf einen anderen Knoten verschiebt sich die Knoten-Markierung entsprechend und es wird wieder das erste Feld der entsprechenden Zeile markiert.

Durch Klick in den Tabellenbereich wird immer das Feld bearbeitet, auf das Sie geklickt haben.

Für beide Verfahren gilt:

Sie können die Vorgangs-Markierung mit den Cursor-Tasten nach oben/unten bzw. der ENTER-Taste verschieben und so die jeweils nächste/vorige Tabellenzeile markieren. Sollte vorher im Tabellenbereich eine andere als die erste Zelle markiert gewesen sein, so bleibt diese Auswahl auch für die neu markierte Zeile erhalten. Innerhalb einer markierten

Tabellenzeile bewegen die Cursortasten nach rechts/links die Markierung jeweils ein Feld weiter/zurück.

Hinweis: Mit Hilfe der Taste ESC werden alle Markierungen aufgehoben.

- **Ändern eines Feldinhaltes**

Zum Ändern eines Feldinhaltes können Sie entweder noch einmal auf das Feld klicken oder die Taste F2 betätigen.

Bei einigen Feldtypen ist dies allerdings gar nicht mehr nötig. Bei Datumsfeldern lässt sich durch Klick auf den nebenstehenden Drop-Down-Pfeil das Datum-Steuererelement anzeigen. Für weitere Informationen zum Gebrauch des Datum-Steuererelementes s. Kap. 4.40 Dialogfeld "Stichtaglinien festlegen".

Bei numerischen Feldern finden sich Pfeil-Schaltflächen, mit deren Hilfe Sie den Wert jeweils erhöhen bzw. verringern können.

Hinweis: Mit Hilfe der Taste ESC verlassen Sie die editierten Felder, ohne die Änderungen zu übernehmen.

- **Einfügen von neuen Tabellenzeilen**

Mit Hilfe der Taste EINGABE lässt sich oberhalb der jeweils markierten Zeile eine neue Tabellenzeile einfügen. Wenn nichts markiert ist, wird die neue Zeile unterhalb der Tabelle eingefügt.

Tabellenspalten optimierbar

Wenn Sie dieses Kontrollkästchen aktivieren, kann der Anwender durch einen Doppelklick auf die Trennlinie zwischen zwei Spalten erreichen, dass die Breite der linken Spalte automatisch an die Länge der darin enthaltenen Texte angepasst wird.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowTableColumn-WidthOptimization** gesetzt werden

Strg-C, -X und -V verarbeiten

Aktivieren Sie dieses Kontrollkästchen, damit die Tastenkombinationen Strg+C, Strg+X und Strg+V automatisch in die Zwischenablage-Operationen **CopyNodesToClipboard**, **CutNodesToClipboard** bzw. **PasteNodesFromClipboard** übersetzt werden. Dieses Verhalten kann abgeschaltet werden, damit in Visual Basic kein Konflikt mit Bearbeitungsmethoden für Menüpunkte entsteht, die dieselben Tastaturkombinationen benutzen. Diese Option kann auch über die Eigenschaft **VcGantt.CtrlCXVProcessing** gesetzt werden.

Neue Boxen zulassen

Wenn Sie dieses Kontrollkästchen aktivieren, können zur Laufzeit neue Boxen angelegt werden. Zum Anlegen neuer Boxen zur Laufzeit muss im Kontextmenü (für das Diagramm) der **Modus: Box erzeugen** gewählt werden oder **InteractionMode** auf **VcCreateBox** gesetzt werden.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowNewBoxes** gesetzt werden.

Mehrfache Box-Markierung zulassen

Wenn Sie dieses Kontrollkästchen aktivieren, können zur Laufzeit mehrere Boxen gleichzeitig durch Anklicken markiert werden, ohne gleichzeitig die STRG-Taste gedrückt halten zu müssen. Diese Option ist standardmäßig deaktiviert.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowMultipleBox-Marking** gesetzt werden.

Kontextmenü für Boxen anzeigen

Aktivieren Sie dieses Kontrollkästchen, um die Anzeige des Kontextmenüs für Boxen zur Laufzeit zu ermöglichen.

Diese Option kann auch über die Eigenschaft **VcGantt.ContextMenuFor-BoxesEnabled** gesetzt werden.

Zeitskalen-Dialog anzeigen

Aktivieren Sie dieses Kontrollkästchen, wenn das Dialogfeld **Zeitskala bearbeiten** bei einem Doppelklick auf die Zeitskala erscheinen soll. Diese Option kann auch über die Eigenschaft **VcGantt.ShowTimeScaleDialog** gesetzt werden.

Zeitskala skalierbar

Nur wenn Sie diese Option wählen, lassen Sie zu, dass der Anwender die Auflösung der Zeitskala interaktiv verändern kann.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowTimeScale-Rescale** gesetzt werden.

Numerische Skala skalierbar

Legen Sie hier fest, ob die numerische Skala des Histogramms skalierbar sein soll.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowNumericScale-Rescale** gesetzt werden.

Zoomen per Mausrad zulassen

Aktivieren Sie dieses Kontrollkästchen, um das Zoomen per Mausrad zuzulassen. Um zu zoomen, muss der Anwender die Strg-Taste festhalten und das Mausrad drehen.

Diese Option kann auch über die Eigenschaft **VcGantt.ZoomingPerMouseWheelAllowed** gesetzt werden.

OnToolTipText-Ereignisse

Aktivieren Sie dieses Kontrollkästchen, um das Ereignis **OnToolTipText** freizuschalten. Mit Hilfe dieses Ereignisses können Sie die Texte der Tooltips, die bei Objekten angezeigt werden sollen, festlegen.

Diese Option kann auch über die Eigenschaft **VcGantt.ShowToolTip** gesetzt werden.

Scroll-Ereignisse

Mithilfe dieses Kontrollkästchens können Sie die Scroll Events an- oder abschalten. Dies kann auch durch die Eigenschaft **VcGantt.ScrollEvents-Enabled** geschehen.

Hinweis: Die Scroll-Events sind standardmäßig **nicht** aktiviert.

OnSupplyTextEntry-Ereignisse

Aktivieren Sie dieses Kontrollkästchen, um das Ereignis **OnSupplyTextEntry** freizuschalten. Mit Hilfe dieses Ereignisses können Sie die Texte aller Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen, die zur Laufzeit erscheinen, verändern, beispielsweise um sie in unterschiedliche Sprachen zu übersetzen.

Diese Option kann auch über die Eigenschaft **VcGantt.EnableSupplyTextEntryEvent** gesetzt werden.

Ereignis-Sicherheitsprüfung

Aktivieren Sie dieses Kontrollkästchen, wenn eine Sicherheitsprüfung der Ereignisse **OnNodeModify** und **OnNodeModifyEx** vorgenommen werden soll. Dann werden in diesen Ereignissen die Zugriffe auf Set-Calls des gleichen Objekttyps unterdrückt.

Diese Option kann auch über die Eigenschaft **VcGantt.EventsSecurityCheck** gesetzt werden.

Automatische Reduzierung der Zeilenhöhen zulassen

Diese Option beeinflusst das Verfahren zur Berechnung der Zeilenhöhe im Diagramm. Ist sie nicht ausgewählt, werden die vertikalen Offsets der Layer ausgehend von einer gedachten Nulllinie in der vertikalen Mitte einer Knotenzeile wirksam. Damit diese Nulllinie stets in der Mitte einer Zeile bleibt, kann so entweder der obere oder der untere Rand einer Zeile sehr groß erscheinen, wobei aber Layer mit vertikalem Offset von 0 immer vertikal zentriert bleiben.

Ist das Kontrollkästchen aktiviert, wird zwar weiterhin die gedachte Nulllinie verwendet, diese befindet sich jedoch nicht mehr zwingend in der Zeilenmitte, sondern wird so positioniert, dass für die Zeilenhöhe das kleinstmögliche Maß verwendet werden kann. Dadurch kann es vorkommen dass Layer mit einem vertikalen Offset von 0 nicht mehr auf gleicher Höhe liegen wie der vertikal zentrierte Text der zugehörigen Tabellenzeile.

Diese Option kann auch über die Eigenschaft **VcGantt.RowHeightReductionEnabled** gesetzt werden.

Anti-Aliasing bei Schriften

Mit dieser Option können Sie das Glätten von Schriftzeichen erlauben oder unterdrücken. Wenn dies bei bestimmten Schriftarten, insbesondere bei nichtlateinischen Zeichen, zu verminderter Lesefähigkeit führt, sollte die Eigenschaft abgeschaltet werden.

Die Glättung per GDI+ bewirkt zusätzlich, dass die Texte bei jeder Zoomstufe die gleiche relative Ausdehnung haben, so dass immer dieselbe Anzahl Zeichen z.B. in ein Tabellenfeld passt. Wenn diese Eigenschaft aber auf **False** steht, dann wird stattdessen die Einstellung des Betriebssystems übernommen (einstellbar in der **Systemsteuerung**, Dialogfeld **Anzeige**, Reiter **Darstellung: Effekte**). Wenn dort eine Kantenglättung eingeschaltet ist, dann werden also Texte weiterhin geglättet. Es kann dann aber sein, dass

bei manchen Zoomstufen mehr Text sichtbar ist als bei anderen, weil die systemeigene Kantenglättung dies nicht garantiert.

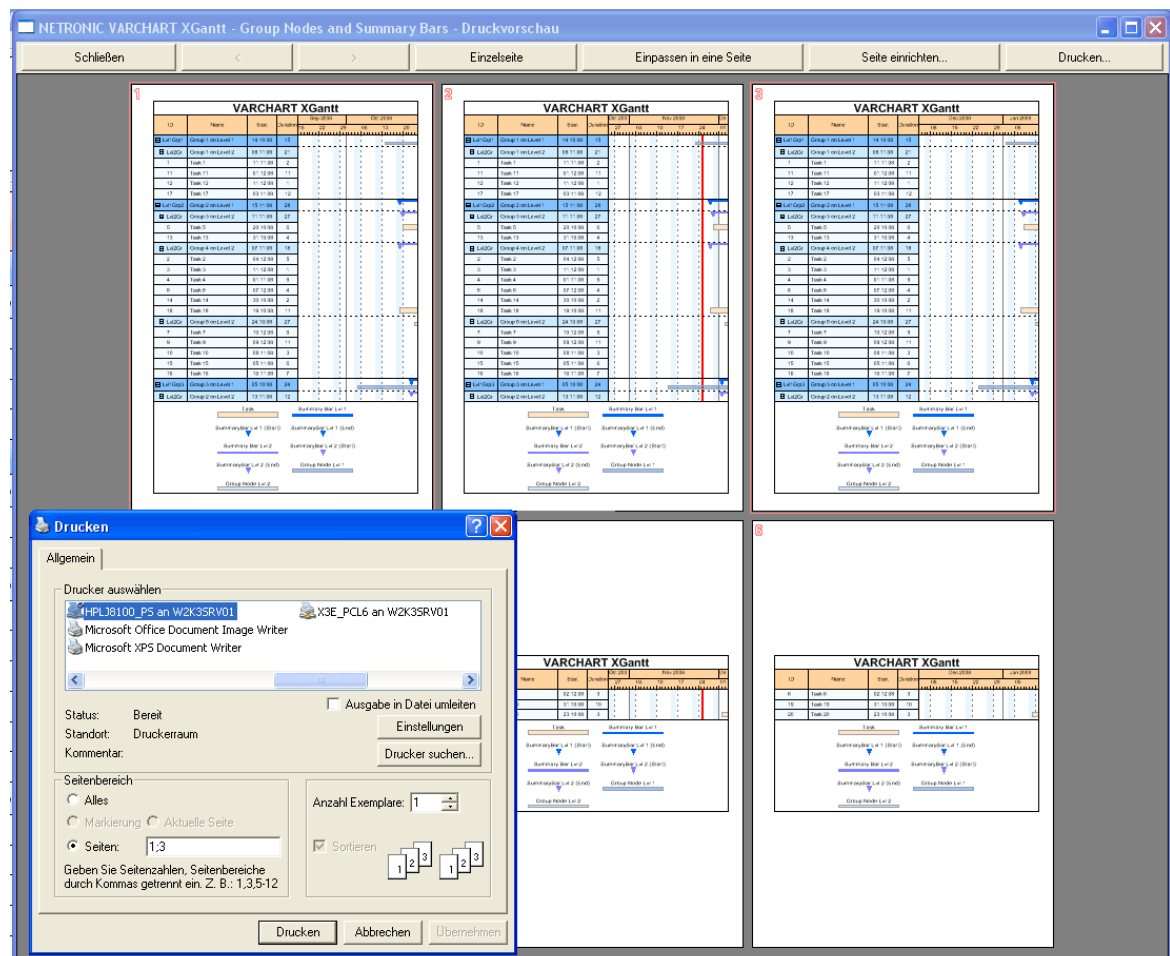
Diese Option kann auch über die Eigenschaft **VcGantt.FontAntiAliasing-Enabled** gesetzt werden

PrintDlgEx Dialog verwenden

Wenn diese Option ausgewählt ist, kann zur Laufzeit der Dialog **Drucker einrichten** weder aus dem Kontextmenü noch aus der Druckvorschau aufgerufen werden, da er sich nun im (erweiterten) **Drucken**-Dialog befindet. Bei neuen Projekten ist die Option standardmäßig aktiviert, bei bestehenden Projekten ist sie aus Gründen der Kompatibilität ausgeschaltet.

In der Druckvorschau können nun einzelne oder mehrere Seiten durch Klick bzw. STRG+Klick ausgewählt werden. Diese Seiten sind dann im Dialog **Drucken** unter **Seitenbereich** bereits voreingestellt.

Wenn man aus der Druckvorschau den **Drucken**-Dialog aufruft, sind alle Blätter mit einer Seitennummer versehen, um so die Seitenauswahl zu erleichtern.



Diese Option kann **nicht** an der API gesetzt werden.

Abgerundete Schrägen bei Verbindungen verwenden

Aktivieren Sie dieses Kontrollkästchen, damit Schrägen bei Verbindungen vom Routing-Typ **vcLRTOrthogonalDistinguishable** als Viertelkreise statt als gerade Linien dargestellt werden. Diese Option kann auch zur Laufzeit über die VcGantt-Eigenschaft **RoundedLinkSlantsEnabled** gesetzt werden.

Optimierung von Gruppen bei Interaktionen

Ist diese Eigenschaft eingeschaltet, dann werden bei Interaktionen wie Aufziehen von Knoten, Knotenverschieben horizontal oder vertikal und Ändern des Start- oder Endtermins eines Layers die Knoten der gesamten Zielgruppe automatisch erneut optimiert, wenn sie bisher bereits optimiert dargestellt wird. Wird diese Eigenschaft abgeschaltet, dann wird bei denselben Interaktionen der Knoten unter der Cursorposition platziert, wenn dies nicht zu einer Überlappung führt. Bei einer möglichen Überlappung wird der Knoten in die nächste Zeile zu anderen Knoten platziert, wenn dies dort keine Überlappung verursacht. Wenn doch, dann bekommt der Knoten eine eigene Zeile unter der, an der der Mauscursor sich befindet.

Diese Option kann auch über die Eigenschaft **VcGantt.GroupOptimization-OnInteractionsEnabled** gesetzt werden.

Verbindungstyp berücksichtigen beim Knotenverschieben

Wählen Sie diese Option, werden beim Ziehen von Knoten und eingeschalteten Verbindungen die Phantomlinien, die die Verbindungen zu verschobenen Knoten und deren Anschlussknoten dargestellt werden, typgerecht dargestellt, d.h. sie beginnen/enden nicht in der Mitte der Knotenphantome, sondern auf deren linken oder rechten Seite.

Diese Option kann auch über die Eigenschaft **VcGantt.ConsiderLink-RelationTypesOnNodeDragging** gesetzt werden.

Wartecursor bei zeitkritischen Operationen eingeschaltet

Wählen Sie diese Option, wenn bei zeitkritischen Aufrufen (wie ScheduleProject) von uns intern ein Wartecursor gesetzt werden soll.

Die Option kann auch über die Eigenschaft **VcGantt.WaitCursorEnabled** gesetzt werden.

Panning-Modus zulassen

Diese Option muss aktiviert werden, damit Sie zur Laufzeit einen bestimmten Bildschirmausschnitt verschieben können. Im Kontextmenü des Diagramms erscheint dann der zusätzliche Eintrag **Verschiebe-Modus**.

Der Verschiebemode gilt standardmäßig für **alle** grafischen Grundelemente. Mithilfe der Eigenschaft **VcGantt.VcViewComponent** kann die Verwendung auf einzelne Elemente beschränkt werden.

Die Option kann auch über die Eigenschaft **VcGantt.AllowPanningMode** gesetzt werden.

Selektion mit Gummirechteck erlaubt

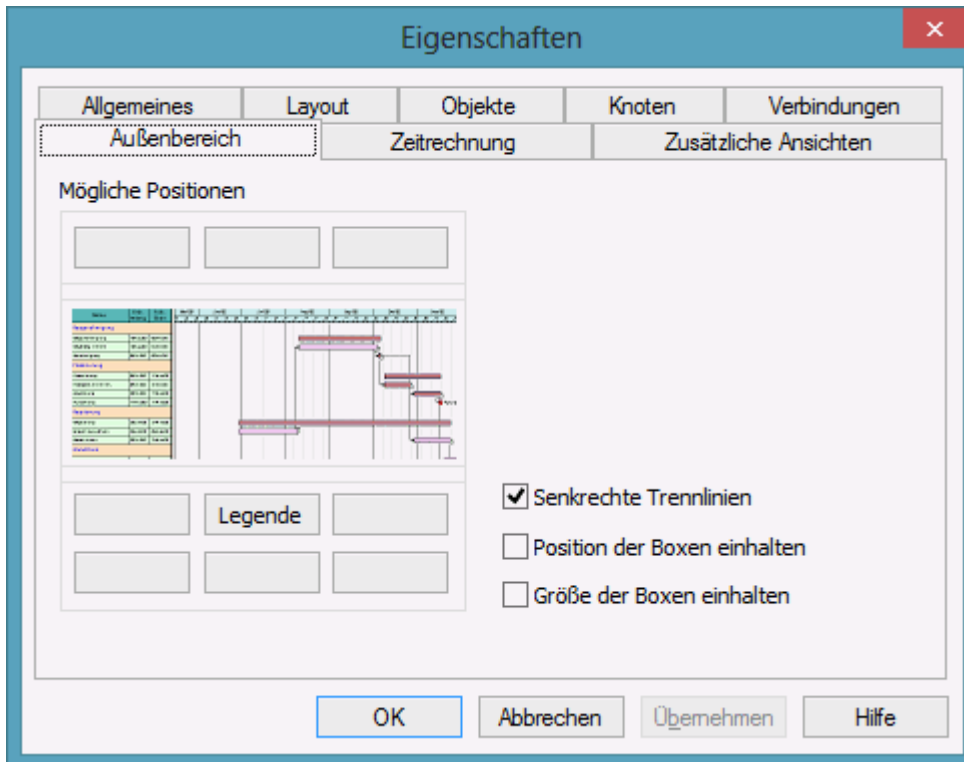
Wählen Sie diese Option, lässt sich die Mehrfachselektion von Knoten mithilfe eines Gummirechtecks aktivieren bzw. unterbinden.

Die Option kann auch über die Eigenschaft **VcGantt.AllowSelectionViaRubberRect** gesetzt werden.

Lizenzierung

Über diese Schaltfläche gelangen Sie in den Dialog **Lizenzierung**. Weitere Informationen zur Lizenzierung finden Sie in Kapitel **Dialogfeld "Lizenzierung"**

4.3 Eigenschaftenseite "Außenbereich"



Mögliche Positionen

Oberhalb der Grafik stehen Ihnen drei und unterhalb der Grafik sechs Bereiche zur Verfügung, in denen Sie Texte, Grafiken oder eine Legende platzieren können. Jeder dieser Bereiche wird in diesem Dialog durch je eine Schaltfläche repräsentiert. Alle diese Bereiche werden nur in der Seitenansicht und im Ausdruck angezeigt. Klicken Sie auf eine der Schaltflächen ober- bzw. unterhalb der Grafik, um den Dialog **Texte, Grafiken und Legende festlegen** zu öffnen.

Senkrechte Trennlinien

Aktivieren Sie dieses Kontrollkästchen, wenn die Bereiche für Texte, Grafiken oder Legende durch senkrechte Trennlinien getrennt werden sollen.

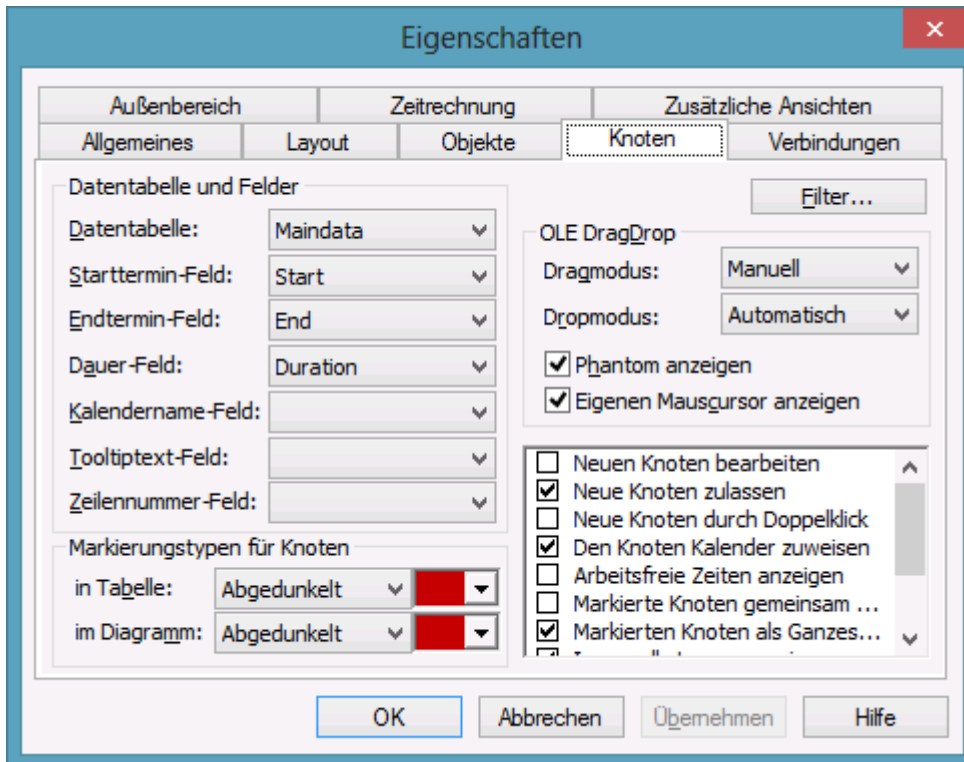
Position der Boxen einhalten

Aktivieren Sie dieses Kontrollkästchen, wenn die Position der Boxen möglichst genau eingehalten werden soll. Andernfalls wird der vorhandene Platz proportional auf die in der jeweiligen Zeile vorhandenen Elemente verteilt.

Größe der Boxen einhalten

Aktivieren Sie dieses Kontrollkästchen, wenn die Größe der Boxen möglichst genau eingehalten werden soll. Gegebenenfalls wird das Diagramm vergrößert und/oder die Texte in den Boxen abgeschnitten.

4.4 Eigenschaftenseite "Knoten"



Datentabelle

Wählen Sie hier die Datentabelle aus, die für die Darstellung der Knoten herangezogen werden soll.

Diese Option kann auch über die Eigenschaft **VcGantt.NodesDataTable-Name** festgelegt werden.

Starttermin-Feld

Vereinbaren Sie hier das Datenfeld, in das der Anfangstermin eines interaktiv angelegten Vorgangs geschrieben werden soll. Es werden nur Datumsfelder angeboten.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeStartDateData-FieldIndex** festgelegt werden.

Endtermin-Feld

Vereinbaren Sie hier das Datenfeld, in das der Endtermin eines interaktiv angelegten Vorgangs geschrieben werden soll. Es werden nur Datumsfelder angeboten.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeEndDateDataFieldIndex** festgelegt werden.

Dauer-Feld

Vereinbaren Sie hier das Datenfeld, in das die Dauer eines interaktiv angelegten Vorgangs geschrieben werden soll. Es werden nur numerische Felder angeboten.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeDurationDataFieldIndex** festgelegt werden.

Kalendername-Feld

Wenn individuelle Kalender für Knoten verwendet werden sollen, können Sie hier das Datenfeld auswählen, das den Namen des für einen Knoten zu verwendenden Kalenders enthalten soll.

Dazu muss die Option **Den Knoten Kalender zuweisen** aktiviert sein. Außerdem müssen die Kalender vor dem Laden der Vorgänge erzeugt worden sein.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeCalendarNameDataFieldIndex** festgelegt werden..

Tooltiptext-Feld

Das Datenfeld, das Sie hier auswählen, wird als Tooltip angezeigt, wenn Sie eine VMF-Datei mit dem WebViewer ansehen und dort auf einen Knoten rechtsklicken. Es sind keine weiteren Einstellungen notwendig.

Das VMF- (Viewer Metafile) Format ist ein Vektorformat, in denen ein Diagramm auflösungsunabhängig gespeichert werden kann. Es kann mittels des als Java Applet ausgeführten GRANEDA WebViewers plattformunabhängig mit jedem Java-fähigen Internet-Browser angesehen werden.

Um in Ihrer Applikation in dem VARCHART ActiveX Tooltips anzuzeigen, müssen Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **OnToolTipText-Ereignisse** aktivieren bzw. die Eigenschaft **ShowToolTip** auf **True** setzen und im **OnToolTipText**-Ereignis programmieren, welche Datenfelder angezeigt werden sollen.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeToolTipTextField** gesetzt werden.

Zeilennummer-Feld

Wählen Sie hier das Datenfeld aus, in das die Zeilennummer jedes Vorgangs geschrieben werden soll. Damit Änderungen wirksam werden, muss eine Aktualisierung über die Methode **VcGantt.UpdateRowNumberFields** vorgenommen werden.

Diese Option kann auch über die Eigenschaft **VcGantt.NodeRowNumber-DataFieldIndex** festgelegt werden.

Markierungstypen für Knoten in Tabelle

Wählen Sie in dem linken Feld aus, ob und welche Art der interaktiven Markierung von Knoten in der Tabelle dem Anwender geboten werden soll. Folgende Alternativen stehen zur Verfügung:

- Ohne
- Einrahmen innen
- Invertieren
- Abgedunkelt (um 25 %)
- Aufgehellert (um 25%)
- Pickmarks innen

In dem rechten Feld können Sie eine Farbe für die Markierung setzen.

Markierungstypen für Knoten im Diagramm

Wählen Sie in dem linken Feld aus, ob und welche Art der interaktiven Markierung von Knoten im Diagramm dem Anwender geboten werden soll. Folgende Alternativen stehen zur Verfügung:

- Ohne
- Einrahmen
- Einrahmen innen
- Invertieren
- Abgedunkelt (um 25 %)
- Aufgehellert (um 25%)
- Pickmarks
- Pickmarks innen

In dem rechten Feld können Sie eine Farbe für die Markierung setzen.

Filter

Über diese Schaltfläche öffnen Sie das Dialogfeld **Filter verwalten**. Der für die Vorselektion der Knoten wirksame Filter kann ausschließlich zur Laufzeit über die Eigenschaft **ActiveNodeFilter** des Objekts **VcGantt** eingestellt werden.

Dragmodus

Mit dem OLE-Dragmodus können Sie festlegen, ob das Ziehen eines Knotens über die Grenze des VARCHART-XGantt-Steuerelements hinaus erlaubt sein soll. Mögliche Alternativen sind:

- **Manuell:** Bei diesem Modus müssen Sie die Methode **OLEDrag** aufrufen, um das Ziehen eines Knotens zu starten.
- **Automatisch:** Das Ziehen eines Knotens über die Grenzen des VARCHART-XGantt-Steuerelements wird automatisch gestartet.

Beim Start des Vorgangs füllt die Quellkomponente das **DataObject** mit den Daten des gezogenen Knotens und setzt den Effekt-Parameter, um damit das OLEStartDrag-Ereignis sowie andere quellenseitige OLE Drag & Drop-Ereignisse auszulösen. Dies gibt Ihnen die Kontrolle über die Drag&Drop-Operation und erlaubt Ihnen, einzugreifen, z. B. um andere Datenformate hinzuzufügen.

VARCHART XGantt verpackt die Daten u.a. in das Standard-Zwischenablage-Format CF_TEXT (für Visual-Basic-Benutzer: das vbCFText-Format), das mühelos gelesen werden kann.

Während des Ziehens kann der Benutzer mit Hilfe der Strg-Taste festlegen, ob das Objekt verschoben oder kopiert werden soll.

Das OLE Drag & Drop-Verhalten des VARCHART XGantt ist kompatibel zu dem in Visual Basic üblichen, d. h. die Methoden, Eigenschaften und Ereignisse tragen dieselben Namen und haben dieselbe Bedeutung wie bei den Standardobjekten aus Visual Basic.

Diese Option kann auch über die Eigenschaft **VcGantt.OLEDragMode** festgelegt werden.

Dropmodus

Mit dem OLE-Dropmodus können Sie festlegen, ob ein Knoten aus einer anderen VARCHART-XGantt-Komponente in die aktuelle Komponente herein gezogen werden darf.

Mögliche Alternativen sind:

- **Kein:** Knoten aus einer anderen VARCHART-XGantt-Komponente können nicht in die aktuelle Komponente herein gezogen werden.
- **Manuell:** Sie erhalten beim Dropping das Ereignis **OLEDragDrop**, so dass Sie die übertragenen Daten selbst weiterverarbeiten können, um z. B. einen Knoten zu erzeugen oder eine Datei einzulesen. Wenn Quell- und Zielkomponente identisch sind, erhalten Sie wie bei abgeschaltetem OLE Drag&Drop eins der Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**.
- **Automatisch:** Der Dropping-Vorgang wird von der Komponente selbst verarbeitet, d. h. es wird, falls möglich, ein Knoten an entsprechender Mausposition erzeugt.

Diese Option kann auch über die Eigenschaft **VcGantt.OLEDropMode** festgelegt werden.

Phantom anzeigen

Mit Hilfe dieses Kontrollkästchens können Sie festlegen, ob während eines OLE-Drag-Vorgangs ein Phantom erscheinen soll oder nicht. Das Abschalten des Phantoms ist für Anwendungen gedacht, die beim Hineindraggen eines Objekts kein neues Objekt erzeugen, sondern beispielsweise den Knoten, auf dem dann ein Objekt fallen gelassen wird, nur neu attributieren.

Diese Option kann auch über die Eigenschaft **VcGantt.OLEDragWithPhantom** festgelegt werden.

Eigenen Mauscursor anzeigen

Mit Hilfe dieses Kontrollkästchens können Sie festlegen, ob während eines OLE-Drag-Vorgangs der Mauscursor in der Zielkomponente gesetzt werden soll. Bei OLE Drag & Drop ist es möglich, den Cursor in der Quellkomponente über das Ereignis **OLEGiveFeedback** zu setzen. Daher würde ein Setzen durch die Zielkomponente zu einem Flimmern der konkurrierenden Cursor führen. Über dieses Kontrollkästchen kann man dieses Verhalten beeinflussen. Ausserdem können bei eingeschaltetem Mauscursor und bei der auf **vcOLEDropManual** gesetzten Eigenschaft Objekte außerhalb der Anlagerungsstellen eines Knotens nicht fallen gelassen werden, während dies bei eingeschaltetem Mauscursor möglich ist.

Diese Option kann auch über die Eigenschaft **VcGantt.OLEDragWithOwn-MouseCursor** festgelegt werden.

Neuen Knoten bearbeiten

Wenn Sie diese Option wählen, öffnet sich das Dialogfeld **Vorgänge bearbeiten** automatisch, wenn der Anwender interaktiv einen neuen Knoten erzeugt hat.

Nachdem ein Knoten erzeugt wurde, kann das Dialogfeld **Vorgänge bearbeiten** - auch wenn die Option deaktiviert ist - jederzeit durch einen Doppelklick auf den Knoten oder über sein Kontextmenü geöffnet werden.

Diese Option kann auch über die Eigenschaft **VcGantt.EditNewNode** festgelegt werden.

Neue Knoten zulassen

Nur wenn Sie diese Option wählen, lassen Sie zu, dass der Anwender in einem geöffneten Projekt interaktiv neue Knoten erzeugen kann. Neue Knoten lassen sich entweder im Modus **Knoten erzeugen** interaktiv erzeugen oder durch Doppelklick auf die gewünschte Position, wenn die Option **Neue Knoten durch Doppelklick** angeschaltet ist.

Diese Option kann auch über die Eigenschaft **VcGantt.AllowNewNodes** festgelegt werden.

Neue Knoten durch Doppelklick

Hier können Sie festlegen, dass neue Knoten per Doppelklick erzeugt werden können. Ein per Doppelklick erzeugter neuer Knoten wird auf der aktuellen Cursor-Position eingefügt und hat die Dauer von einer Zeiteinheit.

Diese Option kann auch über die Eigenschaft **VcGantt.NewNodesViaDoubleClick** festgelegt werden.

Den Knoten Kalender zuweisen

Wenn den Knoten Kalender zugewiesen werden sollen, aktivieren Sie dieses Kontrollkästchen. Die Kalenderzuweisung wirkt sich zum einen beim Verschieben von Vorgängen aus: Anfang und Ende der Vorgänge werden nicht auf arbeitsfreie Tage gelegt. Zum anderen werden beim Berechnen der Dauer von Vorgängen die arbeitsfreien Zeiten berücksichtigt. Auch bei der Zeitrechnung wirkt sich die Kalenderzuweisung aus. Derzeit ist standardmäßig ein Fünf-Tage-Kalender definiert ("WeekCalendar").

Diese Option kann auch über die Eigenschaft **VcGantt.Assign CalendarToNodes** festgelegt werden.

Ist kein individueller Kalender pro Knoten zugewiesen worden, dann wird der in der CalendarCollection als aktiv definierte Kalender verwendet.

Arbeitsfreie Zeiten anzeigen

Aktivieren Sie dieses Kontrollkästchen, wenn arbeitsfreie Zeiten hervorgehoben werden sollen. Sie werden dann in der Form angezeigt, die im **Layer bearbeiten**-Dialog ausgewählt wurde.

Diese Option kann auch über die Eigenschaft **VcGantt.ShowNonWorkInterval** festgelegt werden.

Markierte Knoten gemeinsam verschieben

Wählen Sie diese Option, damit mehrere markierte Knoten gemeinsam verschoben werden können. Bei Deaktivierung können nur einzelne Layer oder Knoten (abhängig davon, ob **Markierten Knoten als Ganzes verschieben** aktiviert wurde) mit der Maus bewegt werden, selbst wenn mehrere Knoten markiert sind.

Diese Option kann auch über die Eigenschaft **VcGantt.MoveAllSelectedNodes** festgelegt werden.

Markierten Knoten als Ganzes verschieben

Aktivieren Sie dieses Kontrollkästchen, damit alle Layer eines markierten Knotens zusammen verschoben werden können. Ein Knoten kann durch einen Klick mit der linken Maustaste auf einen seiner Layer markiert werden.

Wenn dieses Kontrollkästchen nicht aktiviert ist, kann jeder Layer eines markierten Knotens nur einzeln verschoben werden. Sollen trotzdem alle Layer eines markierten Knotens gemeinsam verschoben werden, muss beim Verschieben die Umschalt-Taste gedrückt werden. (Voraussetzung: Die Option **Layer bei gedrückter Shift-Taste als Knoten verschieben** wurde ausgewählt).

Diese Option kann auch über die Eigenschaft **VcGantt.MoveNodeWhenMarked** festgelegt werden.

Immer alle Layer gemeinsam verschieben

Wenn dieses Kontrollkästchen aktiviert ist, können immer alle Layer eines Knotens gemeinsam verschoben werden, ohne dass sie zuvor markiert werden müssen.

Diese Option kann auch über die Eigenschaft **VcGantt.MoveNodeAlways** festgelegt werden.

Layer bei gedrückter Shift-Taste als Knoten verschieben

Wenn Sie diese Option auswählen, können alle Layer eines markierten Knotens gemeinsam verschoben werden, indem während des Verschiebens die Umschalt-Taste gedrückt wird.

Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.MoveLayersAsNodeWithShiftKey** gesetzt werden.

Einrastziele bei Interaktionen

Wenn Sie diese Option auswählen, kann beim Verschieben von Knoten/Layern die Einrastfunktionalität genutzt werden, d.h. es wird festgelegt, ob ein Knoten/Layer beim Verschieben an den jeweils definierten Einrastzielen der Objekte einrasten soll oder nicht.

Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.UseSnapTargetsInInteractions** gesetzt werden.

Einrastlinien anzeigen

Wenn Sie diese Option auswählen, können beim Verschieben von Knoten/Layern mit aktivierter Einrastfunktionalität entsprechende Linien gezeigt werden, damit der/die definierten Einrastziele besser zu erkennen sind.

Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.ShowSnapLines** gesetzt werden.

Einrastmarkierungen anzeigen

Wenn Sie diese Option auswählen, können beim Verschieben von Knoten/Layern bei aktivierter Einrastfunktionalität am Zielknoten/-layer entsprechende Markierungen angezeigt werden, damit der/die definierten Einrastziele besser zu erkennen sind.

Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.ShowSnapMarkings** gesetzt werden.

Verändern der Knotenreihenfolge über das Diagramm erlaubt

Wenn Sie diese Option auswählen, können Sie im Diagrammbereich Knoten mit der Maus von einer Zeile in eine andere verschieben und damit ihre Reihenfolge oder Gruppenzugehörigkeit ändern.

Wenn ein Knoten aus mehreren Layern besteht, muss beim vertikalen Verschieben zusätzlich die Umschalt-Taste gedrückt werden.

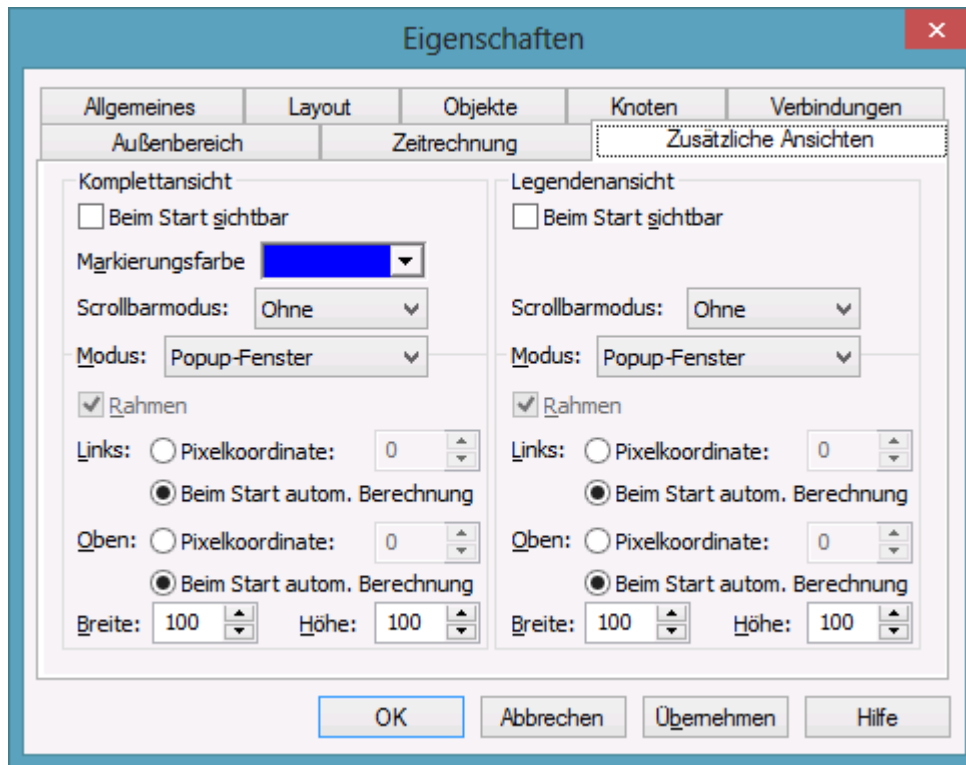
Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.AllowVerticalNodeMovement** gesetzt werden.

Verändern der Knotenreihenfolge über die Tabelle erlaubt

Wenn Sie diese Option auswählen, können Sie im Tabellenbereich Knoten mit der Maus von einer Zeile in eine andere verschieben und damit ihre Reihenfolge oder Gruppenzugehörigkeit ändern. Wenn ein Knoten aus mehreren Layern besteht, muss beim vertikalen Verschieben zusätzlich die Umschalt-Taste gedrückt werden.

Diese Option kann auch zur Laufzeit über die Eigenschaft **VcGantt.AllowVerticalNodeMovementViaTable** gesetzt werden.

4.5 Eigenschaftenseite "Zusätzliche Ansichten"



Auf dieser Eigenschaftenseite können Sie die Eigenschaften der Komplettansicht (World View) sowie der Legendenansicht (Legend View) festlegen.

Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm und/oder das Histogramm angezeigt wird. Der eine Rahmen darin zeigt an, welcher Ausschnitt des Gantt-Graphen gerade im Hauptfenster dargestellt wird, der andere, welcher Ausschnitt des Histogramms.

Mithilfe der Legendenansicht lässt sich, ebenfalls in einem zusätzlichen Fenster, eine Legende auf dem Bildschirm darstellen.

Um die Ansichten anzeigen zu lassen, wählen Sie zur Laufzeit im Standard-Kontextmenü für das Diagramm oder für das Histogramm **Komplettansicht anzeigen** bzw. für die Legende **Legendenansicht anzeigen**. Über diese Menüpunkte können die Ansichten auch wieder ausgeschaltet werden (alternativ über die **Schließen**-Schaltfläche in der Titelleiste des jeweiligen Fensters).

im Folgenden sind die möglichen Einstellungen für beide Ansichten gleichzeitig beschrieben. Sollte ein Kriterium nicht für beide Ansichten gelten, so wird gesondert darauf hingewiesen.

Beim Start sichtbar

Aktivieren Sie dieses Kontrollkästchen, damit die Ansicht beim Start des Programms sichtbar ist.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Visible** bzw. **VcLegendView.Visible** der Programmierschnittstelle gesetzt werden.

Markierungsfarbe (nur für Komplettansicht)

Wählen Sie hier die Farbe der Linie des Rechtecks aus, das in der Komplettansicht den aktuell gewählten Ausschnitt anzeigt.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.MarkingColor** bzw. **VcLegendView.MarkingColor** der Programmierschnittstelle gesetzt werden.

Scrollbarmodus

Wählen Sie hier, ob und welche Bildlaufleisten in der Ansicht dargestellt werden sollen. Durch die Verwendung von Bildlaufleisten werden Leerbereiche vermieden und das Diagramm bzw. die Legende ist besser zu erkennen, weil es bzw. sie größer dargestellt wird. Folgende Möglichkeiten stehen zur Verfügung:

- **Ohne:** In der Ansicht wird immer alles vollständig dargestellt. Dadurch können Leerbereiche entstehen, wenn die Ansicht in ihren Proportionen nicht denen des Charts(oder der Legende) entspricht.
- **Horizontal:** Es wird, wenn notwendig, eine horizontale Bildlaufleiste dargestellt.
- **Vertikal:** Es wird, wenn notwendig, eine vertikale Bildlaufleiste dargestellt.
- **Automatisch:** Es wird, wenn notwendig, eine horizontale oder eine vertikale Bildlaufleiste dargestellt.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.ScrollBarMode** bzw. **VcLegendView.ScrollBarMode** der Programmierschnittstelle gesetzt werden.

Modus

Wählen Sie hier den Modus für die Ansicht aus. Es gibt folgende Möglichkeiten:

- **fest an linker Seite:** Die Ansicht wird links im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
- **fest an rechter Seite:** Die Ansicht wird rechts im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
- **fest an oberer Seite:** Die Ansicht wird oben im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
- **fest an unterer Seite:** Die Ansicht wird unten im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
- **nicht fest positioniert:** Die Ansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des VARCHART ActiveX und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft **VcWorldView.ParentHWnd** geändert werden.
- **Popup-Fenster:** Die Ansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Es kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die **Schließen**-Schaltfläche in der Titelleiste ausgeschaltet werden.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Mode** bzw. **VcLegendView.Mode** der Programmierschnittstelle gesetzt werden.

Rahmen

*Nicht aktiviert, wenn der Modus **Popup-Fenster** gewählt wurde.* Aktivieren Sie dieses Kontrollkästchen, wenn die Ansicht einen Rahmen erhalten soll. Die Rahmenfarbe kann aus der Drop-Down-Liste gewählt werden.

Diese Optionen können auch über die Aufrufe **VcWorldView.Border** und **VcWorldView.Border.Color** bzw. **VcLegendView.Border** und **VcLegendView.Border.Color** der Programmierschnittstelle gesetzt werden.

Links

*Nur aktiviert, wenn der Modus **nicht fest positioniert** oder **Popup-Fenster** gewählt wurde.* Legen Sie hier die linke Position der Ansicht fest. Dabei gibt es zwei Möglichkeiten:

1. Geben Sie unter **Pixelkoordinate** einen Wert an. Dabei handelt es sich um Gerätekoordinaten.
2. Wählen Sie die Option **Beim Start automat. Berechnung**, damit die Position der Ansicht beim Programmstart automatisch berechnet wird.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Left** bzw. **VcLegendView.Left** der Programmierschnittstelle gesetzt werden.

Oben

*Nur aktiviert, wenn der Modus **nicht fest positioniert** oder **Popup-Fenster** gewählt wurde.* Legen Sie hier die obere Position der Ansicht fest. Dabei gibt es zwei Möglichkeiten:

1. Geben Sie unter **Pixelkoordinate** einen Wert an. Dabei handelt es sich um Gerätekoordinaten.
2. Wählen Sie die Option **Beim Start automat. Berechnung**, damit die Position der Ansicht beim Programmstart automatisch berechnet wird.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Top** bzw. **VcLegendView.Top** der Programmierschnittstelle gesetzt werden.

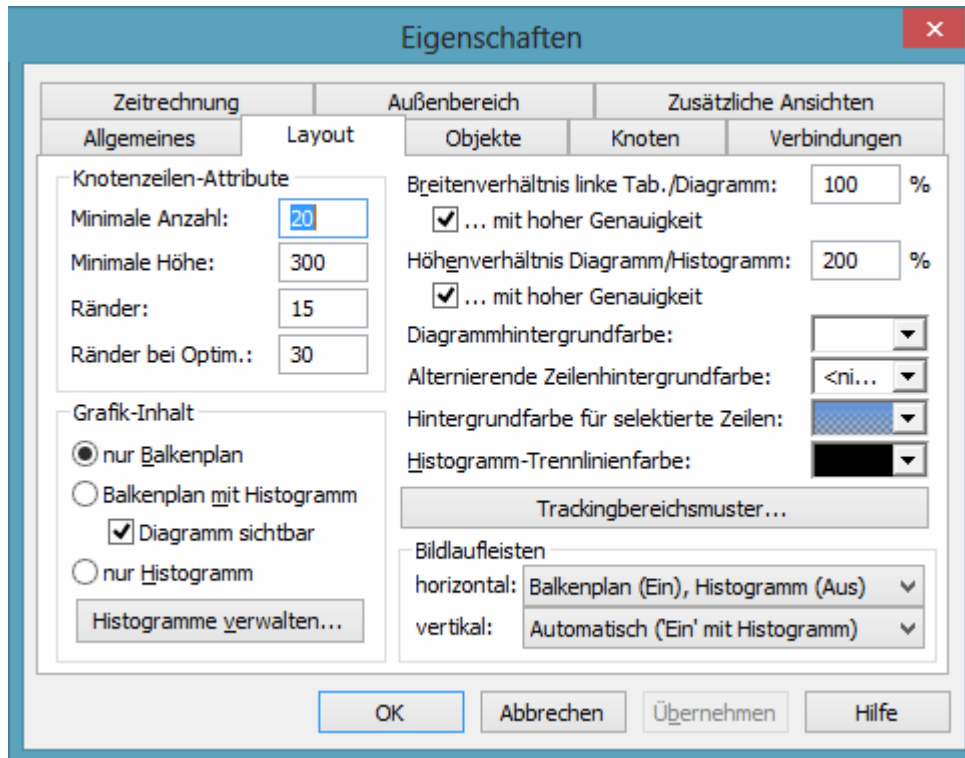
Breite

*Nicht aktiviert, wenn der Modus **fest an oberer/unterer Seite** gewählt wurde.* Legen Sie hier die horizontale Ausdehnung der Ansicht fest. Der Wert wird in Pixelkoordinaten (Gerätekoordinaten) angegeben. Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Width** bzw. **VcLegendView.Width** der Programmierschnittstelle gesetzt werden.

Höhe

*Nicht aktiviert, wenn der Modus **fest an linker/rechter Seite** gewählt wurde.* Legen Sie hier die vertikale Ausdehnung der Ansicht fest. Der Wert wird in Pixelkoordinaten (Gerätekoordinaten) angegeben. Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Height** bzw. **VcLegendView.Height** der Programmierschnittstelle gesetzt werden.

4.6 Eigenschaftenseite "Layout"



Auf dieser Eigenschaftenseite können Sie das Layout Ihrer Grafik festlegen.

Minimale Anzahl

Legen Sie hier fest, wie viele Knotenzeilen mindestens beim Start des Programms im Diagrammbereich dargestellt werden sollen.

Diese Option kann auch über die Eigenschaft **VcGantt.NoOfInitialRows** festgelegt werden.

Minimale Höhe

Legen Sie hier die minimale Höhe der Knotenzeilen in 1/100 mm fest. Die einstellbaren Werte liegen zwischen 2 und 1000.

Die minimale Höhe ist nur dann wirksam, wenn sich kein Vorgang oder nur Vorgänge mit geringerer grafischer Höhe in der Zeile befinden. In allen anderen Fällen wird die Zeilenhöhe entsprechend dem erforderlichen Platzbedarf automatisch vergrößert.

Diese Option kann auch über die Eigenschaft **VcGantt.-MinimumRowHeight** festgelegt werden.

Ränder

Legen Sie hier den minimalen vertikalen Abstand zwischen dem Knoten und der oberen/ unteren Knotenzeilengrenze in 1/100 mm fest. Diese Eigenschaft kann auch zur Laufzeit über die Eigenschaft **MinimumRowHeight** des Objektes **VcGantt** gesetzt werden. Die einstellbaren Werte liegen zwischen 2 und 1000.

Die minimale Höhe ist nur dann wirksam, wenn sich kein Vorgang oder nur Vorgänge mit geringerer grafischer Höhe in der Zeile befinden. In allen anderen Fällen wird die Zeilenhöhe entsprechend dem erforderlichen Platzbedarf automatisch vergrößert.

Diese Option kann auch über die Eigenschaft **VcGantt.RowMargins** festgelegt werden.

Ränder bei optimierter Darstellung

Mit dieser Eigenschaft können Sie den vertikalen Abstand von Subzeilen in 1/100 mm festlegen. Diese Subzeilen existieren nur, wenn Gruppen optimiert dargestellt werden und daraus Knoten in einer jeweiligen Gruppenzeile in mehrere Subzeilen verteilt werden, um nicht zu überlappen.

Diese Option kann auch über die Eigenschaft **VcGantt.SubRowMargins** festgelegt werden.

Grafik-Inhalt

Wählen Sie aus, was in der Grafik dargestellt werden soll:

- nur der Balkenplan
- Balkenplan und Histogramm
- nur das Histogramm.

Histogramme verwalten

Der Dialog **Histogramme verwalten** erscheint.

Breitenverhältnis linke Tabelle/Diagramm

Legen Sie hier das prozentuale Verhältnis von der Breite der Tabelle zur Breite des Gesamtdiagramms (Tabelle plus Diagrammbereich) beim Start fest. Mit dem Eintrag "-1" legen Sie fest, dass die Tabelle beim Start komplett gezeigt wird.

Diese Option kann auch über die Eigenschaft **VcGantt.LeftTableDiagramWidthRatio** festgelegt werden.

...mit hoher Genauigkeit

Wenn dieser Option gewählt ist, werden zur Ermittlung der Größenverhältnisse zwischen Tabelle und Diagramm die genaueren Methoden **LeftTableDiagramWidthRatioEx** und **RightTableDiagramWidthRatioEx** bzw. das Ereignis **VcTableWidthChangingEx** verwendet, die jeweils einen Wert vom Datentyp "Double" zurückgeben.

Ist die Eigenschaft nicht aktiviert, werden die Methoden **LeftTableDiagramWidthRatio** und **RightTableDiagramWidthRatio** bzw. das Ereignis **VcTableWidthChanging** verwendet.

Diese Option kann auch über die Eigenschaft **VcGantt.UseHigherTableDiagramWidthRatioPrecision** gesetzt werden.

Höhenverhältnis Diagramm/Histogramm

Legen Sie hier das prozentuale Verhältnis von der Höhe des Diagrammbereichs (also des Diagramms ohne Histogramm) und der Höhe des Histogramms beim Start fest. Wenn Sie hier "-1" wählen, wird das Histogramm beim Start komplett angezeigt.

Diese Option kann auch über die Eigenschaft **VcGantt.DiagramHistogramHeightRatio** festgelegt werden.

...mit hoher Genauigkeit

Wenn diese Option ausgewählt ist, werden zur Ermittlung des Höhenverhältnisses zwischen Diagramm und Histogramm die genauere Methode **DiagramHistogramHeightRatioEx** bzw. das Ereignis **VcHistogramHeightChangingEx** verwendet, die jeweils einen Wert vom Datentyp "Double" zurückgeben.

Beim Standardwert "False" wird die Methode **DiagramHistogramHeightRatio** bzw. das Ereignis **OnHistogramHeight** verwendet.

Diese Option kann auch über die Eigenschaft **VcGantt.UseHigherDiagramHistogramHeightRatioPrecision** gesetzt werden.

Ansichtenhintergrundfarbe

Hier können Sie eine Farbe für den Diagrammhintergrund festlegen. Zusammen mit der **Alternierenden Zeilenhintergrundfarbe** können Sie ein zeilenweise alternierendes Farbmuster einrichten.

Diese Option kann auch über die Eigenschaft **VcGantt.DiagramBackColor** oder **VcGantt.ViewComponentsBackColor** festgelegt werden.

Ansichtenrandfarbe

Hier können Sie eine Rahmenfarbe für alle Fenster gemeinsam festlegen.

Diese Option kann auch über die Eigenschaft **VcGantt.ViewComponentsBorderColor** festgelegt werden.

Alternierende Zeilenhintergrundfarbe

Hier können Sie eine zweite Hintergrundfarbe für das Diagramm setzen oder erfragen, die mit der **Diagrammhintergrundfarbe** ein zeilenweise alternierendes Farbmuster bildet.

Diese Option kann auch über die Eigenschaft **VcGantt.DiagramAlternatingRowBackColor** festgelegt werden.

Hintergrundfarbe für selektierte Zeilen

Hier können Sie eine Hintergrundfarbe für die jeweils aktuelle Zeile des Diagramms, d.h. die Zeile, in der Sie gerade arbeiten, auswählen.

Diese Option kann auch über die Eigenschaft **VcGantt.SelectedRowBackColorAsARGB** festgelegt werden.

Trackingbereichsmuster

Diese Schaltfläche öffnet den Dialog **Musterattribute bearbeiten**, in dem das Aussehen des freien Bereichs, der bei Interaktionen im LiveUpdate hin und wieder für kurze Zeit am oberen oder unteren Rand entsteht (Standardaussehen: rote Punkte auf gelbem Hintergrund) festgelegt werden kann.

Das Muster kann auch über die entsprechenden Eigenschaften **VcGantt.TrackingSpaceBackColorAsARGB**, **VcGantt.TrackingSpacePattern** und **VcGantt.TrackingSpacePatternColor!AsARGB>** festgelegt werden.

Bildlaufleisten

Hier können Sie festlegen, wo sich die Bildlaufleisten des Histogramms befinden sollen. Für den horizontalen Bereich können Sie zwischen folgenden Optionen wählen:

1. **Balkenplan (ein), Histogramm (aus)** - die horizontale Bildlaufleiste befindet sich zwischen Histogramm und Balkenplan
2. **Balkenplan (aus), Histogramm (ein)** - die horizontale Bildlaufleiste befindet sich unterhalb des Histogramms
3. **Keine** - es gibt keine horizontale Bildlaufleiste.

Für den vertikalen Bereich können Sie zwischen folgenden Optionen wählen:

1. **Automatisch (aber 'ein' mit Histogramm)** - eine vertikale Bildlaufleiste wird im Bedarfsfall automatisch rechts am Gantt-Graphen zugeschaltet; eine weitere ist stets am Histogramm vorhanden
2. **ein** - je eine vertikale Bildlaufleiste ist rechts am Gantt-Graphen und am Histogramm zugeschaltet
3. **aus** - die vertikalen Bildlaufleisten sind ausgeschaltet.

4.7 Eigenschaftenseite "Objekte"



Datentabellen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Datentabellen verwalten**.

Filter

Über diese Schaltfläche öffnen Sie das Dialogfeld **Filter verwalten**.

Zuordnungstabellen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Zuordnungstabellen verwalten**.

Kalender

Über diese Schaltfläche öffnen Sie das Dialogfeld **Kalender festlegen**.

Kalenderprofile

Über diese Schaltfläche öffnen Sie das Dialogfeld **Kalenderprofile verwalten**.

Gruppierung

Über diese Schaltfläche öffnen Sie das Dialogfeld **Gruppierung**.

Update-Verhalten

Über diese Schaltfläche öffnen Sie das Dialogfeld **Aktualisierungsverhalten verwalten**.

Layer

Über diese Schaltfläche öffnen Sie das Dialogfeld **Balkenaussehen verwalten**.

Tabellen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Tabelle festlegen**.

Zeitskalen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Zeitskala festlegen**.

Boxen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Boxen verwalten**.

Boxformate

Über diese Schaltfläche öffnen Sie das Dialogfeld **Boxformate verwalten**.

Stichtaglinien

Über diese Schaltfläche öffnen Sie das Dialogfeld **Stichtaglinien festlegen**.

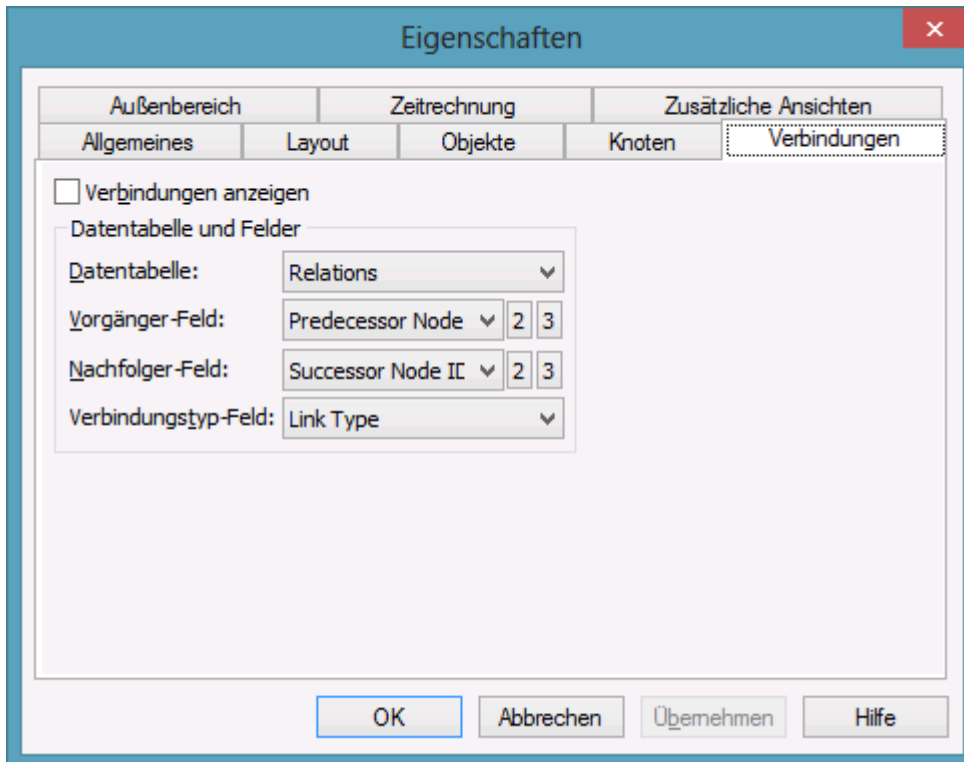
Linienformate

Über diese Schaltfläche öffnen Sie das Dialogfeld **Linienformate verwalten**.

Verbindungsaussehen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Verbindungsaussehen verwalten**.

4.8 Eigenschaftenseite "Verbindungen"



Auf dieser Eigenschaftenseite können Sie wählen, ob die Verbindungen zwischen Knoten dargestellt werden, und ihr Aussehen festlegen.

Verbindungen anzeigen

Wenn Sie dieses Kontrollkästchen aktivieren, werden sowohl Verbindungen als auch Phantomlinien für Links bei Interaktionen angezeigt.

Diese Option kann auch über die Eigenschaft der Programmierschnittstelle **VcLinkAppearance.Visible** festgelegt werden, betrifft jedoch nur die Verbindungen, nicht die Phantomlinien..

Datentabelle

Wählen Sie hier eine Datentabelle aus, die die Felder der Verbindungen enthält. Diese Option kann auch über die Eigenschaft **VcGantt.LinksDataTableName** festgelegt werden.

Vorgänger-Feld

Wählen Sie hier das Datenfeld aus der **Relations**-Tabelle aus, in dem die Identifizierung des Vorgängerknotens der Verbindungen enthalten ist. Diese

Option kann auch über die Eigenschaft **VcGantt.LinkPredecessorDataFieldIndex** festgelegt werden. Diese Eigenschaft ist eine indizierte Eigenschaft, die in C# über die beiden Methoden `set_LinkPredecessorDataFieldIndex (identifierIndex, pvn)` und `get_LinkPredecessorDataFieldIndex (identifierIndex)` angesprochen wird.

Nachfolger-Feld

Wählen Sie hier das Datenfeld bzw. die Datenfelder aus der zuvor eingestellten Datentabelle aus, in dem die Identifizierung des Nachfolgerknotens der Verbindung enthalten ist/enthalten sind. Diese Option kann auch über die Eigenschaft **VcGantt.LinkSuccessorDataFieldIndex** festgelegt werden. Diese Eigenschaft ist eine indizierte Eigenschaft, die in C# über die beiden Methoden `set_LinkSuccessorDataFieldIndex (identifierIndex, pvn)` und `get_LinkSuccessorDataFieldIndex (identifierIndex)` angesprochen wird.

Verbindungstyp-Feld

Wählen Sie hier das Datenfeld, das den Verbindungstyp enthält. Diese Option kann auch über die Eigenschaft **VcGantt.LinkTypeDataFieldIndex** festgelegt werden.

Vorgänger-Portsymbol

Wählen Sie hier für jedes Verbindungsausssehen ein Portsymbol, das die Einmündung zum Vorgängerknoten kennzeichnet.

Diese Option kann auch über die Eigenschaft **Link AppearancePrePort-Symbol** festgelegt werden.

Nachfolger-Portsymbol

Wählen Sie hier für jedes Verbindungsausssehen ein Portsymbol, das die Einmündung zum Nachfolgerknoten kennzeichnet.

Diese Option kann auch über die Eigenschaft **Link AppearanceSuccPort-Symbol** festgelegt werden.

4.9 Eigenschaftenseite "Zeitrechnung"

Zeitrechnungseingabe		Zeitrechnungsergebnis	
Eingabe	aus Feld	Ausgabe	in Feld
Vorgänger (Teil 1)	Relations:Prede...	früh. Start	
Vorgänger (Teil 2)		früh. Ende	
Vorgänger (Teil 3)		spät. Start	
Nachfolger (Teil 1)	Relations:Succe...	spät. Ende	
Nachfolger (Teil 2)		Freier Puffer	
Nachfolger (Teil 3)		Gesamtpuffer	
Verbindungstyp	Relations:Link T...		
Verbindungsdauer			
Dauer			
akt. Start			
akt. Ende			
Start nicht früher als			
Ende nicht später als			

☐ Nur Knoten mit Vorgängern berechnen
☐ Automatische Zeitrechnung

Mit Hilfe dieser Eigenschaftenseite können Sie die Zeitrechnung des VARCHART XGantt an Ihre Schnittstelle anpassen, indem Sie festlegen, welche Datenfelder für die Eingabe (**Zeitrechnungseingabe**) und die Ausgabe (**Zeitrechnungsergebnis**) der Zeitberechnung (des Schedulers) verwendet werden sollen. (Vgl. "Wichtige Begriffe: Zeitrechnung".)

Nur Knoten mit Vorgängern berechnen

Wenn dieses Kontrollkästchen aktiviert ist, werden beim Zeitrechnen nur die Vorgänge, die Vorgänger besitzen, berechnet. Ein "Projektstart" beim Aufruf der Zeitrechnung wird ignoriert.

Wenn dieses Kontrollkästchen nicht aktiviert ist, werden beim Scheduling alle Vorgänge berechnet.

Automatische Zeitrechnung

Wenn diese Option aktiviert ist, werden beim Anlegen oder Löschen einer Verbindung oder beim Ändern der Dauer eines Vorgangs automatisch die abhängigen Termine neu berechnet.

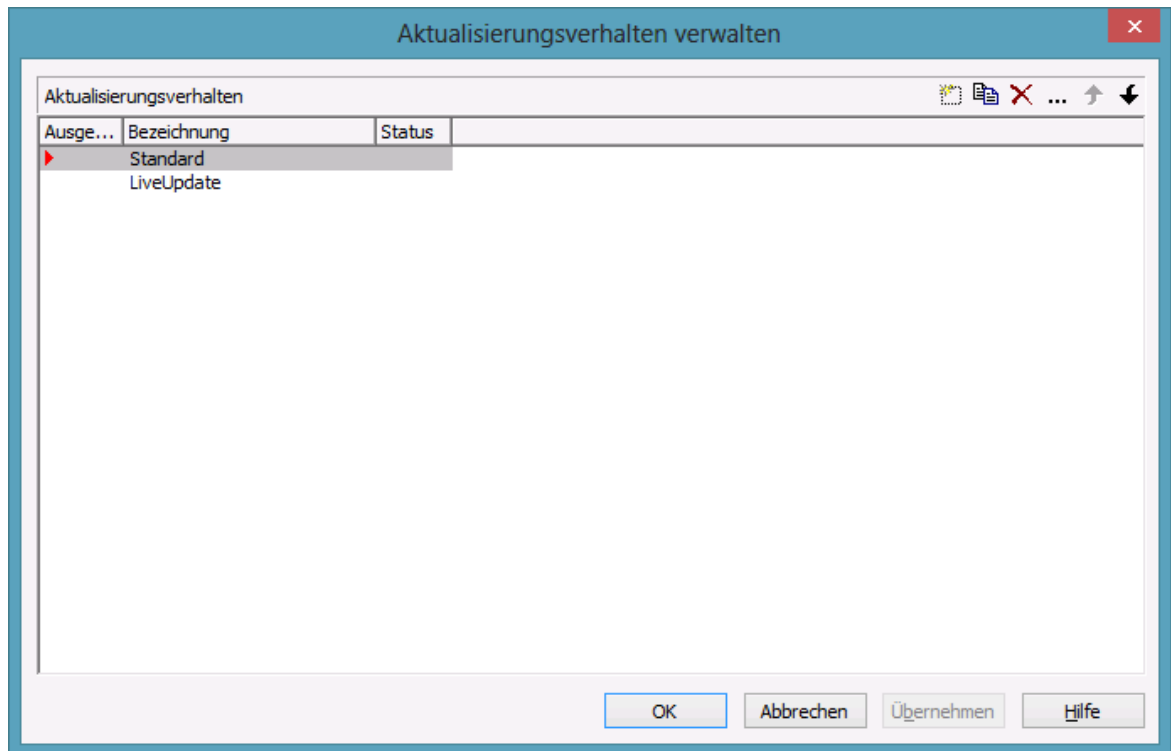
Zeitrechnungseingabe

Wählen Sie hier für jede Eingabe aus, aus welchem Feld sie entnommen werden soll. Als Eingaben für die Zeitrechnung verwendet der Scheduler Datenfelder der jeweils eingestellten Knoten- und Verbindungstabellen.. Die Grundlage der Berechnung sind die Dauer der einzelnen Vorgänge, deren logische Abhängigkeiten und der Projektanfang. Die Felder **Vorgänger** und **Nachfolger** können in der **Zeitrechnungseingabe**-Tabelle nicht bearbeitet werden. Sie geben nur die auf der Eigenschaftenseite **Verbindungen** vorgenommenen Einstellungen wieder.

Zeitrechnungsergebnis

Wählen Sie hier für jedes Ergebnis aus, in welches Feld es geschrieben werden soll. Die Ausgabe des Schedulers erfolgt nur in Datenfelder der **Maindata**-Tabelle. Aus der Dauer der einzelnen Vorgänge, deren logischen Abhängigkeiten und dem Projektanfang werden die frühesten bzw. spätesten Start- und Endtermine sowie der Gesamtpuffer und der freie Puffer berechnet.

4.10 Dialogfeld "Aktualisierungsverhalten verwalten"



Sie gelangen in diesen Dialog über die Eigenschaftenseite **Objekte**. Hier können Sie eigene Aktualisierungsverhalten anlegen, kopieren, löschen und verschieben. Über die Schaltfläche **...** öffnen Sie den Dialog **Aktualisierungsverhalten bearbeiten**.

4.11 Dialogfeld "Aktualisierungsverhalten bearbeiten"

Aktualisierungsverhalten "LiveUpdate" bearbeiten

Kontexte (Objekte und Funktionen)		
Bezeichnung	Aktualisierungsmodus	Verzögerungszeit
Tabellen		
Spaltenbreite ändern	Beim Verschieben der Maus	500 ms
Zeitskalen		
Einheitenbreite ändern	Beim Verschieben der Maus	500 ms
Anfangsdatum des Zeitskalenabschnitts ändern	Beim Verschieben der Maus	500 ms
Stichtaglinien		
Datum ändern	Beim Verschieben der Maus	500 ms
Boxen		
Größe ändern	Beim Verschieben der Maus	500 ms
Position ändern	Beim Verschieben der Maus	500 ms
Verankerungsknoten ändern	Beim Verschieben der Maus	500 ms
Knoten		
Termine/Dauer ändern	Beim Verschieben der Maus	500 ms
Gruppe ändern	Beim Verharren während d...	500 ms
Filtern/Mappen	Beim Verharren während d...	500 ms
Gruppieren	Beim Verharren während d...	500 ms
Automatische Terminberechnung	Beim Verharren während d...	500 ms
Verbindungen		
Nachfolgerknoten ändern	Beim Verschieben der Maus	500 ms
Knotenzeilen-Layout		
Sortierreihenfolge der Knoten ändern	Beim Verharren während d...	500 ms
Hierarchie-Layout		
Berechnung von Summenbalken	Beim Verschieben der Maus	500 ms
Automatisches Kollabieren von Gruppen	Beim Verharren während d...	500 ms
Restaurieren von automatisch kollabierten Gru...	Beim Lösen der Maustaste	0 ms
Automatisches Expandieren einer Ziel-Gruppe	Beim Verharren während d...	500 ms
Restaurieren von automatisch expandierten G...	Beim Verharren während d...	500 ms
Gruppenzeilen-Layouts		
Sortierreihenfolge der Gruppen ändern	Beim Verharren während d...	500 ms
Knotenoptimierung	Beim Verharren während d...	500 ms
Berechnung von Summenbalken	Beim Verschieben der Maus	500 ms
Sortierung von überlappenden Knoten	Beim Verharren während d...	500 ms
Automatisches Kollabieren von Gruppen	Beim Verharren während d...	500 ms
Restaurieren von automatisch kollabierten Gru...	Beim Lösen der Maustaste	0 ms
Automatisches Expandieren einer Ziel-Gruppe	Beim Verharren während d...	500 ms
Restaurieren von automatisch expandierten G...	Beim Verharren während d...	500 ms
Histogramme		
Berechnung Layer-basierter Kurven	Beim Verschieben der Maus	500 ms
Numerische Skalen		
Einheitenhöhe ändern	Beim Verschieben der Maus	500 ms
Kurven		
X-Koordinate ändern	Beim Verschieben der Maus	500 ms
Y-Koordinate ändern	Beim Verschieben der Maus	500 ms
X- und Y-Koordinate ändern	Beim Verschieben der Maus	500 ms
Sashes		
Position ändern	Beim Verschieben der Maus	500 ms
Komplettansicht		
Größe ändern	Beim Verschieben der Maus	500 ms
Position ändern	Beim Verschieben der Maus	500 ms

OK Abbrechen Hilfe

Diesen Dialog erreichen Sie über den Dialog **Aktualisierungsverhalten verwalten**. Hier können Sie für individuell angelegte Aktualisierungsverhalten den Update-Modus und/oder die Verzögerungszeit ändern.

Verzögerungszeit

Nur wenn bei **Update-Modus** die Option **Verharren während des Verschiebens** ausgewählt wurde, kann hier die Verzögerungszeit, nach der während des Verschiebvorgangs des Mauszeigers die Aktualisierung durch das Live Update erfolgen soll, eingestellt werden.

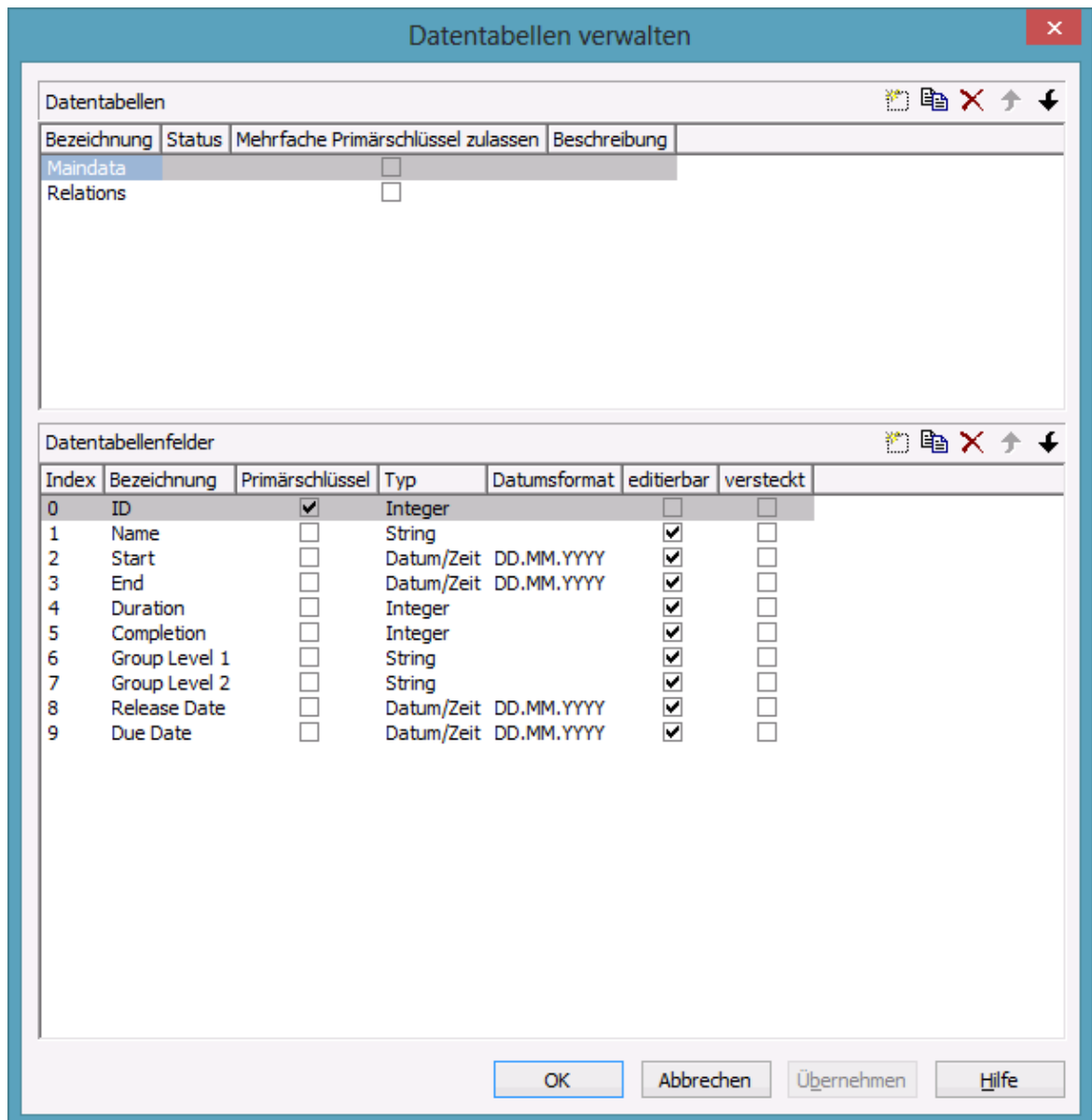
Update-Modus

Hier können Sie bei einem selbst erstellten Aktualisierungsverhalten auswählen, bei welcher Aktion des Cursors die Aktualisierung durch das Live Update erfolgen soll.

Bezeichnung

In dieser Spalte finden Sie die Namen der Objekte und zugehörigen Funktionen, die vom Live Update betroffen sind. Diese Bezeichnungen sind **nicht** editierbar.

4.12 Dialogfeld "Datentabellen verwalten"



Sie gelangen in diesen Dialog über die Eigenschaftenseite **Objekte**. Sie können hier Datentabellen sowie die zugehörigen Datenfelder anlegen und bearbeiten.

Datentabellen

- **Bezeichnung:** In dieser Spalte stehen die Namen aller vorhandenen Datentabellen. Die Namen sind editierbar.
- **Status:** In der Spalte **Status** wird jede Dateantabelle gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (📄) und/oder geändert (⚠️) worden ist.

- **Mehrfache Primärschlüssel zulassen:** Bestimmen Sie hier, ob der Primärschlüssel der Tabelle aus **einem** oder **mehreren (maximal 3)** Feldern bestehen soll. Sobald Sie **Mehrfache Primärschlüssel zulassen** ausgewählt haben, sind im Bereich **Datentabellenfelder** bis zu 3 Felder für den Primärschlüssel wählbar. Die Einstellung **Mehrfache Primärschlüssel zulassen** kann erst dann wieder deaktiviert werden, wenn im Bereich **Datentabellenfelder** nur noch ein Feld für den Primärschlüssel ausgewählt ist.
- **Beschreibung:** Geben Sie hier eine Beschreibung für die Datentabelle ein.

Datentabelle hinzufügen / kopieren / löschen / nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie Datentabellen hinzufügen, kopieren, löschen und in der Liste nach oben oder unten verschieben.

Datentabellenfelder

Hier können Sie für die im Bereich **Datentabellen** ausgewählte Datentabelle Datentabellenfelder anlegen und bearbeiten.

- **Index:** Der Index der Datentabellenfelder ist nicht veränderbar, da er intern als Referenz dient. In der API werden Datenfelder über den Index angesprochen.
- **Bezeichnung:** Diese Spalte zeigt die Namen der Felder der Datentabelle. Nach Anklicken können Sie diese ändern.
- **Primärschlüssel:** Hier können Sie festlegen, welches Feld in der Spalte der Primärschlüssel der Datensätze dieser Tabelle sein soll.
- **Typ:** Hier können Sie den Datentyp für jedes Datentabellenfeld festlegen. Zur Auswahl stehen:

Alphanumerisch

Integer

Datum/Zeit

Double

- **Datumsformat:** Für die Datentabellenfelder vom Typ **Datum/Zeit** können Sie hier jeweils das Datumsformat an das Datumsformat Ihrer Knotendaten anpassen. Einige gebräuchliche Datumsformate stehen zur

Auswahl. Sie können außerdem ein eigenes Datumsformat definieren, z. B. "DD.MM.YY hh:mm".

Das Datumsformat wird aus den Kombinationen **YY** (zweistellige Jahreszahl), **YYYY** (vierstellige Jahreszahl), **MM** (zweistellige Monatszahl), **MMM** (dreistelliges Monatsnamenkürzel), **DD** (zweistellige Tageszahl), **hh** (zweistellige Stundenzahl), **mm** (zweistellige Minutenzahl) und **ss** (zweistellige Sekundenzahl) zusammengesetzt.

Beachten Sie bitte, dass das Datumsformat, das Sie hier festlegen, mit dem Datumsformat Ihrer Knotendaten übereinstimmen muss.

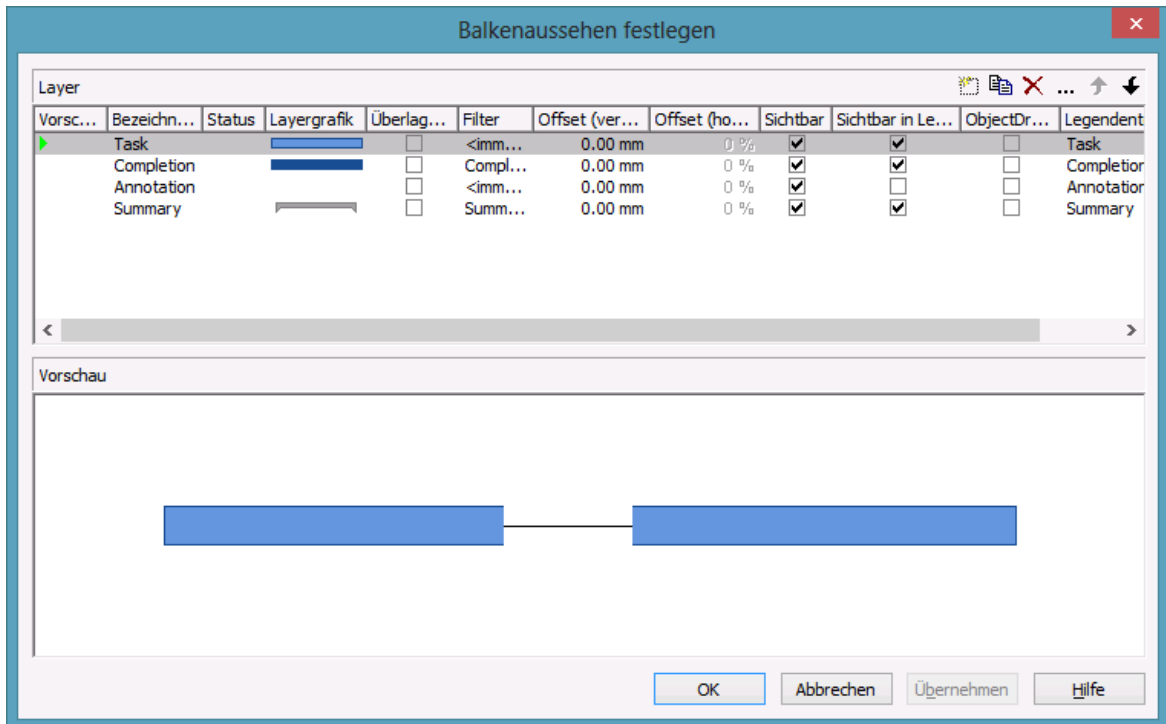
Dieses Datumsformat ist nur für die Dateneingabe, aber nicht für die Darstellung von Daten in Ihrer Grafik relevant.

- **Editierbar:** Aktivieren Sie dieses Kontrollkästchen für alle Datentabellenfelder, die der Anwender im Dialogfeld **Vorgänge bearbeiten** bearbeiten können soll.
- **Versteckt:** Aktivieren Sie dieses Kontrollkästchen für alle Datentabellenfelder, die dem Anwender im Dialogfeld **Vorgänge bearbeiten** verborgen bleiben sollen.
- **Beziehung:** Hier können Sie eine Verknüpfung zu einer anderen Tabelle festlegen, so dass die Datensätze dieser Tabelle über das als Primärschlüssel definierte Feld einer anderen verknüpft sind. Aus diesem Grund werden Ihnen zur Auswahl alle Tabellen angeboten, für die Sie einen Primärschlüssel definiert haben.

Datentabellenfeld hinzufügen / kopieren / löschen / nach oben / unten

 Mit Hilfe dieser Schaltflächen können Sie Datentabellenfelder hinzufügen, kopieren, löschen und in der Liste nach oben oder unten verschieben.

4.13 Dialogfeld "Balkenaussehen festlegen"



Vorgänge werden durch Balken dargestellt. Das grafische Design eines Balkens wird im Balkenaussehen festgelegt. Ein Vorgang setzt sich aus einem oder mehreren Layern zusammen, die dem Vorgang dynamisch über Filter zugewiesen werden.

Layer stellen Termine (Symbolayer) oder Terminpaare (Rechteck-Layer oder Linien-Layer) grafisch dar. Diese Termine kommen aus den Datenfeldern, die Sie im Dialogfeld **Layer bearbeiten** auswählen.

Layer beinhalten sowohl grafische Attribute (Form, Linienfarbe, Muster, etc.) als auch eine Beschriftung. Außerdem können sie unterschiedliche Höhen und einen Offset haben, um sicherzustellen, dass alle Layer, die einem Vorgang zugewiesen sind, sichtbar sind.

Wenn ein Vorgang durch mehrere Layer dargestellt wird, werden die einzelnen Layer nacheinander gezeichnet, wobei sie einander u. U. überlagern. Der Layer, der ganz oben in der **Layer**-Tabelle steht, wird zuerst gezeichnet. Der letzte Layer in der **Layer**-Tabelle wird zuletzt gezeichnet und verdeckt u. U. die zuvor gezeichneten Layer. Aus der grafischen Überlagerung aller Layer, deren Filter auf einen Vorgang zutreffen, ergibt sich das Balkenaussehen.

Layer

In jeder Zeile dieser Tabelle können Sie einen Layer definieren.

Vorschau

Alle Layer, die in dieser Spalte durch eine kleine Pfeilspitze markiert sind, werden im Vorschaufenster in der unteren Hälfte des Dialogs angezeigt.

Durch eine grüne Pfeilspitze wird jeweils der Layer gekennzeichnet, auf dem der Cursor gerade steht. Er wird im Vorschaufenster angezeigt.

Wenn Sie auf Layer in der Spalte **Vorschau** klicken, erscheinen rote Pfeilspitzen, die anzeigen, dass die Layer mit ihren aktuellen Einstellungen im Vorschaufenster dargestellt werden.

Bezeichnung

In dieser Spalte werden die Namen aller verfügbaren Layer aufgeführt. Die Namen sind editierbar.

Status

In dieser Spalte wird jeder Layer gekennzeichnet, der seit dem Aufruf des Dialogs hinzugefügt () oder geändert () worden ist.

Layergrafik

In dieser Spalte wird das Aussehen jedes Layers (Layergrafik) angezeigt. Um die Layergrafik eines Layers zu verändern, klicken Sie auf die **Layer bearbeiten**-Schaltfläche oberhalb der Tabelle oder doppelklicken Sie auf den Eintrag **Layergrafik**. Sie gelangen in den Dialog **Layer bearbeiten**, in dem Sie die grafischen Attribute und die Beschriftung des Layers festlegen können.

Überlagerungslayer

Sie können im Modus **Alle Knoten in einer Zeile** und bei ausgeschalteter **optimiert**-Option die Überlagerung von Layern (und das dadurch bedingte Ausmaß an Verdeckung) durch einen unterhalb des Knotens befindlichen, kleineren Überlagerungslayer anzeigen lassen. Nur einer der in der Liste vorhandenen Layer kann der Überlagerungslayer sein. Für den Überlagerungslayer wird kein Filter verwendet.

Filter

Der Filter, der mit einem Layer verbunden ist, bestimmt, welche Vorgänge durch den betreffenden Layer dargestellt werden. Um den Filter für einen Layer auszusuchen, markieren Sie das **Filter**-Feld. Dann erscheinen zwei Schaltflächen:



Öffnen Sie die Kombobox mit allen verfügbaren Filtern und wählen Sie daraus einen Filter für den Layer aus.



Oder klicken Sie auf die **Bearbeiten**-Schaltfläche, um den Dialog **Filter verwalten** aufzurufen. Dort können Sie Filter bearbeiten, kopieren, neu definieren oder löschen.

Beispiele für Filter sind "Standard", "Kritisch", "Meilenstein". Der gewählte Filter gibt die Bedingung an, unter der der Layer erscheinen soll. Haben Sie beispielsweise für den Layer "Früh" den Filter "kritisch" gewählt, so werden die "Früh"-Layer nur für kritische Vorgänge angezeigt.

Offset (vertikal)

Der vertikale Offset (Versatz des Layers gegenüber seiner horizontalen Mittellinie) wird in Millimetern angegeben. Positive Werte bedeuten eine Verschiebung nach oben, negative nach unten.

Sobald Sie das **Offset (vertikal)**-Feld eines Layers markieren, erscheinen neben dem Eintrag Einstell-Schaltflächen, mit denen Sie den vertikalen Offset des markierten Layers schrittweise erhöhen oder verringern können.



Außerdem erscheint diese Schaltfläche, über die Sie in den Dialog **Zuordnung einstellen** gelangen, in dem Sie eine datenabhängige Zuordnung des vertikalen Offsets vereinbaren können.



Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt.

Offset (horizontal)

(nur für Symbol-Layer) Sobald Sie das Feld **Offset (horizontal)** eines Symbollayers markieren, erscheint neben dem Eintrag eine Schaltfläche mit je einem Pfeil nach oben und nach unten. Sie können hier den horizontalen Offset des markierten Layers (seinen horizontalen Versatz gegenüber dem Layerdatum) in Schritten von jeweils 1 % erhöhen oder verringern (-50 bis +50 %).

Sichtbar

Deaktivieren Sie dieses Kontrollkästchen, damit der Layer unsichtbar ist. So können Sie einen Layer ausblenden, ohne ihn zu löschen.

Sichtbar in Legende

Aktivieren Sie dieses Kontrollkästchen, damit der Layer in der Legende dargestellt wird.

ObjectDraw-Ereignisse

Aktivieren Sie dieses Kontrollkästchen, damit die Ereignisse **OnObjectDraw** und **OnObjectDrawComplete** bei Knoten, die mit diesem Layer gezeichnet werden, auftreten können.

Legendentext

Hier können Sie einen Legendentext für den Layer festlegen.

Layer hinzufügen



Ein neuer Layer wird erzeugt.

Layer kopieren



Der markierte Layer wird kopiert.

Layer löschen



Der markierte Layer wird gelöscht.

Layer bearbeiten



Der Dialog **Layer bearbeiten** wird geöffnet.

Layer früher/später zeichnen

Wenn ein Knoten aus mehreren Layern besteht, werden die einzelnen Layer dieses Knotens schichtweise übereinander gezeichnet. Dabei wird mit dem obersten Layer der Tabelle angefangen. Je tiefer die Position eines Layers in

der Liste, desto mehr andere Layer überlagert er. Die Zeichenreihenfolge der Layer entspricht also ihrer Reihenfolge in der Tabelle.

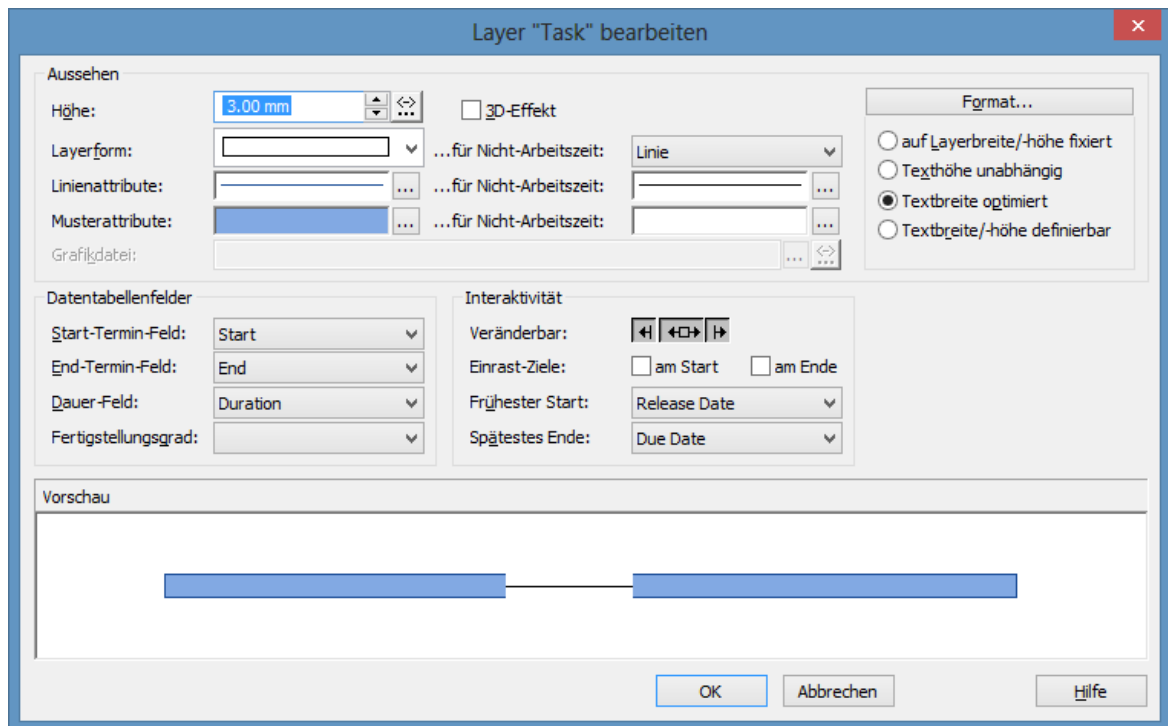
⬆️ Klicken Sie die Schaltfläche **Layer früher zeichnen** an, um den markierten Layer in der Tabelle eine Position höher zu setzen. In der Darstellung gelangt er dabei um eine Stelle weiter in den Hintergrund. Wenn er ganz oben in der Tabelle steht, wird er von allen anderen Layern überlagert.

⬇️ Klicken Sie die Schaltfläche **Layer später zeichnen** an, um den markierten Layer in der Tabelle eine Position tiefer zu setzen. In der Darstellung gelangt er dabei um eine Stelle weiter in den Vordergrund. Wenn er ganz unten in der Tabelle steht, überlagert er alle anderen Layer.

Vorschaufenster

Das Vorschaufenster zeigt alle Layer, die in der **Vorschau**-Spalte markiert sind, mit den durch die Festlegung der Zeichnungsreihenfolge und des Offsets bestimmten Überlagerungen an.

4.14 Dialogfeld "Layer bearbeiten"



Sie erreichen dieses Dialogfeld über die entsprechende Schaltfläche im Dialogfeld **Balkenaussehen festlegen**. In der Kopfzeile wird der Name des zu bearbeitenden Layers angezeigt.

Höhe

In diesem Feld können Sie die Höhe des Layers in Millimetern definieren.

Dies geschieht entweder durch direkte Eingabe des gewünschten Wertes in das Feld oder über die beiden Pfeil-Schaltflächen nach oben/nach unten.

 Über diese Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**, in dem Sie eine datenabhängige Zuordnung der Layer-Höhe vereinbaren können.

 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche gefüllt dargestellt.

3D-Effekt

Sie können hier wählen, ob die Layer einen räumlichen Eindruck erhalten sollen oder nicht.

Layerform

Aus der **Layerform**-Liste können Sie die Form des Layers auswählen. Zur Auswahl stehen:

- **Bitmap-Layer:** Sie können unter **Grafikdatei** die zu verwendende Bitmapdatei auswählen.
- **Unsichtbares Symbol:** Der Layer ist mit Ausnahme seiner Beschriftung unsichtbar und wird auch nicht in der Legende dargestellt.
- **Rechtecklayer**
- **Keilförmige Layer:** aufsteigend oder absteigend keilförmig
- **Linienlayer**
- Verschiedene Typen von **Symbollayern**.

Rechtecklayer, keilförmige Layer und Linienlayer dienen zur Darstellung von Zeitspannen. Keilförmige Layer können verwendet werden, um ansteigende bzw. ausklingende Aktivitäten darzustellen, beispielsweise am Anfang bzw. am Ende von Vorgängen. Symbollayer dienen zur Darstellung von Zeitpunkten.

Form für arbeitsfreie Zeiten

Wählen Sie hier, in welcher Form arbeitsfreie Zeiten in Rechtecklayern dargestellt werden. Zuvor muss auf der Eigenschaftenseite **Knoten** die Option **Layer zeigen arbeitsfreie Zeiten anzeigen** ausgewählt werden.

Zur Auswahl stehen die Formen <Rechteck>, <Linie>, <freie Bereich> oder <keine>. <Keine> bewirkt, dass der Layer durchgezogen wird. In Kombination mit der oben erwähnten Option kann man so für einzelne Layer die arbeitsfreien Zeiten darstellen und für andere nicht.

Linienattribute

Hier wird die ausgewählte Linie der Layerumrandung angezeigt. Um die Linie zu ändern (Typ, Dicke, Farbe), klicken Sie auf die **Bearbeiten**-Schaltfläche (⋮). Dann öffnet sich der Dialog **Linienattribute bearbeiten**.

Linienattribute für arbeitsfreie Zeiten

Bestimmen Sie hier, wie Layer für arbeitsfreie Zeiten umrandet werden. Durch Klick auf ⋮ gelangen Sie in den Dialog **Linienattribute bearbeiten**.

Musterattribute

Hier wird das ausgewählte Muster des Layers angezeigt. Durch Klick auf  gelangen Sie in den Dialog <bMusterattribute bearbeiten, in dem Sie Muster, Musterfarbe, Hintergrundfarbe oder die 2. Musterfarbe einstellen können.

Musterattribute für arbeitsfreie Zeiten

Bestimmen Sie hier, ob und wie Layer für arbeitsfreie Zeiten gefüllt werden. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**

Grafikdatei

*(nur aktiv, wenn unter **Layerform** die Option <Bitmap-Layer> gewählt wurde)* Hier können Sie eine Grafikdatei auswählen, mit der der Layer dargestellt werden soll.

Auch relative Pfadangaben sind möglich. Wird ein relativer Dateiname angegeben, so wird die Datei zur Laufzeit zuerst in dem Verzeichnis gesucht, das in der VARCHART-Eigenschaft **FilePath** gesetzt ist. Wird sie dort nicht gefunden, wird sie zuerst im gerade aktiven Arbeitsverzeichnis der Applikation und dann im Installationsverzeichnis des Steuerelementes gesucht.

 Klicken Sie auf diese Schaltfläche, um den Windows-Dialog **Grafikdatei auswählen** zu öffnen.

 Über diese Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Grafik vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt ().

Bei der Darstellung der Grafik wird die Farbe des Pixels ihrer Ecke links oben durch die Diagramm-Hintergrundfarbe ersetzt, d. h. alle Pixel der Grafik in dieser Farbe werden transparent dargestellt.

auf Layerbreite/höhe fixiert

Wenn Sie diese Option wählen, werden Höhe und Breite der Beschriftung durch die Layerhöhe bzw. -breite bestimmt.

Texthöhe unabhängig

Wenn Sie diese Option wählen, ist die Höhe der Beschriftung außerhalb eines Layers unabhängig von der Layerhöhe, während ihre Breite durch die Layerbreite bestimmt wird. Die Höhe einer Beschriftung innerhalb eines Layers wird immer durch die Layerhöhe beschränkt.

Textbreite optimiert

Wenn Sie diese Option wählen, ist die Breite der Beschriftung außerhalb eines Layers unabhängig von der Layerbreite, während ihre Höhe durch die Layerhöhe bestimmt wird. Die Breite einer Beschriftung innerhalb eines Layers wird immer durch die Layerbreite beschränkt.

Textbreite/-höhe definierbar

Wenn Sie diese Option wählen, sind Breite und Höhe der Beschriftung unabhängig von der Layerbreite bzw. -höhe. Sie können dann die Breite und die Zeilenanzahl für die Felder einzeln im Dialog **Layerformat bearbeiten** oder über die Eigenschaften **MinimumWidth** und **TextLineCount** in Objekten vom Typ **VcLayerFormatField** festlegen.

Start-Termin-Feld

Definieren Sie hier den Anfangstermin des gewählten Layers, bspw. Frühester Anfang, Spätester Anfang, Geplanter Start.

Format

Über diese Schaltfläche öffnen Sie den Dialog **Layerformat bearbeiten**.

End-Termin-Feld

Definieren Sie hier den Endtermin des gewählten Layers, bspw. Frühestes Ende, Spätestes Ende, Geplantes Ende.

Ein Rechteck- bzw. Linien-Layer wird durch die Angabe eines Start-Termin-Feldes und eines End-Termin-Feldes oder eines Start-Termin-Feldes und einer Dauer definiert. Sind sowohl das End-Termin-Feld als auch die Dauer spezifiziert, so hat die Dauer Vorrang vor dem End-Termin-Feld. Bei Interaktionen wird dann aber nicht nur das Dauer-Feld aktualisiert, sondern auch das End-Termin-Feld.

Dauer

Die Einheit der Dauer wird in Abhängigkeit von der Einstellung der Zeiteinheit auf der Eigenschaftenseite **Allgemeines** interpretiert.

Wählen Sie aus der Liste das Datenfeld aus, das die Dauer des gewählten Layers enthält.

Fertigstellungsgrad

(nicht aktiv für Symbol- und Bitmaplayer) Wenn der aktuelle Layer den prozentualen Fertigstellungsgrad anzeigen soll, wählen Sie aus der Liste das Datenfeld aus, das den prozentualen Fertigstellungsgrad des gewählten Layers enthält.

Der durch den Layer dargestellte Endtermin wird aus dem Start-Termin-Feld, dem Endtermin-Feld bzw. der Dauer und dem Fertigstellungsgrad berechnet. Die dem Vorgang zu Grunde liegenden Daten bleiben unverändert.

Veränderbar

Sie können hier festlegen, ob der Anwender mit der Maus den gesamten Layer, den Layer-Anfang und/oder das Layer-Ende verschieben darf.

Sie können drei Optionen für den Anwender freigeben oder sperren:

1. den Layer-Anfang verändern
2. den gesamten Layer (d. h. Anfang und Ende des Layers gemeinsam) verschieben
3. das Layer-Ende verändern

Die gedrückten Schaltflächen zeigen die für den Anwender freigegebenen Optionen an.

Einrastziele

Bestimmen Sie hier, ob der Layer Start- und/oder Endedatum als Einrastziel definiert.

Frühester Start

Das im hier ausgewählten Datenfeld hinterlegte Datum mit Uhrzeit stellt die untere Grenze für die Startzeit des Layers beim interaktiven Verschieben des Layers bzw. Knotens dar.

Diese Option kann auch über die Eigenschaft **VcLayer.MinimumStartDataFieldIndex** festgelegt werden.

Spätestes Ende

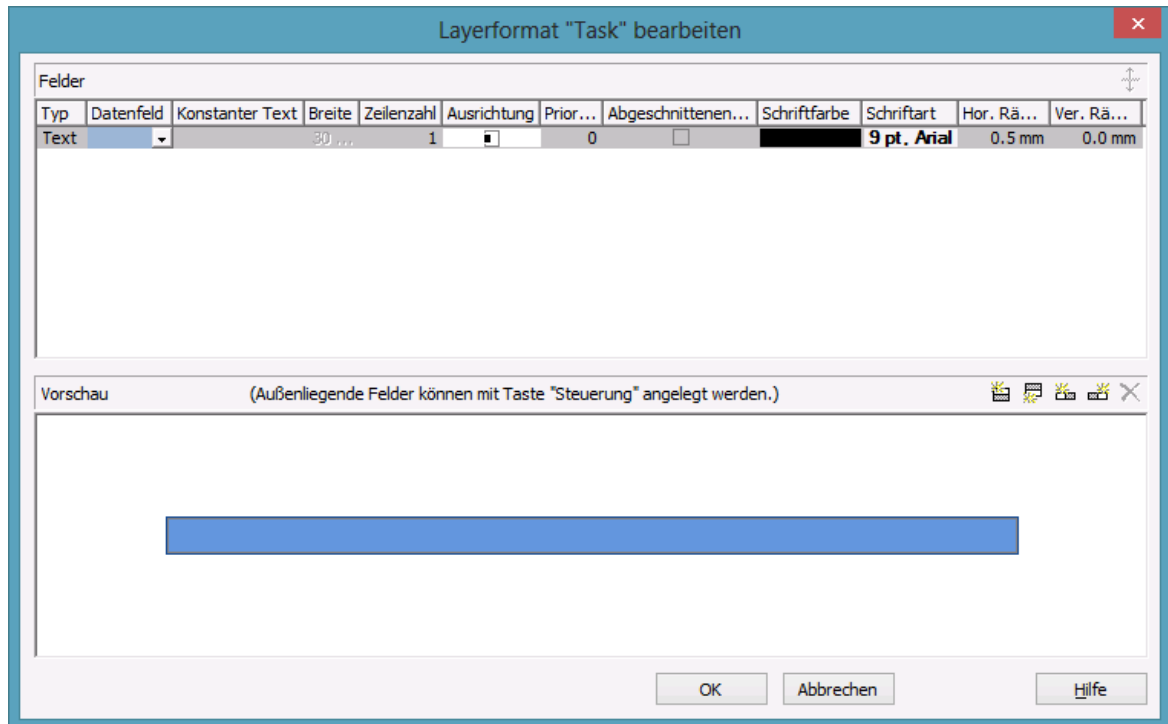
Das im hier ausgewählten Datenfeld hinterlegte Datum mit Uhrzeit stellt die obere Grenze für die Endzeit des Layers beim interaktiven Verschieben des Layers bzw. Knotens dar.

Diese Option kann auch über die Eigenschaft **VcLayer.MaximumEndDataFieldIndex** festgelegt werden.

Vorschau

In der Vorschau wird der Layer mit den hier gewählten Einstellungen dargestellt. Dabei werden Balkenlayer in der Vorschau immer von einer durchgezogenen Linie durchbrochen. Diese Linie zeigt an, wie der Layer zur Laufzeit aussieht, wenn arbeitsfreie Zeiten angezeigt werden und wenn den Knoten ein Kalender zugewiesen wird. (Diese beiden Einstellungen können Sie auf der Eigenschaftenseite **Knoten** vornehmen. Die Darstellung in der Vorschau ist allerdings unabhängig von diesen Einstellungen.)

4.15 Dialogfeld "Layerformat bearbeiten"



Sie erreichen dieses Dialogfeld, indem Sie im Dialogfeld **Layer bearbeiten** auf die **Format**-Schaltfläche klicken.

Typ

Hier wird der Feldtyp (Text) angezeigt.

Datenfeld

Wählen Sie hier das Datenfeld, dessen Inhalt in dem aktuellen Feld ausgegeben werden soll. Außer den Datenfeldern, die Sie in der Datendefinition vereinbart haben, gibt es die Option <Zeilennummer>: Dabei wird die Zeilennummer des Balkens ausgegeben.

passt der Inhalt des Datenfeldes nicht in das Feld, wird der Überhang bei der Ausgabe abgeschnitten.

Konstanter Text

(nur wenn kein Datenfeld gewählt wurde) Sie können hier einen konstanten Text eingeben, der in dem Feld ausgegeben werden soll.

Breite

Legen Sie hier die Breite in Millimetern für das markierte Feld fest. Die maximale Breite eines Feldes beträgt 90 mm.

Hinweis:Nur nutzbar, wenn im Dialog **Layer bearbeiten** die Option **Textbreite/-höhe definierbar** gewählt ist.

Zeilenanzahl

Bestimmen Sie die Anzahl der Textzeilen des aktuellen Feldes.

Hinweis:Nur nutzbar, wenn im Dialog **Layer bearbeiten** die Option **Textbreite/-höhe definierbar** gewählt ist.

Nur für außenstehende Felder eines Layers: Sie können die Anzahl der Textzeilen hier auch dynamisch gestalten, d.h. abhängig von der Textlänge. Dazu haben Sie zwei Möglichkeiten:

1. Sie können die Zeilenzahl des Feldes direkt ermitteln lassen, in ein Feld schreiben und den Inhalt hier verwenden
2. Sie können die Zeilenzahl in einer Zuordnungstabelle niederlegen und hier zuordnen

Zu 1: Über die Methode **VcLayerFormatField.CalculateLineCount(...)** können Sie Anzahl der Zeilen ermitteln lassen und in einem Feld ablegen. Das Feld können Sie über den Zuweisungsdialog zuweisen, indem Sie in dem Feld **Zeilenanzahl** auf die rechte Schaltfläche mit dem Doppelpfeil drücken:



Daraufhin öffnet sich der Zuordnungsdialog, in welchem Sie oben den Namen des Datenfeldes auswählen. Das darunterliegende Feld für eine Zuordnungstabelle lassen Sie leer.

Zu 2: Die Verwendung einer Zuordnungstabelle setzt ihre vorherige Anlage und Bestückung voraus; zudem muss sie vom Typ **vcNumberMap** sein. Bei diesem Typ werden Texten Zahlen zugeordnet. Wenn ein hier niedergelegter Text in einem (noch zuzuweisenden) Datenfeld gefunden wird, erhält das Feld die zugeordnete Anzahl von Textzeilen. Zuordnungstabellen legen Sie über die Eigenschaftenseite **Objekte** und die dortige Schaltfläche **Zuordnungstabellen...** an. Die erstellte Zuordnungstabelle weisen Sie nun ebenfalls in dem Feld **Zeilenanzahl** durch Drücken auf die rechte Schaltfläche mit dem Doppelpfeil zu. In dem sich öffnenden Zuordnungsdialog können Sie das Datenfeld und die Zuordnungstabelle

auswählen und weisen auf diese Weise das Datenfeld zu, dessen Inhalt mit den Textzeichen der Zuordnungstabelle abgeglichen wird. Sie können in diesem Dialog den Inhalt der ausgewählten Tabelle einsehen und weiterführend auch ändern.

Ausrichtung

Bestimmen Sie hier die Ausrichtung des Inhalts in dem markierten Feld (rechtsbündig, zentriert oder linksbündig).

Priorität

Legen Sie hier die Priorität des Layerfeldes fest. Sie können Prioritätswerte zwischen -9 und +9 vergeben. Falls die Gesamtbreite des Layers nicht ausreichen sollte, um die Inhalte aller Layerfelder darzustellen, entscheidet die Priorität darüber, von welchen Layerfeldern der Inhalt vorrangig dargestellt wird. Zunächst wird der Inhalt des Feldes mit der höchsten Priorität nach Möglichkeit vollständig dargestellt. Dann wird der Inhalt von Feldern mit niedrigeren Prioritäten möglichst vollständig dargestellt. Kann der Inhalt eines Feldes nicht ganz dargestellt werden, wird er abgeschnitten oder unterdrückt, je nach Einstellung unter **Abgeschnittenen Text unterdrücken**.

Abgeschnittenen Text unterdrücken

Bestimmen Sie hier, ob ein Text, der nicht ganz in das Feld passt, unterdrückt oder abgeschnitten werden soll.

Schriftfarbe

Die Schriftfarbe des Feldes wird hier angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:



Die Farbzuidnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.



Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Farbe vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt ().

Schriftart

Die Schriftart des Feldes wird hier angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:



Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**.



Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Schriftart vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt ().

selektierte Eigenschaft in alle Felder übernehmen



Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

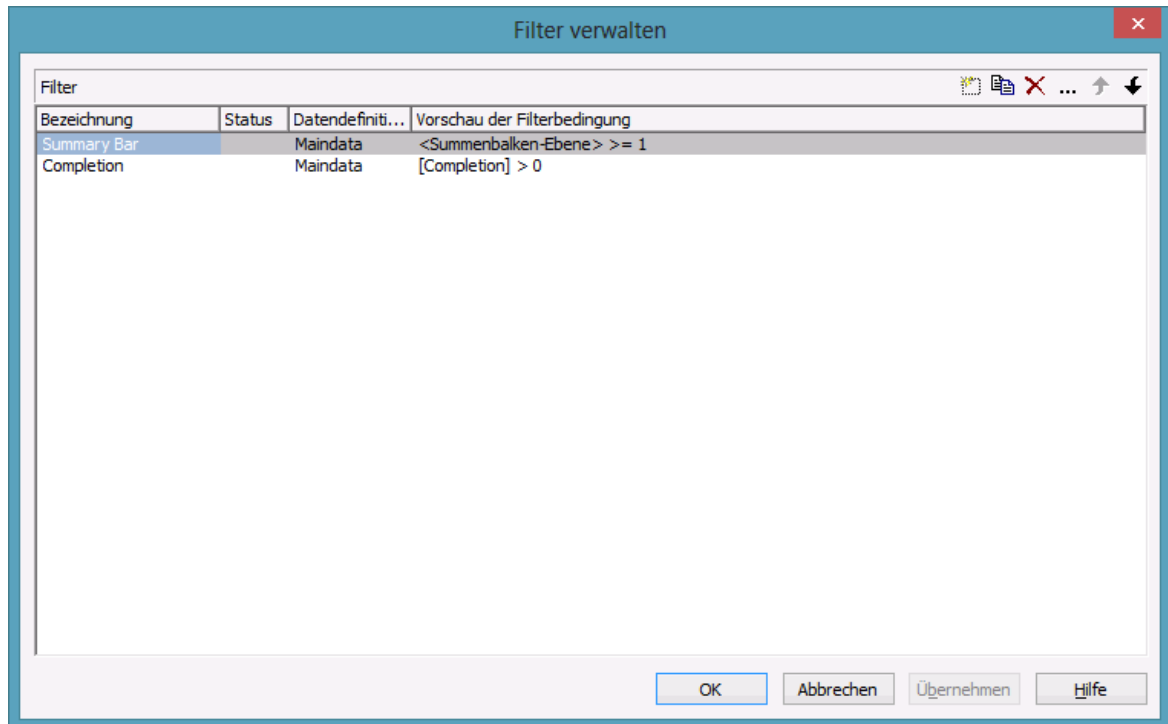
Vorschau

Hier werden die aktuellen Felder dargestellt. Wenn Sie im Vorschaufenster ein Feld anklicken, können Sie dessen Attribute in der **Felder**-Tabelle bearbeiten.



Mit Hilfe der Schaltflächen oberhalb der Vorschau können Sie neue Felder anlegen oder das markierte Feld löschen. Wenn Sie Felder außerhalb des Layers anlegen möchten, so ist zusätzlich die Strg-Taste zu drücken. Zum Löschen von Feldern können Sie auch die Entf-Taste benutzen.

4.16 Dialogfeld "Filter verwalten"



Sie erreichen dieses Dialogfeld

- von der Eigenschaftenseite **Objekte**
- für Layer: vom Dialogfeld **Balkenaussehen festlegen**
- für Tabellenformate: vom Dialogfeld **Tabelle bearbeiten**
- für Verbindungen: von der Eigenschaftenseite **Verbindungen** über die **Filter**-Schaltfläche
- für Histogrammkurven: vom Dialogfeld **Histogramm bearbeiten** über die **Filter**-Kombobox
- für Knoten: von der Eigenschaftenseite **Knoten** über die **Filter**-Schaltfläche

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Filter. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jeder Filter gekennzeichnet, der seit dem Aufruf des Dialogs hinzugefügt (🌟) oder geändert (⚠️) worden ist.

Vorschau der Filterbedingung

In dieser Spalte wird die Bedingung jedes Filters angezeigt. Sie kann hier nicht editiert werden. Um die Filterbedingung zu bearbeiten, klicken Sie auf die **Filter bearbeiten**-Schaltfläche.

Filter hinzufügen



Ein neuer Filter mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Neue Filter werden kontextabhängig angelegt, d. h. es wird immer die jeweils passende Datendefinitionstabelle verwendet.

Filter kopieren



Der markierte Filter wird kopiert.

Filter löschen



Der Filter, den Sie in der Liste markiert haben, wird gelöscht. Es können nur Filter gelöscht werden, die zur Zeit nicht benutzt werden.

Filter bearbeiten



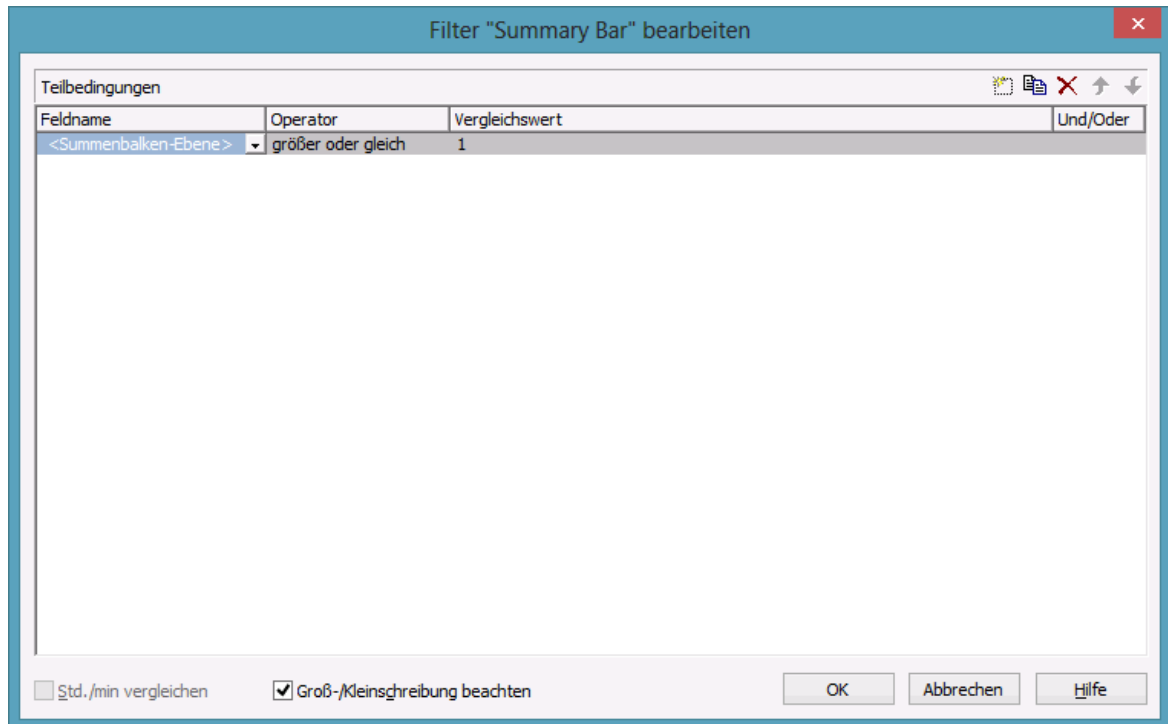
Um die Bedingungen eines Filters anzusehen oder zu ändern, klicken Sie auf die Schaltfläche **Filter bearbeiten**. Es erscheint das Dialogfeld **Filter bearbeiten**. Nun können Sie die für den Filter formulierten Bedingungen bearbeiten.

Filter eine Zeile nach oben/unten



Mit Hilfe dieser Schaltflächen können Sie den markierten Filter eine Zeile nach oben/unten schieben.

4.17 Dialogfeld "Filter bearbeiten"



Sie erreichen dieses Dialogfeld in dem Sie entweder

- von der Eigenschaftenseite **Objekte**
- aus dem Dialog **Knotenaussehen verwalten oder**
- aus dem Dialog **Verbindungsaussehen verwalten**

den **Filter verwalten**-Dialog aktivieren und anschließend die Schaltfläche **Filter bearbeiten** anklicken. In der Kopfzeile dieses Dialogfeldes steht der Name des aktuellen Filters.

Teilbedingung hinzufügen

 Vor der markierten Zeile wird eine neue Zeile für eine Teilbedingung eingefügt.

Teilbedingung kopieren

 Die markierte Teilbedingung wird kopiert und in der Liste eingefügt.

Teilbedingung löschen

 Die markierte Teilbedingung wird gelöscht.

Teilbedingung früher/später abarbeiten

 Wenn ein Filter mehrere Teilbedingungen enthält, werden die einzelnen Teilbedingungen in der Reihenfolge, in der sie in der **Teilbedingungen**-Tabelle stehen, abgearbeitet.

Klicken Sie die Schaltfläche **Teilbedingung früher/später abarbeiten** an, um die markierte Teilbedingung in der Tabelle eine Position höher/tiefer zu setzen, damit sie früher/später abgearbeitet wird.

Feldname

Hier können Sie das Datenfeld auswählen, das mit dem Vergleichswert verglichen werden soll. Die Liste enthält alle verfügbaren Datenfelder sowie eine Reihe von vordefinierten Einträgen:

- Der Eintrag <Gantt: Summenbalken-Ebene> kann für Gantt-Diagramme verwendet werden, um die Summenbalken darzustellen. Definieren Sie beispielsweise den Filter "<Gantt: Summenbalken-Ebene> größer oder gleich 1" und weisen Sie ihn einem Layer (z.B. "Summenbalkenebene 1") zu, um damit die Summenbalken der 1. Ebene darzustellen. Beachten Sie bitte, dass im Dialog **Gruppierung bearbeiten** die Option **Summenbalken anzeigen** aktiviert sein muss, damit die Summenbalken tatsächlich dargestellt werden.
- Filter mit dem Eintrag <Gantt: Gruppierungs-Ebene> können beispielsweise für Gantt-Diagramme im Dialog **Tabelle bearbeiten** als Zeilenfilter für Basiszeilen verwendet werden.
- <Gantt: Kollabiert>: für kollabierte Gruppen
- <Gantt: Jeder Knoten in eigener Zeile>: für eine Darstellung, bei der jeder Knoten in einer eigenen Zeile dargestellt wird
- <Gantt: Knoten überlappend>: für eine Darstellung bei der Knoten einander ggf. überlappen
- <Gantt: Zeile>: Eintrag, um Filter zeilenweise definieren zu können
- <Gantt: Sammelvorgang>: Eintrag, um Filter für Sammelvorgänge definieren zu können
- <Knoten schreibgeschützt>: Eintrag, um Filter für schreibgeschützte Knoten definieren zu können

Diese Option kann auch zur Laufzeit über die VcFilterSubCondition-Eigenschaft **DataFieldIndex** gesetzt werden.

Operator

Wählen Sie hier den Vergleichsoperator für den Vergleich des unter **Feldname** gewählten Datenfeldes mit dem unter **Vergleichswert** angegebenen Wertes bzw. Datenfeldes aus.

Vergleichswert

Hier wird der aktuelle Vergleichswert angezeigt. Im Feld **Vergleichswert** werden in eckigen Klammern alle Felder des passenden Datentyps angeboten, die jeweils als Vergleichswert eingesetzt werden können. Wurde unter **Feldname** beispielsweise das Feld "Frühester Anfang" eingetragen, werden als Vergleichswerte nur Terminfelder (z. B. "Frühestes Ende") und die Optionen <heute> und <Eingabe> angeboten.

Mit Hilfe des Vergleichswertes <Eingabe> können Sie einen variablen Filter definieren. In variablen Filtern sind nur der Feldname und der Operator, aber nicht der Vergleichswert definiert. Diesen können Sie bei Bedarf angeben. Sie können einen variablen Filter verwenden, wenn Sie ein Projekt öffnen und die darzustellenden Vorgänge begrenzen möchten.

Termine werden in dem Format, das auf der Eigenschaftenseite **Allgemeines** unter **Datumsausgabeformat** festgelegt wurde, eingegeben. Wenn Sie unter **Feldname** ein Terminfeld gewählt haben, erscheinen im Feld **Vergleichswert** zwei Pfeil-Schaltflächen, sobald Sie dieses Feld anklicken. Über die erste Pfeil-Schaltfläche öffnen Sie eine Kombobox, die Ihnen alle verfügbaren Termin-Datenfelder anzeigt. Über die zweite Pfeil-Schaltfläche öffnen Sie den Datumsdialog, aus dem Sie das gewünschte Datum per Mausklick auswählen können. Sie können das Datum aber auch direkt editieren.

Zahlenwerte oder Texte müssen direkt eingegeben werden.

Bei den Operatoren "gleich" und "ungleich" können Sie für Textfelder auch Wildcards verwenden:

*: kein oder mehrere beliebige Zeichen

?: genau ein beliebiges Zeichen

Wenn Sie die Zeichen * oder ? nicht als Wildcards verwenden, sondern auf diese Zeichen abfragen wollen, müssen Sie dies durch einen vorangestellten Backslash kenntlich machen:

*: *

\?: ?

Wenn dem Backslash kein * oder ? folgt, wird nach dem Vorhandensein von \ gefragt.

Beispiele:

Vorgang 1 : Name = "Baugenehmigung"

Vorgang 2 : Name = "*Baugenehmigung"

Mögliche Filter für Vorgang 1:

[Name] = B*

[Name] = B?ugenehmigung

Mögliche Filter für Vorgang 2:

[Name] = *B*

[Name] = **

[Name] = ?B*

Und/Oder

Hier wird die logische Verknüpfung der aktuellen mit der jeweils nächsten Teilbedingung in der Tabelle angezeigt.

Wenn Sie den UND-Operator wählen, werden nur die Objekte ausgewählt, die beide Teilbedingungen erfüllen. Wählen Sie den ODER-Operator, werden die Objekte ausgewählt, die mindestens eine der verknüpften Teilbedingungen erfüllen.

Haben Sie mehrere Teilbedingungen formuliert und diese zum Teil mit UND, zum Teil mit ODER verknüpft, werden erst die UND-Verknüpfungen abgearbeitet. (UND bindet stärker als ODER.)

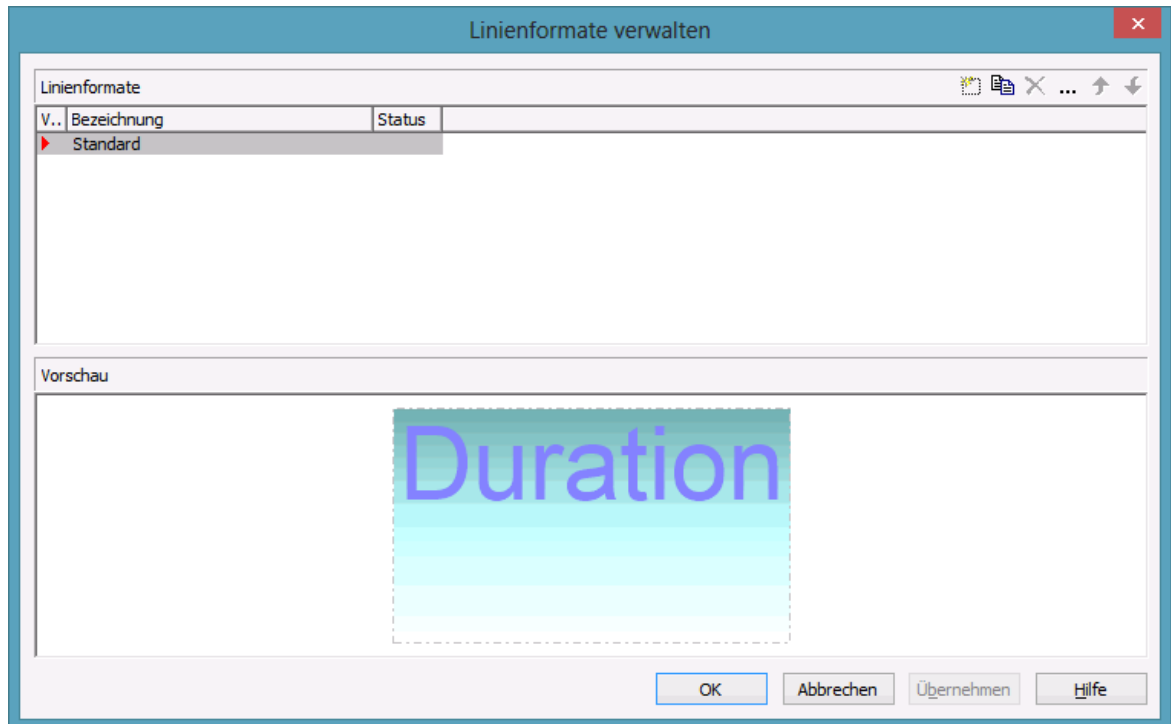
Std./min vergleichen

Aktivieren Sie dieses Kontrollkästchen, wenn beim Datumsvergleich auch Stunde und Minute berücksichtigt werden sollen.

Groß-/Kleinschreibung beachten

Aktivieren Sie dieses Kontrollkästchen, wenn beim Vergleich die Groß-/Kleinschreibung beachtet werden soll.

4.18 Dialogfeld "Linienformate verwalten"



Sie erreichen dieses Dialogfeld

- von der Eigenschaftenseite **Objekte**
- aus dem Dialog **Liniengitter verwalten**

Vorschau

In dieser Spalte wird das jeweils im Vorschaufenster angezeigte Linienformat gekennzeichnet.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Linienformate. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jeder Filter gekennzeichnet, der seit dem Aufruf des Dialogs hinzugefügt (☀️) und/oder geändert (⚠️) worden ist.

Linienformat hinzufügen



Ein neues Linienformat mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Linienformat kopieren



Das markierte Linienformat wird kopiert.

Linienformat löschen



Der Filter, den Sie in der Liste markiert haben, wird gelöscht. Es können nur Filter gelöscht werden, die zur Zeit nicht benutzt werden.

Linienformat bearbeiten



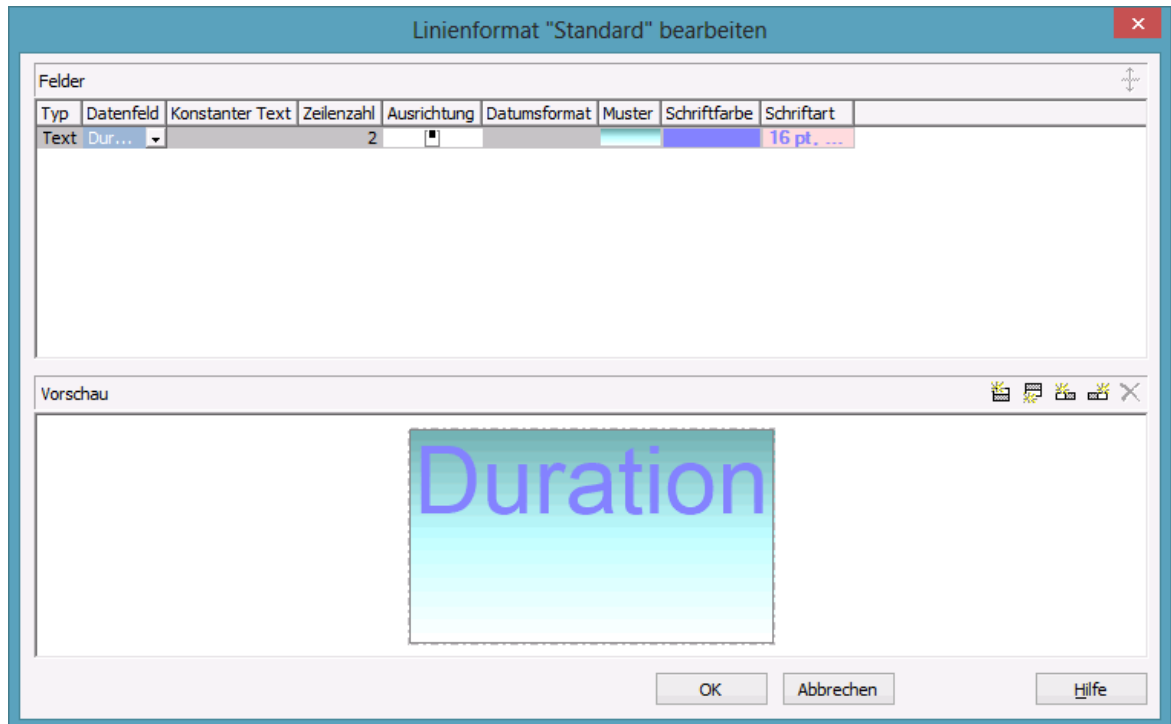
Sie öffnen den Dialog **Linienformat bearbeiten**, in dem Sie die Attribute des Linienformats (Farbe, Muster etc.) auswählen können

Filter eine Zeile nach oben/unten



Mit Hilfe dieser Schaltflächen können Sie das markierte Linienformat eine Zeile nach oben/unten schieben.

4.19 Dialogfeld "Linienformat bearbeiten"



Sie erreichen dieses Dialogfeld

- durch Klick auf **Linienformate** auf der Eigenschaftenseite **Objekte** und im dann folgenden Dialog **Linienformate verwalten** auf die Schaltfläche **...**
- aus dem Dialog **Liniengitter verwalten**

Typ

Hier wird der Feldtyp (Text) angezeigt.

Datenfeld

Wählen Sie hier das Datenfeld, dessen Inhalt für die Beschriftung der Liniengitter genutzt werden soll. Außer den Datenfeldern, die Sie in der Datendefinition vereinbart haben, gibt es die Optionen **<Datum>** und **<Gruppentitel>**: Dabei wird das aktuelle Datum bzw. bei einer bestehenden Gruppierung der Gruppentitel ausgegeben.

Passt der Inhalt des Datenfeldes nicht in das Feld, wird der Überhang bei der Ausgabe abgeschnitten.

Konstanter Text

(nur wenn kein Datenfeld gewählt wurde) Sie können hier einen konstanten Text eingeben, der in dem Feld ausgegeben werden soll.

Zeilenanzahl

Bestimmen Sie die Anzahl der Textzeilen des aktuellen Feldes.

Ausrichtung

Bestimmen Sie hier die Ausrichtung des Inhalts in dem markierten Feld (rechtsbündig, zentriert oder linksbündig).

Datumsformat

Wenn Sie als Datenfeld für die Beschriftung der Liniengitter <Datum> gewählt haben, können Sie hier für dieses Datum ein Ausgabeformat wählen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM: Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- MM: zweistellige Monatsnummer: 01-12
- MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)
- YY: zweistellige Jahreszahl
- YYYY: vierstellige Jahreszahl
- WW: zweistellige Nummer der Kalenderwoche: 01-53
- TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- Q: einstellige Quartalsnummer: 1-4
- TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

- angepasst werden)
- hh: Stunde zweistellig im 24-Stunden-Format: 00-23
- HH: Stunde zweistellig im 12-Stunden-Format: 01-12
- Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- mm: Minute zweistellig: 00-59
- ss: Sekunde zweistellig: 00-59
- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- xC/XC: Sie können das Datum als maximal zehngliedrige, einfache Aufwärtsszählung ab einem bestimmten Referenzdatum gestalten, z.B. "15:05:07:16:00". Dieses Beispiel datiert 15 Monate, 5 Tage, 7 Stunden, 16 Minuten und 0 Sekunden später als das gesetzte Referenzdatum. Die Notation dazu ist: **xC44:C33:C22:C11:C00**. Es sollen je 2 Stellen für die Glieder 4...0 vorgesehen werden, mit einem vorausgehenden "-" Symbol falls der Wert negativ ist. Die Trennzeichen sind variabel und könnten durch andere ersetzt werden. "x" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, aber kein "+"Symbol, falls er positiv ist. "X" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, und ebenso ein "+"Symbol, falls er positiv ist. Im Dialog **Zeitskalenabschnitt ... bearbeiten** sollten die Felder **Referenzdatum verwenden** und **Hauptmarkierung am Referenzdatum ausrichten** aktiviert sein, ebenso sollte **Serielle Beschriftungen** auf **Keine** gesetzt sein. Das Referenzdatum wird zur Laufzeit über die Methode **VcRibbon.set ReferenceDate** gesetzt und überschreibt die Werte des Dialogs.

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

Muster

Legen Sie hier Hintergrundfarbe und -muster fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**. Hier können Sie durch Klick auf  neben dem jeweiligen Attribut ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.

Durch Klick auf die Schaltfläche  gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung des Musters und der Farben vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt ().

Schriftfarbe

Die Schriftfarbe des Feldes wird hier angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Die Farbzuoordnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Farbe vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

Schriftart

Hier wird die aktuelle Schriftart angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Schriftart vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

selektierte Eigenschaft in alle Felder übernehmen

 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

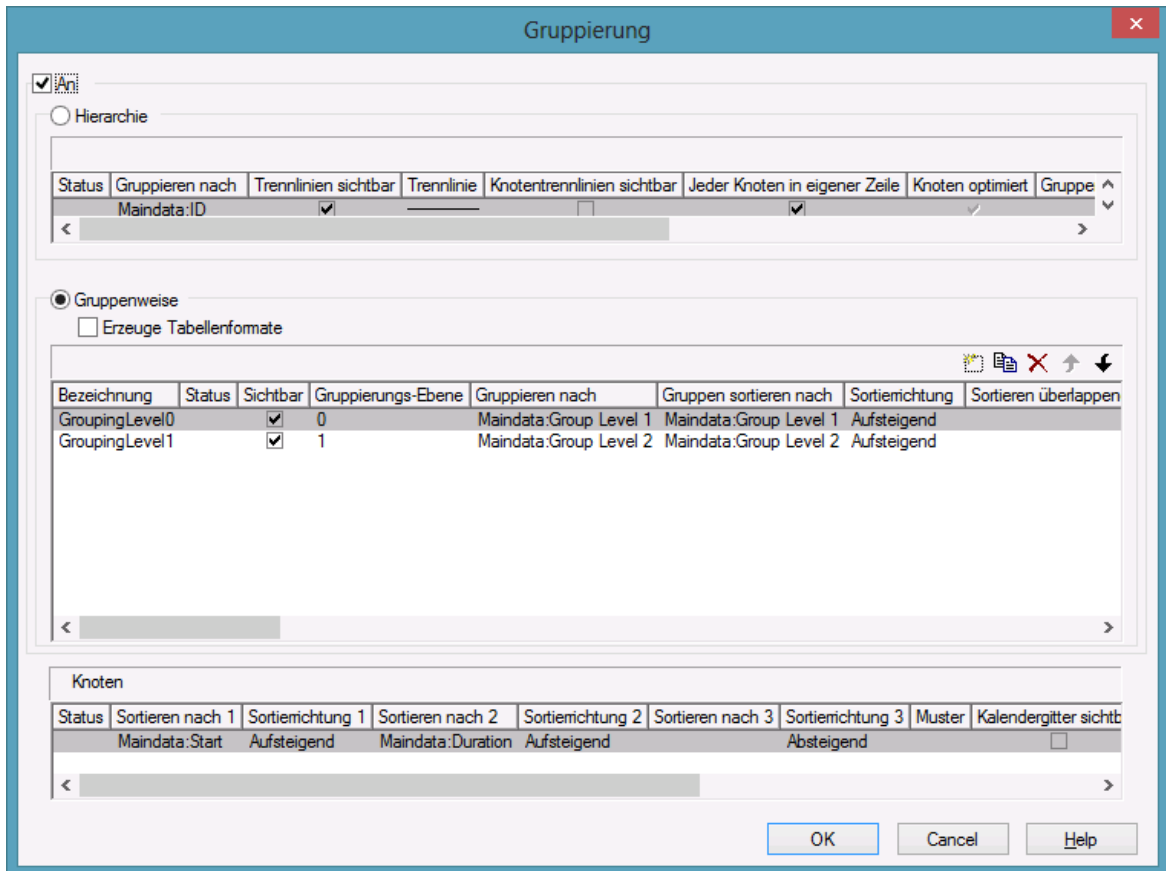
Vorschau

Hier werden die aktuellen Beschriftungen dargestellt.



Mit Hilfe der Schaltflächen oberhalb der Vorschau können Sie neue Felder anlegen oder das markierte Feld löschen. Zum Löschen von Feldern können Sie auch die **Entf**-Taste benutzen.

4.20 Dialogfeld "Gruppierung"



In diesem Dialog können Sie die Optionen zur Hierarchie, Gruppierung, Sortierung und zum Layout dieser Strukturen wählen.

Der Dialog enthält drei Bereiche: **Hierarchie**, **Gruppenweise** und **Knoten**, in denen Sie die entsprechenden Eigenschaften setzen können.

An

Das Gruppieren von Knoten in Form einer Hierarchie (mit einem Hierarchie-Code) oder das Gruppieren nach anderen Kriterien werden hier grundsätzlich an- oder ausgeschaltet.

> Hierarchie

Hier können Sie festlegen, ob die Anordnung der Vorgänge hierarchisch (nach einem Hierarchie-Code) erfolgen soll. Die einzelnen Hierarchieebenen werden im Hierarchie-Code durch Punkte voneinander getrennt. Wenn Sie diese Option wählen, wird die **Gruppenweise** Anordnung automatisch deaktiviert.

In der Tabelle unterhalb der Option **Hierarchie** können Sie weitere Einstellungen für die hierarchische Anordnung vornehmen.

Gruppieren nach

Legen Sie das Feld fest, nach dem die Vorgänge hierarchisch angeordnet werden sollen.

Trennlinien sichtbar

Legen Sie hier fest, ob nach jeder Hierarchieebene Trennlinien ausgegeben werden.

Diese Option kann auch über die Eigenschaft **VcHierarchyLevelLayout.ShowSeparationLines** gesetzt werden..

Trennlinie

Durch Klick auf die Schaltfläche  öffnet sich der Dialog **Linienattribute**, in dem Sie Art und Aussehen der Trennlinien bestimmen können.

Die dort wählbaren Linienattribute lassen sich auch über die entsprechenden Eigenschaften **VcHierarchyLevelLayout.SeparationLineColor**, **VcHierarchyLevelLayout.SeparationLineThickness** und **VcHierarchyLevelLayout.SeparationLineType** festlegen.

Knoten in Kopfzeilen

Bestimmen Sie hier ob jeder Knoten einer Gruppe in einer eigenen Zeile ausgegeben wird oder nicht.

Wenn die Option gewählt ist, entfällt der Tabellenteil für die Vorgänge und man muss die Layerbeschriftung oder den Tooltip verwenden, um die Vorgänge für den Anwender zu identifizieren.

Knoten überlagernd

Mithilfe dieser Option können Sie bestimmen, ob auf dieser Hierarchieebene das Knotenlayout optimiert ist, oder ob Knoten überlappen.

Rümpfe kollabiert

Wenn Sie diese Option wählen, werden bei Programmstart alle Ebenen ab der zweiten Hierarchieebene kollabiert dargestellt. Sie können danach interaktiv expandiert werden.

Summenbalken

Wenn Sie dieses Kontrollkästchen aktivieren, werden Summenbalken für alle Ebenen angezeigt. Wenn Sie Summenbalken ebenenweise definieren möchten, müssen Sie einen Layer mit einer entsprechenden Filterbedingung (<Summenbalken-Ebene> = ...) definieren.

Diese Option kann auch über die Eigenschaft **VcHierarchyLevelLayout.-SummaryBarsVisible** gesetzt werden.

Automatisches Kollabieren von Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens/einer Gruppe alle Gruppen ausser der gerade berührten kollabiert.

Wiederherstellen von automatisch kollabierten Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens/einer Gruppe die zuvor automatisch kollabierten wiederhergestellt.

Automatisches Expandieren von Gruppen

Wenn Sie diese Option wählen, wird beim interaktiven Verschieben eines Knotens/einer Gruppe die Zielgruppe automatisch expandiert.

Wiederherstellen von automatisch expandierten Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens/einer Gruppe die zuvor automatisch expandierten wiederhergestellt.

Seitenumbruch nach jeder Gruppe

Nach Klick auf  stehen folgende Möglichkeiten zur Auswahl:

- **Kein:** es wird kein Seitenumbruch ausgeführt

- **bei voller Seite:** wenn eine Gruppe durch einen Seitenumbruch getrennt würde, wird der Seitenumbruch bereits nach der vorhergehenden Gruppe durchgeführt
- **nach jeder Gruppe:** nach jeder Gruppe wird ein Seitenumbruch ausgeführt

Diese Optionen können auch über die Eigenschaft **VcHierarchyLevelLayout.PageBreakMode** gesetzt werden.

Maximaler Level für Seitenumbrüche

Geben Sie hier an, bis zu welcher Hierarchieebene höchstens Seitenumbrüche nach den Gruppen durchgeführt werden sollen. Geben Sie hier z. B. eine 4 an, wird ab Ebene 5 kein Seitenumbruch mehr eingefügt.

Ist der Standardwert -1 angegeben, werden auf allen Ebenen Seitenumbrüche durchgeführt.

Diese Option kann auch über die Eigenschaft **VcHierarchyLevelLayout.LevelMaximumForPagebreaks** gesetzt werden.

> Gruppenweise

Hier legen Sie fest, ob die Anordnung der Vorgänge in Gruppen (gruppiert nach unterschiedlichen Kriterien) erfolgen soll. Wenn Sie diese Option wählen, wird die **Hierarchie** -Anordnung automatisch deaktiviert.

Im Bereich unterhalb von **Gruppenweise** können Sie alle weiteren Einstellungen für die Gruppierung - vor allem für das Aussehen (Muster, Kalendergitter, Liniengitter etc.) - treffen. Für jede Gruppierungsebene können eigene Kriterien definiert werden. Über die Schaltflächen  können Ebenen hinzugefügt, kopiert, gelöscht und verschoben werden.

Erzeuge Tabellenformate

Wenn dieses Kontrollkästchen aktiviert ist, werden für jede Gruppierungsebene, die Sie hier neu festlegen, eigene Tabellenformate angelegt: Subtitle_n, Collapsed_n. Sie müssen diese dann ggf. im Dialog **Tabellenformat bearbeiten** noch anpassen, insbesondere das Datenfeld.

Ist dieses Kontrollkästchen nicht aktiviert, werden keine Tabellenformate angelegt, wenn eine neue Gruppierungsebene definiert wird. Sie müssen sie dann ggf. selbst anlegen. Mit Hilfe dieser Option können Sie mit nur zwei Gruppierungstabellenformaten (Subtitle und Collapsed) auskommen, und

diese ggf. für jede Gruppierungsebene über Zuordnungstabellen und Filter modifizieren.

Bezeichnung

Legen Sie hier einen Namen für die Gruppierungsebene fest.

Sichtbar

Wählen Sie, ob die Gruppen dieses Levels angezeigt werden oder nicht.

Gruppierungs-Ebene

Hier wird die Ebene angegeben, für die die in dieser Zeile vorgenommenen Einstellungen gelten sollen. Wenn Sie die Reihenfolge der Ebenen ändern möchten, benutzen Sie die Pfeilschaltflächen oberhalb der Tabelle.

Gruppieren nach

Wählen Sie das Datenfeld aus, nach dem Vorgänge auf der gewählten Gruppierungsebene gruppiert werden sollen. Wenn Sie den Leereintrag wählen, werden die Vorgänge auf der eingestellten Ebene nicht gruppiert.

Gruppen sortieren nach

Wählen Sie hier das Datenfeld aus, nach dem die Gruppen beim Programmstart auf der gewählten Gruppierungsebene sortiert werden sollen. Wird hier nichts eingestellt, ergibt sich die Reihenfolge aus der Reihenfolge der Vorgänge beim Laden.

Sortierrichtung

Legen Sie hier die Sortierungsrichtung auf der gewählten Gruppierungsebene fest (aufsteigend oder absteigend).

Sortieren überlappendender Knoten nach

Wählen Sie hier das Datenfeld aus, nach dem die Knoten der Gruppe, die in einer Zeile angeordnet sind, sortiert werden sollen. Wird hier nichts eingestellt, ergibt sich die Reihenfolge aus dem Anfangsdatum und der Dauer der Vorgänge, d.h. die frühesten und kürzesten Vorgänge liegen am weitesten

vorn. Diese Eigenschaft ist nur wirksam, wenn die Eigenschaft **VcGroup-LevelLayout.NodesArrangedOptimized** auf **False** gesetzt wurde.

Sortierrichtung überlappender Knoten

Legen Sie hier die Sortierungsrichtung der sich überlappenden Knoten fest (aufsteigend oder absteigend).

Sortieren optimierter Knoten nach

Wählen Sie hier das Datenfeld aus, nach dem die Knoten der Gruppe, die in einer Zeile angeordnet sind, sortiert werden sollen. Wird hier nichts eingestellt, ergibt sich die Reihenfolge aus dem Anfangsdatum und der Dauer der Vorgänge, d.h. die frühesten und kürzesten Vorgänge liegen am weitesten vorn. Diese Eigenschaft ist nur wirksam, wenn die Eigenschaft **VcGroup-LevelLayout.NodesArrangedOptimized** auf **True** gesetzt wurde.

Sortierrichtung optimierter Knoten

Legen Sie hier die Sortierungsrichtung der optimierte Knoten fest (aufsteigend oder absteigend).

Muster

Wenn Sie auf  klicken, öffnet sich das Dialogfeld **Musterattribute**. Hier können Sie ein Hintergrundmuster und zwei Musterfarben für die Gruppentitelzeile festlegen sowie über die Schaltfläche  eine datenabhängige Zuordnung der jeweiligen Eigenschaft vereinbaren.

Kalender

Wählen Sie hier das Datenfeld aus, das den Namen des Kalenders enthält, der für den Gruppenknoten verwendet werden soll.

Liniengitter sichtbar

Hier können Sie festlegen, ob ein Liniengitter angezeigt wird.

Liniengitter mit Untergruppen

Wählen Sie, ob das Liniengitter auch für Untergruppen dargestellt werden soll.

Liniengitter

Hier können Sie entweder aus der Drop-Down-Liste ein bestehendes Liniengitter für die Gruppierungsebene auswählen oder im Dialog **Liniengitter verwalten** (zu erreichen über ) ein neues Liniengitter anlegen. Nähere Informationen zu diesem Dialog finden Sie im Kapitel **Dialogfeld Liniengitter verwalten**.

Kalendergitter sichtbar

Bestimmen Sie, ob ein Kalendergitter angezeigt wird. Bitte beachten Sie, dass als Voraussetzung dafür die Eigenschaft **VcCalendarGrid.Visible** auf **True** gesetzt sein muss.

Kalendergitter mit Untergruppen

Wählen Sie, ob das Kalendergitter auch für Untergruppen dargestellt werden soll.

Kalendergitter

Hier können Sie entweder aus der Drop-Down-Liste ein bestehendes Kalendergitter für die Gruppe auswählen oder im Dialog **Kalendergitter verwalten** (zu erreichen über ) ein neues Kalendergitter anlegen. Nähere Informationen zu diesem Dialog finden Sie im Kapitel **Dialogfeld Kalendergitter verwalten**.

Bei Auswahl der Option <aus der Skala> wird das erste nicht sichtbare Kalendergitter aus der Zeitskala angezeigt.

Trennlinien sichtbar

Wenn Sie diese Option wählen, wird bei jedem Gruppenwechsel eine Trennlinie ausgegeben.

Trennlinien oben

Wenn Sie diese Option wählen, wird die Trennlinie oberhalb von Gruppen verschiedener Gruppierungsebenen gezogen (statt unterhalb).

Trennlinie

Sie können das Aussehen der Trennlinien bearbeiten, indem Sie auf die **Bearbeiten**-Schaltfläche klicken und so das Dialogfeld **Linie bearbeiten** öffnen.

Knoten in Kopfzeilen

Bestimmen Sie hier ob jeder Knoten einer Gruppe in einer eigenen Zeile ausgegeben wird oder nicht.

Wenn diese Option gewählt ist, entfällt der Tabellenteil für die Vorgänge und man muss die Layerbeschriftung oder den Tooltip verwenden, um die Vorgänge für den Anwender zu identifizieren.

Knoten überlagernd

Mithilfe dieser Option können Sie bestimmen, ob auf dieser Gruppierungsebene das Knotenlayout optimiert ist, oder ob Knoten überlappen.

Rümpfe kollabiert

Wenn Sie diese Option wählen, werden die einzelnen Gruppen zunächst kollabiert dargestellt, d. h. es sind nur die Gruppentitel sichtbar, aber nicht die Vorgänge.

Änderungen erlaubt

Wenn Sie dieses Kontrollkästchen aktivieren, kann der Anwender expandierte Gruppen kollabieren oder kollabierte Gruppen wieder expandieren. Der Anwender kann Gruppen durch einen Doppelklick auf die Gruppenüberschrift im Tabellenteil oder durch einen einfachen Klick auf das Minus- bzw. Plus-Zeichen neben der Gruppenüberschrift oder über das Kontextmenü für Gruppen kollabieren bzw. expandieren.

Summenbalken

Wenn Sie dieses Kontrollkästchen aktivieren, werden Summenbalken angezeigt. Um Summenbalken ebenenweise zu definieren, müssen Sie einen Layer mit einer entsprechenden Filterbedingung (<Summenbalken-Ebene> = ...) definieren.

Gruppenknoten sichtbar

Wenn Sie diese Option wählen, wird für Gruppen, für die eine eigene Datentabelle angelegt wurde, im Diagramm je ein Balken dargestellt. Aktivieren Sie dazu vorher auch die Option **Erweiterte Datentabellen zulassen** auf der Eigenschaftenseite **Allgemeines**.

Vertikal Verschieben über den Tabellenbereich

Wenn diese Option ausgewählt ist, können Sie Gruppen im Tabellenbereich mit der Maus verschieben und damit ihre Reihenfolge ändern.

Automatisches Kollabieren von Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens alle Gruppen ausser der gerade berührten kollabiert.

Wiederherstellen von automatisch kollabierten Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens die zuvor automatisch kollabierten wiederhergestellt.

Automatisches Expandieren von Gruppen

Wenn Sie diese Option wählen, wird beim interaktiven Verschieben eines Knotens die Zielgruppe automatisch expandiert.

Wiederherstellen von automatisch expandierten Gruppen

Wenn Sie diese Option wählen, werden beim interaktiven Verschieben eines Knotens die zuvor automatisch expandierten wiederhergestellt.

Vertikal Verschieben über den Diagrammbereich

Wenn Sie diese Option auswählen, können Sie Gruppen im Diagrammbereich mit der Maus verschieben und damit ihre Reihenfolge ändern.

Seitenumbruch nach jeder Gruppe

Nach Klick auf  stehen folgende Möglichkeiten zur Auswahl:

- **Kein:** es wird kein Seitenumbruch ausgeführt
- **bei voller Seite:** wenn eine Gruppe durch einen Seitenumbruch getrennt würde, wird der Seitenumbruch bereits nach der vorhergehenden Gruppe durchgeführt
- **nach jeder Gruppe:** nach jeder Gruppe wird ein Seitenumbruch ausgeführt

Diese Optionen können auch über die Eigenschaft **VcGroupLevelLayout.-PageBreakMode** gesetzt werden.

> Knoten

Die folgenden Einstellungen beschreiben die Optionen, die Sie für ungruppierte Knoten oder Knoten in Gruppen auswählen können. Dies betrifft vor allem die Sortierung sowie das Aussehen der Knotenzeilen.

Hinweis: Beachten Sie bitte, dass die Einstellungen nur für die Sortierung der Vorgänge beim Öffnen des Diagramms gelten. Wenn Sie die Vorgänge danach neu sortieren möchten, verwenden Sie dazu bitte die VcGantt-Methode **SortNodes**.

Sortieren nach 1-3

Legen Sie hier fest, nach welchen Datenfeldern die Vorgänge beim Öffnen des Diagramms sortiert werden. Sie können bis zu drei Datenfelder als Sortierungskriterien auswählen. Für jedes dieser Sortierungskriterien können Sie auswählen, ob absteigend oder aufsteigend sortiert werden soll (**Sortierreihenfolge 1 bis 3**). Bei Angabe eines Feldes für **gruppiert nach** erfolgt die Sortierung innerhalb jeder Gruppe.

Muster

Wenn Sie auf  klicken, öffnet sich das Dialogfeld **Musterattribute**. Hier können Sie ein Hintergrundmuster und zwei Musterfarben für die Knotenzeile festlegen sowie über die Schaltfläche  eine datenabhängige Zuordnung der jeweiligen Eigenschaft vereinbaren.

Kalendergitter sichtbar

Bestimmen Sie, ob ein Kalendergitter angezeigt wird.

Kalendergitter

Hier können Sie entweder aus der Drop-Down-Liste ein bestehendes Kalendergitter für den Knoten auswählen oder im Dialog **Kalendergitter verwalten** (zu erreichen über ) ein neues Kalendergitter anlegen. Nähere Informationen zu diesem Dialog finden Sie im Kapitel **Dialogfeld Kalendergitter verwalten**.

Bei Auswahl der Option <aus der Skala> wird das erste nicht sichtbare Kalendergitter aus der Zeitskala angezeigt.

Trennlinien sichtbar

Bestimmen Sie, ob eine Trennlinie angezeigt wird.

Trennlinien oben

Wenn Sie diese Option wählen, wird die Trennlinie oberhalb eines Knotens gezogen (statt unterhalb).

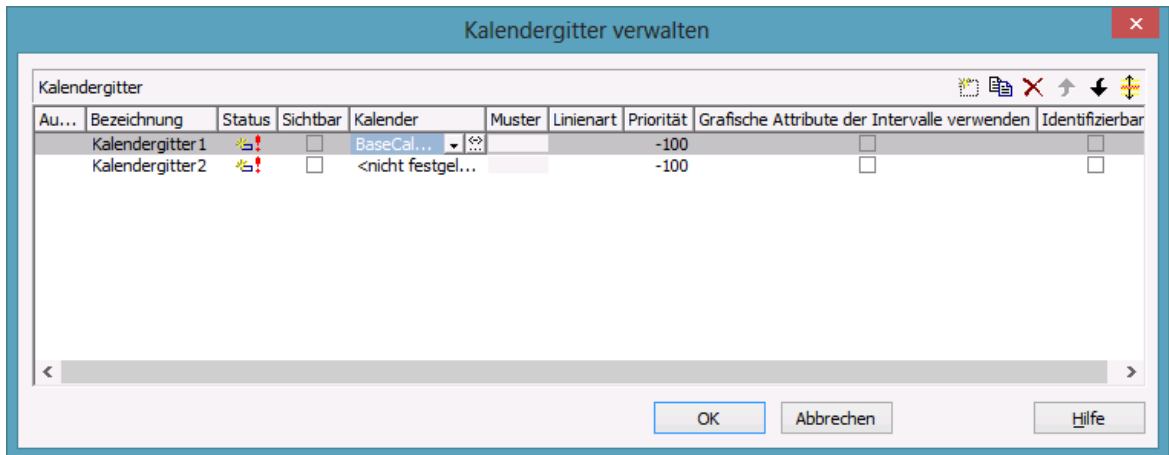
Trennlinie

Sie können das Aussehen der Trennlinien bearbeiten, indem Sie auf die **Bearbeiten**-Schaltfläche klicken und so das Dialogfeld **Linie bearbeiten** öffnen.

Trennlinienschrittweite

Hier können Sie festlegen, nach wie vielen Vorgängen jeweils eine Trennlinie gezogen werden soll.

4.21 Dialogfeld "Kalendergitter verwalten"



Sie gelangen in diesen Dialog, indem Sie im Dialog **Gruppierung** im Bereich **Gruppenweise** im Feld **Kalendergitter** auf **...** klicken.

Mithilfe der Schaltflächen    können Sie Kalendergitter hinzufügen, kopieren oder löschen.

Mit den Pfeilschaltflächen   können Sie ein Kalendergitter nach unten oder oben verschieben, während die Schaltfläche  es ermöglicht, die ausgewählte Eigenschaft allen angezeigten Kalendergittern zuzuweisen.

Folgende Einstellungen können für die Kalendergitter vorgenommen werden:

Ausgewählt

Durch Klicken in dieses Feld können Sie dieses Kalendergitter für die Gruppierungsebene auswählen. Ein roter Pfeil zeigt die Auswahl an.

Bezeichnung

Vergeben Sie hier einen Namen für das Kalendergitter.

Status

Status: In dieser Spalte wird jedes Kalendergitter gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Sichtbar

Aktivieren Sie dieses Kontrollkästchen für jedes Kalendergitter, das angezeigt werden soll.

Kalender

Ein hier ausgewählter Kalender wird auf alle Gruppen dieser Ebene angewendet. Wenn Sie keinen Kalender auswählen, wird der Kalender der Ebene verwendet, der das Kalendergitter zugeordnet ist.

Muster

Legen Sie hier die Hintergrundfarbe und -muster des Kalendergitters fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen können. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung, und Hintergrund der Streifen des Kalendergitters festlegen.

Linienart

Wenn Sie auf die Schaltfläche dieses Feldes () klicken, öffnet sich das Dialogfeld **Linienattribute des Kalendergitters**. Hier können Sie Farbe, Typ und Dicke der Randlinien der vertikalen Linien festlegen.

Priorität

Hier können Sie die Priorität jedes Kalendergitters in Relation zu den anderen Gittern und Layern festlegen (> 0: vor den Layern, < 0: hinter den Layern)

Grafische Attribute der Intervalle verwenden

Legen Sie fest, ob die für die Intervalle eingestellten grafischen Attribute dargestellt werden sollen.

Identifizierbar

Hier können Sie festlegen, ob das Kalendergitter mittels der VcGantt-Methode **IdentifyObjectAt** identifizierbar ist. So kann z.B. ein Tooltip-Text

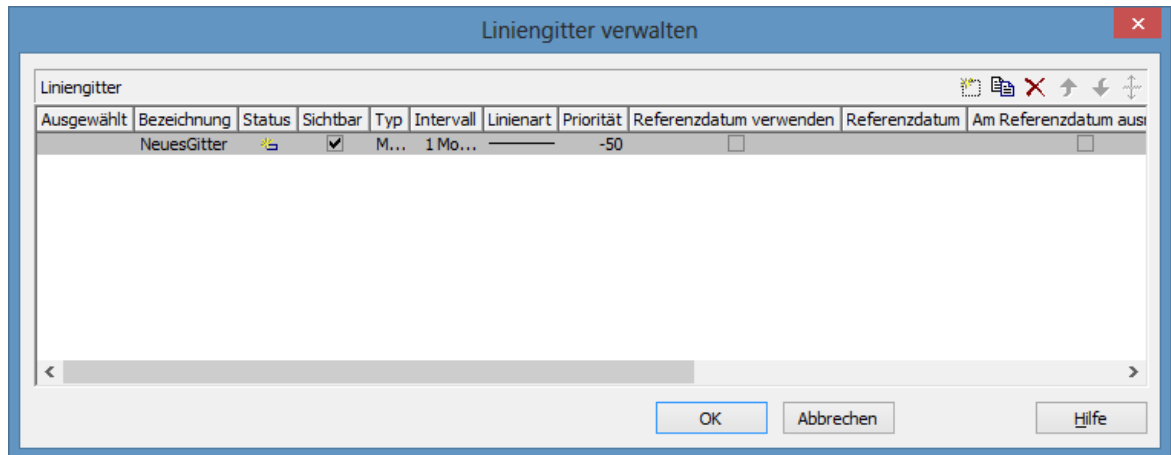
nur bei einem identifizierbaren Gitter erscheinen; Analoges gilt für das Erscheinen des Kontext-Menüs bei einem Rechtsklick.

Für einen Tooltip-Text muss zudem das entsprechende Intervall identifiziert werden: s. **Kalendergitter**-Bereich des Dialogs **Zeitskalenabschnitt bearbeiten**.

Einrastziele Start/Ende

Wenn Sie diese Optionen wählen, definiert das Kalendergitter seine relevanten Positionen als "Einrastziele" für zu verschiebende Knoten/Layer.

4.22 Dialogfeld "Liniengitter verwalten"



Sie gelangen in diesen Dialog, indem Sie im Dialog **Gruppierung** im Bereich **Gruppenweise** im Feld **Liniengitter** auf klicken.

Mithilfe der Schaltflächen können Sie Liniengitter hinzufügen, kopieren oder löschen.

Mit den Pfeilschaltflächen können Sie ein Liniengitter nach unten oder oben verschieben, während die Schaltfläche es ermöglicht, die ausgewählte Eigenschaft allen angezeigten Liniengittern zuzuweisen. Folgende Einstellungen können für die Liniengitter vorgenommen werden:

Ausgewählt

Durch Klicken in dieses Feld können Sie dieses Liniengitter für die Gruppierungsebene auswählen. Ein roter Pfeil zeigt die Auswahl an.

Bezeichnung

Vergeben Sie hier einen Namen für das Liniengitter.

Status

In dieser Spalte wird jedes Liniengitter gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Sichtbar

Aktivieren Sie dieses Kontrollkästchen für jedes Liniengitter, das angezeigt werden soll.

Typ

Hier können Sie die Grundeinheit jedes Liniengitters festlegen, z. B. Tage, Wochen, etc.

Intervall

Hier können Sie das Intervall zwischen den einzelnen Linien in ganzzahligen Vielfachen der Grundeinheit des Liniengitters festlegen.

Linienart

Wenn Sie auf die Schaltfläche dieses Feldes () klicken, öffnet sich das Dialogfeld **Linienattribute des Liniengitters**. Hier können Sie Farbe, Typ und Dicke der Randlinien der Streifen festlegen.

Priorität

Hier können Sie die Priorität jedes Liniengitters in Relation zu anderen Gittern und zu den Layern festlegen (> 0: vor den Layern, < 0: hinter den Layern).

Referenzdatum verwenden

Aktivieren Sie dieses Kontrollkästchen, wenn der Ursprung des Liniengitters auf das Referenzdatum gelegt werden soll.

Referenzdatum

Wählen Sie das Referenzdatum mit Hilfe des Datumsdialogs aus.

Am Referenzdatum ausrichten

Aktivieren Sie dieses Kontrollkästchen, wenn die Linien des Liniengitters am Referenzdatum ausgerichtet werden sollen, d.h. die Position kann vom Anfang einer Zeiteinheit abweichen, z.B. um 13:17 jedes Tages.

Ist diese Option nicht gewählt, werden die Linien eines Liniengitters standardmäßig am Anfang der jeweiligen Zeiteinheit positioniert, z.B. immer um 00:00 Uhr jedes Tages.

Linienformat

Hier können Sie entweder aus der Drop-Down-Liste ein bestehendes Linienformat für das Liniengitter auswählen oder im Dialog **Linienformate vewalten** (zu erreichen über ) ein neues Kalendergitter anlegen. Nähere Informationen zu diesem Dialog finden Sie im Kapitel **Dialogfeld Linienformate verwalten**.

Drehen

Wenn Sie diese Option auswählen, kann die Beschriftungen an den Linien des Liniengitters um 90 Grad (vertikal) gedreht werden.

Ausrichtung

Hier können Sie die horizontale Ausrichtung für die Beschriftungen der Linien festlegen.

Oben

Wenn Sie diese Option wählen, erscheint die Beschriftung der Linien im Liniengitter am oberen Rand des Gantt-Graphen.

Mitte

Wenn Sie diese Option wählen, erscheint die Beschriftung der Linien im Liniengitter in der Mitte des Gantt-Graphen.

Unten

Wenn Sie diese Option wählen, erscheint die Beschriftung der Linien im Liniengitter am unteren Rand des Gantt -Graphen.

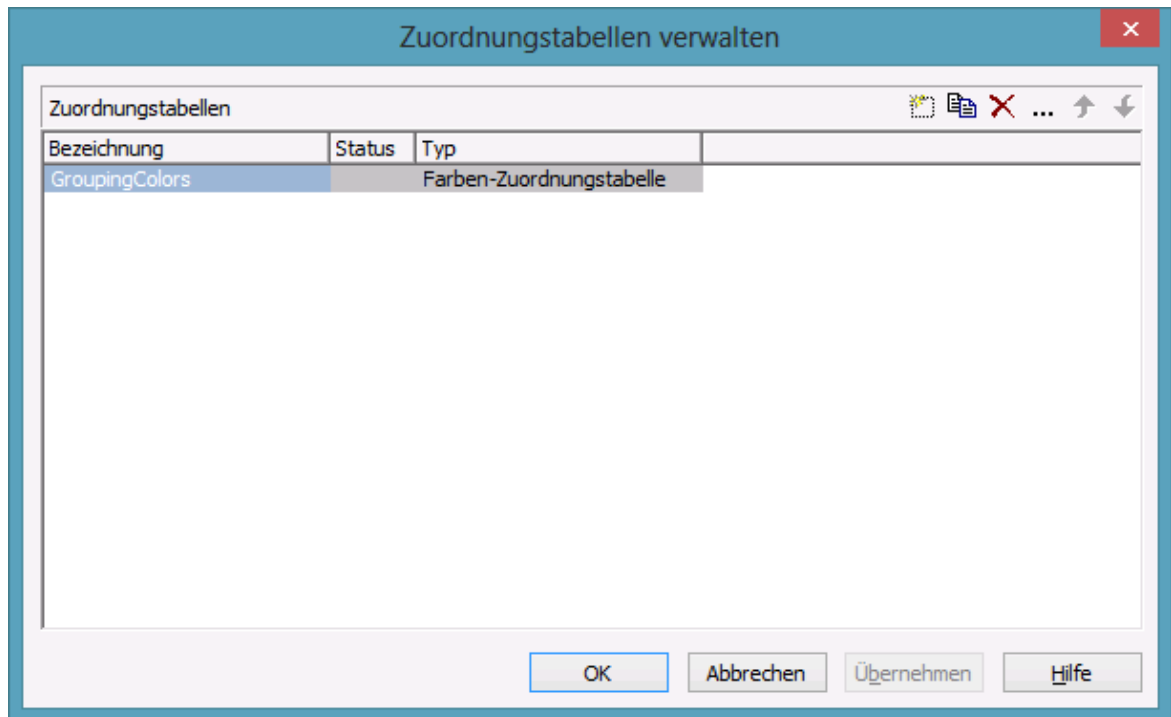
Sommerzeit beachten

Wenn Sie diese Option wählen, wird die Sommerzeit berücksichtigt.

Einrastziel

Wenn Sie diese Option wählen, definiert das Liniengitter seine relevanten Positionen als "Einrastziele" für zu verschiebende Knoten/Layer.

4.23 Dialogfeld "Zuordnungstabellen verwalten"



Sie gelangen über die Eigenschaftenseite **Objekte** in dieses Dialogfeld oder indem Sie im Dialogfeld **Zuordnung einstellen** auf die Schaltfläche **Zuordnungstabellen** klicken.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Zuordnungstabellen. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jede Zuordnungstabelle gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Typ

Wählen Sie hier den Typ der Zuordnungstabelle aus:

- Farben-Zuordnungstabelle
- Schraffuren-Zuordnungstabelle

- Grafikdateien-Zuordnungstabelle
- Schriften-Zuordnungstabelle
- Millimeter-Zuordnungstabelle
- Nummern-Zuordnungstabelle

Zuordnungstabelle hinzufügen



Eine neue Zuordnungstabelle mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Zuordnungstabelle kopieren



Die markierte Zuordnungstabelle wird kopiert.

Zuordnungstabelle löschen



Die Zuordnungstabelle, die Sie in der Liste markiert haben, wird gelöscht. Es können nur die Zuordnungstabellen gelöscht werden, die zur Zeit nicht benutzt werden.

Zuordnungstabelle bearbeiten



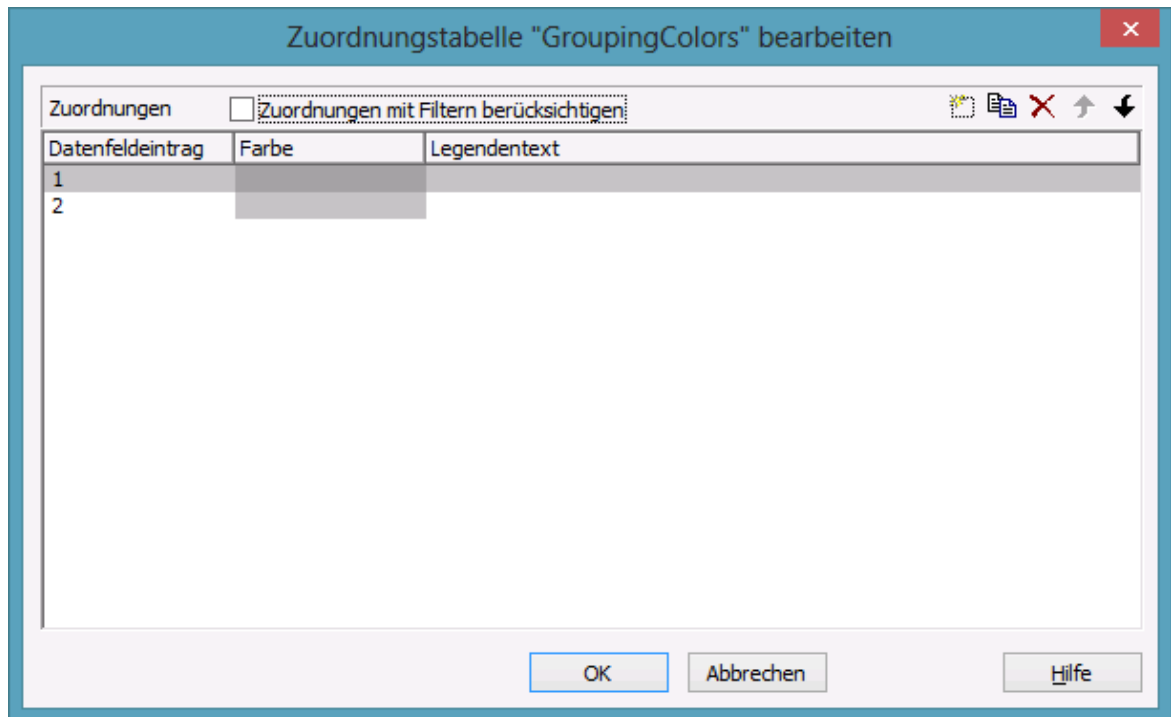
Sie gelangen in das Dialogfeld **Zuordnungstabelle bearbeiten**.

Zuordnungstabelle eine Zeile nach oben / unten



Mit Hilfe dieser Schaltfläche können Sie die markierte Zuordnungstabelle in der Tabelle eine Zeile nach oben/unten schieben.

4.24 Dialogfeld "Zuordnungstabelle bearbeiten"



Sie gelangen in dieses Dialogfeld, indem Sie im Dialogfeld **Zuordnungstabellen verwalten** auf die Schaltfläche **Zuordnungstabelle bearbeiten** (...) klicken.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie mehr Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

Zuordnungen mit Filtern berücksichtigen

Wenn diese Option ausgewählt ist, werden nicht nur die in der Liste der Datenfeldeinträge angegebenen festen Werte als Schlüsselwerte berücksichtigt, sondern auch Filter, die aus der Dropdown-Liste ausgewählt werden können. Dadurch hängen die Ausführungswerte nicht mehr nur von einem konkreten Wert ab, sondern von einem Wertebereich.

Datenfeldeintrag

In dieser Spalte werden die Einträge des gewählten Datenfeldes aufgeführt, für die Sie eine Farbe bzw. ein Muster und den Legendentext vereinbaren können.

Farbe

Um die Farbe für einen Datenfeldeintrag festzulegen, klicken Sie auf das entsprechende Feld. Dann erscheint eine Schaltfläche, über die Sie in den Dialog zur Auswahl der Farbe gelangen. Darin stehen auch transparente Farben zur Verfügung.

Legendentext

Hier können Sie für jeden Datenfeldeintrag einen Legendentext festlegen.

Zuordnung hinzufügen



Eine neue Zuordnung mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Zuordnung kopieren



Die markierte Zuordnung wird kopiert.

Zuordnung löschen



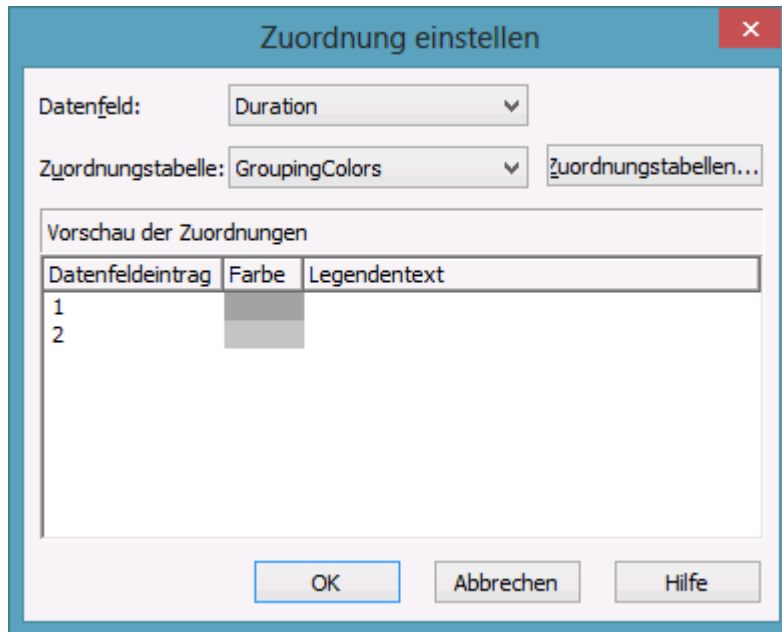
Die Zuordnung, die Sie in der Liste markiert haben, wird gelöscht. Es können nur die Zuordnungen gelöscht werden, die zur Zeit nicht benutzt werden.

Zuordnung eine Zeile nach oben / unten



Die markierte Zuordnung wird in der Tabelle eine Zeile nach oben/unten geschoben.

4.25 Dialogfeld "Zuordnung einstellen"



Verbinden Sie in diesem Dialogfeld das Datenfeld eines Knotens mit einer Zuordnungstabelle. Sie erreichen es über den Dialog **Layer bearbeiten**. Klicken Sie dort beim gewünschten Attribut auf die Schaltfläche .

Datenfeld

Wählen Sie hier das Datenfeld aus, von dessen Einträgen die gewünschten Attribute des zu bearbeitenden Objekts abhängen soll.

Zuordnungstabelle

(nur aktiv, wenn ein Datenfeld gewählt wurde) Wählen Sie hier die Zuordnungstabelle aus, die in Abhängigkeit ihres Typs den einzelnen Datenfeldeinträgen die entsprechenden Attribute zuordnet.

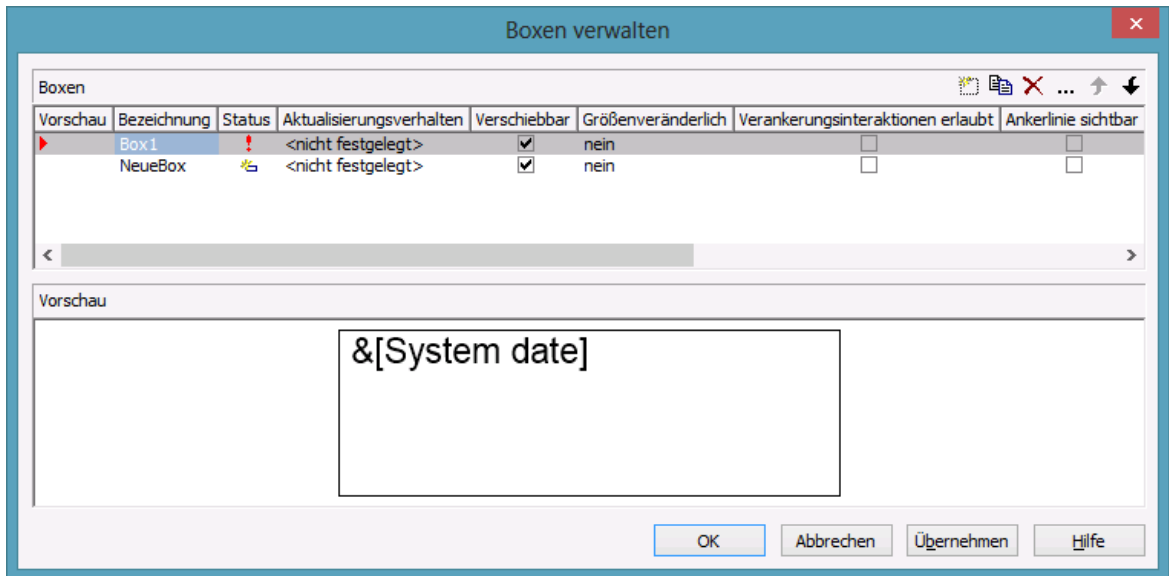
Zuordnungstabellen

Klicken Sie auf diese Schaltfläche, um das Dialogfeld **Zuordnungstabellen verwalten** aufzurufen. Hier können Sie Zuordnungstabellen erstellen, bearbeiten, kopieren oder löschen.

Vorschau der Zuordnungen

Hier wird die gewählte Zuordnungstabelle dargestellt: alle Datenfelder sowie die diesen zugeordneten Farben bzw. Muster und die Legendentexte.

4.26 Dialogfeld "Boxen verwalten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**. Im Diagrammbereich können beliebig viele Boxen dargestellt werden, die Sie in diesem Dialogfeld verwalten können.

Vorschau

Die in dieser Spalte markierte Box wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Boxen. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jede Box gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Aktualisierungsverhalten

Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog **Aktualisierungsverhalten bearbeiten** für Boxen festgelegt wurde.

Verschiebbar

Durch Bewegen einer Box wird ihr Offset verändert. Legen Sie hier fest, ob die Box zur Laufzeit im Diagrammbereich frei beweglich sein soll. Wenn die Box nicht mehr verschiebbar sein soll, nachdem Sie sie positioniert haben, schalten Sie diese Option ab.

Größenveränderlich

Legen Sie hier fest, ob die Größe einer Box interaktiv geändert werden kann, wobei sie wählen können, ob nur die Höhe, nur die Breite oder Höhe und Breite veränderbar sein sollen. Sobald Sie mit der Maus auf den Rand der Box zeigen, wird der Mauszeiger zu einem Doppelpfeil. Vergrößern oder verkleinern Sie nun die Box, indem Sie den Rahmen mit gedrückter linker Maustaste in die gewünschte Richtung ziehen.

Tipp: Wenn Sie **Breite und Höhe** ausgewählt haben, können Sie den Mauszeiger auch auf eine Ecke der Box positionieren und Höhe und Breite gleichzeitig ändern.

Verankerungsinteraktionen erlaubt

Legen Sie fest, ob Ver- oder Entankerungsinteraktionen (per Maus oder Kontextmenü) durchgeführt werden können. Dadurch wird es dem Anwender ermöglicht, Boxen an Knoten zu verankern bzw. sie wieder zu lösen.

Ankerlinie sichtbar

Legen Sie hier fest, ob bei Verankerungen die eingestellten Referenzpunkte beim Knoten und bei der Box durch eine Linie verbunden werden sollen.

Knoten-ID

Hier können Sie eine beliebige Zeichenkette eingeben, die dann als Knoten-ID interpretiert wird und dazu dient, den Knoten zu identifizieren, an dem die jeweilige Box verankert werden soll. Eine leere Zeichenkette bedeutet, dass die Box nicht verankert ist.

Hinweis: Es wird nicht überprüft, ob die Syntax korrekt ist oder ob der Knoten existiert. Wenn der Knoten nicht existiert, wird nicht verankert.

Ursprung

Mithilfe der Eigenschaften **Ursprung**, **Referenzpunkt**, **X-Offset** und **Y-Offset** können Sie jede einzelne Box im Diagrammbereich positionieren, wobei die relative Position der Box zum Diagramm unabhängig von der Diagrammgröße ist.

Legen Sie hier den Ursprung fest, d. h. den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird. Mögliche Werte des Ursprungs: links oben, mittig oben, rechts oben, links mittig, mittig mittig, rechts mittig, links unten, mittig unten, rechts unten.

Referenzpunkt

Legen Sie hier den Referenzpunkt der Box fest, d. h. den Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird. Mögliche Werte des Referenzpunkts: oben links, oben mittig, oben rechts, mittig links, mittig mittig, mittig rechts, unten links, unten mittig, unten rechts.

X-Offset

Legen Sie hier den Abstand zwischen Ursprung und Referenzpunkt in x-Richtung fest (in mm).

Y-Offset

Legen Sie hier den Abstand zwischen Ursprung und Referenzpunkt in y-Richtung fest (in mm).

Rahmen

Wenn Sie auf dieses Feld klicken, erscheint eine **Bearbeiten**-Schaltfläche, über die Sie in den Dialog **Linie bearbeiten** gelangen. Hier können Sie den Typ, die Dicke und die Farbe der Umrandungslinie der Box festlegen.

Priorität

Legen Sie hier die relative Zeichnungspriorität der Box zu anderen Objekten im Diagramm (Knoten, Liniengitter etc.) fest. Die Priorität von Knoten ist 0. Falls die Boxen eine höhere Priorität als die Knoten haben, überdecken sie die Knoten so, dass darauf interaktiv nicht mehr zugegriffen werden kann.

Sichtbar

Legen Sie fest, ob die Box zur Laufzeit sichtbar sein soll.

Boxformat

Hier wird das Boxformat der Box angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:



Aus der Kombobox können Sie ein vorhandenes Boxformat wählen.



Über die **Bearbeiten**-Schaltfläche gelangen Sie in den Dialog **Boxformate verwalten**.

Box hinzufügen



Eine neue Box mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Box kopieren



Die markierte Box wird kopiert.

Box löschen



Die Box, die Sie in der Liste markiert haben, wird gelöscht.

Box bearbeiten



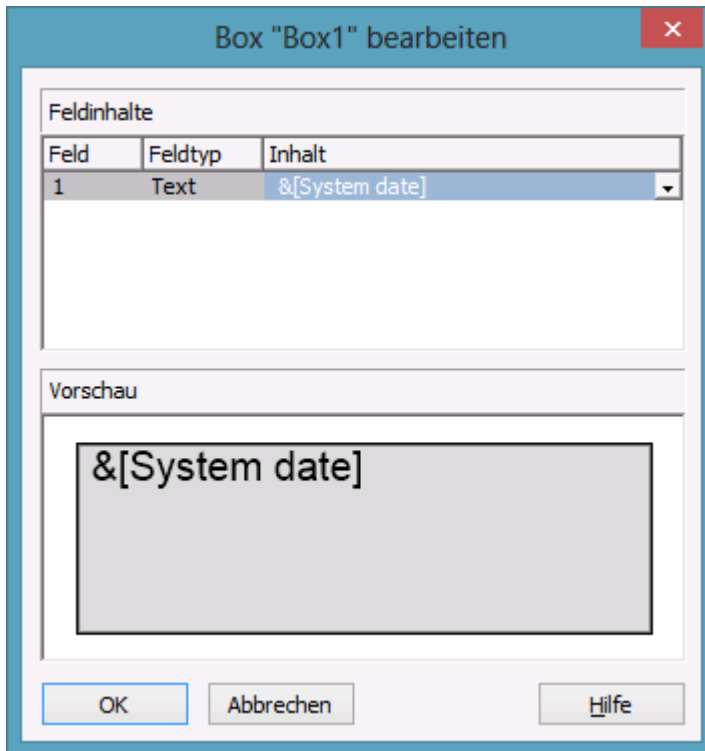
Sie gelangen in das Dialogfeld **Box bearbeiten**.

Box eine Zeile nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie die markierte Box in der Tabelle eine Zeile nach oben bzw. unten schieben.

4.27 Dialogfeld "Box bearbeiten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte** und das Dialogfeld **Boxen verwalten**, indem Sie dort auf die **Box bearbeiten**-Schaltfläche klicken. Dieser Dialog erscheint auch zur Laufzeit, wenn sie auf eine Box doppelklicken.

Feld

In dieser Spalte werden die Nummern aller Felder der Box aufgeführt. (Die Anzahl der Felder hängt vom gewählten Boxformat ab.)

Feldtyp

Hier wird der Feldtyp jedes Feldes angezeigt (Text oder Grafik).

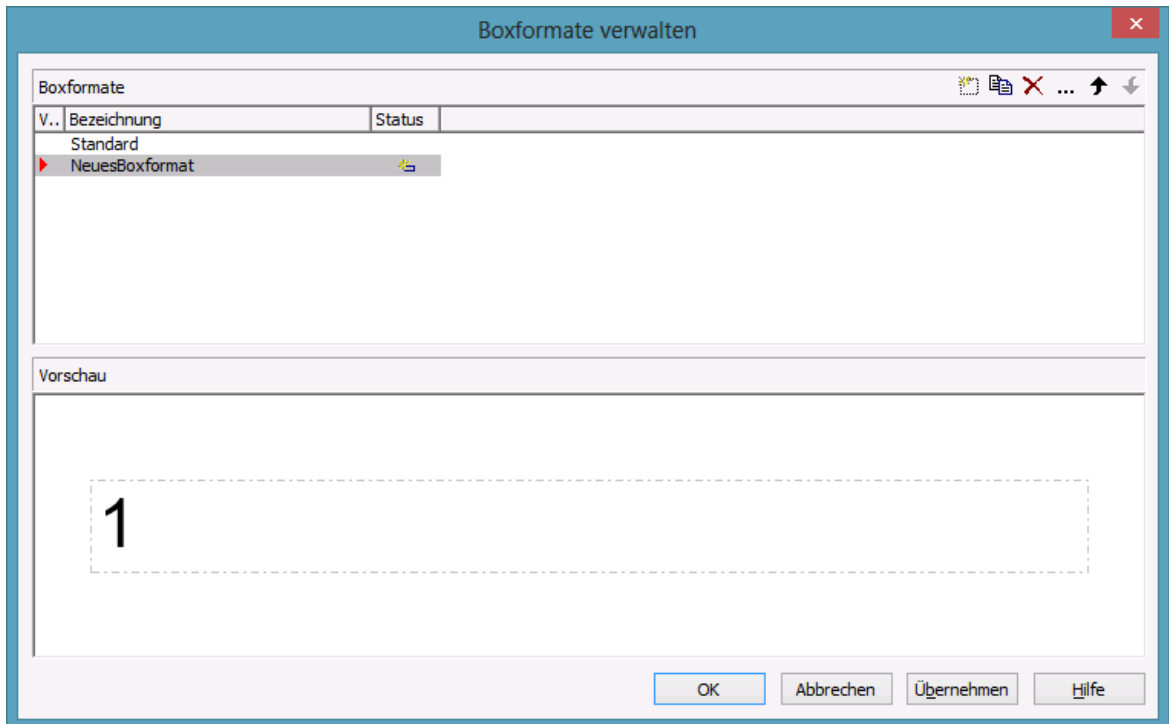
Inhalt

Hier können Sie den Inhalt des Feldes bzw. den Namen einer Grafikdatei eingeben.

Bei mehrzeiligen Textfeldern müssen für einen erzwungenen Umbruch die einzelnen Zeilen des Textfelds mit "\n" im String getrennt sein (Beispiel: "Zeile1\nZeile2"). Ohne erzwungenen Umbruch wird automatisch an Leerzeichen umgebrochen.

Mögliche Grafikformate: WMF, JPG, BMP, GIF, PCX, PNG, TIF.

4.28 Dialogfeld "Boxformate verwalten"



Sie gelangen in dieses Dialogfeld über die Eigenschaftenseite **Objekte**.

Vorschau

Das in der Vorschauspalte markierte Format wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Formate. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jedes Format gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Boxformat hinzufügen

 Ein neues Format mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Boxformat kopieren

 Das markierte Format wird kopiert.

Boxformat löschen

 Das Format, das Sie in der Liste markiert haben, wird gelöscht. Es können nur die Formate gelöscht werden, die zur Zeit nicht benutzt werden.

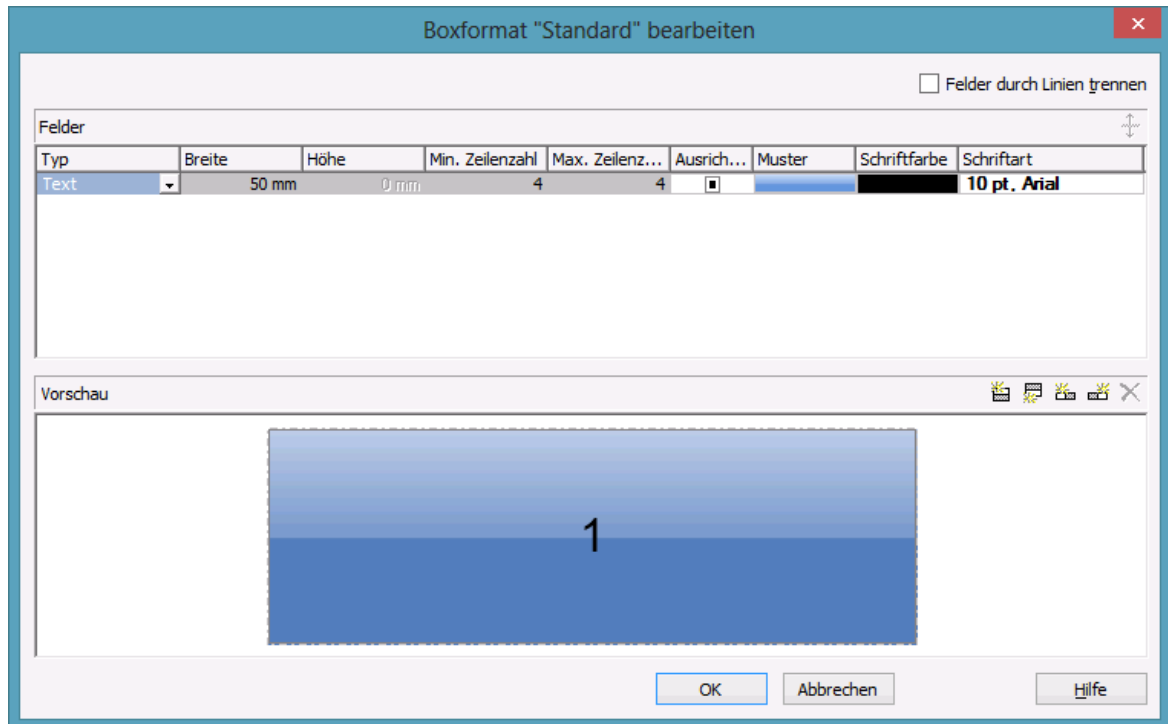
Boxformat bearbeiten

 Sie gelangen in das Dialogfeld **Boxformat bearbeiten**.

Boxformat eine Zeile nach oben / unten

  Mit Hilfe dieser Schaltflächen können Sie das markierte Format in der Tabelle eine Zeile nach oben bzw. unten schieben.

4.29 Dialogfeld "Boxformat bearbeiten"



Sie erreichen dieses Dialogfeld, indem Sie auf der Eigenschaftenseite **Objekte** das Dialogfeld **Boxformate verwalten** aktivieren und dort auf die **Boxformat bearbeiten**-Schaltfläche klicken.

Felder durch Linien trennen

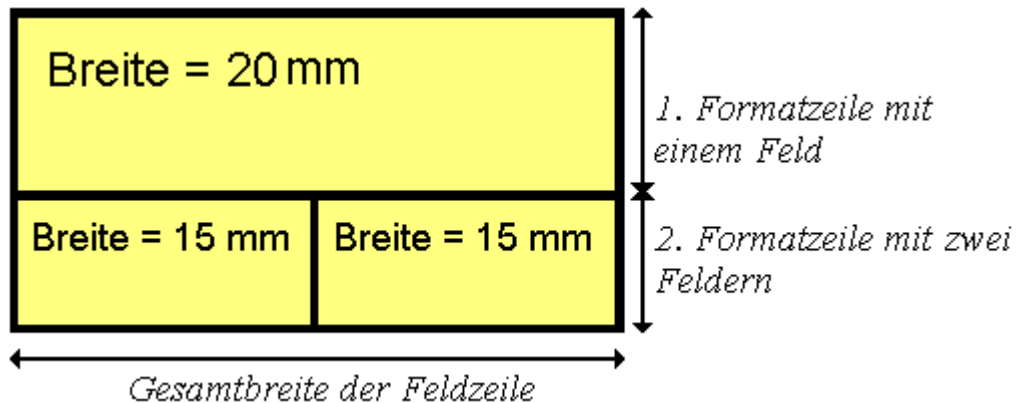
Aktivieren Sie dieses Kontrollkästchen, wenn die Felder der Box durch Linien getrennt werden sollen.

Typ

Wählen Sie hier den Feldtyp (Text oder Grafik).

Breite

Legen Sie hier die Breite in Millimetern für das markierte Feld fest. Die maximale Breite eines Feldes beträgt 200 mm. Ist eine Feldzeile in zwei oder mehr Felder unterteilt und die Gesamtbreite der einzelnen Feldzeilen unterschiedlich groß, so richtet sich die Gesamtbreite nach der insgesamt breitesten Feldzeile.



Höhe

(nur für den Typ Grafik) Legen Sie hier die minimale Höhe in Millimetern für das markierte Feld fest. Die maximale Höhe eines Feldes beträgt 200 mm.

Min./Max. Zeilenzahl

(nur für den Typ Text) Bestimmen Sie die minimale bzw. die maximale Anzahl der Textzeilen des aktuellen Feldes. Die maximale Zeilenzahl eines Feldes beträgt neun.

Ausrichtung

Wählen Sie hier die Ausrichtung des Inhalts in dem markierten Feld (9 Möglichkeiten).

Muster

Legen Sie hier die Hintergrundfarbe und -muster des Feldes fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen können. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.

Schriftfarbe

(nur für den Typ Text) Die Schriftfarbe des Feldes wird hier angezeigt.



Die Farbzuordnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.

Schriftart

(nur für den Typ Text) Die Schriftart des Feldes wird hier angezeigt.

 Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**.

selektierte Eigenschaft in alle Felder übernehmen

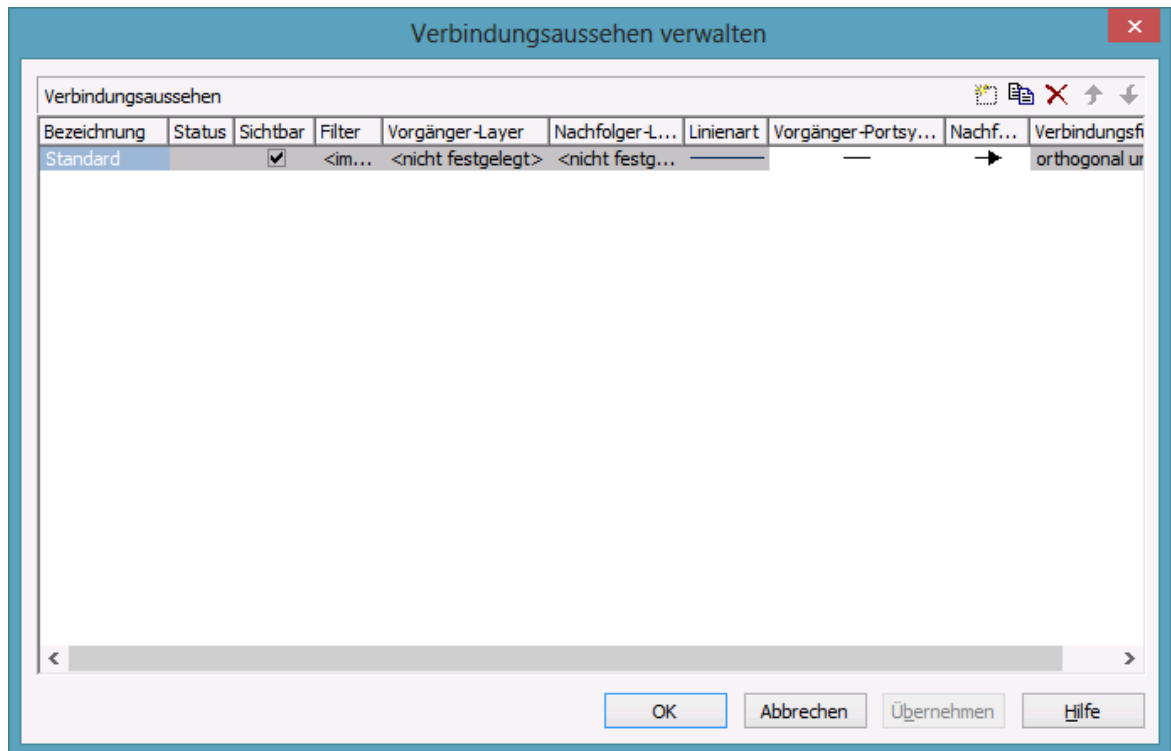
 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

Vorschau

Hier werden die aktuellen Felder des Boxformats dargestellt. Wenn Sie im Vorschaufenster ein Feld anklicken, können Sie dessen Attribute in der **Felder**-Tabelle bearbeiten.

    Mit Hilfe der Schaltflächen oberhalb der Vorschau können Sie neue Felder anlegen oder das markierte Feld löschen. Zum Löschen von Feldern können Sie auch die Entf-Taste benutzen.

4.30 Dialogfeld "Verbindungsaussehen verwalten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**.

Bezeichnung

In dieser Spalte werden die Namen der verfügbaren Verbindungsaussehen angezeigt. Die Namen sind editierbar.

Diese Option kann auch über die Eigenschaft **VcLinkAppearance.Name** festgelegt werden.

Status

In der Spalte **Status** wird jedes Verbindungsaussehen gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Sichtbar

Wenn Sie dieses Kontrollkästchen aktivieren, werden die Verbindungen angezeigt.

Diese Option kann auch über die Eigenschaft **VcLinkAppearance.Visible** festgelegt werden.

Filter

Wählen Sie hier den Filter, der für ein Verbindungsaussehen verwendet werden soll aus.

Diese Option kann auch über die Eigenschaft **VcLinkAppearance.FilterName** festgelegt werden.

Vorgänger-Layer

Wählen Sie hier aus, an welchen Layer des Vorgängerknotens die Verbindung angeknüpft werden soll. Falls der gewählte Layer einem Knoten nicht zugewiesen wurde, wird die Verbindung an den ersten sichtbaren Layer des jeweiligen Knotens geknüpft.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.PredecessorLayerName** festgelegt werden.

Nachfolger-Layer

Wählen Sie hier aus, an welchen Layer des Nachfolgerknotens die Verbindung angeknüpft werden soll. Falls der gewählte Layer einem Knoten nicht zugewiesen wurde, wird die Verbindung an den ersten sichtbaren Layer des jeweiligen Knotens geknüpft.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.SuccessorLayerName** festgelegt werden.

Linienart

Wenn Sie auf den Eintrag dieses Feldes klicken, erscheint die **Bearbeiten**-Schaltfläche . Durch Klick auf diese Schaltfläche gelangen Sie in das Dialogfeld **Linienattribute bearbeiten**, in dem Sie das Aussehen der Verbindungslinien festlegen können.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.LineType** festgelegt werden.

Vorgänger-Portsymbol

Wählen Sie hier für jedes Verbindungsaussehen ein Portsymbol, das die Einmündung zum Vorgängerknoten kennzeichnet.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.PredecessorPortSymbol** festgelegt werden.

Nachfolger-Portsymbol

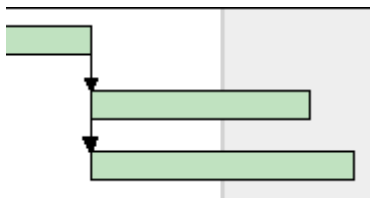
Wählen Sie hier für jedes Verbindungsausssehen ein Portsymbol, das die Einmündung zum Nachfolgerknoten kennzeichnet.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.SuccessorPortSymbol** festgelegt werden.

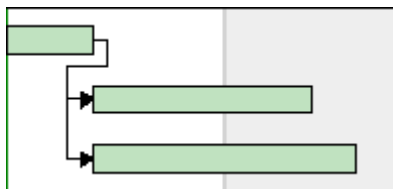
Verbindungsführung

Wählen Sie hier die Art der Verbindungsführung. Der Eintrag <nicht festgelegt> ist erst ab der zweiten Zeile der Tabelle mit den Verbindungsausssehen wählbar, da die erste Zeile immer das Standard-Verbindungsausssehen enthält. Ist <nicht festgelegt> ausgewählt, wird ein in der Liste der LinkAppearance-Objekte weiter vorne stehender Verbindungsführungstyp verwendet.

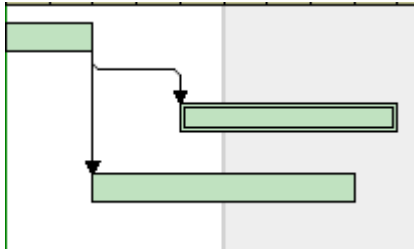
Sie können die Art der Verbindungsführung auch über die **VcLinkAppearance**-Eigenschaft **RoutingType** setzen.



Geradliniger Verbindungstyp



Orthogonaler Verbindungstyp



Orthogonal unterscheidbarer Verbindungstyp

Verbindungsausssehen hinzufügen



Ein neues Verbindungsausssehen mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Verbindungsausssehen kopieren



Das markierte Verbindungsausssehen wird kopiert.

Verbindungsausssehen löschen



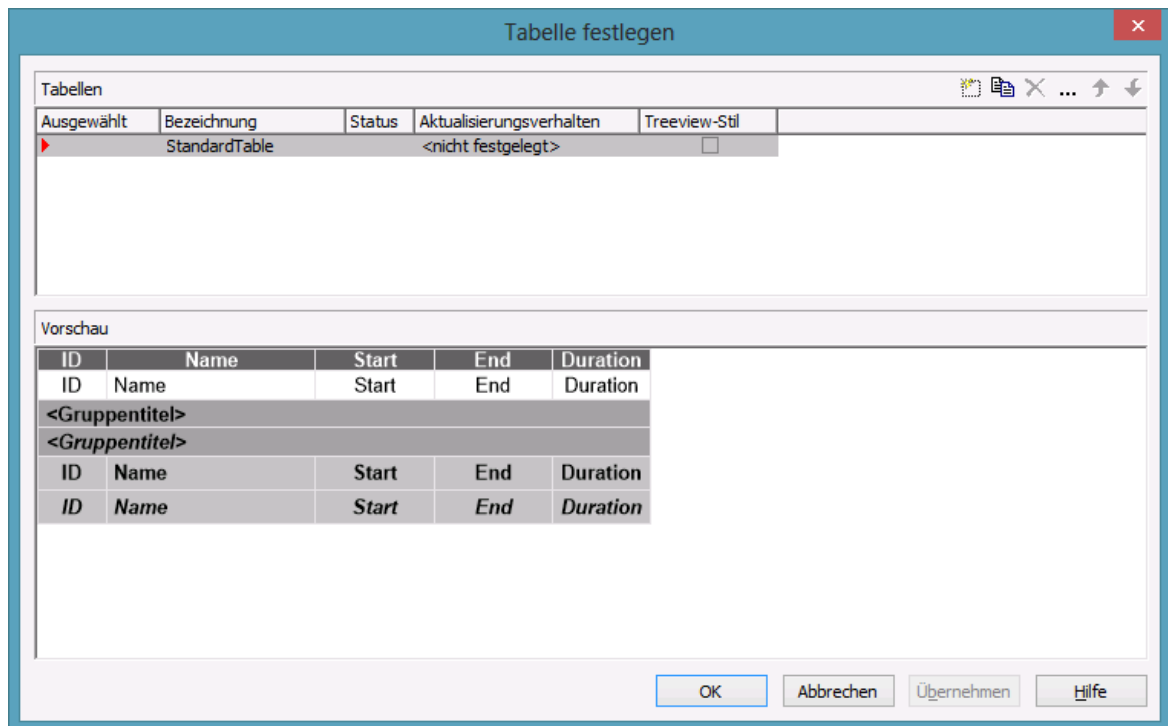
Das Verbindungsausssehen, den Sie in der Liste markiert haben, wird gelöscht. Es können nur Verbindungsausssehen gelöscht werden, die zur Zeit nicht benutzt werden.

Verbindungsausssehen eine Zeile nach oben/unten



Mit Hilfe dieser Schaltflächen können Sie das markierte Verbindungsausssehen eine Zeile nach oben/unten schieben.

4.31 Dialogfeld "Tabelle festlegen"



In diesem Dialogfeld können Sie die Tabellen festlegen und verwalten.

Vorschau

Die in dieser Spalte mit einem roten Pfeil markierte Tabelle wird im Vorschauenfenster in der unteren Hälfte des oben dargestellten Dialogs angezeigt. Sie ist gleichzeitig die Tabelle, die aktuell bearbeitet wird.

Bezeichnung

In dieser Spalte werden die Namen aller verfügbaren Tabellen aufgeführt. Die Namen sind editierbar.

Status

In dieser Spalte wird jede Tabelle gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (📁) und/oder geändert (🔴) worden ist.

Aktualisierungsverhalten

Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog **Aktualisierungsverhalten bearbeiten** für Tabellen festgelegt wurde.

Treeview-Stil

Wenn dieses Kontrollkästchen aktiviert ist, werden Knoten im TreeView-Stil angeordnet, bei dem die logische Baumstruktur durch Linen gezeichnet wird. Ebenen werden in beiden Darstellungen mit Plus- oder Minus-Zeichen versehen.

ID	Name
[-]	
1	Snyder
2	Smith

ID	Name
[-]	
1	Snyder
2	Smith

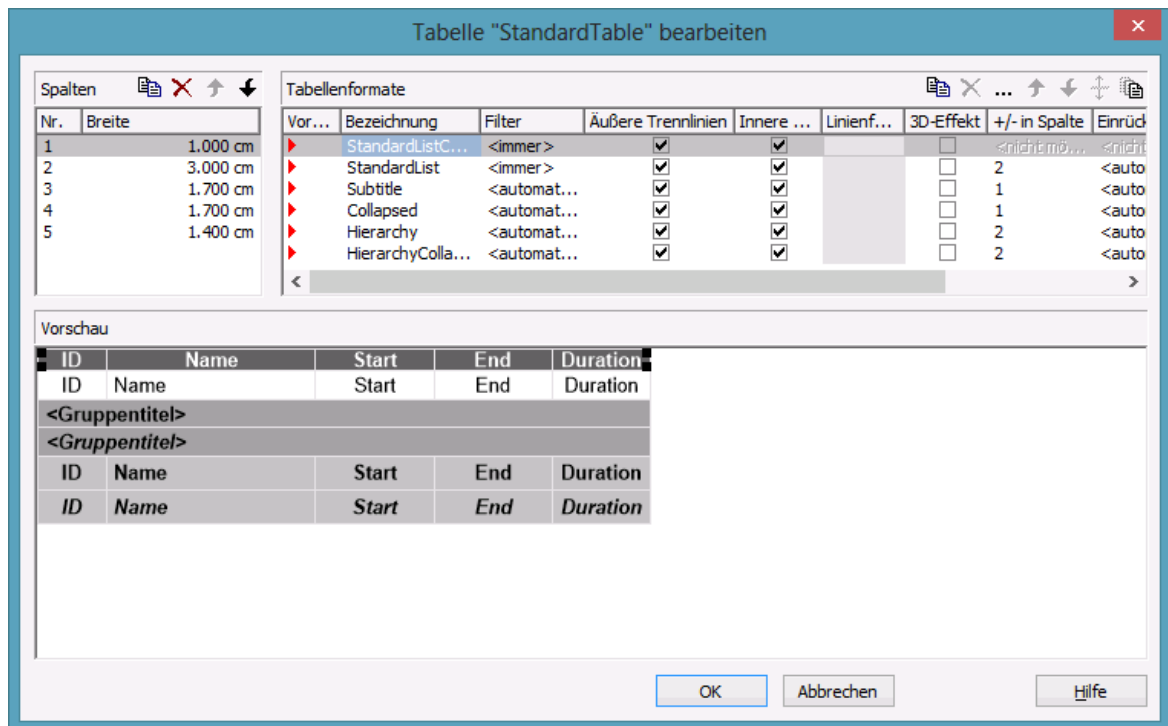
Bilder oben: Gruppierung mit und ohne Treeview-Stil

Tabelle hinzufügen / kopieren / löschen / eine Zeile nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie die markierte Tabelle hinzufügen, kopieren, löschen oder in der Tabelle nach oben bzw. unten verschieben. Letzteres kann z.B. der Sortierung und damit einer besseren Übersicht dienen, hat aber keinerlei Funktion hinsichtlich der Priorität.

4.32 Dialogfeld "Tabelle bearbeiten"



In diesem Dialogfeld können Sie eine Tabelle bearbeiten.

Spalten

In der **Spalten**-Liste werden die **Nr.** und die **Breite** jeder Tabellenspalte aufgelistet. Die Breite kann im Bereich von 0 bis 10 cm in Schritten von 1 mm variiert werden.

Maximal 100 Spalten sind möglich. Die Reihenfolge der Tabellenspalten in der **Spalten**-Liste entspricht der Reihenfolge der Spalten in der Grafik.



Mit Hilfe der Schaltflächen oberhalb der **Spalten**-Liste können Sie Tabellenspalten kopieren, löschen oder ihre Position verändern.

Tabellenformate

In der **Tabellenformate**-Liste können Sie verschiedene Zeilentypen festlegen:

- **Vorschau:** Jedes Tabellenformat, das in dieser Liste mit einem roten Dreieck markiert ist, wird in der Vorschau dargestellt.
- **Bezeichnung:** Jedes Tabellenformat hat standardmäßig eine Bezeichnung. **StandardListCaption** ist das Tabellenformat für die Tabellenüberschrift. Die Bezeichnung der Tabellenformate ist editierbar

bei den Tabellenformaten **ListFormat2** und **ListFormat3** sowie bei allen selbst definierten Tabellenformaten.

- **Filter:** Mit jedem Tabellenformat ist ein Filter verbunden, der die Vorgänge auswählt, für die das entsprechende Tabellenformat verwendet werden soll. Wenn mehrere Filter dieser Liste auf einen Vorgang zutreffen, wird das Tabellenformat mit der jeweils höchsten Priorität verwendet. Die Reihenfolge der Filter in der Auflistung des Fensters von unten nach oben bestimmt ihre Priorität. Der obere Filter besitzt somit die niedrigste Priorität. Es gibt folgende vordefinierte Filter: <Anschlussknoten>: das Format wird nur auf Knoten angewendet, die an die selektierten Knoten angeschlossen sind. <Nie>: der Filter kommt nie zur Anwendung und kann praktischerweise als Kopiervorlage für neue, selbstdefinierte Filter dienen. <Automatisch>: das Format wird auf Knoten angewendet, die einer zu spezifizierenden Gruppierungsebene angehören. <Immer>: das Format ist das Default-Format und wird auf alle Knoten angewendet, die von keinem anderen Filter abgefangen werden. Der Filter <immer> ist daher nicht löschtbar. Es ist sinnvoll, diesen Filter ganz nach oben zu setzen.
- **Äußere/Innere Trennlinien:** Legen Sie für jedes Tabellenformat fest, ob die Felder darin durch äußere/innere Trennlinien voneinander getrennt werden sollen.
- **Linienfarbe:** Hier können Sie die Farbe der Trennlinien für jedes Format individuell setzen.
- **3D-Effekt:** Legen Sie hier fest, ob die einzelnen Tabellenfelder durch einen 3D-Effekt hervorgehoben werden sollen.
- **+/- in Spalte:** Legen Sie hier fest, ob in einer Spalte +/- zum Ein- bzw. Ausblenden untergeordneter Zeilen angezeigt werden soll. Wählen Sie aus der Liste die gewünschte Spaltennummer.
- **Einrückungsspalte:** Legen Sie hier fest, welche Spalte einen Einzug bekommen soll. Dies wirkt sich nur aus, wenn der Zeile bzw. dem Knoten andere Zeilen/Knoten untergeordnet sind. Dann wird die erste untergeordnete Zeile eingerückt. Bei Auswahl von **automatisch** wird die Spalte eingerückt, in der +/- angezeigt wird.
- **Einrückungsbreite:** Geben Sie hier das Maß des Einzugs in mm an.

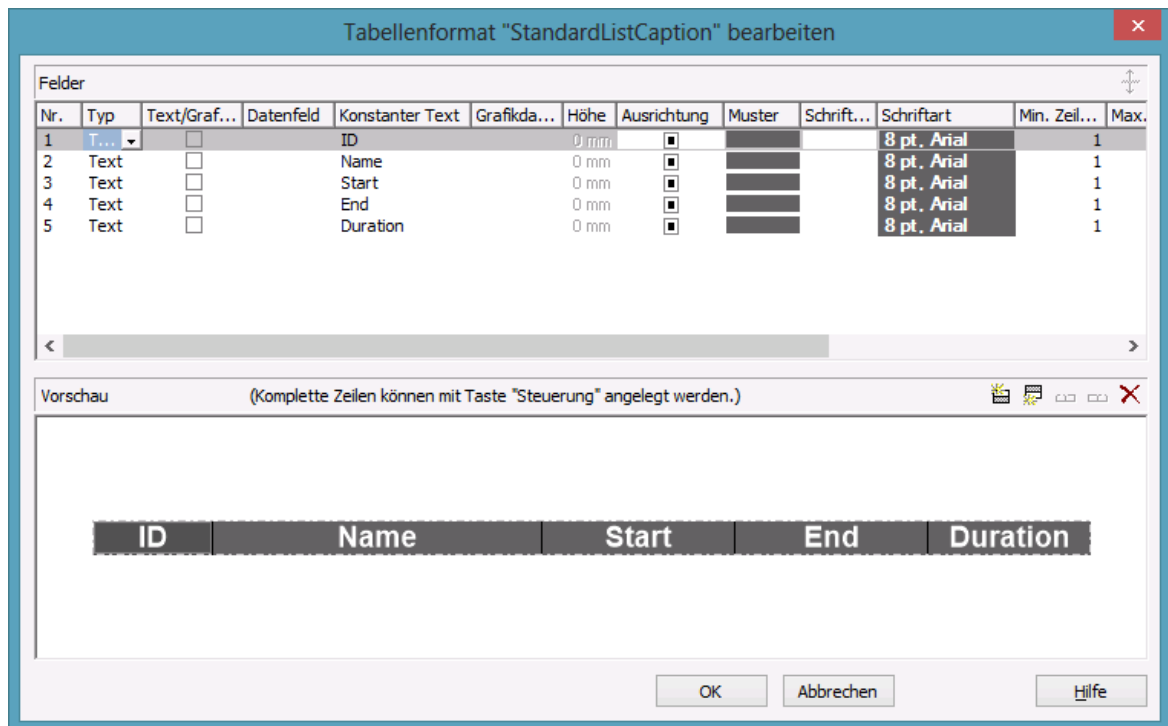
 Mit Hilfe dieser Schaltflächen oberhalb der **Tabellenformate**-Liste können Sie das markierte Tabellenformat kopieren oder löschen oder dafür den Dialog **Tabellenformat bearbeiten** aufrufen.

Hinweis: Beim Tabellenformat **StandardListCaption** (Tabellenüberschrift) sind die Attribute der Tabellenformate nicht über Zuordnungstabellen einstellbar.

 Mit Hilfe der Pfeil-Schaltflächen können Sie alle Tabellenformate mit Ausnahme der beiden ersten in der Liste positionieren.

 Wenn Sie die Eigenschaft **Äußere Trennlinien** oder **Innere Trennlinien** für ein Tabellenformat geändert haben und dann auf diese Schaltfläche klicken, wird diese Änderung auf alle anderen Tabellenformate übertragen.

4.33 Dialogfeld "Tabellenformat bearbeiten"



In diesem Dialogfeld können Sie ein Tabellenformat (einen Zeilentyp) bearbeiten.

Nr.

Nummer des Tabellenformatfeldes: Diese Nummer ist nicht editierbar. Sie dient als Index, über den Sie über die API auf dieses Tabellenformatfeld zugreifen können.

Wenn Sie im Vorschauenfenster ein neues Tabellenformatfeld anlegen, erhält dieses in der Liste zunächst statt einer Nummer ein Fragezeichen. Erst nachdem Sie den Dialog mit **OK** verlassen und wieder geöffnet haben, wird das Fragezeichen durch eine neue Nummer ersetzt.

Typ

Wählen Sie hier den Feldtyp: **Text**, **Grafik** oder **MultiState**. MultiState-Felder werden verwendet, um z.B. beim Anklicken eines Tabellenfeldes eine zyklische Abfolge von Zuständen wie auch der zugehörigen Datenfelder auszulösen.

Kombi-Feld

Wenn dieses Kontrollkästchen aktiviert ist, können in dem Tabellenformatfeld ein Text und eine Grafik kombiniert werden, und zwar folgendermaßen:

- **Typ:** Text, **Kombi-Feld:** nein: Nur Text wird ausgegeben (wie unter **Datenfeld** oder unter **Konstanter Text** angegeben).
- **Typ:** Grafik, **Kombi-Feld:** nein: Nur eine Grafik wird ausgegeben (wie unter **Grafikdateiname** angegeben).
- **Typ:** Text, **Kombi-Feld:** ja: Text (wie unter **Datenfeld** oder unter **Konstanter Text** angegeben) und Grafik (wie unter **Grafikdateiname** angegeben) werden ausgegeben.
- **Typ:** Grafik, **Kombi-Feld:** ja: Nur eine Grafik wird ausgegeben (wie unter **Grafikdateiname** angegeben). Text (unter **Datenfeld** angegeben) ist nur im Tooltip sichtbar; er kann aber, falls geeignet, als Hyperlink genutzt werden.)

Datenfeld

Wählen Sie hier das Datenfeld, dessen Inhalt in dem aktuellen Feld ausgegeben werden soll. Außer den Datenfeldern, die Sie in der Datendefinition vereinbart haben, gibt es folgende Optionen:

- <Gruppentitel>: der bei der jeweiligen Gruppierungsebene eingestellte Code
- <Zeilennummer>: fortlaufende Zeilennummer

Passt der Inhalt des Datenfeldes nicht in das Feld, wird der Überhang bei der Ausgabe abgeschnitten.

Konstanter Text

(nur wenn kein Datenfeld gewählt wurde) Sie können hier einen konstanten Text eingeben, der in dem Feld ausgegeben werden soll.

Grafikdateiname

Hier werden Name und Pfad der Grafikdatei angezeigt, die in dem gewählten Tabellenfeld dargestellt werden soll.

Wenn Sie auf ein **Grafikdateiname**-Feld klicken, erscheinen zwei Schaltflächen:

 Der Windows-Dialog **Grafikdatei auswählen** erscheint. Hier können Sie eine Grafikdatei auswählen, die in dem aktuellen Formatfeld erscheinen soll.

Wird ein relativer Dateiname angegeben, so wird die Datei zur Laufzeit zuerst in dem Verzeichnis gesucht, das in der VARCHART-ActiveX-Eigenschaft **FilePath** gesetzt ist. Wird sie dort nicht gefunden, wird sie zuerst im gerade aktiven Arbeitsverzeichnis der Applikation und dann im Installationsverzeichnis des VARCHART ActiveX gesucht.

 Klicken Sie auf diese Schaltfläche (**Zuordnungen einstellen**), um eine Zuordnungstabelle zu verwenden, um Grafiken abhängig vom Inhalt eines Datenfelds in einem Tabellenfeld erscheinen zu lassen.

Wird im Dialog **Zuordnung einstellen** nur ein Datenfeld, aber keine Zuordnungstabelle ausgewählt, so wird der Inhalt des eingestellten Datenfelds direkt als Name einer Grafikdatei interpretiert. Findet sich in dem Datenfeld bzw. in der Zuordnungstabelle kein gültiger Name einer Grafikdatei, dann wird der direkt in Spalte angegebene Dateiname verwendet.

Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der 2. Schaltfläche fett dargestellt ().

 Sobald Sie die entsprechende Zeile verlassen, zeigt ein Symbol im Feld **Grafikdateiname** an, dass eine Zuordnung vorgenommen worden ist.

Bei der Darstellung der Grafik wird die Farbe des Pixels von ihrer Ecke links oben durch die Diagramm-Hintergrundfarbe ersetzt, d. h. alle Pixel der Grafik in dieser Farbe werden transparent dargestellt.

Höhe

(nur für den Typ Grafik) Legen Sie hier die minimale Höhe in Millimetern für das markierte Feld fest. Die maximale Höhe eines Feldes beträgt 99 mm.

Ausrichtung

Wählen Sie hier die Ausrichtung des Inhalts in dem markierten Feld (9 Möglichkeiten).

Muster

Legen Sie hier die standardmäßige Hintergrundfarbe und -muster des Tabellenformates fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**. Hier können Sie durch Klick auf  neben dem jeweiligen Attribut ein Muster, eine Hintergrundfarbe sowie eine 2.

Musterfarbe auswählen. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.

Durch Klick auf die Schaltfläche  gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung des Musters und der Farben vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

Schriftfarbe

Die Schriftfarbe des Feldes wird hier angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Die Farbzusordnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Farbe vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

Schriftart

Die Schriftart des Feldes wird hier angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Schriftart vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

Min./Max. Zeilenzahl

(nur für den Typ Text) Bestimmen Sie die minimale bzw. die maximale Anzahl der Textzeilen des aktuellen Feldes. Die maximale Zeilenzahl eines Feldes beträgt neun.

Zeilenabstand

Wählen Sie hier den prozentualen Zeilenabstand.

Umbruch

Legen Sie fest, ob ggf. die Zeilen umgebrochen werden sollen.

Hor. Ränder (links/ rechts)/ Ver. Ränder (oben/ unten)

Legen Sie die Breite der Ränder der Tabellenformatfelder fest.

+/- in Spalte

Wählen Sie aus, ob in der Spalte +/- zum Ein- bzw. Ausblenden der weiteren Zeilen angezeigt werden soll

Einrückungsspalte

Legen Sie hier fest, ob die Spalte eingerückt werden soll.

Vorschau

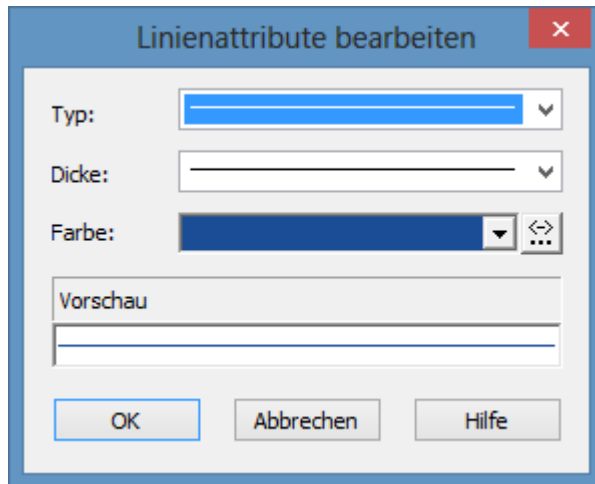
Hier werden die aktuellen Felder des Tabellenformats dargestellt. Wenn Sie im Vorschaufenster ein Feld anklicken, können Sie dessen Attribute in der **Felder**-Tabelle bearbeiten.



Mit Hilfe der Schaltflächen oberhalb des Vorschaufensters können Sie neue Felder anlegen oder das markierte Feld löschen. Zum Löschen von Feldern können Sie auch die Entf-Taste benutzen.

Die vier ersten Schaltflächen, die zum Hinzufügen neuer Felder dienen, sind immer nur dann aktiviert, wenn an der markierten Stelle tatsächlich Felder hinzugefügt werden können. Dies hängt davon ab, wie viele Spalten für das betreffende Tabellenformat im übergeordneten Dialog **Tabelle bearbeiten** festgelegt worden sind, wie viele Felder also maximal nebeneinander in diesem Tabellenformat möglich sind.

4.34 Dialogfeld "Linienattribute bearbeiten"



Der Dialog zum Bearbeiten der Linienattribute, der jeweils über die Schaltfläche aufgerufen werden kann, steht zur Verfügung für Hierarchie und Gruppierung, für Kalendergitter, für das Balkenaussehen, für Kurvenfüllung und die numerischen Skalen beim Histogramm, für das Verbindungsaussehen, für Intervalle sowie für den Rahmen von Boxen.

Typ

Wählen Sie hier den Typ der Linie aus (durchgezogen, gestrichelt, etc.).

Dicke

Wählen Sie hier die Linienstärke aus.

Farbe

Wählen Sie hier die Farbe der Linien aus.



Über diese Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**, in dem Sie eine datenabhängige Zuordnung der Linienfarbe vereinbaren können.



Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt.

Vorschau

Hier wird das Aussehen der Linie mit den gewählten Einstellungen angezeigt.

4.35 Dialogfeld "Musterattribute bearbeiten"



Der Muster-Dialog, der über die Schaltfläche  aufgerufen werden kann, steht zur Verfügung für die Kurvenfüllung beim Histogramm, für Kalendergitter, die Titelzeile einer Gruppe, Intervalle, Zeitskalenabschnitte, Box-, Linien- und Tabellenformate, Layer sowie für Knotenzeilen.

 Über diese Schaltfläche gelangen Sie jeweils in den Dialog **Zuordnung einstellen**, in dem Sie eine datenabhängige Zuordnung für Muster, Musterfarbe oder Hintergrundfarbe oder Musterfarbe2 vereinbaren können.

 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt.

Muster

Hier können Sie ein Hintergrundmuster auswählen.

Musterfarbe

Wählen Sie hier die Vordergrund-Farbe des Musters aus.

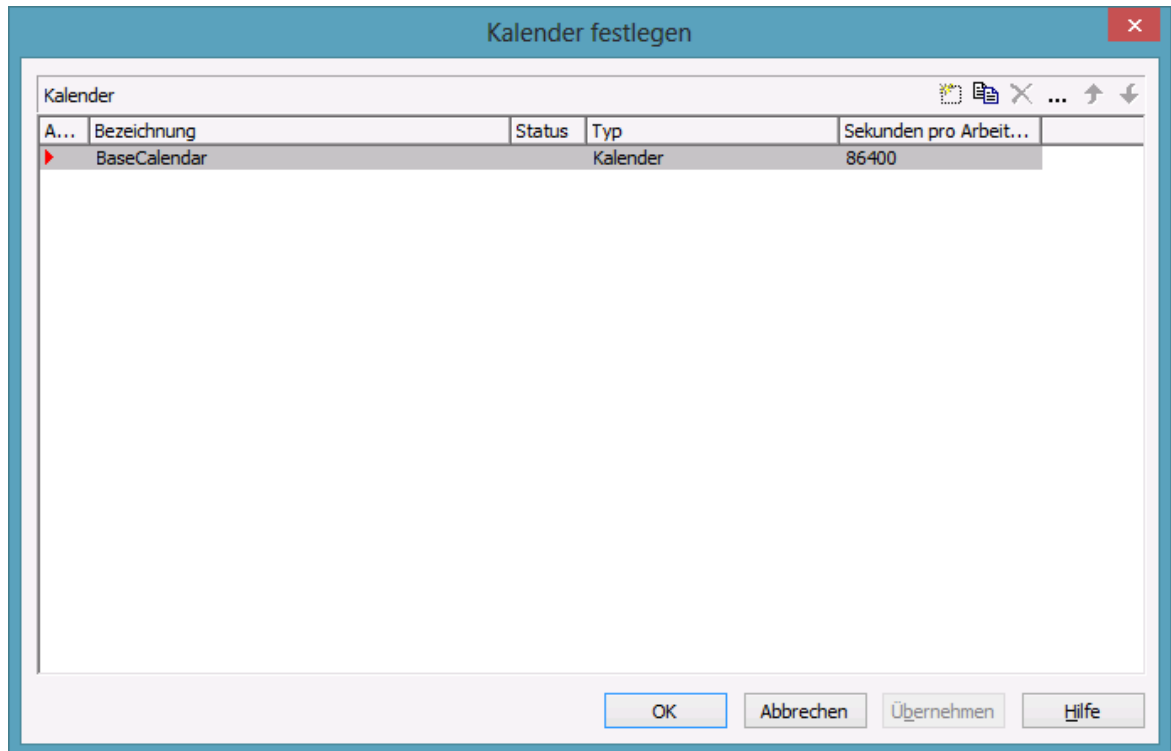
Hintergrundfarbe oder Musterfarbe 2

Wählen Sie hier die Hintergrund-Farbe oder eine zweite Musterfarbe aus.

Vorschau

Hier wird das Aussehen des Musters mit den gewählten Einstellungen angezeigt.

4.36 Dialogfeld "Kalender festlegen"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**. In jeder Zeile der Tabelle können Sie einen Kalender definieren.

Ausgewählt

Der Kalender, den Sie in dieser Spalte durch eine kleine Pfeilspitze markieren, wird für das Kalendergitter verwendet.

Bezeichnung

In dieser Spalte werden die Namen aller definierten Kalender aufgeführt.

Status

In der Spalte **Status** wird jeder Kalender gekennzeichnet, der seit dem Aufruf des Dialogs hinzugefügt (🌟) und/oder geändert (⚠️) worden ist.

Typ

Legen Sie hier den Kalendertyp fest. Außer normalen Kalendern sind auch Schichtkalender möglich.

Sekunden pro Arbeitstag

Legen Sie hier fest, wie viele Sekunden der Arbeitstag hat.

Kalender hinzufügen



Um einen neuen Kalender zu definieren, klicken Sie auf diese Schaltfläche.

Kalender kopieren



Der markierte Kalender wird kopiert.

Kalender löschen



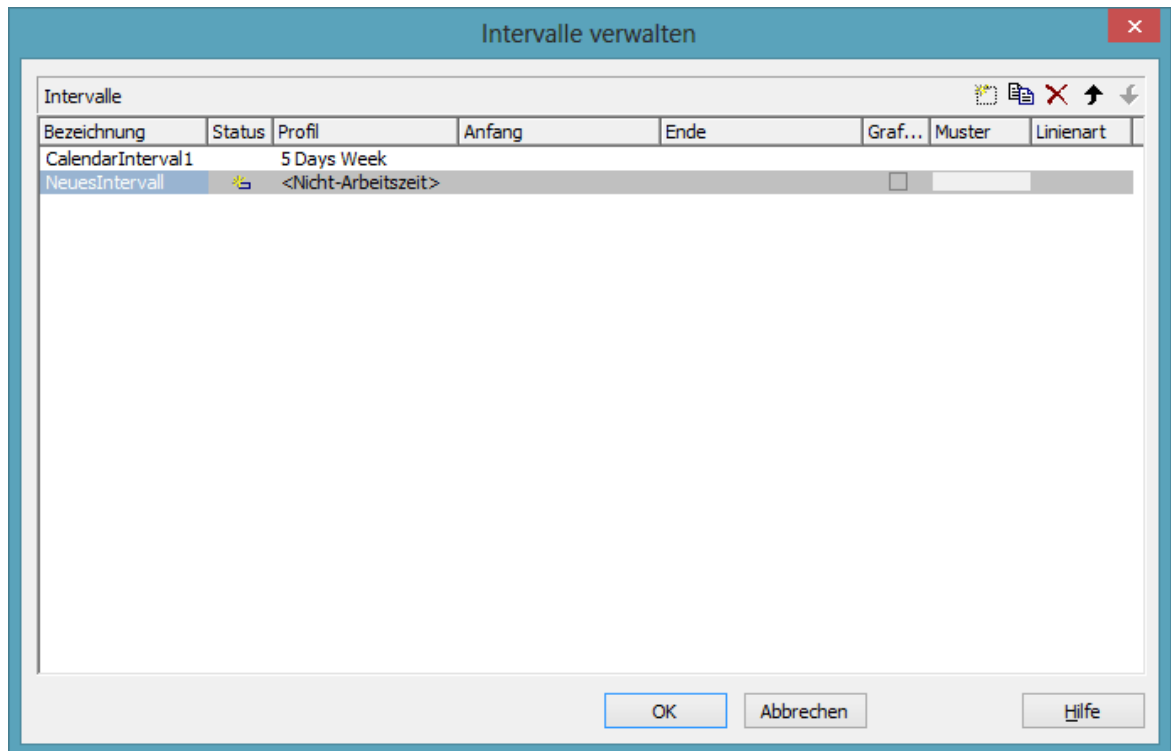
Der markierte Kalender wird gelöscht.

Kalender bearbeiten



Wenn Sie diese Schaltfläche anklicken, öffnet sich der Dialog **Kalender bearbeiten**.

4.37 Dialogfeld "Intervalle verwalten" (Kalender)



In diesem Dialogfeld können Sie Intervalle einrichten und verändern.

Bezeichnung

In dieser Spalte werden die Bezeichnungen aller Intervalle aufgeführt. Die Bezeichnungen sind editierbar.

Status

In dieser Spalte wird jedes Intervall gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (🌟) und/oder geändert (🔴) worden ist.

Profil

Weisen Sie hier dem Intervall ein Kalenderprofil zu. Über  öffnen Sie eine Liste mit Profilen, aus der Sie das gewünschte Profil auswählen können. Zum Bearbeiten des Profils klicken Sie neben dem Profilnamen auf  um den Dialog **Kalenderprofile verwalten** zu öffnen.

Anfang/Ende

Legen Sie hier für das jeweilige Intervall den Anfang bzw. das Ende fest. Das Datum lässt sich komfortabel mithilfe eines Spincontrols eingeben bzw. ändern.

Grafische Attribute verwenden

Wenn Sie diese Option wählen, können für das Intervall Muster und Linientyp eingestellt und angezeigt werden. Die Option ist nur aktiv für die Profiltypen <Arbeitszeit> und <Nicht-Arbeitszeit>.

Muster

Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**.

Linienart

Durch Klick auf  gelangen Sie in den Dialog **Linienattribute bearbeiten**.

Intervall hinzufügen

 Ein neues Intervall mit einem Standardnamen wird angelegt. Doppelklicken Sie auf den markierten Namen um ihn zu verändern.

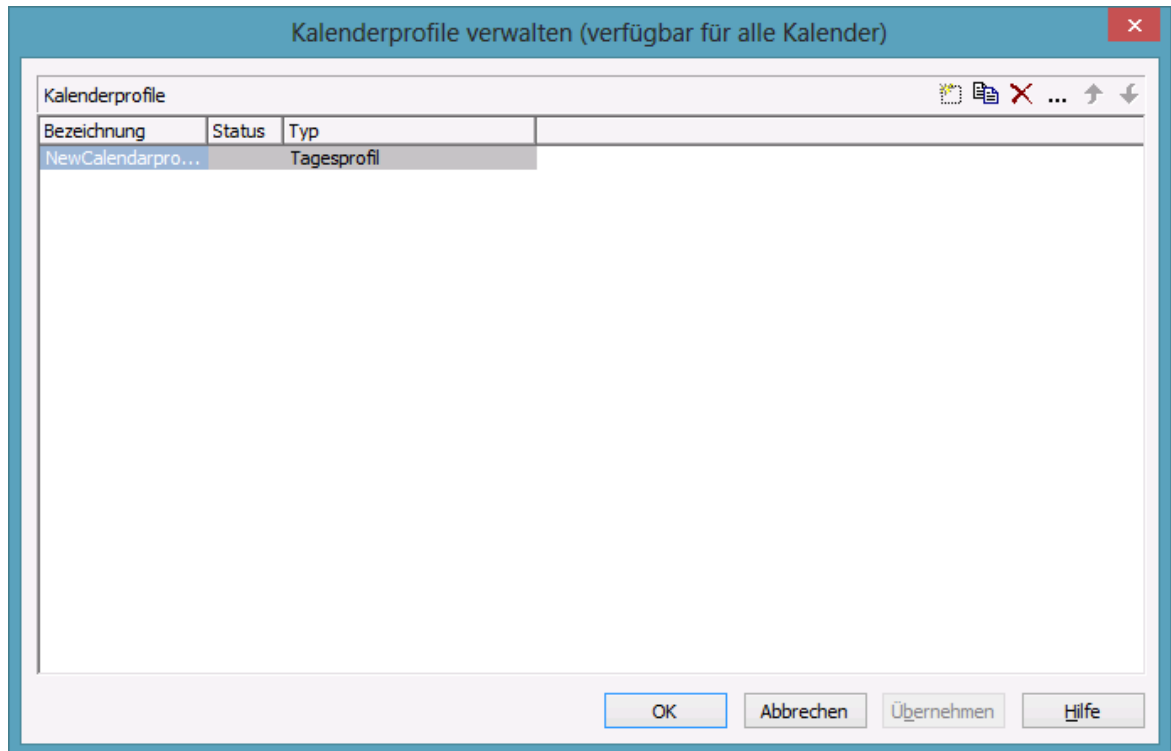
Intervall kopieren

 Es wird eine Kopie des markierten Intervalls angelegt.

Intervall löschen

 Das markierte Intervall wird gelöscht.

4.38 Dialogfeld "Kalenderprofile verwalten"



In diesem Dialog können Sie Kalenderprofile einrichten und verändern.

Bezeichnung

In dieser Spalte werden die Bezeichnungen aller Kalenderprofile aufgeführt. Die Bezeichnungen sind editierbar.

Status

In dieser Spalte wird jedes Kalenderprofil gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt (🌟) und/oder geändert (🔴) worden ist.

Typ

Wählen Sie hier durch Klick auf  den Typ des Kalenderprofils aus. Zur Verfügung stehen die Profiltypen <Tagesprofil>, <Wochenprofil>, <Jahresprofil> und <Variables Profil>.

Kalenderprofil hinzufügen

 Ein neues Kalenderprofil mit einem Standardnamen wird angelegt. Doppelklicken Sie auf den markierten Namen um ihn zu verändern.

Kalenderprofil kopieren



Es wird eine Kopie des markierten Kalenderprofils angelegt.

Kalenderprofil löschen



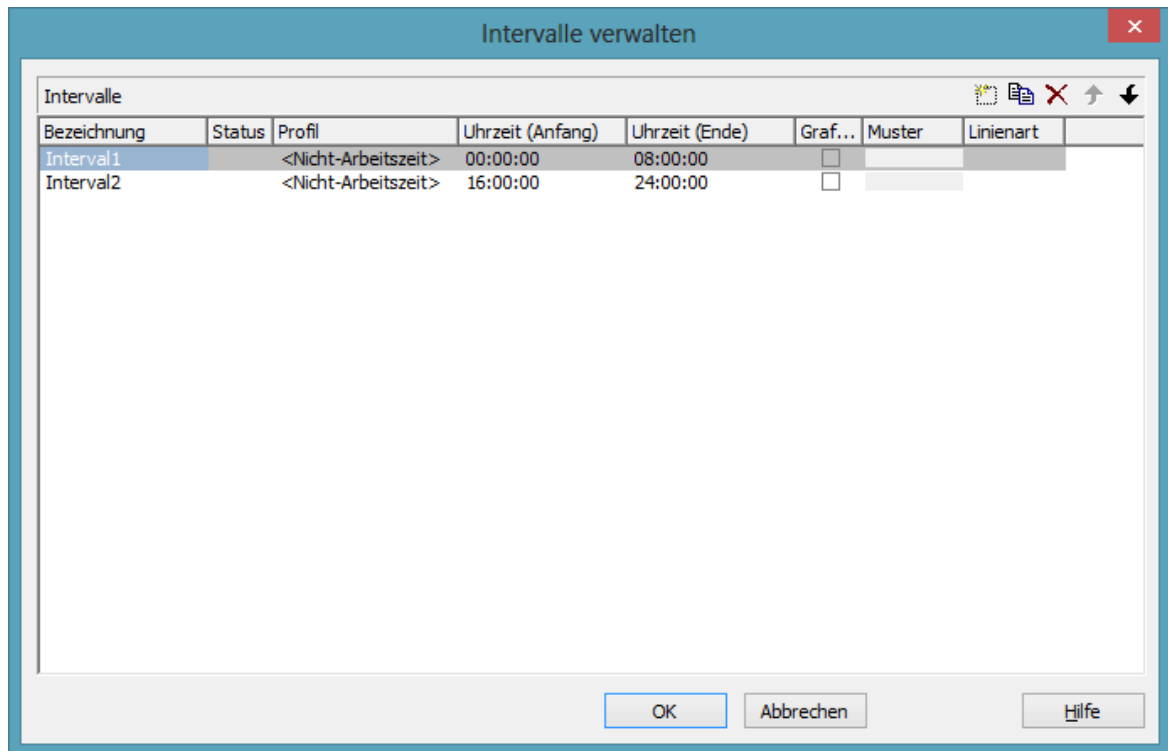
Das markierte Kalenderprofil wird gelöscht.

Kalenderprofil bearbeiten



Wenn Sie diese Schaltfläche anklicken, öffnet sich der Dialog **Intervalle verwalten** (Kalenderprofile).

4.39 Dialogfeld "Intervalle verwalten" für Tagesprofil



Zu diesem Dialogfeld gelangen Sie, wenn Sie auf der Eigenschaftenseite "Objekte" das Dialogfeld "Kalenderprofile verwalten" aktivieren und dann die "Bearbeiten"-Schaltfläche eines Kalenderprofils drücken. Die unterschiedlichen Profiltypen bieten entsprechende Optionen; dieses Dialogfeld dient dazu, für ein Tagesprofil Intervalle einzurichten und zu verändern.

Bezeichnung

In dieser Spalte werden die Bezeichnungen aller Intervalle aufgeführt. Die Bezeichnungen sind editierbar.

Status

In dieser Spalte wird jedes Intervall gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Profil

Weisen Sie hier dem Intervall ein Kalenderprofil zu. Über  öffnen Sie eine Liste mit Profilen, aus der Sie das gewünschte Profil auswählen können.

Uhrzeit Anfang/Uhrzeit Ende

Legen Sie hier für das jeweilige Intervall mithilfe der Pfeiltasten die Start- bzw. Endezeit fest.

Grafische Attribute verwenden

Wenn Sie diese Option wählen, können für das Intervall Muster und Linientyp eingestellt und angezeigt werden. Die Option ist nur aktiv für die Profiltypen <Arbeitszeit> und <Nicht-Arbeitszeit>.

Muster

Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**.

Linienart

Durch Klick auf  gelangen Sie in den Dialog **Linienattribute bearbeiten**.

Intervall hinzufügen

 Ein neues Intervall mit einem Standardnamen wird angelegt. Doppelklicken Sie auf den markierten Namen um ihn zu verändern.

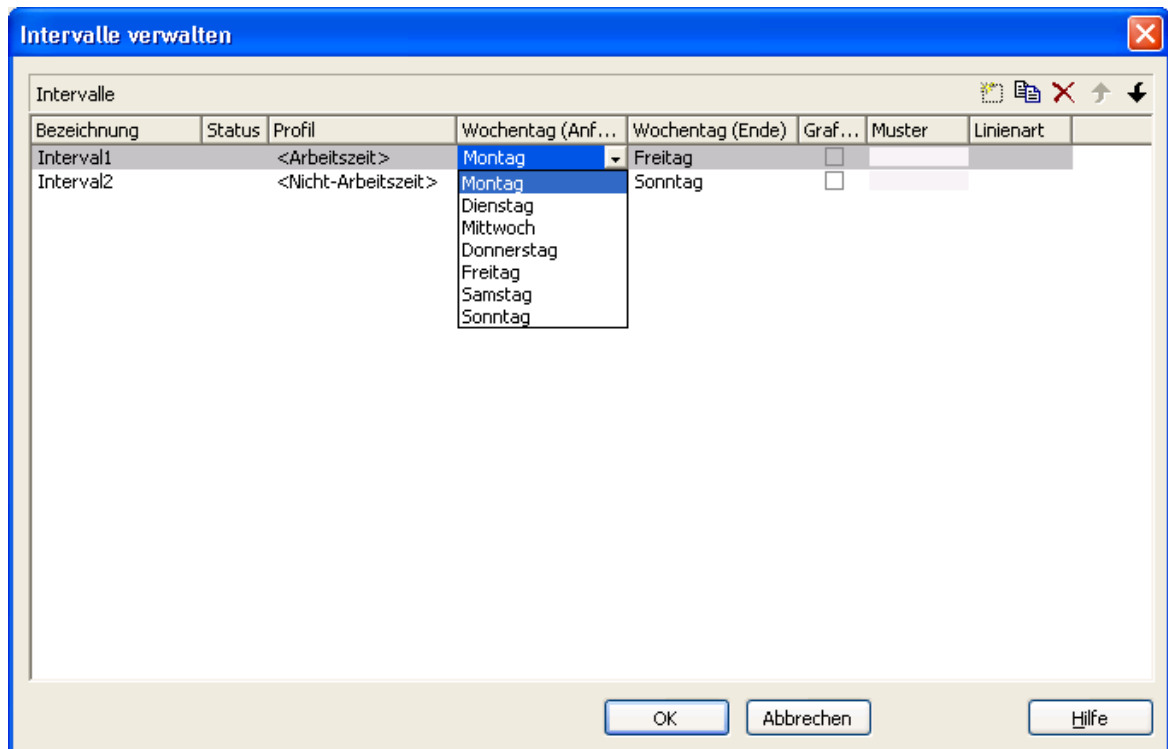
Intervall kopieren

 Es wird eine Kopie des markierten Intervalls angelegt.

Intervall löschen

 Das markierte Intervall wird gelöscht.

4.40 Dialogfeld "Intervalle verwalten" für Wochenprofil

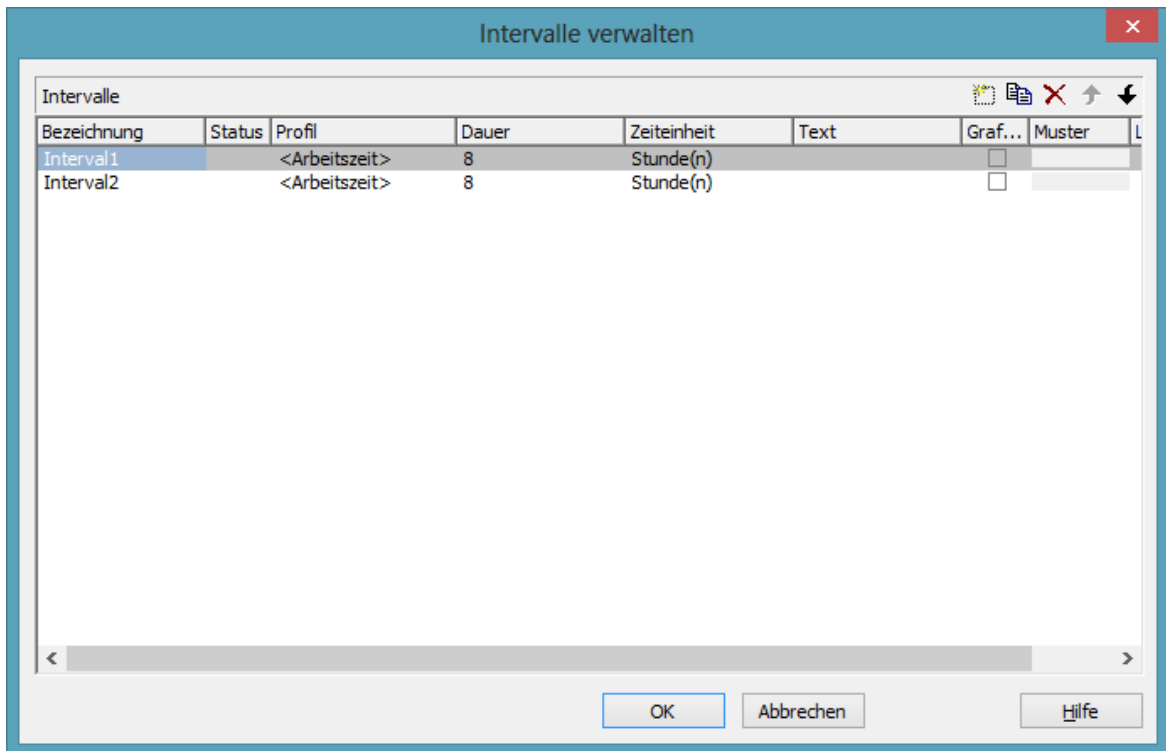


Zu diesem Dialogfeld gelangen Sie, wenn Sie auf der Eigenschaftenseite "Objekte" das Dialogfeld "Kalenderprofile verwalten" aktivieren und dann die "Bearbeiten"-Schaltfläche eines Kalenderprofils drücken. Die unterschiedlichen Profiltypen bieten entsprechende Optionen; dieses Dialogfeld dient dazu, für ein Wochenprofil Intervalle einzurichten und zu verändern.

Wochentag Anfang/Wochentag Ende

Durch Klick auf  können Sie den ersten/letzten Wochentag des Intervalls festlegen.

4.41 Dialogfeld "Intervalle verwalten" für Variables Profil



Zu diesem Dialogfeld gelangen Sie, wenn Sie auf der Eigenschaftenseite "Objekte" das Dialogfeld "Kalenderprofile verwalten" aktivieren und dann die "Bearbeiten"-Schaltfläche eines Kalenderprofils drücken. Die unterschiedlichen Profiltypen bieten entsprechende Optionen; dieses Dialogfeld dient dazu, für ein Variables Profil Intervalle einzurichten und zu verändern.

Dauer

Legen Sie hier die Dauer des Intervalls fest. Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden.

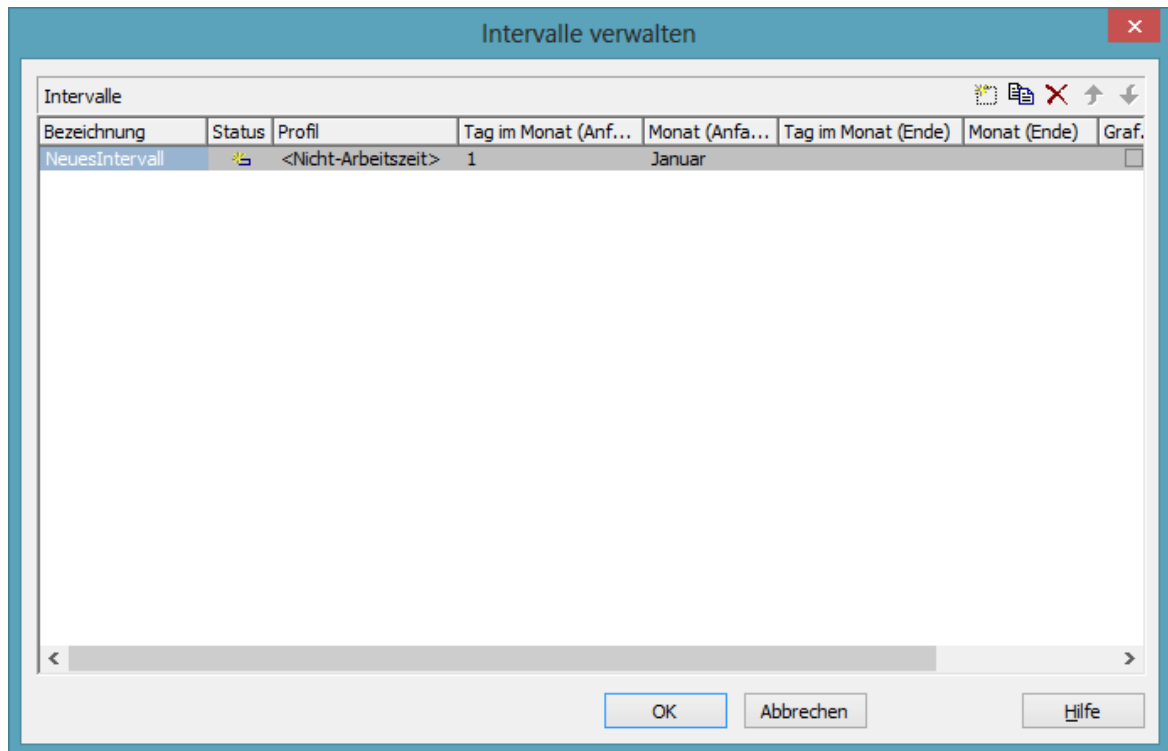
Zeiteinheit

Legen Sie hier die Zeiteinheit des Intervalls fest. Diese Option kann auch über die Eigenschaft **VcInterval.TimeUnit** gesetzt werden.

Text

Legen Sie hier einen Text fest, der im Zeitstreifen des Intervalls erscheinen soll. Diese Option kann auch über die Eigenschaft **VcInterval.Text** gesetzt werden.

4.42 Dialogfeld "Intervalle verwalten" für Jahresprofil



Zu diesem Dialogfeld gelangen Sie, wenn Sie auf der Eigenschaftenseite "Objekte" das Dialogfeld "Kalenderprofile verwalten" aktivieren und dann die "Bearbeiten"-Schaltfläche eines Kalenderprofils drücken. Die unterschiedlichen Profiltypen bieten entsprechende Optionen; dieses Dialogfeld dient dazu, für ein Jahresprofil Intervalle einzurichten und zu verändern.

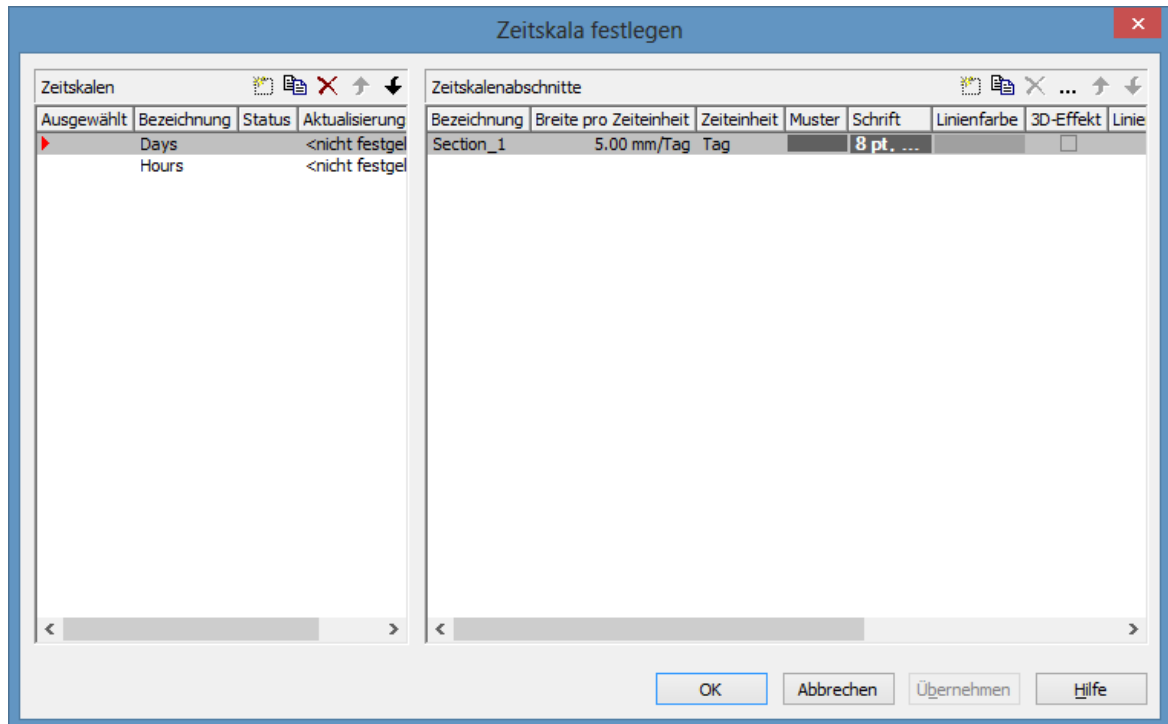
Tag im Monat (Anfang)/Tag im Monat (Ende)

Durch Klick auf  können Sie den Tag des ersten Monats des Intervalls festlegen. Diese Option kann auch über die Eigenschaft **VcInterval.DayInStart/EndMonth** gesetzt werden.

Monat (Anfang)/Monat (Ende)

Durch Klick auf  können Sie den Start-/ Endmonat des Intervalls festlegen. Diese Option kann auch über die Eigenschaft **VcInterval.Start/EndMonth** gesetzt werden.

4.43 Dialogfeld "Zeitskala festlegen"



Sie erreichen dieses Dialogfeld von der Eigenschaftenseite **Objekte**. Sie können hier verschiedene Zeitskalen und deren Zeitskalenabschnitte festlegen.

Zeitskalen

- **Ausgewählt:** Die Zeitskala, die Sie in dieser Spalte durch eine kleine Pfeilspitze markieren, wird für das Diagramm verwendet. Beachten Sie bitte, dass die Zeitskala, die Sie hier wählen, und die Zeiteinheit (Eigenschaftenseite **Allgemeines**) zueinander passen sollten.
- **Aktualisierungsverhalten:** > Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog Aktualisierungsverhalten bearbeiten für Zeitskalen festgelegt wurde.
- **Bezeichnung:** In dieser Spalte werden die Namen aller verfügbaren Zeitskalen aufgeführt. Die Namen sind editierbar.
- **Status:** In dieser Spalte wird jede Zeitskala gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (🌟) und/oder geändert (🔴) worden ist.

Zeitskala hinzufügen / kopieren / löschen / eine Zeile nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie Zeitskalen hinzufügen, kopieren, löschen und in der Tabelle nach oben oder unten verschieben.

Zeitskalenabschnitte

Diese Tabelle enthält alle für die ausgewählte Zeitskala definierten Zeitskalenabschnitte mit ihren Eigenschaften. Die folgenden Eigenschaften können Sie hier festlegen:

- **Bezeichnung** des Zeitskalenabschnitts
- **Breite pro Zeiteinheit:** Die Breite der Grundeinheit eines Zeitskalenabschnitts wird in mm/Zeiteinheit festgelegt. Die Grundeinheit ist die kleinste Einheit, in die die Zeitskala unterteilt wird. Die Breite der Grundeinheit wird in Millimetern angegeben und lässt sich in Schritten von Hundertstel Millimetern verändern. Die minimale Breite der Grundeinheiten beträgt 0,01 mm.
- **Zeiteinheit:** des Zeitskalenabschnitts: Sekunden, Minuten, Stunden, Tage.
- **Muster** des Zeitskalenabschnitts: Sie können ein anderes Muster auswählen, indem Sie auf die Schaltfläche (...) klicken und damit das Dialogfeld **Musterattribute bearbeiten** öffnen. Das hier gewählte Muster wird für alle Zeitstreifen dieses Zeitskalenabschnitts verwendet. Falls die Muster der Zeitstreifen zuvor unterschiedlich waren, werden sie damit gleich gesetzt.
- **Schrift:** Hier werden die Schriftart und -farbe angezeigt. Über die erste Schaltfläche (▼) gelangen Sie in das Dialogfeld **Farbe**, in dem Sie die Schriftfarbe festlegen können. Über die zweite Schaltfläche (...) gelangen Sie in den Windows-Dialog **Schriftart**, in dem Sie die Schriftart für die Beschriftung des Zeitskalenabschnitts auswählen können. Die hier gewählte Schrift (Schriftfarbe und -art) wird für alle Zeitstreifen dieses Zeitskalenabschnitts verwendet. Falls die Schriften der Zeitstreifen zuvor unterschiedlich waren, werden sie damit gleich gesetzt.
- **3D-Effekt:** Mit Hilfe dieser Option können Sie wählen, ob der Zeitskalenabschnitt durch einen 3D-Effekt einen räumlichen Eindruck erhalten soll.

- **Liniengitter:** Hier können Sie festlegen, ob die vordefinierten vertikalen Gitternetzlinien im Diagrammbereich unter dem Zeitskalenabschnitt dargestellt werden sollen.
- **Kalendergitter:** Hier können Sie festlegen, ob im Diagrammbereich unter dem Zeitskalenabschnitt ein vordefiniertes Kalendergitter angezeigt werden soll. Dadurch werden Wochenenden und sonstige arbeitsfreie Tage durch Flächen besonders markiert.
- **Kollabieren der arbeitsfreien Zeiten:** Hier können Sie für den Zeitskalenabschnitt festlegen, ob arbeitsfreie Zeiten ausgeblendet werden sollen oder nicht. Der Kalender, der die arbeitsfreien Zeiten bestimmt, wird im Dialog **Kalender festlegen** ausgewählt.

Zeitskalenabschnitt hinzufügen/ kopieren/ löschen/ bearbeiten/ früher/ später



Mit Hilfe dieser Schaltflächen können Sie Zeitskalenabschnitte hinzufügen, kopieren, löschen, bearbeiten und in der Tabelle nach oben bzw. unten verschieben.

Die Reihenfolge der Zeitskalenabschnitte in der Tabelle entspricht der Reihenfolge in der Darstellung.

Wenn Sie einen neuen Zeitskalenabschnitt definiert haben, werden anschließend alle Zeitskalenabschnitte in ihrer zeitlichen Ausdehnung ungefähr gleich breit dargestellt. Durch Ziehen mit der Maus können Sie diese Aufteilung noch modifizieren. Über die API können Sie den Anfang jedes Zeitskalenabschnitts festlegen bzw. bearbeiten.

4.44 Dialogfeld "Zeitskalenabschnitt bearbeiten"

Zeitskalenabschnitt "Section_1" bearbeiten

Breite pro Zeiteinheit: ☐ Kollabieren arbeitsfreier Zeiten

Zeitsstreifen mit 3D-Effekt: ☐

Index	Typ	Position	Minimale Höhe	Hauptmarkierungen	Nebenmarkierungen	Muster	Schrift	Datumsformat	Einheitentrennung
0	Monate	Oben	5.2 mm	1	1		8 pt., Arial	TM YYYY	durchgezogene Linien
1	Tage	Oben	5.2 mm	1	1		8 pt., Arial N...	DD	durchgezogene Linien

Liniengitter

Index	Sichtbar	Typ	Intervall	Linienart	Priorität	Referenzdatum verwenden	Referenzdatum	Am Referenzdatum ausrichten	Sommerzeit be
0	☒	Wochen	1 Wochen		-20	☐		☐	nicht festgele
1	☒	Monate	1 Monate		-10	☐		☐	nicht festgele

Kalendergitter

Index	Sichtbar	Kalender	Muster	Linienart	Priorität	Grafische Attribute der Intervalle verwenden	Identifizierbar	Einrast-Ziel am Start
0	☒	<nicht festgelegt>			-25	☐	☐	☐

OK Abbrechen Hilfe

In diesem Dialogfeld können Sie Zeitskalenabschnitte erstellen, einrichten und verändern.

Breite pro Zeiteinheit

Hier kann die Breite pro Zeiteinheit für den gewählten Zeitskalenabschnitt eingestellt werden. Der neue Wert wird in das entsprechende Feld der **Zeitskalenabschnitte**-Tabelle des Dialogs **Zeitskala festlegen** übertragen.

Kollabieren arbeitsfreier Zeiten

Legen Sie für den gewählten Zeitskalenabschnitt fest, ob arbeitsfreie Zeiten ausgeblendet werden sollen oder nicht. Der Kalender, der die arbeitsfreien Zeiten bestimmt, ist der, der im Dialog **Kalender festlegen** ausgewählt ist.

Wenn Sie hier eine Festlegung treffen, wird sie in die **Zeitskalenabschnitte**-Tabelle des Dialogs **Zeitskala festlegen** übernommen.

Zeitstreifen

Zeitstreifen dienen zur Beschriftung der Zeitskala. Jeder Zeitskalenabschnitt kann mehrere Zeitstreifen besitzen (beispielsweise je einen mit einer Monats- und einer Tagesskala).



Mit Hilfe dieser Schaltflächen können Sie Zeitstreifen hinzufügen, kopieren, löschen und in der Tabelle nach oben bzw. unten verschieben.

In der Tabelle können Sie Folgendes für die einzelnen Zeitstreifen des gewählten Zeitskalenabschnitts festlegen:

- **Index:** fortlaufende Nummer der Zeitstreifen (nicht editierbar)
- **Typ** des Zeitstreifens: Sekunden, Minuten, Stunden, Tage, Wochen, Monate, Quartale, Jahre, Schichten, Fiskalische Quartale, Fiskalische Jahre.
- **Position:** Legen Sie hier für jeden Zeitstreifen fest, ob er oberhalb (**oben**) oder unterhalb (**unten**) des Diagrammbereichs oder überhaupt nicht dargestellt werden soll (**keine**).
- **Minimale Höhe** des Zeitstreifens in mm
- **Hauptmarkierungen:** Legen Sie hier fest, nach wie vielen Zeiteinheiten eine Hauptmarkierung erscheint, z. B. nach jeweils sieben Tagen. (Die Zeiteinheit hängt vom Typ des verwendeten Zeitstreifens ab.) Die Hauptmarkierungen werden beschriftet, wenn ausreichend Platz vorhanden ist.
- **Nebenmarkierungen:** Legen Sie hier fest, nach wie vielen Zeiteinheiten eine unbeschriftete Nebenmarkierung erscheint, z. B. nach jeweils einem Tag. (Die Zeiteinheit hängt vom Typ des verwendeten Zeitstreifens ab.)
- **Muster:** Legen Sie hier das Muster des Zeitstreifens fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen können. Solange Sie kein anderes Muster festlegen, haben alle Zeitstreifen des Zeitskalenabschnitts das Muster, das im Dialog **Zeitskala festlegen** ausgewählt wurde. Wenn Sie für den ersten Zeitstreifen eines Zeitskalenabschnitts ein neues Muster auswählen, wird dieses auch für das **Muster**-Feld der **Zeitskalenabschnitte**-Tabelle im Dialog **Zeitskala festlegen** übernommen.
- **Schrift:** Hier werden die Schriftart und -farbe jedes Zeitstreifens angezeigt. Solange nichts anderes vereinbart wurde, haben alle Zeitstreifen des Zeitskalenabschnitts die Schriftart, die im Dialog **Zeitskala festlegen** dafür ausgewählt wurde. Um für einen Zeitstreifen eine andere

Schriftfarbe festzulegen, klicken Sie auf die erste Schaltfläche (). Sie gelangen dann in das Dialogfeld **Farbe**. Um für einen Zeitstreifen eine andere Schriftart festzulegen, klicken Sie auf die zweite Schaltfläche (). Sie gelangen dann in den Windows-Dialog **Schriftart**. Die Schrift, die Sie für den ersten Zeitstreifen eines Zeitskalenabschnitts vereinbaren, wird in das **Schrift**-Feld der **Zeitskalenabschnitte**-Tabelle im Dialog **Zeitskala festlegen** übertragen.

- **Datumsformat** des Zeitstreifens: Welche Formate möglich sind, hängt von dem gewählten Typ des Zeitstreifens ab. Für das Datum stehen folgende Kürzel zur Verfügung:

D:	Wochentagsname erster Buchstabe
TD:	Wochentagsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
DD:	Tagesnummer zweistellig: 01-31 (nicht anpassbar)
DDD:	die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
M:	erster Buchstabe des Monatsnamens (nicht anpassbar)
TM:	Monatsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
MM:	zweistellige Monatsnummer: 01-12
MMM:	erste drei Buchstaben des Monatsnamens (nicht anpassbar)
YY:	zweistellige Jahreszahl
YYYY:	vierstellige Jahreszahl
WW:	zweistellige Nummer der Kalenderwoche: 01-53
TW:	Text für "Kalenderwoche" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
Q:	einstellige Quartalsnummer: 1-4
TQ:	Quartalsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
hh:	Stunde zweistellig im 24-Stunden-Format: 00-23
HH:	Stunde zweistellig im 12-Stunden-Format: 01-12
Th:	Text für "Uhr" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis OnSupplyTextEntry angepasst werden)

- angepasst werden)
- mm: Minute zweistellig: 00-59
- ss: Sekunde zweistellig: 00-59
- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- xC/XC: Sie können das Datum als maximal zehngliedrige, einfache Aufwärtsszählung ab einem bestimmten Referenzdatum gestalten, z.B. "15:05:07:16:00". Dieses Beispiel datiert 15 Monate, 5 Tage, 7 Stunden, 16 Minuten und 0 Sekunden später als das gesetzte Referenzdatum. Die Notation dazu ist: **xC44:C33:C22:C11:C00**. Es sollen je 2 Stellen für die Glieder 4...0 vorgesehen werden, mit einem vorausgehenden "-" Symbol falls der Wert negativ ist. Die Trennzeichen sind variabel und könnten durch andere ersetzt werden. "x" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, aber kein "+"Symbol, falls er positiv ist. "X" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, und ebenso ein "+"Symbol, falls er positiv ist. Im Dialog **Zeitskalenabschnitt ... bearbeiten** sollten die Felder **Referenzdatum verwenden** und **Hauptmarkierung am Referenzdatum ausrichten** aktiviert sein, ebenso sollte die **Serielle Beschriftungen** auf **Keine** gesetzt sein. Das Referenzdatum wird zur Laufzeit über die Methode **VcRibbon.set ReferenceDate** gesetzt und überschreibt die Werte des Dialogs.

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

- **Einheitentrennung:** Zur Unterteilung des Zeitstreifens können Sie zwischen drei Möglichkeiten wählen: **durchgezogene Linien**, **Ticks** und **keine Linien**.
- **Tick-Position:** Legen Sie fest, ob die Ticks und deren Beschriftungen am oberen oder unteren Rand ausgegeben werden sollen.

- **Markierungsfarbe:** Legen Sie die Farbe der Markierungen fest.
- **Ausrichtung:** Für die Ausrichtung der Beschriftung an den Hauptmarkierungen stehen folgende Möglichkeiten zur Verfügung: **Zentriert, Rechts, Links, An den Ticks** (Hauptmarkierungen).
- **Serielle Beschriftung:** Legen Sie fest, ob auf dem Zeitstreifen serielle Nummern statt des Datums angezeigt werden sollen, und falls ja, ob Null am ggf. gesetzten Referenzdatum den Anfang bilden soll.
- **Referenzdatum verwenden:** Aktivieren Sie dieses Kontrollkästchen, falls ggf. der Ursprung dieser seriellen Nummern (oder des fiskalischen Jahrs oder oder Quartals) auf das Referenzdatum gelegt werden soll. Andernfalls wird er auf den Anfang des Zeitskalenabschnitts gelegt.
- **Referenzdatum:** Wählen Sie ggf. das Referenzdatum mit Hilfe des Datumsdialogs aus.
- **Am Referenzdatum ausrichten:** Aktivieren Sie dieses Kontrollkästchen, wenn die Linien des Liniengitters am Referenzdatum ausgerichtet werden sollen, d.h. die Position kann vom Anfang einer Zeiteinheit abweichen, z.B. um 13:17 jedes Tages.
- Ist diese Option nicht gewählt, werden die Linien eines Liniengitters standardmäßig am Anfang der jeweiligen Zeiteinheit positioniert, z.B. immer um 00:00 Uhr jedes Tages.
- **Kalender:** Falls Sie einen Schichten-Zeitstreifen darstellen möchten, wählen Sie hier einen der Schichtkalender, die Sie im Dialog **Kalender festlegen** definiert haben.
- **Sommerzeit beachten:** Aktivieren Sie dieses Kontrollkästchen, wenn auf dem Zeitstreifen die Sommerzeit berücksichtigt werden soll.

Liniengitter

Im Diagrammbereich und im Histogramm können ein oder mehrere Liniengitter (bestehend aus vertikalen Linien) unter dem gewählten Zeitskalenabschnitt dargestellt werden.



Mit Hilfe dieser Schaltflächen können Sie Liniengitter hinzufügen, kopieren, löschen und in der Tabelle nach oben bzw. unten verschieben.

In der Tabelle können Sie die Liniengitter für den gewählten Zeitskalenabschnitt festlegen:

- **Index:** fortlaufende Nummer der Liniengitter (nicht editierbar)

- **Sichtbar:** Aktivieren Sie dieses Kontrollkästchen für jedes Liniengitter, das angezeigt werden soll.
- **Typ:** Hier können Sie die Grundeinheit jedes Liniengitters festlegen, z. B. Tage, Wochen, etc.
- **Intervall:** Hier können Sie das Intervall zwischen den einzelnen Linien in ganzzahligen Vielfachen der Grundeinheit des Liniengitters festlegen.
- **Linienart:** Wenn Sie auf die Schaltfläche dieses Feldes (...) klicken, öffnet sich das Dialogfeld **Linienattribute des Liniengitters**. Hier können Sie Farbe, Typ und Dicke der Randlinien der Streifen festlegen.
- **Priorität:** Hier können Sie die Priorität jedes Liniengitters in Relation zu anderen Gittern und zu den Layern festlegen (> 0 : vor den Layern, < 0 : hinter den Layern).
- **Referenzdatum:** Das Referenzdatum versetzt den Anfang des Liniengitters um den angegebenen Abstand zu dem standardmäßigen Beginn am Montag 00.00 Uhr.
- **Sommerzeit beachten:** Aktivieren Sie dieses Kontrollkästchen, wenn für dieses Liniengitter die Sommerzeit berücksichtigt werden soll.
- **Liniengitter:** Das Liniengitter definiert seine relevanten Positionen als "Einrastziele" für zu verschiebende Knoten/Layer.

Kalendergitter

Im Diagrammbereich und im Histogramm können unter dem gewählten Zeitskalenabschnitt ein oder mehrere Kalendergitter dargestellt werden, die arbeitsfreie Zeiten durch vertikale Streifen darstellen.



Mit Hilfe dieser Schaltflächen können Sie Kalendergitter hinzufügen, kopieren, löschen und in der Tabelle nach oben bzw. unten verschieben.

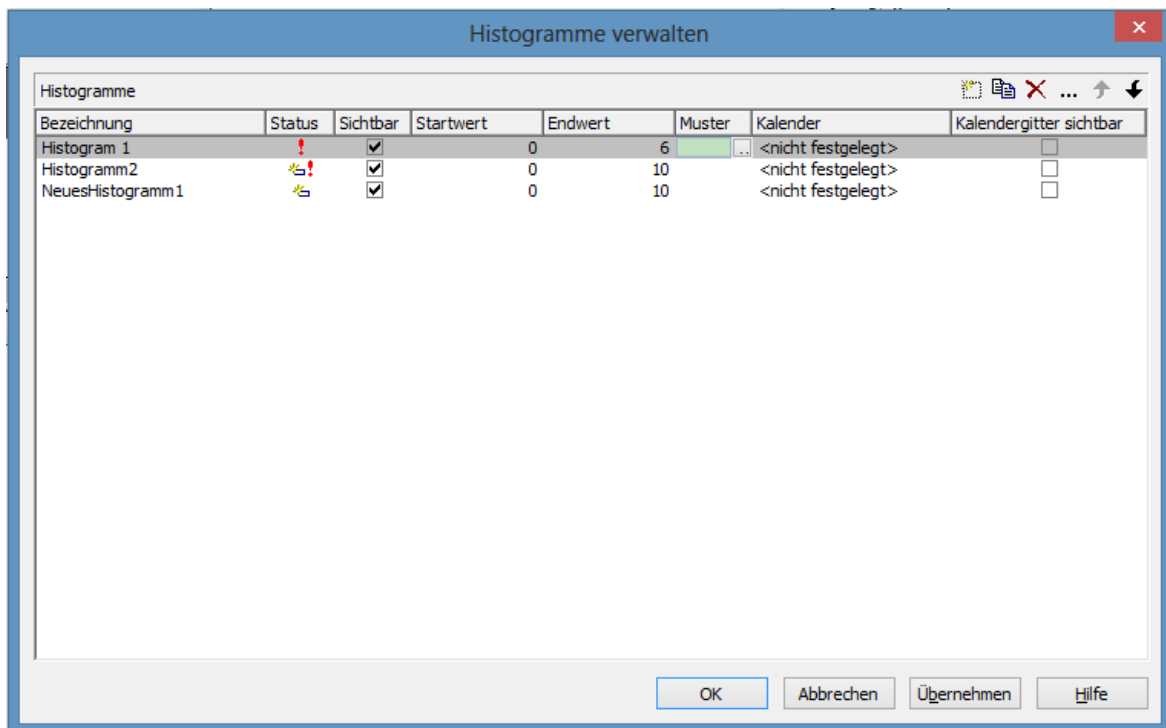
Sie können hier Folgendes für jedes Kalendergitter festlegen:

- **Index:** fortlaufende Nummer der Kalendergitter (nicht editierbar)
- **Sichtbar:** Aktivieren Sie dieses Kontrollkästchen für jedes Kalendergitter, das angezeigt werden soll.
- **Kalender:** Wählen Sie den Kalender aus, der die arbeitsfreien Zeiten bestimmen soll. Beim Eintrag <nicht festgelegt> wird der Kalender verwendet, der im Dialog **Kalender festlegen** ausgewählt ist.
- **Muster:** Wenn Sie auf die Schaltfläche dieses Feldes (...) klicken, öffnet sich das Dialogfeld **Musterattribute des Kalendergitters**. Hier können

Sie Typ, Vordergrund und Hintergrund der Streifen des Kalendergitters festlegen.

- **Linienart:** Wenn Sie auf die Schaltfläche dieses Feldes (...) klicken, öffnet sich das Dialogfeld **Linienattribute des Kalendergitters**. Hier können Sie Farbe, Typ und Dicke der Randlinien der vertikalen Linien festlegen.
- **Priorität:** Hier können Sie die Priorität jedes Kalendergitters in Relation zu den anderen Gittern und Layern festlegen (> 0: vor den Layern, < 0: hinter den Layern).
- **Kalendergitter>!: Das Kalendergitter definiert seine relevanten Positionen als "Einrastziele" für zu verschiebende Knoten/Layer.**

4.45 Dialogfeld "Histogramme verwalten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite Layout. Sie können hier mehrere Histogramme festlegen und auswählen, welche(s) davon angezeigt werden soll(en).

Vorschau

Das in der Vorschauspalte markierte Histogramm wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte werden die Namen aller verfügbaren Histogramme aufgeführt. Die Namen sind editierbar.

Status

In dieser Spalte wird jedes Histogramm gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt (☆) und/oder geändert (!) worden ist.

Sichtbar

Mit Hilfe dieses Kontrollkästchens können Sie festlegen, ob das gewählte Histogramm dargestellt werden soll oder nicht.

Startwert

Legen Sie hier den niedrigsten Wert der numerischen Skala des Histogramms fest. Gegebenenfalls wird der niedrigste Wert der numerischen Skala an die Kurvenwerte angepasst.

Endwert

Legen Sie hier den höchsten Wert der numerischen Skala des Histogramms fest. Gegebenenfalls wird der höchste Wert der numerischen Skala an die Kurvenwerte angepasst.

Muster

Hier können für jedes Histogramm Muster und Farbe festgelegt werden.

Histogramm hinzufügen



Ein neues Histogramm wird erzeugt.

Histogramm kopieren



Das markierte Histogramm wird kopiert.

Histogramm löschen



Das markierte Histogramm wird gelöscht.

Histogramm bearbeiten



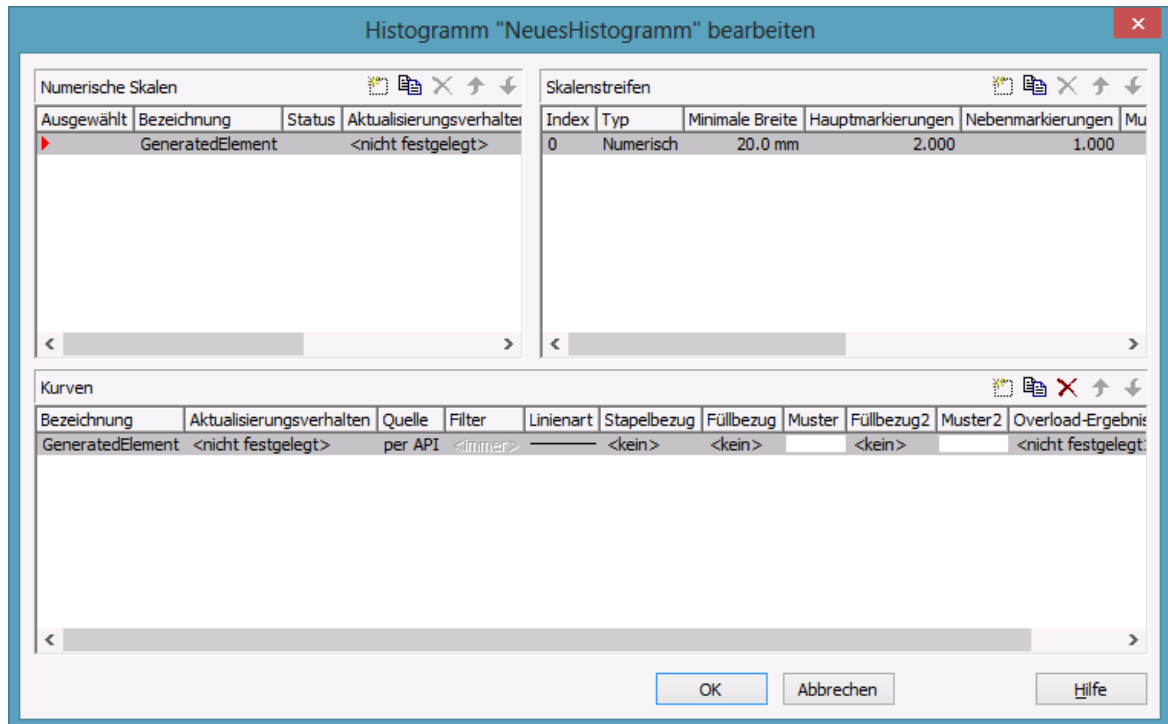
Der Dialog **Histogramm bearbeiten** wird geöffnet.

Histogramm eine Zeile nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie das markierte Histogramm eine Zeile nach oben/unten schieben. Die Position der Histogramme in der Tabelle bestimmt deren Ausgabereihenfolge.

4.46 Dialogfeld "Histogramm bearbeiten"



Sie erreichen dieses Dialogfeld, indem Sie im Dialogfeld **Histogramme verwalten** auf die **Histogramm bearbeiten**-Schaltfläche (...) klicken.

Sie können hier für das zu bearbeitende Histogramm verschiedene numerische Skalen mit jeweils einem oder mehreren Skalenstreifen definieren und auswählen, welche davon dargestellt werden soll.

Das Histogramm kann mehrere Kurven enthalten. Sie können für jede Kurve festlegen, aus welcher Quelle die Daten dafür stammen sollen. Durch Filter kann für jede Kurve festgelegt werden, welche Vorgänge zu der Kurve beitragen sollen. Außerdem können Sie das Aussehen der Kurven definieren.

Numerische Skalen

- **Ausgewählt:** Die rote Pfeilspitze markiert die zu verwendende numerische Skala.
- **Bezeichnung** der numerischen Skala
- **Status:** In dieser Spalte wird jede numerische Skala gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (🔍) und/oder geändert (🔴) worden ist.
- **Aktualisierungsverhalten:** Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das

Aktualisierungsverhalten, das im Dialog Aktualisierungsverhalten bearbeiten für die numerische Skala festgelegt wurde.

- **Breite pro Einheit** in mm, legt den Abstand zwischen den Hauptmarkierungen fest
- **Einheit gibt die Schrittweite der Hauptmarkierungen an**
- **Linienfarbe** Legen Sie hier die Markierungsfarbe aller numerischer Skalenstreifen von Histogrammen fest
- **Liniengitter:** Legen Sie hier fest, ob ein Liniengitter angezeigt werden soll oder nicht.
- **Linienart:** Hier wird der Linientyp des Liniengitters angezeigt. Um ihn zu ändern, klicken Sie auf die Schaltfläche (...) im Feld. Dann öffnet sich der Dialog **Linie bearbeiten**.

Skalenstreifen

Legen Sie hier für jeden Skalenstreifen der markierten numerischen Skala Folgendes fest:

- **Index:** fortlaufende Nummer des Skalenstreifens (nicht editierbar)
- **Typ** des Skalenstreifens: **Numerisch** oder **Textuell**. Über die Schaltfläche öffnen Sie einen Dialog, in dem Sie den Typ festlegen können.
- **Minimale Breite** in mm
- **Hauptmarkierungen:** Legen Sie hier fest, nach wie vielen Einheiten eine Hauptmarkierung (beschrifteter Teilstrich) erscheint.
- **Nebenmarkierungen:** Legen Sie hier fest, nach wie vielen Einheiten eine Nebenmarkierung (ein kleinerer, unbeschrifteter Teilstrich) erscheint.
- **Muster:** Durch Klick auf ... gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Musterfarbe sowie eine Hintergrundfarbe oder 2. Musterfarbe für den Skalenstreifen auswählen können. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.
- **Schrift:** Die Schriftart des Skalenstreifens wird hier angezeigt. Über die Schaltfläche (...) erreichen Sie den Windows-Dialog **Schriftart**.

- **Doubleformat:** Hier können Sie aus verschiedenen Zahlenausgabeformaten wählen, wobei **I** für die Ziffern vor dem Dezimaltrenner und **D** für je eine Ziffer hinter dem Dezimaltrenner steht.
- **Markierungsfarbe:** Legen Sie hier die Markierungsfarbe aller numerischer Skalenstreifen fest
- **ObjectDraw-Ereignisse:** Wählen Sie diese Option, wenn Sie die beiden Ereignisse **OnObjectDrawEx** und **OnObjectDrawCompleteEx** auslösen möchten.

Mit dem Ereignis **OnObjectDrawEx** wird der Standardskalenstreifen durch einen benutzerdefinierten ersetzt, das Ereignis **OnObjectDrawCompleteEx** wird hierbei nicht benötigt.

Das Ereignis **OnObjectDrawCompleteEx** ermöglicht es Ihnen, dem von VARCHART XGantt gezeichneten Skalenstreifen noch etwas hinzuzufügen.

- **Einheitenbezeichnung:** Beschriftung der Einheiten der numerischen Skala.

Kurven

- **Name:** In dieser Spalte werden die Namen aller verfügbaren Kurven angezeigt.
- **Aktualisierungsverhalten:** Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog **Aktualisierungsverhalten bearbeiten** für Kurven festgelegt wurde.
- **Quelle:** Hier können Sie für jede einzelne Kurve festlegen, aus welcher Quelle die Daten für die Berechnung der einzelnen Kurven stammen sollen. Sie können zwischen zwei grundsätzlichen Alternativen wählen:
 - 1. per Layer:** Die Kurve wird aus den Daten der gewählten Layer aller Vorgänge, die die Layer-Filterkriterien erfüllen, generiert. Über die Wahl eines Filters können Sie diese Vorgänge noch weiter eingrenzen.
 - 2. per API:** Sie können die Werte über die Programmierschnittstelle (API) mit der VcCurve-Methode **SetValues** eingeben.

Über die **Bearbeiten**-Schaltfläche () können Sie in beiden Fällen den Dialog **Datenquelle der Kurve einstellen** aufrufen.
- **Filter:** Wählen Sie für jede Kurve den Filter aus, der bestimmt, welche Vorgänge zu der Kurve beitragen sollen. Über die **Bearbeiten**-Schaltfläche () können Sie ggf. den Dialog **Filter verwalten** aufrufen.

- **Linienart:** Wenn Sie auf den Eintrag dieses Feldes klicken, öffnet sich das Dialogfeld **Linie bearbeiten**, in dem Sie das Aussehen der Kurvenlinie festlegen können.
- **Stapelbezug:** Wenn Sie die Kurven stapeln möchten, geben Sie für jede einzelne Kurve an, auf welche Kurve sie gestapelt werden soll. Sie können so sukzessive die Kurven aufeinander stapeln, um so beispielsweise die Gesamtauslastung Ihrer Ressourcen (z. B. Arbeitsgruppen) darzustellen. Soll eine Kurve nicht auf eine andere Kurve gestapelt werden, wählen Sie hierfür den Eintrag "<kein>". Wählen Sie für alle Kurven den Eintrag "<kein>", werden die Kurven nicht aufeinander gestapelt, sondern überlagern einander gegebenenfalls. Um die Kurven dennoch voneinander unterscheiden zu können, sollten Sie sie durch unterschiedliche Muster kennzeichnen.
- **Füllbezug:** Hier können Sie festlegen, bis wohin die Füllung unterhalb jeder Kurve reichen soll. Wählen Sie hier für eine Kurve den Eintrag "<kein>", wird keine Füllung unter dieser Kurve dargestellt. Wählen Sie "<Null-Linie>", so reicht die Füllung unter dieser Kurve bis zur Null-Linie. Geben Sie hier eine andere Kurve an, reicht die Füllung nur bis zur dieser Kurve.
- **Muster:** Hier können Sie das Muster der Füllung unter jeder Kurve festlegen. Über die **Bearbeiten**-Schaltfläche () erreichen Sie den Dialog **Muster**, in dem Sie das Muster bearbeiten können.
- **Füllbezug 2:** Wählen Sie hier eine 2. Bezugskurve aus. Die Füllung zur 2. Bezugskurve wird nur dargestellt, wenn die y-Werte der in dieser Zeile definierten Kurve größer sind als die der 2. Bezugskurve.
- **Muster 2:** Hier können Sie das Muster und die Farbe der Füllung über der 2. Bezugskurve festlegen.
- **Overload-Ergebnis** Wählen Sie aus der Drop-Down-Liste einen von Ihnen zuvor erstellten Kalender, in den die Intervalle, die sich aus den berechneten Overload-Terminen ergeben, geschrieben werden. Dieser Kalender kann z.B. für die Darstellung eines Kalendergitters in der Gruppe verwendet werden.

Beispiele zur Verwendung von Histogrammen finden Sie unter "Tutorium: Histogramme erstellen" sowie "Tutorium: Kapazitätsengpässe darstellen".

Numerische Skala/Zeitstreifen/Kurve hinzufügen



Ein neues Objekt wird erzeugt.

Numerische Skala/Zeitstreifen/Kurve kopieren

 Das markierte Objekt wird kopiert.

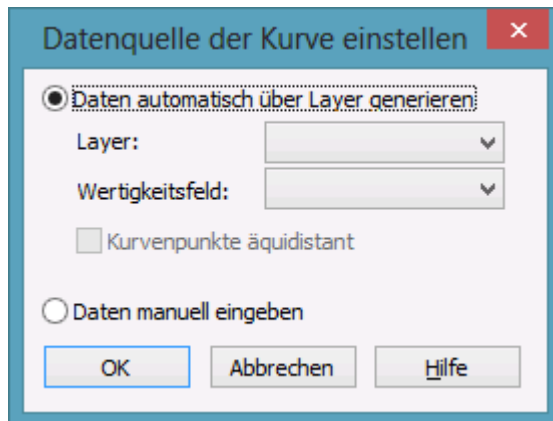
Numerische Skala/Zeitstreifen/Kurve löschen

 Das markierte Objekt wird gelöscht.

Numerische Skala/Zeitstreifen/Kurve eine Zeile nach oben/unten

  Mit Hilfe dieser Schaltflächen wird das markierte Objekt eine Zeile nach oben/unten geschoben.

4.47 Dialogfeld "Datenquelle der Kurve einstellen"



Sie erreichen dieses Dialogfeld über das Dialogfeld **Histogramm bearbeiten**.

Daten automatisch über Layer generieren

Wählen Sie diese Option, wenn die Daten automatisch über Layer generiert werden sollen. Beim Aufsummieren der Vorgänge zu einer Kurve werden für jeden Vorgang die Anfangs- und Enddaten des gewählten Layertyps (z.B. des Layers "Start-Ende") übernommen. Legen Sie dann Folgendes fest:

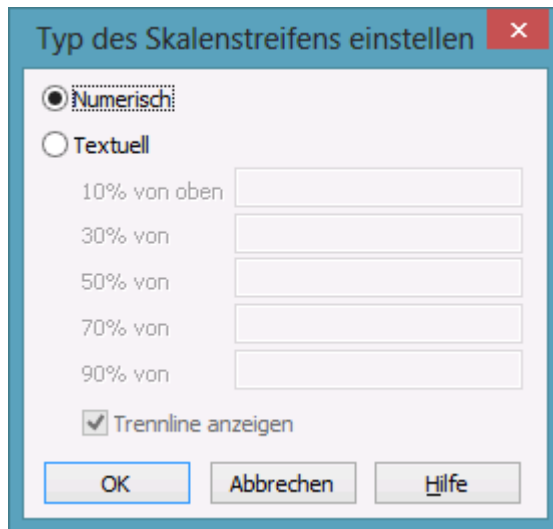
- **Layer**
- **Wertigkeitsfeld:** Datenfeld, aus dem pro Vorgang die Wertigkeit für die Addition der Kapazitäten entnommen wird.

Daten manuell eingeben

Wählen Sie diese Option, wenn die Daten manuell (per API) eingegeben werden sollen. Für diese Option können Sie außerdem die Option **Kurvenpunkte äquidistant** wählen. Andernfalls werden Kurvenpunkte nur an den Stellen erzeugt, an denen sich der y-Wert ändert.

Weitere Informationen finden Sie im Kapitel "Wichtige Begriffe: Histogramme".

4.48 Dialogfeld "Typ des Skalenstreifens einstellen"



Sie erreichen dieses Dialogfeld über das Dialogfeld **Histogramm bearbeiten**.

Numerisch

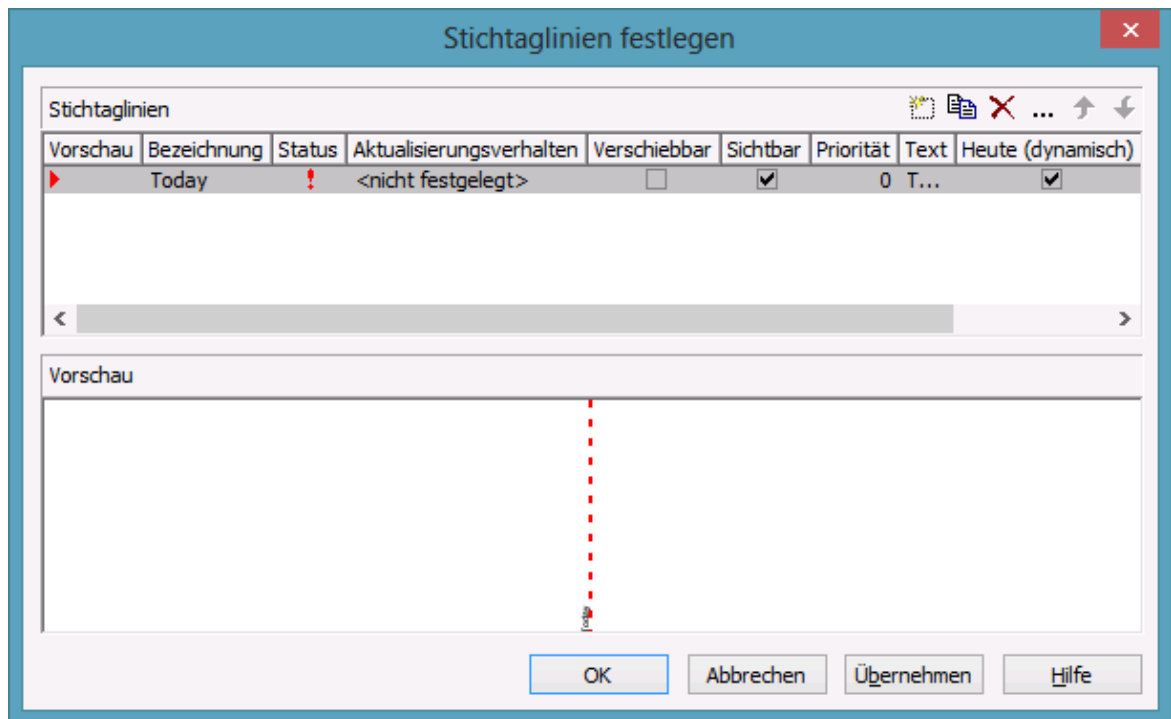
Wählen Sie diese Option, wenn der gewählte Skalenstreifen numerisch beschriftet werden soll.

Textuell

Wählen Sie diese Option, wenn der gewählte Skalenstreifen mit Texten Ihrer Wahl beschriftet werden soll. Sie können Text an insgesamt fünf vorgegebenen Positionen ausgeben (10%, 30%, 50%, 70 %, 90 % von oben).

Wenn Sie im Dialog **Histogramme bearbeiten** mehrere Skalenstreifen definiert haben, können Sie durch Anklicken der Option **Trennlinie anzeigen** bestimmen, ob rechts von dem jeweiligen Streifen eine senkrechte Trennlinie gezogen werden soll.

4.49 Dialogfeld "Stichtaglinien festlegen"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**.

Durch Stichtaglinien (vertikale Linien im Diagramm) können Sie bestimmte Daten (das aktuelle Tagesdatum oder beliebig gesetzte Termine) besonders hervorheben. Alle Stichtaglinien, die in der Grafik dargestellt werden, werden in diesem Dialogfeld verwaltet.

Vorschau

Die Stichtaglinie, die hier durch eine kleine rote Pfeilspitze gekennzeichnet ist, wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte werden die Namen aller Stichtaglinien aufgeführt, die in der Grafik dargestellt werden. Die Bezeichnungen sind editierbar.

Status

In dieser Spalte wird jede Stichtaglinie gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt (☀️) und/oder geändert (❗️) worden ist.

Aktualisierungsverhalten

Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog **Aktualisierungsverhalten bearbeiten** für Stichtaglinien festgelegt wurde.

Verschiebbar

Aktivieren Sie dieses Kontrollkästchen, wenn die Stichtaglinie zur Laufzeit verschiebbar sein soll.

Sichtbar

Aktivieren Sie dieses Kontrollkästchen, wenn die Stichtaglinie zur Laufzeit sichtbar sein soll.

Priorität

Hier können Sie die Priorität jeder Stichtaglinie festlegen (> 0: vor den Layern, < 0: hinter den Layern).

Text

Sie können hier einen Text eingeben, der an der Stichtaglinie ausgegeben werden soll.

Heute (dynamisch)

Aktivieren Sie dieses Kontrollkästchen, wenn die Stichtaglinie beim Start immer die aktuelle Systemzeit anzeigen soll. Das Feld **Datum** ist dann deaktiviert.

Datum

Sie können das Datum der Stichtaglinie editieren, indem Sie einen Teil des Datums markieren und dann mit Hilfe der Cursortasten einen neuen Wert auswählen. Oder klicken Sie auf die Pfeil-Schaltfläche () . Dann erscheint das Datum-Steuerelement. Darin ist das gewählte Datum markiert bzw. wenn kein Datum gewählt ist, das aktuelle Datum. Sie können per Mausklick einen Tag des dargestellten Monats auswählen und ggf. mit Hilfe der Pfeil-Schaltflächen von einem Monat zum anderen wechseln. Wenn Sie auf den Monatsnamen klicken, öffnet sich eine Kombobox mit den Monatsnamen.

Wenn Sie auf die Jahreszahl klicken, erscheinen Pfeil-Schaltflächen, mit denen Sie die Jahreszahl jeweils um ein Jahr erhöhen bzw. verringern können. Wenn Sie auf **Heute** klicken, wird das aktuelle Tagesdatum gewählt.



Datum

Aktivieren Sie dieses Kontrollkästchen, wenn die Stichtaglinie mittels der VcGantt-Methode **IdentifyObjectAt** identifizierbar sein soll.

Die Option kann auch mithilfe der Eigenschaft **VcDateLine.Identifiable** festgelegt werden.

Einrastziel

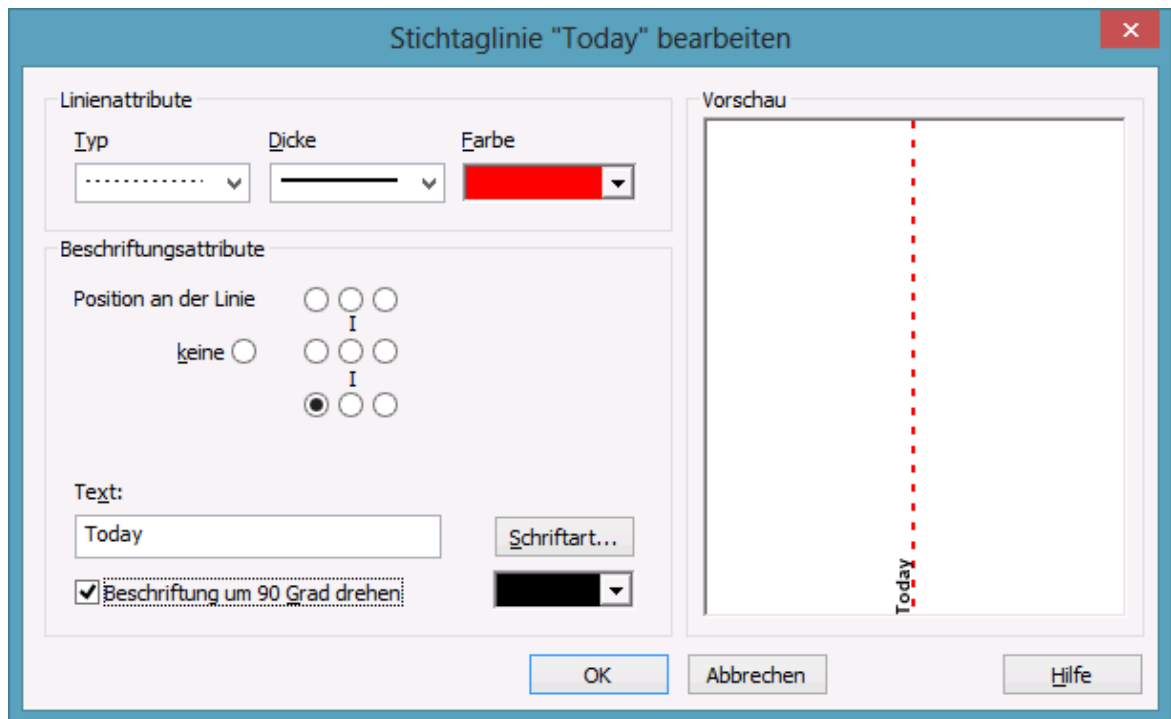
Legen Sie hier fest, ob die Stichtaglinie ihre Position (und damit ihr Datum) als "Einrast"-Ziel für das Verschieben eines Knotens/Layers definiert.

Stichtaglinie hinzufügen / kopieren / löschen / eine Zeile nach oben / unten



Mit Hilfe dieser Schaltflächen können Sie Stichtaglinien hinzufügen, kopieren, löschen und in der Tabelle nach oben bzw. unten verschieben.

4.50 Dialogfeld "Stichtaglinie bearbeiten"



In diesem Dialogfeld können Sie Stichtaglinien erstellen, einrichten und verändern.

Linienattribute

Wählen Sie hier **Typ**, **Dicke** und **Farbe** der Stichtaglinie aus.

Position an der Linie

Wählen Sie hier aus, an welcher Position ggf. ein Text ausgegeben werden soll. Wenn kein Text ausgegeben werden soll, aktivieren Sie den Radiobutton **keine**. Er ist standardmäßig aktiviert, wenn kein Text für die Stichtaglinie eingegeben wurde. Wenn Sie einen Text für die Stichtaglinie eingegeben und das **Text**-Feld wieder verlassen haben, wird standardmäßig die Position oben rechts an der Linie gewählt. Sie können ggf. eine andere Position an der Linie wählen.

Text

Sie können hier einen Text eingeben, der an der Stichtaglinie ausgegeben werden soll. Standardmäßig ist das **Text**-Feld leer. Sobald Sie eine Position an der Linie wählen, wird der Name der Stichtaglinie in das **Text**-Feld gesetzt. Sie können den Text beliebig editieren.

Schriftart

Über diese Schaltfläche gelangen Sie in den Windows-Dialog **Schriftart**, in dem Sie die Schriftart für die Beschriftung der Stichtaglinie festlegen können. Über die darunter liegende Schaltfläche gelangen Sie in den Windows-Farbselektor, über den Sie die Schriftfarbe wählen oder eine neue Farbe einrichten können.

Beschriftung um 90 Grad drehen

Aktivieren Sie dieses Kontrollkästchen, wenn die Beschriftung der Stichtaglinie in vertikaler Richtung ausgegeben werden soll.

4.51 Dialogfeld "Texte, Grafiken und Legende festlegen"

Sie erreichen dieses Dialogfeld, indem Sie auf der Eigenschaftenseite **Außenbereich** auf eine der neun Schaltflächen ober- bzw. unterhalb der Grafik klicken.

Art des Inhalts

Wählen Sie hier die Art des Inhalts, der in dem zuvor gewählten Bereich der Darstellung ausgegeben werden soll:

- **Leer:** Der gewählte Diagrammbereich bleibt leer.
- **Text:** In dem gewählten Diagrammbereich wird der Text der sechs Textzeilen dargestellt.
- **Grafik:** Sie können in dem gewählten Bereich eine Grafik (z.B. Ihr Firmenlogo) platzieren. Grafiken werden immer mittig ausgerichtet.

- **Legende:** Eine Legende wird in dem gewählten Diagrammbereich ausgegeben. Sie dokumentiert die Layer, die in der aktuellen Darstellung auftreten.

Die jeweils nicht benötigten Bereiche des Dialogs werden abhängig von Ihrer Auswahl deaktiviert. Dabei bleiben alle Angaben erhalten.

Legendenattribute

*Nur aktiv, wenn das Kontrollkästchen **Legende** angeklickt wurde.* Sie gelangen in den gleichnamigen Dialog, der für die Legende weitere Gestaltungsmöglichkeiten bietet.

Grafikdatei

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Tragen Sie hier den Namen der Grafikdatei ein. Falls sich die gewünschte Grafikdatei nicht im Installationsverzeichnis des VARCHART ActiveX befindet, müssen Sie das Laufwerk und den Pfad auch angeben.

Durchsuchen

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Wenn Sie auf diese Schaltfläche klicken, erscheint der Windows-Dialog **Grafikdatei auswählen**, mit dessen Hilfe Sie das Laufwerk, das Verzeichnis und den Dateinamen der Grafik festlegen können.

Textzeilen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Vereinbaren Sie den Text (max. 6 Zeilen), mit denen der gewählte Diagrammbereich beschriftet werden soll. Sie können einen beliebigen Text eintragen, aber auch Platzhalter (z.B. &[System-Datum]) für Projektdetails einsetzen. Sind alle sechs Textzeilen leer, wird dieser Bereich nicht dargestellt.

Projektdetails

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Hier können Sie dem Diagramm verschiedene Informationen (Anzahl der Seiten, Seitennummer, Systemdatum) hinzufügen, indem Sie aus der Kombobox den gewünschten Platzhalter auswählen und auf die Schaltfläche **Hinzufügen** klicken. Die Platzhalter werden in der Druckvorschau oder beim Ausdruck

durch die entsprechenden Daten ersetzt und stets auf dem aktuellen Stand gehalten.

Hinzufügen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Nachdem Sie aus der Liste ein Projektdetail ausgewählt haben, bestätigen Sie Ihre Wahl mit der Schaltfläche **Hinzufügen**. Die Projektdetails werden in die Zeile geschrieben, in der sich der Cursor gerade befindet.

Textausrichtung

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über die Kontrollkästchen können Sie die Textzeilen linksbündig, zentriert oder rechtsbündig ausrichten.

Schrift für alle Zeilen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**. Hier können Sie die Schriftart, Schriftgröße usw. für alle sechs Zeilen festlegen. Beim Ausführen dieser Aktion werden die Einstellungen für die Schrift der einzelnen Zeilen überschrieben.

Schrift für Zeile 1...6

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**. Hier können Sie die Schriftart, Schriftgröße usw. für die Zeile festlegen, in der sich der Cursor befindet.

Alle Texte löschen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Wenn Sie diese Schaltfläche anklicken, werden die Texte aller sechs Textzeilen gelöscht.

Max. Höhe (mm)

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Falls mehrere Felder für Text, Grafik oder Legende vereinbart worden sind, können Sie hier die maximale Höhe des aktuellen Feldes festlegen. So können Sie verhindern, dass Feldinhalte abgeschnitten werden.

Max. Breite (mm)

*Nur aktiv, wenn das Kontrollkästchen **Text** oder **Grafik** angeklickt wurde. Falls mehrere Felder für Text, Grafik oder Legende vereinbart worden sind, können Sie hier die max. Breite des aktuellen Feldes festlegen. So können Sie verhindern, dass Feldinhalte abgeschnitten werden.*

4.52 Dialogfeld "Legendenattribute"

Sie erreichen dieses Dialogfeld zur Laufzeit über das Kontextmenü der Legende oder zur Designzeit über den Dialog **Texte, Grafiken und Legende festlegen** durch Klick auf die entsprechende Schaltfläche. Diese wird erst wählbar, nachdem Sie bei **Art des Inhalts Legende** ausgewählt haben.

Legendentitel sichtbar

Legen Sie hier fest, ob ein Legendentitel angezeigt werden soll und geben Sie einen Text ein. Durch Klick auf **Schriftart** öffnen Sie den gleichnamigen Windows-Dialog, in dem Sie die Schriftattribute für den Legendentitel festlegen können.

Anordnung

- nach Zeilenanzahl: Geben Sie hier an, ob und mit wie vielen Zeilen die Legende dargestellt werden soll.
- nach Spaltenanzahl: Geben Sie hier an, ob und mit wie vielen Spalten die Legende dargestellt werden soll.

- Nach Zeilen- und Spaltenanzahl: Die Legende wird in Spalten und Zeilen dargestellt. Ist die hier eingegebene Zahl niedriger als die vorhandenen Layern, so werden die überzähligen Layer nicht dargestellt.

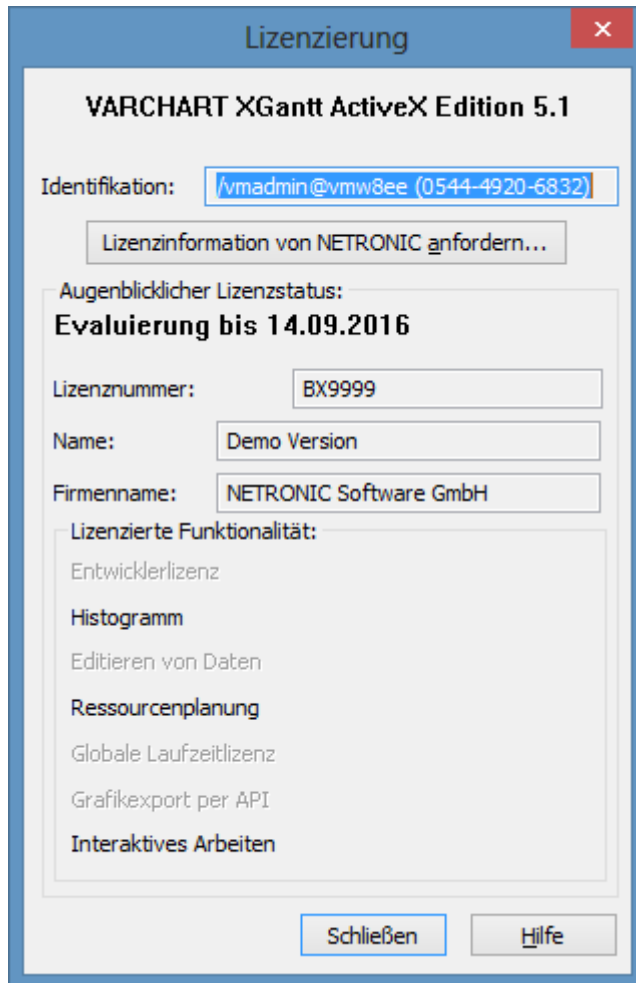
Ränder

- oberer Rand: Geben Sie ein Maß für den oberen Rand des Legendenelements an.
- unterer Rand: Geben Sie ein Maß für den unteren Rand des Legendenelements an.

Schriftart

Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**, in dem Sie die Schriftattribute für die Legende festlegen können.

4.53 Dialogfeld "Lizenzierung"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Allgemeines**.

Vor der Lizenzierung ist das Programm nur als Demoversion lizenziert. Folgende Einschränkungen gelten gegenüber der Vollversion: Die Nutzungsdauer zum Testen des Produkts ist auf 30 Tage begrenzt. Nach Ablauf dieses Zeitraums erscheint das Wort "Demo" im Diagramm.

Hardware-Identifikation

(nicht editierbar) Die Nummer, die in diesem Feld angezeigt wird, wird aus der Hardware-Konfiguration Ihres Rechners berechnet. Sie wird von NETRONIC Software GmbH für die Lizenzierung benötigt.

Bei Änderungen an Ihrer Hardware ist eine neue Lizenzierung erforderlich. Der Kundendienst von NETRONIC Software GmbH hilft Ihnen dann gern weiter.

Anfordern

Um die Lizenzierung vorzunehmen, klicken Sie auf diese Schaltfläche. Dann erscheint der Dialog **Lizenzinformationen anfordern**.

Lizenznummer/Name/Firmenname

(nicht editierbar) Hier werden Ihre Lizenznummer, Ihr Name und der Name Ihrer Firma angezeigt.

Lizenzierte Funktionalität

Hier wird angezeigt, welche Module für Sie freigegeben wurden. Wenn Sie die Lizenzierung vorgenommen haben, sind die freigegebenen Module aktiviert.

- **Entwicklerlizenz**
- **Histogramm**
- **Globale Laufzeitlizenz** (das VARCHART ActiveX-Steuerlelement läuft im Runtime-Modus auf jedem anderen Rechner.)
- **Einzelplatz-Laufzeitlizenzen** (das VARCHART ActiveX-Steuerlelement muss auf jedem Rechner, auf dem es im Runtime-Modus laufen soll, einzeln lizenziert werden.)
- **Grafikexport per API**
- **Interaktives Arbeiten**

Schließen

Sie verlassen den Dialog.

4.54 Dialogfeld "Lizenzinformationen anfordern"

Lizenzinformationen anfordern

VARCHART XGantt ActiveX Edition 5.0

Hardware-Identifikation: 0544-4920-6832

Erster Schritt: Geben Sie Ihre Benutzerdaten ein:

Lizenznummer:

Name:

Firmenname:

Zweiter Schritt: Fordern Sie Ihre Lizenzinformationen an:

Wenn Sie keine E-Mail direkt von Ihrem Computer versenden können, kontaktieren Sie NETRONIC Software GmbH unter Angabe der obigen vier Einträge:

E-Mail: license@netronic.com
Telefon: +49/2408/141-0
Fax: +49/2408/141-33

Dritter Schritt: Wenn Sie die Lizenzinformationsdatei erhalten haben, kopieren Sie diese in das gleiche Verzeichnis wie die OCX-Datei.

Geben Sie Ihre Lizenznummer, Ihren Namen und den Namen Ihrer Firma an und klicken Sie auf **E-Mail an NETRONIC senden**. Damit wird automatisch eine E-Mail generiert, die Sie nur noch absenden müssen. Sobald wir Ihre E-Mail erhalten haben, werden wir unverzüglich eine Datei mit Ihren Lizenzinformationen (**vcgantt.lic**) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei **vcgantt.ocx** steht.

Wenn Sie die neue Lizenzierung vorgenommen haben, müssen Sie diese in jedem Ihrer Projekte aktivieren. Öffnen Sie dazu in jedem Ihrer Projekte eine beliebige Eigenschaftenseite, nehmen Sie dort eine beliebige Änderung vor und speichern Sie diese. Nun ist die neue Lizenzierung aktiviert.

5 Benutzerschnittstelle

5.1 Übersicht

Die folgende Liste gibt einen Überblick über die Interaktionsmöglichkeiten für Anwender.

- Navigation in Diagramm und Tabelle
- Zoomen
- Knoten bzw. Layer markieren
- Knoten erzeugen
- Knoten verschieben
- Layer verschieben
- Anfangs-/Enddatum verändern
- Knoten löschen, ausschneiden, kopieren und einfügen
- Knotendaten bearbeiten
- Verbindungen bearbeiten
- Boxen an Knoten verankern
- Gruppendaten bearbeiten
- Gruppen expandieren/kollabieren
- Gruppen verschieben
- Breitenverhältnis Tabelle/Diagramm bearbeiten
- Breite der Tabellenspalte bearbeiten
- Bearbeiten von Feldinhalten in der Tabelle
- Einfügen von Tabellenzeilen
- Zeitskala bearbeiten
- Skalierung und Grenzen von Zeitskalenabschnitten bearbeiten
- Stichtaglinien verschieben
- Legende bearbeiten
- Seite einrichten

- Druckvorschau verwenden

Kontextmenüs (rechte Maustaste):

- für das Diagramm
- für Knoten
- für Verbindungen
- für Gruppen
- für die Zeitskala
- für das Histogramm
- für die Legende
- für Boxen

Weitere Informationen zu Interaktionsmöglichkeiten bei gruppierter oder hierarchischer Anordnung finden Sie im Kapitel "Wichtige Begriffe" unter "Gruppierung" bzw. "Hierarchie".

Bei allen Interaktionen wird ein Ereignis ausgelöst, sodass Sie im Programm darüber informiert werden und ggf. darauf reagieren können.

5.2 Navigation in Diagramm und Tabelle

Folgende Tasten und Tastenkombinationen stehen für das Navigieren in Diagramm und Tabelle zur Verfügung:

- die Pfeiltasten verschieben die Markierung von Knoten oder Tabellenfeldern in die entsprechende Richtung (für weitere Informationen dazu, insbesondere auch zum Markieren innerhalb von Gruppen s. Kapitel 5.4 "Knoten bzw. Layer markieren")
- **Pos1**: an den linken Diagrammrand scrollen
- **Strg + Pos1**: an die linke obere Ecke des Diagramms scrollen
- **Ende**: an den rechten Diagrammrand scrollen
- **Strg + Ende**: an die rechte untere Ecke des Diagramms scrollen
- **Bild rauf/runter**: eine Bildschirmseite rauf/runter scrollen
- **Strg + Shift J**: zur nächsten Stichtaglinie scrollen
- **Strg + *** (Nummernblock): der Bildschirmausschnitt wird ggf. so verschoben, dass der Anfang des Knotens sichtbar ist

Auch mit der Maus kann navigiert werden:

- Drehen Sie das Mousrad, um vertikal im Diagramm bzw. im Histogramm (abhängig von der Cursorposition) zu scrollen
- Drücken Sie das Mousrad (bzw. die mittlere Maustaste) und bewegen die Maus entsprechend, um in sämtliche Richtungen zu scrollen.

5.3 Zoomen

Mithilfe der folgenden Tastenkombinationen lässt sich die Darstellung vergrößern bzw. verkleinern:

- **Strg** + - (Nummernblock): Verkleinern
- **Strg** + + (Nummernblock): Vergrößern

Auch die Maus kann zum Zoomen genutzt werden:

- Drehen Sie das Mausrad während Sie die Strg-Taste gedrückt halten. Dazu muss das Zoomen per Mausrad zugelassen sein. Dies geschieht entweder über die Option **Zoomen per Mausrad zulassen** auf der Eigenschaftenseite **Allgemeines** oder über die API-Eigenschaft **VcGantt1.-ZoomingPerMouseWheelAllowed**. (Diese Eigenschaft ist standardmäßig deaktiviert.)

Weitere Information zu den Zoommöglichkeiten für den Druck finden Sie in Kapitel 5.21 "Seite einrichten".

5.4 Knoten bzw. Layer markieren

Einen einzelnen Knoten markieren Sie durch Anklicken des Knotens mit der linken Maustaste. Dabei wird auch das jeweils erste Feld in der entsprechenden Tabellenzeile markiert.

Sie können jedoch auch direkt auf ein bestimmtes Feld in der Tabelle klicken - damit wird gleichzeitig der entsprechende Vorgang im Diagrammbereich markiert.

Mehrere Knoten, die sich im Diagrammbereich, unter- oder übereinander befinden, werden markiert, indem Sie bei gedrückter Umschalt-Taste diese Knoten im Diagrammbereich bzw. die entsprechenden Zeilen im Tabellenbereich anklicken, oder indem Sie mit der Maus ein Rechteck um die Knoten bzw. Zeilen aufziehen.

Mehrere Knoten die sich nicht unter- oder übereinander befinden, markieren Sie, indem Sie sie bei gedrückter Strg-Taste im Diagrammbereich oder im Tabellenbereich die entsprechenden Zeilen anklicken.

Die Markierungen können sowohl im Diagramm- als auch im Tabellenbereich mit Hilfe der Cursortasten oder der Enter-Taste in die entsprechende Richtung verschoben werden.

Bei Gruppen im Modus **Knoten in einer Zeile** und **optimiert** gilt Folgendes: Von oben kommend wird zuerst der erste Knoten der Gruppe markiert; von unten kommend zuerst der letzte Knoten der Gruppe. Innerhalb dieser Art von Gruppen kann mit den Pfeiltasten nach rechts bzw. links navigiert werden.

Hinweis: Durch erneutes Anklicken der Knoten bzw. Tabellenfelder/-zeilen oder mit Hilfe der Taste ESC werden die Markierungen wieder aufgehoben.

5.5 Knoten erzeugen

Um Knoten erzeugen zu können, müssen Sie erst den Erzeugemodus wählen, i. d. R. über das Standard-Kontextmenü für das Diagramm.

Der Modus **Knoten erzeugen** kann nur eingeschaltet werden, wenn auf der Eigenschaftenseite **Knoten** die Option **Neue Knoten zulassen** gesetzt ist.

Im Modus **Knoten erzeugen** nimmt der Cursor die Form eines Kreuzchens an. Durch das Ziehen der Maus mit gedrückter linker Maustaste lassen sich in diesem Modus neue Knoten erzeugen. An der Position der Maus erscheint zunächst eine Box, die den aktuellen Anfangs- und Endtermin sowie die Dauer des neuen Knotens anzeigt.

Knoten neu anlegen	
Anfang:	09.09.2007
Ende:	11.09.2007
Dauer:	2 Tage

Wenn Sie bei mehrstufiger Gruppierung in einer kollabierten Gruppe einen Knoten anlegen, erscheint ein kleiner Pfeil an dem Kreuzchen. Dieser Pfeil zeigt an, ob der neu anzulegende Knoten als erster Knoten (Pfeil nach oben) bzw. als letzter Knoten (Pfeil nach unten) in die Gruppe eingefügt wird.

Bei expandierten Gruppen wird der neue Knoten immer als erster Knoten eingefügt, wenn sich der Cursor in einer Gruppentitelzeile befindet.

Bei hierarchischer Gruppierung können Sie den neuen Knoten je nach Richtung des kleinen Pfeils immer vor oder hinter dem Bezugsknoten einfügen.

Wenn die Option **Neuen Knoten bearbeiten** auf der Eigenschaftenseite **Knoten** gewählt ist, öffnet sich das Dialogfeld **Vorgänge bearbeiten**, sobald Sie die Maustaste loslassen. Hier können Sie alle Daten des neu angelegten Knotens bearbeiten.

Wenn Sie nichts anderes vereinbaren, erscheint der neu angelegte Vorgang an der durch die Maus bestimmten Position.

Der **Modus: Knoten erzeugen** lässt sich auch über das Setzen der Eigenschaft **InteractionMode** auf den Wert **VcCreateNode** aktivieren.

Das Ereignis **OnNodeCreate** tritt ein, wenn der Anwender interaktiv einen Knoten erzeugt hat. Das Knotenobjekt wird als Parameter übergeben, sodass eine Datenvalidierung (der Dialog **Vorgänge bearbeiten** ist ggf. vorher aktiviert) vorgenommen werden kann. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten wieder gelöscht.

5.6 Knoten verschieben mit der Maus

Verschieben von Knoten im Diagramm

Das Verschieben von Knoten im Diagramm führt je nach den Einstellungen auf der Eigenschaftenseite **Knoten** zu unterschiedlichen Ergebnissen. Im Folgenden werden die Möglichkeiten beschrieben, die Ihnen bei den Standardeinstellungen zur Verfügung stehen (für Informationen zu den davon abweichenden Einstellungen lesen Sie bitte das Kapitel 4.4: "Eigenschaftenseite Knoten"):

- Markierte Knoten als Ganzes verschieben
- Layer bei gedrückter Shift-Taste als Knoten verschieben

Wenn Sie mit der Maus auf einen Knoten zeigen, wird der Mauszeiger zu einem kleinen Quadrat mit je einem Pfeil rechts und links bzw. vier Pfeilen, falls der Knoten nur durch einen einzigen Layer dargestellt wird. Nun können Sie den gewünschten Layer mit gedrückter linker Maustaste verschieben.



Wenn Sie den kompletten Knoten (mit allen Layern) verschieben möchten, zeigen Sie mit gedrückter Umschalt-Taste auf den gewünschten Knoten. Der Cursor wird zu einem kleinen Quadrat mit vier Pfeilen . Halten Sie die Umschalt-Taste weiter gedrückt und ziehen den Knoten mit Hilfe der Maus an eine andere Position. Es erscheint eine Infobox, die Ihnen das aktuelle Anfangs- und Enddatum des Knotens angibt. Sobald Sie die Maustaste loslassen, bleibt der Knoten an der zuletzt gewählten Position, und die Infobox schließt sich wieder.

Vorgang verschieben	
Anfang:	01.09.2007
Ende:	09.09.2007

Hinweis: Die Umschalt-Taste muss nur gedrückt werden, wenn Sie einen Knoten verschieben möchten, der aus mehreren Layern besteht.

Wenn auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Verändern der Knotenreihenfolge über das Diagramm zulassen** aktiviert ist, können Knoten im Diagramm auch in **vertikaler** Richtung verschoben werden.

Wenn Sie einen Knoten vertikal verschieben, zeigt ein Cursor mit entsprechenden Pfeilen an, wie der Knoten relativ zu den anderen angelagert wird:   .

Verschieben von Knoten in der Tabelle

Auch in der Tabelle lassen sich Knoten verschieben, wenn Sie auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Verändern der Knotenreihenfolge über die Tabelle zulassen** aktiviert haben. Allerdings lassen sich bislang in der Tabelle nur komplette Knoten nur vertikal verschieben.

Wenn Sie einen Knoten vertikal verschieben, zeigt ein Cursor mit entsprechenden Pfeilen an, wie der Knoten relativ zu den anderen angelagert wird:   .

Wenn ein Knoten interaktiv verändert worden ist (Verschieben, Veränderung des Anfangs- und/oder Enddatums des Knotens oder Veränderung eines Wertes im Dialog **Vorgänge bearbeiten**), wird das Ereignis **OnNodeModifyEx** ausgelöst. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten nicht verändert.

5.7 Knoten verschieben und Dauer ändern mit der Tastatur

Standardmäßig sind die Pfeiltasten <links> und <rechts> mit navigierenden Funktionen wie Scrollen im Diagramm oder Verschieben der Markierung von Knoten, Layern oder eines Tabellenfeldes belegt. Sie können den Pfeiltasten aber über die Eigenschaft **VcGantt.ArrowKeyMode** einen Modus zuweisen, in dem Knoten verändert werden können.

In diesem Modus kann der Benutzer einen Knoten verschieben, vergrößern oder verkleinern:

- Verschieben

Der Knoten wird beim Tastendruck direkt verschoben. Die kleinste Schrittweite entspricht der des Verschiebens mit der Maus. Um den Knoten schneller zu verschieben, lässt sich die Schrittweite über die Eigenschaft **VcGantt.ArowKeyStepMultiplier** vergrößern und durch zusätzliches Drücken der <Strg>-Taste aktivieren.

Tastenfunktionen:

Pfeiltaste links/rechts: Knoten verschieben

<Strg> + Pfeiltaste <links/rechts>: Schrittweite modifizieren

- Ändern der Dauer

Die Dauer kann nur für alle **sichtbaren** Layer und nur, wenn nur ein Knoten markiert ist, über die Pfeiltasten verändert werden. Auch hier kann man mit dem oben beschriebenen Multiplikator arbeiten, der durch <Strg> aktiviert wird.

Tastenfunktionen:

<Umschalttaste> + Pfeiltaste <links/rechts>: Dauer ändern

<Umschalttaste> + <Strg> + Pfeiltaste <links/rechts>: Schrittweite modifizieren

Ein Info-Fenster mit Informationen zur Position bleibt noch kurze Zeit nach Beendigung der Interaktion sichtbar, damit die Information gelesen werden kann.

Nähere Informationen zu den entsprechenden API-Eigenschaften finden Sie in der API-Referenz.

Wenn ein Knoten interaktiv verändert worden ist (Verschieben, Veränderung des Anfangs- und/oder Enddatums des Knotens oder Veränderung eines

Wertes im Dialog **Vorgänge bearbeiten**), wird das Ereignis **OnNodeModifyEx** ausgelöst. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten nicht verändert.

5.8 Layer verschieben

Wenn Sie den Cursor auf einem Layer positionieren, erscheint der Verschiebecursor. Nun können Sie den Layer bei gedrückter linker Maustaste mit der Maus horizontal verschieben. Sobald Sie die Maustaste wieder loslassen, wird der Layer an der aktuellen Position positioniert. Während Sie den Layer verschieben, zeigt Ihnen die Infobox **Layer verschieben** die aktuellen Start- und Enddaten des Layers an. Die Dauer bleibt dabei erhalten. Layer können nur innerhalb einer Zeile, aber nicht von einer Zeile in eine andere verschoben werden.

Layer verschieben	
Anfang:	09.09.2007
Ende:	23.09.2007

Wenn Sie einen Symbol-Layer verschieben, sieht die Infobox **Layer verschieben** folgendermaßen aus:

Layer verschieben	
Datum:	09.09.2007

5.9 Anfangs-/Enddatum verändern

Sie können auf dieselbe Weise auch nur das Anfangs- bzw. nur das Enddatum eines Layers verändern, wenn Sie den Cursor am äußeren rechten oder linken Rand des Layers positioniert haben. Dann erscheint eine Infobox, die Ihnen den aktuellen Anfangs- bzw. Endtermin anzeigt. Die Dauer wird dabei verändert.

Anfang verschieben	Ende verschieben
Anfang: 01.09.2007	Ende: 09.09.2007
Dauer: 18 Tage	Dauer: 6 Tage

Wenn ein Knoten interaktiv verändert worden ist (Veränderung des Anfangs- und/oder Enddatum des Knotens oder Veränderung eines Wertes im Dialog **Vorgänge bearbeiten**), wird das Ereignis **OnNodeModify** ausgelöst. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** können Sie die Veränderung unterdrücken.

5.10 Knoten löschen, ausschneiden, kopieren und einfügen

Mit der Entf-Taste können Sie markierte Knoten löschen.

Mit Strg-X können Sie markierte Knoten ausschneiden, mit Strg-C kopieren.

Wenn Sie die Umschalt-Taste gedrückt halten, können Sie mit den Pfeiltasten (oben bzw. unten) mehrere Knoten markieren.

Liegt innerhalb des gewählten Bereichs eine Gruppe im Modus **Knoten in einer Zeile** und **optimiert**, werden alle Knoten dieser Gruppe markiert.

Ist der erste Knoten einer solchen Gruppe markiert und geht man mit gedrückter Umschalt-Taste in eine andere Zeile, werden alle Knoten sowohl in der Ziel- als auch in der Startzeile markiert.

Sie können kodierte oder ausgeschnittene Knoten mit Strg-V oberhalb der Zielzeile (der Zeile, in der ein Knoten markiert ist) einfügen.

Sollen die Knoten unterhalb der Zielzeile eingefügt werden, ist Umschalt-Strg-V zu verwenden.

Die Einfügeposition relativ zum Bezugsknoten wird durch entsprechende Pfeile am Cursorsymbol angezeigt.

Beim Einfügen von Knoten wird die Reihenfolge von gruppierten Knoten beibehalten.

Knoten können nicht in eine leere Gruppe im Modus **Knoten in einer Zeile** und **optimiert** eingefügt werden.

5.11 Knotendaten bearbeiten

Alle Daten eines Knotens können Sie im Dialogfeld **Vorgänge bearbeiten** bearbeiten. Sie erreichen das Dialogfeld durch einen Doppelklick auf einen Knoten oder über den Befehl <Bearbeiten> seines Kontextmenüs.

Um die Daten mehrerer Knoten zu bearbeiten, markieren Sie die gewünschten Knoten und wählen Sie aus dem Kontextmenü eines der markierten Knoten ebenfalls den Befehl **Bearbeiten** um das Dialogfeld **Vorgänge bearbeiten** zu öffnen. Jetzt können Sie nacheinander die Daten aller markierten Knoten bearbeiten

Datenfelder	Werte
ID	2
Name	
Start	06.01.2014
End	11.01.2014
Duration	5
Completion	
Group Level 1	
Group Level 2	
Release Date	
Due Date	

Durch einen Doppelklick auf einen Knoten wird das Ereignis **OnNodeLDbClick** ausgelöst.

Wenn ein Knoten interaktiv verändert worden ist (hier durch die Veränderung eines Wertes im Dialog **Vorgänge bearbeiten**), wird das Ereignis **OnNodeModify** ausgelöst. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Veränderung unterdrückt.

Datenfelder

In dieser Spalte werden alle Datenfelder angezeigt, durch die der markierte Knoten beschrieben wird und die **nicht** im Dialog **Datentabellen verwalten**

als **versteckt** definiert wurden. Welche Datenfelder verfügbar sind, hängt von Ihrer Datendefinition ab.

Werte

Sie können in dieser Spalte alle Werte des markierten Knotens direkt bearbeiten, sofern sie im Dialog **Datentabellen verwalten** als **editierbar** definiert worden sind.

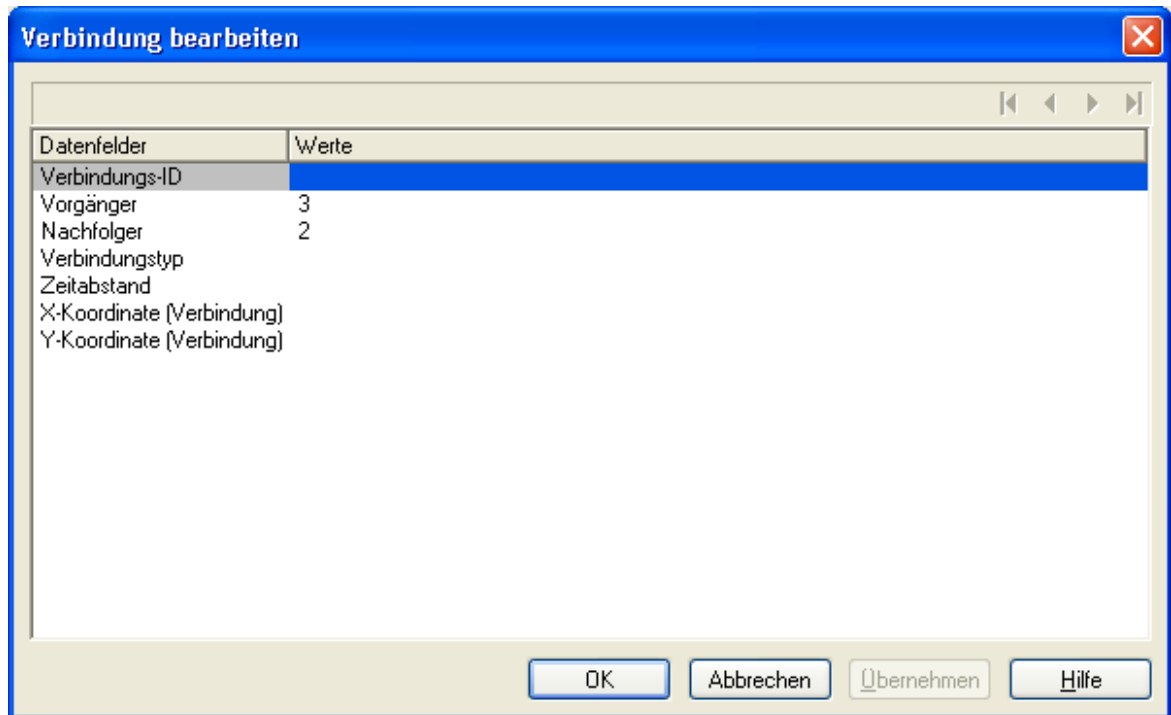
Wenn Sie hier ein Datenfeld vom Typ **Datum/Zeit** bearbeiten, erscheint ein Datumsdialog, in dem Sie das gewünschte Datum anklicken können. Fehler durch die Eingabe eines falschen Datumsformats werden so vermieden.



Das **Datumsausgabeformat** wird auf der Eigenschaftenseite **Allgemeines** festgelegt.

Wenn Sie ein Datenfeld vom Typ **Integer** bearbeiten, erscheint ein Spincontrol, mit dem Sie den gewünschten Wert einstellen können.

5.12 Verbindungen bearbeiten



Dieses Dialogfeld können Sie mit der Methode **VcGantt.EditLink** aufrufen. Hier können Sie die Daten einer markierten Verbindung ansehen und bearbeiten.

An erster Stelle wird die Identifikation (ID) der markierten Verbindung angezeigt.

5.13 Box an Knoten verankern

Sie können eine Box entweder interaktiv (mit der Maus + gedrückter Umschalt-Taste oder über das Kontextmenü) oder über die API mit einem Knoten verankern.

- **Verankern mit der Maus:** Zeigen Sie mit der Maus auf die gewünschte Box und drücken Sie die Umschalt-Taste. Ein kleiner Anker erscheint. Nun ziehen Sie mit gedrückter linker Maustaste eine Linie von der Box bis zu dem gewünschten Knoten und lassen die Maustaste wieder los. Die Box ist nun mit dem Knoten verankert und bei aktivierter Option **Ankerlinie sichtbar** im Dialog **Boxen verwalten** erscheint eine Verbindungslinie. Um die Verankerung wieder zu lösen, verfahren sie genauso.
- **Verankern über das Kontextmenü:** Markieren Sie den Knoten, mit dem die Box verankert werden soll und wählen dann aus dem Kontextmenü der Box den Befehl **Box am markierten Knoten verankern**. Sollte das Kontextmenü nicht erscheinen, muss auf der Eigenschaftenseite **Allgemeines** die Option **Kontextmenü für Boxen anzeigen** aktiviert werden.



Zum Lösen der Box von dem Knoten wählen Sie aus dem Kontextmenü den entsprechenden Befehl.

Wenn Sie die Box mit einem anderen Knoten verankern möchten, verfahren Sie genauso wie oben beschrieben

- **Verankern über API:** Bitte lesen Sie dazu in der API-Referenz die Beschreibung der Eigenschaft **AnchoringInteractionsAllowed** sowie der Methode **AnchorToNode** des Objektes **VcBox**

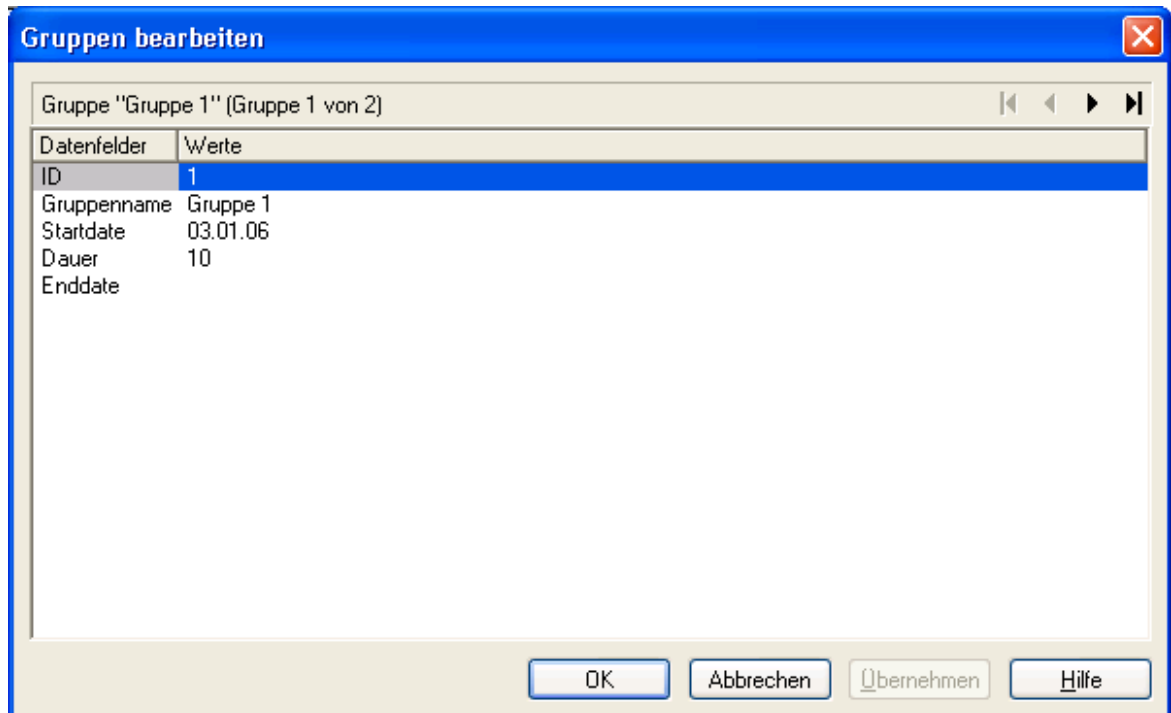
Eine verankerte Box kann weiterhin interaktiv verschoben werden (Voraussetzung: die Option **Verschiebbar** ist aktiviert).

Wird ein Knoten mit einer verankerten Box verschoben, so wird die Box entsprechend mitverschoben. Wird der Knoten kollabiert, wird auch die Box kollabiert d.h. sie ist nicht mehr sichtbar. Sobald der Knoten expandiert wird, ist auch die Box wieder sichtbar.

Bei interaktiver Ver/Entankerung bleibt die Position der Box auf dem Bildschirm erhalten - die zugrunde liegenden Offset-Werte werden entsprechend der Referenzpunkte (Origin, ReferencePoint) umgerechnet. Wenn sich also eine Textbox mit einem bestimmten Offset z.B. auf das Diagramm oben links (Origin) bezieht und dann mit einem Knoten verankert wird, wird automatisch ein Offset zum Knoten oben links berechnet und dieser Offset führt dazu, dass die Position auf dem Bildschirm beibehalten wird. Wird die Box vom Knoten gelöst, erfolgt eine Rückberechnung.

Bei Verwendung der API-Eigenschaft **AnchorToNode** wird genauso verfahren, nicht aber bei der Eigenschaft **NodeID**.

5.14 Gruppendaten bearbeiten



Dieses Dialogfeld erreichen Sie entweder über das Kontextmenü für Gruppen oder durch einen Doppelklick auf einen Gruppenbalken (der nur angezeigt wird, wenn im Dialog **Gruppierung** die Option **Gruppenknoten sichtbar** aktiviert wurde).

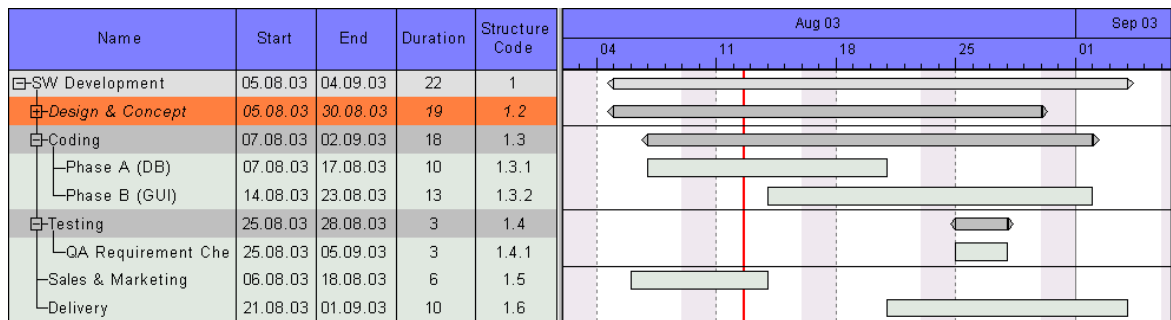
Hier können die Daten einer Gruppe, bzw. wenn mehrere Gruppen markiert sind, sukzessive die Daten aller markierter Gruppen bearbeitet werden.

Oberhalb der Tabelle wird angezeigt um die wievielte Gruppe der markierten Gruppen es sich handelt.

Mit Hilfe der Pfeil-Schaltflächen können Sie zwischen den Gruppen navigieren.

5.15 Gruppen kollabieren bzw. expandieren

Wenn bei gruppierter Darstellung im Dialog **Gruppierung** das Kontrollkästchen **Änderungen erlaubt** aktiviert ist, können Sie durch einen Doppelklick auf die Gruppenüberschrift oder durch Anklicken des **Plus**- oder **Minus**-Symbols neben der Gruppenüberschrift eine kollabierte Gruppe expandieren bzw. eine expandierte Gruppe kollabieren.



Das Ereignis **OnGroupModify** tritt ein, wenn der Anwender eine Gruppe interaktiv verändert. Das getroffene Group-Objekt, die Art der Veränderung und der Rückgabestatus werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Veränderung unterdrückt.

5.16 Gruppen verschieben

Gruppen lassen sich vertikal sowohl im Tabellen- als auch im Diagrammteil (dort über den Summenbalken) verschieben, wenn im Dialog **Gruppierung** die Optionen **Vertikal Verschieben über den Tabellenbereich** und/oder **Vertikal Verschieben über den Diagrammbereich** ausgewählt wurden. Beim Verschieben wird durch einen entsprechenden Cursor angezeigt, wo die Gruppe eingefügt wird .

Hinweis. Eine Gruppe lässt sich nur innerhalb einer Parentgruppe verschieben.

5.17 Bearbeiten von Feldinhalten in der Tabelle

Zum Bearbeiten eines Feldinhaltes klicken Sie das gewünschte Feld an und geben entweder einen neuen Inhalt ein oder ändernd den bestehenden ab.

Weitere Möglichkeiten, den Feldinhalt zu bearbeiten, bieten sich Ihnen, wenn Sie die Option **Erweitertes Editieren in Tabelle** auf der Eigenschaftenseite **Allgemeines** aktiviert haben.

Dann erscheint bei Datumsfeldern ein Drop-Down-Pfeil, den Sie anklicken können, um das Datum-Steuerelement anzuzeigen. Für weitere Informationen zum Gebrauch des Datum-Steuerelementes s. Kapitel 4.40 Dialogfeld "Stichtaglinien festlegen".

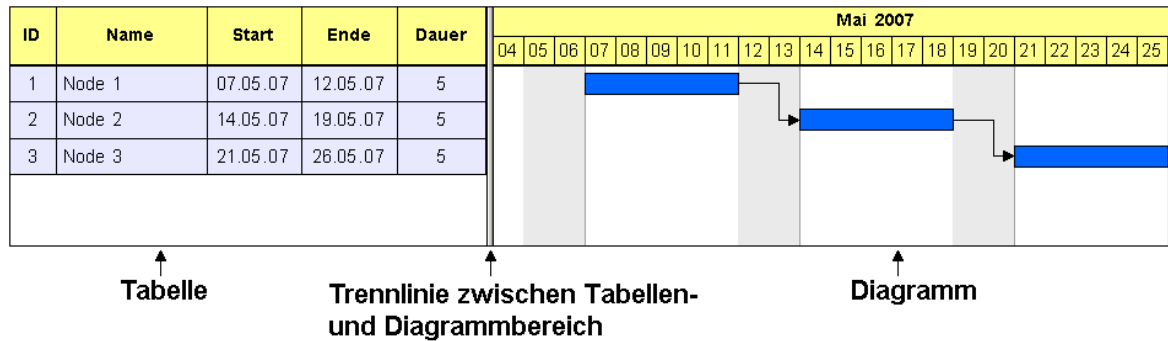
Bei numerischen Feldern finden sich Pfeil-Schaltflächen, mit deren Hilfe Sie den Wert jeweils erhöhen bzw. verringern können.

Eine ausführliche Beschreibung der erweiterten Editiermöglichkeiten in der Tabelle finden Sie in Kapitel 4.2 "Eigenschaftenseite Allgemeines".

Hinweis: Mit Hilfe der Taste ESC verlassen Sie die editierten Felder, ohne die Änderungen zu übernehmen.

5.18 Breitenverhältnis Tabelle/ Diagramm bearbeiten

Die Tabellentitel und das Diagramm werden durch eine vertikale Trennlinie (Splitterbar) voneinander getrennt.



Wenn Sie im Bereich der Tabellenzeilen oder des Diagramms mit dem Mauszeiger darüber fahren, wird der Zeiger zu einer vertikalen Doppellinie mit je einem Pfeil von links und rechts.

Name	Dauer	04	05	06
Node 1	5			
Node 2	5			

Nun können Sie das Breitenverhältnis von Tabellen und Diagramm durch Ziehen mit der Maus verändern. (Der Tabellenbereich kann maximal so breit werden, wie im Dialog **Tabelle bearbeiten** als Summe der einzelnen Tabellenspaltenbreiten definiert wurde.)

5.19 Breite der Tabellenspalte bearbeiten

Die Breite einer Tabellenspalte können Sie interaktiv verändern, indem Sie im Bereich der Tabellenüberschrift mit dem Cursor die Trennlinie zwischen zwei Spalten verschieben.

ID	Name	↔	Start	Ende
----	------	---	-------	------

Die Breite einer Tabellenspalte kann nur im Bereich der Tabellentitel verändert werden.

Das Ereignis **VcTableWidthChanging** tritt ein, wenn die Tabellenbreite interaktiv worden ist. Die Tabelle und das neue Diagramm-Breitenverhältnis werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

Das Ereignis **VcTableColumnWidthChanging** tritt ein, wenn der Anwender die Breite einer Tabellenspalte interaktiv verändert hat. Die Tabelle, der Index der geänderten Spalte und die neue Spaltenbreite (in 1/100 mm) werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

Sie können die Spaltenbreite auch automatisch berechnen lassen. Die Voraussetzung dafür ist, dass auf der Eigenschaftenseite **Allgemeines** die Option **Tabellenspalten optimierbar** aktiviert ist. Dann kann der Anwender durch einen Doppelklick auf die Trennlinie zwischen zwei Spalten erreichen, dass die Breite der linken Spalte automatisch an die Länge der darin enthaltenen Texte angepasst wird. Dabei wird das Ereignis **OnOptimizeTableColumnWidth** ausgelöst. Wenn die Optimierung erfolgt ist, wird das Ereignis **VcTableColumnWidthChanging** ausgelöst.

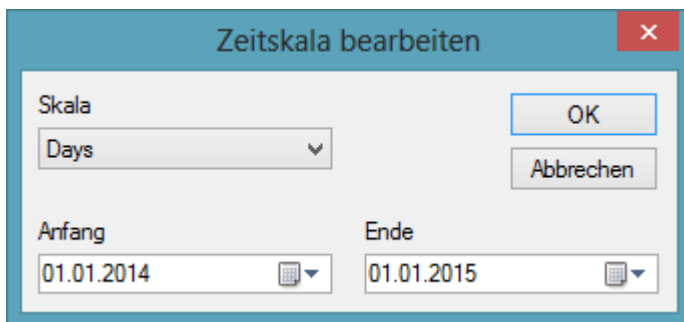
5.20 Einfügen von neuen Tabellenzeilen

Wenn auf der Eigenschaftenseite **Allgemeines** die Option **Erweitertes Editieren in Tabelle** aktiviert ist, lässt sich mithilfe der Taste EINFÜG oberhalb der jeweils markierten Zeile eine neue Tabellenzeile einfügen. Wenn nichts markiert ist, wird die neue Zeile am Ende der Tabelle eingefügt.

5.21 Zeitskala bearbeiten

Im Dialogfeld **Zeitskala bearbeiten** können Sie eine vordefinierte Zeitskala (Minuten, Stunden, Tage, Wochen, Monate) auswählen sowie deren Anfang und Ende festlegen. Wenn mehrere Zeitskalenabschnitte definiert sind, darf hier der Anfang der Zeitskala nicht in den zweiten Zeitskalenabschnitt gelegt werden.

Sie erreichen das Dialogfeld durch einen Doppelklick auf die Zeitskala oder über ihr Kontextmenü.



Durch einen Doppelklick auf die Zeitskala wird das Ereignis **OnTimeScaleLDbClick** ausgelöst. Das getroffene TimeScale-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Das integrierte **Zeitskala bearbeiten**-Dialogfeld kann durch Setzen des Rückgabestatus auf **vcRetStatFalse** unterdrückt werden.

Das Dialogfeld "Zeitskala bearbeiten"

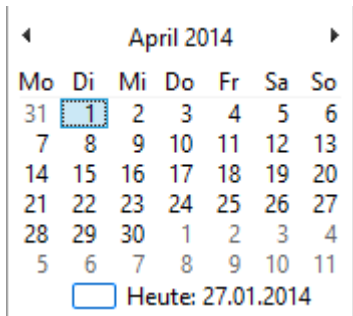
Skala

Hier können Sie die gewünschte Zeitskala auswählen. Zur Auswahl stehen Minuten, Stunden, Tage, Wochen und Monate.

Anfang

In diesem Feld legen Sie den Anfangstermin der Zeitskala fest.

Wenn Sie auf die Pfeil-Schaltfläche klicken, erscheint ein kleiner Datumsdialog, in dem Sie das gewünschte Datum anklicken können. Fehler durch die Eingabe eines falschen Datumsformats werden so vermieden.



Das Datumsausgabeformat wird auf der Eigenschaftenseite **Allgemeines** festgelegt.

Ende

In diesem Feld legen Sie den Endtermin der Zeitskala fest.

Wenn Sie auf die Pfeil-Schaltfläche klicken, erscheint ein kleiner Datumsdialog, in dem Sie das gewünschte Datum anklicken können. Fehler durch die Eingabe eines falschen Datumsformats werden so vermieden.

Das Datumsausgabeformat wird auf der Eigenschaftenseite **Allgemeines** festgelegt.

5.22 Skalierung und Grenzen von Zeitskalenabschnitten bearbeiten

Skalierung von Zeitskalenabschnitten bearbeiten



Sie können einen Zeitskalenabschnitt interaktiv skalieren, indem Sie den Cursor auf diesem Zeitskalenabschnitt positionieren und dann bei gedrückter linker Maustaste die Maus nach links oder rechts ziehen. Dabei ändert der Cursor seine Form zu einer vertikalen Linie mit je einem Pfeil nach links und rechts. Ziehen Sie den Cursor nun nach links, wird die Einheitenbreite des Zeitskalenabschnitts verkleinert; ziehen Sie den Cursor nach rechts, wird sie vergrößert. Während des Ziehens mit der Maus zeigt Ihnen eine Infobox, um welchen Prozentfaktor die Einheitenbreite des Zeitskalenabschnitts (Sektion) verändert wird.

Hinweis: Wenn der Cursor im Bereich der Zeitskala in der Nähe des Sash steht und Sie den Cursor dann nach rechts bzw. links ziehen, wird die Zeitskala stark gedehnt bzw. gestaucht. Diese Reskalierung können Sie wieder rückgängig machen, indem Sie mit dem Cursor in der Zeitskala in die entgegengesetzte Richtung ziehen.

Das Ereignis **OnTimeScaleSectionRescale** tritt ein, wenn der Anwender einen Abschnitt der Zeitskala interaktiv skaliert. Die Zeitskala, die Nummer des Zeitskalenabschnitts und die neue Grundeinheitsbreite (**BasicUnitWidth**) werden als Parameter übergeben, sodass Sie beispielsweise die Zulässigkeit der Skalierung prüfen können. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

Grenzen von Zeitskalenabschnitten bearbeiten



Die Grenze zwischen zwei Zeitskalenabschnitten können Sie interaktiv verändern, indem Sie mit der Maus die Linie zwischen diesen beiden Zeitskalenabschnitten verschieben. Dabei ändert der Cursor seine Form zu einer vertikalen Doppellinie mit je einem Pfeil nach links und rechts.

Das Ereignis **OnTimeScaleSectionStartModify** tritt ein, wenn der Anwender den Anfang eines Zeitskalenabschnitts interaktiv verändert. Die Zeitskala, die Nummer des Zeitskalenabschnitts und das neue Anfangsdatum

werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

5.23 Stichtaglinie verschieben

Das Stichtagdatum können Sie interaktiv verändern, indem Sie mit der Maus die Stichtaglinie verschieben. Die Voraussetzung dafür ist, dass im Dialog **Stichtaglinien festlegen** das Kontrollkästchen **Frei verschiebbar** für die betreffende Stichtaglinie aktiviert ist.

Stichtaglinien können auch per API gesetzt werden.

Das Ereignis **OnDateLineModify** tritt ein, wenn eine Stichtaglinie durch den Anwender interaktiv verschoben wurde. Die veränderte Stichtaglinie wird als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf **vcRetStat-False** wird die Änderung rückgängig gemacht.

5.24 Seite einrichten

Alle Einstellungen zum Seitenlayout können Sie im Dialog "Seite einrichten" vornehmen. Sie gelangen in diesen Dialog entweder über den entsprechenden Befehl im Diagramm-Kontextmenü über aus der Druckvorschau durch Klick auf die gleichnamige Schaltfläche.

Seite einrichten

Skalierung

Modus: Anpassen an Seitenanzahl

Zoomfaktor: 100,0 % Aktuell: 9,97

Maximale Breite: 1 Seite(n)

Maximale Höhe: 1 Seite(n)

☐ Titel/Tabelle/Zeitskala/Legende wiederholen

☒ Tabelle anzeigen

☐ Aussehen von Bildschirmansicht übernehmen

Tabellenspalten (1-5;7):

☒ Diagramm anzeigen

☐ Zeitskalenstart: 01.01.2014

☐ Zeitskalenende: 01.01.2015

☐ Zeitskala an Breite der Seiten anpassen

Seitenaufteilung

☒ Seiten mit Leerraum auffüllen

☒ Rahmen außen

Ausrichtung: Mittig

☐ Zuschnittmarken

☐ Faltmarkierungen (DIN 824): Form A

Fußzeile

☐ Seitennummerierung: Zeile.Spalte

☐ Text:

☐ Druckdatum

Seitenränder

Links: 1,5 cm Oben: 1,0 cm

Rechts: 1,0 cm Unten: 1,0 cm

OK Abbrechen

Modus

Durch Auswahl einer Skalierungsart aus der Drop-Down-Liste und der entsprechenden Werte bei **Zoomfaktor** bzw. **Maximale Breite/Höhe** bestimmen Sie den Maßstab der Darstellung bei der Ausgabe. Nach Klick auf **Übernehmen** werden unter **Aktuell** die Werte angezeigt, die sich aus Ihren Einstellungen ergeben.

Zoomfaktor

Ein Skalierungsfaktor von 100 % entspricht der Originalgröße, ein kleinerer Wert bewirkt eine entsprechende Verkleinerung, ein größerer Wert eine Vergrößerung.

Anpassen an Seitenzahl

Durch Auswahl dieser Option können Sie die Anzahl der Seiten in Höhe und Breite vorgeben, auf die das Diagramm bei der Ausgabe maximal aufgeteilt werden soll (**Maximale Breite, Maximale Höhe**). Die Diagramme werden so groß wie möglich, aber ohne Verzerrungen dargestellt.

Zoomen mit horizontaler Anpassung

Wählen Sie einen Zoomfaktor sowie eine feste Anzahl an Druckseiten in der Breite, die beim Druck durch Stauchen oder Strecken der Zeitskala eingehalten wird.

Titel/Tabelle/Zeitskala/Legende wiederholen

Aktivieren Sie dieses Kontrollkästchen, damit bei der Ausgabe des Diagramms auf mehreren Seiten auf jeder Seite Titel, Tabelle, Zeitskala und Legende ausgegeben werden.

Tabelle anzeigen

Bestimmen sie, ob die Tabelle gedruckt wird oder nicht. Wenn die Option nicht ausgewählt ist, wird die Tabelle nicht gedruckt.

Aussehen von Bildschirmansicht übernehmen

Hier können Sie festlegen, ob bei der Druckvorschau und beim Druck die aktuell auf dem Bildschirm sichtbare Breite der Tabelle übernommen werden soll.

Diese Option kann auch über die Eigenschaft **VcPrinter.TableWidth-AdoptionFromViewOnScreen** festgelegt werden.

Tabellenspalten

Hier können Sie festlegen, welche Tabellenspalten gedruckt werden sollen. Geben Sie dazu die gewünschten Spalten einzeln oder in Bereichen durch Komma oder Semikolon getrennt an. Beispiel: "1;5-7;3" selektiert die Spalten 1, 3 und 5 bis 7.

Diagramm anzeigen

Bestimmen sie, ob das Diagramm (Zeitskala und Balken) mit ausgedruckt werden soll oder nicht.

Zeitskalenstart

Mithilfe dieser Option kann der Starttermin des zu druckenden Zeitraums festgelegt werden. Dabei ist es nur möglich, den Zeitraum gegenüber dem am Bildschirm gezeigten einzuschränken, d.h. nur ein Startdatum, das nach dem über die Eigenschaft **VcGantt.TimeScaleStart** gesetzten liegt, führt zu einer Änderung des Ausdrucks.

Diese Option kann auch über die Eigenschaft **VcPrinter.TimeColumnStart-Date** festgelegt werden.

Zeitskalenende

Mithilfe dieser Option kann der Endtermin des zu druckenden Zeitraums festgelegt werden. Dabei ist es nur möglich, den Zeitraum gegenüber dem am Bildschirm gezeigten einzuschränken, d.h. nur ein Enddatum, das vor dem über die Eigenschaft **VcGantt.TimeScaleEnd** gesetzten liegt, führt zu einer Änderung des Ausdrucks.

Diese Option kann auch über die Eigenschaft **VcPrinter.TimeColumnEnd-Date** festgelegt werden.

Zeitskala an Breite der Seiten anpassen

Diese Option führt dazu, dass der Platz auf den Druckseiten besser ausgenutzt wird:

- Bei einer Skalierung über eine feste Anzahl Seiten: Der Zoomfaktor wird so berechnet, dass die eingestellte Anzahl Seiten in der Höhe voll

bedruckt wird. Gleichzeitig wird die Zeiskala so gestaucht oder gestreckt, dass die eingestellte Anzahl Seiten in der Breite voll ausgenutzt wird.

- Bei einer Skalierung über einen Zoomfaktor: Die Zeitskala wird so gestaucht oder gestreckt, dass die eingestellte Anzahl Seiten in der Breite voll ausgenutzt wird.

Seiten mit Leerraum auffüllen

Mithilfe dieser Option können Sie festlegen, ob zwischen dem Diagramm und den Boxen für Titel und Legende so viel Platz gelassen wird, dass die Boxen auf jeder Druckseite immer in voller Breite gedruckt werden können und fest am Blattrand positioniert sind. Wenn diese Option nicht ausgewählt ist, werden die Boxen ohne Zwischenraum am Diagramm gedruckt und können dann je nach Diagramm auf den verschiedenen Druckseiten in der Breite variieren.

Rahmen außen

Aktivieren Sie dieses Kontrollkästchen, damit ein Rahmen um das Diagramm herum ausgegeben wird. Wenn die Option **Titel/Tabelle/Zeitskala wiederholen** ausgewählt ist, erhält jede Seite einen Rahmen, andernfalls wird ein Rahmen um das gesamte Diagramm gezogen.

Ausrichtung

Bestimmen Sie die Ausrichtung des Diagramms auf dem Blatt.

Zuschnittmarken

Wurde dieses Kontrollfeld aktiviert, wird das Diagramm mit Zuschnittmarken versehen, die das Zusammenkleben der ausgedruckten Einzelseiten zu einer Gesamtgrafik erleichtern.

Faltmarkierungen (DIN 824)

In der DIN Norm 824 ist für Bauzeichnungen eine ganz bestimmte Art der Faltung vorgeschrieben, mit der man die Zeichnung auf DIN A4-Größe zusammenfalten kann. Die Ausgabe von entsprechenden Faltmarkierungen auf Ihrem Diagramm erleichtert Ihnen die Faltung. Folgende Formate stehen zur Verfügung:

- **Form A:** mit Heftrand auf der linken Seite, damit die gefaltete Zeichnung gelocht und ohne Heftstreifen in einem Ordner abgeheftet werden kann.

- **Form B:** insgesamt etwas schmaler, damit ein Heftstreifen angebracht werden kann, der dann gemeinsam mit der Zeichnung die Breite von DIN A4 erreicht.
- **Form C:** die gefaltete Zeichnung wird nicht gelocht, sondern in eine Sichthülle gelegt

Die vorliegenden Faltmarkierungen können für jedwedes Zielformat ausgegeben werden, während die DIN 824 explizit nur die Formate DIN A0 bis A3 kennt.

Seitennummerierung

Ist dieses Kontrollkästchen aktiviert, wird auf jeder Seite unten links die Seitennummer ausgegeben. Folgende Möglichkeiten stehen dabei zur Auswahl:

- **Zeile.Spalte:** Sinnvoll, wenn das Diagramm sich auf mehr als eine Seite in der Länge als auch in der Breite erstreckt. Vor dem Punkt wird die Position der Seite in der vertikalen, dann die in der horizontalen Reihenfolge ausgegeben.
- **Spalte.Zeile:** Sinnvoll, wenn das Diagramm sich auf mehr als eine Seite in der Länge als auch in der Breite erstreckt. Vor dem Punkt wird die Position der Seite in der horizontalen, dann die in der vertikalen Reihenfolge ausgegeben.
- **Seite/Anzahl:** Zuerst erscheint die aktuelle Seitenzahl, danach die Anzahl der Gesamtseiten: 1/6, 2/6 etc.

Text

Aktivieren Sie dieses Kontrollkästchen, um jede Seite unten links mit einem beliebigen Text zu versehen. Dieser Text wird ggf. rechts von der Seitennummer ausgegeben.

Für die Seitennummerierung können Sie in die Zeile **Zusatztext** folgende Platzhalter eingeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{PAGE}	= fortlaufende Seitennummer
{NUMPAGES}	= Gesamtanzahl der Seiten
{ROW}	= Zeilenposition des Ausschnitts im Gesamtdiagramm
{COLUMN}	= Spaltenposition des Ausschnitts im Gesamtchart

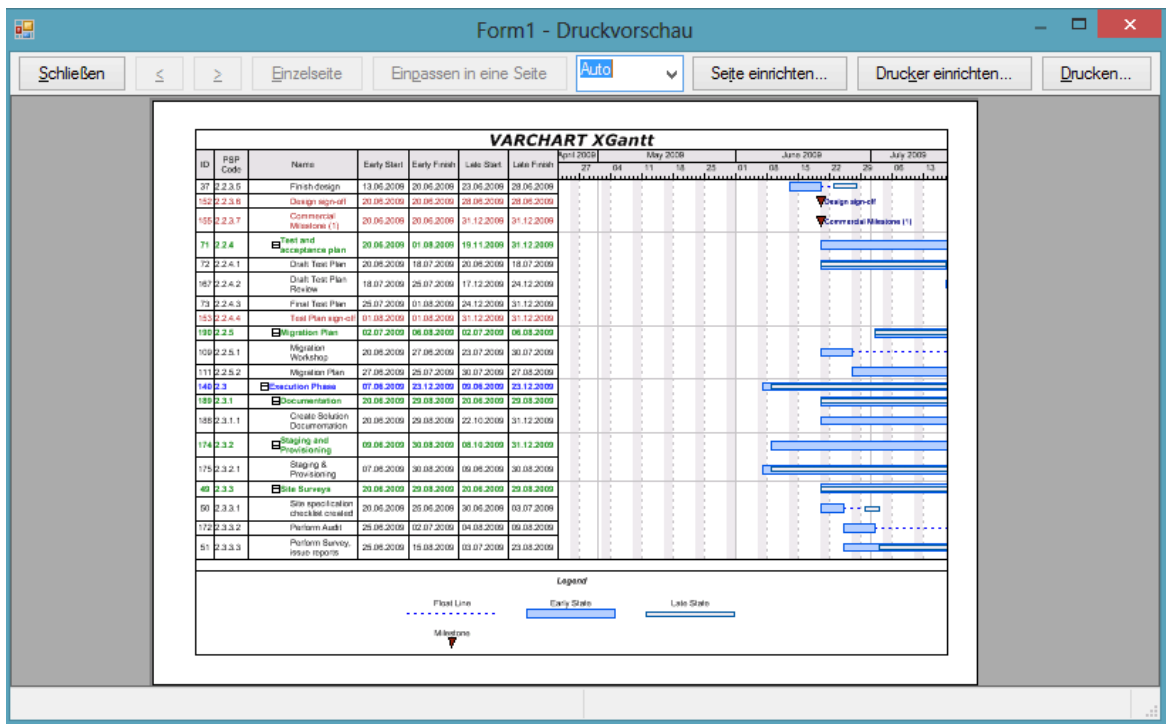
Druckdatum

Ist dieses Kontrollkästchen aktiviert, wird auf jeder Seite unten links das Druckdatum ausgegeben. Das Druckdatum wird ggf. rechts von der Seitennummer und dem Zusatztext ausgegeben.

Seitenränder

Über die Felder **Oben**, **Unten**, **Links** und **Rechts** legen Sie den Raum zwischen Papierrand und dem Diagramm in cm fest.

5.25 Druckvorschau



Vor dem Drucken können Sie das Diagramm in der Druckvorschau prüfen. Es wird so auf dem Bildschirm dargestellt, wie es im Dialogfeld **Seite einrichten** festgelegt ist und wie es gedruckt wird.

Sie können hier einzelne Seiten oder die Gesamtübersicht über alle Seiten Ihrer Darstellung ansehen oder einen Ausschnitt Ihres Diagramms interaktiv vergrößern und anschliessend drucken.

Die Statuszeile informiert über die Gesamtseitenzahl und die horizontale und vertikale Aufteilung der Seiten. In der Ansicht **Einzelseite** wird zusätzlich noch die aktuelle Seitenzahl angezeigt.

Schließen

Sie verlassen die Druckvorschau und gelangen zurück in die Darstellung.



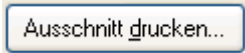
Nur aktiv, wenn die Schaltfläche **Einzel** gedrückt wurde. Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten einzeln ansehen. Mit Hilfe dieser Schaltfläche gelangen Sie zur vorangehenden Seite. Sie bewegen sich über die Seiten von rechts nach links in aufsteigenden Zeilen.

>

Nur aktiv, wenn die Schaltfläche **Einzel** gedrückt wurde. Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten einzeln ansehen. Mit Hilfe dieser Schaltfläche gelangen Sie zur nächsten Seite. Sie bewegen sich über die Seiten von links nach rechts in absteigenden Zeilen.

Einzelseite/Übersicht

Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten entweder einzeln oder in der Übersicht ansehen. In der **Übersicht** sehen Sie alle Seiten - je nach Seitenanzahl mehr oder weniger stark verkleinert, im Darstellungsmodus **Einzelseite**, wird zunächst die erste Seite des Diagramms einzeln und vergrößert dargestellt. Mit  oder  können Sie dann durch die Seiten blättern. Mit einem Doppelklick auf eine Seite wechseln Sie bequem zwischen den beiden Darstellungsarten **Einzelseite** und **Übersicht**.

Ein Ausschnitt Ihres Diagramms lässt sich im Darstellungsmodus **Einzelseite** interaktiv vergrößern, indem Sie bei gedrückter linker Maustaste ein Rechteck um den zu vergrößernden Bildausschnitt ziehen. Sobald Sie die linke Maustaste loslassen, wird der eingerahmte Bildausschnitt entsprechend vergrößert und statt der Schaltfläche **Drucken** erscheint die Schaltfläche  über die Sie den Bildausschnitt dann in der aktuellen Vergrößerung drucken können. Beachten Sie bitte, dass der in der Druckvorschau interaktiv gewählte Vergrößerungsfaktor nicht den Skalierungsfaktor im Dialogfeld **Seite einrichten** verändert.

Einpassen in eine Seite

Mit dieser Schaltfläche lässt sich ein mehrseitiges Diagramm auf eine Seite verkleinern. Auch hier können Teile des Diagramms, wie unter **Einzelseite/Übersicht** beschrieben, interaktiv vergrößert und anschliessend gedruckt werden.

Zoomfaktor

Wählen Sie einen Zoomfaktor aus der Liste oder definieren einen individuellen, um die Darstellungsgröße ihres Diagramms für die Druckvorschau zu verändern. Dies ist nur im Modus "Einzelseite" möglich. Der Zoomfaktor lässt sich auch durch Drehen des Mausekkrades bei gedrückter <STRG>-Taste verändern. Er hat keinen Einfluss auf den späteren Druck. Je

nach gewählter Größe werden vertikale und/oder horizontale Bildlaufleisten angezeigt. Auch das Mausrad kann zum Bewegen des Bildes verwendet werden (ohne Umschalttaste vertikal, mit Umschalttaste horizontal).

Der Zoomfaktor **Auto** ist voreingestellt und verkleinert bzw. vergrößert das Blatt immer so, dass es bildschirmfüllend dargestellt wird.

Seite einrichten

Sobald Sie auf diese Schaltfläche klicken, gelangen Sie in das Dialogfeld **Seite einrichten** und können dort Änderungen am Seitenlayout vornehmen.

Drucker einrichten

*Nur sichtbar, wenn auf der Eigenschaftenseite **Allgemeines** die Option **PrintDlgEx-Dialog verwenden** nicht gewählt wurde.*

Sobald Sie auf diese Schaltfläche klicken, gelangen Sie in das Windows-Dialogfeld **Drucker einrichten** und können dort Änderungen an den Drucker-einstellungen vornehmen.

Drucken/Ausschnitt drucken

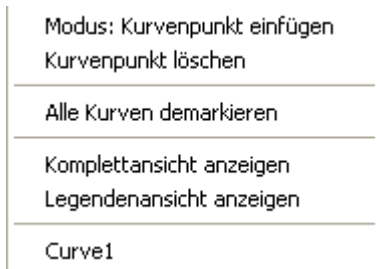
Über diese Schaltfläche gelangen Sie in den Windows-Dialog **Drucken** und können von dort aus den Druckvorgang einleiten.

Wenn Sie in der Druckvorschau einen Ausschnitt interaktiv gezoomt haben, ändert die Schaltfläche ihre Beschriftung zu **Ausschnitt drucken**. Wenn Sie sie anklicken, ist im Windows-Dialog **Drucken** die Option **Markierung** bereits ausgewählt. Durch Klick auf **OK** wird der am Bildschirm dargestellte Ausschnitt gedruckt.

Beachten Sie bitte, dass der in der Druckvorschau interaktiv gewählte Vergrößerungsfaktor nicht den Skalierungsfaktor im Dialogfeld **Seite einrichten** verändert.

5.26 Kontextmenü für die Kurve

Wenn Sie mit der rechten Maustaste in einen leeren Histogrammbereich oder auf eine Kurve klicken, erscheint das folgende Kontextmenü:



Das Ereignis **OnHistogramRClick** oder **OnCurveRClick** wird ausgelöst, wenn der Anwender mit der rechten Maustaste auf das Histogramm oder auf die Kurve klickt. Das getroffene Histogramm- oder Kurven-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Das integrierte Kontextmenü kann an der gegebenen Position durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt und ein eigenes Kontextmenü anzeigen werden.

Modus: Kurvenpunkt einfügen

In diesem Modus können Sie mit jedem Linksklick auf die Verfügbarkeitskurve einen neuen Stützpunkt für diese definieren.

Kurvenpunkt löschen

Um einen Stützpunkt zu löschen, klicken Sie diesen mit der rechten Maustaste an und wählen Sie im Kontextmenü den Befehl **Kurvenpunkt löschen**.

Alle Kurven demarkieren

Alle Kurven werden demarkiert.

Komplettansicht anzeigen

Über diesen Menüpunkt können Sie die Komplettansicht ein- oder ausschalten. Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm und/oder Histogramm angezeigt wird. Ein Rahmen zeigt an, welchen Ausschnitt das Hauptfenster gerade anzeigt.

Legendenansicht anzeigen

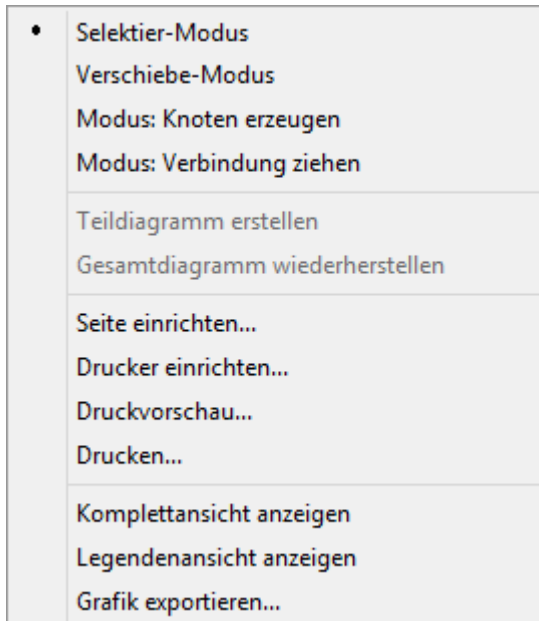
Über diesen Menüpunkt können Sie die Legendenansicht ein- oder ausschalten. Die Legendenansicht ist ein zusätzliches Fenster, in dem die Legende angezeigt wird.

Kurven

Falls vorhanden, werden die API-Kurven am Ende des Kontextmenüs angezeigt. Über den entsprechenden Menüeintrag können sie markiert werden.

5.27 Kontextmenü für das Diagramm

Wenn Sie die rechte Maustaste drücken, wenn der Cursor im Diagrammbereich (jedoch nicht auf einem Knoten) steht, öffnet sich das folgende Kontextmenü:



Das Ereignis **OnDiagramRClick** tritt ein, wenn der Anwender mit der rechten Maustaste im Diagrammbereich klickt und dabei keinen Knoten trifft. Die Position (x,y-Koordinaten) wird als Parameter übergeben. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt werden.

Selektier-Modus

Der Selektier-Modus ist der Standardmodus, in dem alle Interaktionen außer dem Erzeugen von Knoten und Verbindungen möglich sind.

Verschiebe-Modus

Im Verschiebemode lässt sich mithilfe eines Cursors in Handform der darunter befindliche Bildschirmausschnitt verschieben.

Dieser Modus muss auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

Modus: Knoten erzeugen

Dieser Modus kann nur eingeschaltet werden, wenn auf der Eigenschaftenseite **Knoten** die Option **Neue Knoten zulassen** gesetzt ist.

Der Cursor nimmt die Form eines Kreuzchens an. Durch das Ziehen der Maus mit gedrückter linker Maustaste lassen sich in diesem Modus neue Knoten erzeugen. An der Position der Maus erscheint zunächst eine Box, die den aktuellen Anfangs- und Endtermin sowie die Dauer des neuen Knotens anzeigt.

Knoten neu anlegen	
Anfang:	09.09.2007
Ende:	11.09.2007
Dauer:	2 Tage

Wenn Sie im Modus **Knoten erzeugen** in einer kollabierten Gruppe einen Knoten anlegen, erscheint ein kleiner Pfeil an dem Kreuzchen. Dieser Pfeil zeigt an, ob der neu anzulegende Knoten als erster Knoten (Pfeil nach oben) bzw. als letzter Knoten (Pfeil nach unten) in die Gruppe eingefügt wird.

Wenn die Option **Neuen Knoten bearbeiten** auf der Eigenschaftenseite **Knoten** gewählt ist, öffnet sich das Dialogfeld **Vorgänge bearbeiten**, sobald Sie die Maustaste loslassen. Hier können Sie alle Daten des neu angelegten Knotens bearbeiten.

Wenn Sie nichts anderes vereinbaren, erscheint der neu angelegte Vorgang an der durch die Maus bestimmten Position.

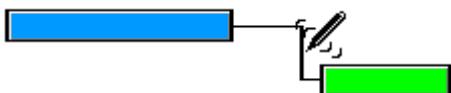
Der **Modus: Knoten erzeugen** lässt sich auch über das Setzen der Eigenschaft **InteractionMode** auf den Wert **VcCreateNode** aktivieren.

Das Ereignis **OnNodeCreate** tritt ein, wenn der Anwender interaktiv einen Knoten erzeugt hat. Das Knotenobjekt wird als Parameter übergeben, sodass eine Datenvalidierung (der Dialog **Vorgänge bearbeiten** ist ggf. vorher aktiviert) vorgenommen werden kann. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten wieder gelöscht.

Modus: Verbindung ziehen

Dieser Modus kann nur eingeschaltet werden, wenn auf der Eigenschaftenseite **Verbindungen** die Option **Verbindungen anzeigen** gesetzt ist.

Der Cursor nimmt die Form eines Bleistiftes an. Verbinden Sie nun mit der Maus zwei Knoten, so wird eine Ende-Anfang-Verbindung zwischen ihnen erstellt.



Das Ereignis **OnLinkCreate** tritt ein, wenn der Anwender interaktiv eine Verbindung zwischen zwei Knoten erzeugt hat. Das neu erzeugte Link-Objekt wird als Parameter zurückgegeben, sodass eine Validierung und ggf. ein Datenbank-Eintrag vorgenommen werden können. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Verbindung wieder gelöscht.

Modus: Box erzeugen

Dieser Modus kann nur eingeschaltet werden, wenn auf der Eigenschaftenseite **Allgemeines** die Option **Neue Boxen zulassen** gesetzt ist.

Durch das Ziehen der Maus mit gedrückter linker Maustaste lassen sich in diesem Modus neue Boxen erzeugen.

Der **Modus: Box erzeugen** lässt sich auch über das Setzen der Eigenschaft **InteractionMode** auf den Wert **VcCreateBox** aktivieren.

(siehe auch Ereignisse **OnBoxCreate** und **OnBoxCreateComplete**).

Teildiagramm erstellen

(nur aktiv, wenn Knoten markiert sind) Wählen Sie diese Option, um ein Teildiagramm aus den markierten Knoten darzustellen.

Gesamtdiagramm wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teildiagramm erstellen** gewählt wurde)* Wählen Sie diese Option, um das Gesamtdiagramm wiederherzustellen.

Seite einrichten

Sie gelangen in das Dialogfeld **Seite einrichten**.

Sie können dieses Dialogfeld auch mit der API-Methode **PageLayout** aufrufen.

Drucker einrichten

*Nur aktiv, wenn auf der Eigenschaftenseite **Allgemeines** die Option **PrintDlgEx Dialog verwenden** nicht gewählt wurde.*

Sie gelangen in das Windows-Dialogfeld **Drucker einrichten**. Sie können dieses Dialogfeld auch mit der VcGantt-Methode **PrinterSetup** aufrufen.

Druckvorschau

Sie gelangen in das Dialogfeld **Druckvorschau**. Sie können dieses Dialogfeld auch mit der VcGantt-Methode **PrintPreview** aufrufen.

Drucken

Wenn Sie diese Option wählen, gelangen Sie in das Windows-Dialogfeld **Drucken**. Sie können dieses Dialogfeld auch mit der VcGantt-Methode **PrintIt** aufrufen.

Komplettansicht anzeigen

Über diesen Menüpunkt können Sie die Komplettansicht ein- oder ausschalten. Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm angezeigt wird. Ein farbiger Rahmen markiert den aktuell im Hauptfenster dargestellten Diagrammausschnitt. Wenn Sie diesen Rahmen mit der Maus verschieben, wird auch im Hauptfenster der entsprechende Diagrammausschnitt angezeigt.

Die Komplettansicht lässt sich auch über die Eigenschaft **VcWorldView.Visible** an- oder abschalten.

Legendenansicht anzeigen

Über diesen Menüpunkt können Sie die Legendenansicht ein- oder ausschalten. Die Legende erscheint in einem zusätzlichen Fenster.

Sie können die Legendenansicht auch über die Eigenschaft **VcLegendView.Visible** an- oder abschalten.

Diagramm exportieren

Wenn Sie diese Option wählen, gelangen Sie in das Windows-Dialogfeld **Speichern unter**, in dem Sie das dargestellte Diagramm als Grafikdatei speichern können.

Sie können diesen Dialog auch mit der VcGantt Methode **ShowExportGraphicsDialog** aufrufen.

Beim Exportieren wird die Größe des exportierten Diagramms in Pixeln wie folgt berechnet:

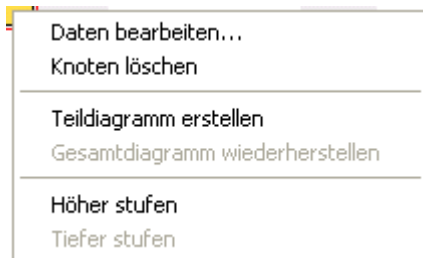
- PNG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen.

- GIF, TIFF, BMP, JPEG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.
- WMF: Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.
- EMF/EMF+: Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand verwendet.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten finden Sie im Kapitel: "Wichtige Konzepte: Grafikformate".

5.28 Kontextmenü für Knoten

Wenn Sie mit der rechten Maustaste einen oder mehrere markierte Knoten anklicken, erscheint das folgende Menü:



Wenn der Anwender auf einen Knoten (location = vcInDiagram) oder auf einen vorgangsbezogenen Tabelleneintrag (location = vcInTable) mit der rechten Maustaste klickt, tritt das Ereignis **OnNodeRClick** ein. Das getroffene Knotenobjekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt werden.

Daten bearbeiten

Wenn Sie diese Option wählen, öffnet sich das Dialogfeld **Vorgänge bearbeiten**. Wenn mehrere Knoten markiert sind, können Sie darin sukzessive die Daten aller markierter Knoten bearbeiten.

Knoten löschen

Wenn Sie diese Option wählen, werden die markierten Knoten gelöscht.

Teildiagramm erstellen

Wählen Sie diese Option, um ein Teildiagramm aus den markierten Knoten darzustellen.

Gesamtdiagramm wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teildiagramm erstellen** gewählt wurde)*
Wählen Sie diese Option, um das Gesamtdiagramm wiederherzustellen.

Höher stufen

(nur für Hierarchie) Der markierte Knoten wird in der Hierarchie eine Ebene höher eingestuft.

Tiefer stufen

(nur für Hierarchie) Der markierte Knoten wird in der Hierarchie eine Ebene tiefer eingestuft.

5.29 Kontextmenü für Verbindungen

Wenn Sie eine Verbindung mit der rechten Maustaste anklicken, erscheint das Kontextmenü **Verbindung löschen**. Wenn Sie die markierte Verbindung löschen möchten, bestätigen Sie dies durch einen Klick mit der linken Maustaste.

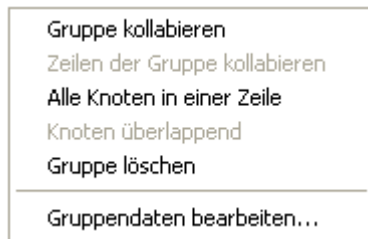


Verbindung löschen

Das Ereignis **OnLinkRClickCltn** tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Verbindung bzw. mehrere sich überlagernde Verbindungen klickt. Ein `LinkCollection`-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Cursors als Parameter übergeben. Sie können so an der betreffenden Position Ihr eigenes Kontextmenü anzeigen. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt werden.

5.30 Kontextmenü für Gruppen

Wenn Sie eine Gruppenüberschrift in der Tabelle oder einen Gruppenbalken im Diagramm (der nur angezeigt wird, wenn im Dialog **Gruppierung** die Option **Gruppenknoten sichtbar** aktiviert wurde) mit der rechten Maustaste anklicken, erscheint ein Kontextmenü, das die wichtigsten Optionen für Gruppen enthält:



Das Ereignis **OnGroupRClick** tritt ein, wenn der Anwender in der Tabelle auf eine Gruppenüberschrift mit der rechten Maustaste klickt. Das getroffene Group-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt werden.

Gruppe kollabieren/Gruppe expandieren

Mit dieser Option können Sie eine kollabierte Gruppe expandieren bzw. eine expandierte Gruppe kollabieren.

Zeilen der Gruppe expandieren/kollabieren

Mit dieser Option können Sie die Zeilen einer kollabierten Gruppe expandieren bzw. die Zeilen einer expandierten Gruppe kollabieren.

Wenn für die Gruppe alle Knoten in einer Zeile dargestellt werden, bewirkt diese Option, dass nur die vorhandenen Untergruppen in der gewählten Gruppe ausgeblendet werden.

Alle Knoten in einer Zeile/Jeder Knoten in eigener Zeile

Wenn Sie die Option **Alle Knoten in einer Zeile** wählen, werden alle Vorgänge einer Gruppe in einer Zeile ausgegeben. Überschneiden sich Vorgänge einer Gruppe, so werden sie im expandierten Status der Gruppe automatisch untereinander angeordnet, um Überlagerungen zu vermeiden.

Wird die Gruppe kollabiert, so sind Überschneidungen von Vorgängen möglich. Bei dieser Art der Anordnung entfällt der Tabellenteil für die Vorgänge. Man muss also die Layerbeschriftung oder den Tooltip verwenden, um die Vorgänge für den Anwender zu identifizieren.

Mit der Option **Jeder Knoten in eigener Zeile** erreichen Sie, dass jeder Knoten in einer eigenen Zeile dargestellt wird.

Knoten überlappend/Knoten optimiert

*nur aktiv, wenn **Alle Knoten in einer Zeile** gewählt wurde.* Wenn Sie die Option **Knoten überlappend** wählen, werden die Knoten einer Gruppe in einer Zeile dargestellt, auch wenn es dabei zu Überschneidungen kommt. Wenn Sie die Option **Knoten optimiert** wählen, werden die Knoten einer Gruppe ohne Überschneidungen dargestellt, auch wenn sie dann u. U. nicht in einer Zeile dargestellt werden können.

Gruppe löschen

Mit dieser Option können Sie die markierte Gruppe löschen, vorausgesetzt sie ist leer.

Gruppendaten bearbeiten

Mit dieser Option rufen Sie den gleichnamigen Dialog auf.

5.31 Kontextmenü für die Zeitskala

Wenn Sie mit der rechten Maustaste auf die Zeitskala klicken, erscheint das folgende Menü:



Das Ereignis **OnTimeScaleRClick** tritt ein, wenn der Anwender mit der rechten Maustaste auf die Zeitskala klickt. Das getroffene TimeScale-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. So können Sie an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrückt werden.

Zeitskala bearbeiten

Wenn Sie diese Option wählen, erscheint das Dialogfeld **Zeitskala bearbeiten**.

Stichtaglinie

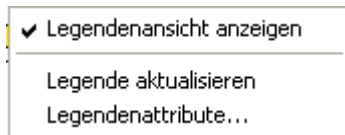
Wählen Sie, ob eine Stichtaglinie in Ihrer Darstellung angezeigt werden soll.

Gitternetzlinien

Wählen Sie, ob Gitternetzlinien in Ihrer Darstellung dargestellt werden sollen.

5.32 Kontextmenü für die Legende

Wenn Sie mit der rechten Maustaste auf die Legende klicken, erscheint das folgende Menü:



Legendenansicht anzeigen

Über diesen Menüpunkt können Sie die Legendenansicht ein- oder ausschalten.

Legende aktualisieren

Über diesen Menüpunkt können Sie die Legendenansicht aktualisieren.

Eine Aktualisierung über das Menü kann nötig sein, da nach Änderungen im Diagramm die Legende nicht automatisch aktualisiert wird. Werden also zum Beispiel Knoten hinzugefügt oder gelöscht, muss eine Aktualisierung entweder über das Kontextmenü oder durch Aus- und Einschalten der Legende durchgeführt werden. Dies gilt auch für das Laden von Knoten. Wenn für die Legendenansicht auf der Eigenschaftenseite **Zusätzliche Ansichten** die Option **Beim Start sichtbar** eingestellt wurde, aber zum Zeitpunkt des Aufbaus noch keine Knoten geladen waren, bleibt die Legende bis zur Aktualisierung leer.

Legendenattribute

Über diesen Menüpunkt öffnen Sie den gleichnamigen Dialog, in dem Sie Einstellungen zum Legendentitel, den Legendenelementen und zu den Rändern der Legende vornehmen können. Weitere Information zu diesem Dialog finden Sie in Kapitel 4.44 "Dialogfeld Legendenattribute".

5.33 Kontextmenü für Boxen

Wenn Sie mit der rechten Maustaste auf eine Box klicken, erscheint das folgende Menü:



Sollte das Kontextmenü nicht erscheinen, muss auf der Eigenschaftenseite **Allgemeines** die Option **Kontextmenü für Boxen anzeigen** aktiviert werden.

Box am markierten Knoten verankern

Mit diesem Befehl können Sie eine Box am markierten Knoten verankern. Dies ist nur möglich, wenn zuvor im Dialogfeld **Boxen verwalten** die Option **Verankerungsinteraktionen erlaubt** ausgewählt wurde.

Box vom Knoten lösen

Mit diesem Befehl können Sie die Verankerung wieder lösen.

6 Häufig gestellte Fragen

6.1 Wie kann das Steuerelement neu lizenziert werden? Was ist bei Problemen mit der Lizenzierung zu tun?

Wenn Sie die Nutzungsdauer des VARCHART ActiveX-Steuerelementes verlängern oder ein neu erworbenes Modul lizenzieren möchten, öffnen Sie bitte den Dialog **Lizenzierung**.

Sie erreichen diesen Dialog über die Eigenschaftenseite **Allgemeines**. Um die Lizenzierung vorzunehmen, klicken Sie auf die **Anfordern**-Schaltfläche. Dann erscheint der Dialog **Lizenzinformationen anfordern**.

Geben Sie dort Ihre Lizenznummer, Ihren Namen und den Namen Ihrer Firma an und klicken sie auf **E-Mail an NETRONIC senden**. Damit wird automatisch eine E-Mail generiert, die Sie nur noch absenden müssen. Sobald wir Ihre E-Mail erhalten haben, werden wir unverzüglich eine Datei mit Ihren Lizenzinformationen (vcgantt.lic) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei *vcgantt.ocx* steht.

Wenn Sie die neue Lizenzierung vorgenommen haben, müssen Sie die neue Lizenzierung noch aktivieren. Öffnen Sie dazu eine beliebige Eigenschaftenseite, nehmen Sie dort eine beliebige Änderung vor und speichern Sie diese. Nun ist die neue Lizenzierung aktiviert.

Wenn Sie beim Lizenzieren des VARCHART ActiveX-Steuerelementes die Fehlermeldung "REGSVR32 Error Return: 0X0000007e" erhalten, ist die DLL *vcwin32u.dll* nicht vorhanden oder steht nicht in einem im PATH angegebenen Verzeichnis. Sollte die Datei fehlen, wenden Sie sich bitte an den Kundendienst der NETRONIC Software GmbH.

6.2 Wie kann erreicht werden, dass eine veränderte ini-Datei des VARCHART ActiveX-Steuerelementes verwendet wird?

Einige Einstellungen für das VARCHART ActiveX-Steuerelement können auf den Eigenschaftenseiten nicht eingestellt werden, lassen sich durch das Editieren der ini-Datei aber trotzdem verändern. Gehen Sie dazu folgendermaßen vor:

1. Öffnen Sie die Eigenschaftenseite **Allgemeines**. Dort wird im Feld **Konfigurationsdatei** die aktuell verwendete Konfigurationsdatei (z.B. *projekt.ini*) angezeigt.
2. Klicken Sie auf die Schaltfläche **Durchsuchen**. Das Dialogfeld **Laden bzw. Speichern** öffnet sich. Geben Sie hier unter **Dateiname** den Namen für eine Konfigurationsdatei an, die nur vorübergehend verwendet werden soll, z. B. *dummy.ini*. Klicken Sie auf **Speichern**.
3. Klicken Sie nun auf der Eigenschaftenseite **Allgemeines** auf **OK** bzw. **Übernehmen**. Die Konfigurationsdatei *dummy.ini* wird nun automatisch erzeugt und verwendet.
4. Nun können Sie Ihre ini-Datei (z. B. *projekt.ini*) in einem Editor (z. B. Microsoft WordPad) bearbeiten und Ihre Änderungen speichern.
5. Wählen Sie anschließend auf der Eigenschaftenseite **Allgemeines** unter **Konfigurationsdatei** Ihre bearbeitete Konfigurationsdatei (*projekt.ini*) wieder aus und klicken Sie auf **OK**. Nun wird die bearbeitete ini-Datei (*projekt.ini*) des VARCHART ActiveX-Steuerelementes verwendet.

6.3 Was müssen Benutzer von Borland Delphi beim Upgrade auf eine neue Version von VARCHART XGantt tun?

Nach dem Upgrade oder Update von VARCHART XGantt auf eine neue Version ist es notwendig, diese neue Version in dem Delphi Package Borland Anwenderkomponenten neu zu installieren. Dazu gehen Sie bitte wie folgt vor:

- Starten Sie Borland Delphi.
- Klicken Sie auf **Komponente** und **ActiveX importieren**.
- Selektieren Sie in der Liste der ActiveX Controls *NETRONIC VARCHART XGantt* und entfernen Sie die Registrierung durch Klick auf die Schaltfläche **Entfernen**. Verlassen Sie dann den Dialog mit **Abbrechen**.
- Öffnen Sie nun den Dialog **Komponente > Packages installieren** und wählen Sie das Package *Borland Anwenderkomponenten*. (Dieses Package ist in der Datei *dclusr*0.bpl* gespeichert, wobei das '*' im Dateinamen abhängig von Ihrer Delphi-Version ist: 5, 6 oder 7.)
- Klicken Sie auf **Bearbeiten**. Damit wird die Datei *dclusrX0.dpk* geöffnet.
- Selektieren Sie nun nacheinander die Dateien *VcGanttLib_TLB.pas* und *VcGanttLib_TLB.dcr* und entfernen Sie sie jeweils durch einen Rechtsklick aus dem Projekt.
- Kompilieren Sie nun das Package und schließen Sie danach den Dialog. Dadurch werden die Änderungen in dem Projekt *dclusrX0* gespeichert.
- Öffnen Sie nun wieder den Dialog **Komponente > ActiveX importieren**.
- Klicken Sie auf **Hinzufügen**, wählen Sie *vcgantt.ocx* aus und klicken Sie auf **Öffnen**. Jetzt sehen Sie *NETRONIC VARCHART XGantt* wieder in der Liste der registrierten ActiveX Controls.
- Klicken Sie auf **Installieren...**. Damit wird das Package *dclusrX0.bpl* neu kompiliert.
- Schließen Sie den Dialog. Damit wird das Projekt *dclusrX0* gespeichert.

6.4 Was tun, wenn das Steuerelement unerwartet nicht in allen Benutzerkonten eines Rechners funktioniert?

Wenn Sie feststellen, dass das Steuerelement nicht ansprechbar ist, wenn ein anderer Benutzer die gleiche, das Steuerelement nutzende Applikation aufruft, dann liegt es daran, dass das Steuerelement nicht für alle Benutzer installiert wurde. Bei der Erstellung des Setup-Programms, mit dem Sie das Steuerelement auf dem Kundenrechner installieren, sollte die Option "Für alle Benutzer installieren" ausgewählt sein.

Nachträglich oder von Hand lässt sich eine Installation für mehrere Benutzerkonten freischalten, indem Sie die Sicherheitseinstellungen der zum Steuerelement gehörenden Dateien so erweitern, dass andere Benutzerkonten als das, unter dem die Installation stattfand, auf die Dateien zugreifen dürfen. Die Sicherheitseinstellungen können Sie einerseits über den Menüpunkt "Eigenschaften" des Kontextmenüs zur jeweiligen Datei ändern oder per Kommandozeile über den Befehl 'cacls'. Die zum Steuerelement gehörenden Dateien finden Sie im Kapitel "Shipping the Application" am Anfang dieses Buches.

6.5 Wie lässt sich in VARCHART XGantt der XP Visual Style aktivieren?

Der XP Visual Style ist abhängig von der verwendeten Version der Common Controls von Windows. In Windows XP werden die Versionen 5 und 6 mitgeliefert; der Kompatibilität wegen wird aber standardmäßig Version 5 benutzt. Das Umschalten einer Applikation auf Version 6 geschieht durch eine Eintragung in einem sogenannten Manifest, das seit XP von Windows unterstützt wird. Dieses Manifest kann entweder einkompiliert werden (Ressource mit ID 24 bzw. RT_MANIFEST) oder als Datei im selben Verzeichnis wie die EXE-Datei abgelegt werden (der Name setzt sich dann zusammen aus dem EXE-Name + ".manifest", z.B. notepad.exe.manifest). Der Inhalt des Manifests besitzt XML-Format und sieht folgendermaßen aus:

Code-Beispiel

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1"
manifestVersion="1.0">
  <assemblyIdentity version="1.0.0.0" processorArchitecture="x86"
                    name="appname.exe" type="win32" />
  <description>Hier Applikationsbeschreibung eintragen</description>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity type="win32" name="Microsoft.Windows.Common-
Controls" version="6.0.0.0"
                        processorArchitecture="x86"
publicKeyToken="6595b64144ccf1df"
                        language="*" />
    </dependentAssembly>
  </dependency>
</assembly>
```

Man sollte darauf achten, dass der Win32-API-Befehl **InitCommonControls** beim Applikationsstart aufgerufen wird, da sonst evtl. die alten Common Controls in Version 5.0 geladen werden, bevor das Manifest ausgewertet wird. In VB 6 kann man ihn z.B. im Ereignis **Form_Initialize** der Start-Form eintragen, vorher ist allerdings noch folgende Zeile notwendig:

Code-Beispiel

```
Private Declare Sub InitCommonControls Lib "comctl32.dll" ()
```

Das Manifest selbst sollte angepasst werden auf Applikationsnamen und -version (s. <assemblyIdentity> und <description>).

Weitere Informationen finden Sie auch auf den folgenden Internet-Seiten:

http://www.activevb.de/tutorials/tut_xpstyles/xpstyles.html

<http://support.microsoft.com/?id=309366>

<http://support.microsoft.com/?id=303636>

6.6 Wie kann die Spreizung der Zeitskala begrenzt werden?

Wenn die Zeitskala sehr weit links im sichtbaren Bereich mit gedrückter linker Maustaste angepackt wird, um die Zeitskala zu spreizen, kann leicht ein Faktor weit über 1.000% erreicht werden. Das lässt sich im Ereignis **OnTimeScaleSectionRescale** leicht abfangen. Im folgenden Beispiel wird jeweils maximal eine Verdopplung der Auflösung zugelassen.

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleSectionRescale(ByVal timeScale As _  
    VcGanttLib.VcTimeScale, ByVal sectionIndex As _  
    Integer, ByVal newBasicUnitWidth As Long, _  
    returnStatus As Variant)  
  
    Dim nOldUnitWidth As Long  
  
    nOldUnitWidth = timeScale.Section(sectionIndex).UnitWidth  
  
    If newBasicUnitWidth > (2 * nOldUnitWidth) Then  
        timeScale.Section(sectionIndex).UnitWidth = 2 * nOldUnitWidth  
        returnStatus = vcRetStatFalse  
    End If  
  
End Sub
```

6.7 Wie kann beim Klick in die Tabelle der zugehörige Balken in den sichtbaren Bereich gebracht werden?

Beim Ereignis **OnNodeLClick** wird neben dem Knoten auch die Information **InTable** oder **InDiagram** übergeben. Fand der Klick in der Tabelle statt (**InTable**), wird aus dem Knoten das relevante Datum ausgefragt und mit der Methode **ScrollToDate** an das VARCHART ActiveX-Objekt übergeben.

Code-Beispiel

```
Private Sub VcGantt1_OnNodeLClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)
    Dim myDataDef As VcDataDefinition
    Dim myDataDefTable As VcDataDefinitionTable
    Dim myDataField As VcDefinitionField
    Dim myIndex As Integer

    If location = vcInTable Then
        ' falls der Index des Feldes "Start" nicht bekannt ist
        Set myDataDef = VcGantt1.DataDefinition
        Set myDataDefTable = myDataDef.DefinitionTable(vcMaindata)
        Set myDataField = myDataDefTable.FieldByName("Start")
        myIndex = myDataField.ID
        VcGantt1.ScrollToDate node.DataField(myIndex), vcLeftAligned, 2
    End If
End Sub
```

6.8 Wie werden Überschneidungen von Vorgängen einer Gruppe sichtbar?

Um Engpässe durch die Überschneidung des Urlaubs mehrerer Mitarbeiter oder durch die Mehrfachbelegung von Maschinen zu verhindern, können die Überschneidungen in einer Gruppe von Vorgängen sichtbar gemacht werden. Überschneidungen von Vorgängen kann es grundsätzlich nur geben, wenn eine Gruppierung verwendet wird und im Dialog **Gruppierung** die Option **Jeder Knoten in eigener Zeile nicht** gewählt wurde. Bei dieser Einstellung können Gruppen kollabiert oder expandiert dargestellt werden. Bei kollabierten Gruppen sind Überschneidungen nicht zu erkennen, während sie bei expandierten Gruppen durch eine versetzte Anordnung sichtbar gemacht werden.

Um Überschneidungen sichtbar zu machen, deaktivieren Sie ggf. im Dialog **Gruppierung** die Option **Jeder Knoten in eigener Zeile**, um die Vorgänge einer Gruppe in einer Zeile auszugeben. Überschneiden sich Vorgänge einer Gruppe, so werden sie auch bei dieser Option untereinander angeordnet. Dadurch erkennen Sie die Überschneidungen auf einen Blick.

Wenn die Vorgänge kollabiert werden, sind Überschneidungen von Vorgängen jedoch nicht zu erkennen. Um sicherzustellen, dass der Anwender die Überschneidung von Vorgängen nicht übersieht, sollten Sie die Option **Änderungen erlaubt** abschalten, um das Umschalten zwischen diesen beiden Darstellungsmodi zu verhindern. Deaktivieren Sie außerdem die Option **Zu Anfang kollabiert**. Dann erscheinen alle Gruppen expandiert, und Überschneidungen fallen sofort ins Auge, da die Vorgänge an den entsprechenden Stellen untereinander dargestellt werden.

6.9 Wie lässt sich die Reihenfolge der Vorgänge sichern und laden?

Die Voraussetzung für das Sichern und erneute Laden von Vorgängen ist, dass die Vorgänge aus einer Datei geladen werden.

Um die Reihenfolge der Vorgänge zu sichern und zu laden, wählen Sie auf der Eigenschaftenseite **Sortierung** unter **Feld für Zeilennummer** ein Datenfeld aus. In dieses Datenfeld schreibt das VARCHART ActiveX-Steuererelement dann die Identifikation.

Wenn die Knotenreihenfolge interaktiv verändert wurde, lässt sie sich mit der Methode **UpdateRowNumberField** aktualisieren. (Gruppierung und Hierarchie dürfen dabei nicht eingeschaltet sein.)

Ergänzen Sie schließlich die folgenden Code-Zeilen:

Code-Beispiel

```
Private Sub Form_Unload ()  
    VcGantt1.UpdateRowNumberField  
    VcGantt1.SaveAs (" ")  
End Sub
```

6.10 Wieso können Knoten u. U. nicht interaktiv erzeugt werden?

Wenn Sie zur Laufzeit keine Knoten mit der Maus anlegen können, stellen Sie sicher, dass auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Neue Knoten zulassen** aktiviert ist.

Wenn außerdem das Kontrollkästchen **Neue Knoten durch Doppelklick** aktiviert ist, können Sie zur Laufzeit Knoten durch Doppelklick erzeugen.

Außerdem lassen sich keine Knoten in arbeitsfreien Bereichen anlegen, wenn ein Kalender aktiviert ist.

Kontrollieren Sie außerdem, ob in Ihrem Programmcode die Eigenschaft **AllowNewNodes** auf **False** gesetzt wurde; dann können ebenfalls keine neuen Knoten angelegt werden.

6.11 Wie lassen sich die Standard-Kontextmenüs abschalten?

Die vordefinierten Kontextmenüs können Sie durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrücken.

Code-Beispiel

```
' Kontextmenü für das Diagramm abschalten
Private Sub VcGantt1_OnDiagramRClick(ByVal x As Long, _
                                     ByVal y As Long, returnStatus As
Variant)
    returnStatus = vcRetStatNoPopup
End Sub

' Kontextmenü für Verbindungen abschalten
Private Sub VcGantt1_OnLinkRClickCltn(ByVal linkCltn As _
                                       VcGanttLib.VcLinkCollection, _
                                       ByVal x As Long, _
                                       ByVal y As Long, _
                                       returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub

' Kontextmenü für Knoten abschalten
Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
                                   ByVal location As _
                                   VcGanttLib.LocationEnum, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub
```

6.12 Was ist bei Problemen mit dem Drucken zu tun?

Wenn Sie Ihr Diagramm nicht drucken können oder den Drucker nicht einrichten können, prüfen Sie bitte, ob die Datei *vcprct32.dll* vorhanden ist. Prüfen Sie außerdem, ob sie in dem im PATH angegebenen Verzeichnis enthalten ist, und ob der Standard-Windows-Drucker eingerichtet ist.

Sollte die Datei *vcprct32.dll* fehlen, wenden Sie sich bitte an den Kundendienst der NETRONIC Software GmbH.

6.13 Wie lässt sich die Performance verbessern?

> SuspendUpdate

Bei größeren Datenmengen kann es unter Umständen sehr lange dauern, wenn bei einer großen Anzahl von Vorgängen dieselbe Aktion durchgeführt wird. Nicht jeder automatische Aktualisierungsvorgang im Diagramm ist notwendig; in einem solchen Fall können Sie Aktualisierungen für den Einzelfall unterdrücken und nach der Abarbeitung einer Code-Sequenz final einmal durchführen. Unterdrückung und Wiedereinsatz der Aktualisierung erfolgen mit der Methode **SuspendUpdate**, die zu Beginn der Code-Sequenz auf **True** bzw. am Ende auf **False** gesetzt wird. Auf diese Weise können Sie die Performance insgesamt beträchtlich erhöhen.

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcGantt1.SuspendUpdate (True)

For Each dataRecord In dataRecordCltn
    dataRecord.DataField(4) = "10"
    dataRecord.UpdateDataRecord
Next

VcGantt1.SuspendUpdate (False)
```

Die Performance lässt sich analog auch beim Update von vielen Verbindungen verbessern.

Auch beim Modifizieren von Tabellenformaten lässt sich die Performance steigern.

Code-Beispiel

```
Private Sub ModifyTable_Click()
    Dim formatCol As VcTableFormatCollection
    Dim aFormat As VcTableFormat
    Dim index As Integer
    VcGantt1.SuspendUpdate True
    Set formatCol = VcGantt1.Table.TableFormatCollection
    For Each aFormat In formatCol
        For index = 1 To aFormat.NoOfColumns
            aFormat.FieldBackgroundColor(index) = vbGreen
            aFormat.FieldFontBody(index) = vbBold
            aFormat.FieldFontColor(index) = vbRed
            aFormat.FieldFontName(index) = "Arial"
            aFormat.FieldFontSize(index) = 14
        Next
    Next
    VcGantt1.SuspendUpdate False
End Sub
```


452 Häufig gestellte Fragen

```
                aFormat.FieldHorAlignment(index) = vcHorCenterAligned
            Next
        Next
        VcGantt1.SuspendUpdate False
    End Sub
```

Auch bei nicht äquidistanten Histogrammkurven erreichen Sie mit dieser Methode eine signifikante Performancesteigerung.

Code-Beispiel

```
Private Sub CommandCreateCurve_Click()

    Dim myCurve As VcCurve
    Set myCurve =
    VcGantt1.HistogramCollection.FirstHistogram.CurveCollection. _
        CurveByName("Curve1")
    Dim index As Integer
    Dim aDate As Date

    'aktuelles Datum
    aDate = Date
    VcGantt1.SuspendUpdate True

    For index = 1 To 3000
        ' verschieben um 2h:24min
        aDate = aDate + 1 / 10
        myCurve.SetValues aDate, index
    Next
    VcGantt1.SuspendUpdate False

End Sub
```

Auch bei der Verwendung von Kalendern können Sie durch diese Methode das Update beschleunigen, denn wenn die Knoten schon geladen sind, erfordern Mehrfachmodifikationen an Kalendern einige Zeit, weil alle Knoten auf Kalenderabhängigkeiten geprüft werden.

> Grafiken

Eine Ursache für eine zu geringe Performance können Grafiken beispielsweise in Tabellen-, Knoten- oder Boxfeldern sein, die zu groß sind oder eine zu große Pixelzahl haben.

6.14 Fehlermeldungen

> Fehlermeldungen zur Laufzeit, vom Entwickler verursacht

Fehlerursache	Meldung
Lizenzierung fehlgeschlagen	Sie nutzen eine unlizenzierte Version von *. Bitte wenden Sie sich an NETRONIC, um eine vollständig lauffähige Version zu erhalten.
	Die Lizenzierung meldet einen Fehler. Bitte setzen Sie sich mit NETRONIC in Verbindung.
	Die Nutzungsberechtigung ist abgelaufen. Bitte setzen Sie sich mit NETRONIC in Verbindung.
	Ihre Identifizierung hat sich von * in * geändert, bitte wenden Sie sich an NETRONIC!
	Das in diesem Programm benutzte ActiveX-Steuerelement hat keine Runtime-Lizenz!
ActiveX-Installation unvollständig oder ältere DLL-Versionen im Systempfad	Die DLL * wurde nicht gefunden
	Das Interface (*) konnte nicht geladen werden
	Die Interface- DLL (Version **) ist zu alt. Dieses Programm benötigt Version * oder jünger.
Programminstallation unvollständig oder absoluter Pfad fehlerhaft	Die Gruppentiteldatei konnte nicht gefunden werden.
	Die Datei * ist keine gültige Grafikdatei.
	Grafikfile ist nicht angegeben oder existiert nicht.
Fehler bei der Zuweisung einer neuen INI-Datei	Konfigurationsfile * nicht gefunden, Programm erzeugt es mit Standardkonfiguration.
INI-Datei hat Fehler	Die Knotenkennzeichnung/Die Tabelle/Der Layer * verlangt den nicht existierenden Filter *. Der Filtereintrag wird auf <immer> korrigiert.
	Die Knotenkennzeichnung/Die Tabelle * verlangt die nicht existierende Knotenbeschriftung *. Der Knotenbeschriftungseintrag wird auf <nicht festgelegt> korrigiert.
	Der Name * für Layer ist doppelt vorhanden. Bitte Konfigurationsdatei prüfen.
	Highlight * existiert nicht
	Der Name * für Verbindungsaussehen ist doppelt vorhanden. Bitte Konfigurationsdatei(en) prüfen.
	Your configuration file * is corrupt. [*] must be unique.

.

> Fehlermeldungen zur Laufzeit, verursacht vom Endanwender oder vom Entwickler

Fehlerursache	Meldung
Zyklen in der Methode ScheduleProject entdeckt	Das Projekt enthält Zyklen!
Interaktives Knotenverschieben	Es können keine neuen Gruppen angelegt werden.

6.15 Sind alle Fonts verwendbar?

Bedingt durch die Umstellung auf GDI+ gibt es Einschränkungen bzgl. der Darstellung von Schriftarten. So ist GDI+ nicht in der Lage, Fonts darzustellen, die als Postscript- oder gar Bitmapfonts vorliegen. Zu ersteren gehören auch Fonts, die zwar vom Typ **OpenType** sind, aber als "klassische Fonts" intern eine Art Postscript-Struktur besitzen (wie z.B. "Warnock Pro"). Zu den letzteren gehören z.B. die "alten" Windows-Fonts "Courier" und "Times", aber auch "System" und "MS Sans Serif".

Diese Fonts werden von den Fontauswahldialogen in VARCHART XGantt daher nicht angeboten. Werden sie dennoch über die API gesetzt, so wird ein Ersatzfont angezeigt. Bei den "alten" Windows-Fonts sind von NETRONIC Ersetzungsregeln eingebaut worden, die einen ähnlichen "neuen" Font auswählen, bei externen Fonts wird dann "Arial" als Ersatz genommen, damit überhaupt Text auf dem Bildschirm erscheint.

Es ist zwar unwahrscheinlich, aber nicht auszuschließen, dass in zukünftigen Versionen von GDI+ Unterstützung für die bisher nicht unterstützten Fonts eingebaut wird. Informationen sind bei Microsoft leider nur in Blogs und Newsgruppen zu erhalten, nicht jedoch bei MSDN.

7 API Referenz

7.1 Objekttypen

- DataObject
- DataObjectFiles
- VcBorderArea
- VcBorderBox
- VcBox
- VcBoxCollection
- VcBoxFormat
- VcBoxFormatCollection
- VcBoxFormatField
- VcCalendar
- VcCalendarCollection
- VcCalendarGrid
- VcCalendarGridCollection
- VcCalendarProfile
- VcCalendarProfileCollection
- VcCurve
- VcCurveCollection
- VcDataDefinition
- VcDataDefinitionTable
- VcDataDefinitionTable
- VcDataRecord
- VcDataRecordCollection
- VcDataTable
- VcDataTableCollection
- VcDataTableField
- VcDataTableFieldCollection
- VcDateLine
- VcDateLineCollection
- VcDateLineGrid
- VcDateLineGridCollection
- VcDefinitionField
- VcField

- VcFilter
- VcFilterCollection
- VcFilterSubCondition
- VcGantt
- VcGroup
- VcGroupCollection
- VcGroupLevelLayout
- VcGroupLevelLayoutCollection
- VcHierarchyLevelLayout
- VcHistogram
- VcHistogramCollection
- VcInfoWindow
- VcInterval
- VcIntervalCollection
- VcLayer
- VcLayerCollection
- VcLayerFormat
- VcLayerFormatField
- VcLegendView
- VcLineFormat
- VcLineFormatCollection
- VcLineFormatField
- VcLink
- VcLinkAppearance
- VcLinkAppearanceCollection
- VcLinkCollection
- VcMap
- VcMapCollection
- VcMapEntry
- VcNode
- VcNodeCollection
- VcNodeLevelLayout
- VcNumericScale
- VcNumericScaleCollection
- VcPrinter
- VcRect
- VcResourceScheduler2
- VcRibbon
- VcScheduler
- VcSection

- VcTable
- VcTableCollection
- VcTableFormat
- VcTableFormatCollection
- VcTableFormatField
- VcTimeScale
- VcTimeScaleCollection
- VcUpdateBehavior
- VcUpdateBehaviorCollection
- VcUpdateBehaviorContext
- VcWorldView

7.2 DataObject

DataObject

Durch die OLE-Drag&Drop-Technik können Sie ausgewählte Knoten in ein anderes VARCHART-ActiveX-Steuerelement ziehen und dort ablegen. Der Transportspeicher für diese Daten ist das DataObject-Objekt. Zu diesem Zweck stellt das Objekt die passenden Methoden und Eigenschaften bereit: **Files**, **Clear**, **GetData**, **GetFormat** und **SetData**.

Sie können auch Daten mit anderen OLE-Drag&Drop-fähigen Steuerelementen und Anwendungen austauschen. Bedenken Sie bitte dabei, dass die VARCHART-ActiveX-Steuerelemente die Daten immer im CSV-Textformat ablegen und interpretieren.

Damit OLE Drag & Drop einsetzbar wird, müssen im Eigenschaften-Fenster die Eigenschaften **OLEDragMode** und **OLEDropMode** aktiviert werden. Auf der Eigenschaftsseite **Knoten** bestimmt die Option **Move all selected nodes**, ob nur ein einzelner oder gleichzeitig mehrere Knoten verschoben werden können.

Nähere Erläuterungen zu diesem Thema finden Sie im Kapitel **Wichtige Begriffe** unter der dem Thema **OLE-Drag&Drop**.

Eigenschaften

- DropEndDate
- DropStartDate
- Files

Methoden

- Clear
- GetData
- GetFormat
- SetData

Eigenschaften

DropEndDate

Nur-Lese-Eigenschaft von DataObject

Diese Eigenschaft gibt das Enddatum des Drop-Vorgangs an. Wenn **OLE-DropMode** auf **vcOLEDropManual** gesetzt wurde, kann diese Eigenschaft dazu verwendet werden, das Enddatum des Phantoms auszufragen, um dieses einem neu erzeugten Knoten mitzugeben.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Enddatum

DropStartDate

Nur-Lese-Eigenschaft von DataObject

Diese Eigenschaft gibt das Startdatum des Drop-Vorgangs an. Wenn **OLEDropMode** auf **vcOLEDropManual** gesetzt wurde, kann diese Eigenschaft dazu verwendet werden, das Startdatum des Phantoms auszufragen, um dieses einem neu erzeugten Knoten mitzugeben.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Startdatum

Files

Nur-Lese-Eigenschaft von DataObject

Diese Eigenschaft gibt eine DataObjectFiles-Auflistung zurück, die wiederum eine Liste aller Dateinamen enthält, die von einem DataObject-Objekt verwendet werden (wie beispielsweise Namen von Dateien, die ein Benutzer in oder aus dem Windows-Explorer zieht). Diese Eigenschaft kann nur benutzt werden, wenn das DataObject Daten im Format **15 (Dateiliste, s. Eigenschaft GetFormat)** enthält.

	Datentyp	Beschreibung
Eigenschaftswert	DataObjectFiles	Liste der verfügbaren Dateien

Methoden

Clear

Methode von DataObject

Mit dieser Methode können Sie den Inhalt des DataObject-Objekts löschen. Diese Methode steht nur für Drag-Operationen zur Verfügung, das heißt, für die Ereignisse **OLEStartDrag**, **OLESetData**, **OLEGiveFeedback** und **OLECompleteDrag**.

	Datentyp	Beschreibung
Rückgabewert	Void	

GetData

Methode von DataObject

Diese Methode gibt Daten vom Typ **Variant** aus einem DataObject zurück und steht nur für DataObject-Objekte der Ereignisse **OLEDragOver** und **OLEDragDrop** zur Verfügung.

Die Methode **GetData** kann auch andere als die unten aufgeführten Datenformate verwenden, wozu unter anderem auch benutzerdefinierte Formate gehören, die in Windows über die API-Funktion **RegisterClipboardFormat()** registriert sind. Hierbei sind jedoch einige Einschränkungen zu berücksichtigen:

Die **GetData**-Methode gibt Daten immer in Form eines Byte-Datenfelds zurück, wenn sie ein Format haben, das die Methode nicht erkennt.

Das von der **GetData**-Methode zurückgegebene Byte-Datenfeld ist ggf. größer als die tatsächlichen Daten, wobei sich am Ende des Datenfelds beliebige Bytes befinden. Die Ursache dafür liegt darin, dass VARCHART ActiveX das Format der Daten nicht kennt und lediglich weiß, wieviel Speicher das Betriebssystem für die Daten reserviert hat. Dieser reservierte Speicher ist häufig größer als der tatsächlich für die Daten erforderliche Speicher. Aus diesem Grund können sich am Ende des reservierten Speichersegments möglicherweise überflüssige Bytes befinden. Dies führt dazu, dass Sie die richtigen Funktionen für eine sinnvolle Interpretation der zurückgegebenen Daten verwenden müssen (z. B. in Visual Basic das

Abschneiden einer Zeichenfolge an einer bestimmten Stelle mit der **Left**-Funktion, falls die Daten im Textformat vorliegen).

Hinweis: Nicht alle Anwendungen unterstützen die Formate **2** (Bitmap) oder **9** (Farbpalette), daher sollten Sie möglichst immer **8** (Geräteunabhängige Bitmap) verwenden.

	Datentyp	Beschreibung
Parameter: ⇒ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vbCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF Mögliche Werte: Datenfeldindex
Rückgabewert	Variant	geholte Daten

GetFormat

Methode von DataObject

Diese Methode gibt einen booleschen Wert zurück, der anzeigt, ob im DataObject Daten eines bestimmten Formats vorhanden sind. Sie steht nur für DataObject-Objekte der Ereignisse **OLEDragOver** und **OLEDragDrop** zur Verfügung.

	Datentyp	Beschreibung
Parameter: ⇒ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vcCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF Mögliche Werte: Datenfeldindex
Rückgabewert	Boolean	Die Methode GetFormat liefert True zurück, wenn ein Element des DataObject-Objektes dem spezifizierten Format entspricht, andernfalls wird False zurückgegeben.

SetData

Methode von DataObject

Diese Methode fügt Daten im angegebenen Datenformat zum DataObject hinzu und steht nur für DataObject-Objekte der Ereignisse **OLEStartDrag**, **OLESetData**, **OLEGiveFeedback** und **OLECompleteDrag** zur Verfügung.

Die Methode **SetData** kann auch andere als die unter **Format** aufgeführten Datenformate verwenden, wozu unter anderem auch benutzerdefinierte Formate gehören, die in Windows über die API-Funktion **RegisterClipboardFormat()** registriert sind. Hierbei sind jedoch einige Einschränkungen zu berücksichtigen:

Für die **SetData**-Methode müssen die Daten in Form eines Byte-Datenfelds vorliegen, wenn das angegebene Datenformat nicht erkannt wird.

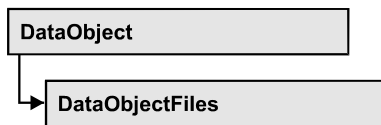
Nicht alle Anwendungen unterstützen **2** (Bitmap) oder **9** (Farbpalette), daher sollten Sie möglichst immer **8** (geräteunabhängige Bitmap) verwenden.

	Datentyp	Beschreibung
Parameter: ⇒ data	Variant	Zu setzende Daten oder Empty , falls bekannt gegeben werden soll, dass das Format über das Ereignis OLESetData bei Bedarf gesetzt werden kann.

466 API-Referenz: DataObject

⇒ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vcCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF Mögliche Werte: Datenfeldindex
Rückgabewert	Void	

7.3 DataObjectFiles



Dieses Objekt verwaltet eine Liste aller Dateinamen, die in einem DataObject gespeichert sind, wenn dieses Daten im Format **15** (Dateiliste) enthält. Über **For Each Item in DataObjectFiles** können Sie in einer Schleife auf alle Dateinamen zugreifen.

Eigenschaften

- `_NewEnum`
- `Count`
- `Item`

Methoden

- `Add`
- `Clear`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von DataObjectFiles

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datenobjekt-Dateien iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```

Private Sub VcGantt1_OLEDragOver(ByVal data As VcGanttLib.DataObject, effect As
Long, ByVal button As Integer, ByVal Shift As Integer, ByVal x As Long, ByVal y
As Long, ByVal state As VcGanttLib.OLEDragStateEnum)

Dim fileName as String
For Each fileName In DataObject.DataObjectFiles
    Debug.Print fileName
Next

End Sub

```

Count**Nur-Lese-Eigenschaft von DataObjectFiles**

Diese Eigenschaft gibt die Anzahl der verwalteten Dateinamen zurück.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Dateinamen

Item**Eigenschaft von DataObjectFiles**

Diese Eigenschaft setzt oder erfragt einen der verwalteten Dateinamen über den angegebenen Index. Da dies die definierte Standardeigenschaft des Objekts ist, kann in vielen Entwicklungsumgebungen (z. B. Visual Basic) der Eigenschaftsname weggelassen werden. Beispiel: DataObjectFiles(0) erfragt den ersten Dateinamen.

	Datentyp	Beschreibung
Parameter: ⇒ index	Long	Index des Dateinamens {0...Count-1}
Eigenschaftswert	String Mögliche Werte:	Dateiname Name der Farbzusordnungstabelle

Methoden

Add

Methode von DataObjectFiles

Mit dieser Methode können Sie den angegebenen Dateinamen zur Liste der Dateinamen hinzufügen. Ist ein Index (Integer, Wertebereich 0 bis .Count-1) angegeben, dann wird der Dateiname an der entsprechenden Stelle eingefügt. Andernfalls wird er am Ende der Liste angefügt.

	Datentyp	Beschreibung
Parameter:		
⇒ index	Variant	Index der Position in der Liste, an der der Dateiname eingefügt werden soll (optional)
⇒ fileName	String	Name der Datei
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Void	

Clear

Methode von DataObjectFiles

Mit dieser Methode können Sie alle Dateinamen der Liste löschen.

	Datentyp	Beschreibung
Rückgabewert	Void	

Remove

Methode von DataObjectFiles

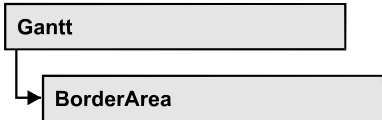
Mit dieser Methode können Sie den Dateinamen an dem angegebenen Index (Wertebereich 0 bis .Count-1) entfernen.

	Datentyp	Beschreibung
Parameter:		
⇒ index	Long	Index der Position in der Liste, von der der Dateiname gelöscht werden soll.

470 API-Referenz: DataObjectFiles

Rückgabewert	Void	
---------------------	------	--

7.4 VcBorderArea



Ein Objekt vom Typ **VcBorderArea** bezeichnet den Titel- bzw. Legendenbereich der Grafik.

Methoden

- **BorderBox**

Methoden

BorderBox

Methode von VcBorderArea

Diese Methode ermöglicht den Zugriff auf ein BorderBox-Objekt.

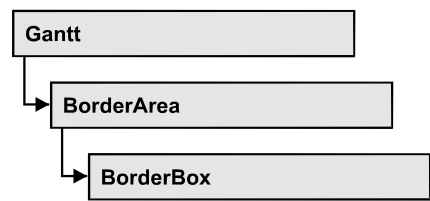
	Datentyp	Beschreibung
Parameter: boxPosition	BorderBoxPositionEnum Mögliche Werte: vcBBXPBottomBottomCentered 8 vcBBXPBottomBottomLeft 7 vcBBXPBottomBottomRight 9 vcBBXPBottomTopCentered 5 vcBBXPBottomTopLeft 4 vcBBXPBottomTopRight 6 vcBBXPLegend 51 vcBBXPTopCentered 2 vcBBXPTopLeft 1 vcBBXPTopRight 3	Position der Box zweite Zeile im unteren Bereich, mittig zweite Zeile im unteren Bereich, links zweite Zeile im unteren Bereich, rechts erste Zeile im unteren Bereich, mittig erste Zeile im unteren Bereich, links erste Zeile im unteren Bereich, rechts Legende oben mittig oben links oben rechts
Rückgabewert	VcBorderBox	Box des Titel- und Legendenbereichs

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
```

7.5 VcBorderBox



Ein Objekt vom Typ **VcBorderBox** bezeichnet eine der Boxen des Titel- bzw. Legendenbereichs der Grafik.

Eigenschaften

- Alignment
- GraphicsFileName
- LegendElementsArrangement
- LegendElementsBottomMargin
- LegendElementsMaximumColumnCount
- LegendElementsMaximumRowCount
- LegendElementsTopMargin
- LegendFont
- LegendTitle
- LegendTitleFont
- LegendTitleVisible
- Text
- TextFont
- Type

Eigenschaften

Alignment

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Ausrichtung dieses BorderBox-Objektes festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	BorderBoxAlignmentEnum	Ausrichtung der BorderBox
	Mögliche Werte: vcBBXACentered -1 vcBBXALeft -3	unten mittig links

vcBBXARight -2	rechts
----------------	--------

GraphicsFileName

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie den Namen der Grafikdatei angeben oder erfragen, die in dem aktuellen VcBoundingBox-Objekt verwendet wird. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile)
- *.WMF mit eingebautem EMF

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei
	Mögliche Werte:	Name der Farbzustimmungstabelle

Code-Beispiel

```

Dim borderArea As VcBorderArea
Dim bBoxTR As VcBoundingBox

Set borderArea = VcGantt1.BorderArea
Set bBoxTR = borderArea.BoundingBox(vcBBXPTopRight)
bBoxTR.Type = vcBBXTGraphics
bBoxTR.GraphicsFilename = "Asterix.jpg"

```

LegendElementsArrangement**Eigenschaft von VcBoundingBox**

Mit dieser Eigenschaft können Sie die Anordnung der Elemente der Legende setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	LegendElementsArrangementEnum	Typ der Anordnung der Legendenelemente
	Mögliche Werte:	
	vcLEAFixedToColumns 1	Die Legendenelemente werden nur in Spalten ausgerichtet.
	vcLEAFixedToRows 0	Die Legendenelemente werden nur in Zeilen ausgerichtet.
	vcLEAFixedToRowsAndColumns 2	Die Legendenelemente werden in Zeilen und Spalten ausgerichtet.

LegendElementsBottomMargin**Eigenschaft von VcBoundingBox**

Mit dieser Eigenschaft können Sie den Abstand der Elemente zum unteren Rand festlegen oder erfragen (Einheit: mm).

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des unteren Randes
	Mögliche Werte:	
		Datenfeldindex

LegendElementsMaximumColumnCount**Eigenschaft von VcBoundingBox**

Mit dieser Eigenschaft können Sie die Anzahl der Spalten setzen oder erfragen, über die sich die Elemente der Legende verteilen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Spalten
	Mögliche Werte:	Datenfeldindex

LegendElementsMaximumRowCount

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie die Anzahl der Zeilen setzen oder erfragen, über die sich die Elemente der Legende verteilen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Zeilen
	Mögliche Werte:	Datenfeldindex

LegendElementsTopMargin

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie den Abstand der Elemente zum oberen Rand festlegen oder erfragen (Einheit: mm).

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des oberen Randes
	Mögliche Werte:	Datenfeldindex

LegendFont

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie die Schriftattribute der Legende angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute der Legende

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTLegend
logThis (bBoxBBL.LegendFont.Name)
```

LegendTitle

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Legendentitel angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendentitel
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
```

LegendTitleFont

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Schriftattribute des Legendentitels angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute des Legendentitels

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTLegend
logThis (bBoxBBL.LegendTitleFont.Name)
```

LegendTitleVisible

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft legen Sie fest, ob der Legendentitel sichtbar ist oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Legendentitel sichtbar (True)/nicht sichtbar (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBoundingBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BoundingBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitleVisible = False
```

Text

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie den Text einer Titelzeile (oberhalb oder unterhalb des Diagramms) angeben oder erfragen. Für die Seitennummerierung oder die Ausgabe des Systemdatums können Sie folgende Platzhalter angeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{COLUMN} = Seitennummer in der Breite (einer zweidimensionalen Seitenanordnung)

{NUMPAGES} = Gesamtanzahl der Seiten

{PAGE} = fortlaufende Seitennummer

{ROW} = Seitennummer in der Höhe (einer zweidimensionalen Seitenanordnung)

{SYSTEMDATE} = Systemdatum

	Datentyp	Beschreibung
Parameter: rowIndex	Integer	Zeilenindex {0...6}

	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	String	Text in Textfeldern
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBoundingBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BoundingBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTText
bBoxBBL.Text(index) = "Department A"
```

TextFont**Eigenschaft von VcBoundingBox**

Mit dieser Eigenschaft können Sie die Schriftattribute einer Titelzeile (oberhalb oder unterhalb des Diagramms) angeben oder erfragen.

Dies ist eine indizierte Eigenschaft, die in C# über die beiden Methoden **set_TextFont (rowIndex, pvn)** und **get_TextFont (row-Index)** angesprochen wird.

	Datentyp	Beschreibung
Parameter: rowIndex	Integer Mögliche Werte:	Zeilenindex {0...6} Datenfeldindex
Eigenschaftswert	StdFont	Schriftattribute des Textes

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxTL As VcBoundingBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BoundingBox(vcBBXPBottomBottomLeft)

bBoxTL.TextFont(i).Bold = False
bBoxTL.TextFont(i).Italic = False
bBoxTL.TextFont(i).Name = "Symbol"
```

Code Sample in C#

```
/ Text for Title
VcBoundingBox borderBox =
VcGantt1.BorderArea.BoundingBox(VcBoundingBoxPosition.vcBBXPTopCentered);
borderBox.Type = VcBoundingBoxType.vcBBXTText;
```

```
Font titleFont1 = new Font("Arial", 20, FontStyle.Bold);

borderBox.set_Text(1, "Time Scheduler");
borderBox.set_TextFont(1, titleFont1);
```

Type

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Typ des BorderBox-Objekts angeben oder erfragen.

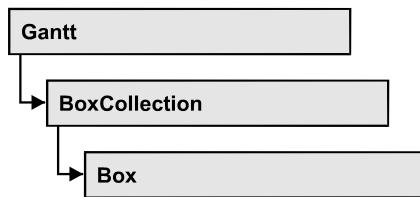
	Datentyp	Beschreibung
Eigenschaftswert	BorderBoxTypeEnum Mögliche Werte: vcBBXTGraphics 3 vcBBXTLegend 4 vcBBXTNothing 0 vcBBXTText 1 vcBBXTTextWithGraphics 2	Typ der Box Grafik Legende Leer Text Text und Grafik

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcGantt1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTGraphics
```

7.6 VcBox



Ein Objekt vom Typ **VcBox** beschreibt eine Box, in der Texte und Grafiken ausgegeben werden können.

Eigenschaften

- AnchoringInteractionsAllowed
- AnchoringLineVisible
- FieldText
- FormatName
- LineColor
- LineThickness
- LineType
- MarkBox
- Moveable
- Name
- NodeID
- Origin
- Priority
- ReferencePoint
- Resizing
- Specification
- UpdateBehaviorName
- Visible

Methoden

- AnchorToNode
- GetActualExtent
- GetTopLeftPixel
- GetXYOffset
- GetXYOffsetAsVariant
- IdentifyFormatField
- SetXYOffset
- SetXYOffsetByTopLeftPixel

Eigenschaften

AnchoringInteractionsAllowed

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob eine Box interaktiv mit einem Knoten verankert werden kann.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Box kann/kann nicht interaktiv mit einem Knoten verankert werden Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.AnchoringInteractionsAllowed = False
```

AnchoringLineVisible

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob bei einer Verankerung die eingestellten Referenzpunkt bei Knoten und Box durch eine Linie verbunden werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verankerungslinie zwischen Knoten und Box wird/wird nicht angezeigt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
```

```
box.AnchoringLineVisible = False
```

FieldText

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Inhalt eines Feldes einer Box erfragen oder festlegen. Diese Eigenschaft können Sie auch im Dialog **Box bearbeiten** festlegen.

Bei mehrzeiligen Textfeldern müssen für einen erzwungenen Umbruch die einzelnen Zeilen des Textfelds mit "\n" im String getrennt sein (Beispiel: "Zeile1\nZeile2"). Ohne erzwungenen Umbruch wird automatisch an Leerzeichen umgebrochen.

	Datentyp	Beschreibung
Parameter: ⇨ fieldIndex	Integer Mögliche Werte:	Feldindex Datenfeldindex
Eigenschaftswert	String Mögliche Werte:	Feldinhalt Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.boxCollection
Set box = boxCltn.FirstBox
box.FieldText(0) = "User: "
```

FormatName

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen des Boxformats erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxFormat	BoxFormat-Objekt oder Nothing

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox
```

```
boxCltn = VcGantt1.BoxCollection
box = boxCltn.FirstBox
box.FormatName = "Standard"
```

LineColor

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Farbe der Randlinie der Box festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255})

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineColor = RGB(255, 0, 0)
```

LineThickness

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Stärke der Randlinie der Box erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineThickness = 2
```

LineType

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Typ der Randlinie der Box festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp Standardwert: vcSolid
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100 vcLineType1 101 vcLineType10 110 vcLineType11 111 vcLineType12 112 vcLineType13 113 vcLineType14 114 vcLineType15 115	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0  Linientyp 1  Linientyp 10  Linientyp 11  Linientyp 12  Linientyp 13  Linientyp 14  Linientyp 15 

vcLineType16 116	Linientyp 16 -----
vcLineType17 117	Linientyp 17 -----
vcLineType18 118	Linientyp 18 -----
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3 -----
vcLineType4 104	Linientyp 4 -----
vcLineType5 105	Linientyp 5 -----
vcLineType6 106	Linientyp 6 -----
vcLineType7 107	Linientyp 7 -----
vcLineType8 108	Linientyp 8 -----
vcLineType9 109	Linientyp 9 -----
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineType = vcDotted
```

MarkBox**Eigenschaft von VcBox**

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob eine Box markiert ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	True: Box markiert; false: Box nicht markiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.MarkBox = True
```

Moveable

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Box interaktiv verschiebbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	verschiebbar (True)/ nicht verschiebbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Moveable = False
```

Name

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen einer Box erfragen oder setzen. Diese Eigenschaft können Sie im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Box
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox
Dim boxName As String

Set boxCltn = VcGantt1.boxCollection
Set box = boxCltn.FirstBox
boxName = box.Name
MsgBox boxName
```

NodeID

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die ID des Knotens mit dem die Box verankert ist, festlegen oder abfragen. Diese Eigenschaft können Sie auch im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	ID des Knotens, mit dem die Box verankert ist
	Mögliche Werte:	Name der Farbzustimmungstabelle

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox
Dim box.NodeID As String

Set boxCltn = VcGantt1.boxCollection
Set box = boxCltn.FirstBox
box.NodeID = 3
```

Origin

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Ursprungspunkt der Box festlegen oder erfragen, d. h. den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird.

Mithilfe der Eigenschaften **Origin** und **ReferencePoint** sowie der Methode **GetXyOffset** können Sie jede einzelne Box im Diagrammbereich positionieren, wobei die relative Position der Box zum Diagramm unabhängig von der Diagrammgröße ist.

	Datentyp	Beschreibung
Eigenschaftswert	BoxOriginEnum	Ursprungspunkt der Box
	Mögliche Werte: vcBOBottomCenter 28 vcBOBottomLeft 27 vcBOBottomRight 29 vcBOCenterCenter 25 vcBOCenterLeft 24 vcBOCenterRight 26 vcBOTopCenter 22 vcBOTopLeft 21 vcBOTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Origin = vcBOTopCenter
```

Priority**Eigenschaft von VcBox**

Mit dieser Eigenschaft können Sie die Priorität der Box erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Prioritätsstufe
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Priority = 3
```

ReferencePoint**Eigenschaft von VcBox**

Mit dieser Eigenschaft können Sie den Referenzpunkt der Box festlegen oder erfragen, d. h. den Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird.

	Datentyp	Beschreibung
Eigenschaftswert	BoxReferencePointEnum	Referenzpunkt der Box
	Mögliche Werte: vcBRPBottomCenter 28 vcBRPBottomLeft 27 vcBRPBottomRight 29 vcBRPCenterCenter 25 vcBRPCenterLeft 24 vcBRPCenterRight 26 vcBRPTopCenter 22 vcBRPTopLeft 21 vcBRPTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.ReferencePoint = vcBRPCenterRight

```

Resizing**Eigenschaft von VcBox**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob und wie die Größe einer Box verändert werden kann.

	Datentyp	Beschreibung
Eigenschaftswert	BoxResizingEnum Mögliche Werte: vcBRHeight 23 vcBRNo 0 vcBRWidth 24 vcBRWidth/Height 1050	Interaktive Größenänderung der Box Die Höhe der Box kann interaktiv verändert werden. Die Größe der Box kann nicht interaktiv verändert werden. Die Breite der Box kann interaktiv verändert werden. Breite und Höhe der Box können interaktiv verändert werden.

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Resizing = vcBRWidth

```

Specification**Nur-Lese-Eigenschaft von VcBox**

Mit dieser Eigenschaft können Sie die Spezifikation dieser Box auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung einer Box mit der Methode **VcBoxCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Boxspezifikation Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
MsgBox box.Specification
```

UpdateBehaviorName

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Visible

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Box sichtbar sein soll. Diese Eigenschaft können Sie auch im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Box sichtbar/unsichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.FirstBox
box.Visible = False
```

Methoden

AnchorToNode

Methode von VcBox

Mit dieser Methode können Sie Boxen an Knoten verankern bzw. sie wieder lösen. Um eine Box wieder vom Knoten zu lösen, müssen Sie als Parameter "NULL" übergeben.

Eine verankerte Box kann weiterhin interaktiv verschoben werden (Voraussetzung: die Eigenschaft **Moveable** ist gesetzt).

Wird ein Knoten mit einer verankerten Box verschoben, so wird die Box entsprechend mitverschoben. Wird der Knoten kollabiert, wird auch die Box kollabiert d.h. sie ist nicht mehr sichtbar. Sobald der Knoten expandiert wird, ist auch die Box wieder sichtbar.

Beim Ver-/Entankern bleibt die Position der Box auf dem Bildschirm erhalten - die zu Grunde liegenden Offset-Werte werden entsprechend der Referenzpunkte (Origin, ReferencePoint) umgerechnet. Wenn sich also eine Textbox mit einem bestimmten Offset z.B. auf das Diagramm oben links (Origin) bezieht und dann mit einem Knoten verankert wird, wird automatisch ein Offset zum Knoten oben links berechnet und dieser Offset führt dazu, dass die Position auf dem Bildschirm beibehalten wird. Wird die Box vom Knoten gelöst, erfolgt eine Rückberechnung

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knotenobjekt, mit dem die Box verankert ist
Rückgabewert	Boolean	Box wird am Knoten verankert/vom Knoten gelöst

GetActualExtent

Methode von VcBox

Mit dieser Methode können Sie die Breite und Höhe der Box erfragen (Einheit: 1/100 mm).

Werden diese Werte beim XY-Offset berücksichtigt, kann man z.B. den Referenzpunkt der Verankerungslinie ändern, ohne die Position der Box zu verändern.

	Datentyp	Beschreibung
Parameter:		
↔ width	Integer	Breite der Box
	Mögliche Werte:	Datenfeldindex
↔ height	Integer	Höhe der Box
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Ausdehnung der Box wird zurückgegeben/wird nicht zurückgegeben

GetTopLeftPixel

Methode von VcBox

Mit dieser Methode können Sie den gespeicherten XY-Offset für die obere linke Ecke in Pixel umrechnen und ausgeben.

Der x-Wert kann dann z.B. mit der Methode **VcGantt.GetDate** weiter verwendet werden, um ein Datum zu erhalten

	Datentyp	Beschreibung
Parameter:		
↔ x	Integer	x-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
↔ y	Integer	y-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Offset wird zurückgegeben/nicht zurückgegeben

GetXYOffset

Methode von VcBox

Mit dieser Methode können Sie den Abstand zwischen Ursprung und Referenzpunkt in x- und y-Richtung erfragen (Einheit: 1/100 mm).

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetXYOffsetAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
↔ xOffset	Integer	x-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
↔ yOffset	Integer	y-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Offset wird zurückgegeben/nicht zurückgegeben

GetXYOffsetAsVariant

Methode von VcBox

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetXYOffset**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↔) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

IdentifyFormatField

Methode von VcBox

Mit dieser Methode können Sie den Index des an der bezeichneten Position befindlichen Formatfeldes erfragen. Falls sich an der bezeichneten Position ein Feld befindet, wird **True** zurückgegeben, ansonsten **False**.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate der Position
⇒ y	Long	Y-Koordinate der Position

⇐ format	VcBoxFormat	Identifiziertes Format
⇐ formatFieldIndex	Integer	Format-Feldindex
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Ein Formatfeld befindet sich/befindet sich nicht an der angegebenen Position

SetXYOffset

Methode von VcBox

Mit dieser Methode können Sie den Abstand zwischen Ursprung und Referenzpunkt in x- und y-Richtung festlegen (Einheit: 1/100 mm).

Den Offset können Sie auch im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Parameter:		
⇒ xOffset	Integer	x-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
⇒ yOffset	Integer	y-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Offset wird gesetzt (True)/nicht gesetzt (False)

Code-Beispiel

```
Dim OffsetSet As Boolean
OffsetSet = VcGantt1.boxCollection.FirstBox.SetXYOffset(100, 100)
```

SetXYOffsetByTopLeftPixel

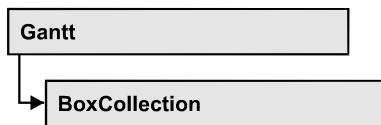
Methode von VcBox

Mit dieser Methode können Sie den angegebenen Pixelwert der oberen linken Ecke intern in einen XY-Offset umrechnen und speichern.

Damit kann man z.B. an einer XY-Koordinate aus einem Ereignis eine Box positionieren.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Integer	x-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
⇒ y	Integer	y-Wert des Offsets
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Boolean	Offset wird gesetzt (True)/nicht gesetzt (False)

7.7 VcBoxCollection



In einem Objekt des Typs VcBoxCollection sind alle verfügbaren Boxen zusammengefasst. Über **For Each box In BoxCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Boxen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **BoxByName** und **BoxByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Boxen kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Boxen.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- BoxByIndex
- BoxByName
- Copy
- FirstBox
- NextBox
- Remove
- Update

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcBoxCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Box-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim box As VcBox

For Each box In VcGantt1.BoxCollection
    Debug.Print box.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcBoxCollection

Mit dieser Eigenschaft können Sie die Anzahl der Boxobjekte in der Box-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Boxen

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim numberOfBoxes As Long

Set boxCltn = VcGantt1.BoxCollection
Dim numberOfBoxes = boxCltn.Count
```

Methoden

Add

Methode von VcBoxCollection

Mit dieser Methode können Sie eine neue Box in der BoxCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Boxobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die neu angelegte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ boxName	String Mögliche Werte:	Name der Box Name der Farbzusordnungstabelle
Rückgabewert	VcBox	Neues Boxobjekt

Code-Beispiel

```
Set newBox = VcGantt1.BoxCollection.Add("box1")
```

AddBySpecification**Methode von VcBoxCollection**

Mit dieser Methode können Sie eine Box über eine Box-Spezifikation erzeugen. Dies dient der Persistenz von Box-Objekten. Die Spezifikation einer Box kann erfragt (siehe VcBox-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Box mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden. Um die neu angelegte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Boxspezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcBox	Neues Boxobjekt

BoxByIndex**Methode von VcBoxCollection**

Mit dieser Methode können Sie auf eine einzelne Box über ihren Index zugreifen. Existiert keine Box unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Box

		Datenfeldindex
Rückgabewert	VcBox	Ermitteltes Boxobjekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(2)
box.LineThickness = 2
```

BoxByName**Methode von VcBoxCollection**

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Box zugreifen. Existiert keine Box unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ boxName	String	Name der Box
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcBox	Box

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByName("Box 1")
```

Copy**Methode von VcBoxCollection**

Mit dieser Methode können Sie eine Box kopieren. Wenn die Box mit dem angegebenen Namen existiert und der Name der neuen Box noch nicht verwendet wird, wird das neue Boxobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die kopierte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ boxName	String	Name der zu kopierenden Box
		Mögliche Werte: Name der Farbzusordnungstabelle
	String	Name der neuen Box
		Mögliche Werte: Name der Farbzusordnungstabelle
Rückgabewert	VcBox	Boxobjekt

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.Copy("BoxOne", "NewBox")
boxCltn.Update

```

FirstBox**Methode von VcBoxCollection**

Mit dieser Methode können Sie auf die erste Box der Box-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextBox** über die nachfolgenden Boxen zu iterieren. Existiert keine Box in der Box-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBox	erste Box

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.FirstBox

```

NextBox**Methode von VcBoxCollection**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Boxen der Box-Auflistung zugreifen, nachdem Sie mit der Methode **FirstBox** den Initialwert erfasst haben. Sind alle Boxen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBox	Nachfolgendes Box-Objekt

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.FirstBox

While Not box Is Nothing
    Listbox.AddItem box.Name
    Set box = boxCltn.NextBox
Wend

```

Remove**Methode von VcBoxCollection**

Mit dieser Methode können Sie eine Box löschen. Um das Löschen der Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ boxName	String	Name der Box
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Box gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```

Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(2)
boxCltn.Remove (box.Name)
boxCltn.Update

```

Update**Methode von VcBoxCollection**

Mit dieser Methode können Sie eine BoxCollection aktualisieren, nachdem Sie sie verändert haben.

502 API-Referenz: VcBoxCollection

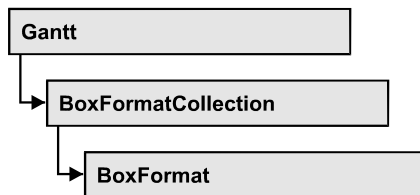
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.BoxByIndex(2)
boxCltn.Remove (box.Name)
boxCltn.Update
```

7.8 VcBoxFormat



Ein Objekt vom Typ **VcBoxFormat** beschreibt die Formate von Boxen.

Eigenschaften

- `_NewEnum`
- `FieldsSeparatedByLines`
- `FormatField`
- `FormatFieldCount`
- `Name`
- `Specification`

Methoden

- `CopyFormatField`
- `RemoveFormatField`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von VcBoxFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Boxformatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim formatField As VcBoxFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

FieldsSeparatedByLines**Eigenschaft von VcBoxFormat**

Mit dieser Eigenschaft können Sie festlegen, ob die Felder durch sichtbare Linien getrennt werden (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Boxfelder werden durch Linien getrennt (True)/ nicht getrennt (False).
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxFormat As VcBoxFormat

Set boxFormat = VcGantt1.BoxFormatCollection.FormatByIndex(2)
boxFormat.FieldsSeparatedByLines = True
```

FormatField**Nur-Lese-Eigenschaft von VcBoxFormat**

Mit dieser Eigenschaft können Sie ein VcBoxFormatField-Objekt per Index holen. Der Index muss im Bereich von 0 bis .FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht (wie in den Feldeigenschaften vorher) von 1 bis .FormatFieldCount!

	Datentyp	Beschreibung
Parameter: index	Integer	Index des Boxformatfeldes 0FormatFieldCount-1
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	VcBoxFormatField	Boxformatfeld

Code-Beispiel

```

Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcGantt1.BoxFormatCollection.FirstFormat
Set formatField = boxFormat.formatField(0)
MsgBox formatField.FormatName

```

FormatFieldCount**Nur-Lese-Eigenschaft von VcBoxFormat**

Mit dieser Eigenschaft können Sie die Anzahl der Felder eines Boxformats ermitteln.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Felder im Boxformat
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```

Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcGantt1.BoxFormatCollection.FirstFormat
MsgBox boxFormat.FormatFieldCount

```

Name**Eigenschaft von VcBoxFormat**

Mit dieser Eigenschaft können Sie den Namen eines Boxformats erfragen oder setzen. Diese Eigenschaft können Sie auch im Dialog **Boxformate verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Boxformats
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim boxFormat As VcBoxFormat

For Each boxFormat In VcGantt1.BoxFormatCollection
    List1.AddItem (boxFormat.Name)
Next

```

Specification

Nur-Lese-Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Boxformats auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Boxformats mit der Methode **VcBoxFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Boxformats
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Methoden

CopyFormatField

Methode von VcBoxFormat

Mit dieser Methode können Sie ein Boxformatfeld kopieren. Das neue VcBoxFormatField-Objekt wird zurückgegeben. Es erhält den nächsten, noch nicht vergebenen Index.

	Datentyp	Beschreibung
Parameter: ⇒ position	FormatFieldInnerPositionEnum	Position des neuen Boxformatfeldes
	Mögliche Werte: vcInnerAbove 1 vcInnerBelow 3 vcInnerLeftOf 0 vcInnerRightOf 4	oberhalb unterhalb links von rechts von
⇒ refIndex	Integer	Index des Referenz-Boxformatfeldes
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcBoxFormatField	Boxformatfeld-Objekt

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField
```

```
Set boxFormat = VcGantt1.BoxFormatCollection.FormatByIndex(2)
Set formatField = boxFormat.CopyFormatField(vcInnerRightOf, 0)
```

RemoveFormatField

Methode von VcBoxFormat

Mit dieser Methode können Sie ein Boxformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Boxformatfelder neu festgesetzt, so dass sie wieder fortlaufend numeriert sind.

	Datentyp	Beschreibung
Parameter:		
⇒ index	Integer	Index des zu löschenden Boxformatfeldes
	Mögliche Werte:	Datenfeldindex

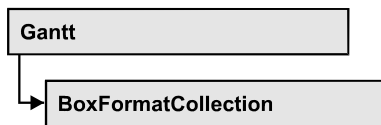
Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim i As Integer

boxFormat = VcGantt1.BoxFormatCollection.FirstFormat

For i = 0 To boxFormat.FormatFieldCount - 1
    boxFormat.RemoveFormatField (i)
Next
```


7.9 VcBoxFormatCollection



Im VcBoxFormat-Auflistungsobjekt sind alle verfügbaren Boxformate zusammengefasst. Über **For Each boxFormat In BoxFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Formate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **BoxFormatByName** und **BoxFormatByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Boxformate kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Boxformaten.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstFormat
- FormatByIndex
- FormatByName
- NextFormat
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcBoxFormatCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Boxformatobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim format As VcBoxFormat

For Each format In VcGantt1.BoxCollection
    Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcBoxFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Boxformatobjekte in der BoxFormat-Auflistung abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Boxformate

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim numberOfBoxformats As Long

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Dim numberOfBoxformats = boxFormatCltn.Count
```

Methoden

Add

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein neues Boxformat in der BoxFormatCollection anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Boxformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String	Name des Boxformats

	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcBoxFormat	Neues Boxformatobjekt

Code-Beispiel

```
Set newBoxFormat = VcGantt1.BoxFormatCollection.Add("boxFormat1")
```

AddBySpecification**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie ein Boxformat über eine Boxformat-Spezifikation erzeugen. Dies dient der Persistenz von Boxformatobjekten. Die Spezifikation eines Boxformats kann erfragt (siehe VcBoxFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Boxformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ formatSpecification	String Mögliche Werte:	Boxformatspezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcBoxFormat	Neues Boxformatobjekt

Copy**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie ein Boxformat kopieren. Wenn das Boxformat mit dem angegebenen Namen existiert und der Name des neuen Boxformats noch nicht verwendet wird, wird das neue Boxformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String Mögliche Werte:	Name des zu kopierenden Boxformats Name der Farbzuoordnungstabelle
⇒ newFormatName	String	Name des neuen Boxformats

	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcBoxFormat	Boxformatobjekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormat = boxFormatCltn.Copy("CurrentBoxFormat", "NewBoxFormat")
```

FirstFormat**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie auf das erste Boxformat der BoxFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Boxformate zu iterieren. Existiert kein Boxformat in der BoxFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBoxFormat	erstes Boxformat

Code-Beispiel

```
Dim format As VcBoxFormat

Set format = VcGantt1.BoxFormatCollection.FirstFormat
```

FormatByIndex**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie auf ein einzelnes Boxformat über seinen Index zugreifen. Existiert kein Boxformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Boxformats
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcBoxFormat	Ermitteltes Boxformat-Objekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set format = boxFormatCltn.FormatByIndex(2)
```

FormatByName**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Boxformat zugreifen. Existiert kein Boxformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ formatName	String	Name des Boxformats
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcBoxFormat	Boxformat

Code-Beispiel

```
Dim formatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set formatCltn = VcGantt1.BoxFormatCollection
Set format = formatCltn.FormatByName("Standard")
```

NextFormat**Methode von VcBoxFormatCollection**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Boxformate der BoxFormat-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFormat** den Initialwert erfasst haben. Sind alle Formate durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBoxFormat	Nachfolgendes Boxformat

Code-Beispiel

```
Dim formatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set formatCltn = VcGantt1.BoxFormatCollection
Set format = formatCltn.FirstFormat
```

```

While Not format Is Nothing
    List1.AddItem format.Name
    Set format = formatCltn.NextFormat
Wend

```

Remove

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein Boxformat löschen. Wenn das Boxformat noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String Mögliche Werte:	Name des Boxformats Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Boxformat gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

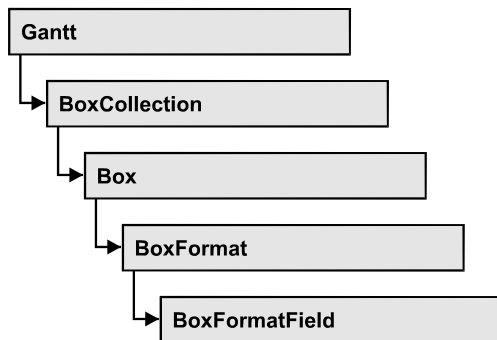
```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormat = boxFormatCltn.FormatByIndex(1)
boxFormatCltn.Remove (boxFormat.Name)

```

7.10 VcBoxFormatField



Ein Objekt vom Typ **VcBoxFormatField** stellt ein Boxformatfeld, also ein Feld eines VcBoxFormat-Objekts dar. Ein Boxformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Boxformat untergebracht ist.

Eigenschaften

- Alignment
- FormatName
- GraphicsHeight
- Index
- MaximumTextLineCount
- MinimumTextLineCount
- MinimumWidth
- PatternBackgroundColorAsARGB
- PatternColorAsARGB
- PatternEx
- TextFont
- TextFontColor
- Type

Eigenschaften

Alignment

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Boxformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum	Ausrichtung des Feldinhalts
	Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	 unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.Alignment = vcFFACenter

```

FormatName**Nur-Lese-Eigenschaft von VcBoxFormatField**

Mit dieser Eigenschaft können Sie den Namen des Boxformats erfragen, zu dem dieses Boxformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Boxformats
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
MsgBox boxFormatField.FormatName

```

GraphicsHeight**Eigenschaft von VcBoxFormatField**

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** die Höhe der Grafik in dem Boxformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe der Grafik in mm
	Mögliche Werte:	0 ... 200 Datenfeldindex

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTGraphics
boxFormatField.GraphicsHeight = 150

```

Index**Nur-Lese-Eigenschaft von VcBoxFormatField**

Mit dieser Eigenschaft können Sie den Index des Boxformatfelds im zugehörigen Boxformat erfragen.

	Datentyp	Beschreibung

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
MsgBox boxFormatField.Index

```

MaximumTextLineCount**Eigenschaft von VcBoxFormatField**

Mit dieser Eigenschaft können Sie die maximale Anzahl der Zeilen in dem Boxformatfeld setzen oder erfragen, falls das Boxformatfeld vom Typ **vcFFTText** ist. Bitte sehen Sie auch die Eigenschaft **MinimumTextLineCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Maximale Zeilenzahl
	Mögliche Werte:	0 ... 20

Datenfeldindex

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTText
boxFormatField.MaximumTextLineCount = 5
```

MinimumTextLineCount

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die minimale Anzahl der Zeilen in dem Boxformatfeld setzen oder erfragen, falls der Typ des Boxformatfeldes auf **vcFFTText** gesetzt wurde. Ist in einem Knoten mehr Text vorhanden, als in die minimale Anzahl der Zeilen hineinpasst, wird dieses Feld für diesen Knoten dynamisch bis zur maximalen angegebenen Anzahl der Zeilen ausgedehnt. Wenn Sie dieser Eigenschaft einen Wert zuweisen, sollten Sie anschließend auch erneut der Eigenschaft **MaximumTextLineCount** den gewünschten Wert setzen, sonst könnte es vorkommen, dass das Maximum durch das Minimum überschrieben wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Zeilenzahl
	Mögliche Werte:	0 ... 20
		Datenfeldindex

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTText
boxFormatField.MinimumTextLineCount = 3
```

MinimumWidth

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Boxformatfeldes in mm festlegen oder erfragen. Die Breite des Feldes kann sich vergrößern,

wenn unter oder über dem Feld andere Felder größere minimale Breiten besitzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Minimale Breite des Boxformatfeldes
		0 ... 200
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.MinimumWidth = 100
```

PatternBackgroundColorAsARGB

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Boxformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Boxformats besitzen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Hintergrundfarbe des Boxformatfelds
		Standardwert: -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.BackColor = RGB(0, 255, 0)
```

PatternColorAsARGB

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Boxformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Boxformats besitzen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Boxformatfeldes

Code-Beispiel

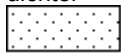
```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

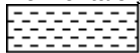
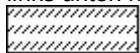
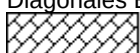
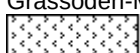
Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.PatternColor = RGB(0, 255, 0)
```

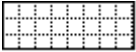
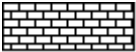
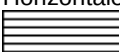
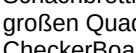
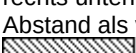
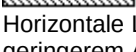
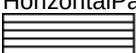
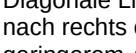
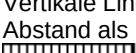
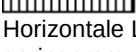
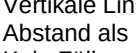
PatternEx

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Boxformatfeldes ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Mustertyp Standardwert: Wie im Dialog definiert Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 

vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
vcCrossPattern 6	Kreuzschraffur 
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 

vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern 
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 

vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

TextFont

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Boxformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Boxformatfeldes

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.TextFont.Bold = True
```

TextFontColor

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Boxformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Boxformatfeldes Standardwert: -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.TextFontColor = RGB(0, 255, 0)
```

Type

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie den Typ des Boxformatfeldes erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldTypeEnum	Typ des Boxformatfeldes
	Mögliche Werte: vcFFTGraphics 64	Grafik

524 API-Referenz: VcBoxFormatField

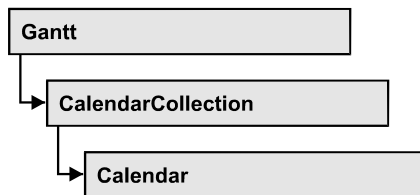
	vcFFText 36	Text
--	-------------	------

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTextGraphics
boxFormatField.GraphicsHeight = 200
```

7.11 VcCalendar



Ein Kalender dient der Definition von Arbeits- und Nichtarbeitszeiten. Er wird durch eine lückenlose Aneinanderreihung von Arbeitszeiten und Nichtarbeitszeiten zusammengesetzt. Diese werden in der Regel durch Arbeitstage und Arbeitswochen (Workday- bzw. Workweek-Objekte) dargestellt, aber auch Intervalle sind möglich.

Ein neu angelegter Kalender enthält standardmäßig ein Arbeitszeitintervall, das den Projektzeitraum umfaßt. Sie können einen Kalender für Zeitberechnungen verwenden, z.B. um den Zeitunterschied zwischen zwei verschiedenen Terminen in Arbeitstagen zu ermitteln. Durch den Kalender übernehmen die Objekte des Gantt-Graphen wie Kalendergitter, Balken und Layer die festgelegten Zeitintervalle. So können Balken durch arbeitsfreie Intervalle unterbrochen dargestellt werden. Sie können einen Kalender für Zeitberechnungen verwenden, z.B. um den Zeitunterschied zwischen zwei verschiedenen Terminen in Arbeitstagen zu ermitteln.

Eigenschaften

- CalendarProfileCollection
- IntervalCollection
- Name
- SecondsPerWorkday
- Specification
- Type

Methoden

- AddDuration
- CalcDuration
- Clear
- GetEndOfPreviousWorktime
- GetNextIntervalBorder
- GetPreviousIntervalBorder
- GetStartOfInterval
- GetStartOfNextWorktime
- IsWorktime

- Update

Eigenschaften

CalendarProfileCollection

Nur-Lese-Eigenschaft von VcCalendar

Mit dieser Eigenschaft haben Sie Zugriff auf die Kalenderprofilauflistung, in der alle zur Verfügung stehenden Kalenderprofile dieses Kalenderobjektes enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcCalendarProfileCollection	CalendarProfileCollection-Objekt

IntervalCollection

Nur-Lese-Eigenschaft von VcCalendar

Mit dieser Eigenschaft haben Sie Zugriff auf die Intervallauflistung, in der alle zur Verfügung stehenden Intervalle enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcIntervalCollection	IntervalCollection-Objekt

Name

Nur-Lese-Eigenschaft von VcCalendar

Mit dieser Eigenschaft können Sie den Namen eines Kalenders erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalenders
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim calendar As VcCalendar
Dim calendarName As String

Set calendar = VcGantt1.CalendarCollection.FirstCalendar
```

```
calendarName = calendar.Name
```

SecondsPerWorkday

Eigenschaft von VcCalendar

Mit dieser Eigenschaft lässt sich die Anzahl der Sekunden eines Arbeitstag setzen oder erfragen. Die Option kann auch im Dialog **Kalender festlegen** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	"Sekunden eines Arbeitstages"

Specification

Nur-Lese-Eigenschaft von VcCalendar

Mit dieser Eigenschaft können Sie die Spezifikation dieses Kalenders erfragen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Kalenders mit der Methode **VcCalendarCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Spezifikation des Kalenders Name der Farbzusordnungstabelle

Type

Eigenschaft von VcCalendar

Mit dieser Eigenschaft können Sie den Kalendertyp setzen oder erfragen. Wenn Sie den Typ ändern, gehen alle für den Kalender gesetzten Eigenschaften verloren.

	Datentyp	Beschreibung
Eigenschaftswert	CalendarTypeEnum Mögliche Werte: vcNormalCalendar 139	Kalendertyp

vcShiftCalendar 12

Code-Beispiel

```

Dim calendarCltn As VcCalendarCollection
Dim calendar As VcCalendar

Set calendarCltn = VcGantt1.CalendarCollection
Set calendar = calendarCltn.CalendarByIndex(2)
calendar.Type = vcNormalCalendar

```

Methoden

AddDuration

Methode von VcCalendar

Mit dieser Methode können Sie zu einem Termin eine Dauer addieren, um so unter Berücksichtigung der Kalenderdefinition einen neuen Termin zu berechnen. Wenn Sie beispielsweise zu einem Freitag die Dauer von drei Arbeitstagen hinzufügen, würde der neue Termin - bei arbeitsfreien Wochenenden - den darauffolgenden Mittwoch ergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ Date	Date/Time	Datum, an dem die Dauer eingefügt werden soll
⇒ Duration	Long	Anzahl der Zeiteinheiten (z.B. Tage) der Dauer
Rückgabewert	Date/Time	Datum, an dem die Dauer eingefügt wurde

Code-Beispiel

```

Dim calendar As VcCalendar
Dim newDate As Date

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
newDate = calendar.AddDuration("16.06.2014", 3)

```

CalcDuration

Methode von VcCalendar

Mit dieser Methode können Sie die Anzahl der Arbeitszeitelemente (z. B. Arbeitstage) berechnen, die zwischen zwei Terminen liegen. Die Einheit (z. B. Tage) des berechneten Wertes entspricht der auf der Eigenschaftenseite **Allgemeines** festgelegten Zeiteinheit im gleichnamigen Feld.

	Datentyp	Beschreibung
Parameter:		
⇒ fromDate	Date/Time	Anfangsdatum der Zeitdauer, aus der die Anzahl der Arbeitszeitelemente ermittelt werden soll
⇒ toDate	Date/Time	Enddatum der Zeitdauer, aus der die Anzahl der Arbeitszeitelemente ermittelt werden soll
Rückgabewert	Long	Anzahl der Zeiteinheiten (z.B. Tage) der Dauer

Code-Beispiel

```
Dim calendar As VcCalendar
Dim duration As Long
```

```
Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
duration = calendar.CalcDuration("01.01.2013", "31.12.2014")
```

Clear**Methode von VcCalendar**

Diese Methode löscht alle Profile und Intervalle aus dem Kalender, wodurch er komplett geleert wird (=>100% Arbeitszeit). Damit sich die Änderungen auch optisch zeigen (also z.B. bei Kalendergittern), muss noch ein Update aufgerufen werden.

	Datentyp	Beschreibung

GetEndOfPreviousWorktime**Methode von VcCalendar**

Mit dieser Methode können Sie das Ende der dem Referenzdatum vorangehenden Arbeitsperiode erfragen. Dabei muss das Referenzdatum in einer Nichtarbeitszeit liegen.

	Datentyp	Beschreibung
Parameter:		
⇒ Date	Date/Time	Referenzdatum, zu dem die vorangehende Arbeitszeit ermittelt werden soll
Rückgabewert	Date/Time	Enddatum der vorangegangenen Arbeitszeit

Code-Beispiel

```
Dim calendar As VcCalendar
Dim endOfWork As Date
```

```
Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
endOfWork = calendar.GetEndOfPreviousWorktime("18.06.2014")
```

GetNextIntervalBorder

Methode von VcCalendar

Mit dieser Methode können Sie den Anfangstermin des dem Referenzdatum nachfolgenden Intervalls erfragen. Wenn das Intervall, in dem das Referenzdatum liegt, eine Nichtarbeitszeit war, ist das zurückgegebene Datum der Anfang der folgenden Arbeitszeit, und umgekehrt.

	Datentyp	Beschreibung
Parameter: ⇒ Date	Date/Time	Referenzdatum, zu dem das Anfangsdatum der folgenden Intervallgrenze ermittelt werden soll
Rückgabewert	Date/Time	Anfangsdatum der folgenden Intervallgrenze

Code-Beispiel

```
Dim calendar As VcCalendar
Dim nextIntervalBorder As Date

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
nextIntervalBorder = calendar.GetNextIntervalBorder("18.06.2014")
```

GetPreviousIntervalBorder

Methode von VcCalendar

Mit dieser Methode können Sie den Endtermin des dem Referenzdatum vorangehenden Intervalls erfragen. Wenn das Intervall, in dem das Referenzdatum liegt, eine Nichtarbeitszeit war, ist das zurückgegebene Datum das Ende der vorausgehenden Arbeitszeit, und umgekehrt.

	Datentyp	Beschreibung
Parameter: ⇒ Date	Date/Time	Referenzdatum, zu dem das Enddatum der vorausgehenden Intervallgrenze ermittelt werden soll
Rückgabewert	Date/Time	Enddatum der vorausgehenden Intervallgrenze

Code-Beispiel

```
Dim calendar As VcCalendar
Dim previousIntervalBorder As Date

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
previousIntervalBorder = calendar.GetPreviousIntervalBorder("18.06.2014")
```

GetStartOfInterval

Methode von VcCalendar

Mit dieser Methode können Sie den Anfang des Intervalls erfragen, in dem das Referenzdatum liegt.

	Datentyp	Beschreibung
Parameter: ⇒ Date	Date/Time	Referenzdatum des Intervalls, zu dem das Anfangsdatum ermittelt werden soll
Rückgabewert	Date/Time	Anfangsdatum des Intervalls

Code-Beispiel

```
Dim calendar As VcCalendar
Dim startOfInterval As Date

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
startOfInterval = calendar.GetStartOfInterval("18.06.2014")
```

GetStartOfNextWorktime

Methode von VcCalendar

Mit dieser Methode können Sie den Anfang der dem Referenzdatum nachfolgenden Arbeitszeit erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ Date	Date/Time	Referenzdatum, zu dem das Anfangsdatum der folgenden Arbeitszeit ermittelt werden soll
Rückgabewert	Date/Time	Anfangsdatum der folgenden Arbeitszeit

Code-Beispiel

```
Dim calendar As VcCalendar
Dim startOfNextWorktime As Date

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
startOfNextWorktime = calendar.GetStartOfNextWorktime("18.06.2014")
```

IsWorktime

Methode von VcCalendar

Mit dieser Methode können Sie erfragen, ob sich das übergebene Datum in einer Arbeitszeit befindet.

532 API-Referenz: VcCalendar

	Datentyp	Beschreibung
Parameter: ⇒ Date	Date/Time	Datum, das darauf geprüft werden soll, ob es in einer Arbeitszeit liegt
Rückgabewert	Boolean	Übergebenes Datum fällt/fällt nicht in eine Arbeitszeit

Code-Beispiel

```
Dim calendar As VcCalendar
Dim isWorktime As Boolean

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
isWorktime = calendar.isWorktime("18.06.2014")
```

Update

Methode von VcCalendar

Mit dieser Methode können Sie einen Kalender aktualisieren, nachdem Sie ihn verändert haben. Diese Methode stellt sicher, dass alle Objekte, die den Kalender verwenden (z. B. das Raster), ebenfalls aktualisiert werden.

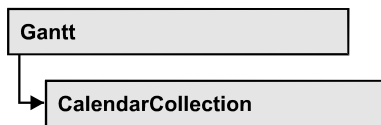
	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
Dim calendar As VcCalendar

Set calendar = VcGantt1.CalendarCollection.CalendarByName("WeekCalendar")
calendar.Update
```

7.12 VcCalendarCollection



In einem Objekt des Typs **VcCalendarCollection** sind alle mit der Methode **CreateCalendar** angelegten Kalender zusammengefasst. Über **For Each calendar In CalendarCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Kalender zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **CalendarByName** und **CalendarByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Kalender kann über die Eigenschaft **Count** erfragt werden. Mit der Eigenschaft **Active** können Sie den dem Kalendergitter zu Grunde liegenden Kalender erfragen.

Eigenschaften

- **_NewEnum**
- **Active**
- **Count**

Methoden

- **Add**
- **AddBySpecification**
- **CalendarByIndex**
- **CalendarByName**
- **Copy**
- **FirstCalendar**
- **NextCalendar**
- **Remove**
- **Update**

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcCalendarCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface **IEnumVariant** implementiert. Mittels dieses Objekts kann man über alle ent-

haltenen Kalenderobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim calendar As VcCalendar

For Each calendar In VcGantt1.CalendarCollection
    Debug.Print calendar.Name
Next
```

Active

Eigenschaft von VcCalendarCollection

Mit dieser Eigenschaft kann der aktuelle Standardkalender erfragt oder gesetzt werden, der für Vorgänge verwendet wird, wenn kein anderer Kalender zugewiesen wurde.

	Datentyp	Beschreibung
Eigenschaftswert	VcCalendar	Aktuell verwendeter Kalender

Code-Beispiel

```
Dim workday As VcWorkday
Dim freeday As VcWorkday
Dim workweek As VcWorkweek
Dim calendarCltn As VcCalendarCollection
Dim calendar As VcCalendar

Set workday = VcGantt1.WorkdayCollection.CreateWorkday("Work day")
workday.AddNonWorkInterval "00:00:00", "00:00:00"
workday.AddWorkInterval "08:00:00", "16:30:00"

Set freeday = VcGantt1.WorkdayCollection.CreateWorkday("Workfree day")
freeday.AddNonWorkInterval "00:00:00", "00:00:00"

Set calendarCltn = VcGantt1.calendarcollection
Set calendar = calendarCltn.AddCalendar("New Calendar")

Set workweek = VcGantt1.WorkweekCollection.CreateWorkweek("Work week")
workweek.AddWorkday workday, vcMonday, vcFriday
workweek.AddWorkday freeday, vcSaturday, vcSunday

calendar.AddWorkweek workweek, "01.01.13", "31.12.14"

calendar.Update

Set calendarCltn.Active = calendar
```

Count

Nur-Lese-Eigenschaft von VcCalendarCollection

Mit dieser Eigenschaft können Sie die Anzahl der Kalenderobjekte in der Kalender-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Kalender

Methoden

Add

Methode von VcCalendarCollection

Mit dieser Methode können Sie einen neuen Kalender in der FilterCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Kalenderobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇨ calendarName	String	Name des Kalenders
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcCalendar	Neues Kalenderobjekt

AddBySpecification

Methode von VcCalendarCollection

Mit dieser Methode können Sie einen Kalender über eine Kalender-Spezifikation erzeugen. Dies dient der Persistenz von Kalenderobjekten. Die Spezifikation eines Kalenderobjektes kann erfragt (siehe VcCalendar-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann der gleiche Kalender mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Kalenderspezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcCalendar	Neues Kalenderobjekt

CalendarByIndex

Methode von VcCalendarCollection

Mit dieser Methode können Sie auf ein einzelnes Kalender über seinen Index zugreifen. Existiert kein Kalender unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Kalenders Datenfeldindex
Rückgabewert	VcCalendar	Ermitteltes Kalenderobjekt

CalendarByName

Methode von VcCalendarCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf einen bestimmten Kalender zugreifen. Existiert kein Kalender unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ CalendarName	String Mögliche Werte:	Name des Kalenders Name der Farbzusordnungstabelle
Rückgabewert	VcCalendar	Kalender

Code-Beispiel

```
Dim calendarCltn As VcCalendarCollection

Set calendarCltn = VcGantt1.CalendarCollection
calendarCltn.Active = calendarCollection.CalendarByName("Calendar_1")
```

Copy**Methode von VcCalendarCollection**

Mit dieser Methode können Sie einen Kalender kopieren. Wenn der Kalender mit dem angegebenen Namen existiert und der Name des neuen Kalenders noch nicht verwendet wird, wird das neue Kalenderobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ calendarName	String	Name des zu kopierenden Kalenders
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ newCalendarName	String	Name des neuen Kalenders
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcCalendar	Kalenderobjekt

FirstCalendar**Methode von VcCalendarCollection**

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Kalender der Kalender-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextCalendar1** über die nachfolgenden Kalender zu iterieren. Existiert kein Kalender in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendar	Erster Kalender

NextCalendar

Methode von VcCalendarCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Kalender der kalenderauflistung zugreifen, nachdem Sie mit der Methode **FirstCalendar** den Initialwert erfasst haben. Sind alle Kalender durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendar	Nachfolgender Kalender

Code-Beispiel

```
Dim calendarCltn As VcCalendarCollection
Dim calendar As VcCalendar

Set calendarCltn = VcGantt1.CalendarCollection
Set calendar = calendarCltn.FirstCalendar

While Not calendar Is Nothing
    List1.AddItem (calendar.Name)
    Set calendar = calendarCltn.NextCalendar
Wend
```

Remove

Methode von VcCalendarCollection

Mit dieser Methode können Sie einen Kalender löschen. Wenn der Kalender noch in irgendeinem anderen Objekt verwendet wird, kann er nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Kalender gelöscht (True)/nicht gelöscht (False)

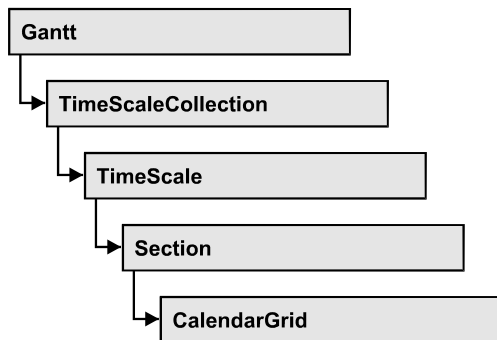
Update

Methode von VcCalendarCollection

Mit dieser Methode können Sie eine Kalender-Collection aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

7.13 VcCalendarGrid



Ein Objekt vom Typ **VcCalendarGrid** ist ein Kalendergitter, durch das arbeitsfreie Tage durch farbige Flächen besonders gekennzeichnet werden.

Eigenschaften

- BackColorAsARGB
- BackColorDataFieldIndex
- BackColorMapName
- CalendarName
- CalendarNameDataFieldIndex
- CalendarNameMapName
- EndSnapTarget
- Identifiable
- LineColor
- LineColorDataFieldIndex
- LineColorMapName
- LineThickness
- LineType
- Name
- Pattern
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternDataFieldIndex
- PatternMapName
- Priority
- SnapTarget
- Specification
- StartSnapTarget
- UseGraphicalAttributesOfIntervals

- Visible
- VisibleDataFieldIndex
- VisibleMapName

Methoden

- IdentifyInterval
- IdentifyIntervalAsVariant

Eigenschaften

BackColorAsARGB

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Farbe der Flächen des Kalendergitters erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getPatternColorAsARGB**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255} Standardwert: &hFFD8D8D8 (grau)
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim section As VcSection
Dim calendarGrid As VcCalendarGrid

Set section = VcGantt1.TimeScaleCollection.Active.Section(0)
Set calendarGrid = section.CalendarGrid
calendarGrid.BackgroundColorAsARGB = &h88FF0A06
```

BackColorDataFieldIndex

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **BackColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

BackColorMapName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuhordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **BackColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **BackColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

CalendarName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie dem Kalendergitter einen Kalender zuweisen um dessen arbeitsfreie Zeiten farblich hervorzuheben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Kalendernamen übergibt
	Mögliche Werte:	

	Name der Farbzuoordnungstabelle
--	---------------------------------

CalendarNameDataFieldIndex

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das den Namen des zu verwendenden Kalenders für eine Gruppierungsebene enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Kalender** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Namen des zu verwendenden Kalenders enthält

CalendarNameMapName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Namen einer Kalenderzuordnungstabelle (Typ vcTextMap) setzen oder erfragen. Wird hier "" angegeben, wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Kalenderzuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **CalendarNameDataFieldIndex** angegeben ist, wird der Kalender aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird der Kalender verwendet, der dem Kalendergitter der Gruppierungsebene zugewiesen ist.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Kalenderzuordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

EndSnapTarget

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Endtermin dieses Kalendergitters als Einrastziel definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Endtermin des Kalendergitters wird/wird nicht als Einrastziel definiert.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Identifiable

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Kalendergitter identifizierbar ist. Wenn die Eigenschaft auf **True** gesetzt wurde, kann das Kalendergitter mittels der VcGantt-Methode **IdentifyObjectAt** identifiziert werden. Auch ein Tooltip-Text wird über **OnTooltipText** nur bei einem identifizierbaren Gitter angefordert. Ebenso wird das Ereignis **OnCalendarGridRClick** nur dann ausgelöst, wenn das Kalendergitter identifizierbar ist.

Für einen passenden Tooltip-Text muss zudem der entsprechende Intervall identifiziert werden: s. VcGantt-Methode **IdentifyInterval**.

Diese Eigenschaft kann ebenfalls im **Kalendergitter**-Bereich des Dialogs **Zeitskalenabschnitt bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kalendergitter identifizierbar / nicht identifizierbar Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim section As VcSection
Dim calendarGrid As VcCalendarGrid

Set section = VcGantt1.TimeScaleCollection.Active.Section(0)
Set calendarGrid = section.CalendarGrid
calendarGrid.Identifiable = True
```

LineColor

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Farbe eines Kalendergitters erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Linienattribute des Kalendergitters** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: Wie im Dialog definiert

LineColorDataFieldIndex

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuoordnungstabelle in der Eigenschaft **LineColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

LineColorMapName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle für die Linienfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **LineColorDataFieldIndex -1** angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuoordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

LineThickness

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Linienstärke eines Kalendergitters erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Diese Eigenschaft können Sie ebenfalls in dem Dialog **Attribute des Kalendergitters** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

LineType

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Linientyp eines Kalendergitters erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Linienattribute des Kalendergitters** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp
	Mögliche Werte:	
	vcDashed 4	Linientyp gestrichelt
	vcDashedDotted 5	Linientyp gestrichelt-gepunktet
	vcDotted 3	Linientyp gepunktet
	vcLineType0 100	Linientyp 0

	vcLineType1 101	Linientyp 1

	vcLineType10 110	Linientyp 10

	vcLineType11 111	Linientyp 11

	vcLineType12 112	Linientyp 12

	vcLineType13 113	Linientyp 13

	vcLineType14 114	Linientyp 14

	vcLineType15 115	Linientyp 15

	vcLineType16 116	Linientyp 16

	vcLineType17 117	Linientyp 17

	vcLineType18 118	Linientyp 18

	vcLineType2 102	Linientyp 2
		
	vcLineType3 103	Linientyp 3

	vcLineType4 104	Linientyp 4

	vcLineType5 105	Linientyp 5

	vcLineType6 106	Linientyp 6

	vcLineType7 107	Linientyp 7

	vcLineType8 108	Linientyp 8

	vcLineType9 109	Linientyp 9

	vcNone 1	Kein Linientyp
	vcNotSet -1	Kein Linientyp zugewiesen
	vcSolid 2	Linientyp durchgezogen

Name

Eigenschaft von VcCalendarGrid

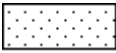
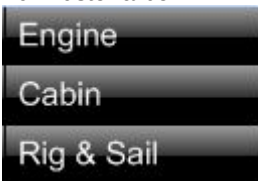
Mit dieser Eigenschaft können Sie den Namen eines Kalendergitters erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalendergitters
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Pattern

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie das Muster des Kalendergitters festlegen oder erfragen. S. auch **set/getPatternColor**.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 

vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
vcDivotPattern 2036	Grassoden-Muster
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern

vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschildel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf

vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

PatternColorAsARGB

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Musterfarbe des Kalendergitters festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getBackgroundColorAsARGB** und **set/getPattern**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim section As VcSection
Dim calendarGrid As VcCalendarGrid
```

```
Set section = VcGantt1.TimeScaleCollection.Active.Section(0)
Set calendarGrid = section.CalendarGrid
calendarGrid.PatternColorAsARGB = &h88FF0A06
```

PatternColorDataFieldIndex

Nur-Lese-Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

PatternColorMapName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuhordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Kalendergitters aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

PatternDataFieldIndex

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

PatternMapName

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie den Namen einer Musterzuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn der Name einer Farbzuhordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternDataFieldIndex** angegeben sind, wird das Muster des Kalendergitters aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **Pattern** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Priority

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Priorität des Kalendergitters erfragen oder festlegen. Wenn zwei Objekte dieselbe Position im Diagramm haben, liegt das Objekt mit der höheren Priorität über dem Objekt mit der niedrigeren Priorität. Kalendergitter haben standardmäßig die niedrigste Priorität. Knoten besitzen mit dem Wert 0 die höchste Priorität von allen

Objekten. Soll ein Kalendergitter vor den Knoten liegen, müssen Sie seine Priorität auf einen positiven Wert setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Prioritätsstufe {-100 ... 100} Standardwert: -20
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim section As VcSection
Dim calendarGrid As VcCalendarGrid

Set section = VcGantt1.TimeScaleCollection.Active.Section(0)
Set calendarGrid = section.CalendarGrid
calendarGrid.Priority = 3
```

SnapTarget

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob für dieses Kalendergitter ein Einrastziel am Datum definiert wird.

	Datentyp	Beschreibung

Specification

Nur-Lese-Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie die Spezifikation dieses Kalendergitters erfragen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Kalendergitters mit der Methode **VcCalendarGridCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung

StartSnapTarget

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Starttermin dieses Kalendergitters als Einrastziel definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Starttermin des Kalendergitters wird/wird nicht als Einrastziel definiert.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UseGraphicalAttributesOfIntervals

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob die für die Intervalle eingestellten grafischen Attribute dargestellt werden. Diese Option kann auch im Dialog **Intervalle verwalten** (zu erreichen über den Dialog **Kalender festlegen**) eingestellt werden. Wenn die Eigenschaft auf **False** gesetzt wurde, bleiben die Einstellungen der Eigenschaft **VcInterval.UseGraphicalAttributes** wirkungslos.

	Datentyp	Beschreibung

Visible

Eigenschaft von VcCalendarGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Kalendergitter sichtbar ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kalendergitter sichtbar/unsichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim section As VcSection
Dim calendarGrid As VcCalendarGrid

Set section = VcGantt1.TimeScaleCollection.Active.Section(0)
Set calendarGrid = section.CalendarGrid
calendarGrid.Visible = True

```

VisibleDataFieldIndex**Eigenschaft von VcCalendarGrid**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das den Wert "1" (für "sichtbar") oder 0 (für "unsichtbar") enthält. Diese Eigenschaft kann auch im Dialog **Kalendergitter** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Sichtbarkeitsmodus enthält

VisibleMapName**Eigenschaft von VcCalendarGrid**

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle (Typ vcTextMap) für den Sichtbarkeitsmodus setzen oder erfragen. Wird hier "" angegeben, wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **VisibleDataFieldIndex** angegeben ist, wird der Sichtbarkeitsstatus aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Kalendergitter auf "sichtbar" gesetzt. Diese Eigenschaft kann auch im Dialog **Kalendergitter** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Zuordnungstabelle für den Sichtbarkeitsmodus
	Mögliche Werte:	Name der Farbzusordnungstabelle

Methoden

IdentifyInterval

Methode von VcCalendarGrid

Mit dieser Methode können Sie an den übergebenen Koordinaten ein Intervall-Objekt des Kalenders identifizieren, das dem Kalendergitter zugewiesen wurde. Da Intervalle wiederholt werden, sind meist nicht eindeutig (z.B. kann sich dasselbe Wochenendintervall in einem Zeitbereich von einem Jahr 52 mal wiederholen). Daher liefert die Methode zusätzlich die Anfangs- und Endtermine des betroffenen Intervalls.

Die Methode kann z.B. innerhalb des TooltipText-Ereignisses aufgerufen werden, um das Intervall unter dem Mauszeiger zu ermitteln.

Falls sich an der bezeichneten Position ein Intervall befindet, wird **True** zurückgegeben, andernfalls **False**.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der by-Reference Parameter nur die analoge Methode **IdentifyIntervalAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇐ identifiedIntervalParam	VcInterval	Identifiziertes Intervall
⇐ startDateParam	Date/Time	Anfangsdatum des identifizierten Intervalls
⇐ endDateParam	Date/Time	Enddatum des identifizierten Intervalls
Rückgabewert	Boolean	Intervall konnte identifiziert / nicht identifiziert werden

Code-Beispiel

```
Private Sub VcGantt1_DragDrop(Source As Control, X As Single, Y As Single)

    Dim identifiedObj As Object
    Dim identifiedObjType As VcObjectTypeEnum
    Dim identifiedIntervalParam As VcInterval
    Dim startDateParam As System.Date
    Dim endDateParam As System.Date
    Dim xPix, yPix As Long

    xPix = X / Screen.TwipsPerPixelX
    yPix = Y / Screen.TwipsPerPixelY
```

```

    Call VcGantt1.IdentifyInterval(xPix, yPix, identifiedIntervalParam, _
                                   startDateParam, endDateParam)
    If Not identifiedIntervalParam Is Nothing Then
        MsgBox ("This Interval "" + identifiedIntervalParam() + _
                """, ranges from "" + startDateParam + _
                """, to ""'+endDateParam")
    Else
        MsgBox ("At this position no interval was identified.")
    End If
End Sub

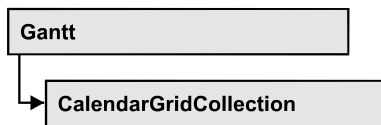
```

IdentifyIntervalAsVariant

Methode von VcCalendarGrid

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyInterval**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript by-Reference-Parameter (gekennzeichnet durch ) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

7.14 VcCalendarGridCollection



In einem Objekt des Typs `VcCalendarGridCollection` sind alle verfügbaren Kalendergitter zusammengefasst. Über **For Each calendarGrid In CalendarGridCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Kalendergitter zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **CalendarGridByName** und **CalendarGridByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Kalendergitter kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Kalendergittern.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `CalendarGridByIndex`
- `CalendarGridByName`
- `Copy`
- `FirstCalendarGrid`
- `NextCalendarGrid`
- `Remove`
- `Update`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcCalendarGridCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Kalendergitterobjekte iterieren. In Visual Basic wird diese Eigen-

schaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim calendarGrid As VcCalendarGrid

For Each calendarGrid In VcGantt1.CalendarGrid
    Debug.Print calendarGrid.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcCalendarGridCollection

Mit dieser Eigenschaft kann die Anzahl der Kalenderliniengitter in der Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der CalendarGrids

Code-Beispiel

```
Dim calendarGridCltn As Vc CalendarGridCollection
Dim numberOfCalendarGrids As Long

Set calendarGridCltn = VcGantt1.CalendarGridCollection
numberOfCalendarGrids = calendarGridCltn.Count
```

Methoden

Add

Methode von VcCalendarGridCollection

Mit dieser Methode können Sie ein neues Kalenderliniengitter in der CalendarGridCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Kalenderliniengitterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ calendarGridName	String Mögliche Werte:	Name des CalendarGrids Name der Farbzuoordnungstabelle
Rückgabewert	VcCalendarGrid	Neues CalendarGrid-Objekt

Code-Beispiel

```
Set newCalendarGrid = VcGantt1.CalendarGridCollection.Add("Grid1")
```

AddBySpecification**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie ein Kalenderliniengitter über eine Kalenderliniengitter-Spezifikation erzeugen. Dies dient der Persistenz von Kalenderliniengitter-Objekten. Die Spezifikation eines Kalenderliniengitters kann erfragt und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Kalenderliniengitter mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ specification	String Mögliche Werte:	CalendarGrid-Spezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcCalendarGrid	Neues CalendarGrid-Objekt

CalendarGridByIndex**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie auf eine einzelne Kalenderliniengitter über ihren Index zugreifen. Existiert kein Kalenderliniengitter unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des CalendarGrids

		Datenfeldindex
Rückgabewert	VcCalendarGrid	Ermitteltes CalendarGridobjekt

Code-Beispiel

```

Dim calendarGridCltn As VcCalendarGrid
Dim calendar As VcCalendar

Set calendarGridCltn = VcGantt1.CalendarGrid
Set calendarGrid = calendarGridCltn.CalendarGridByIndex(2)
MsgBox calendarGrid.Name

```

CalendarGridByName**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie unter Verwendung des Namens des Kalenderliniengitters auf ein bestimmtes Kalenderliniengitter zugreifen. Existiert kein CalendarGrid-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ calendarGridName	String	Name des CalendarGrids
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcCalendarGrid	CalendarGrid

Code-Beispiel

```

Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
Set calendarGrid = calendarGridCltn.CalendarGridByName("Grid 4")

```

Copy**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie ein Kalenderliniengitter kopieren. Wenn das Kalenderliniengitter mit dem angegebenen Namen existiert und der Name des neuen Kalenderliniengitters noch nicht verwendet wird, wird das neue Kalenderliniengitterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ calendarGridName	String	Name des zu kopierenden CalendarGrids
	Mögliche Werte:	Name der Farbzordnungstabelle
	String	Name des neuen CalendarGrids
	Mögliche Werte:	Name der Farbzordnungstabelle
⇒ newCalendarGridName	String	Name des neuen CalendarGrids
	Mögliche Werte:	Name der Farbzordnungstabelle
Rückgabewert	VcCalendarGrid	CalendarGridobjekt

Code-Beispiel

```

Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
Set calendarGrid = calendarGridCltn.Copy("CurrentCalendarGrid",
"NewCalendarGrid")

```

FirstCalendarGrid**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie auf das erste Kalenderliniengitter der Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextCalendarGrid** über die nachfolgenden Kalenderliniengitter zu iterieren. Existiert kein Kalenderliniengitter in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendarGrid	Erstes CalendarGrid

Code-Beispiel

```

Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
calendarGridCltn.CalendarGrids (vcAnyCalendarGrid)
Set calendarGrid = calendarGridCltn.FirstCalendarGrid

```

NextCalendarGrid**Methode von VcCalendarGridCollection**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Kalenderliniengitter der Auflistung zugreifen, nachdem Sie mit der Methode

FirstCalendarGrid den Initialwert erfasst haben. Sind alle Levels durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendarGrid	Nachfolgendes CalendarGrid

Code-Beispiel

```
Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
Set calendarGrid = calendarGridCltn.FirstCalendarGrid

While Not calendarGrid Is Nothing
    Listbox.AddItem calendarGrid.Name
    Set calendarGrid = calendarGridCltn.NextCalendarGrid
Wend
```

Remove

Methode von VcCalendarGridCollection

Mit dieser Methode können Sie ein Kalenderliniengitter löschen. Wenn das Terminliniengitter noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ calendarGridName	String	Name des CalendarGrids
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	CalendarGrid gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
Set calendarGrid = calendarGridCltn.FormatByIndex(1)
calendarGridCltn.Remove (calendarGrid.Name)
```

Update

Methode von VcCalendarGridCollection

Mit dieser Methode können Sie die Darstellung aller Objekte, die durch die verwendeten Kalenderliniengitter bestimmt werden, aktualisieren. Wenn Sie diese Methode nicht aufrufen, werden die Änderungen der

Kalenderliniengitter zur Laufzeit nicht ausgeführt. Sie sollten diese Methode erst am Ende des Codes zur Festlegung der Kalenderliniengitter und der Level-Auflistung verwenden, damit die Aktualisierung nicht schon ausgeführt wird, bevor alle Festlegungen der Kalenderliniengitter ausgeführt worden sind.

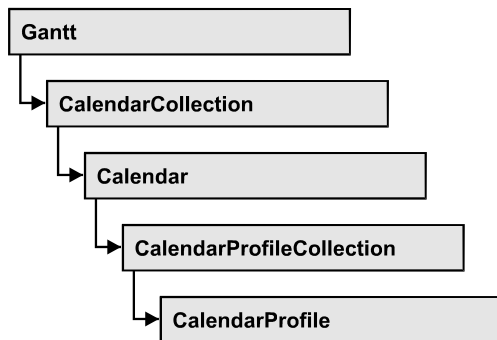
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim calendarGrid As VcCalendarGrid
```

```
Set calendarGrid = VcGantt1.CalendarGrid.Collection.CalendarGridByName("Grid 3")  
calendarGrid.Update
```

7.15 VcCalendarProfile



In einem Objekt vom Typ **VcCalendarProfile** legen Sie ein Kalenderprofil fest.

Eigenschaften

- IntervalCollection
- Name
- Specification
- Type

Methoden

- PutInOrderAfter

Eigenschaften

IntervalCollection

Nur-Lese-Eigenschaft von VcCalendarProfile

Mit dieser Eigenschaft haben Sie Zugriff auf die Intervallauflistung, in der alle zur Verfügung stehenden Intervalle enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcIntervalCollection	IntervalCollection-Objekt

Name

Nur-Lese-Eigenschaft von VcCalendarProfile

Mit dieser Eigenschaft können Sie den Namen des Kalenderprofils setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalenderprofils
	Mögliche Werte:	Name der Farbzusordnungstabelle

Specification

Nur-Lese-Eigenschaft von VcCalendarProfile

Mit dieser Eigenschaft können Sie die Spezifikation dieses Kalenderprofils erfragen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Kalenderprofils mit der Methode **VcCalendarProfileCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Kalenderprofils
	Mögliche Werte:	Name der Farbzusordnungstabelle

Type

Eigenschaft von VcCalendarProfile

Mit dieser Eigenschaft können Sie den Typ des Kalenderprofils setzen oder erfragen. Wenn Sie den Typ ändern, gehen alle für das Kalenderprofil gesetzten Eigenschaften verloren.

	Datentyp	Beschreibung
Eigenschaftswert	CalendarProfileTypeEnum	Typ des Kalenderprofils
	Mögliche Werte: vcDayProfile 4 vcShiftProfile 5 vcWeekProfile 3	

vcYearProfile 2

Methoden

PutInOrderAfter

Methode von VcCalendarProfile

Mit dieser Methode können Sie dieses Kalenderprofil in der Auflistung aller Kalenderprofile hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Kalenderprofil an die erste Stelle gesetzt. Die Reihenfolge der Kalenderprofile in der Auflistung entscheidet darüber, in welcher Reihenfolge sie in Kalendern angewendet werden.

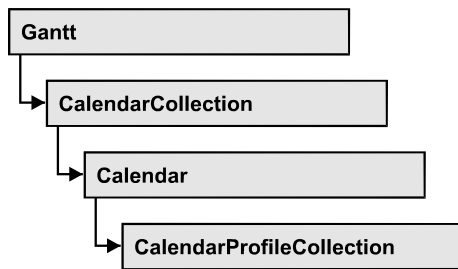
	Datentyp	Beschreibung
Parameter:		
refNameParam	String	Name des Kalenderprofils, hinter das das aktuelle Kalenderprofil gesetzt werden soll.
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim calProfCltn As VcCalendarProfileCollection
Dim calProf1 As VcCalendarProfile
Dim calProf2 As VcCalendarProfile

calProfCltn = VcGantt1.CalendarProfileCollection()
calProf1 = calProfCltn.Add("calProf1")
calProf2 = calProfCltn.Add("calProf2")
calProf1.PutInOrderAfter("calProf2")
calProfCltn.Update()
```

7.16 VcCalendarProfileCollection



In einem Objekt vom Typ `VcCalendarProfileCollection` sind automatisch alle verfügbaren Kalenderprofile zusammengefasst. Über **For Each calendar-Profile In CalendarProfileCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Kalenderprofile zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **CalendarProfileByName** und **CalendarProfileByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Kalenderprofile kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Kalenderprofilen.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `CalendarProfileByIndex`
- `CalendarProfileByName`
- `Copy`
- `FirstCalendarProfile`
- `NextCalendarProfile`
- `Remove`
- `SelectCalendarProfiles`
- `Update`

Eigenschaften

_NewEnum

Eigenschaft von VcCalendarProfileCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Kalenderprofilobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Count

Nur-Lese-Eigenschaft von VcCalendarProfileCollection

Mit dieser Eigenschaft können Sie die Anzahl der Kalenderprofilobjekte in der Kalenderprofil-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der CalendarProfile-Objekte

Methoden

Add

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie ein neues Kalenderprofil in der FilterCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Kalenderprofilobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ profileName	String Mögliche Werte:	Name des Kalenderprofils Name der Farbzusordnungstabelle
Rückgabewert	VcCalendarProfile	Neues Kalenderprofilobjekt

AddBySpecification

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie ein Kalenderprofil über eine Kalenderprofil-Spezifikation erzeugen. Dies dient der Persistenz von Kalenderprofilobjekten. Die Spezifikation eines Kalenderprofilobjektes kann erfragt (siehe VcCalendarProfile-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Kalenderprofil mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Kalenderprofilspezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcCalendarProfile	Neues Kalenderprofilobjekt

CalendarProfileByIndex

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie auf ein einzelnes Kalenderprofil über seinen Index zugreifen. Existiert kein Kalenderprofil unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Kalenderprofils Datenfeldindex

Rückgabewert	VcCalendarProfile	Ermitteltes CalendarProfile-Objekt
---------------------	-------------------	------------------------------------

CalendarProfileByName

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Kalenderprofil zugreifen. Existiert kein Kalenderprofil unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ profileName	String Mögliche Werte:	Name des CalendarProfile-Objekts Name der Farbzusordnungstabelle
Rückgabewert	VcCalendarProfile	Zurückgegebenes CalendarProfile-Objekt

Copy

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie ein Kalenderprofil kopieren. Wenn das Kalenderprofil mit dem angegebenen Namen existiert und der Name des neuen Kalenderprofils noch nicht verwendet wird, wird das neue Kalenderprofilobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ profileName	String Mögliche Werte:	Name des zu kopierenden Kalenderprofils Name der Farbzusordnungstabelle
⇒ newprofileName	String Mögliche Werte:	Name des neuen Kalenderprofils Name der Farbzusordnungstabelle
Rückgabewert	VcCalendarProfile	CalendarProfile-Objekt

FirstCalendarProfile

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. das erste Kalenderprofil der Kalenderprofil-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextCalendarProfile** über die nachfolgenden Kalenderprofile zu iterieren. Existiert kein Kalenderprofil in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendarProfile	Erstes CalendarProfile-Objekt

NextCalendarProfile

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Kalenderprofile der Kalenderprofil-Auflistung zugreifen, nachdem Sie mit der Methode **FirstCalendarProfile** den Initialwert erfasst haben. Sind alle Kalenderprofile durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCalendarProfile	Nachfolgendes CalendarProfile-Objekt

Remove

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie ein Kalenderprofil löschen. Wenn das Kalenderprofil noch in irgendeinem anderen Objekt verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ profileName	String	Name des Kalenderprofils
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	Kalenderprofil gelöscht (True)/nicht gelöscht (False)

SelectCalendarProfiles

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie festlegen, welche Kalenderprofile in der Auflistung der Kalenderprofile verfügbar sein sollen.

	Datentyp	Beschreibung
Parameter: ⇒ selectionType	CalendarProfileTypeEnum Mögliche Werte: vcDayProfile 4 vcShiftProfile 5 vcWeekProfile 3 vcYearProfile 2	Auszuwählender Kalenderprofiltyp
Rückgabewert	Long	Anzahl der ausgewählten Kalenderprofile

Code-Beispiel

```
Dim calendarProfileCltn As VcCalendarProfileCollection

Set calendarProfileCltn = VcGantt1.CalendarProfileCollection
calendarProfileCltn.SelectCalendarProfile (vcSelected)
```

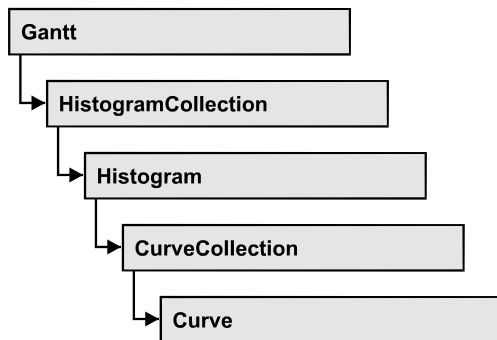
Update

Methode von VcCalendarProfileCollection

Mit dieser Methode können Sie eine CalendarProfile-Collection aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

7.17 VcCurve



Ein Objekt vom Typ VcCurve stellt eine Kurve dar, mit der Sie beispielsweise die Ressourcenauslastung und -verfügbarkeit in einem Histogramm darstellen können. Die Werte von Histogrammkurven können Sie entweder direkt setzen oder aus einzelnen Layern ableiten lassen.

Um die Werte direkt zu setzen, wählen Sie im Dialog **Datenquelle der Kurve einstellen** die Option **Daten manuell eingeben** und weisen Sie in Ihrer Anwendung der Kurve mit der Methode **SetValues** Werte zu.

Um die Werte aus den Vorgängen berechnen zu lassen, wählen Sie im Dialog **Datenquelle der Kurve einstellen** die Option **Daten über Layer generieren** und wählen Sie dann einen der angebotenen Layer aus.

Eigenschaften

- Addend
- CurveSource
- CurveType
- Fill2Color
- Fill2Pattern
- Fill2ReferenceName
- FillColor
- FillPattern
- FillReferenceName
- FilterName
- Histogram
- LayerName
- LineColor
- LineThickness
- LineType
- MarkCurve
- Name

- OverloadResultsCalendarName
- Pattern2Color
- PatternColor
- PointsEquidistant
- Specification
- StackReferenceName
- TimeUnit
- UnitsPerStep
- UpdateBehaviorName
- ValencyDataFieldIndex
- Visible

Methoden

- Clear
- DeletePoint
- DeletePointAsVariant
- GetFirstOverload
- GetFirstOverloadAsVariant
- GetFirstOverloadEx
- GetNextOverload
- GetNextOverloadAsVariant
- GetNextOverloadEx
- GetValues
- GetValuesAsVariant
- GetValuesEx
- SetValues

Eigenschaften

Addend

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie alle y-Werte einer per API generierten Histogrammkurve um den übergebenen Wert erhöhen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Wert, um den die y-Werte der Histogrammkurve erhöht werden sollen

Code-Beispiel

```

Dim histogram As VcHistogram
Dim fixCurve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set fixCurve = histogram.CurveCollection.CurveByName("Availability")

fixCurve.Addend ("1")

```

CurveSource**Nur-Lese-Eigenschaft von VcCurve**

Mit dieser Eigenschaft können Sie die Quelle, aus der die Daten einer Histogrammkurve entnommen werden, erfragen. Setzen können Sie diese Eigenschaft im Dialog **Datenquelle der Kurve einstellen**. Wenn Sie **vcSetCurve** (erzeugt durch die Option **Daten über Layer generieren** im Dialog **Datenquelle der Kurve einstellen**) zurückerhalten, können Sie die Daten in Ihrer Anwendung mit der Methode **SetValues** setzen. Wenn Sie **vcCalculateFromLayer** zurückerhalten (Option **Daten automatisch über Layer generieren**), werden die Daten aus den Layern berechnet.

	Datentyp	Beschreibung
Eigenschaftswert	CurveSourceEnum Mögliche Werte: vcCalculateFromLayer 1 vcSetCurve 3	Art der Werteberechnung, Kurvenwerte setzen Werteberechnung aus Layer Die Werte werden manuell (per API) gesetzt.

Code-Beispiel

```

Dim histogram As VcHistogram
Dim curve As VcCurve
Dim curveSource As Long

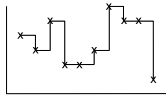
Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curveSource = curve.CurveSource

```

CurveType**Nur-Lese-Eigenschaft von VcCurve**

Mit dieser Eigenschaft können Sie den Typ einer Histogrammkurve erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	CurveTypeEnum	Kapazitätskurve Standardwert: vcCapacityCurve
	Mögliche Werte: vcCapacityCurve 215	Kapazitätskurve 
	vcLineCurve 214	Linienkurve 
	vcStepCurve 216	Stufenkurve 

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
Dim curveType As Long
```

```
Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")
```

```
curveType = curve.CurveType
```

Fill2Color**Eigenschaft von VcCurve**

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Musters oberhalb der zweiten Bezugskurve erfragen oder festlegen.

Die Füllung zur 2. Bezugskurve wird nur dargestellt, wenn die y-Werte der aktuellen Kurve größer sind als die der 2. Bezugskurve.

Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** unter **Muster 2** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255}) Standardwert: Wie im Dialog definiert

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
```

```
Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.Fill2Color = RGB(150, 100, 170)
```

Fill2Pattern

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie das Füllmuster des Bereichs zwischen einer Histogrammkurve und der 2. Bezugskurve erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Art des Füllmusters
		Standardwert: Wie im Dialog definiert
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben
	vcCrossPattern 6	Kreuzschraffur
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke

vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
vcDivotPattern 2036	Grassoden-Muster
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern

vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel

vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.Fill2Pattern = vcDiagCrossPattern
```

Fill2ReferenceName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Namen der 2. Bezugskurve einer Histogrammkurve erfragen oder setzen. Zusammen mit der Histogrammkurve begrenzt diese eine Fläche, die mit einem Muster gefüllt werden kann. Diese Eigenschaft wird im Dialog **Histogramm bearbeiten** festgelegt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der 2. Bezugskurve
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
Dim fillRef As Object

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")
Set referenceCurve = histogram.CurveCollection._
    CurveByName(curve.Fill2ReferenceName)
```

FillColor

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Farbe des Bereichs zwischen einer Histogrammkurve und dem gewählten Bezugsobjekt erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** unter **Muster** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: Wie im Dialog definiert

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

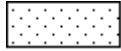
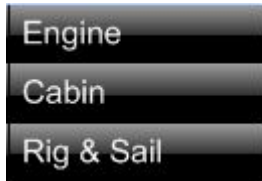
Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

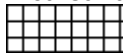
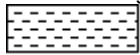
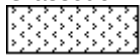
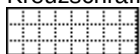
curve.FillColor = RGB(150, 100, 170)
```

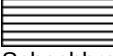
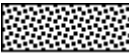
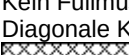
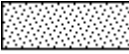
FillPattern

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie das Füllmuster des Bereichs zwischen einer Histogrammkurve und dem gewählten Bezugsobjekt erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp Standardwert: Wie im Dialog definiert Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11 vcAeroGlassPattern 40
		Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter  Vertikaler Farbverlauf in der Füllmusterfarbe 

vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
vcCrossPattern 6	Kreuzschraffur 
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 

vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern 
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 

vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.FillPattern = vcDiagCrossPattern
```

FillReferenceName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Namen des Bezugsobjekts (x-Achse oder eine Kurve) einer Histogrammkurve für eine Flächenfüllung erfragen.

Zusammen mit der Histogrammkurve begrenzt das Bezugsobjekt eine Fläche, die anschließend farbig und/oder mit Mustern gefüllt werden kann. Diese Eigenschaft wird im Dialog **Histogramm bearbeiten** festgelegt.

Hinweis: Der Name der x-Achse als Bezugsobjekt muss mit "VC_AXIS" angegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Bezugskurve
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Private Sub VcGantt2_OnDiagramRClick(ByVal x As Long, ByVal y As Long,
returnStatus As Variant)
    Dim histogram As VcHistogram
    Dim curve As VcCurve

    Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
    Set curve = histogram.CurveCollection.CurveByName("Curve_1")
    curve.FillReferenceName = "VC_AXIS"
End Sub
```

FilterName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie der Kurve einen Filter zuweisen oder den vorhandenen erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Filtername
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.histogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")
curve.FilterName = "Critical"
```

Histogram

Nur-Lese-Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie erfragen, zu welchem Histogramm eine Kurve gehört.

	Datentyp	Beschreibung
Eigenschaftswert	VcHistogram	Histogrammobjekt

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set curve = VcGantt1.HistogramCollection.FirstHistogram.curveCollection. _
    CurveByName("Curve1")
Set histogram = curve.Histogram
```

LayerName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie der Kurve einen Layer zuweisen oder den vorhandenen erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Layers
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.histogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")
curve.LayerName = "Start-End"
```

LineColor

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Farbe einer Histogrammkurve erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {{0...255},{0...255},{0...255}} Standardwert: Wie im Dialog definiert

Code-Beispiel

```

Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.LineColor = RGB(200, 0, 180)

```

LineThickness**Eigenschaft von VcCurve**

Mit dieser Eigenschaft können Sie die Linienstärke einer Histogrammkurve erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Diese Eigenschaft können Sie ebenfalls in dem Dialog **Histogramm bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.LineType = vcSolid
curve.LineThickness = 3

' or:
curve.LineType = vcLineType5
curve.LineThickness = 20
```

LineType

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Linientyp einer Histogrammkurve erfragen oder festlegen. Soll z. B. bei gestapelten Kurven die Linie nicht dargestellt werden, wählen Sie **vcNone**. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp Standardwert: vcSolid
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100 vcLineType1 101 vcLineType10 110 vcLineType11 111 vcLineType12 112 vcLineType13 113 vcLineType14 114	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0 Linientyp 1 Linientyp 10 Linientyp 11 Linientyp 12 Linientyp 13 Linientyp 14

vcLineType15 115	Linientyp 15
vcLineType16 116	Linientyp 16
vcLineType17 117	Linientyp 17
vcLineType18 118	Linientyp 18
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3
vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.LineType = vcSolid
```

MarkCurve

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Markierungszustand einer per API generierten Histogrammkurve erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kurve markiert/nicht markiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim histogram As VcHistogram
Dim fixCurve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
```

```
Set fixCurve = histogram.CurveCollection.CurveByName("Availability")

fixCurve.MarkCurve = True
```

Name

Nur-Lese-Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Namen einer Histogrammkurve erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Kurvenname
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
Dim curveName As String

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curveName = curve.Name
```

OverloadResultsCalendarName

Eigenschaft von VcCurve

Mithilfe dieser Eigenschaft können Sie einen Kalender setzen oder erfragen, in den die Intervalle, die sich aus den berechneten Overload-Terminen ergeben, geschrieben werden. Dieser Kalender kann z.B. für die Darstellung eines Kalendergitters in der Gruppe verwendet werden

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalenderobjektes für die Overload-Ergebnisse
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Pattern2Color

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Vordergrundfarbe des Musters des Bereichs über der zweiten Bezugskurve erfragen oder festlegen. Die Füllung zur 2. Bezugskurve wird nur dargestellt, wenn die y-Werte der aktuellen Kurve größer sind als die der 2. Bezugskurve.

Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: Wie im Dialog definiert

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.Pattern2Color = RGB(150, 150, 110)
```

PatternColor

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Vordergrundfarbe des Musters des Bereichs zwischen einer Histogrammkurve und dem eingestellten Bezugsobjekt erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Histogramm bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: Wie im Dialog definiert

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.PatternColor = RGB(150, 150, 110)
```

PointsEquidistant

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Punkte der Kurve äquidistant sein sollen. Bei **False** werden Kurvenpunkte nur an den Stellen erzeugt, an denen sich der y-Wert ändert. Sie können diese Eigenschaft auch im Dialog **Datenquelle der Kurve einstellen** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kurvenpunkte äquidistant (True) / nicht äquidistant (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.FirstHistogram
Set curve = histogram.CurveCollection.CurveByName("Curve1")
curve.PointsEquidistant = False
```

Specification

Nur-Lese-Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Spezifikation dieser Kurve auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung einer Kurve mit der Methode **VcCurveCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Kurvenspezifikation
	Mögliche Werte:	Name der Farbzuoordnungstabelle

StackReferenceName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Namen der Stapelbezug-Kurve einer Histogrammkurve setzen oder erfragen. Er legt für jede Kurve fest, auf welche andere Kurve sie gestapelt werden soll. Er muss angegeben werden, wenn Kurven aufeinander gestapelt werden sollen. Diese Eigenschaft kann auch im Dialog **Histogramm bearbeiten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Stapelbezugskurve
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
Dim referenceCurve As Object

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

Set referenceCurve =
histogram.CurveCollection.CurveByName(curve.StackReferenceName)
```

TimeUnit

Nur-Lese-Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Zeiteinheit einer Histogrammkurve erfragen. Sie ist nur auf Kurven anwendbar, die per API generiert wurden. Wenn Sie die Eigenschaft auf eine aus Layerwerten generierte Kurve anwenden, wird das Ergebnis -1 zurückgeliefert. Die Zeiteinheit können Sie auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	TimeUnitEnum	Zeiteinheit Standardwert: Wie im Dialog definiert
	Mögliche Werte: vcDay 5 vcHour 6 vcMinute 7 vcSecond 8	Zeiteinheit Tag Zeiteinheit Stunde Zeiteinheit Minute Zeiteinheit Sekunde

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve
```

```

Dim timeUnit As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

timeUnit = curve.timeUnit

```

UnitsPerStep

Nur-Lese-Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie die Anzahl der Einheiten pro Schritt für eine Histogrammkurve erfragen. Die Anzahl kann auf der Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl Einheiten Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```

Dim histogram As VcHistogram
Dim curve As VcCurve
Dim unitsPerStep As Integer

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

unitsPerStep = curve.UnitsPerStep

```

UpdateBehaviorName

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

ValencyDataFieldIndex

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie das Wertigkeitsfeld einer automatisch generierten Kurve erfragen oder setzen, d. h. das Datenfeld, aus dem pro Vorgang die Wertigkeit für die Addition der Kapazitäten entnommen wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Wertigkeitsfeldes
	Mögliche Werte:	Datenfeldindex

Visible

Eigenschaft von VcCurve

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Kurve sichtbar sein soll oder nicht. Diese Eigenschaft kann auch im Dialog **Histogramme verwalten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kurve sichtbar/unsichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

curve.Visible = True
```

Methoden

Clear

Methode von VcCurve

Mit dieser Methode werden die y-Werte aller Punkte der Kurve auf Null gesetzt. Diese Methode kann nur auf Kurven angewendet werden, deren Werte per Automation definiert worden sind.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
Dim histogram As VcHistogram
Dim fixCurve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set fixCurve = histogram.CurveCollection.CurveByName("Availability")

fixCurve.Clear
```

DeletePoint

Methode von VcCurve

Mit dieser Methode können Sie bei einer per API erzeugten Kurve den Kurvenpunkt, der der angegebenen x-Koordinate am nächsten ist, löschen.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **DeletePointAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	x-Wert des zu löschenden Kurvenpunkts
⇒ y	Long	y-Wert des zu löschenden Kurvenpunkts
⇐ pointDate	Date	Datum des Kurvenpunktes, der gelöscht worden ist
Rückgabewert	Boolean	Kurvenpunkt erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
Private Sub VcGantt1_OnCurveRClick(ByVal curve As VcGanttLib.VcCurve, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)

    Dim pointDate As Date
    Dim deleted As Boolean
```

```

returnStatus = vcRetStatNoPopup
deleted = curve.DeletePoint(x, y, pointDate)
If deleted Then Call MsgBox(-pointDate)
End If
End Sub

```

DeletePointAsVariant

Methode von VcCurve

Diese Methode ist bis auf die Parameter identisch mit der Methode **DeletePoint**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GetFirstOverload

Methode von VcCurve

Ein Overload ist ein Zeitbereich, in dem die aktuelle Kurve höhere Werte aufweist als die Bezugskurve. Die Bezugskurve ist die Kurve, die im Dialog **Histogramm bearbeiten** als **2. Füllbezug** zur aktuellen Kurve definiert ist.

Mit dieser Methode können Sie auf den ersten Overload zugreifen, um anschließend in einer Schleife mit der Methode **GetNextOverload** über die nachfolgenden Overloads zu iterieren.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetFirstOverloadAsVariant** benutzen.

Hinweis: Für Fließkommazahlen in den Parametern **fromValue** und **toValue** verwenden Sie bitte die Methode **GetFirstOverloadEx**.

	Datentyp	Beschreibung
Parameter:		
 fromDate	Date/Time	Anfangsdatum des Overload-Bereiches
 fromValue	Long	Y-Wert am Anfangsdatum des Overload-Bereiches
 toDate	Date/Time	Enddatum des Overload Bereiches
 toValue	Long	y-Wert am Enddatum des Overload-Bereiches
Rückgabewert	Boolean	Overload erfolgreich (True) / nicht erfolgreich (False) erfragt

Code-Beispiel

```

Dim histogram As VcHistogram

Dim curve1 As VcCurve, fixCurve As VcCurve

Dim yValues As String
Dim bOK As Boolean

Dim fromDate As Date, toDate As Date
Dim fromValue As Long, toValue As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve1 = histogram.CurveCollection.CurveByName("LayerCurve")
Set fixCurve = histogram.CurveCollection.CurveByName("FixCurve")

yValues = "6;6;6;6;6;0;0;6;6;6;6;0;0;6;6;6;6;0;0;6;6;6;6;0;0;6;6;6;6;0;0;6;6;6;6;6"
fixCurve.SetValues "31.08.14", yValues

bOK = curve1.GetFirstOverload(fromDate, fromValue, toDate, toValue)

```

GetFirstOverloadAsVariant**Methode von VcCurve**

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetFirstOverload**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GetFirstOverloadEx**Methode von VcCurve**

Ein Overload ist ein Zeitbereich, in dem die aktuelle Kurve höhere Werte aufweist als die Bezugskurve. Die Bezugskurve ist die Kurve, die im Dialog **Histogramm bearbeiten** als **2. Füllbezug** zur aktuellen Kurve definiert ist.

Mit dieser Methode können Sie auf den ersten Overload zugreifen, um anschließend in einer Schleife mit der Methode **GetNextOverloadEx** über die nachfolgenden Overloads zu iterieren.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetFirstOverloadAsVariant** benutzen.

Hinweis: Im Vergleich zu der Methode **GetFirstOverload** erlaubt diese Methode Fließkommazahlen in den Parametern **fromValue** und **toValue**.

	Datentyp	Beschreibung
Parameter:		
⇐ fromDate	Date/Time	Anfangsdatum des Overload-Bereiches
⇐ fromValue	Double	Y-Wert am Anfangsdatum des Overload-Bereiches
⇐ toDate	Date/Time	Enddatum des Overload Bereiches
⇐ toValue	Double	y-Wert am Enddatum des Overload-Bereiches
Rückgabewert	Boolean	Overload erfolgreich (True) / nicht erfolgreich (False) erfragt

Code-Beispiel

```

Dim histogram As VcHistogram

Dim curve1 As VcCurve, fixCurve As VcCurve

Dim yValues As String
Dim bOK As Boolean

Dim fromDate As Date, toDate As Date
Dim fromValue As Long, toValue As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve1 = histogram.CurveCollection.CurveByName("LayerCurve")
Set fixCurve = histogram.CurveCollection.CurveByName("FixCurve")

yValues = "6;6;6;6;6;0;0;6;6;6;6;6;0;0;6;6;6;6;6;0;0;6;6;6;6;6;0;0;6;6;6;6;6"
fixCurve.SetValues "31.08.14", yValues

bOK = curve1.GetFirstOverload(fromDate, fromValue, toDate, toValue)

```

GetNextOverload**Methode von VcCurve**

Ein Overload ist ein Zeitbereich, in dem die aktuelle Kurve höhere Werte aufweist als die Bezugskurve. Die Bezugskurve ist die Kurve, die im Dialog **Histogramm bearbeiten** als **2. Füllbezug** zur aktuellen Kurve definiert ist.

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Overloads zugreifen, nachdem Sie mit der Methode **GetFirstOverload** den Initialwert erfasst haben.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetNextOverloadAsVariant** benutzen.

Hinweis: Für Fließkommazahlen in den Parametern **fromValue** und **toValue** verwenden Sie bitte die Methode **GetNextOverloadEx**.

	Datentyp	Beschreibung
Parameter:		
↔ fromDate	Date/Time	Anfangsdatum des Overload-Bereiches
↔ fromValue	Long	Y-Wert am Anfangsdatum des Overload-Bereiches
↔ toDate	Date/Time	Enddatum des Overload Bereiches
↔ toValue	Long	y-Wert am Enddatum des Overload-Bereiches
Rückgabewert	Boolean	Overload erfolgreich (True) / nicht erfolgreich (False) erfragt.

Code-Beispiel

```

...
bOK = curve1.GetFirstOverload(fromDate, fromValue, toDate, toValue)

MsgBox fromDate & " (" & fromValue & ") - " & toDate & " (" & toValue & ")"

While bOK
    bOK = curve1.GetNextOverload(fromDate, fromValue, toDate, toValue)
    If bOK Then MsgBox fromDate & " (" & fromValue & ") - " & toDate & " (" _
        & toValue & ")"
Wend

```

GetNextOverloadAsVariant**Methode von VcCurve**

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetNextOverload**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↔) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GetNextOverloadEx**Methode von VcCurve**

Ein Overload ist ein Zeitbereich, in dem die aktuelle Kurve höhere Werte aufweist als die Bezugskurve. Die Bezugskurve ist die Kurve, die im Dialog **Histogramm bearbeiten** als **2. Füllbezug** zur aktuellen Kurve definiert ist.

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Overloads zugreifen, nachdem Sie mit der Methode **GetFirstOverloadEx** den Initialwert erfasst haben.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetNextOverloadAsVariant** benutzen.

Hinweis: Im Vergleich zu der Methode **GetNextOverload** erlaubt diese Methode Fließkommazahlen in den Parametern **fromValue** und **toValue**.

	Datentyp	Beschreibung
Parameter:		
↔ fromDate	Date/Time	Anfangsdatum des Overload-Bereiches
↔ fromValue	Double	Y-Wert am Anfangsdatum des Overload-Bereiches
↔ toDate	Date/Time	Enddatum des Overload Bereiches
↔ toValue	Double	y-Wert am Enddatum des Overload-Bereiches
Rückgabewert	Boolean	Overload erfolgreich (True) / nicht erfolgreich (False) erfragt.

Code-Beispiel

```

...
bOK = curve1.GetFirstOverload(fromDate, fromValue, toDate, toValue)

MsgBox fromDate & " (" & fromValue & ") - " & toDate & " (" & toValue & ")"

While bOK
    bOK = curve1.GetNextOverload(fromDate, fromValue, toDate, toValue)
    If bOK Then MsgBox fromDate & " (" & fromValue & ") - " & toDate & " (" & _
        & toValue & ")"
Wend

```

GetValues

Methode von VcCurve

Mit dieser Methode können Sie den Wert einer Histogrammkurve zu einem bestimmten Datum erfragen. Da nicht sichergestellt ist, dass die Kurve an diesem bestimmten Datum auch einen Stützwert besitzt, werden jeweils das Datum und der Wert des nächsten Stützpunktes vor bzw. nach dem angegebenen Datum ausgegeben. Wird ein Stützpunkt genau getroffen, so wird der zugehörige Wert zwei Mal, das heißt, als voriger und nächster Wert ausgegeben.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetValuesAsVariant** benutzen.

Hinweis: Für Fließkomma-Werte verwenden Sie bitte die Eigenschaft **GetValuesEx**.

	Datentyp	Beschreibung
Parameter:		
⇒ inputDate	Date/Time	Datum, an dem der Wert der Histogrammkurve erfragt werden soll
⇐ leftDate	Date/Time	Datum des letzten Stützpunktes vor dem gewünschten Datum
⇐ leftValue	Long	Wert des letzten Stützpunktes vor dem gewünschten Datum
⇐ rightDate	Date/Time	Datum des nächsten Stützpunktes nach dem gewünschten Datum
⇐ rightValue	Long	Wert des nächsten Stützpunktes nach dem gewünschten Datum
Rückgabewert	void	

Code-Beispiel

```

Dim histogram As VcHistogram
Dim curve1 As VcCurve
Dim inputDate As String

Dim leftDate As Date, rightDate As Date
Dim leftValue As Long, rightValue As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")

Set curve1 = histogram.CurveCollection.CurveByName("LayerCurve")

inputDate = InputBox("Date: ")

curve1.GetValues CDate(inputDate), leftDate, leftValue, rightDate, rightValue

MsgBox leftDate & " (" & leftValue & ") " & rightDate & " (" & rightValue & ") "
```

GetValuesAsVariant

Methode von VcCurve

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetValues**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇐) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GetValuesEx

Methode von VcCurve

Mit dieser Methode können Sie den Wert einer Histogrammkurve zu einem bestimmten Datum erfragen. Im Vergleich zur Methode **GetValues** können

Sie hier auch Fließkomma-Zahlen verwenden. Da nicht sichergestellt ist, dass die Kurve an diesem bestimmten Datum auch einen Stützwert besitzt, werden jeweils das Datum und der Wert des nächsten Stützpunktes vor bzw. nach dem angegebenen Datum ausgegeben. Wird ein Stützpunkt genau getroffen, so wird der zugehörige Wert zwei Mal, das heißt, als voriger und nächster Wert ausgegeben.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetValuesAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ inputDate	Date/Time	Datum, an dem der Wert der Histogrammkurve erfragt werden soll
⇐ leftDate	Date/Time	Datum des letzten Stützpunktes vor dem gewünschten Datum
⇐ leftValue	Double	Wert des letzten Stützpunktes vor dem gewünschten Datum
⇐ rightDate	Date/Time	Datum des nächsten Stützpunktes nach dem gewünschten Datum
⇐ rightValue	Double	Wert des nächsten Stützpunktes nach dem gewünschten Datum
Rückgabewert	void	

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curve1 As VcCurve
Dim inputDate As String

Dim leftDate As Date, rightDate As Date
Dim leftValue As Long, rightValue As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")

Set curve1 = histogram.CurveCollection.CurveByName("LayerCurve")

inputDate = InputBox("Date: ")

curve1.GetValues CDate(inputDate), leftDate, leftValue, rightDate, rightValue

MsgBox leftDate & " (" & leftValue & ") " & rightDate & " (" & rightValue & ") "
```

SetValues

Methode von VcCurve

Mit dieser Methode können Sie die Werte einer per API generierten Histogrammkurve setzen. Eine mittels **SetValues** gesetzte Kurve kann

beispielsweise zur Anzeige der Maschinenkapazität verwendet werden oder als Bezugskurve dienen.

Von dem Kontrollkästchen **Kurvenpunkte äquidistant** im Dialog **Datenquelle der Kurve einstellen** hängt ab, wie diese Methode anzuwenden ist:

Kurvenpunkte äquidistant: Man kann mit der Methode einen Startwert (**startValue**) und einen String, der durch Semikola getrennt die y-Werte enthält, übergeben. Aus dem Startwert für x und dem String mit den y-Werten werden zusammen mit der **Zeiteinheit** und dem Wert für das **Kleinste Zeitintervall** (Eigenschaftenseite **Allgemeines**) die Wertepaare der Histogrammkurve ermittelt. So gesetzte Kurven können nicht interaktiv verändert werden.

Kurvenpunkte nicht äquidistant: Man muss die Methode für jedes Wertepaar einzeln aufrufen und jeweils Paare von x-y-Werten übergeben. Die **Zeiteinheit** und das **Kleinste Zeitintervall** sind hier irrelevant. Die Kurve kann hier interaktiv verändert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ startDate	Date/Time	Startdatum
⇒ values	String	y-Werte als Sting
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Werte erfolgreich (True) / nicht erfolgreich (False) gesetzt

Code-Beispiel

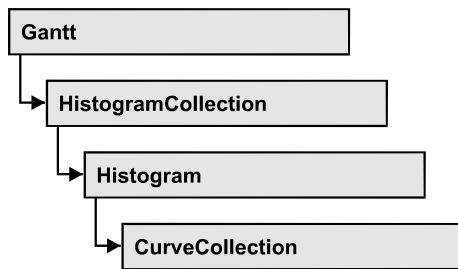
```
Dim histogram As VcHistogram
Dim curve As VcCurve
Dim yValues As String

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curve = histogram.CurveCollection.CurveByName("Curve1")

' If the option Curve points equidistant is checked for the curve:
yValues = "5;1;1;2;2;2;4;5;5;3;2;1;"
curve.SetValues("2012/02/14 12:05:00", yValues)

' If the option Curve points equidistant is not checked for the curve:
curve.SetValues("2012/02/14 12:05:00", "5")
curve.SetValues("2012/02/14 12:07:00", "1")
curve.SetValues("2012/02/14 12:23:00", "1")
curve.SetValues("2012/02/14 13:05:00", "2")
```

7.18 VcCurveCollection



In einem Objekt vom Typ `VcCurveCollection` sind alle Kurven eines Histogramms zusammengefasst. Über **For Each curve In CurveCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Kurven zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **CurveByName** und **CurveByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Kurven kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Kurven.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `CurveByIndex`
- `CurveByName`
- `FirstCurve`
- `NextCurve`
- `Remove`

Eigenschaften

NewEnum

Nur-Lese-Eigenschaft von VcCurveCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Kurvenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim curve As VcCurve

For Each curve In VcGantt1.CurveCollection
    Debug.Print curve.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcCurveCollection

Mit dieser Eigenschaft können Sie die Anzahl der Kurven in der KurvenAuflistung abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Kurven

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection
Dim numberOfCurves As Long

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curveCltn = histogram.CurveCollection

numberOfCurves = curveCltn.Count
```

Methoden

Add

Methode von VcCurveCollection

Mit dieser Methode können Sie eine neue Kurve in der CurveCollection anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Kurvenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ curveName	String Mögliche Werte:	Name der Kurve Name der Farbzuoordnungstabelle
Rückgabewert	VcCurve	Neues Kurvenobjekt

Code-Beispiel

```
Set newCurve = VcGantt1.CurveCollection.Add("test1")
```

AddBySpecification

Methode von VcCurveCollection

Mit dieser Methode können Sie eine Kurve über eine Kurven-Spezifikation erzeugen. Dies dient der Persistenz von Kurvenobjekten. Die Spezifikation einer Kurve kann erfragt (siehe VcCurve-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Kurve mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Kurvenspezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcCurve	Neues Kurvenobjekt

Copy

Methode von VcCurveCollection

Mit dieser Methode können Sie eine Kurve kopieren. Wenn die Kurve mit dem angegebenen Namen existiert und der Name der neuen Kurve noch nicht verwendet wird, wird das neue Kurvenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ curveName	String	Name der zu kopierenden Kurve
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ newCurveName	String	Name der neuen Kurve
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcCurve	Kurvenobjekt

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.FirstHistogram
Set curveCltn = histogram.CurveCollection
Set curve = curveCltn.Copy("CurrentCurve", "NewCurve")
```

CurveByIndex

Methode von VcCurveCollection

Mit dieser Methode können Sie auf eine einzelne Kurve über ihren Index zugreifen. Existiert keine Kurve unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇒ index	Integer	Index der Kurve
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcCurve	Ermitteltes Kurvenobjekt

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection
```

```
Dim curve As VcCurve
```

```
Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curveCltn = histogram.CurveCollection
Set curve = curveCltn.CurveByIndex(2)
```

CurveByName

Methode von VcCurveCollection

Mit dieser Methode können Sie unter Verwendung des Kurvennamens auf eine bestimmte Kurve zugreifen. Existiert keine Kurve unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇨ curveName	String Mögliche Werte:	Name der Kurve Name der Farbzusordnungstabelle
Rückgabewert	VcCurve	Kurve

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curveCltn = histogram.CurveCollection

Set curve = curveCltn.CurveByName("Curve1")
```

FirstCurve

Methode von VcCurveCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. die erste Kurve der Kurvenauflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextCurve** über die nachfolgenden Kurven zu iterieren. Existiert keine Kurve in der Kurvenauflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCurve	Erste Kurve

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection
```

```

Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curveCltn = histogram.CurveCollection

Set curve = curveCltn.FirstCurve

```

NextCurve

Methode von VcCurveCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Kurven der Curve-Auflistung zugreifen, nachdem Sie mit der Methode **FirstCurve** den Initialwert erfasst haben. Sind alle Kurven durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcCurve	Nachfolgende Kurve

Code-Beispiel

```

Dim histogram As VcHistogram
Dim curveCollection As VcCurveCollection
Dim curve As VcCurve

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set curveCollection = histogram.CurveCollection

Set curve = curveCollection.FirstCurve

While Not curve Is Nothing
    Set curve = curveCollection.NextCurve
Wend

```

Remove

Methode von VcCurveCollection

Mit dieser Methode können Sie eine Kurve löschen. Wenn die Kurve noch in irgendeinem anderen Objekt benutzt wird, kann sie nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

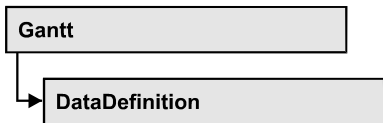
	Datentyp	Beschreibung
Parameter: ⇒ curveName	String	Name der Kurve
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Kurve gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection

Set histogram = VcGantt1.HistogramCollection.FirstHistogram
Set curveCltn = histogram.CurveCollection
curveCltn.Remove ("CurrentCurve")
```

7.19 VcDataDefinition



Die Definition der Daten von Knoten und Verbindungen wird über den Dialog **Datentabellen verwalten**, erreichbar über die Eigenschaftenseite **Objekte: Datentabellen...**, eingestellt. Das Objekt ermöglicht den Zugriff auf die Namen und Typen der verfügbaren Felder. Die Datendefinition eines VcGantt-Objektes enthält zwei Datendefinitionstabellen: **vcMaindata** und **vcRelations**.

Eigenschaften

- DefinitionTable

Eigenschaften

DefinitionTable

Nur-Lese-Eigenschaft von VcDataDefinition

- Diese Eigenschaft ermöglicht den Zugriff auf die beiden Tabellen des Datendefinitionsobjekts: **vcMaindata**: Definitionen der Knotendaten
- **vcRelations**: Definitionen der Verbindungsdaten

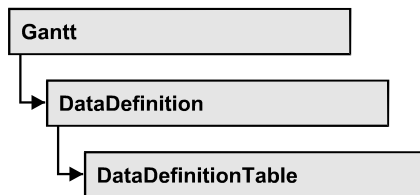
	Datentyp	Beschreibung
Parameter: ⇒ tableType	DataTableEnum Mögliche Werte: vcMaindata 0 vcRelations 1	Typ der Datendefinitionstabelle Tabellentyp vcMaindata (für Knoten) Tabellentyp vcRelations (für Verbindungen)
Eigenschaftswert	VcDataDefinitionTable	Datendefinitionstabelle

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable

Set dataDefinition = VcGantt1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
```

7.20 VcDataDefinitionTable



Ein Objekt vom Typ `VcDataDefinitionTable` ist ein Element der Datendefinition und stellt eine Tabelle aus Datendefinitionsfeldern dar. Auf diese Felder können Sie einzeln über die Methoden **FieldByIndex** oder **FieldByName** zugreifen oder über eine Schleife mit **FirstField** und **NextField** alle Felder abfragen. Über die Eigenschaft **Count** erhalten Sie die Anzahl der Felder. Die Datendefinitionstabelle wird auf der Eigenschaftenseite **Datentabellen verwalten** voreingestellt.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `CreateDataField`
- `FieldByIndex`
- `FieldByName`
- `FirstField`
- `NextField`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcDataDefinitionTable`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datendefinitionsfelder iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim datdeftable As VcDataDefinitionTable

For Each datdeftable In VcGantt1.VcDataDefinition
    Debug.Print datdeftable.Count
Next
```

Count**Nur-Lese-Eigenschaft von VcDataDefinitionTable**

Mit dieser Eigenschaft kann die Anzahl der Felder in der Datendefinitionstabelle abgefragt werden. Anlegen können Sie Datenfelder im Dialog **Datentabellen verwalten** oder zur Laufzeit mit Hilfe der Methode **CreateDataField**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Felder

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim numberOfFields As Long

Set dataDefinition = VcGantt1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)

numberOfFields = dataDefinitionTable.Count
```

Methoden**CreateDataField****Methode von VcDataDefinitionTable**

Mit dieser Methode kann zur Laufzeit ein neues Datenfeld an das Ende der Datendefinitionstabelle angefügt werden. Das neue Datenfeld hat standardmäßig den Datentyp Integer; er kann aber mit Hilfe der Eigenschaft **Type** von **VcDefinitionField** geändert werden.

	Datentyp	Beschreibung
Parameter: ⇒ newfieldName	String Mögliche Werte:	Name des neuen Feldes Name der Farbzuordnungstabelle
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = _VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.CreateDataField("Description")
dataDefinitionField.Type = vcDefFieldAlphanumericType
VcGantt1.DataTableCollection.Update

```

FieldByIndex**Methode von VcDataDefinitionTable**

Mit dieser Methode können Sie über den Index auf ein beliebiges Feld der Datentabelle zugreifen. Jedes Feld kann über seinen Namen oder über seinen Index angesprochen werden. Der Index für das erste Feld ist 0. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	Beschreibung
Parameter: ⇒ fieldIndex	Integer Mögliche Werte:	Index des Feldes Datenfeldindex
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField
Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.FieldByIndex(I)
Next

```

FieldByName

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie unter Verwendung des Feldnamens auf ein bestimmtes Feld der Datentabelle zugreifen. Existiert kein Feld unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**). Jedes Feld kann über seinen Namen oder seinen Index angesprochen werden. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	Beschreibung
Parameter: ⇨ fieldName	String Mögliche Werte:	Feldname Name der Farbzusordnungstabelle
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim definitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set definitionField = dataDefinitionTable.FieldByName("Start")
```

FirstField

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie auf das erste Feld der Datentabelle zugreifen, um anschließend in einer Schleife mit der Methode **NextField** über die nachfolgenden Felder zu iterieren. Existiert kein Feld in der Datentabelle, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Erstes Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FirstField
```

NextField

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Felder der Datentabelle zugreifen, nachdem Sie die Methode **FirstField** aufgerufen haben.

Wenn kein weiteres Feld mehr existiert, wird ein Leerobjekt zurückgegeben (in Visual Basic: Nothing).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Nachfolgendes Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
While Not dataDefinitionField Is Nothing
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Wend

or

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Next
```

7.21 VcDataDefinitionTable

Ein Objekt vom Typ VcDataDefinitionTable ist ein Element der Datendefinition und stellt eine Tabelle aus Datendefinitionsfeldern dar. Auf diese Felder können Sie einzeln über die Methoden **FieldByIndex** oder **FieldByName** zugreifen oder über eine Schleife mit **FirstField** und **NextField** alle Felder abfragen. Über die Eigenschaft **Count** erhalten Sie die Anzahl der Felder. Die Definitionstabelle wird auf der Eigenschaftenseite **Datentabellen verwalten** voreingestellt.

Eigenschaften

- _NewEnum
- Count

Methoden

- CreateDataField
- FieldByIndex
- FieldByName
- FirstField
- NextField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcDataDefinitionTable

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datendefinitionsfelder iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim datdeftable As VcDataDefinitionTable
```



```

For Each datdeftable In VcGantt1.VcDataDefinition
    Debug.Print datdeftable.Count
Next

```

Count

Nur-Lese-Eigenschaft von VcDataDefinitionTable

Mit dieser Eigenschaft kann die Anzahl der Felder in der Datendefinitionstabelle abgefragt werden. Anlegen können Sie Datenfelder im Dialog **Datentabellen verwalten** oder zur Laufzeit mit Hilfe der Methode **CreateDataField**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Felder

Code-Beispiel

```

Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim numberOfFields As Long

Set dataDefinition = VcGantt1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)

numberOfFields = dataDefinitionTable.Count

```

Methoden

CreateDataField

Methode von VcDataDefinitionTable

Mit dieser Methode kann zur Laufzeit ein neues Datenfeld an das Ende der Datendefinitionstabelle angefügt werden. Das neue Datenfeld hat standardmäßig den Datentyp Integer; er kann aber mit Hilfe der Eigenschaft **Type** von **VcDefinitionField** geändert werden.

	Datentyp	Beschreibung
Parameter: ⇒ newfieldName	String Mögliche Werte:	Name des neuen Feldes Name der Farbzusordnungstabelle
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = _ VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.CreateDataField("Description")
dataDefinitionField.Type = vcDefFieldAlphanumericType
VcGantt1.DataTableCollection.Update

```

FieldByIndex**Methode von VcDataDefinitionTable**

Mit dieser Methode können Sie über den Index auf ein beliebiges Feld der Datentabelle zugreifen. Jedes Feld kann über seinen Namen oder über seinen Index angesprochen werden. Der Index für das erste Feld ist 0. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	Beschreibung
Parameter: ⇒ fieldIndex	Integer Mögliche Werte:	Index des Feldes Datenfeldindex
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField
Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.FieldByIndex(I)
Next

```

FieldByName**Methode von VcDataDefinitionTable**

Mit dieser Methode können Sie unter Verwendung des Feldnamens auf ein bestimmtes Feld der Datentabelle zugreifen. Existiert kein Feld unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**). Jedes Feld kann über seinen Namen oder seinen Index angesprochen werden. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	Beschreibung
Parameter: ⇒ fieldName	String Mögliche Werte:	Feldname Name der Farbzusordnungstabelle
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim definitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set definitionField = dataDefinitionTable.FieldByName("Start")

```

FirstField**Methode von VcDataDefinitionTable**

Mit dieser Methode können Sie auf das erste Feld der Datentabelle zugreifen, um anschließend in einer Schleife mit der Methode **NextField** über die nachfolgenden Felder zu iterieren. Existiert kein Feld in der Datentabelle, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Erstes Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FirstField

```

NextField**Methode von VcDataDefinitionTable**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Felder der Datentabelle zugreifen, nachdem Sie die Methode **FirstField** aufgerufen haben.

Wenn kein weiteres Feld mehr existiert, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Nachfolgendes Datendefinitionsfeld

Code-Beispiel

```

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
While Not dataDefinitionField Is Nothing
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Wend

or

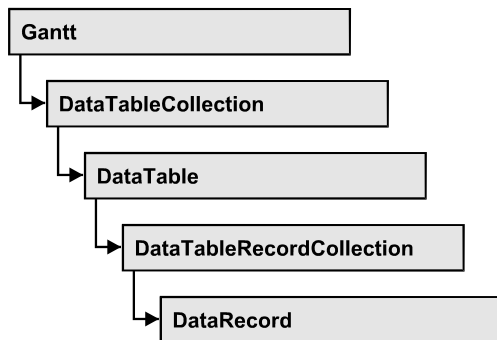
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Next

```

7.22 VcDataRecord



Ein Datensatz ist das logische Grundelement eines Objektes in einem Gantt Diagram, z.B. eines Knotens, eines Gruppenknotens, einer Verbindung, einer Aufgabe, Operation etc. Die Objekte besitzen spezifische Eigenschaften, die in den Feldern des Datensatzes beschrieben werden. Zu den Datenfeldern des Datensatzes existieren entsprechende Beschreibungen, die Datentabellenfelder. Datensätze und Datentabellenfelder werden jeweils zu Collection-Objekten zusammengefasst und bilden eine Datentabelle.

Eigenschaften

- AllData
- DataField
- DataTableName
- ID

Methoden

- DeleteDataRecord
- IdentifyObject
- RelatedDataRecord
- UpdateDataRecord

Eigenschaften

AllData

Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können alle Daten eines Datensatzes gesetzt oder erfragt werden. Beim Setzen ist ein CSV-String (Semikolon als Trennzeichen) oder der Datentyp "Variant" erlaubt, der in einem Array alle

Datenfelder des Datensatzes erhält. Beim Erfragen wird eine Zeichenkette (String) zurückgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Variant	Alle Daten des Datensatzes

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRecValue() As Variant
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata1")
Set dataRecCltn = dataTable.DataRecordCollection
ReDim dataRecValue(dataTable.DataTableFieldCollection.Count)
dataRecValue(0) = 1
dataRecValue(1) = "Node One"

'Variant
Set dataRecord = dataRecCltn.Add(dataRecValue)
'CSV
dataRecord.AllData = "1;Node One;"

dataRecord.UpdateDataRecord
```

DataField

Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie einem Datenfeld des Datensatzes einen Wert zuweisen oder einen gesetzten Wert erfragen. Wenn ein Datensatz durch diese Methode einen neuen Wert erhalten hat, muss anschließend die grafische Darstellung mit der Methode **UpdateDataRecord** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Datenfeldes
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)
```

```
dataRecord.DataField(1) = "Node Two"
dataRecord.UpdateDataRecord
```

DataTableName

Nur-Lese-Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie den Namen der Datentabelle erfragen, zu der dieser Datensatz gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der zugehörigen Tabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

MsgBox dataRecord.DataTableName
```

ID

Nur-Lese-Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie die ID eines Datensatzes erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datensatz-ID
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

MsgBox dataRecord.ID
```

Methoden

DeleteDataRecord

Methode von VcDataRecord

Mit dieser Methode können Sie einen Datensatz löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Datensatz erfolgreich (true) / nicht erfolgreich (false) gelöscht

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

dataRecord.DeleteDataRecord
```

IdentifyObject

Methode von VcDataRecord

Mit dieser Methode kann man erfragen, ob und welches datenbasierte Objekt aus dem Datensatz erzeugt wurde.

Die Methode liefert als Rückgabewert **true**, wenn ein datenbasiertes Objekt ermittelt werden konnte, d.h. wenn aus dem Datensatz für die Grafik ein datenbasiertes Objekt hergestellt wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ establishedObject Param	Object	Erkanntes Objekt
establishedObjectTypeParam	VcObjectTypeEnum	Typ des erkannten Objekts
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeCalendarGrid 18 vcObjTypeCurve 12 vcObjTypeDateLine 9 vcObjTypeGroup 7 vcObjTypeGroupInDiagram 11 vcObjTypeGroupInTable 7 vcObjTypeHistogram 13 vcObjTypeLayer 8 vcObjTypeLinkCollection 3	Objektyp Box Objektyp Kalendergitter Objektyp Kurve Objektyp Stichtaglinie Objektyp Gruppe Objektyp Gruppe im Knotenbereich Objektyp Gruppe im Tabellenbereich Objektyp Histogramm Objektyp Layer Objektyp LinkCollection

	vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14 vcObjTypeTable 4 vcObjTypeTableCaption 5 vcObjTypeTimeScale 6	Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Knoten im Tabellenbereich kein Objekt Objekttyp Werteskala Objekttyp Summenbalken Objekttyp Tabelle Objekttyp Tabellenüberschrift Objekttyp Zeitskala
Rückgabewert	Boolean	datenbasiertes Objekt wurde/wurde nicht erzeugt

RelatedDataRecord

Methode von VcDataRecord

Mit dieser Eigenschaft können Sie einem Datensatz einen weiteren zuordnen oder einen zugeordneten Datensatz erfragen. Bei der Verwendung von erweiterten Tabellen (extended data tables) können Datensätze einer Tabelle über einen Primärschlüssel den Datensätzen einer anderen Tabelle zugeordnet werden.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Datenfeldes Datenfeldindex
Rückgabewert	VcDataRecord	Zugeordneter Datensatz

Code-Beispiel

```

Private Sub VcGantt1_OnNodeLClick(ByVal node As VcGanttLib.VcNode, ByVal
location As VcGanttLib.LocationEnum, ByVal x As Long, ByVal y As Long,
returnStatus As Variant)

    Dim dataTable As VcDataTable
    Dim dataRecordCltn As VcDataRecordCollection
    Dim firstDataRecord As VcDataRecord
    Dim secondDataRecord As VcDataRecord

    Set dataTable = VcGantt1.DataTableCollection.DataTableByIndex(0)
    Set dataRecordCltn = dataTable.DataRecordCollection

    Set firstDataRecord = dataRecordCltn.DataRecordByID(node.DataField(0))
    Set secondDataRecord = firstDataRecord.RelatedDataRecord(2)

    MsgBox secondDataRecord.AllData

End Sub

```

UpdateDataRecord

Methode von VcDataRecord

Nachdem Sie ein oder mehrere Datenfelder eines Datensatzes mit der Eigenschaft **DataField** verändert haben, aktualisieren Sie die grafische Darstellung im Diagramm mit **UpdateDataRecord**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Datensatz erfolgreich (true) / nicht erfolgreich (false) aktualisiert

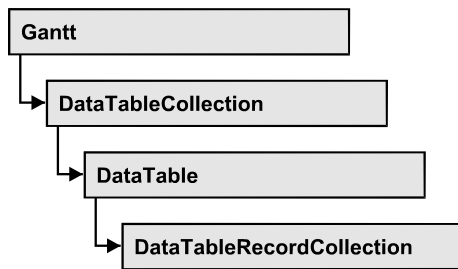
Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

dataRecord.DataField(1) = "Node Two"
dataRecord.UpdateDataRecord
```

7.23 VcDataRecordCollection



In einem Objekt vom Typ **VcDataRecordCollection** sind die Datensätze einer Datentabelle zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Datensätze im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataRecord** und **NextDataRecord** können Sie iterativ auf die Datensätze zugreifen sowie mit **DataRecordByID** auf einzelne Datensätze; die Methoden **Add** und **Remove** ermöglichen das Hinzufügen und Entfernen von Datensätzen und mit **Update** können Sie die grafische Darstellung der Datensätze mit neu eingegebenen Daten aktualisieren.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `DataRecordByID`
- `FirstDataRecord`
- `GetNewUniqueID`
- `NextDataRecord`
- `Remove`
- `Update`

Eigenschaften

`_NewEnum`

Eigenschaft von **VcDataRecordCollection**

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle ent-

haltenen Datensätze iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

For Each dataRecord In dataRecordCltn
    Debug.Print dataRecord.AllData
Next dataRecord
```

Count

Nur-Lese-Eigenschaft von VcDataRecordCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datensätze in der DataRecord-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datensätze im Collection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
MsgBox "Number of DataRecords: " & dataRecordCltn.Count
```

Methoden

Add

Methode von VcDataRecordCollection

Mit dieser Methode können Sie einen neuen Datensatz in der DataRecordCollection anlegen. Wenn die Datensatzbeschreibung die Anlage eines neuen

Datensatzes erlaubt, wird der neue Datensatz zurückgegeben, andernfalls wird eine **VcPrimaryKeyNotUniqueException** erzeugt.

Nach dem Hinzufügen des Datensatzes muss die Methode **VcGantt.EndLoading** aufgerufen werden, damit die Änderung wirksam wird.

	Datentyp	Beschreibung
Parameter: ⇒ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)
Rückgabewert	VcDataRecord	Neue angelegter Datensatz

Code-Beispiel

```
Const Main_ID = 0
Const Main_Name = 1
Const Main_Start = 2
Const Main_Duration = 4

' ...

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim dataRecVal() As Variant

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection

ReDim dataRecVal(dataTable.DataTableFieldCollection.Count)

dataRecVal(Main_ID) = 1
dataRecVal(Main_Name) = "Node 1"
dataRecVal(Main_Start) = DateSerial(2014, 1, 8)
dataRecVal(Main_Duration) = 8
Set dataRec1 = dataRecCltn.Add(dataRecVal)
VcGantt1.EndLoading()

' equivalent
' dataRec1 = dataRecCltn.Add("1;Node 1;01.08.14;;8")
```

DataRecordByID

Methode von VcDataRecordCollection

Mit dieser Methode können Sie auf einen einzelnen Tabellendatensatz über seine Identifikation zugreifen. Existiert kein Datensatz unter der angegebenen ID, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter: ⇒ dataRecordID	String	ID des Datensatzes
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcDataRecord	Datensatzobjekt

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(0)

```

FirstDataRecord**Methode von VcDataRecordCollection**

Mit dieser Methode können Sie auf den ersten Datensatz der Datenstz-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataRecord** über die nachfolgenden Datensätze zu iterieren. Existiert kein Datensatz in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Erster Datensatz

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.FirstDataRecord

```

GetNewUniqueID**Methode von VcDataRecordCollection**

Mit dieser Methode können Sie eine eindeutige ID für einen Datensatz generieren lassen. Die Methode ist nützlich, wenn Sie z.B. mit **Add** einen neuen Datensatz generieren und die ID nicht manuell vergeben.

	Datentyp	Beschreibung
Rückgabewert	Long	Neue Datensatz-ID

NextDataRecord

Methode von VcDataRecordCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datensätze der Datensatzauflistung zugreifen, nachdem Sie mit der Methode **FirstDataRecord** den Initialwert erfasst haben. Sind alle Datensätze durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Nachfolgender Datensatz

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcGantt1.SuspendUpdate True

Set dataRecord = dataRecordCltn.FirstDataRecord
While Not dataRecord Is Nothing
    dataRecord.DataField(4) = "10"
    dataRecord.UpdateDataRecord
    Set dataRecord = dataRecordCltn.NextDataRecord
Wend

VcGantt1.SuspendUpdate False
```

Remove

Methode von VcDataRecordCollection

Mit dieser Methode können Sie einen Datensatz löschen. Die Methode liefert **true** wenn gelöscht wurde und **false**, wenn nicht gelöscht wurde. Der Datensatzinhalt im Übergabeparameter wird dazu verwendet, um anhand der Identifizierung das Objekt zu finden.

	Datentyp	Beschreibung
Parameter: ⇒ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)
Rückgabewert	Boolean	True

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcGantt1.SuspendUpdate True

Set dataRecord = dataRecordCltn.FirstDataRecord
While Not dataRecord Is Nothing
    dataRecord.DataField(4) = "10"
    dataRecord.UpdateDataRecord
    Set dataRecord = dataRecordCltn.NextDataRecord
Wend

VcGantt1.SuspendUpdate False
VcGantt1.EndLoading()

```

Update**Methode von VcDataRecordCollection**

Mit dieser Methode können Sie einen Datensatz in der Datensatzliste aktualisieren, nachdem mittels der Methode **Add()** der Datensatz vorher hinzugefügt wurde. Falls der zu aktualisierende Datensatz nicht existiert, d.h. also nicht vorher angelegt wurde, wird er mittels Update() nun neu angelegt. Siehe auch **VcDataRecordCollection.Add()**.

Nach dem Aktualisieren des Datensatzes muss die Methode **VcGantt.EndLoading** aufgerufen werden, damit die Änderung wirksam wird.

	Datentyp	Beschreibung
Parameter: ⇒ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)

Code-Beispiel

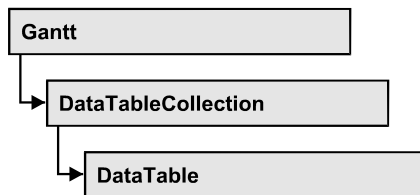
```

Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
dataRecordCltn.Update("1;1.8.2017;;8")
VcGantt1.EndLoading()

```


7.24 VcDataTable



Eine Datentabelle umfasst **Datensätze** (data records) mit ihren Datenfeldern und ihren Inhalten sowie die Beschreibungen der Datenfelder, die **Tabellendatenfelder** (data table fields) genannt werden. Datensätze und Datentabellenfelder können in der Form von eigenen Auflistungen (collection) verwaltet werden.

Datentabellen ihrerseits können ebenfalls in eigenen Auflistungen verwaltet werden.

Eigenschaften

- DataRecordCollection
- DataTableFieldCollection
- Description
- MultiplePrimaryKeysAllowed
- Name

Eigenschaften

DataRecordCollection

Nur-Lese-Eigenschaft von VcDataTable

Diese Eigenschaft gibt die in der Datentabelle enthaltene Datensatz-Auflistung zurück. Die Datensatz-Auflistung enthält alle existierenden Datensätze einer Tabelle. Zu Programmbeginn ist sie leer.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataRecordCollection	DataRecordCollection-Objekt

Code-Beispiel

```

Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable()
MsgBox dataTable.DataRecordCollection.Count
  
```

DataTableFieldCollection

Nur-Lese-Eigenschaft von VcDataTable

Diese Eigenschaft gibt die in der Datentabelle enthaltene Datentabellenfeld-Auflistung zurück. Die Datentabellenfeld-Auflistung enthält die Definition der Datenfelder eines Datensatzes der Tabelle. Zu Programmbeginn enthält sie die bereits zur Designzeit vereinbarten Datenfelder. Weitere Datenfelder können zur Laufzeit über die Methode **Add** des Objekts **DataTableFieldCollection** hinzugefügt werden. Die Definition der Datentabellenfelder muss abgeschlossen sein, bevor die Tabelle mit Datensätzen gefüllt wird.

	Datentyp	Beschreibung
Eigenschaftswert	VcTableFieldCollection	DataTableFieldCollection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.DataTableByIndex(0)
MsgBox dataTable.DataTableFieldCollection.Count
```

Description

Eigenschaft von VcDataTable

Mit dieser Eigenschaft können sie eine Beschreibung der Datentabelle setzen oder erfragen. Sprechende Namen, z.B. der Name der Tabelle, sind häufig sehr lang und werden daher bei Previews nicht vollständig angezeigt, so dass ihr Nutzen nicht zum Tragen kommen kann. Damit Sie für die vollständige Anzeige kurze Namen verwenden können und trotzdem nicht auf die gewünschte Information verzichten müssen, können Sie in diesem Feld zusätzliche Informationen zum Tabellennamen speichern. Sein Inhalt wird im Dialog zur Datentabelle angezeigt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Beschreibung der Datentabelle Standardwert: Leere Zeichenkette
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
dataTable.Description = "This table contains data for nodes"
```

MultiplePrimaryKeysAllowed

Eigenschaft von VcDataTable

Mit dieser Eigenschaft können Sie angeben oder erfragen, ob die Verwendung zusammengesetzter Primärschlüssel möglich ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verwendung von zusammengesetzten Primärschlüsseln erlaubt (true)/nicht erlaubt (false) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Name

Eigenschaft von VcDataTable

Mit dieser Eigenschaft können sie den Namen der Datentabelle setzen oder erfragen. Ein Tabellename ist obligat und muss eindeutig sein; zudem ist eine leere Zeichenkette nicht erlaubt. Unterscheidungen in der Groß- und Kleinschreibung führen zu unterschiedlichen Namen. Mit der Methode **DataTableByName** des Objekts **DataTableCollection** können Sie über den Tabellennamen eine Referenz auf das Datentabellenobjekt erhalten.

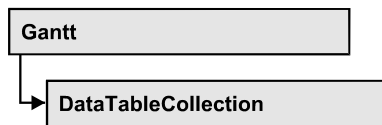
	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle Standardwert: Leere Zeichenkette
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.DataTableByIndex(0)
MsgBox dataTable.Name
```

7.25 VcDataTableCollection



In einem Objekt vom Typ `VcDataTableCollection` sind die vorhandenen Datentabellen zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Tabellen im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataTable** und **NextDataTable** können Sie iterativ auf die Tabellen zugreifen sowie mit **DataTableByName** und **DataTableByIndex** auf einzelne Tabellen; die Methoden **Add** und **Copy** ermöglichen das Hinzufügen und Kopieren von Tabellen und mit **Update** können Sie dem XGantt-Objekt die neuen Änderungen der Datenstrukturen bekanntgeben, das heißt aktualisieren.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `Copy`
- `DataTableByIndex`
- `DataTableByName`
- `FirstDataTable`
- `NextDataTable`
- `Update`

Eigenschaften

`_NewEnum`

Eigenschaft von `VcDataTableCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts können Sie über alle enthaltenen Datentabellen iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator**

angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
For Each dataTable In dataTableCltn
    List1.AddItem (dataTable.Name)
Next
```

Count

Eigenschaft von VcDataTableCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datentabellen in der DataTable-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datentabellen im Collection-Objekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection

Set dataTableCltn = VcGantt1.DataTableCollection
MsgBox (dataTableCltn.Count)
```

Methoden

Add

Methode von VcDataTableCollection

Mit dieser Methode können Sie eine neue Datentabelle in der DataTable-Auflistung anlegen. Wenn der Tabellename noch nicht verwendet wurde, wird ein Objekt vom Typ **VcDataTable** zurückgegeben, andernfalls **Nothing** (in Visual Basic) oder **0** (in anderen Sprachen). Nur wenn die Eigenschaft **ExtendedDataTables** bei dem Objekt VcGantt auf **True** gesetzt ist, können Tabellen angelegt werden. Maximal 90 Datentabellen sind insgesamt in der Auflistung möglich.

	Datentyp	Beschreibung
Parameter: ⇒ dataTableName	String	Name der Datentabelle
	Mögliche Werte:	Name der Farbzuordnungstabelle
Rückgabewert	VcDataTable	Neu angelegte Datentabelle

Code-Beispiel

```

Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.Add("Resources")
dataTableCltn.Update

```

Copy**Methode von VcDataTableCollection**

Mit dieser Methode können Sie eine Datentabelle kopieren. Es wird nur die Tabellendefinition kopiert, jedoch nicht die eventuell bereits existierende Datensätze. Nur wenn die Eigenschaft **ExtendedDataTables** bei dem Objekt VcNet auf **True** gesetzt ist, können Tabellen angelegt werden. Wenn die Tabelle kopiert werden konnte, wird ein neues Objekt vom Typ **VcDataTable** zurückgegeben, andernfalls **Nothing** (in Visual Basic) oder **0** (in anderen Sprachen). Es wird bei den Tabellennamen generell zwischen Groß- und Kleinschreibung unterschieden.

	Datentyp	Beschreibung
Parameter: ⇒ dataTableName	String	Name der zu kopierenden Datentabelle (Quelltabelle)
		Mögliche Werte:
	String	Name der Farbzuordnungstabelle
		Mögliche Werte:
⇒ newDataTableName	String	Name der neu zu erstellenden Datentabelle (Zieltabelle)
		Mögliche Werte:
		Name der Farbzuordnungstabelle
Rückgabewert	VcDataTable	Neu erstelltes Datentabellen-Objekt

Code-Beispiel

```

Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.Copy("Resources", "NewResources")

```

```
dataTableCltn.Update
```

DataTableByIndex

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf eine einzelne Datentabelle über ihren Index zugreifen. Der Index der ersten Tabelle ist 0. Existiert keine Tabelle unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (**Nothing** in Visual Basic oder **0** in anderen Sprachen).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Datentabelle Datenfeldindex
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByIndex(2)
MsgBox (dataTable.Name)
```

DataTableByName

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf eine einzelne Datentabelle über ihren Namen zugreifen. Existiert keine Tabelle unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (**Nothing** in Visual Basic oder **0** in anderen Sprachen).

	Datentyp	Beschreibung
Parameter: ⇒ dataTableName	String Mögliche Werte:	Name der Tabelle Name der Farbzusordnungstabelle
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable
```

```

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByName("Resources")
MsgBox (dataTable.Description)

```

FirstDataTable

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf die erste Datentabelle des DataTable-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataTable** über die nachfolgenden Tabellen zu iterieren. Existiert keine Datentabelle in der DataTable-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataTable	Erste Datentabelle

Code-Beispiel

```

Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.FirstDataTable

```

NextDataTable

Methode von VcDataTableCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datentabellen der DataTable-Auflistung zugreifen, nachdem Sie mit der Methode **FirstDataTable** den Initialwert erfasst haben. Sind alle Tabellen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataTable	Nachfolgende Datentabelle

Code-Beispiel

```

Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable
Dim i As Integer

Set dataTableCltn = VcGantt1.DataTableCollection
Set dataTable = dataTableCltn.FirstDataTable
For i = 0 To dataTableCltn.Count
    List1.AddItem (dataTable.Name)
    Set dataTable = dataTableCltn.NextDataTable
Next i

```


Update

Methode von VcDataTableCollection

Mit dieser Methode können Sie neue Änderungen an Datenstrukturen aktualisieren. Der Aufruf ist notwendig, damit durchgeführte Änderungen an der Datentabellendefinition und an den Datentabellenfeldern in der VARCHART Komponente wirksam werden. Auf diese Weise werden bei mehreren Änderungen unnötige Aktualisierungen vermeiden.

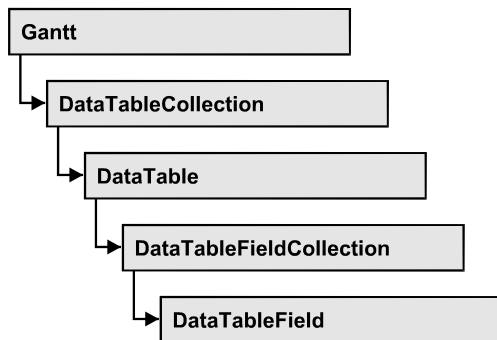
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

dataTableCltn = VcGantt1.DataTableCollection
dataTable = dataTableCltn.Add("Resources")
dataTable.DataTableFieldCollection.Add ("Id")
dataTableCltn.Update
```

7.26 VcDataTableField



Ein Objekt vom Typ **VcDataTableField** legt die Eigenschaften eines Feldes der Datentabelle fest. Zur Definition eines Datentabellenfeldes gehört der Name, der Datentyp und die Festlegung, ob das Datenfeld als Primärschlüssel dient, der zur eindeutigen Identifizierung eines Datensatzes herangezogen werden kann. Unter Verwendung des Primärschlüssels z.B. kann von anderen Datentabellen auf diese Datentabelle Bezug genommen werden. Um eine Beziehung zu einer Datentabelle mit Primärschlüssel aufzubauen, muss der Primärschlüssel im Feld **RelationshipFieldIndex** benannt werden.

Die DataTableField-Objekte einer Datentabelle werden über die Auflistung **DataTableFieldCollection** verwaltet.

Eigenschaften

- DataTableName
- DateFormat
- Editable
- Hidden
- Index
- Name
- PrimaryKey
- RelationshipFieldIndex
- Type

Eigenschaften

DataTableName

Nur-Lese-Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Namen der zugehörigen Datentabelle erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
MsgBox dataTable.DataTableFieldCollection.FirstDataTableField.DataTableName
```

DateFormat

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie das Datumsformat des Datentabellenfeldes festlegen oder erfragen. Das Datumsformat wird beim Lesen und Schreiben von CSV-Dateien verwendet und wenn der Formattyp **String** beim Hinzufügen eines Datensatzes über die Methode **Add** zur Datensatzauflistung verwendet wird. Diese Eigenschaft ist nur wirksam, wenn der Datentyp des Feldes auf **vcDataTableFieldDateTime** eingestellt ist.

Hinweis: Es sollte zuerst die Eigenschaft **Type** gesetzt werden, bevor die Eigenschaft **DateFormat** vereinbart wird.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat {DMYhms;.;/} Standardwert: DD.MM.YYYY hh:mm:ss
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
```

```

Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Type = vcDataTableFieldDateTimeType
'DateFormat = "01.12.2014"
dataTableField.DateFormat = "DD.MM.YYYY"

```

Editable

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das spezifizierte Datenfeld zur Laufzeit in der Tabelle (des Diagramms) und des Dialogs **Knoten bearbeiten** editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Feld editierbar (True) / nicht editierbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Editable = False
VcGantt1.DataTableCollection.Update

```

Hidden

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Datenfeld zur Laufzeit in den Dialogen **EditNode** oder **EditLink** angezeigt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Feld wird nicht angezeigt (True) / angezeigt (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Hidden = True
VcGantt1.DataTableCollection.Update

```

Index**Nur-Lese-Eigenschaft von VcDataTableField**

Mit dieser Eigenschaft können Sie den Index des Datentabellenfelds in der zugehörigen Datentabelle erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datentabellenfeldes.
	Mögliche Werte:	Datenfeldindex

Name**Eigenschaft von VcDataTableField**

Mit dieser Eigenschaft können Sie den Namen des Datenfelds setzen oder erfragen. Der Name des Datenfelds erscheint innerhalb von Laufzeitdialogen wie beispielsweise dem **EditNode**-Dialog. In der API erfolgt der Zugriff auf die Datenfeldinhalte eines Datensatzes jedoch immer über den Index, den dieses Feld im **DataTableFieldCollection**-Objekt besitzt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Feldes Standardwert: Empty string
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField = dataTable.DataTableFieldCollection.Add("Start")
VcGantt1.DataTableCollection.Update

```

PrimaryKey

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob dieses Datenfeld den Primärschlüssel enthält, der zur eindeutigen Identifikation eines Datensatz herangezogen werden kann. In einer Datentabelle kann immer nur ein Datenfeld die Kennung Primärschlüssel tragen. Die aktuelle Setzung hebt eine eventuell vorhandene Setzung bei einem anderen Datenfeld der gleichen Tabelle auf. Die Festlegung eines Primärschlüssels ist unerlässlich, wenn eine Beziehung von einer anderen Tabelle zu dieser Tabelle hergestellt werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Das Feld dient (True) / dient nicht (False) als Primärschlüssel Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField
Dim isPrimaryKey As Boolean

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Id")
dataTableField.PrimaryKey = True
VcGantt1.DataTableCollection.Update
```

RelationshipFieldIndex

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft verbinden Sie ein Datenfeld und seine Beschreibung. Dazu setzen Sie hier den Index des Datensatzfeldes, auf welches sich die gesetzten Eigenschaften des Datentabellenfeldes beziehen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datensatzfeldes, auf das sich die Datendefinition des Datentabellenfeldes bezieht. Standardwert: -1

Code-Beispiel

```
Dim dataTableTask As VcDataTable
Dim dataTaskFieldId As VcDataTableField
Dim dataTaskFieldName As VcDataTableField
```

650 API-Referenz: VcDataTableField

```
Dim dataTableOperation As VcDataTable
Dim dataOperationFieldId As VcDataTableField
Dim dataOperationFieldName As VcDataTableField
Dim dataOperationFieldTaskId As VcDataTableField

'Create table Task
dataTableTask = VcGantt1.DataTableCollection.Add("Task")
dataTaskFieldId = dataTableTask.DataTableFieldCollection.Add("Id")
dataTaskFieldId.PrimaryKey = True
dataTaskFieldName = dataTableTask.DataTableFieldCollection.Add("Name")
dataTaskFieldName.Type = vcDataTableFieldAlphanumericType

'Create table Operation
dataTableOperation = VcGantt1.DataTableCollection.Add("Operation")
dataOperationFieldId = dataTableOperation.DataTableFieldCollection.Add("Id")
dataOperationFieldId.PrimaryKey = True
dataOperationFieldName = dataTableOperation.DataTableFieldCollection.Add("Name")
dataOperationFieldName.Type = vcDataTableFieldAlphanumericType
dataOperationFieldTaskId =
dataTableOperation.DataTableFieldCollection.Add("TaskId")
dataOperationFieldTaskId.Type = vcDataTableFieldIntegerType

'Link tables Task and Operations
dataOperationFieldTaskId.RelationshipFieldIndex =
VcGantt1.DetectFieldIndex("Task", "Id")
```

Type

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Datentyp des Feldes setzen oder erfragen.

Hinweis: Durch Setzen der Eigenschaft **Type** kann sich die Eigenschaft **DateFormat** ändern. Durch Setzen des Eigenschaftswertes auf **vcDataTableAlphanumeric** oder **vcDataTableFieldInteger** wird ein eventuell eingestelltes Datumsformat auf "" gesetzt.

	Datentyp	Beschreibung
Eigenschaftswert	DataTableFieldTypeEnum	Datentyp des Feldes, kann maximal 512 Zeichen enthalten Standardwert: VcDataTableFieldIntegerType
	Mögliche Werte: vcDataTableFieldAlphanumericType 1 vcDataTableFieldDateTimeType 3 vcDataTableFieldIntegerType 2	Datentyp Alphanumerisch: "" Datentyp Datum: DD.MM.YYYY Datentyp Integer (32 Bit): ""

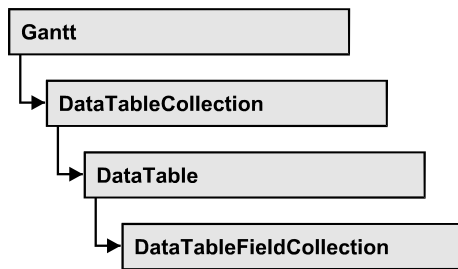
Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
```

```
dataTableField.Type = vcDataTableFieldDateTimeType  
VcGantt1.DataTableCollection.Update
```


7.27 VcDataTableFieldCollection



In einem Objekt vom Typ `VcDataTableFieldCollection` sind die Datentabellenfelder einer Datentabelle zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Felder im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataField** und **NextDataField** können Sie iterativ auf die Felder zugreifen sowie mit **DataFieldByName** und **DataFieldByIndex** auf einzelne Felder; die Methoden **Add** und **Copy** ermöglichen das Hinzufügen und Kopieren von Feldern.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `Copy`
- `DataTableFieldByIndex`
- `DataTableFieldByName`
- `FirstDataTableField`
- `NextDataTableField`

Eigenschaften

`_NewEnum`

Eigenschaft von `VcDataTableFieldCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datentabellenfelder iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator**

angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
For Each dataTableField In dataTable.DataTableFieldCollection
    List1.AddItem (dataTableField.Name)
Next
```

Count

Nur-Lese-Eigenschaft von VcDataTableFieldCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datentabellenfelder in der DataTableField-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datentabellenfelder im Collection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
MsgBox ("Number of data fields: " & dataTable.DataTableFieldCollection.Count)
```

Methoden

Add

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie ein neues Datentabellenfeld in der DataTableFieldCollection anlegen. Wenn der Name noch nicht verwendet wurde, wird das neue Datenfeld zurückgegeben, andernfalls "Nothing" (Visual Basic) oder "0" (andere Sprachen). Maximal 9.999 Datenfelder können pro Tabelle angelegt werden.

	Datentyp	Beschreibung
Parameter: ⇒ dataTableFieldName	String Mögliche Werte:	Name des neuen Datentabellenfeldes Name der Farbzusordnungstabelle
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Add("Priority")
VcGantt1.DataTableCollection.Update

```

Copy**Methode von VcDataTableFieldCollection**

Mit dieser Methode können Sie ein Datentabellenfeld kopieren. Die Identifizierung des Feldes erfolgt über den Namen.

	Datentyp	Beschreibung
Parameter: ⇒ dataTableFieldName	String Mögliche Werte:	Name des zu kopierenden Datentabellenfeldes (Quellfeld) Name der Farbzusordnungstabelle
⇒ newDataTableFieldName	String Mögliche Werte:	Name des neu zu erstellenden Datentabellenfeldes (Zielfeld) Name der Farbzusordnungstabelle
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Copy("Name", "NewName")
VcGantt1.DataTableCollection.Update

```

DataTableFieldByIndex

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf ein einzelnes Datentabellenfeld über seinen Index zugreifen. Existiert kein Feld an dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ Index	Integer Mögliche Werte:	Index des Datentabellenfeldes Datenfeldindex
Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.DataTableFieldByIndex(1)
MsgBox (dataTableField.Name)
```

DataTableFieldByName

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf ein einzelnes Datentabellenfeld über seinen Namen zugreifen. Existiert kein Feld unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ dataTableFieldName	String Mögliche Werte:	Name des Datentabellenfeldes Name der Farbzusordnungstabelle
Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.DataTableFieldBy("Name")
dataTableField.Editable = False
VcGantt1.DataTableCollection.Update
```

FirstDataTableField

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf das erste Datenfeld der DataTableField-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataTableField** über die nachfolgenden Datenfelder zu iterieren. Existiert kein Datenfeld in der DataTableField-Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataTableField	Erstes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.FirstDataTableField
```

NextDataTableField

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datentabellenfelder der DataTableField-Auflistung zugreifen, nachdem Sie mit der Methode **FirstDataTableField** den Initialwert erfasst haben. Sind alle Felder durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

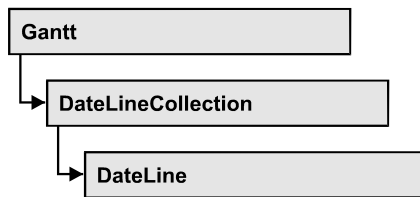
	Datentyp	Beschreibung
Rückgabewert	VcDataTableField	Nachfolgendes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableFieldCltn As VcDataTableFieldCollection
Dim dataTableField As VcDataTableField
Dim i As Integer

Set dataTable = VcGantt1.DataTableCollection.FirstDataTable
Set dataTableFieldCltn = dataTable.DataTableFieldCollection
Set dataTableField = dataTableFieldCltn.FirstDataTableField
For i = 0 To dataTableFieldCltn.Count
    List1.AddItem (dataTableField.Name)
    Set dataTableField = dataTableFieldCltn.NextDataTableField
Next i
```

7.28 VcDateLine



Ein Objekt vom Typ VcDateLine stellt eine Terminstichwertlinie dar. Eine Terminstichwertlinie ist eine vertikale Stichwertlinie im Balkendiagramm zur Kennzeichnung eines Termins.

Eigenschaften

- AlwaysCurrentDate
- Date
- DateDataFieldIndex
- Font
- FontColor
- Identifiable
- LabelPosition
- LineColor
- LineThickness
- LineType
- Moveable
- Name
- Priority
- SnapTarget
- Specification
- Text
- TurningAnnotationEnabled
- UpdateBehaviorName
- Visible
- VisibleDataFieldIndex
- VisibleMapName

Methoden

- PutInOrderAfter

Eigenschaften

AlwaysCurrentDate

Nur-Lese-Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie erfragen, ob die Stichtaglinie immer das aktuelle Datum (mit Uhrzeit) zur Zeit des Aufrufs des VARCHART ActiveX anzeigt. Diese Eigenschaft können Sie auch im Dialog **Stichtaglinien festlegen** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dateLine As VcDateLine
Dim DateLineTimer As Timer

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")

If dateLine.AlwaysCurrentDate = True Then DateLineTimer.Enabled = True
```

Date

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie die Position der Stichtaglinie setzen oder erfragen. Bitte beachten Sie, dass zwischen Datum und Uhrzeit ein Leerzeichen stehen muss. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Datum {1.1.1970...31.12.2035} Standardwert: none or current date

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
dateLine.Date = "30.09.14 12:00:00"
```

DateDataFieldIndex

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das Datum für eine individuelle Stichtaglinie enthält.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das Datum enthält

Font

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie die Schriftattribute einer Stichtaglinie erfragen oder festlegen. Diese Eigenschaft können Sie auch im Dialog **Stichtaglinien bearbeiten** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute der Stichtaglinientexte

Code-Beispiel

```
Dim newFont As New StdFont

newFont.Name = "Times New Roman"
newFont.Italic = True
newFont.Bold = True
newFont.Size = 12

Set VcDateLine.Font = newFont
```

FontColor

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie die Schriftfarbe einer Stichtaglinie erfragen oder festlegen. Mit dieser Eigenschaft können Sie die Schriftattribute einer Stichtaglinie erfragen oder festlegen. Diese Eigenschaft können Sie auch im Dialog **Stichtaglinien bearbeiten** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
```



```
dateLine.FontColor = RGB(120, 100, 150)
```

Identifiable

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Stichtaglinie identifizierbar ist. Wenn die Eigenschaft auf **True** gesetzt wurde, kann die Stichtaglinie mittels der VcGantt-Methode **IdentifyObjectAt** identifiziert werden.

Diese Option kann auch im Dialog **Stichtaglinien festlegen** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Stichtaglinie identifizierbar / nicht identifizierbar Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

LabelPosition

Nur-Lese-Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie festlegen oder erfragen, an welcher Position ggf. ein Text ausgegeben werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	LabelPositionEnum	Position der Stichtaglinienbeschriftung

LineColor

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie die Farbe einer Stichtaglinie erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Stichtaglinie bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte Standardwert: 255. Visual Basic: RGB (255, 0, 0)

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
dateLine.LineColor = RGB(120, 100, 150)
```

LineThickness**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie die Linienstärke einer Stichtaglinie erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Diese Eigenschaft können Sie auch in dem Dialog **Stichtaglinie bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
dateLine.LineType = vcSolid
dateLine.LineThickness = 3
```

LineType

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie den Linientyp einer Stichtaglinie erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Stichtaglinie bearbeiten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp Standardwert: vcSolid
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100 vcLineType1 101 vcLineType10 110 vcLineType11 111 vcLineType12 112 vcLineType13 113 vcLineType14 114 vcLineType15 115 vcLineType16 116 vcLineType17 117 vcLineType18 118 vcLineType2 102 vcLineType3 103 vcLineType4 104 vcLineType5 105 vcLineType6 106 vcLineType7 107	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0  Linientyp 1  Linientyp 10  Linientyp 11  Linientyp 12  Linientyp 13  Linientyp 14  Linientyp 15  Linientyp 16  Linientyp 17  Linientyp 18  Linientyp 2  Linientyp 3  Linientyp 4  Linientyp 5  Linientyp 6  Linientyp 7 

vcLineType8 108	Linientyp 8 -----
vcLineType9 109	Linientyp 9 -----
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
dateLine.LineType = vcSolid
```

Moveable**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Stichtaglinie interaktiv verschiebbar sein soll. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	verschiebbar (True)/ nicht verschiebbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
If chkMoveable.Value = vbUnchecked Then
    dateLine.Moveable = False
Else
    dateLine.Moveable = True
End If
```

Name**Nur-Lese-Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie den Namen der Stichtaglinie erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection

For Each dateLine in dateLineCltn
    ListBox.AddItem (dateLine.Name)
Next dateLine

```

Priority**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie die Priorität einer Stichtaglinie setzen oder erfragen. Wenn zwei Objekte dieselbe Position im Diagramm haben, liegt das Objekt mit der höheren Priorität über den Objekten mit einer niedrigeren Priorität. Gitter haben standardmäßig die niedrigste Priorität; Knoten besitzen mit dem Wert 0 die höchste Priorität von allen Objekten. Stichwertlinien liegen standardmäßig vor Kalender- und Terminliniengittern, aber hinter den Knoten. Soll eine Stichwertlinie vor den Knoten liegen, müssen Sie ihre Priorität auf einen positiven Wert setzen. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Prioritätsstufe Standardwert: 0
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```

Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("dateLine1")
dateLine.Priority = 10

```

SnapTarget**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob für diese Stichtaglinie ein Einrastziel am Datum definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Einrastziel wird/wird nicht am Datum der Stichtaglinie definiert. Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Specification

Nur-Lese-Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie die Spezifikation dieser Stichtaglinie auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung einer Stichtaglinie mit der Methode **VcDateLineCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Spezifikation der Stichtaglinie Name der Farbzuoordnungstabelle

Text

Eigenschaft von VcDateLine

Mit dieser Eigenschaft können Sie einen Erläuterungstext zu der Stichtaglinie setzen oder erfragen. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	String Mögliche Werte:	Beschriftungstext Name der Farbzuoordnungstabelle
Eigenschaftswert	String Mögliche Werte:	Text für die Stichtaglinie Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim dateLine As VcDateLine
```

```
Set dateLine = VcGantt1.DateLineCollection.DateLineByName("DateLine1")
dateLine.Text = "Stichtag"
```

TurningAnnotationEnabled**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie einstellen oder erfragen ob die Beschriftung der Stichtaglinie um 90 Grad gedreht wird. Diese Eigenschaft kann auch im Dialog **Stichtaglinie bearbeiten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Beschriftung der Stichtaglinie ist/ist nicht um 90 Grad gedreht
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UpdateBehaviorName**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Visible**Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie die Sichtbarkeit einer Stichtaglinie setzen oder erfragen. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Stichtaglinie sichtbar/unsichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim dateLine As VcDateLine

Set dateLine = VcGantt1.DateLineCollection.DateLineByName("dateLine1")
If chkVisible.Value = vbUnchecked Then
    dateLine.Visible = False
Else
    dateLine.Visible = True
End If

```

VisibleDataFieldIndex**Nur-Lese-Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können sie den Index des Datenfeldes für die individuelle Stichtaglinie setzen oder erfragen, das den Wert "1" (für "sichtbar") oder 0 (für "unsichtbar") enthält. Sie können diese Eigenschaft auch im Dialog **Stichtaglinien festlegen** setzen.

	Datentyp	Beschreibung

VisibleMapName**Nur-Lese-Eigenschaft von VcDateLine**

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle (Typ vcTextMap) für den Sichtbarkeitsmodus setzen oder erfragen. Wird hier "" angegeben, wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **VisibleDataFieldIndex** angegeben ist, wird der Sichtbarkeitsstatus aus der Zuordnungstabelle ausgewählt. Diese Eigenschaft kann auch im Dialog **Stichtaglinien festlegen** festgelegt werden. Trifft kein Datenfeldeintrag in der Zuordnungstabelle zu, wird der Wert aus dem Dialog verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Zuordnungstabelle für den Sichtbarkeitsmodus
	Mögliche Werte:	Name der Farbzusordnungstabelle

Methoden

PutInOrderAfter

Methode von VcDateLine

Mit dieser Methode können Sie die Terminline in der Auflistung aller Terminlinien hinter den durch den Namen angegebenen setzen. Wenn als Name "" übergeben wird, wird die Terminlinie an die erste Stelle gesetzt. Die Reihenfolge der Terminlinien in der Auflistung entscheidet darüber, in welcher Reihenfolge sie dargestellt werden.

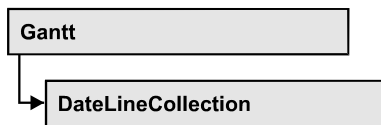
	Datentyp	Beschreibung
Parameter: ↳ refName	String	Name der Stichtaglinie, hinter den die aktuelle Stichtaglinie eingeordnet werden soll.
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Void	

Code-Beispiel

```
Dim datLinCltn As VcDateLineCollection
Dim datLin1 As VcDateLine
Dim datLin2 As VcDateLine

datLinCltn = VcGantt1.DateLineCollection()
datLin1 = datLinCltn.Add("datLin1")
datLin2 = datLinCltn.Add("datLin2")
datLin1.PutInOrderAfter("datLin2")
datLinCltn.Update()
```

7.29 VcDateLineCollection



In einem Objekt vom Typ **VcDateLineCollection** sind alle verfügbaren Terminlinien automatisch enthalten. Über **For Each dateLine In DateLineCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Terminlinien zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **DateLineByName** und **DateLineFormatByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Terminlinien kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Terminlinien.

Eigenschaften

- **_NewEnum**
- **Count**

Methoden

- **Add**
- **AddBySpecification**
- **Copy**
- **DateLineByIndex**
- **DateLineByName**
- **FirstDateLine**
- **NextDateLine**
- **Remove**
- **Update**

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcDateLineCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Stichtaglinienobjekte iterieren. In Visual Basic wird diese Eigen-

schaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dateline As VcDateLine

For Each dateline In VcGantt1.DateLineCollection
    Debug.Print dateline.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcDateLineCollection

Mit dieser Eigenschaft können Sie die Anzahl von Stichtaglinien in der DateLine-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Stichtaglinien

Code-Beispiel

```
Dim numberOfDateLines As Long

numberOfDateLines = VcGantt1.DateLineCollection.Count
```

Methoden

Add

Methode von VcDateLineCollection

Mit dieser Methode können Sie eine neue Stichtaglinie in der DateLineCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Stichtaglinienobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die neu angelegte Stichtaglinie im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ dateLineName	String Mögliche Werte:	Name der Stichtaglinie Name der Farbzusordnungstabelle
Rückgabewert	VcDateLine	Neues Stichtaglinienobjekt

Code-Beispiel

```
Set newDateLine = VcGantt1.DateLineCollection.Add("DateLine1")
```

AddBySpecification**Methode von VcDateLineCollection**

Mit dieser Methode können Sie eine Stichtaglinie über eine Stichtaglinien-Spezifikation erzeugen. Dies dient der Persistenz von Stichtaglinien-Objekten. Die Spezifikation einer Stichtaglinie kann erfragt (siehe VcDateLine-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Stichtaglinie mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden. Um die neu angelegte Stichtaglinie im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ specification	String Mögliche Werte:	Stichtaglinien-Spezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcDateLine	Neues Stichtaglinienobjekt

Copy**Methode von VcDateLineCollection**

Mit dieser Methode können Sie eine Stichtaglinie kopieren. Wenn die Stichtaglinie mit dem angegebenen Namen existiert und der Name der neuen Stichtaglinie noch nicht verwendet wird, wird das neue Stichtaglinienobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die kopierte Stichtaglinie im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ dateLineName	String	Name der zu kopierenden Stichtaglinie
	Mögliche Werte:	Name der Farbzuordnungstabelle
⇒ newDateLineName	String	Name der neuen Stichtaglinie
	Mögliche Werte:	Name der Farbzuordnungstabelle
Rückgabewert	VcDateLine	Stichtaglinienobjekt

Code-Beispiel

```

Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.Copy("DateLineOne", "NewDateLine")
dateLineCltn.Update

```

DateLineByIndex**Methode von VcDateLineCollection**

Mit dieser Methode können Sie auf eine einzelne Terminlinie über ihren Index zugreifen. Existiert keine Terminlinie unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index der Stichtaglinie
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcDateLine	Ermitteltes Stichtaglinienobjekt

Code-Beispiel

```

Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.DateLineByIndex(0)
MsgdateLine DateLine.Name

```

DateLineByName

Methode von VcDateLineCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Stichtaglinie zugreifen. Existiert keine Stichtaglinie unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇨ dateLineName	String	Name der Stichtaglinie
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcDateLine	Stichtaglinie

Code-Beispiel

```
Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.DateLineByName("DateLineOne")
MsgdateLine DateLine.Name
```

FirstDateLine

Methode von VcDateLineCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. die erste Stichtaglinie der Dateline-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDateLine** über die nachfolgenden Stichtaglinien zu iterieren. Existiert keine Stichtaglinie in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDateLine	Erste Stichtaglinie

Code-Beispiel

```
Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.FirstDateLine
While Not dateLine Is Nothing
    ListdateLine.AddItem (dateLine.Name)
    Set dateLine = dateLineCltn.NextDateLine
Wend
```

NextDateLine

Methode von VcDateLineCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Stichtaglinien der VcDateLine-Auflistung zugreifen, nachdem Sie mit der Methode **FirstDateLine** den Initialwert erfasst haben. Sind alle Stichtaglinien durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDateLine	Nachfolgende Stichtaglinie

Code-Beispiel

```
Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection

Set dateLine = dateLineCltn.FirstDateLine
While Not dateLine Is Nothing
    ListdateLine.AddItem (dateLine.Name)
    Set dateLine = dateLineCltn.NextDateLine
Wend
```

Remove

Methode von VcDateLineCollection

Mit dieser Methode können Sie eine Stichtaglinie löschen. Um das Löschen der Stichtaglinie im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ dateLineName	String	Name der Stichtaglinie
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Stichtaglinie gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.DateLineByIndex(2)
dateLineCltn.Remove (dateLine.Name)
dateLineCltn.Update
```

Update

Methode von VcDateLineCollection

Mit dieser Methode können Sie eine DateLineCollection aktualisieren, nachdem Sie sie verändert haben.

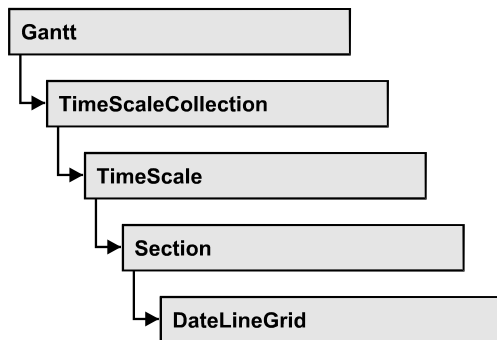
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
Set dateLine = dateLineCltn.DateLineByIndex(2)
dateLineCltn.Remove (DateLine.Name)
dateLineCltn.Update
```


7.30 VcDateLineGrid



Ein Objekt vom Typ **VcDateLineGrid** ist ein vordefiniertes Liniengitter, in dem Zeitabschnitte (Tage, Wochen, Monate, etc.) durch vertikale Linien markiert werden.

Eigenschaften

- AdjustToReferenceDate
- AnnotationAtBottom
- AnnotationAtCenter
- AnnotationAtTop
- FormatName
- HorAlignment
- LineColor
- LineColorDataFieldIndex
- LineColorMapName
- LineThickness
- LineType
- Name
- ObserveDST
- Period
- Priority
- ReferenceDate
- SnapTarget
- TurningAnnotationEnabled
- Unit
- UseReferenceDate
- Visible
- VisibleDataFieldIndex
- VisibleMapName

Eigenschaften

AdjustToReferenceDate

Eigenschaft von VcDateLineGrid

Die Linien eines Liniengitters werden standardmäßig am Anfang der jeweiligen Zeiteinheit positioniert, z.B. immer um 00:00 Uhr jedes Tages. Mit dieser Eigenschaft können die Linien des Liniengitters am Referenzdatum ausgerichtet werden, d.h. die Position kann vom Anfang einer Zeiteinheit abweichen, z.B. um 13:17 jedes Tages. Das Referenzdatum wird gesetzt über die Eigenschaft **set/getReferenceDate**.

Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitter auf Referenzdatum positioniert (True)/nicht positioniert (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AnnotationAtBottom

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Beschriftung der Linien im Liniengitter nach unten setzen oder erfragen, ob die Beschriftung am unteren Rand des Gantt Graphen ausgerichtet ist. S. auch **set/getAnnotationAtCenter** und **set/getAnnotationAtTop**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitterbeschriftung unten (True)/nicht unten (False) positioniert Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AnnotationAtCenter

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Beschriftung der Linien im Liniengitter in die Mitte des Gantt Graphen setzen oder erfragen, ob die Beschriftung mittig ausgerichtet ist. S. auch **set/getAnnotationAtBottom** und **set/getAnnotationAtTop**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitterbeschriftung mittig (True) / nicht mittig (False) positioniert Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AnnotationAtTop

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Beschriftung der Linien im Liniengitter nach oben setzen oder erfragen, ob die Beschriftung am oberen Rand des Gantt Graphen ausgerichtet ist. S. auch **set/getAnnotationAtCenter** und **set/getAnnotationAtBottom**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitterbeschriftung oben (True)/nicht oben (False) positioniert Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

FormatName

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Namen des Linienformats dieses Terminliniengitters erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Linienformats
	Mögliche Werte:	Name der Farbzuoordnungstabelle

HorAlignment

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft wird für die Beschriftungen der Linien die horizontale Ausrichtung festgelegt oder erfragt.

	Datentyp	Beschreibung
Eigenschaftswert	HorizontalAlignmentEnum	Art der Ausrichtung
	Mögliche Werte: vcHorCenterAligned -1 vcLeftAligned -3 vcRightAligned -2	horizontal mittig linksbündig rechtsbündig

LineColor

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Farbe eines Terminliniengitters erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte
		Standardwert: 255. Visual Basic: RGB (255, 0, 0)

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid =
VcGantt1.TimeScaleCollection.Active.Section(0).dateLineGrid(0)

dateLineGrid.LineColor = RGB(130, 80, 200)
```

LineColorDataFieldIndex

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuoordnungstabelle in der Eigenschaft **LineColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

LineColorMapName

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle für die Linienfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **LineColorDataFieldIndex** **-1** angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

LineThickness

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Linienstärke eines Terminliniengitters erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm

Wert	Punkte	mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Diese Eigenschaft können Sie ebenfalls in dem Dialog **Kalendergitter bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid = VcGantt1.TimeScaleCollection._
                    Active.Section(0).dateLineGrid(0)

dateLineGrid.LineThickness = 2
```

LineType

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Linientyp eines Terminliniengitters erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp Standardwert: vcDashed
	Mögliche Werte: vcDashed 4 vcDashedDotted 5	Linientyp gestrichelt Linientyp gestrichelt-gepunktet

vcDotted 3	Linientyp gepunktet
vcLineType0 100	Linientyp 0
vcLineType1 101	Linientyp 1
vcLineType10 110	Linientyp 10
vcLineType11 111	Linientyp 11
vcLineType12 112	Linientyp 12
vcLineType13 113	Linientyp 13
vcLineType14 114	Linientyp 14
vcLineType15 115	Linientyp 15
vcLineType16 116	Linientyp 16
vcLineType17 117	Linientyp 17
vcLineType18 118	Linientyp 18
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3
vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid =
VcGantt1.TimeScaleCollection.Active.Section(0).dateLineGrid(0)

dateLineGrid.LineType = vcSolid
```

Name

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Namen eines Terminliniengitters erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Terminliniengitters
	Mögliche Werte:	Name der Farbzuoordnungstabelle

ObserveDST

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob für dieses Liniengitter die Sommerzeit in der Darstellung berücksichtigt wird oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	DateLineGridObserveDSTEnum	Sommerzeit wird/wird nicht berücksichtigt.
	Mögliche Werte: vcGODDefault 9999 vcGODNo 0 vcGODYes 1	Standardeinstellung aus der .INI-Datei wird verwendet Sommerzeit wird nicht berücksichtigt Sommerzeit wird berücksichtigt

Period

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Zeiteinheiten (Unit) jeweils eine Linie des Terminliniengitters gezogen werden soll. Der Abstand zwischen den Rasterlinien ist gleich dem Produkt aus der Zeiteinheit des Gitters (Unit) und der Periode (Period).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Periodenwert
		Standardwert: 1

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid =
VcGantt1.TimeScaleCollection.Active.Section(0).dateLineGrid(0)

dateLineGrid.Unit = vcGridUnitDay
dateLineGrid.Period = 1
```


Priority

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Priorität eines Terminliniengitters erfragen oder festlegen. Wenn zwei Objekte dieselbe Position im Diagramm haben, liegt das Objekt mit der höheren Priorität über den Objekten mit einer niedrigeren Priorität. Gitter haben standardmäßig die niedrigste Priorität; Knoten besitzen mit dem Wert 0 die höchste Priorität von allen Objekten. Liniengitter liegen standardmäßig vor Kalenderliniengittern, aber hinter Stichtaglinien und Knoten. Soll ein Liniengitter vor den Knoten liegen, müssen Sie seine Priorität auf einen positiven Wert setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Prioritätsstufe {-1000...+1000} Standardwert: -20

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid = VcGantt1.TimeScaleCollection.Active.Section(0). _
    dateLineGrid(0)

dateLineGrid.Priority = 10
```

ReferenceDate

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie das Referenzdatum des Liniengitters erfragen oder festlegen. Für die tatsächliche Verwendung dieses Datums muss die Eigenschaft **UseReferenceDate** gesetzt werden. Um das Liniengitter an diesem Datum auszurichten, s. Eigenschaft **AdjustToReferenceDate**.

Das Referenzdatum versetzt den Anfang des Gitters um den angegebenen Abstand zu dem standardmäßigen Beginn am Montag 00.00 Uhr. Dabei spielt das absolute Datum keine Rolle, sondern nur die Differenz zum Standardbeginn. Beispiel: Soll das Gitter an einem Dienstag beginnen, setzen Sie das Referenzdatum z.B. auf den 6.5.2014. Das gleiche Ergebnis können Sie erzielen, wenn Sie das Referenzdatum eine Woche früher setzen, also auf den 29.4.2014. Ausschlaggebend ist die Differenz zu Montag, sie entspricht in dem Beispiel jeweils 1 Tag. Der Abstand muss nicht in Tagen angegeben werden, auch Uhrzeiten sind möglich, z.B. 29.4.2014 8:15 Uhr. Wenn ein

Stundengitter vorliegt, sind nur die Minuten relevant und bei letzterem Beispiel wird das Gitter um 15 Minuten versetzt.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Referenzdatum

SnapTarget

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob für dieses Liniengitter ein Einrastziel am Datum definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Einrastziel wird/wird nicht am Datum des Liniengitters definiert. Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

TurningAnnotationEnabled

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die Beschriftungen an den Linien des Liniengitters um 90 Grad (vertikal) gedreht werden können.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Die Beschriftungen können gedreht werden (True) / wurden bereits gedreht (False) Standardwert: True Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Unit

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie die Zeiteinheit des Liniengitters erfragen oder festlegen. Der Abstand zwischen den einzelnen Rasterlinien ist gleich dem Produkt aus der Zeiteinheit des Liniengitters (Unit) und der Periode (Period).

	Datentyp	Beschreibung
Eigenschaftswert	GridUnitEnum	Zeiteinheit Standardwert: vcGridUnitWeek
	Mögliche Werte: vcGridUnitDay 5 vcGridUnitHour 6 vcGridUnitMinute 7 vcGridUnitMonth 3 vcGridUnitQuarter 2 vcGridUnitSecond 8 vcGridUnitWeek 4 vcGridUnitYear 1	Gittereinheit Tag Gittereinheit Stunde Gittereinheit Minute Gittereinheit Monat Gittereinheit Quartal Gittereinheit Sekunde Gittereinheit Woche Gittereinheit Jahr

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid =
VcGantt1.TimeScaleCollection.Active.Section(0).DateLineGrid(0)

dateLineGrid.Period = 1
dateLineGrid.Unit = vcGridUnitDay
```

UseReferenceDate

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Liniengitter ein Referenzdatum verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitter verwendet (True)/verwendet kein (False) Referenzdatum Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Visible

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Liniengitter sichtbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Liniengitter sichtbar (True)/unsichtbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

Set dateLineGrid = VcGantt1.TimeScaleCollection.Active.Section(0). _
    DateLineGrid(0)

dateLineGrid.Visible = True
```

VisibleDataFieldIndex

Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das den Wert "1" (für "sichtbar") oder 0 (für "unsichtbar") enthält. Diese Eigenschaft kann auch im Dialog **Terminliniengitter** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Sichtbarkeitsmodus enthält

VisibleMapName

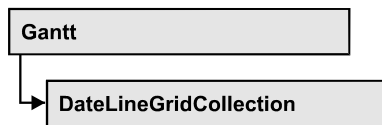
Eigenschaft von VcDateLineGrid

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle (Typ vcTextMap) für den Sichtbarkeitsmodus setzen oder erfragen. Wird hier "" angegeben, wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **VisibleDataFieldIndex** angegeben ist, wird der Sichtbarkeitsstatus aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das

Terminliniengitter auf "sichtbar" gesetzt. Diese Eigenschaft kann auch im dialog **DateLineGrid** festgelegt werden.

	Datentyp	Beschreibung
Parameter: ⇔ Rückgabewert	String Mögliche Werte:	Name der Zuordnungstabelle für den Sichtbarkeitsmodus Name der Farbzusordnungstabelle
Eigenschaftswert	String Mögliche Werte:	Name der Zuordnungstabelle für den Sichtbarkeitsmodus Name der Farbzusordnungstabelle

7.31 VcDateLineGridCollection



In einem Objekt des Typs `VcDateLineGridCollection` sind alle verfügbaren Terminliniengitter zusammengefasst. Über **For Each dateLineGrid In DateLineGridCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Terminliniengitter zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **DateLineGridByName** und **DateLineGridByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Terminliniengitter kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Terminliniengittern.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `DateLineGridByIndex`
- `DateLineGridByName`
- `FirstDateLineGrid`
- `NextDateLineGrid`
- `Remove`
- `Update`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcDateLineGridCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese

Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid

For Each dateLineGrid In VcGantt1.DateLineGrid
    Debug.Print dateLineGrid.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcDateLineGridCollection

Mit dieser Eigenschaft kann die Anzahl der Terminliniengitter in der Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der DateLineGrids

Code-Beispiel

```
Dim dateLineGridCltn As Vc DateLineGridCollection
Dim numberOfDateLineGrids As Long

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
numberOfDateLineGrids = dateLineGridCltn.Count
```

Methoden

Add

Methode von VcDateLineGridCollection

Mit dieser Methode können Sie ein neues Terminliniengitter in der DateLineGridCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Terminliniengitterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ dateLineGridName	String Mögliche Werte:	Name des DateLineGrids Name der Farbzusordnungstabelle
Rückgabewert	VcDateLineGrid	Neues DateLineGrid-Objekt

Code-Beispiel

```
Set newDateLineGrid = VcGantt1.DateLineGridCollection.Add("Grid1")
```

AddBySpecification**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie ein Terminliniengitter über eine Terminliniengitter-Spezifikation erzeugen. Dies dient der Persistenz von Terminliniengitter-Objekten. Die Spezifikation eines Terminliniengitters kann erfragt und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Terminliniengitter mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ specification	String Mögliche Werte:	DateLineGrid-Spezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcDateLineGrid	Neues DateLineGrid-Objekt

Copy**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie ein Terminliniengitter kopieren. Wenn das Terminliniengitter mit dem angegebenen Namen existiert und der Name des neuen Terminliniengitters noch nicht verwendet wird, wird das neue Terminliniengitterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ dateLineGridName	String	Name des zu kopierenden DateLineGrids

⇒ newDateLineGridName	Mögliche Werte:	Name der Farbzuoordnungstabelle
	String	Name des neuen DateLineGrids
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcDateLineGrid	DateLineGridobjekt

Code-Beispiel

```
Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
Set dateLineGrid = dateLineGridCltn.Copy("CurrentDateLineGrid",
"NewDateLineGrid")
```

DateLineGridByIndex**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie auf eine einzelne Terminliniengitterüber ihren Index zugreifen. Existiert kein Terminliniengitter unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇒ index	Integer	Index des DateLineGrids
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcDateLineGrid	Ermitteltes DateLineGridobjekt

Code-Beispiel

```
Dim dateLineGridCltn As VcDateLineGrid
Dim dateLine As VcDateLine

Set dateLineGridCltn = VcGantt1.DateLineGrid
Set dateLineGrid = dateLineGridCltn.DateLineGridByIndex(2)
MsgBox dateLineGrid.Name
```

DateLineGridByName**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie unter Verwendung des Namens des Terminliniengitters auf ein bestimmtes Terminliniengitter zugreifen. Existiert

kein DateLineGrid-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ DateLineGridName	String	Name des DateLineGrids
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcDateLineGrid	DateLineGrid

Code-Beispiel

```
Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
Set dateLineGrid = dateLineGridCltn.DateLineGridByName("Grid 4")
```

FirstDateLineGrid

Methode von VcDateLineGridCollection

Mit dieser Methode können Sie auf das erste Terminliniengitter der Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDateLineGrid** über die nachfolgenden Terminliniengitter zu iterieren. Existiert kein Terminliniengitter in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDateLineGrid	Erstes DateLineGrid

Code-Beispiel

```
Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
dateLineGridCltn.DateLineGrids (vcAnyDateLineGrid)
Set dateLineGrid = dateLineGridCltn.FirstDateLineGrid
```

NextDateLineGrid

Methode von VcDateLineGridCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Terminliniengitter der Auflistung zugreifen, nachdem Sie mit der Methode **FirstDateLineGrid** den Initialwert erfasst haben. Sind alle Levels durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDateLineGrid	Nachfolgendes DateLineGrid

Code-Beispiel

```

Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
Set dateLineGrid = dateLineGridCltn.FirstDateLineGrid

While Not dateLineGrid Is Nothing
    Listbox.AddItem dateLineGrid.Name
    Set dateLineGrid = dateLineGridCltn.NextDateLineGrid
Wend

```

Remove**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie ein Terminliniengitter löschen. Wenn das Terminliniengitter noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ dateLineGridName	String	Name des DateLineGrids
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	DateLineGrid gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```

Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineGridCltn = VcGantt1.DateLineGridCollection
Set dateLineGrid = dateLineGridCltn.FormatByIndex(1)
dateLineGridCltn.Remove (dateLineGrid.Name)

```

Update**Methode von VcDateLineGridCollection**

Mit dieser Methode können Sie die Darstellung aller Objekte, die durch die verwendeten Terminliniengitter bestimmt werden, aktualisieren. Wenn Sie diese Methode nicht aufrufen, werden die Änderungen der Terminliniengitter zur Laufzeit nicht ausgeführt. Sie sollten diese Methode erst am Ende des Codes zur Festlegung der Terminliniengitter und der Level-Auflistung

verwenden, damit die Aktualisierung nicht schon ausgeführt wird, bevor alle Festlegungen der Terminliniengitter ausgeführt worden sind.

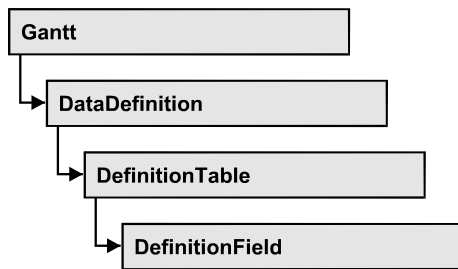
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim dateLineGrid As VcDateLineGrid
```

```
Set dateLineGrid = VcGantt1.DateLineGrid.Collection.DateLineGridByName("Grid 3")  
dateLineGrid.Update
```

7.32 VcDefinitionField



Ein Objekt vom Typ `VcDefinitionField` definiert ein Feld der Datendefinitionstabelle. Die Definition besteht im Kern aus einem Namen und einer Festlegung des Datentyps.

Eigenschaften

- `DateFormat`
- `Editable`
- `Hidden`
- `ID`
- `Name`
- `Type`

Eigenschaften

DateFormat

Eigenschaft von `VcDefinitionField`

Mit dieser Eigenschaft können Sie das Datumsformat des Feldes in einer Datendefinitionstabelle festlegen oder erfragen. Diese Eigenschaft ist nur wirksam, wenn der Datentyp des Feldes auf **`vcDefFieldDateTimeType`** eingestellt ist.

Die `DateFormat`-Einstellung wird beim Lesen und Schreiben von CSV-Dateien verwendet und wenn der Formattyp **`String`** beim Hinzufügen eines Datensatzes über die Methoden **`InsertNodeRecord`** oder **`InsertLinkRecord`** des `Gantt`-Objektes verwendet wird.

Das Format der Datumsausgabe in der Grafik wird über die Eigenschaft **`DateOutputFormat`** bei `VcGantt` gesteuert.

Hinweis: Es sollte zuerst die Eigenschaft Type gesetzt werden, bevor die Eigenschaft DateFormat vereinbart wird.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat {DMYhms;.:/} Standardwert: bei vcDefFieldDateTime DD.MM.YYYY hh:mm:ss
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Type = vcDefFieldDateTimeType
'DateFormat = "DD.MM.YYYY"
dataDefField.DateFormat = "01.12.2014"
```

Editable

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das spezifizierte Datenfeld zur Laufzeit in der Tabelle (des Diagramms) und des Dialogs **Knoten bearbeiten** editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Definitionsfeld editierbar/nicht editierbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Editable = False
```

Hidden

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob ein Datenfeld zur Laufzeit versteckt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Definitionsfeld versteckt/nicht versteckt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Hidden = True
```

ID

Nur-Lese-Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie den Index des Feldes einer Datendefinitionstabelle abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Definitionsfeldes
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
MsgBox dataDefField.ID
```

Name

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie den Namen des Feldes einer Datendefinitionstabelle setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Definitionsfeldes
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.CreateDataField("Start")
```

Type**Eigenschaft von VcDefinitionField**

Mit dieser Eigenschaft können Sie den Typ des Feldes einer Daten-definitionstabelle erfragen oder setzen.

Hinweis: Durch Setzen der Eigenschaft **Type** wird die Eigenschaft **Date-Format** geändert!

vcDefFieldAlphanumericType: DateFormat = ""

vcDefFieldDateTimeType: DateFormat = "DD.MM.YYYY hh:mm:ss"

vcDefFieldIntegerType: DateFormat = ""

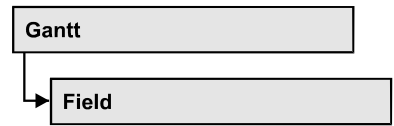
	Datentyp	Beschreibung
Eigenschaftswert	DefinitionFieldTypeEnum	Typ des Definitionsfeldes Standardwert: vcDefFieldIntegerType
	Mögliche Werte: vcDefFieldAlphanumericType 1 vcDefFieldDateTimeType 4 vcDefFieldIntegerType 2	Datentyp Alphanumerisch: "" Datentyp Datum: DD.MM.YYYY Datentyp Integer (32 Bit): ""

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcGantt1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.CreateDataField("Start")
dataDefField.Type = vcDefFieldDateTimeType
```


7.33 VcField



Ein Objekt vom Typ VcField ist ein Feld in einem Datensatz. Die Felder eines Datensatzes sind über Datenfeld-Identifikationsnummern adressierbar.

Eigenschaften

- DataFieldID

Eigenschaften

DataFieldID

Nur-Lese-Eigenschaft von VcField

Mit dieser Eigenschaft können Sie die Identifikation eines Datenfeldes erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeld-Identifikationsnummer
	Mögliche Werte:	Datenfeldindex

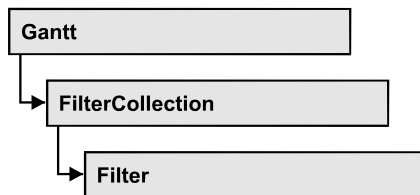
Code-Beispiel

```
Private Sub VcGantt1_OnNodeLClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant)

    Dim identifiedObject As Object
    Dim identifiedObjectType As VcObjectTypeEnum
    Dim dataField As VcField

    If location = vcInTable Then
        Call VcGantt1.IdentifyObjectAt(x, y, identifiedObject, _
            identifiedObjectType)
        Set dataField = VcGantt1.IdentifyField(x, y, identifiedObjectType)
        MsgBox dataField.DataFieldID
    End If
End Sub
```

7.34 VcFilter



Ein Objekt vom Typ VcFilter enthält Filterbedingungen (VcFilterSubCondition), z.B. zulässige Werte für die Datenfelder eines Knotens oder einer Verbindung. Abhängig von den Daten trifft die Filterbedingung für einen Vorgang zu oder nicht. Filter werden verwendet, um z.B. einem Knoten ein bestimmtes Format zuzuweisen.

Nur wenn der Filter nach Änderungen der Filterbedingungen gültig ist, werden diese wirksam. Andernfalls bleiben die bisherigen Filterbedingungen wirksam (prüfbar über die Methoden VcFilter.IsValid und VcFilterSubCondition.IsValid).

Eigenschaften

- _NewEnum
- DataDefinitionTable
- DatesWithHourAndMinute
- Name
- Specification
- StringsCaseSensitive
- SubCondition
- SubConditionCount

Methoden

- AddSubCondition
- CopySubCondition
- Evaluate
- IsValid
- RemoveSubCondition

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcFilter

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Filterbedingungsobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim fiSuCo As VcFilterSubCondition

For Each fiSuCo In filter
    Debug.Print fiSuCo.Index
Next
```

DataDefinitionTable

Eigenschaft von VcFilter

Diese Eigenschaft gibt zurück, ob es sich um einen Filter für Knoten (vcMainData) oder für Verbindungen (vcRelations) handelt. Ändern kann man diese Eigenschaft nur, wenn der Filter keine Bedingungen enthält.

	Datentyp	Beschreibung
Eigenschaftswert	DataTableEnum Mögliche Werte: vcMaindata 0 vcRelations 1	Typ der Datendefinitionstabelle Tabellentyp vcMaindata (für Knoten) Tabellentyp vcRelations (für Verbindungen)

DatesWithHourAndMinute

Eigenschaft von VcFilter

Diese Eigenschaft entscheidet, ob beim Vergleich von Datums-Filterbedingungen Stunden und Minuten berücksichtigt werden. Die Einstellung kann nur geändert werden, wenn mindestens eine Unterbedingung mit einem Datumsvergleich vorhanden ist. Ansonsten ist der Eigenschaftswert immer False.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Stunden und Minuten werden berücksichtigt (True)/ nicht berücksichtigt (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Name

Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie den Namen des Filters erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Filters
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcGantt1.FilterCollection

For Each filter In filterCltn
    ListBox.AddItem filter.name
Next filter
```

Specification

Nur-Lese-Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie die Spezifikation dieses Filters auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken

gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung eines Filters mit der Methode **VcFilterCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Filterspezifikation
	Mögliche Werte:	Name der Farbzuoordnungstabelle

StringsCaseSensitive

Eigenschaft von VcFilter

Diese Eigenschaft entscheidet, ob bei String-Filterbedingungen der Vergleich mit Unterscheidung von Groß- und Kleinschreibung stattfindet.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vergleich mit Unterscheidung von Groß- und Kleinschreibung findet statt (True)/findet nicht statt (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

SubCondition

Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie auf ein VcFilterSubCondition-Objekt per Index zugreifen.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index der Filterbedingung {0 ... VcFilter.SubConditionCount-1}
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	VcFilterSubCondition	Filterbedingungsobjekt

SubConditionCount

Nur-Lese-Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie die Anzahl der Filterbedingungen erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Filterbedingungen
	Mögliche Werte:	Datenfeldindex

Methoden

AddSubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine neue Filterbedingung an der angegebenen Stelle in der Collection der bestehenden Filterbedingungen erzeugen.

Das entsprechende VcFilterSubCondition-Objekt wird zurückgegeben. Die Eigenschaften dieses Objekt sind standardmäßig folgendermaßen gesetzt:

- DataFieldIndex: -1
- Operator: vcInvalidOp
- ComparisonValueAsString: "<INVALID>"
- ConnectionOperator: vcInvalidConnOp.

	Datentyp	Beschreibung
Parameter: ⇒ atIndex	Integer	Index der neuen Filterbedingung {0 to VcFilter.SubConditionCount and –1 for "at the end of the Collection" (identical with the value VcFilter.SubConditionCount)}
	Mögliche Werte:	Datenfeldindex

Rückgabewert	VcFilterSubCondition	Filterbedingungsobjekt
---------------------	----------------------	------------------------

CopySubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine Filterbedingung mit Hilfe der Indexangabe kopieren. Die neue Filterbedingung wird an der angegebenen Stelle in der Collection der bestehenden Filterbedingungen eingefügt und als VcFilterSubCondition-Objekt zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ fromIndex	Integer	Index der zu kopierenden Filterbedingung
	Mögliche Werte:	Datenfeldindex
⇒ atIndex	Integer	Index der neuen Filterbedingung
	Mögliche Werte:	{0 to VcFilter.SubConditionCount and –1 for "at the end of the Collection" (identical with the value VcFilter.SubConditionCount)}
		Datenfeldindex
Rückgabewert	VcFilterSubCondition	Filterbedingungsobjekt

Evaluate

Methode von VcFilter

Mit dieser Methode kann für einen bestimmten Datensatz geprüft werden, ob der gesetzte Filter zutrifft oder nicht. Es sollten sinnvollerweise nur Objekte übergeben werden, die intern mit Datensätzen der Datentabellen verbunden sind. Dies sind: **VcNode**, **VcLink**, **VcGroup**, **VcDataRecord**. Wird ein hier nicht aufgeführter Objekttyp übergeben, wird eine Exception ausgelöst.

	Datentyp	Beschreibung
Parameter:		
⇒ dataObjectParam	Variant	Datensatzobjekt
Rückgabewert	Boolean	Filter trifft für Datensatz zu (True)/nicht zu (False)

IsValid

Methode von VcFilter

Diese Methode prüft, ob alle Filterbedingungen korrekt formuliert sind. Nur wenn das der Fall ist, werden geänderte Filterbedingungen überhaupt wirksam. Andernfalls bleiben die bisherigen Filterbedingungen wirksam.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Filterbedingungen korrekt (True)/ nicht korrekt (False)

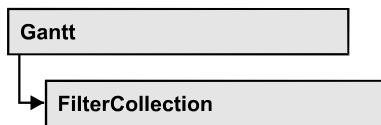
RemoveSubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine Filterbedingung mit Hilfe der Indexangabe löschen.

	Datentyp	Beschreibung
Parameter:		
⇒ index	Integer	Index der zu löschenden Filterbedingung
	Mögliche Werte:	Datenfeldindex

7.35 VcFilterCollection



In einem Objekt vom Typ `VcFilterCollection` sind automatisch alle verfügbaren Filter zusammengefasst. Über **For Each filter In FilterCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Filter zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **FilterByName** und **FilterByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Filter kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Filtern.

Eigenschaften

- `_NewEnum`
- `Count`
- `MarkedNodesFilter`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FilterByIndex`
- `FilterByName`
- `FirstFilter`
- `NextFilter`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcFilterCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Filterobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim filter As VcFilter

For Each filter In VcGantt1.FilterCollection
    Debug.Print filter.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcFilterCollection

Mit dieser Eigenschaft können Sie die Anzahl der Filterobjekte in der Filter-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim numberOfFilters As Long

Set filterCltn = VcGantt1.FilterCollection
numberOfFilters = filterCltn.Count
```

MarkedNodesFilter

Nur-Lese-Eigenschaft von VcFilterCollection

Mit dieser Eigenschaft können Sie einen konstanten Pseudo-Filter holen, der nur bei **ActiveNodeFilter** eingesetzt werden kann und dort das Filtern auf die gerade markierten Knoten auslöst (Teildiagramm).

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Pseudo-Filter

Code-Beispiel

```
Set VcGantt1.ActiveNodeFilter = VcGantt1.FilterCollection.MarkedNodesFilter
```

Methoden

Add

Methode von VcFilterCollection

Mit dieser Methode können Sie einen neuen Filter in der FilterCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Filterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Die referenzierte Datendefinitionstabelle des neuen Filters ist automatisch vcMainData (siehe VcFilter.DataDefinitionTable). Sie kann auf vcRelations geändert werden, solange der Filter noch keine Unterbedingungen enthält.

	Datentyp	Beschreibung
Parameter: ⇨ newName	String Mögliche Werte:	Name des Filters Name der Farbzusordnungstabelle
Rückgabewert	VcFilter	Neues Filterobjekt

Code-Beispiel

```
Set newFilter = VcGantt1.FilterCollection.Add("foo")
```

AddBySpecification

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter über eine Filter-Spezifikation erzeugen. Dies dient der Persistenz von Filterobjekten. Die Spezifikation eines Filters kann erfragt (siehe VcFilter-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann der gleiche Filter mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇨ filterSpecification	String Mögliche Werte:	Filterspezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcFilter	Neues Filterobjekt

Copy

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter kopieren. Wenn der Filter mit dem angegebenen Namen existiert und der Name des neuen Filters noch nicht verwendet wird, wird das neue Filterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ fromName ⇒ newName	String Mögliche Werte: String Mögliche Werte:	Name des zu kopierenden Filters Name der Farbzuoordnungstabelle Name des neuen Filters Name der Farbzuoordnungstabelle
Rückgabewert	VcFilter	Filterobjekt

FilterByIndex

Methode von VcFilterCollection

Mit dieser Methode können Sie auf einen einzelnen Filter über seinen Index zugreifen. Existiert kein Filter unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Filters Datenfeldindex
Rückgabewert	VcFilter	Ermitteltes Filterobjekt

FilterByName

Methode von VcFilterCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf einen bestimmten Filter zugreifen. Existiert kein Filter unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ filterName	String Mögliche Werte:	Name des Filters Name der Farbzusordnungstabelle
Rückgabewert	VcFilter	Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcGantt1.FilterCollection
Set filter = filterCltn.FilterByName("Department A")
```

FirstFilter

Methode von VcFilterCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Filter der Filter-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFilter** über die nachfolgenden Filter zu iterieren. Existiert kein Filter in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcFilter	Erster Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcGantt1.FilterCollection
Set filter = filterCltn.FirstFilter
```

NextFilter

Methode von VcFilterCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Filter der Filter-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFilter** den Initialwert erfasst haben. Sind alle Filter durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcFilter	Nachfolgender Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcGantt1.FilterCollection
Set filter = filterCltn.FirstFilter

While Not filter Is Nothing
    Listbox.AddItem filter.Name
    Set filter = filterCltn.NextFilter
Wend
```

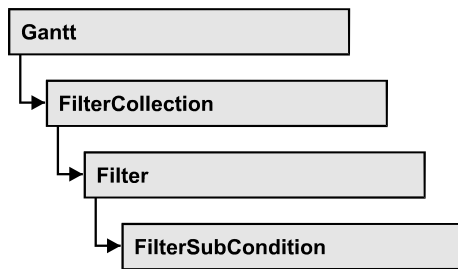
Remove

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter löschen. Wenn der Filter noch in irgendeinem anderen Objekt benutzt wird, kann er nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ name	String	Name des Filters
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Filter gelöscht (True)/nicht gelöscht (False)

7.36 VcFilterSubCondition



Ein Objekt vom Typ VcFilterSubCondition enthält eine einzelne Filterbedingung. Eine Filterbedingung hat im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, der ihre Position im Filter bestimmt.

Im Dialog **Filter bearbeiten** gibt es für jede Filterbedingung eine eigene Zeile. Die dort zur Designzeit dargestellten Eigenschaften sind mit der API hier zur Laufzeit nachträglich veränderbar.

Eigenschaften

- ComparisonValueAsString
- ConnectionOperator
- DataFieldIndex
- FilterName
- Index
- Operator

Methoden

- IsValid

Eigenschaften

ComparisonValueAsString

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Vergleichswert erfragen oder setzen. Dieser String muss einem bestimmten Format entsprechen:

- String: wird in doppelte Anführungszeichen eingeschlossen. Beispiel in VB: ""Aachen""; Beispiel in C/C++: "\"Aachen\""

- Datum: wird in #-Zeichen eingeschlossen. Beispiel: "#18/06/2015;12:34;56;#" (das Datumsformat ist immer "DD.MM.YYYY;hh:mm:ss;", da es sich hierbei um das interne Standardformat, unabhängig vom Betriebssystem und dessen lokalen Einstellungen, handelt). Ein spezieller Datums-Vergleichswert ist "<TODAY>".
- Datenfeld: wird in eckige Klammern eingeschlossen. Beispiel: "[ID]"
- Zahl: wird direkt angegeben. Beispiel: "52076"
- Liste: bei einem der vc...In-Operatoren: wird in geschweifte Klammern eingeschlossen. Die enthaltenen Werte müssen dann alle vom gleichen Typ (String, Datum oder Zahl) sein und können alle obigen Formate besitzen. Beispiel: "{"NETRONIC", [Name]}"
- Ungültig (z. B. nach Neuerzeugen einer Unterbedingung): "<INVALID>"

Der Typ des Vergleichswerts muss dem Datenfeldtyp und dem Typ des Operators entsprechen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Vergleichswert
	Mögliche Werte:	Name der Farbzustimmungstabelle

ConnectionOperator

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Operator für die Verknüpfung mit der folgenden Unterbedingung erfragen oder setzen. Dabei bindet **vcAnd** stärker als **vcOr**.

	Datentyp	Beschreibung
Eigenschaftswert	ConnectionOperatorEnum	Operator für die Verknüpfung mit der folgenden Filterbedingung
	Mögliche Werte: vcAnd 1 vcInvalidConnOp 0 vcOr 2	Und-Operator ungültiger Operator Oder-Operator

DataFieldIndex

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Index des Datenfeldes, dessen Inhalt verglichen werden soll, erfragen oder setzen. Der Datenfeldtyp muss dem Typ des Vergleichswerts und des Operators entsprechen.

Sonderwerte:

- -1: kein Datenfeld (ungültig)
- vcBarGroupLevel: Platzhalter für Gruppenebenennummer
- vcGroupCollapsed: für kollabierte Gruppen
- vcGroupNodeOrSummaryNode: Eintrag, um Filter für Sammelvorgänge definieren zu können
- vcNodesInSeparateRows: für eine Darstellung, bei der jeder Knoten in einer eigenen Zeile dargestellt wird
- vcNodesOverlaid: für eine Darstellung, bei der Knoten einander ggf. überlappen
- vcRowNumber: Eintrag, um Filter zeilenweise definieren zu können
- vcSumBarLevel: Platzhalter für Summenbalken-Ebenennummer

Diese Eigenschaft können Sie auch im **Filter bearbeiten**-Dialog setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, dessen Inhalt verglichen werden soll

FilterName

Nur-Lese-Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Namen des Filters erfragen, zu dem diese Filterbedingung gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Filters
	Mögliche Werte:	Name der Farbzustimmungstabelle

Index

Nur-Lese-Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Index dieser Filterbedingung im zugehörigen Filter erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index der Filterbedingung im zugehörigen Filter
	Mögliche Werte:	Datenfeldindex

Operator

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Operator für den Vergleich erfragen oder setzen. Die über die API verfügbaren Operatoren entsprechen den Operatoren im Dialog **Filter bearbeiten**. Der Typ des Operators muss dem Datenfeldtyp und des Vergleichswerts entsprechen.

	Datentyp	Beschreibung
Eigenschaftswert	OperatorEnum	Vergleichsoperator
	Mögliche Werte: vcDateEarlier 27 vcDateEarlierOrEqual 28 vcDateEqual 25 vcDateIn 31 vcDateLater 29 vcDateLaterOrEqual 30 vcDateNotEqual 26 vcDateNotIn 32 vcIntEqual 9 vcIntGreater 13 vcIntGreaterOrEqual 14 vcIntIn 15 vcIntLess 11 vcIntLessOrEqual 12 vcIntNotEqual 10 vcIntNotIn 16 vcInvalidOp 0 vcStringBeginsWith 3	Datum früher als Datum früher als oder gleich Datum gleich Datum in Datum später als Datum später als oder gleich Datum ungleich Datum nicht in Integer gleich Integer größer Integer größer oder gleich Integer in Integer kleiner als Integer kleiner als oder größer Integer ungleich Integer nicht in ungültiger Operator String beginnt mit

vcStringContains	5	String enthält
vcStringEqual	1	String gleich
vcStringIn	7	String enthält
vcStringNotBeginsWith	4	String beginnt nicht mit
vcStringNotContains	6	String enthält nicht
vcStringNotEqual	2	String nicht gleich
vcStringNotIn	8	String nicht enthalten in

Methoden

IsValid

Methode von VcFilterSubCondition

Diese Methode prüft, ob die Filterbedingung korrekt formuliert ist.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Filterbedingung korrekt (True)/ nicht korrekt (False)

7.37 VcGantt

Gantt

Ein VcGantt-Objekt ist das VARCHART-XGantt-Steuerelement. Über Ereignisse können Sie dessen Interaktionen kontrollieren. Das VcGantt-Objekt kann durch eine Vielzahl von Eigenschaften und Methoden den Anforderungen entsprechend konfiguriert werden.

Eigenschaften

- ActiveNodeFilter
- AllowMultipleBoxMarking
- AllowNewBoxes
- AllowNewNodes
- AllowNumericScaleRescale
- AllowPanningMode
- AllowSelectionViaRubberRect
- AllowTableColumnWidthOptimization
- AllowTimescaleRescale
- AllowVerticalNodeMovement
- AllowVerticalNodeMovementViaTable
- Arrangement
- ArrowKeyMode
- ArrowKeyStepSizeMultiplier
- AssignCalendarToNodes
- BarSeparationGroupBy
- BorderArea
- BoxCollection
- BoxFormatCollection
- CalendarCollection
- CalendarGridCollection
- CalendarProfileCollection
- ConfigurationName
- ConsiderLinkRelationTypesOnNodeDragging
- ContextMenuForBoxesEnabled
- CtrlCXVProcessing
- CurrentVersion
- DataDefinition
- DataTableCollection
- DateLineCollection

- DateLineGridCollection
- DateOutputFormat
- DiagramAlternatingRowBackColor
- DiagramBackColor
- DiagramHistogramHeightRatio
- DiagramHistogramHeightRatioEx
- DiagramVisible
- DialogFont
- DirectDataWritingModeEnabled
- DoubleOutputFormat
- EditNewNode
- Enabled
- EnableSupplyTextEntryEvent
- EventReturnStatus
- EventsSecurityCheck
- EventText
- ExtendedDataTables
- ExtendedEditingBehavior
- FilePath
- FilterCollection
- FontAntiAliasingEnabled
- GroupCollection
- GroupingField
- GroupingModificationsAllowed
- GroupingOrderField
- GroupingSortOrder
- GroupLevelLayoutCollection
- GroupOptimizationOnInteractionsEnabled
- HierarchyDataFieldIndex
- HierarchyLevelLayout
- HistogramCollection
- HistogramSeparationLineColor
- hWnd
- InfoWindow
- InInteractionEventsEnabled
- InPlaceEditingOnGroupsInDiagramEnabled
- InPlaceEditingOnGroupsInTableEnabled
- InPlaceEditingOnNodesInDiagramEnabled
- InPlaceEditingOnNodesInTableEnabled
- InteractionMode

- LayerCollection
- LegendView
- LineFormatCollection
- LinkAppearanceCollection
- LinkCollection
- LinkPredecessorDataFieldIndex
- LinksDataTableName
- LinkSuccessorDataFieldIndex
- LinkTypeDataFieldIndex
- MapCollection
- MinimumRowHeight
- MouseProcessingEnabled
- MoveAllSelectedNodes
- MoveLayersAsNodeWithShiftKey
- MoveNodeAlways
- MoveNodeWhenMarked
- NewNodesViaDoubleClick
- NodeCalendarNameDataFieldIndex
- NodeCollection
- NodeDurationDataFieldIndex
- NodeEndDateDataFieldIndex
- NodeLevelLayout
- NodeRowNumberDataFieldIndex
- NodesDataTableName
- NodeStartDateDataFieldIndex
- NodeTooltipTextField
- NoOfInitialRows
- OLEDragHorizontalMovementAllowed
- OLEDragMode
- OLEDragViaDiagram
- OLEDragViaTable
- OLEDragWithOwnMouseCursor
- OLEDragWithPhantom
- OLEDropMode
- OverlapLayerEnabled
- OverlapLayerName
- PartialLoadThreshold
- PhantomLayerHeight
- Printer
- ResourceScheduler2

- RightTable
- RightTableDiagramWidthRatio
- RightTableDiagramWidthRatioEx
- RoundedLinkSlantsEnabled
- RowHeightReductionEnabled
- RowMargins
- Sash3DStyleEnabled
- SashThickness
- Scheduler
- ScrollEventsEnabled
- SelectedRowBackColorAsARGB
- ShowNonWorkInterval
- ShowSnapLines
- ShowSnapMarkings
- ShowTimeScaleDialog
- ShowToolTip
- SnapTargetNodesSelectionMode
- SortField
- SortOrder
- SubRowMargins
- SummaryBarsVisible
- Table
- TableCollection
- TableDiagramWidthRatio
- TableDiagramWidthRatioEx
- TimeScaleCollection
- TimeScaleEnd
- TimeScaleStart
- TimeUnit
- TimeUnitsPerStep
- ToolTipChangeDuration
- ToolTipDuration
- ToolTipPointerDuration
- ToolTipShowAfterClick
- TrackingSpaceBackColorAsARGB
- TrackingSpacePattern
- TrackingSpacePatternColorAsARGB
- UpdateBehaviorCollection
- UseHigherDiagramHistogramHeightRatioPrecision
- UseHigherTableDiagramWidthRatioPrecision

- UseSnapTargetsInInteractions
- UseTwinLineSashPhantom
- ViewComponentsBackColor
- ViewComponentsBorderColor
- WaitCursorEnabled
- WorldView
- ZoomFactor
- ZoomingPerMouseWheelAllowed

Methoden

- AboutBox
- Clear
- ClearAll
- ConvertDistance
- DeleteLinkRecord
- DeleteNodeRecord
- DetectDataTableFieldName
- DetectDataTableName
- DetectFieldIndex
- DumpConfiguration
- EditGroup
- EditLink
- EditNode
- EndLoading
- ExportGraphicsToFile
- FitChartIntoView
- FitHistogramsIntoView
- FitRangeIntoView
- GetAValueFromARGB
- GetBValueFromARGB
- GetCurrentComponentStart
- GetCurrentViewDates
- GetCurrentViewDatesAsString
- GetCurrentViewDatesAsVariant
- GetDate
- GetDateAsString
- GetGValueFromARGB
- GetLinkByID
- GetLinkByIDs
- GetNodeByID

- GetRValueFromARGB
- GetViewComponentSize
- GetViewComponentSizeAsVariant
- GroupNodes
- HistogramSetMaxYValue
- IdentifyField
- IdentifyLayerAt
- IdentifyLayerAtAsVariant
- IdentifyObject
- IdentifyObjectAt
- IdentifyObjectAtAsVariant
- InsertLinkRecord
- InsertNodeRecord
- MakeARGB
- Open
- OptimizeTimeScaleStartEnd
- PageLayout
- PrintDirectEx
- PrinterSetup
- PrintIt
- PrintPreview
- PrintToFile
- RecalculateAllStructureCodes
- Reset
- SaveAsEx
- Schedule
- ScrollComponentStartTo
- ScrollToDate
- ScrollToGroupLine
- ScrollToNode
- ScrollToNodeLine
- ShowExportGraphicsDialog
- SortGroups
- SortNodes
- SuspendUpdate
- UpdateLinkRecord
- UpdateNodeRecord
- UpdateRowNumberFields
- Zoom

Ereignisse

- Error
- ErrorAsVariant
- KeyDown
- KeyPress
- KeyUp
- OLECompleteDrag
- OLEDragDrop
- OLEDragOver
- OLEGiveFeedback
- OLESetData
- OLEStartDrag
- OnBoxCreate
- OnBoxCreateComplete
- OnBoxLClick
- OnBoxLDbClick
- OnBoxModify
- OnBoxModifyCompleteEx
- OnBoxRClick
- OnCalendarGridRClick
- OnCurveLClick
- OnCurveLDbClick
- OnCurveModifyComplete
- OnCurveModifyEx
- OnCurveModifyEx2
- OnCurveModifyExAsString
- OnCurveRClick
- OnDataRecordCreate
- OnDataRecordCreateComplete
- OnDataRecordDelete
- OnDataRecordDeleteComplete
- OnDataRecordModify
- OnDataRecordModifyComplete
- OnDataRecordNotFound
- OnDateLineModify
- OnDateLineRClick
- OnDeleteCurvePoint
- OnDeleteCurvePointEx
- OnDiagramLClick

- OnDiagramLDbClick
- OnDiagramRClick
- OnGroupDelete
- OnGroupLClick
- OnGroupLDbClick
- OnGroupModify
- OnGroupModifyComplete
- OnGroupModifyEx
- OnGroupRClick
- OnGroupsMark
- OnGroupsMarkComplete
- OnHelpRequested
- OnHistogramLClick
- OnHistogramLDbClick
- OnHistogramRClick
- OnHistogramsHeight
- OnHistogramsHeightChanged
- OnHistogramsHeightModifyEx
- OnInsertCurvePoint
- OnInsertCurvePointEx
- OnInteractionEndComplete
- OnInteractionModeChange
- OnInteractionModeChangeComplete
- OnInteractionObjectChangingComplete
- OnInteractionStartComplete
- OnLegendViewClosed
- OnLinkCreate
- OnLinkCreateComplete
- OnLinkDelete
- OnLinkDeleteComplete
- OnLinkLClickCltn
- OnLinkLDbClickCltn
- OnLinkRClickCltn
- OnModifyComplete
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnNodeCreate
- OnNodeCreateCompleteEx

- OnNodeDelete
- OnNodeDeleteCompleteEx
- OnNodeLClick
- OnNodeLDbClick
- OnNodeModifyComplete
- OnNodeModifyCompleteEx
- OnNodeModifyEx
- OnNodeRClick
- OnNodeResizeStart
- OnNodesMarkComplete
- OnNodesMarkEx
- OnNumericScaleLClick
- OnNumericScaleLDbClick
- OnNumericScaleRClick
- OnNumericScaleRescale
- OnObjectDrawCompleteEx
- OnObjectDrawEx
- OnOptimizeTableColumnWidth
- OnPreScrollComponent
- OnPreScrollDiagramHor
- OnResourceSchedulingProgress
- OnResourceSchedulingWarning
- OnScrollComponent
- OnScrollDiagramHor
- OnSelectField
- OnShowCurveNameInMenu
- OnShowDate
- OnShowInPlaceEditor
- OnStatusLineText
- OnSupplyTextEntry
- OnSupplyTextEntryAsVariant
- OnTableCaptionLClick
- OnTableCaptionLDbClick
- OnTableCaptionRClick
- OnTableColumnWidth
- OnTableColumnWidthModifyComplete
- OnTableWidth
- OnTableWidthModifyEx
- OnTimeScaleChangeComplete
- OnTimeScaleEndModifyComplete

- OnTimeScaleLClick
- OnTimeScaleLDbClick
- OnTimeScaleRClick
- OnTimeScaleSectionRescaleCompleteEx
- OnTimeScaleSectionRescaleEx
- OnTimeScaleSectionStartModify
- OnTimeScaleStartModifyComplete
- OnToolTipText
- OnToolTipTextAsVariant
- OnViewComponentsSizeModifyComplete
- OnWorldViewClosed
- OnZoomFactorModifyComplete

Eigenschaften

ActiveNodeFilter

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Filter festlegen oder erfragen, der die darzustellenden Knoten selektiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Filterobjekt Standardwert: Nothing

Code-Beispiel

```
Set VcGantt1.ActiveNodeFilter = VcGantt1.FilterCollection. _  
    FilterByName("Milestone")
```

AllowMultipleBoxMarking

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Markieren von mehreren Boxen gleichzeitig zur Laufzeit möglich ist. Ist die Eigenschaft nicht gesetzt, ist zum Markieren mehrerer Boxen das zusätzliche Drücken der STRG-Taste erforderlich. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mehrfachmarkierung von Boxen möglich / nicht möglich Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowMultipleBoxMarking = True
```

AllowNewBoxes**Eigenschaft von VcGantt**

Mit dieser Eigenschaft wird dem Anwender das Anlegen neuer Boxen erlaubt (True) oder verboten (False). Wird diese Eigenschaft auf False gesetzt, ist der **Modus: Boxen erzeugen** nicht verfügbar. Außerdem kann der **InteractionMode** nicht auf **VcCreateBox** gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/ nicht in Kraft (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowNewBoxes = False
```

AllowNewNodes**Eigenschaft von VcGantt**

Mit dieser Eigenschaft wird dem Anwender das Anlegen neuer Vorgänge erlaubt (True) oder verboten (False). Wird diese Eigenschaft auf False gesetzt, kann der Interaktionsmodus **Knoten erzeugen** nicht eingeschaltet werden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Neue Knoten zugelassen/nicht zugelassen Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowNewNodes = False
```

AllowNumericScaleRescale**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die numerische Skala zur Laufzeit reskalierbar sein soll.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	numerische Skala reskalierbar (True)/ nicht reskalierbar (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowNumericScaleRescale = True
```

AllowPanningMode**Nur-Lese-Eigenschaft von VcGantt**

Die Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Bildschirmausschnitt verschieben erlaubt (True)/nicht erlaubt (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AllowSelectionViaRubberRect

Eigenschaft von VcGantt

Mithilfe dieser Eigenschaft lässt sich bestimmen, ob im leeren Diagrammbereich zur Selektion von Knoten ein Gummirechteck erscheint, wenn die Maus mit gedrückter linker Maustaste bewegt wird.

Die Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Markieren per Gummirechteck erlaubt (True)/nicht erlaubt (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AllowTableColumnWidthOptimization

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie dem Anwender erlauben, die Breite der Tabellenspalten automatisch optimieren zu lassen. Die Optimierung einer Tabellenspalte wird ausgelöst, wenn der Anwender auf die Trennlinie zwischen der zu optimierenden Spalte und der Spalte rechts davon doppelklickt. Es wird anschließend das Ereignis **OnOptimize-TableColumnWidth** ausgelöst. Wenn die Veränderung der Spaltenbreite erfolgt ist, wird noch das Ereignis **OnTableColumnWidth** ausgelöst.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Optimierung der Spaltenbreite zugelassen/nicht zugelassen
	Mögliche Werte:	Standardwert: True Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowTableColumnWidthOptimization = False
```


AllowTimescaleRescale

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie dem Anwender erlauben, die Zeitskala interaktiv zu skalieren. Wenn der Anwender die Zeitskala skalieren darf, wird nach dem Skalieren das Ereignis **OnTimeScaleRescale** ausgelöst.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Reskalierung zugelassen/nicht zugelassen Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllowTimescaleRescale = False
```

AllowVerticalNodeMovement

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das vertikale Verschieben von Knoten im Diagramm zugelassen ist. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Knoten** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vertikales Verschieben von Knoten im Diagramm zugelassen/nicht zugelassen Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AllowVerticalNodeMovementViaTable

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das vertikale Verschieben von Knoten in der Tabelle zugelassen ist. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Knoten** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Vertikales Verschieben von Knoten in Tabelle zugelassen/nicht zugelassen Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Arrangement

Eigenschaft von VcGantt

Mit dieser Eigenschaft legen Sie fest, ob Vorgänge hierarchisch oder gruppenweise angeordnet werden sollen. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Kontrollkästchen **Hierarchie** eingestellt werden. Diese Eigenschaft wird wirksam, wenn die Eigenschaft **HierarchyDataFieldIndex** bzw. **GroupDataFieldIndex** entsprechend gesetzt wurde.

	Datentyp	Beschreibung
Parameter:		
Eigenschaftswert	VcArrangementType Mögliche Werte: vcArrangementTypeGroupwise 1 vcArrangementTypeHierarchical 2	Anordnung der Vorgänge gruppenweise oder hierarchisch Standardwert: vcArrangementTypeGroupwise Gruppenweise Anordnung der Vorgänge Hierarchische Anordnung der Vorgänge

Code-Beispiel

```
VcGantt1.GroupLevelLayoutCollection.FirstGroupLevelLayout().GroupDataFieldIndex
= VcGantt1.DetectFieldIndex("Maindata", "Department")
VcGantt1.Arrangement = VcArrangementType.vcArrangementTypeGroupwise
VcGantt1.GroupNodes(True)

// alternativ:

VcGantt1.HierarchyDataFieldIndex = VcGantt1.DetectFieldIndex("Maindata",
"StructureCode")
VcGantt1.Arrangement = VcArrangementType.vcArrangementTypeHierarchical
VcGantt1.GroupNodes(True)
```

ArrowKeyMode

Eigenschaft von VcGantt

Mit dieser Eigenschaft wird der Modus der Pfeiltasten <links> und <rechts> festgelegt. Standardmäßig sind die Pfeiltasten mit navigierenden Funktionen

wie Abrollen des Diagramms oder Bewegen eines markierten Feldes in einem Knoten oder in der Tabelle belegt. Sie können den Pfeiltasten durch diese Eigenschaft verändernde Funktionen zuweisen, so dass der Benutzer einen Knoten verschieben, vergrößern oder verkleinern kann. Das Info-Fenster mit Informationen zur Position verbleibt noch eine kurze Zeit nach Beendigung der Interaktion, damit die Information gelesen werden kann.

Kommt der Knoten beim Verschieben an den Rand des Views, rollt das Diagramm automatisch ab (Autoscroll).

Beim einfachen Drücken der Pfeiltasten wird der Knoten verschoben, beim gleichzeitigen Drücken der <Shift> -Taste wird er verlängert oder verkürzt.

Die Belegung der Pfeiltasten mit verändernden Funktionen ist besonders nützlich bei Diagrammen mit niedriger Auflösung, da das Positionieren von Knoten auf ein genaues Datum mit der Maus unpräzise sein kann. Die Positionierung mittels Pfeiltasten ist genauer, weil in dem Informationsfenster jeder Bewegungsschritt mit einem veränderten Datum angezeigt wird und dadurch eine höhere Auflösung angeboten wird als in der Zeitskala sichtbar ist. Die Schrittweite wird dabei gesteuert über die Eigenschaften **VcGantt.TimeUnit** und **VcGantt.TimeUnitsPerStep** sowie **VcGantt.ArrowKeyStepSizeMultiplier**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Modus der Pfeiltasten <links> und <rechts> Standardwert: 0
	Mögliche Werte: vcStandard 127	Die Pfeiltasten <links> und <rechts> befinden sich in ihren Standard-Modus
	vcResizeOrMoveNode 384	Die Pfeiltasten <links> und <rechts> befinden sich in dem Modus, Knoten zu verändern

Code-Beispiel

```
'Assigning the function to an option button
Private Sub OptionEditNode_Click()
    If OptionStandard.Value = True Then
        VcGantt1.ArrowKeyMode = vcStandard
    Else
        VcGantt1.ArrowKeyMode = vcResizeOrMoveNode
    End If
End Sub
```

ArrowKeyStepSizeMultiplier

Eigenschaft von VcGantt

Über diese Eigenschaft können Sie dem Pfeiltasten-Schrittweitenmultiplikator einen Wert setzen oder diesen erfragen. Die Schrittweite beim Bewegen des Cursors mit der Maus oder mit den Pfeiltasten (s. Eigenschaft **VcGantt.ArrowKeyMode**) wird über die beiden Eigenschaften **VcGantt.TimeUnit** und **VcGantt.TimeUnitsPerStep** festgelegt. Die Multiplikation der dort gesetzten Werte ergibt die Schrittweite: Wenn die Zeiteinheit auf "Tag" gesetzt wurde und die Einheiten pro Schritt auf 2, beträgt die Schrittweite 2 Tage. Da bei der Maus eine größere Bewegung durch weiteres Ziehen möglich ist, bei Tasten aber nicht, gibt es den zusätzlichen Multiplikationsfaktor für die Pfeiltasten. Er wird vom Benutzer durch das zusätzliche Drücken der <Strg>-Taste aktiviert. Wenn Sie diesen Wert auf 10 setzen, beträgt die Schrittweite im o.g. Beispiel 20 Tage pro Tastendruck.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Wert des Multiplikationsfaktors
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
'Reducing the time scale resolution and enlarging the step size
Private Sub CommandExtendScale_Click()
    'Filling up the available space for the Gantt graph by extending the time
    scale
    VcGantt1.TimeScaleEnd = DateSerial(2012, 1, 1)
    ' Reducing the resolution of the time scale by the factor 10
    VcGantt1.TimeScaleCollection.Active.Section(0).UnitWidth =
VcGantt1.TimeScaleCollection.Active.Section(0).UnitWidth / 10
    ' Increasing the multiplier for the arrow keys to advance in larger steps
    VcGantt1.ArrowKeyStepSizeMultiplier = 25
End Sub
```

AssignCalendarToNodes

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen, ob den Vorgängen ein Kalender zugewiesen werden soll. Die Kalenderzuweisung wirkt sich beim Verschieben von Vorgängen aus: Anfang und Ende der Vorgänge werden nicht auf arbeitsfreie Tage gelegt. Außerdem werden beim Berechnen der Dauer von Vorgängen die arbeitsfreien Zeiten berücksichtigt. Standardmäßig ist ein Fünf-Tage-Kalender definiert, Sie können aber auch eigene Kalender

definieren. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kalender wird zugewiesen (True) / nicht zugewiesen (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AssignCalendarToNodes = False
```

BarSeparationGroupBy

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen oder erfragen, nach welchem Datenfeld Knoten in Gruppen ggf. separiert werden. Diese Eigenschaft kann nur angewendet werden, wenn die Knoten gruppiert werden (Seite **Objekte**, Schaltfläche **Gruppierung**, Rahmen **Gruppenweise**) und dort die Gruppierungsoptionen **Jeder Knoten in eigener Zeile** und **Knoten optimiert** gewählt worden sind. Dann können Sie für **BarSeparationGroupBy** ein Datenfeld auswählen. Alle Knoten einer Gruppe, die denselben Wert in diesem Datenfeld haben, werden dann in derselben Zeile angeordnet, auch wenn es dabei zu Überlagerungen kommt.

Hinweis: Bitte beachten Sie, dass die Felder vom Datentyp **Integer** oder **Alphanumerisch** sein müssen und der Wertebereich zwischen 1 und long_MAX (2147483647) liegen muss, damit die Eigenschaft zufriedenstellende Ergebnisse liefert. Hat ein Feld den Wert 0, so wird der Knoten nicht mit anderen Knoten der Gruppe zusammengehalten.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Nummer des Feldes, das für die Separierung von Knoten in Gruppen verwendet werden soll
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.BarSeparationGroupBy = 3
```

BorderArea

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf das BorderArea-Objekt, also auf den Titel- und Legendenbereich.

	Datentyp	Beschreibung
Eigenschaftswert	VcBorderArea	Titel- und Legendenbereich

Code-Beispiel

```
Dim borderArea As VcBorderArea
Set borderArea = VcGantt1.BorderArea
```

BoxCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die Box-Auflistung und damit auf die verwendeten Boxen.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxCollection	BoxCollection-Objekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
For Each box In boxCltn
    List1.AddItem (box.Name)
Next
```

BoxFormatCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die Boxformat-auflistung, in der alle zur Verfügung stehenden Boxformate enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxFormatCollection	BoxFormatCollection-Objekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcGantt1.BoxFormatCollection
```

```
For Each boxFormat In boxFormatCltn
    List1.AddItem (boxFormat.Name)
Next
```

CalendarCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die Calendar-Auflistung, in der alle zur Verfügung stehenden Kalender enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcCalendarCollection	CalendarCollection-Objekt

Code-Beispiel

```
Dim calendarCltn As VcCalendarCollection
Dim calendar As VcCalendar

Set calendarCltn = VcGantt1.CalendarCollection
For Each calendar In calendarCltn
    List1.AddItem (calendar.Name)
Next
```

CalendarGridCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die CalendarGrid-Auflistung, in der alle zur Verfügung stehenden Kalenderliniengitter enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcCalendarCollecGridCollection	CalendarGridCollection-Objekt

Code-Beispiel

```
Dim calendarGridCltn As VcCalendarGridCollection
Dim calendarGrid As VcCalendarGrid

Set calendarGridCltn = VcGantt1.CalendarGridCollection
For Each calendarGrid In calendarGridCltn
    List1.AddItem (calendarGrid.Name)
Next
```

CalendarProfileCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die Kalenderprofilauflistung, in der alle zur Verfügung stehenden Kalenderprofile enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcCalendarProfileCollection	CalendarProfileCollection-Objekt

Code-Beispiel

```
Dim calendarProfileCltn As VcCalendarProfileCollection
Dim calendarProfile As VcCalendarProfile

Set calendarProfileCltn = VcGantt1.CalendarProfileCollection
```

ConfigurationName**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie eine Konfigurationsdatei (*.ini) laden, aus der alle relevanten Einstellungen übernommen werden. Das schließt eine zugehörige (ggf. andere) Datenschnittstelle (*.ifd) ein. Als Konfigurationsdatei können Sie eine lokale Datei mit Pfad oder eine URL angeben.

lokale Datei: Die Standard-Konfigurationsdatei *vcgantt.ini* sollte sich im gleichen Verzeichnis befinden wie *vcgantt.ocx*. Wenn Sie keinen Pfad angeben, wird die Konfigurationsdatei im Installationsverzeichnis erwartet. Falls die angegebene Datei nicht existiert, wird versucht, die Standard-Konfiguration zu laden, die aber nicht notwendigerweise beim Endanwender existiert.

URL: Die Angabe einer URL als Konfigurationsdatei ist nur sinnvoll, wenn die Konfiguration über die API zur Laufzeit festgelegt wird, da dann die *ini*- und *ifd*-Dateien von der angegebenen URL heruntergeladen werden. (Gibt man schon zur Designzeit eine URL als Konfigurationsdatei an, werden die *ini*- und *ifd*-Dateien zwar ebenfalls heruntergeladen, aber im Structured Storage (bei VB: *frx*-Datei) gespeichert. Dieser Speicher wird dann zur Laufzeit verwendet, statt die Dateien direkt herunterzuladen.) Sie können also, wenn Sie das VARCHART-ActiveX-Steuerelement in eine HTML-Seite einbetten, die URL für die *ini*- und *ifd*-Datei direkt setzen und müssen keine anderen Mechanismen verwenden, um temporär eine lokale Datei zu erzeugen, was von Browsern als unsicher eingestuft wird.

Siehe auch "Einleitung: Verwendung des VARCHART ActiveX in HTML-Seiten"

Hinweis: Beim Einlesen einer neuen Konfigurationsdatei gehen die bestehenden Daten verloren und müssen ggf. wieder eingelesen werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Dateiname Standardwert: vcgantt.ini
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ConfigurationName = "c:\VARCHART\XGantt\sample.ini"
' or:
VcGantt1.ConfigurationName = "http://members.tripod.de/netronic_te/_
                                xgantt_sample.ini"
```

ConsiderLinkRelationTypesOnNodeDragging**Eigenschaft von VcGantt**

Wenn diese Eigenschaft auf **True** gesetzt wird, werden beim Ziehen von Knoten und eingeschalteten Verbindungen die Phantomlinien, die die Verbindungen zu verschobenen Knoten und deren Anschlussknoten darstellen, typgerecht dargestellt, d.h. sie beginnen/enden nicht in der Mitte der Knotenphantome, sondern auf deren linken oder rechten Seite.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung

Code-Beispiel

```
VcGantt1.ConsiderLinkRelationTypesOnNodeDragging = True
```

ContextMenuForBoxesEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob das Kontextmenü für Boxen erscheinen kann.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kontextmenü für Box wird/wird nicht angezeigt
	Mögliche Werte:	

Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

CtrlCXVProcessing

Eigenschaft von VcGantt

Mit dieser Eigenschaft werden die Tastenkombinationen Strg+C, Strg+X und Strg+V automatisch in die Zwischenablage-Operationen **CopyNodesToClipboard**, **CutNodesToClipboard** bzw. **PasteNodesFromClipboard** übersetzt. Dieses Verhalten kann abgeschaltet werden, damit beispielsweise in Visual Basic-Applikationen keine Konflikte mit Tastenkombinationen (Shortcuts) für Menüpunkte entstehen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Tastenkombinationen werden/werden nicht in Zwischenablage-Operationen übersetzt Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.CtrlCXVProcessing = False
```

CurrentVersion

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die aktuelle Versionsnummer eines VARCHART XGantt-Objekts erfragen. Damit wird es auf einfache Art möglich, die zur Laufzeit im Kundensystem registrierte Version zu erkennen und ggf. eine Reparatur der Installation anzufordern, falls eine zu alte Version erkannt wird. Die Versionsnummer kann ebenfalls über die Eigenschaftenseite der Datei **vcgantt.ocx** im Tab **Version** eingesehen sowie über die FILEVERSION-Ressource dieser Datei gelesen werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Versionsnummer
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
MsgBox VcGantt1.CurrentVersion
```

DataDefinition

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf das aktuelle VcDataDefinition-Objekt, um darin z. B. Feldnamen oder Feldtypen abzufragen.

Die Datendefinition des VcGantt hat zwei Datendefinitionstabellen: vcMaindata und vcRelations.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataDefinition	Datendefinition

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition  
Set dataDefinition = VcGantt1.DataDefinition
```

DataTableCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die DataTable-Auflistung und auf die darin verwendeten Datentabellen.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataTableCollection	Ermitteltes Auflistungsobjekt der Datentabellen

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection  
Dim dataTable As VcDataTable  
  
Set dataTableCltn = VcGantt1.DataTableCollection  
For Each dataTable In dataTableCltn  
    List1.AddItem (dataTable.Name)  
Next
```

DateLineCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die DateLine-Auflistung, in der alle zur Verfügung stehenden Stichtaglinien enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcDateLineCollection	DateLineCollection-Objekt

Code-Beispiel

```

Dim dateLineCltn As VcDateLineCollection
Dim dateLine As VcDateLine

Set dateLineCltn = VcGantt1.DateLineCollection
For Each dateLine In dateLineCltn
    List1.AddItem (dateLine.Name)
Next

```

DateLineGridCollection**Nur-Lese-Eigenschaft von VcGantt**

Mit dieser Eigenschaft haben Sie Zugriff auf die DateLineGrid-Auflistung, in der alle zur Verfügung stehenden Terminliniengitter enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcDateLineGridCollection	DateLineGridCollection-Objekt

Code-Beispiel

```

Dim dateLineGridCltn As VcDateLineGridCollection
Dim dateLineGrid As VcDateLineGrid

Set dateLineCltn = VcGantt1.DateLineCollection
For Each dateLine In dateLineCltn
    List1.AddItem (dateLine.Name)
Next

```

DateOutputFormat**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie das Ausgabeformat von Terminen einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM: Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

angepasst werden)

MM: zweistellige Monatsnummer: 01-12

MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)

YY: zweistellige Jahreszahl

YYYY: vierstellige Jahreszahl

WW: zweistellige Nummer der Kalenderwoche: 01-53

TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

Q: einstellige Quartalsnummer: 1-4

TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

hh: Stunde zweistellig im 24-Stunden-Format: 00-23

HH: Stunde zweistellig im 12-Stunden-Format: 01-12

Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

mm: Minute zweistellig: 00-59

ss: Sekunde zweistellig: 00-59

TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

Das hier definierte Format wirkt sich im Tabellenteil und im Knotenbereich bei Layerbeschriftungen aus. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datum {DMYhms::;}
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.DateOutputFormat = "DD.MM.YY"
```

DiagramAlternatingRowBackColor

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie eine zweite Hintergrundfarbe für das Diagramms setzen oder erfragen, die mit der durch die Eigenschaft **DiagramBackColor** gesetzten Hintergrundfarbe ein zeilenweise alternierendes Farbmuster bildet. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255})
		Standardwert: (255,255,255)

Code-Beispiel

```
VcGantt1.DiagramAlternatingRowBackColor = RGB(255, 0, 0)
```

DiagramBackColor

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Diagramms setzen oder erfragen. Zusammen mit der Eigenschaft **DiagramAlternatingRowBackColor** können Sie ein zeilenweise alternierendes Farbmuster einrichten. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255})
		Standardwert: (255,255,255)

Code-Beispiel

```
VcGantt1.DiagramBackColor = RGB(255, 0, 0)
```

DiagramHistogramHeightRatio**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie das prozentuale Verhältnis von der Höhe des Diagrammbereichs (also des Diagramms ohne Histogramm) und der Höhe des Histogramms beim Start erfragen oder festlegen. Bei den Werten -1 und 0 wird das Histogramm beim Start komplett angezeigt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer {-1, 0, 1, ..., 1000}	Verhältnis von Diagrammhöhe und Histogrammhöhe

Code-Beispiel

```
Dim ratio As Integer
```

```
ratio = VcGantt1.DiagramHistogramHeightRatio
```

DiagramHistogramHeightRatioEx**Eigenschaft von VcGantt**

Mit dieser Eigenschaft lässt sich das Verhältnis der Gesamthöhe des Diagramms (in %) zur Höhe des Histogramms erfragen oder festlegen.

Diese Eigenschaft unterscheidet sich von der Eigenschaft **DiagramHistogramHeightRatio** durch die Rückgabe des Datentyps "Double", mit dem eine höhere Genauigkeit erzielt wird. Die Verwendung dieser Eigenschaft muss zuvor über die Eigenschaft **UseHigherDiagramHistogramHeightRatioPrecision** oder über die Option **Höhere Genauigkeit für das Höhenverhältnis zwischen Diagramm und Histogramm(en)** auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhenverhältnis

Code-Beispiel

```
VcGantt1.DiagramHistogramHeightRatioEx = 40
```

DiagramVisible

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob der Diagrammbereich (Tabelle und GanttGraph) sichtbar sein soll. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Diagrammbereich sichtbar (True) / nicht sichtbar (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.DiagramVisible = False
```

DialogFont

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Schriftgröße und den Schrifttyp der zur Laufzeit erscheinenden Dialogfelder im VARCHART XGantt erfragen oder festlegen. Als Objekt wird ein Font-Objekt Ihrer Programmierungsumgebung erwartet, z. B. bei Visual Basic ein Objekt der Klasse **StdFont**.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute

Code-Beispiel

```
Dim newFont As New StdFont

newFont.Size = 14
newFont.Name = "Verdana"

Set VcGantt1.DialogFont = newFont
```

DirectDataWritingModeEnabled

Eigenschaft von VcGantt

Ist diese Eigenschaft auf "True" gesetzt, werden Datenänderung, die über **VcNode/VcLink/VcDataRecord/.set_DataField** bzw. **.AllData** ausgeführt werden, direkt in den Datenbestand übernommen, OHNE Auswertungen (z.B. Filterauswertungen, Mapping etc.) durchzuführen. Dies führt zu einer Verbesserung der Performance.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Datenänderungen ohne Auswertungen werden (True)/werden nicht (False) ausgeführt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

DoubleOutputFormat

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie das Ausgabeformat von Zahlen als double-Wert im Gantt-Diagramm einstellen oder erfragen. Das Format kann über folgende Zeichen dargestellt werden:

- Text
- I
- D

sowie die Trennzeichen Komma und Punkt. Dabei steht **Text** für einen beliebigen Text, **I** für die Ziffern vor dem Dezimaltrenner und **D** für je eine Ziffer hinter dem Dezimaltrenner. Die erlaubte, generelle Reihenfolge ist **Text I D Text**, wobei Komma und Punkt an beliebiger Stelle eingefügt werden können. Als Beispiel sei die Zahl -284901,3458 gegeben. Über das Format **I,DDDD ppm** wird sie als **-284901,3458 ppm** dargestellt. Über das Format **\$I,III.DD** wird sie als **\$-284,901.35** dargestellt.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenfolge, die das Double-Format beschreibt, z.B. "I,DDDD ppm"
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.DoubleOutputFormat = "I,DDDD ppm"
```

EditNewNode

Eigenschaft von VcGantt

Mit dieser Eigenschaft wird festgelegt, ob bei der Erzeugung eines neuen Knotens das Dialogfeld **Vorgänge bearbeiten** erscheinen soll oder nicht. Die Eigenschaft **AllowNewNodes** muss auf **True** gesetzt sein, damit überhaupt ein neuer Knoten erzeugt werden kann. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vorgänge bearbeiten -Dialogfeld erscheint /erscheint nicht Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.EditNewNode = True
```

Enabled

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie das VARCHART-XGantt-Steuerelement so abschalten, dass es auf Maus- und Tastenbefehle nicht reagiert.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	VARCHART-ActiveX-Steuerelement angeschaltet/abgeschaltet
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Enabled = False
```

EnableSupplyTextEntryEvent

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie das Auftreten des Ereignisses **OnSupplyTextEntry** aktivieren oder deaktivieren. Mit Hilfe dieses Ereignisses können Sie die Texte aller Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen und Monats- und Tagesnamen, die zur Laufzeit

erscheinen, verändern, beispielsweise um sie in unterschiedliche Sprachen zu übersetzen.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Eigenschaft in Kraft/nicht in Kraft Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.EnableSupplyTextEntryEvent = True
```

EventReturnStatus

Eigenschaft von VcGantt

Diese Eigenschaft benötigen Sie nur dann, wenn Sie mit einer Entwicklungsumgebung arbeiten, die keinen Rückgabewert in einer Ereignisprozedur erlaubt. Dies ist beispielsweise bei JavaScript der Fall. Mit dieser Eigenschaft überschreiben Sie innerhalb der Ereignismethode den defaultmäßigen returnStatus mit dem gewünschten Wert. Jede Setzung wirkt nur innerhalb des Ereignisses, in dem sie erfolgt ist.

	Datentyp	Beschreibung
Eigenschaftswert	ReturnStatusEnum Mögliche Werte: vcRetStatDefault 2 vcRetStatFalse 0 vcRetStatNoPopup 4 vcRetStatOK 1	Rückgabewert des Ereignisses Standardwert: vcRetStatOK Das Default-Verhalten wird nicht verändert. Das Default-Verhalten wird nicht durchgeführt. Das Aufspringen des Rechte-Maustasten-Menüs wird unterdrückt. Das Default-Verhalten wird durchgeführt.

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramRClick(ByVal x As Long, ByVal y As Long,
returnStatus As Variant)

    VcGantt1.EventReturnStatus = vcRetStatNoPopup

End Sub
```

EventsSecurityCheck

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Event-Sicherheitsprüfung an- oder abschalten. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Event-Sicherheitsprüfung an/aus Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.EventsSecurityCheck = False
```

EventText

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft benötigen Sie nur dann, wenn Sie mit einer Entwicklungsumgebung arbeiten, die keine Setzung des Übergabeparameters in einer Ereignisprozedur erlaubt. Dies ist beispielsweise bei JavaScript der Fall. Mit dieser Eigenschaft setzen Sie den ToolTipText. Jede Setzung wirkt nur innerhalb des Ereignisses, in dem sie erfolgt ist.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Tool Tip Name der Farbzusordnungstabelle

Code-Beispiel

```
Private Sub VcGantt1_OnSupplyTextEntry(ByVal controlIndex As
VcGanttLib.TextEntryIndexEnum, TextEntry As String, returnStatus As Variant)

    VcGantt1.EventText = "Order189"
End Sub
```

ExtendedDataTables

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie zwischen der Beschränkung auf zwei Datentabellen (Maindata und Relations) und der weiterentwickelten Verwendung von bis zu 90 Datentabellen wechseln. Die Verwendung der

zweiten Option wird empfohlen. Diese Eigenschaft muss zu Beginn des Programms gesetzt werden, bevor Datentabellen und Datensätze angelegt werden.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	False: nur zwei Datentabellen (Maindata und Relations) True: bis zu 99 Datentabellen Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ExtendedDataTables = True
```

ExtendedEditingBehavior

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob zur Laufzeit die erweiterten Möglichkeiten zur Tabellenbearbeitung genutzt werden können. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	erweitertes Editierverhalten in Tabellen an/aus Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ExtendedEditingBehavior = True
```

FilePath

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einen Verzeichnispfad setzen, damit auch Grafikdateien, bei denen nur ein relativer Dateiname angegeben ist, in dem

angegebenen Verzeichnis gefunden werden. Andernfalls wird die Datei in dem gerade aktiven Arbeitsverzeichnis der Applikation und im Installationsverzeichnis des VARCHART-ActiveX-Steuerelements gesucht.

Es empfiehlt sich, diese Eigenschaft beim Start der Applikation während des Initialisierens des VARCHART-ActiveX-Steuerelements zu setzen. Der Einfachheit halber sollte als Verzeichnispfad der Verzeichnispfad der Applikation oder ein Unterverzeichnis davon verwendet werden. Dies hat den Vorteil, dass die Applikation in einem beliebigen Verzeichnis installiert sein kann.

	Datentyp	Beschreibung
Eigenschaftswert	String	Verzeichnispfad Standardwert: " "
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim graphicsPath As String

graphicsPath = App.Path & "\bitmaps"
VcGantt1.FilePath = graphicsPath
```

FilterCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die Filter-Auflistung und damit auf die verwendeten Filter.

	Datentyp	Beschreibung
Eigenschaftswert	VcFilterCollection	FilterCollection-Objekt

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcGantt1.FilterCollection
For Each filter In filterCltn
    List1.AddItem (filter.Name)
Next
```

FontAntiAliasingEnabled

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Schriftzeichen optisch per GDI+ geglättet werden sollen. Wenn dies bei bestimmten Schriftarten, insbesondere bei nichtlateinischen Zeichen, zu verminderter Lesefähigkeit führt, sollte die Eigenschaft auf **False** gesetzt werden.

Die Glättung per GDI+ bewirkt zusätzlich, dass die Texte bei jeder Zoomstufe die gleiche relative Ausdehnung haben, so dass immer dieselbe Anzahl Zeichen z.B. in ein Tabellenfeld passt. Wenn diese Eigenschaft aber auf False steht, dann wird stattdessen die Einstellung des Betriebssystems übernommen (einstellbar in der **Systemsteuerung**, Dialogfeld **Anzeige**, Reiter **Darstellung: Effekte**). Wenn dort eine Kantenglättung eingeschaltet ist, dann werden also Texte weiterhin geglättet. Es kann dann aber sein, dass bei manchen Zoomstufen mehr Text sichtbar ist als bei anderen, weil die systemeigene Kantenglättung dies nicht garantiert.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Schriftzeichen werden optisch geglättet/nicht geglättet.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

GroupCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht bei gruppierter Darstellung den Zugriff auf die Gruppen-Auflistung, in der alle verfügbaren Gruppen enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroupCollection	GroupCollection-Objekt

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
For Each group In groupCltn
```

```
List1.AddItem (group.Name)
Next
```

GroupingField

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen oder erfragen, welches Feld aus der Datendefinitionstabelle als Gruppierkriterium auf einer bestimmten Ebene verwendet werden soll. Die Sortierung der Gruppen erfolgt standardmäßig in der Reihenfolge, in der jeweils der erste Vorgang einer Gruppe eingelesen wird. Dieses Verhalten kann über die Eigenschaft **GroupingOrderField** verändert werden.

Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ GroupingLevel	Integer Mögliche Werte:	Gruppierenebene (beginnend mit 0) Datenfeldindex
Eigenschaftswert	Integer Mögliche Werte:	Feld-ID aus der Datendefinitionstabelle Datenfeldindex

Code-Beispiel

```
Dim definitionTable As VcDataDefinitionTable

Set definitionTable = VcGantt1.DataDefinition.DefinitionTable(vcMaingroup)
VcGantt1.GroupingField(0) = definitionTable.FieldByName("Code 1").ID
VcGantt1.GroupNodes (True)
```

GroupingModificationsAllowed

Eigenschaft von VcGantt

Mit dieser Eigenschaft stellen Sie ein, ob der Anwender expandierte Gruppen kollabieren oder kollabierte Gruppen wieder expandieren kann. Der Anwender kann dann die Gruppen durch einen Doppelklick auf die Gruppenüberschrift im Tabellenteil oder durch einen Klick auf das Minus- bzw. Plus-Zeichen neben der Gruppenüberschrift oder über das Kontextmenü für Gruppen kollabieren bzw. expandieren.

Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ groupingLevel	Integer Mögliche Werte:	Gruppierungsebene Datenfeldindex
Eigenschaftswert	Boolean Mögliche Werte:	Änderungen erlaubt (True)/ nicht erlaubt (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.GroupingModificationsAllowed(0) = False
```

GroupingOrderField**Eigenschaft von VcGantt**

Mit dieser Eigenschaft stellen Sie ein, welches Feld der Datendefinitionstabelle zum Sortieren der Gruppen verwendet werden soll. Durch Verwendung von **GroupingOrderField** können Sie die Sortierung in alphabetischer Reihenfolge auf- oder absteigend nach diesem Feld erzielen. Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ groupingLevel	Integer Mögliche Werte:	Gruppierungsebene Datenfeldindex
Eigenschaftswert	Long	Feldindex aus der Datendefinitionstabelle

Code-Beispiel

```
VcGantt1.GroupingOrderField (0) = 12
VcGantt1.GroupingSortOrder (0) = vcDescending
VcGantt1.SortGroups
```

GroupingSortOrder**Eigenschaft von VcGantt**

Mit dieser Eigenschaft legen Sie fest, ob die Gruppen auf- oder absteigend sortiert werden. Das Feld, nach dem die Gruppen sortiert werden, legen Sie mit der Eigenschaft **GroupingOrderField** fest. Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ groupingLevel	Integer Mögliche Werte:	Gruppierungsebene Datenfeldindex
Eigenschaftswert	SortOrderEnum Mögliche Werte: vcAscending 1 vcDescending 2	aufsteigende oder absteigende Sortierreihenfolge Standardwert: vcAscending aufsteigende Abfolge absteigende Abfolge

Code-Beispiel

```
VcGantt1.GroupingOrderField (0) = 12
VcGantt1.GroupingSortOrder (0) = vcAscending
VcGantt1.SortGroups
```

GroupLevelLayoutCollection**Nur-Lese-Eigenschaft von VcGantt**

Mit dieser Eigenschaft haben Sie Zugriff auf die GroupLevelLayout-Auflistung, in der alle zur Verfügung stehenden Gruppierungsebenenlayouts enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroupLevelLayoutCollection	GroupLevelLayoutCollection-Objekt

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout

Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
For Each groupLevelLayout In groupLevelLayoutCltn
    List1.AddItem (groupLevelLayout.Name)
Next
```

GroupOptimizationOnInteractionsEnabled**Eigenschaft von VcGantt**

Ist diese Eigenschaft eingeschaltet, dann werden bei Interaktionen wie Aufziehen von Knoten, Knotenverschieben horizontal oder vertikal und Ändern des Start- oder Endtermins eines Layers die Knoten der gesamten Zielgruppe automatisch erneut optimiert, wenn sie bisher bereits optimiert dargestellt wird. Wird diese Eigenschaft abgeschaltet, dann wird bei denselben Interaktionen der Knoten unter der Cursorposition platziert, wenn

dies nicht zu einer Überlappung führt. Bei einer möglichen Überlappung wird der Knoten in die nächste Zeile zu anderen Knoten platziert, wenn dies dort keine Überlappung verursacht. Wenn doch, dann bekommt der Knoten eine eigene Zeile unter der, an der der Mauscursor sich befindet.

Die Eigenschaft kann zur Designzeit auch in der Eigenschaftenseite **Allgemeines** eingestellt werden.

Siehe auch Methode **VcGroup.ReOptimizeNodes**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Die optimierte Anordnung nach einer Interaktion erfolgt (True) / erfolgt nicht (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

HierarchyDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, nach dem Vorgänge hierarchisch angeordnet werden sollen. Dies kann auch erfolgen **nachdem** bereits Daten geladen wurden. Damit Änderungen wirksam werden, muss über die Eigenschaft **VcGantt.Arrangement** die hierarchische Anordnung der Vorgänge (**vcArrangementTypeHierarchical**) eingestellt, sowie eine Aktualisierung über die Methode **VcGantt.GroupNodes** vorgenommen werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, nach dem Vorgänge hierarchisch angeordnet werden sollen

Code-Beispiel

```
VcGantt1.HierarchyDataFieldIndex = VcGantt1.DetectFieldIndex("Maindata",
"Hierarchy")
VcGantt1.Arrangement = vcArrangementTypeHierarchical
VcGantt1.GroupNodes True
```

HierarchyLevelLayout

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf das HierarchyLevelLayout-Objekt. Mit diesem Objekt können Sie die Eigenschaften einer hierarchischen Anordnung setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcHierarchyLevelLayout	HierarchyLevelLayout-Objekt

HistogramCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die Histogramm-Auflistung, in der alle zur Verfügung stehenden Histogramme enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcHistogramCollection	HistogramCollection-Objekt

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
For Each histogram In histogramCltn
    List1.AddItem (histogram.Name)
Next
```

HistogramSeparationLineColor

Eigenschaft von VcGantt

Mit dieser Eigenschaft setzen oder erfragen Sie die Farbe für die Trennlinien zwischen Histogrammen. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Layout** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: (255,255,255)

hWnd

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft gibt eine Zugriffsnummer zurück. Die Arbeitsumgebung Microsoft Windows kennzeichnet jedes Formular und jedes Steuerelement in einer Anwendung durch Zuweisen einer Zugriffsnummer mit der Bezeichnung **hWnd**. Die Eigenschaft **hWnd** wird im Zusammenhang mit dem Aufruf der Windows-API verwendet. Viele Funktionen der Windows-Arbeitsumgebung benötigen die hWnd-Zugriffsnummer des aktiven Fensters als Argument.

Hinweis: Der Wert dieser Eigenschaft kann sich während der Programmausführung ändern. Sie sollten den Wert von **hWnd** daher nicht in einer Variablen speichern.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Zugriffsnummer

Code-Beispiel

```
MsgBox (Me(hWnd))
```

InfoWindow

Nur-Lese-Eigenschaft von VcGantt

Über diese Eigenschaft haben Sie Zugriff auf das VcInfoWindow-Objekt, das das Informationsfenster eines Knotens bezeichnet, das in einem Balkenplan beim Anlegen oder Verändern des Knotens erscheint.

	Datentyp	Beschreibung
Eigenschaftswert	VcInfoWindow	InfoWindow-Objekt

InInteractionEventsEnabled

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Verwendung von InInteractionEvents zulassen oder unterbinden. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	InInteractionEvents ein- (True) oder ausgeschaltet (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.InInteractionEventsEnabled = True
```

InPlaceEditingOnGroupsInDiagramEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob zur Laufzeit das direkte Editieren von Gruppendatenfeldern im Diagramm-Bereich möglich ist. Voraussetzung ist die Verwendung von eigenen Datentabellen für Gruppendaten. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Hinweis: Falls einzelne Datenfelder nicht editierbar sein sollen, darf im Dialog **Datentabellen verwalten** das Attribut **editierbar** nicht ausgewählt sein.

S. ebenfalls **InPlaceEditingOnNodesInDiagramEnabled**, **InPlaceEditingOnNodesInTableEnabled** und **InPlaceEditingOnGroupsInTableEnabled**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren möglich (True) / nicht möglich (False)
	Mögliche Werte:	Standardwert: True Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.InPlaceEditingOnGroupsInDiagramEnabled = True
```

InPlaceEditingOnGroupsInTableEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob zur Laufzeit das direkte Editieren von Gruppendatenfeldern in der Tabelle möglich ist.

Voraussetzung ist die Verwendung von eigenen Datentabellen für Gruppendaten. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Hinweis: Falls einzelne Datenfelder nicht editierbar sein sollen, darf im Dialog **Datentabellen verwalten** das Attribut **editierbar** nicht ausgewählt sein.

S. ebenfalls **InPlaceEditingOnNodesInDiagramEnabled**, **InPlaceEditingOnNodesInTableEnabled** und **InPlaceEditingOnGroupsInDiagramEnabled**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren möglich (True) / nicht möglich (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.InPlaceEditingOnGroupsInTableEnabled = True
```

InPlaceEditingOnNodesInDiagramEnabled

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob zur Laufzeit das direkte Editieren von Knotendatenfeldern im Diagramm-Bereich möglich ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Hinweis: Falls einzelne Datenfelder nicht editierbar sein sollen, darf im Dialog **Datentabellen verwalten** das Attribut **editierbar** nicht ausgewählt sein.

S. ebenfalls **InPlaceEditingOnNodesInTableEnabled**, **InPlaceEditingOnGroupsInTableEnabled** und **InPlaceEditingOnGroupsInDiagramEnabled**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren möglich (True) / nicht möglich (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.InPlaceEditingOnNodesInDiagramEnabled = True
```

InPlaceEditingOnNodesInTableEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob zur Laufzeit das direkte Editieren von Knotendatenfeldern in der Tabelle möglich ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Hinweis: Falls einzelne Datenfelder nicht editierbar sein sollen, darf im Dialog **Datentabellen verwalten** das Attribut **editierbar** nicht ausgewählt sein.

S. ebenfalls **InPlaceEditingOnNodesInDiagramEnabled**, **InPlaceEditingOnGroupsInTableEnabled** und **InPlaceEditingOnGroupsInDiagramEnabled**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren möglich (True) / nicht möglich (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.InPlaceEditingOnNodesInTableEnabled = True
```


InteractionMode

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einen der verfügbaren Interaktionsmodi einstellen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	InteractionModeEnum	Interaktionsmodus Standardwert: vcPointer
	Mögliche Werte: vcCreateBox 36 vcCreateLink 4 vcCreateNode 2 vcDeleteLink 5 vcDeleteNode 3 vcPanning 6 vcPointer 0	Erzeugemodus für Boxen Erzeugemodus für Verbindungen Erzeugemodus für Knoten Löschmodus für Verbindungen Löschmodus für Knoten Verschiebemodus Selektiermodus

Code-Beispiel

```
VcGantt1.InteractionMode = vcCreateNode
```

LayerCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die Layer-Auflistung, in der alle verfügbaren Layer enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcLayerCollection	LayerCollection-Objekt

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
For Each layer In layerCltn
    List1.AddItem (layer.Name)
Next
```

LegendView

Nur-Lese-Eigenschaft von VcGantt

Über diese Eigenschaft erhalten Sie Zugriff auf das VcLegendView-Objekt, das die Legendenansicht des Diagramms definiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcLegendView	LegendView-Objekt

Code-Beispiel

```
Dim legendview As VcLegendView

Set legendview = VcGantt1.LegendView
legendview.Visible = True
```

LineFormatCollection**Nur-Lese-Eigenschaft von VcGantt**

Mit dieser Eigenschaft haben Sie Zugriff auf die Linienformatauflistung, in der alle zur Verfügung stehenden Linienformate enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcLineFormatCollection	LineFormatCollection-Objekt

LinkAppearanceCollection**Nur-Lese-Eigenschaft von VcGantt**

Mit dieser Eigenschaft haben Sie Zugriff auf die LinkAppearanceCollection, in der alle zur Verfügung stehenden LinkAppearance-Objekte zusammengefasst sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkAppearanceCollection	LinkAppearanceCollection-Objekt

Code-Beispiel

```
Dim linkAppCltn As VcLinkAppearanceCollection
Dim linkApp As VcLinkAppearance

Set linkAppCltn = VcGantt1.LinkAppearanceCollection
For Each linkApp In linkAppCltn
    List1.AddItem (linkApp.Name)
Next
```

LinkCollection**Nur-Lese-Eigenschaft von VcGantt**

Diese Eigenschaft ermöglicht den Zugriff auf die Link-Auflistung, in der alle Verbindungen enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkCollection	LinkCollection Objekt

Code-Beispiel

```

Dim linkCltn As VcLinkCollection
Dim link As VcLink

Set linkCltn = VcGantt1.LinkCollection
For Each link In linkCltn
    List1.AddItem (link.AllData)
Next

```

LinkPredecessorDataFieldIndex**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, in dem die Identifizierung des Vorgängerknotens der Verbindung enthalten ist. Dies ist nur möglich, solange noch keine Daten geladen wurden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** gesetzt werden.

	Datentyp	Beschreibung
Parameter: identifizierIndex	Integer Mögliche Werte:	Index des Vorgängerknotens {0...2} Datenfeldindex
Eigenschaftswert	Long	Feldindex aus der Datendefinitionstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcGantt2.DataTableCollection.Add("LinkDataTable")
VcGantt1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")
dataTable.DataTableFieldCollection.Add ("Successor")
VcGantt1.DataTableCollection.Update

VcGantt1.LinkPredecessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")
VcGantt1.LinkSuccessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")

'Load Data
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcGantt1.EndLoading

```

LinksDataTableName

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Namen der Tabelle setzen oder erfragen, die die Felder für die Verbindungen enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle aus der die Verbindungen kommen
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcGantt2.DataTableCollection.Add("LinkDataTable")
VcGantt1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add("Predecessor")
dataTable.DataTableFieldCollection.Add("Successor")
VcGantt1.DataTableCollection.Update

VcGantt1.LinkPredecessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")
VcGantt1.LinkSuccessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")

'Load Data
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcGantt1.EndLoading
```

LinkSuccessorDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, in dem die Identifizierung des Nachfolgerknotens der Verbindung enthalten ist. Dies ist nur möglich, solange noch keine Daten geladen wurden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** gesetzt werden.

	Datentyp	Beschreibung
Parameter: identifizierIndex	Integer Mögliche Werte:	Index des Vorgängerknotens {0...2} Datenfeldindex
Eigenschaftswert	Long	Feldindex aus der Datendefinitionstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcGantt2.DataTableCollection.Add("LinkDataTable")
VcGantt1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")
dataTable.DataTableFieldCollection.Add ("Successor")
VcGantt1.DataTableCollection.Update

VcGantt1.LinkPredecessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")
VcGantt1.LinkSuccessorDataFieldIndex(0) =
VcGantt1.DetectFieldIndex("LinkDataTable", "Id")

'Load Data
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcGantt1.EndLoading

```

LinkTypeDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das den Verbindungstyp enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Verbindungstypen enthält

Code-Beispiel

```

Dim dataTable As VcDataTable

'create Link DataTable
Set dataTable = VcGantt1.DataTableCollection.Add("LinkDataTable")
VcGantt1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")

```

```
dataTable.DataTableFieldCollection.Add ("Successor")
dataTable.DataTableFieldCollection.Add ("LinkType")
VcGantt1.DataTableCollection.Update
```

MapCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die Map-Auflistung, in dem bestimmte Zuordnungstabellen enthalten sind. Die enthaltenen Map-Objekte werden durch die Methode **VcMapCollection.SelectMaps** ausgewählt.

	Datentyp	Beschreibung
Eigenschaftswert	VcMapCollection	MapCollection-Objekt

Code-Beispiel

```
Dim mapCltn As VcMapCollection

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
```

MinimumRowHeight

Eigenschaft von VcGantt

Mit dieser Eigenschaft wird die minimale Höhe einer Zeile (Einheit: 1/100 mm) eingestellt. Die eingestellte Höhe sollte etwa der mittleren Höhe der Darstellung eines Vorgangs entsprechen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** eingestellt werden.

Die minimale Höhe ist nur dann wirksam, wenn sich kein Vorgang oder nur Vorgänge mit geringerer grafischer Höhe in der Zeile befinden. In allen anderen Fällen wird die Zeilenhöhe entsprechend dem erforderlichen Platzbedarf automatisch vergrößert. Der einstellbaren Werte liegen zwischen 2 und 1000.

	Datentyp	Beschreibung
Eigenschaftswert	Long	minimale Zeilenhöhe

Code-Beispiel

```
VcGantt1.MinimumRowHeight = 100
```

MouseProcessingEnabled

Eigenschaft von VcGantt

Diese Eigenschaft kann dazu genutzt werden, Ihre eigene Verarbeitung von Mausereignissen zu ermöglichen. Wenn Sie eine eigene Verarbeitung vom Ereignis **OnMouseDown** bis zum **OnMouseUp** durchführen möchten, setzen Sie die Eigenschaft **MouseProcessingEnabled** für diese Zeit auf False. Dann ignoriert VARCHART XGantt bis zum Zurücksetzen auf True alle Mausbewegungen und Klicks.

Diese Eigenschaft kann auch in den OnMouse*-Ereignissen gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/ nicht in Kraft (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub VcGantt1_OnMouseDown(ByVal button As Integer, ByVal Shift As Integer, ByVal x As Long, ByVal y As Long)
```

```
    VcGantt1.MouseProcessingEnabled = False
```

```
End Sub
```

MoveAllSelectedNodes

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob der Anwender mehrere markierte Knoten gemeinsam verschieben können soll. Andernfalls können nur einzelne Layer oder Knoten (je nachdem, ob auf der Eigenschaftenseite **Knoten** die Option **Markierte Knoten als Ganzes verschieben** ausgewählt wurde oder nicht) bewegt werden, auch wenn mehrere Knoten markiert sind).

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mehrere markierte Knoten gemeinsam verschiebbar (True)/nur einzelne Layer bzw. Knoten verschiebbar, auch wenn mehrere Knoten markiert sind (False)
	Mögliche Werte:	

Gruppe unsichtbar/sichtbar
Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.MoveAllSelectedNodes = True
```

MoveLayersAsNodeWithShiftKey**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob alle Layer eines markierten Knotens gleichzeitig mit der Maus verschoben werden können, wenn während des Verschiebens die Umschalttaste gedrückt wird (True) oder ob jeder Layer eines markierten Knotens nur einzeln verschoben werden kann (False). Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verschieben von allen Layern eines Knotens mit Umschalttaste zugelassen/nicht zugelassen
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.MoveLayersAsNodeWithShiftKey = False
```

MoveNodeAlways**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob ein Knoten und damit alle seine Layer verschoben werden können, ohne dass sie zuvor markiert wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Verschieben von kompletten Knoten ohne vorherige Markierung ist an- (true) oder abgeschaltet (false)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.MoveNodeAlways = True
```

MoveNodeWhenMarked**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob ein markierter Knoten als ganzes mit der Maus verschoben werden kann (True) oder ob jeder Layer eines markierten Knotens nur einzeln verschoben werden kann (False). Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Markierter Knoten als ganzes mit der Maus verschiebbar (True)/jeder Layer eines markierten Knotens nur einzeln verschiebbar (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.MoveNodeWhenMarked = True
```

NewNodesViaDoubleClick**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie steuern, ob der Anwender neue Knoten per Doppelklick (im Diagrammbereich) anlegen kann oder nicht. Voraussetzung ist, dass die Eigenschaft **AllowNewNodes** auf True gesetzt ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Anlegen neuer Knoten per Doppelklick zugelassen/nicht zugelassen Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.NewNodesViaDoubleClick = True
```

NodeCalendarNameDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das den Namen des für einen Knoten zu verwendenden Kalenders enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Namen des für einen Knoten zu verwendenden Kalenders enthält

NodeCollection

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf die Knotenauflistung, in der bestimmte Knoten enthalten sind. Diese Knoten werden durch die Methode **VcNodeCollection.SelectNodes** ausgewählt.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	NodeCollection-Objekt

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection

Set nodeCltn = VcGantt1.NodeCollection
nodeCltn.SelectNodes (vcAll)
```

NodeDurationDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das die Dauer eines interaktiv angelegten Knotens enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Dauer eines interaktiv angelegten Knotens enthält

NodeEndDateDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen/erfragen, in das der Endtermin eines interaktiv angelegten Vorgangs geschrieben werden soll. Dies ist nur möglich, solange noch keine Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Endtermin eines interaktiv angelegten Vorgangs enthält

NodeLevelLayout

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf das NodeLevelLayout-Objekt. Mit diesem Objekt können Sie die Eigenschaften einer hierarchischen Anordnung setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeLevelLayout	NodeLevelLayout-Objekt

NodeRowNumberDataFieldIndex

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen/erfragen, in das die Zeilennummer jedes Vorgangs geschrieben werden soll. Dies ist nur möglich, solange noch keine Daten geladen wurden. Damit Änderungen wirksam werden, muss eine Aktualisierung über die Methode **VcGantt.UpdateRowNumberFields** vorgenommen werden.

Die Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Zeilennummer jedes Vorgangs enthält

Code-Beispiel

```

Private Sub Form_Load()

VcGantt1.NodeRowNumberDataFieldIndex =
VcGantt1.DetectFieldIndex("NodeDataTable", "SortNumber")

    'Load data
    Call loadData

    VcGantt1.UpdateRowNumberFields
    VcGantt1.SaveAsEx "C:\ProjectData.txt", vcUnicodeEncoding
End Sub

```

NodesDataTableName**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Namen der Tabelle setzen oder erfragen, die die Felder für die Darstellung der Knoten zur Verfügung stellt. Dies ist nur möglich, solange noch keine Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle aus der die Knotendaten kommen
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Node DataTable
Set dataTable = VcGantt1.DataTableCollection.Add("NodeDataTable")
VcGantt1.NodesDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
'Load Data
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("NodeDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;Node One;")
Set dataRecord = dataTable.DataRecordCollection.Add("2;Node Two;")

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcGantt1.EndLoading

```

NodeStartDateDataFieldIndex**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen/erfragen, in das der Starttermin eines interaktiv angelegten Vorgangs geschrieben werden soll. Dies ist nur möglich, solange noch keine

Daten geladen wurden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Starttermin eines interaktiv angelegten Vorgangs enthält

NodeTooltipTextField

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, welches Datenfeld eines Knotens für Tooltips in VMF-Dateien genutzt werden soll. Drückt man im WebViewer die rechte Maustaste, erscheint der zugeordnete Text. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Knoten-Datenfelds für Tooltiptexte Standardwert: 4
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.NodeTooltipTextField = 1
```

NoOfInitialRows

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Anzahl der Knotenzeilen beim Programmstart setzen oder erfragen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Knotenzeilen beim Programmstart

Code-Beispiel

```
VcGantt1.NoOfInitialRows = 1
```

OLEDragHorizontalMovementAllowed

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob ein Knoten auch dann horizontal verschoben werden kann, wenn das Steuerelement die Zielkomponente eines im Moment laufenden OLE Drag&Drop-Vorgangs ist. Für Verschiebeoperationen, die nur innerhalb eines Gantt-Diagramms stattfinden, ist diese Eigenschaft nicht wirksam.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	OLE Drag&Drop Aktion erlaubt (True) / nicht erlaubt (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.OLEDragHorizontalMovementAllowed = True
```

OLEDragMode

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Ziehen eines Knotens über die Grenze der VARCHART-XGantt-Komponente hinaus erlaubt sein soll.

Wenn **OLEDragMode** auf **vcOLEDragManual** gesetzt wird, ist ein OLE-Drag-Vorgang nicht möglich. Bei **vcOLEDragAutomatic** hingegen wird das Ziehen eines Knotens über die Grenzen der VARCHART XGantt-Komponente automatisch gestartet.

Beim Start des Vorgangs füllt die Quellkomponente das **DataObject** mit den Daten des gezogenen Knotens und setzt den Effekt-Parameter, um damit das OLEStartDrag-Ereignis sowie andere quellenseitige OLE Drag & Drop-Ereignisse auszulösen. Dies gibt Ihnen die Kontrolle über die Drag&Drop-Operation und erlaubt Ihnen, einzugreifen, z. B. um andere Datenformate hinzuzufügen.

VARCHART XGantt verpackt die Daten in das Standard-Zwischenablage-Format CF_TEXT (für Visual-Basic-Benutzer: das vbCFText-Format), das mühelos gelesen werden kann.

Während des Ziehens kann der Benutzer mit Hilfe der **Strg**-Taste entscheiden, ob das Objekt verschoben oder kopiert werden soll.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

Das OLE Drag & Drop-Verhalten des VARCHART-XGantt ist kompatibel zu dem in Visual Basic üblichen, das heißt, die Methoden, Eigenschaften und Ereignisse tragen dieselben Namen und haben dieselbe Bedeutung wie bei den Standardobjekten aus Visual Basic.

	Datentyp	Beschreibung
Eigenschaftswert	OLEDragModeEnum	Verschiebemodus für Bewegungen aus der VARCHART-ActiveX-Komponente heraus Standardwert: vcOLEDragManual
	Mögliche Werte: vcOLEDragAutomatic 1 vcOLEDragManual 0	Methode OLEDrag wird automatisch aufgerufen Methode OLEDrag muß eigens aufgerufen werden.

Code-Beispiel

```
VcGantt1.OLEDragMode = vcOLEDragAutomatic
```

OLEDragViaDiagram

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob OLE-DragDrop im Diagrammbereich erlaubt sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	OLE-DragDrop im Diagramm erlaubt/nicht erlaubt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

OLEDragViaTable

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob OLE-DragDrop im Tabellenbereich erlaubt sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	OLE-DragDrop in Tabelle erlaubt/nicht erlaubt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

OLEDragWithOwnMouseCursor

Eigenschaft von VcGantt

Diese Eigenschaft bestimmt, ob während eines OLE-Drag-Vorgangs der Cursor in der Zielkomponente gesetzt werden soll. Bei OLE Drag & Drop ist es möglich, den Zeiger in der Quellkomponente über das Ereignis **OLEGiveFeedback** zu setzen. Daher würde ein Setzen durch die Zielkomponente zu einem Flimmern der konkurrierenden Cursor führen, was man über diese Eigenschaft beeinflussen kann.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mauszeiger wird/wird nicht in der Zielkomponente gesetzt
	Mögliche Werte:	Standardwert: True Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.OLEDragWithOwnMouseCursor = False
```

OLEDragWithPhantom

Eigenschaft von VcGantt

Diese Eigenschaft bestimmt, ob während eines OLE-Drag-Vorgangs ein Phantom erscheinen soll oder nicht. Das Abschalten des Phantoms ist für Anwendungen gedacht, die beim Hineindraggen eines Objekts kein neues Objekt erzeugen, sondern z. B. den Knoten, auf dem dann ein Objekt fallengelassen wird, nur neu attributieren. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Phantom erscheint/erscheint nicht Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.OLEDragWithPhantom = False
```

OLEDropMode**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob ein Knoten aus einer anderen VARCHART-XGantt-Komponente in die aktuelle Komponente hinein gezogen werden darf.

Bei **OLEDropNone** ist dies nicht erlaubt. Bei **vcOLEDropManual** erhalten Sie beim Dropping das Ereignis **OLEDragDrop**, so dass Sie die übertragenen Daten selbst weiterverarbeiten können, um z. B. einen Knoten zu erzeugen oder eine Datei einzulesen. Wenn Quell- und Zielkomponente identisch sind, erhalten Sie wie bei abgeschaltetem OLE Drag&Drop eins der Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**. Bei **vcOLEDropAutomatic** wird der Dropping-Vorgang von der Komponente selbst verarbeitet, das heißt es wird, falls möglich, ein Knoten an entsprechender Mausposition erzeugt.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

Das OLE Drag & Drop-Verhalten des VARCHART XGantt ist kompatibel zu dem in Visual Basic üblichen, das heißt, die Methoden, Eigenschaften und Ereignisse tragen dieselben Namen und haben dieselbe Bedeutung wie bei den Standardobjekten aus Visual Basic.

	Datentyp	Beschreibung
Eigenschaftswert	OLEDropModeEnum	Verschiebemodus für Bewegungen in die VARCHART-ActiveX-Komponente herein Standardwert: vcOLEDropNone
	Mögliche Werte:	

vcOLEDropAutomatic 2	Die Daten des übertragenen Objektes werden automatisch verarbeitet, d. h. es wird ein den Daten entsprechender Knoten an der Stelle des Droppings erzeugt.
vcOLEDropManual 1	Das Ereignis OLEDragDrop wird aufgerufen, und die Daten des übertragenen Objektes müssen vom Programmierer verarbeitet werden.
vcOLEDropNone 0	Dropping von nicht eigenen Objekten ist in der VARCHART-ActiveX-Komponente nicht zugelassen.

Code-Beispiel

```
VcGantt1.OLEDropMode = vcOLEDropAutomatic
```

OverlapLayerEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Überlappungs-Layer des Diagramms aktivieren. Bitte sehen Sie auch **OverlapLayerName** sowie beim Layer-Objekt **UsedAsOverlapLayer**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Überlappungslayer ein (True) / aus (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

OverlapLayerName**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Layer, der im Diagramm als Überlappungs-Layer dienen soll, über seinen Namen festlegen oder erfragen. Der Überlappungs-Layer muss vorher beim Layer-Objekt angelegt und mittels **UsedAsOverlap Layer** als Überlappungs-Layer gekennzeichnet werden sowie schließlich über **OverlapLayerEnabled** beim Gantt-Objekt aktiviert werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Überlappungsl-Layers
	Mögliche Werte:	Name der Farbzuoordnungstabelle

PartialLoadThreshold

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen, oder erfragen ab welcher Knotenzahl von einem optimierten Teilupdate auf ein komplettes Update der Datensätze umgeschaltet wird.

Standardmäßig wird beim Hinzufügen eines neuen Datensatzes ein Ladezyklus gestartet, der intern für das Laden großer Datenmengen optimiert ist: Strukturen wie Gruppierung und Sortierung, Summenbalkenberechnungen usw. werden verworfen und komplett neu erstellt. Dies ist beim Laden großer Datenmengen in einen leeren Plan vorteilhaft, beim Nachladen weniger Datensätze in eine schon bestehende Datenstruktur jedoch kann damit im Extremfall das Nachladen eines einzigen Knotens in einen vorhandenen Plan mit z.B. 4000 Vorgängen, 500 Gruppen, sortierten Knoten, berechneten Summenbalken, positionierten Links usw. einen kompletten Neuaufbau dieser Strukturen bewirken. Damit würde das Nachladen des einen Knotens genauso lange dauern wie das Laden der 4000 Vorgänge, da der Aufbau dieser Strukturen den Hauptteil der Performance ausmacht.

Die Eigenschaft **PartialLoadThreshold** bietet hier Abhilfe: Eine kleine Datenmenge wird durch ein Teilupdate in eine große vorhandene Datenmenge optimiert eingefügt. Der Wert, der hier angegeben wird, stellt den Schwellenwert für die Anzahl von Datensätzen dar, ab dem von einem "kleinen" auf ein "großes" Update umgeschaltet wird.

Der optimale Schwellenwert, hängt von verschiedenen Faktoren in der jeweiligen Applikation ab und muss vom Anwender individuell ermittelt werden:

- Anzahl der vorhanden Knoten,
- Komplexität des Gantts (z.B. Gruppierung, Sortierung, Summenbalken, Links, Mapping usw.)

Die Eigenschaft sollte in erster Linie genutzt werden, wenn bereits viele Knoten vorhanden sind, und man zur Laufzeit noch einige wenige hinzufügen möchte.

Diese Eigenschaft kann auch bei den Eigenschaften des Steuerlementes gesetzt werden:

OLEDragWithPhantom	True
OLEDropMode	0 - vcOLEDropNone
OverlapLayerEnabled	True
OverlapLayerName	
PartialLoadThreshold	0
PhantomLayerHeight	200
RightTableDiagramWidthRatio	-1
RoundedLinkSlantsEnabled	False
RowHeightReductionEnabled	False
RowMargins	50
ScrollEventsEnabled	True
SelectedRowBackColorAsARGB	0
ShowNonWorkInterval	False
ShowTimeScaleDialog	True

Hinweis. Die Optimierung beschränkt sich zur Zeit noch auf die **Maindata**-Tabelle. Werden also in einem Ladezyklus auch Daten anderer Tabellen oder Verbindungen geladen, wird die Einstellung ignoriert.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Knoten, ab der von Teilupdate auf Vollupdate umgeschaltet wird

PhantomLayerHeight

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Höhe eines Layer-Phantoms beim Aufziehen eines Knotens in 1/100 mm einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe des Layerphantoms
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim phantomLayerHeight As Integer
phantomLayerHeight = VcGantt1.PhantomLayerHeight
```

Printer

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf das Printer-Objekt. Mit diesem Objekt können Sie die Eigenschaften des aktuell verwendeten Druckers erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	VcPrinter	Druckerobjekt

Code-Beispiel

```
Dim printerZoomfactor As Integer
Dim printerCuttingMarks As String

printerZoomfactor = VcGantt1.Printer.ZoomFactor
printerCuttingMarks = VcGantt1.Printer.CuttingMarks
```

ResourceScheduler2

Eigenschaft von VcGantt

Diese Eigenschaft setzt das ResourceScheduler2-Objekt für die Ressourcenplanung.

	Datentyp	Beschreibung
Eigenschaftswert	VcResourceScheduler2	Übergebenes ResourceScheduler2-Objekt

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskDataTableName = "Task"
VcGantt1.ResourceScheduler2.TaskDueDateFieldIndex = 1
VcGantt1.ResourceScheduler2.TaskReleaseDateFieldIndex = 2
'...
VcGantt1.ResourceScheduler2.Process
```

RightTable

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf das rechtsseitige Tabellenobjekt, um auf dessen verwendete Formate zugreifen oder die Tabellenspalten und -überschriften verändern zu können.

	Datentyp	Beschreibung
Eigenschaftswert	VcTable	Zweite, rechtsseitige Tabelle

Code-Beispiel

```
Dim rightTable As VcTable
```

```
Set rightTable = VcGantt1.RightTable
```

RightTableDiagramWidthRatio

Eigenschaft von VcGantt

Mit dieser Eigenschaft lässt sich das Verhältnis der Breite der rechten Tabelle zur Gesamtbreite des Diagramms (in %) erfragen oder festlegen. Beim Wert -1 wird die Tabelle immer komplett dargestellt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breitenverhältnis {-1, 1...100}
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.RightTableDiagramWidthRatio = 40
```

RightTableDiagramWidthRatioEx

Eigenschaft von VcGantt

Mit dieser Eigenschaft lässt sich das Verhältnis der Breite der rechten Tabelle zur Gesamtbreite des Diagramms (in %) erfragen oder festlegen. Beim Wert -1 wird die Tabelle immer komplett dargestellt.

Diese Eigenschaft unterscheidet sich von der Eigenschaft **RightTableDiagramWidthRatio** durch die Rückgabe des Datentyps "Double", mit dem eine höhere Genauigkeit erzielt wird. Die Verwendung dieser Eigenschaft muss zuvor über die Eigenschaft **UseHigherTableDiagramWidthRatioPrecision** oder über die Option **Höhere Genauigkeit für die Breitenverhältnisse zwischenTabelle(n) und Diagramm** auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breitenverhältnis

Code-Beispiel

```
VcGantt1.RightTableDiagramWidthRatioEx = 40
```

RoundedLinkSlantsEnabled

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob die Schrägen bei Verbindungen vom Routing-Typ **vcLRTOrthogonalDistinguishable** als Viertelkreise statt als gerade Linien dargestellt werden sollen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	System.Boolean	Schrägen bei Verbindungen werden/werden nicht als Viertelkreise dargestellt Standardwert: falsch

Code-Beispiel

```
VcGantt1.RoundedLinkSlantsEnabled = True
```

RowHeightReductionEnabled

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft beeinflusst das Verfahren zur Berechnung der Zeilenhöhe im Diagramm. Ist sie auf **false** gesetzt, werden die vertikalen Offsets der Layer ausgehend von einer gedachten Nulllinie in der vertikalen Mitte einer Knotenzeile wirksam. Damit diese Nulllinie stets in der Mitte einer Zeile bleibt, kann so entweder der obere oder der untere Rand einer Zeile sehr groß erscheinen, wobei aber Layer mit vertikalem Offset von 0 immer vertikal zentriert bleiben.

Wenn die Eigenschaft auf **true** gesetzt ist, wird zwar weiterhin die gedachte Nulllinie verwendet, diese befindet sich jedoch nicht mehr zwingend in der Zeilenmitte, sondern wird so positioniert, dass für die Zeilenhöhe das kleinstmögliche Maß verwendet werden kann. Dadurch kann es vorkommen, dass Layer mit einem vertikalen Offset von 0 nicht mehr auf gleicher Höhe liegen wie der vertikal zentrierte Text der zugehörigen Tabellenzeile.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Reduzierung der Zeilenhöhe erlaubt (true)/nicht erlaubt (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar

	Gruppen sichtbar/unsichtbar
--	-----------------------------

Code-Beispiel

```
VcGantt1.RowHeightReductionEnabled = True
```

RowMargins**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den Abstand zwischen dem oberen/ unteren Knotenrand und dem oberen/ unteren Rand der Knotenzeilen in 1/100 mm festlegen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Abstand zwischen dem oberen/ unteren Knotenrand und dem oberen/ unteren Rand der Knotenzeilen in 1/100 mm

Code-Beispiel

```
VcGantt1.RowMargins = 100
```

Sash3DStyleEnabled**Eigenschaft von VcGantt**

Mithilfe dieser Eigenschaft kann gesetzt oder erfragt werden, ob der Sash im 3D-Modus angezeigt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	3D-Ansicht Sash an/aus
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

SashThickness**Eigenschaft von VcGantt**

Mithilfe dieser Eigenschaft kann die Dicke des Sashes gesetzt oder erfragt werden. Wertebereich: 3 - 20 Pixel.

	Datentyp	Beschreibung

Scheduler

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft gibt das VcScheduler-Objekt zurück.

	Datentyp	Beschreibung
Eigenschaftswert	VcScheduler	Gibt das VcScheduler-Objekt zurück

ScrollEventsEnabled

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Scroll Events an- oder abschalten. Dies kann auch auf der Eigenschaftenseite **Allgemeines** geschehen.

Hinweis: Die Scroll-Events sind standardmäßig **nicht** aktiviert!

	Datentyp	Beschreibung
Eigenschaftswert	System.Boolean	Scroll events an/aus

Code-Beispiel

```
VcGantt1.ScrollEventsEnabled = True
```

SelectedRowBackColorAsARGB

Eigenschaft von VcGantt

Diese Eigenschaft gibt die Farbe an, mit der die gerade selektierte Zeile eingefärbt wird. Sie können bei dieser Farbe auch einen Alpha-Wert verwenden, der die Transparenz für die Färbung einstellt, um z.B. einen Farbschleier über die normale Hintergrundfarbe der Zeile zu legen (s. Eigenschaften DiagramBackColor und DiagramAlternatingRowBackColor). Standardmäßig ist das Einfärben abgeschaltet, indem ein volltransparenter Farbwert vorgegeben ist.

Der Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). R, G und B können nicht (!) mit dem üblichen RGB-

Makro zusammengebaut werden. Am einfachsten schreibt man einen Farbwert in hexadezimaler Schreibweise: &haarrggbb (VB6) bzw. 0xaarrggbb (C++), wobei aa, rr, gg und bb jeweils im Bereich 00..FF (entspricht dezimal 0-255) liegen dürfen. Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255, 255 ist weiß.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Farbwert Standardwert: 0

Code-Beispiel

```
VcGantt1.SelectedRowBackgroundColorAsARGB = &h77503DFF
```

ShowNonWorkInterval

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob arbeitsfreie Zeiten im Knoten dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

Hinweis: **AssignCalendarToNodes** muss hierfür auf True gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	arbeitsfreie Intervalle anzeigen (True)/nicht anzeigen (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ShowNonWorkInterval = False
```

ShowSnapLines

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft legen Sie fest, ob beim Verschieben mit definierten Einrastzielen entsprechende Linien gezeigt werden, damit der/die definierten Einrastziele besser zu erkennen sind.

Diese Funktionalität kann auch auf der Eigenschaftenseite **Knoten** aktiviert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Einrastlinien werden/werden nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowSnapMarkings

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft legen Sie fest, ob beim Verschieben mit definierten Einrastzielen an den als Ziel definierten Knoten eine Markierung gezeigt wird, damit der/die definierten Einrastziele besser zu erkennen sind.

Diese Funktionalität kann auch auf der Eigenschaftenseite **Knoten** aktiviert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Einrastmarkierungen werden/werden nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowTimeScaleDialog

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob für den Anwender bei einem Doppelklick auf die Zeitskala das Dialogfeld **Zeitskala bearbeiten** erscheinen soll. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zeitskalendialog erscheint /erscheint nicht
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ShowTimeScaleDialog = False
```

ShowToolTip**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie das Ereignis **OnToolTipText** aktivieren oder deaktivieren. Mit Hilfe dieses Ereignisses können Sie die Texte der Tooltips verändern. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ShowToolTip = True
```

SnapTargetNodesSelectionMode**Eigenschaft von VcGantt**

Mit dieser Eigenschaft legen Sie fest, ob beim Verschieben mit definierten Einrastzielen Knoten automatisch oder manuell selektiert werden.

Mit der Eigenschaft **VcNode.SnapTargetMode** werden die Knoten bei manueller Selektion als mögliche Einrastziele ausgewählt.

	Datentyp	Beschreibung
Eigenschaftswert	SnapTargetNodesSelectionModeEnum	Selektionsmodus für Knoten beim Verschieben mit Einrastzielen Standardwert: vcAutomatically
	Mögliche Werte: vcAutomatically 1 vcUserSelection 2	Automatische Selektion von Knoten Manuelle Selektion von Knoten

SortField

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Sortierfelder für Ihre Knoten festlegen. Es gibt drei Sortierungsebenen, für die jeweils ein Feldindex der Datendefinitionstabelle angegeben werden kann. Die Sortierungsrichtung wird über die Eigenschaft **SortOrder** eingestellt. Die Sortierung wird durch die Methode **SortNodes** ausgelöst.

Die Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Parameter: ⇒ sortLevel	Integer Mögliche Werte:	Sortierebene {0...2} Datenfeldindex
Eigenschaftswert	Integer Mögliche Werte:	Feldindex aus der Datendefinitionstabelle Datenfeldindex

Code-Beispiel

```
VcGantt1.SortField (0) = 11
VcGantt1.SortOrder (0) = vcDescending

VcGantt1.SortNodes
```

SortOrder

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie für jede der drei Sortierebenen die Sortierrichtung festlegen. Die Sortierung wird durch die Methode **SortNodes** ausgelöst. Sie können die Eigenschaft **SortOrder** auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte: vcAscending 1 vcDescending 2	aufsteigende oder absteigende Sortierreihenfolge Standardwert: vcAscending aufsteigende Abfolge absteigende Abfolge

Code-Beispiel

```
VcGantt1.SortField (0) = 11
VcGantt1.SortOrder (0) = vcDescending

VcGantt1.SortNodes
```

SubRowMargins**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie den vertikalen Abstand von Subzeilen in /100 mm festlegen. Diese Subzeilen existieren nur, wenn Gruppen optimiert dargestellt werden und daraus Knoten in einer jeweiligen Gruppenzeile in mehrere Subzeilen verteilt werden, um nicht zu überlappen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Abstand zwischen Subzeilen in 1/100 mm {0...200} Standardwert: 50

Code-Beispiel

```
VcGantt1.SubRowMargins = 100
```

SummaryBarsVisible**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Summenbalken sichtbar sind oder nicht.

	Datentyp	Beschreibung
Parameter: ⇒ GroupingLevel	Integer	(nicht für Hierarchie) Gruppierungsebene (GroupingLevel = -1: beim Setzen: alle Ebenen, beim Lesen: mindestens eine Ebene)
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Boolean	Summenbalken sichtbar (True)/ unsichtbar (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.SummaryBarsVisible (-1) = False
```

Table

Nur-Lese-Eigenschaft von VcGantt

Diese Eigenschaft ermöglicht den Zugriff auf das Tabellenobjekt, um so auf die verwendeten Formate zugreifen und darin die Tabellenspalten oder ihre Überschriften verändern zu können.

	Datentyp	Beschreibung
Eigenschaftswert	VcTable	Tabelle

Code-Beispiel

```
Dim table As VcTable
Set table = VcGantt1.Table
```

TableCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die Tabellenauflistung, in dem alle zur Verfügung stehenden Tabellen enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcTableCollection	TableCollection-Objekt

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim table As VcTable

Set tableCltn = VcGantt1.TableCollection
For Each table In tableCltn
    List1.AddItem (table.Name)
Next
```

TableDiagramWidthRatio

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft lässt sich das Verhältnis der Breite der linken Tabelle zur Gesamtbreite des Diagramms (in %) erfragen oder festlegen. Beim Wert -1 wird die Tabelle immer komplett dargestellt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breitenverhältnis {-1, 1...100}
	Mögliche Werte:	

Datenfeldindex

Code-Beispiel

```
VcGantt1.LeftTableDiagramWidthRatio = 40
```

TableDiagramWidthRatioEx**Eigenschaft von VcGantt**

Mit dieser Eigenschaft lässt sich das Verhältnis der Breite der linken Tabelle zur Gesamtbreite des Diagramms (in %) erfragen oder festlegen. Beim Wert -1 wird die Tabelle immer komplett dargestellt.

Diese Eigenschaft unterscheidet sich von der Eigenschaft **LeftTableDiagramWidthRatio** durch die Rückgabe des Datentyps "Double", mit dem eine höhere Genauigkeit erzielt wird. Die Verwendung dieser Eigenschaft muss zuvor über die Eigenschaft **UseHigherTableDiagramWidthRatio-Precision** oder über die Option **Höhere Genauigkeit für die Breitenverhältnisse zwischenTabelle(n) und Diagramm** auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

	Datentyp	Beschreibung

Code-Beispiel

```
VcGantt1.LeftTableDiagramWidthRatioEx = 40
```

TimeScaleCollection**Nur-Lese-Eigenschaft von VcGantt**

Diese Eigenschaft ermöglicht den Zugriff auf die TimeScale-Auflistung und damit auf die verfügbaren Zeitskalen.

	Datentyp	Beschreibung
Eigenschaftswert	VcTimeScaleCollection	TimeScaleCollection-Objekt

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
Dim timeScale As VcTimeScale

Set timeScaleCltn = VcGantt1.TimeScaleCollection
For Each timeScale In timeScaleCltn
    List1.AddItem (timeScale.Name)
Next
```


TimeScaleEnd

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie das Ende der Zeitskala erfragen oder festlegen. Bei der Festlegung muss das Ende immer nach dem Anfang (s. Eigenschaft **TimeScaleStart**) liegen, die Eigenschaften müssen aber in umgekehrter Reihenfolge gesetzt werden. Es ist empfehlenswert, die vorgegebene Reihenfolge im Programmierbeispiel hier zu verwenden.

Hinweis: Das Enddatum ist nicht eingeschlossen. Geben Sie beispielsweise das Enddatum "31.12.2010" an, ist der letzte Tag, der dargestellt wird, der 30.12.2010.

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Enddatum der Zeitskala {1.1.1980...31.12.2035}

Code-Beispiel

```
' Timescale from 1.10.2014 to 30.11.2014
VcGantt1.TimeScaleEnd = "01.12.14"
VcGantt1.TimeScaleStart = "01.10.14"
VcGantt1.TimeScaleEnd = "01.12.14"
```

TimeScaleStart

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie den Anfang der Zeitskala erfragen oder festlegen. Bei der Festlegung muss der Anfang immer vor dem Ende (s. Eigenschaft **TimeScaleEnd**) liegen, die Eigenschaften müssen aber in umgekehrter Reihenfolge gesetzt werden. Es ist empfehlenswert, die vorgegebene Reihenfolge im Programmierbeispiel hier zu verwenden.

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Anfangsdatum der Zeitskala {1.1.1980...31.12.2035}

Code-Beispiel

```
' Timescale from 1.10.2014 to 30.11.2014
VcGantt1.TimeScaleEnd = "01.12.14"
VcGantt1.TimeScaleStart = "01.10.14"
VcGantt1.TimeScaleEnd = "01.12.14"
```

TimeUnit

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Zeiteinheit für die Berechnung der Dauer (siehe "Layer") und für das interaktive Anlegen und Verändern Ihrer Knoten in der Darstellung setzen oder erfragen. Haben Sie beispielsweise die Zeiteinheit Tage gewählt, lassen sich Knoten nur in Sprüngen von ganzen Tagen anlegen und verändern, und die Dauer von Knoten wird ebenfalls in Tagen berechnet. Die Eigenschaft **TimeUnit** kann auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Änderungen der Eigenschaft sollten vor dem Einlesen von Daten, d.h. am besten beim Start erfolgen, weil nachträgliche Änderungen keine Auswirkung haben.

	Datentyp	Beschreibung
Eigenschaftswert	TimeUnitEnum Mögliche Werte: vcDay 5 vcHour 6 vcMinute 7 vcSecond 8	Zeiteinheit Standardwert: vcDay Zeiteinheit Tag Zeiteinheit Stunde Zeiteinheit Minute Zeiteinheit Sekunde

Code-Beispiel

```
Dim timeUnit As TimeUnitEnum
timeUnit = VcGantt1.TimeUnit
```

TimeUnitsPerStep

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie festlegen, wie viele Zeiteinheiten die kleinstmögliche interaktive Veränderung eines Knotens beträgt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** unter **Kleinstes Zeitintervall** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Anzahl der Zeiteinheiten pro Schritt Standardwert: 1 Datenfeldindex

Code-Beispiel

```
VcGantt1.TimeUnitsPerStep = 4
```

ToolTipChangeDuration**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, die vergeht, bevor das nächste ToolTip-Fenster auf dem Bildschirm erscheint, wenn der Mauszeiger auf das nächste Objekt gesetzt wird. Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Standardwert von 98 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden. Maximalwert = 32767 ms Standardwert: -1
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.ToolTipText = "Object"
VcGantt1.ToolTipChangeDuration = 1000
```

ToolTipDuration**Eigenschaft von VcGantt**

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, während der das ToolTip-Fenster sichtbar bleiben soll (sofern der Mauszeiger innerhalb des umgebenden Rechtecks eines Objektes unbewegt bleibt). Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Standardwert von 5.000 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden. Maximalwert = 32767 ms Standardwert: -1
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.ToolTipText = "Object"
VcGantt1.ToolTipDuration = 1000
```

ToolTipPointerDuration

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, während der der Mauszeiger innerhalb des umgebenden Rechtecks eines Objektes unbewegt bleiben muss, damit das ToolTip-Fenster erscheint. Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Maximalwert von 480 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden Standardwert: -1
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.ToolTipText = "Object"
VcGantt1.ToolTipPointerDuration = 1000
```

ToolTipShowAfterClick

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie einstellen, ob das angezeigte ToolTip-Fenster beim Anklicken des Objektes verschwinden soll (Standard-Verhalten) oder entsprechend seiner eingestellten Zeiten weiterhin angezeigt werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	ToolTip-Fenster verschwindet (false) oder bleibt (true) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ToolTipShowAfterClick = True
```

TrackingSpaceBackColorAsARGB

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Trackingbereichs setzen oder erfragen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	RGB-Farbwerte ({0...255},{0...255},{0...255}) Standardwert: (255,255,255)
	Mögliche Werte:	Datenfeldindex

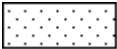
Code-Beispiel

```
VcGantt1.TrackingSpaceBackgroundColor = System.Drawing.Color.Blue
```

TrackingSpacePattern

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie für den Hintergrund desTrackingbereichs ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 

vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
vcDivotPattern 2036	Grassoden-Muster
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien

vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell

vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

TrackingSpacePatternColorAsARGB

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Musterfarbe des Trackingbereichs festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255})
	Mögliche Werte:	Datenfeldindex

UpdateBehaviorCollection

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft haben Sie Zugriff auf die UpdateBehavior-Auflistung, in der alle zur Verfügung stehenden Aktualisierungsverhalten enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcUpdateBehaviorCollection	UpdateBehaviorCollection-Objekt

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.updBehCollection
For Each updBeh In updBehCltn
    List1.AddItem (updBeh.Name)
Next
```

UseHigherDiagramHistogramHeightRatioPrecision

Eigenschaft von VcGantt

Ist diese Eigenschaft auf "True" gesetzt, werden zur Ermittlung des Höhenverhältnisses zwischen Diagramm und Histogramm die genauere Methode **DiagramHistogramHeightRatioEx** bzw. das Ereignis **OnHistogramHeightModifyEx** verwendet, die jeweils einen Wert vom Datentyp "Double" zurückgeben.

Beim Standardwert "False" wird die Methode **DiagramHistogramHeightRatio** bzw. das Ereignis **OnHistogramHeight** verwendet.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Genauere Methoden zur Ermittlung des Höhenverhältnisses Diagramm/Histogramm werden (True)/werden nicht (False) verwendet Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UseHigherTableDiagramWidthRatioPrecision

Eigenschaft von VcGantt

Ist diese Eigenschaft auf "True" gesetzt, werden zur Ermittlung des Breitenverhältnisses zwischen Tabelle(n) und Diagramm die genaueren Methoden **LeftTableDiagramWidthRatioEx** und **RightTableDiagramWidthRatioEx** bzw. das Ereignis **OnTableWidthModifyEx** verwendet, die jeweils einen Wert vom Datentyp "Double" zurückgeben.

Beim Standardwert "False" werden die Methoden **LeftTableDiagramWidthRatio** und **RightTableDiagramWidthRatio** bzw. das Ereignis **OnTableWidth** verwendet.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Genauere Methoden zur Ermittlung des Breitenverhältnisses Tabelle(n)/Diagramm werden (True)/werden nicht (False) verwendet Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UseSnapTargetsInInteractions

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Einrastziele bei Knoten-/Layerinteraktionen verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Einrastziele werden/werden nicht bei Knoten-/Layerinteraktionen verwendet .
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UseTwinLineSashPhantom

Eigenschaft von VcGantt

Mithilfe dieser Eigenschaft kann gesetzt oder erfragt werden, ob beim interaktiven Verschieben eines Sashes im **Standard**-Update-Behavior eine einzelne Phantomlinie oder eine Doppellinie erscheinen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Doppelte Phantomlinie beim interaktiven Verschieben des Sash an/aus Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ViewComponentsBackColor

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Diagramms setzen oder erfragen. Zusammen mit der Eigenschaft **DiagramAlternating-RowBackColor** können Sie ein zeilenweise alternierendes Farbmuster einrichten. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Code-Beispiel

```
VcGantt1.ViewComponentsBackColor = System.Drawing.Color.Blue
```

ViewComponentsBorderColor

Nur-Lese-Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie die Randlinienfarbe aller Fenster gemeinsam setzen oder erfragen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Code-Beispiel

```
VcGantt1.ViewBorderColor = Color.Blue
```

WaitCursorEnabled**Eigenschaft von VcGantt**

Mit dieser Eigenschaft kann man steuern, ob bei zeitkritischen Aufrufen (wie ScheduleProject) ein Wartecursor gesetzt werden soll oder nicht.

Die Eigenschaft kann auch auf der Eigenschaftseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Wartecursor wird gesetzt/wird nicht gesetzt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

WorldView**Nur-Lese-Eigenschaft von VcGantt**

Über diese Eigenschaft erhalten Sie Zugriff auf das VcWorldView-Objekt, das die Komplettansicht des Diagramms definiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcWorldView	Komplettansicht-Objekt

Code-Beispiel

```
Dim worldview As VcWorldView

Set worldview = VcGantt1.WorldView
worldview.Visible = True
```

ZoomFactor

Eigenschaft von VcGantt

Mit dieser Eigenschaft kann der absolute Zoomfaktor der Bildschirmdarstellung in % angegeben oder erfragt werden (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

Der absolute Zoomfaktor ist ein gerundeter Wert und kann daher zu Ungenauigkeiten führen.

Siehe auch die VcGantt-Methoden **FitChartIntoView()** und **Zoom()**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	absoluter Zoomfaktor
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.ZoomFactor = 150
```

ZoomingPerMouseWheelAllowed

Eigenschaft von VcGantt

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das Zoomen per Mausewheel zugelassen ist. Um zu zoomen, muss der Anwender dann die Strg-Taste festhalten und das Mausewheel drehen. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zoomen erlaubt (True)/nicht erlaubt (False)
	Mögliche Werte:	Standardwert: False Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ZoomingPerMouseWheelAllowed = True
```

Methoden

AboutBox

Methode von VcGantt

Mit dieser Methode können Sie die **About**-Box aufrufen. Sie enthält eine Übersicht über die jeweils verwendeten Programm- und Bibliotheksdateien mit absolutem Pfad und Versionsnummer. Dies dient der vereinfachten Hotline-Unterstützung. Die Übersicht kann mit der Maus selektiert und mit <Strg>+<C> herauskopiert werden, um sie z. B. mit <Strg>+<V> in eine Mail einzufügen.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcGantt1.AboutBox
```

Clear

Methode von VcGantt

Mit dieser Methode werden alle bisher an der API erstellten Objekte gelöscht und der zur Entwurfszeit auf den Eigenschaftenseiten eingestellte Zustand wiederhergestellt.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Objekte im Diagramm erfolgreich gelöscht {True}

Code-Beispiel

```
VcGantt1.Clear
```

ClearAll

Methode von VcGantt

Mit dieser Methode werden alle bisher erstellten Objekte gelöscht und der zur Entwurfszeit auf den Eigenschaftenseiten eingestellte Zustand mit Ausnahme der Kalender wiederhergestellt

	Datentyp	Beschreibung
Rückgabewert	Boolean	Objekte im Diagramm erfolgreich gelöscht {True}

Code-Beispiel

```
VcGantt1.ClearAll
```

ConvertDistance**Methode von VcGantt**

Mit dieser Methode können Abstände, die in der Einheit "1/100 Millimeter" vorliegen, in die tatsächliche Anzahl Pixel in X- oder Y-Richtung gewandelt werden und umgekehrt. Die Umrechnung berücksichtigt den aktuell eingestellten Zoomfaktor (s. Eigenschaft **VcGantt.ZoomFactor**).

	Datentyp	Beschreibung
Parameter: ⇒ conversionType	DistanceConversionTypeEnum	Art der Umwandlung
	Mögliche Werte:	
	vcXCentiMillimetersToPixels 1	Umwandlung eines Abstands in X-Richtung von 1/100 mm in Pixel.
	vcXPixelsToCentiMillimeters 3	Umwandlung eines Abstands in X-Richtung von Pixeln in 1/100 Millimeter.
	vcYCentiMillimetersToPixels 2	Umwandlung eines Abstands in Y-Richtung von 1/100 mm in Pixel.
	vcYPixelsToCentiMillimeters 4	Umwandlung eines Abstands in Y-Richtung von Pixeln in 1/100 Millimeter.
⇒ value	Long	Anzahl der Quellen-Einheiten (die umgewandelt werden sollen)
Rückgabewert	Long	Anzahl der Ziel-Einheiten (in welche umgewandelt wurde)

DeleteLinkRecord**Methode von VcGantt**

Mit dieser Methode können Sie eine Verbindung zwischen zwei Knoten löschen. Der Datensatz der Verbindung wird durch die im Dialog **Datentabellen verwalten** festgelegten Primärschlüssel identifiziert.

	Datentyp	Beschreibung
Parameter: ⇒ linkRecordContent	Object	Inhalt des Verbindungsdatensatzes

Rückgabewert	Boolean	Datensatz der Verbindung erfolgreich (True) / nicht erfolgreich (False) gelöscht
---------------------	---------	--

Code-Beispiel

```
VcGantt1.DeleteLinkRecord ("A100;A105;;")
```

DeleteNodeRecord**Methode von VcGantt**

Mit dieser Methode können Sie einen Knoten löschen. Der Knoten wird durch den Primärschlüssel im Datensatz festgelegt. Welches Datenfeld als Primärschlüssel verwendet wird, wird im Dialog **Datentabellen verwalten** festgelegt.

	Datentyp	Beschreibung
Parameter: ⇒ nodeRecordContent	Object	Inhalt des Knotendatensatzes
Rückgabewert	Boolean	Datensatz des Knotens erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
VcGantt1.DeleteNodeRecord "A100;;;;;;"
```

DetectDataTableFieldName**Methode von VcGantt**

Mit dieser Methode können Sie über den Index eines Tabellendatenfeldes seinen Namen erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ fieldIndex	Long	Index des Datentabellenfeldes, dessen Name ermittelt werden soll
Rückgabewert	String	Zurückgegebener Name des Datentabellenfeldes

Code-Beispiel

```
'Find the name of a DataTableField
Dim fieldName As String

fieldName = VcGantt1.DetectDataTableFieldName(0)
```


DetectDataTableName

Methode von VcGantt

Mit dieser Methode können Sie über den Index einer Datentabelle ihren Namen erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ fieldIndex	Long	Index der Datentabelle, deren Name ermittelt werden soll
Rückgabewert	String	Name der Datentabelle

Code-Beispiel

```
'Find the name of a DataTable
Dim tableName As String

tableName = VcGantt1.DetectDataTableName(0)
```

DetectFieldIndex

Methode von VcGantt

Mit dieser Methode können Sie über den Namen einer Datentabelle und den Feldnamen den Index eines Tabellendatenfeldes erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ dataTableNames Mögliche Werte: ⇒ dataTableFieldName	String String	Name der Datentabelle, in der sich das Feld befindet, dessen Index ermittelt werden soll Name der Farbzusordnungstabelle Name des Datentabellenfeldes, dessen Index ermittelt werden soll Name der Farbzusordnungstabelle
Rückgabewert	String	Zurückgegebener Index des Datentabellenfeldes

Code-Beispiel

```
'Find the index of a DataTableField
Dim fieldIndex As Integer

fieldIndex = VcGantt1.DetectFieldIndex("Maindata", "Name")
```

DumpConfiguration

Methode von VcGantt

Mit dieser Methode können Sie die Konfiguration, bestehend aus .INI und .IFD-Datei, speichern.

Die Methode sollte lediglich zu Diagnosezwecken genutzt werden.

	Datentyp	Beschreibung
Parameter: ⇒ FileName ⇒ encoding	String Mögliche Werte: EncodingEnum Mögliche Werte: vcANSIEncoding 1 vcUnicodeEncoding 2	Dateiname, ggf. mit Pfad. Name der Farbzusordnungstabelle Art der Kodierung Wird eine Datei in der ANSI-Kodierung gespeichert, so geschieht dies in Abhängigkeit von den lokalen Einstellungen des Windows-Betriebssystems, d.h. die Datei enthält Zeichen, die nur in der aktuell eingestellten Sprachversion auch wieder korrekt eingelesen werden können. Wird eine Datei in Unicode-Kodierung gespeichert, ist sie unabhängig von irgendwelchen Einstellungen. Dies Verfahren sollte, wenn möglich, bevorzugt werden. Eine in Unicode-Kodierung gespeicherte Datei erfordert jedoch in Visual Basic 6 eine spezielle Behandlung, wenn sie dort unabhängig von der VARCHART Komponente eingelesen werden soll.
Rückgabewert	Boolean	Datei erfolgreich (True)/nicht erfolgreich (False) abgespeichert.

EditGroup

Methode von VcGantt

Mit dieser Methode wird der Dialog **Gruppen bearbeiten** für die angegebene Gruppe aufgerufen.

	Datentyp	Beschreibung
Parameter: ⇒ group	VcGroup	Gruppe, deren Daten editiert werden sollen
Rückgabewert	Boolean	Gruppendaten wurden editiert/Editierung abgebrochen

EditLink

Methode von VcGantt

Mit dieser Methode wird der Dialog **Verbindung bearbeiten** für die angegebene Verbindung aufgerufen.

	Datentyp	Beschreibung
Parameter: ⇒ link	VcLink	Verbindung, deren Daten editiert werden sollen
Rückgabewert	Boolean	Verbindungsdaten wurden editiert/Editierung abgebrochen

Code-Beispiel

```
Private Sub VcGantt1_OnLinkLClickCltn(ByVal linkCltn As _
                                   VcGanttLib.VcLinkCollection, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)

    Dim link As VcLink

    If linkCltn.Count > 0 Then
        For Each link In linkCltn
            VcGantt1.EditLink link
        Next link
    End If

End Sub
```

EditNode

Methode von VcGantt

Mit dieser Methode wird der Dialog **Vorgänge bearbeiten** für den angegebenen Knoten aufgerufen.

	Datentyp	Beschreibung
Parameter: ⇒ node	VcNode	Knoten, dessen Daten editiert werden sollen
Rückgabewert	Boolean	Knotendaten wurden editiert/Editierung abgebrochen

Code-Beispiel

```
Private Sub VcGantt1_OnNodeLClick(ByVal node As VcGanttLib.VcNode, _
                                   ByVal location As VcGanttLib.LocationEnum, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)

    VcGantt1.EditNode node

End Sub
```

EndLoading

Methode von VcGantt

Mit dieser Methode wird das Ende des Ladevorgangs bei **InsertNodeRecord** und **InsertLinkRecord** angezeigt. Dadurch wird eine Aktualisierung der Grafik ausgelöst.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ladevorgang beendet/nicht beendet

Code-Beispiel

```
VcGantt1.EndLoading
```

ExportGraphicsToFile

Methode von VcGantt

Mit dieser Methode können Sie ein Gantt-Diagramm in einer Datei abspeichern, ohne einen **Speichern unter**-Dialog zu erzeugen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. mit eingebautem EMF)

Nur EMF, EMF+ VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

Beim Exportieren in Bitmapgrafikformate kann durch Angabe einer 0 bei der gewünschten Pixelzahl in X- oder Y-Richtung eine verzerrungsfreie Grafik exportiert werden. Sind beide Pixelanzahlen 0, dann wird die Größe der exportierten Grafik in Pixeln von VARCHART XGantt wie folgt berechnet:

- **PNG:** Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Die DPI-Zahl wird auch in der ausgegebenen PNG-Datei abgelegt, so dass Anzeigeprogramme die richtige Anzeigegröße bei gegebenem Zoomfaktor ermitteln können.
- **GIF, TIFF, BMP, JPEG:** Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.

Bei den Vektorgrafikformaten kann keine Pixelanzahl vorgegeben werden, sondern es werden folgende Koordinatenräume benutzt:

- **WMF:** Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.
- **EMF/EMF+:** Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand in beiden Richtungen X und Y verwendet.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten finden Sie im Kapitel: "Wichtige Konzepte: Grafikformate".

	Datentyp	Beschreibung
Parameter: ⇒ FileName	String	Dateiname, ggf. mit Pfad.
	Mögliche Werte:	

⇒ PrintOutputFormat	PrintOutputFormat	Name der Farbzuoordnungstabelle
	Mögliche Werte: vcBMP 2 vcEMF 9 vcEMFPlus 12 vcEMFWithEMFPlusIncluded 11 vcEPS 3 vcGIF 4 vcJPG 5 vcPCX 6 vcPNG 7 vcTIF 8 vcVMF 0 vcWMF 1 vcWMFWithEMFIncluded 10	Format der abzuspeichernden Datei Datei wird im Format BMP geschrieben. Datei wird im Format EMF geschrieben. Datei wird als *.EMF-Datei geschrieben, beinhaltet aber nur EMF+-Format. Datei wird als *.EMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF+-Format Wird nicht mehr unterstützt. Datei wird im Format GIF geschrieben. Datei wird im Format JPG geschrieben. Wird nicht mehr unterstützt. Datei wird im Format PNG geschrieben. Datei wird im Format TIF geschrieben. Datei wird im Format VMF geschrieben. Datei wird im Format WMF geschrieben. Datei wird als *.WMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF-Format.
⇒ SizeX	Integer	Breite des exportierten Diagramms in Pixeln. Nur bei Pixelformaten möglich. Bei Angabe von 0 wird der Wert unter Beachtung des Seitenverhältnisses berechnet.
⇒ SizeY	Integer	Datenfeldindex
	Mögliche Werte:	Höhe des exportierten Diagramms in Pixeln. Nur bei Pixelformaten möglich. Bei Angabe von 0 wird der Wert unter Beachtung des Seitenverhältnisses berechnet. Datenfeldindex
Rückgabewert	Boolean	Datei erfolgreich (True) / nicht erfolgreich (False) abgespeichert.

Code-Beispiel

```
VcGantt1.ExportGraphicsToFile"C:\temp\export", vcVMF, 0, 0
```

FitChartIntoView**Methode von VcGantt**

Mit dieser Methode haben Sie die Möglichkeit, das Diagramm unter Beibehaltung des Seitenverhältnisses so in die Größe des Steuerelementes einzupassen, dass es entweder in der Höhe oder der Breite komplett sichtbar ist. Es wird die relative Vergrößerung bzw. Verkleinerung in Prozent * 1000 zurückgegeben.

Siehe auch die Eigenschaft **ZoomFactor** und die Methode **Zoom()** von VcGantt.

	Datentyp	Beschreibung
Parameter: ⇒ fitMode	FitModeEnum Mögliche Werte: vcFitHeight 23 vcFitMaximumOfWidthAnd Height 1051 vcFitMinimumOfWidthAnd Height 1052 vcFitWidth 24 vcUseLargerZoomFactor 1053 vcUseSmallerZoomFactor 1054	Auswahl des Zoomfaktors Das Diagramm wird in der Höhe an die Fenstergröße angepasst. Das Diagramm wird mit seiner größten Ausdehnung an das Fenster angepasst. Das Diagramm wird mit seiner kleinsten Ausdehnung an das Fenster angepasst. Das Diagramm wird in der Breite an die Fenstergröße angepasst. Der größere der beiden Zoomfaktoren wird verwendet. Damit ist das Diagramm in dieser entsprechenden Ausdehnung immer größer als das Fenster. Der kleinere der beiden Zoomfaktoren wird verwendet. Damit passt das Diagramm mit der entsprechenden Ausdehnung immer komplett in das Fenster.
Rückgabewert	Long	Relativer Zoomfaktor

Code-Beispiel

```
Dim myZoomfactor As Integer
VcGantt1.(FitChartIntoView(VcFitMode.vcFitWidth) / 1000)
```

FitHistogramsIntoView**Methode von VcGantt**

Mit dieser Methode werden alle sichtbaren Histogramm so skaliert, dass sie gemeinsam in einem Fenster sichtbar sind. Die Skalierung erfolgt proportional; ihre Größenverhältnisse werden also beibehalten.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Histogramme mussten (True) / mussten nicht (False) neu skaliert werden.

Code-Beispiel

```
VcGantt1.FitHistogramsIntoView = True
```

FitRangeIntoView**Methode von VcGantt**

Mit dieser Methode wird ein bestimmter Teilbereich der gesamten Zeitskala in das Diagramm eingepaßt und damit sichtbar gemacht. Die Skalierung der Zeitskala wird dabei entsprechend verändert. Mit dem Parameter **gap-AsNoOfTimeUnits** können Sie festlegen, um wie viele Zeiteinheiten der

sichtbare Bereich früher beginnen soll als der durch den Parameter **startValue** festgelegte Anfang des Bereiches und später enden soll als das durch den Parameter **endValue** festgelegte Ende des Bereiches. Die Zeiteinheit selbst wird auf der Eigenschaftenseite **Allgemeines** eingestellt.

	Datentyp	Beschreibung
Parameter:		
⇒ startDate	Date/Time	Anfangsdatum des einzupassenden Bereiches.
⇒ endDate	Date/Time	Enddatum des einzupassenden Bereiches.
⇒ gapAsNoOfTimeUnits	Long	Anzahl der Zeiteinheiten zwischen startDate und dem Anfang des sichtbaren Bereiches bzw. zwischen endDate und dem Ende des sichtbaren Bereiches
Rückgabewert	Boolean	Bereich wurde erfolgreich (True) / nicht erfolgreich (False) eingepasst.

Code-Beispiel

```
VcGantt1.FitRangeIntoView "14.09.14", "21.09.14", 1
```

GetAValueFromARGB

Methode von VcGantt

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Alpha-Wert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter:		
⇒ argb	Long	ARGB- Wert, aus dem der Alpha-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Alpha-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcGantt1.MakeARGB(alpha, red, green, blue)
alpha = VcGantt1.GetAValueFromARGB(argb)
```


GetBValueFromARGB

Methode von VcGantt

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Blauwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter: ⇨ argb	Long	ARGB- Wert, aus dem der Blau-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Blau-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcGantt1.MakeARGB(alpha, red, green, blue)
blue = VcGantt1.GetBValueFromARGB(argb)
```

GetCurrentComponentStart

Methode von VcGantt

Diese Methode liefert den aktuellen Scrollwert (die Startkoordinate, auf die gerade gescrollt ist) eines grafischen Grundelements des VARCHART- Windows-Forms-Steuerelements (Zeitskala, Diagramm, Histogramm, Tabelle, Tabellenüberschrift usw.) in beliebiger Richtung in 1/100 mm.

	Datentyp	Beschreibung
Parameter: ⇨ component	ComponentTypeEnum Mögliche Werte: vcAdditionalListComponent 1 vcBottomListTitleComponent 14 vcBottomRightListTitleComponent 17 vcBottomTimeScaleComponent 15 vcDiagramComponent 4 vcHistogramComponent 8 vcHistogramVerScaleComponent 7	Typ des grafischen Elements zusätzliche Tabelle untere Titelleiste Tabellenbereich unten rechts untere Zeitskala Diagramm Histogramm numerische Skala (vertikale Histogramm-skala)

	vcLegendComponent 10 vcListComponent 0 vcListTitleComponent 2 vcRightListComponent 5 vcRightListTitleComponent 16 vcTimeScaleComponent 3 vcTopTitleComponent 11	Legende (zur Zeit ohne Funktion; Rückgabewerte 00) Tabelle Tabellenüberschrift rechte Tabelle Tabellenüberschrift der rechten Tabelle obere Zeitskala obere Titelleiste
↔ scrollOrientation	ScrollOrientationEnum Mögliche Werte: vcHorizontal 1 vcVertical 2	Richtung, in die gescrollt wird horizontales Scrollen vertikales Scrollen
Rückgabewert	Long	Scrollwert in 1/100 mm

GetCurrentViewDates

Methode von VcGantt

Mit dieser Methode können Sie das Start- oder Enddatum des sichtbaren Bereichs der Zeitskala erfragen.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetCurrentViewDatesAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
↔ leftDate	Date	Startdatum des sichtbaren Bereichs der Zeitskala
↔ rightDate	Date	Enddatum des sichtbaren Bereichs der Zeitskala
Rückgabewert	Boolean	Start- bzw. Enddatum des sichtbaren Bereichs der Zeitskala werden zurückgegeben/nicht zurückgegeben.

Code-Beispiel

```

Dim bGetCurrentViewDates As Boolean
Dim leftdate As Date
Dim rightdate As Date

GetCurrentViewDates = VcGantt1.GetCurrentViewDates(leftdate, rightdate)

```

GetCurrentViewDatesAsString

Methode von VcGantt

Mit dieser Methode können Sie das Start- oder Enddatum des sichtbaren Bereichs der Zeitskala erfragen. Diese Methode ist identisch mit der Methode

GetCurrentViewDates, abgesehen davon, dass die Parameter hier als String übergeben werden.

Der Datumsstring hat das feste Format "DD.MM.YYYY;hh:mm:ss".

	Datentyp	Beschreibung
Parameter: ↔ leftDate	String	Startdatum des sichtbaren Bereichs der Zeitskala
	Mögliche Werte:	Name der Farbzuoordnungstabelle
↔ rightDate	String	Enddatum des sichtbaren Bereichs der Zeitskala
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	Start- bzw. Enddatum des sichtbaren Bereichs der Zeitskala werden zurückgegeben/nicht zurückgegeben.

Code-Beispiel

```
Dim bGetCurrentViewDates As Boolean
Dim leftdate As String
Dim rightdate As String

bGetCurrentViewDates = VcGantt1.GetCurrentViewDates(leftdate, rightdate)
```

GetCurrentViewDatesAsVariant

Methode von VcGantt

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetCurrentViewDates**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↔) nur verwenden kann, wenn diese Parameter vom Typ Variant sind.

Code-Beispiel

```
Dim bGetCurrentViewDates As Boolean
Dim leftdate As Variant
Dim rightdate As Variant

bGetCurrentViewDates = VcGantt1.GetCurrentViewDates(leftdate, rightdate)
```

GetDate

Methode von VcGantt

Mit dieser Methode können Sie innerhalb des Diagrammbereichs das Datum zu einer beliebigen X-Koordinate erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ x	Long	X-Koordinate im Gantt-Diagramm, zu der das Datum erfragt werden soll
Rückgabewert	Date/Time	Erfragtes Datum

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramLClick(ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    Labell1.Caption = VcGantt1.GetDate(x)

End Sub
```

GetDateAsString**Methode von VcGantt**

Mit dieser Methode können Sie innerhalb des Diagrammbereichs das Datum zu einer beliebigen X-Koordinate erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ x	Long	X-Koordinate im Gantt-Diagramm, zu der das Datum erfragt werden soll
Rückgabewert	String	Erfragtes Datum

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramLClick(ByVal x As Long, ByVal y As Long,
    returnStatus As Variant)
    MsgBox VcGantt1.GetDateAsString(x)
End Sub
```

GetValueFromARGB**Methode von VcGantt**

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Grünwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter: ⇒ argb	Long	ARGB- Wert, aus dem der Grün-Wert ermittelt werden soll

Rückgabewert	Integer	Zurückgegebener Grün-Wert
---------------------	---------	---------------------------

Code-Beispiel

```

Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcGantt1.MakeARGB(alpha, red, green, blue)
green = VcGantt1.GetGValueFromARGB(argb)

```

GetLinkByID**Methode von VcGantt**

Mit dieser Methode können Sie auf eine einzelne Verbindung über ihre Identifikation zugreifen, die im Dialog **Datentabellen verwalten** festgelegt wurde. Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter:		
⇒ linkID	Variant	Identifikation der Verbindung
Rückgabewert	VcLink	Verbindung

Code-Beispiel

```

Dim link1 As VcLink
Dim successor As Integer

Set link1 = VcGantt1.GetLinkByID(" 1")
successor = link1.datafield(2)

```

GetLinkByIDs**Methode von VcGantt**

Mit dieser Methode können Sie auf eine einzelne Verbindung über die IDs ihres Vorgänger- und ihres Nachfolgerknotens zugreifen. Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter: ⇒ predecessorID Mögliche Werte: ⇒ successorID	String Mögliche Werte: String Mögliche Werte:	Identifikation des Vorgängerknotens Name der Farbzusordnungstabelle Identifikation des Nachfolgerknotens Name der Farbzusordnungstabelle
Rückgabewert	VcLink	Verbindung

Code-Beispiel

```
Dim link As VcLink

Set link = VcGantt1.GetLinkByIds(" 2", " 3")
```

GetNodeById**Methode von VcGantt**

Mit dieser Methode können Sie auf einen einzelnen Knoten über seine Identifikation zugreifen, die im Dialog **Datentabellen verwalten** festgelegt wurde. Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter: ⇒ nodeID	Variant	Identifikation des Knotens
Rückgabewert	VcNode	Knoten

Code-Beispiel

```
Dim node As VcNode

Set node = VcGantt1.GetNodeById("10")
```

GetRValueFromARGB

Methode von VcGantt

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Rotwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter: ⇒ argb	Long	ARGB- Wert, aus dem der Rot-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Rot-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcGantt1.MakeARGB(alpha, red, green, blue)
red = VcGantt1.GetRValueFromARGB(argb)
```

GetViewComponentSize

Methode von VcGantt

Mit dieser Methode können Sie zur Laufzeit die Größe und Position eines grafischen Grundelements des VARCHART-ActiveX-Steuerelements (Zeitskala, Diagramm, Histogramm, Tabelle, Tabellenüberschrift usw.) erfragen (siehe Ereignis **OnViewComponentsSizeModifyComplete**).

Hinweise:

1. Die Position bezieht sich auf den Nullpunkt des VARCHART-XGantt-Steuerelements.
2. Die Werte werden in Pixel zurückgegeben.
3. Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetViewComponentSizeAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter: ⇒ viewComponent	ComponentTypeEnum Mögliche Werte: vcAdditionalListComponent 1 vcBottomListTitleComponent 14 vcBottomRightListTitleComponent 17 vcBottomTimeScaleComponent 15 vcDiagramComponent 4 vcHistogramComponent 8 vcHistogramVerScaleComponent 7 vcLegendComponent 10 vcListComponent 0 vcListTitleComponent 2 vcRightListComponent 5 vcRightListTitleComponent 16 vcTimeScaleComponent 3 vcTopTitleComponent 11	Komponententyp zusätzliche Tabelle untere Titelleiste Tabellenbereich unten rechts untere Zeitskala Diagramm Histogramm numerische Skala (vertikale Histogramm-skala) Legende (zur Zeit ohne Funktion; Rückgabewerte 00) Tabelle Tabellenüberschrift rechte Tabelle Tabellenüberschrift der rechten Tabelle obere Zeitskala obere Titelleiste
⇨ x	Long	X-Koordinate der Komponente
⇨ y	Long	Y-Koordinate der Komponente
⇨ width	Long	Breite der Komponente
⇨ height	Long	Höhe der Komponente
Rückgabewert	Void	

Code-Beispiel

```

Private Sub handleHideHistogram()
    Dim x As Long
    Dim y As Long
    Dim width As Long
    Dim height As Long
    Dim scMod As Long
    scMod = ScaleMode
    ScaleMode = vbPixels

    VcGantt1.GetViewComponentSize vcHistogramVerScaleComponent, x, y, _
        width, height

    ' plus 6 because of the sash
    Text1.Top = VcGantt1.Top + y + 6
    Text1.Left = VcGantt1.Left + x
    ' minus 25 because of the numeric scale
    Text1.width = width - 25
    ' minus 6 because of the sash
    Text1.height = height - 6
    ScaleMode = scMod
End Sub

```

GetViewComponentSizeAsVariant

Methode von VcGantt

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetViewComponentSize**. Die gesonderte Implementierung wurde notwendig, weil

beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch $\Leftrightarrow$) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GroupNodes

Methode von VcGantt

Mit dieser Methode können Sie die Gruppierung an- oder abschalten. Wenn Sie mit der Eigenschaft **GroupingField** ein Gruppierfeld und/oder mit der Eigenschaft **GroupingOrderField** eine Gruppierreihenfolge eingestellt haben, so müssen Sie danach mit **GroupNodes** die Gruppierung aktivieren.

	Datentyp	Beschreibung
Parameter: $\Rightarrow$ onOff	Boolean Mögliche Werte:	Gruppierung ein/aus Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
Rückgabewert	Boolean	Knoten erfolgreich (True) / nicht erfolgreich (False) gruppiert

Code-Beispiel

```
VcGantt1.GroupingField = 11
VcGantt1.GroupingOrderField = 12

VcGantt1.GroupNodes (True)
```

HistogramSetMaxYValue

Methode von VcGantt

Mit dieser Methode können Sie den Maximalwert der numerischen Skala des (ersten) Histogramms festlegen. Diesen Wert können Sie auch im Dialog **Histogramme verwalten** unter **Endwert** setzen.

	Datentyp	Beschreibung
Parameter: $\Rightarrow$ yValue	Long	Maximaler Y-Wert
Rückgabewert	Long	Maximaler Y-Wert gesetzt (1)/nicht gesetzt (0)

Code-Beispiel

```
VcGantt1.HistogramSetMaxYValue (40)
```

IdentifyField

Methode von VcGantt

Mit dieser Methode können Sie das Tabellenfeld eines Knotens identifizieren, das sich an einer gegebenen Cursorposition befindet.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ IdentifiedObjectType	Integer	Identifizierter Objekttyp, für zukünftige Verwendung
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcField	Identifiziertes Feld

Code-Beispiel

```
Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
                                ByVal location As VcGanttLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    Dim field As VcField
    Dim objtype As Long

    Set field = VcGantt1.IdentifyField(x, y, objtype)
    If Not field Is Nothing then
        Labell1.Caption = node.dataField(field.DataFieldID)
    End If
End Sub
```

IdentifyLayerAt

Methode von VcGantt

Mit dieser Methode können Sie einen Layer identifizieren. Wenn Sie mit der Methode **IdentifyObjectAt** einen Knoten identifiziert haben, können Sie ihn als Referenzobjekt benutzen, um mit **IdentifyLayerAt** den Layer dieses Knotens zu erfragen, der auf den angegebenen Koordinaten liegt.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der by-Reference-Parameter nur die analoge Methode **IdentifyLayerAtAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

⇒ referenceNode	VcNode	Bezugsknoten
⇐ identifiedLayer	VcLayer	Identifizierter Layer
Rückgabewert	Boolean	Objekt identifiziert/Objekt nicht identifiziert

Code-Beispiel

```
Private Sub VcGantt1_DragDrop(Source As Control, X As Single, Y As Single)

    Dim identifiedObj As Object
    Dim identifiedObjType As VcObjectTypeEnum
    Dim identifiedLayer As VcLayer
    Dim xPix, yPix As Long

    xPix = X / Screen.TwipsPerPixelX
    yPix = Y / Screen.TwipsPerPixelY

    Call VcGantt1.IdentifyObjectAt(xPix, yPix, identifiedObj, identifiedObjType)

    Select Case identifiedObjType
        Case vcObjTypeNodeInDiagram
            Call VcGantt1.IdentifyLayerAt(xPix, yPix, identifiedObj, _
                identifiedLayer)
            If Not identifiedLayer Is Nothing Then
                MsgBox ("The Node """" + identifiedObj.DataField(0) + _
                    """, Layer """" + identifiedLayer.Name + _
                    """, was identified in the diagram area.")
            Else
                MsgBox ("The Node """" + identifiedObj.DataField(0) + _
                    """, was identified in diagram area; " + _
                    "no layer was identified.")
            End If
        Case vcObjTypeNodeInTable
            MsgBox ("The Node """" + identifiedObj.DataField(0) + _
                """, was identified via the table.")
        Case Else
            MsgBox ("No node was identified.")
    End Select
End Sub
```

IdentifyLayerAtAsVariant**Methode von VcGantt**

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyLayerAt**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇐) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

IdentifyObject

Methode von VcGantt

Mit dieser Methode können Sie ein Objekt identifizieren, das sich im Tabellen- oder Diagrammbereich befindet. Der Typ des Objekts wird zurückgegeben. Wenn Sie mit dieser Methode einen Knoten identifiziert haben, können Sie ihn als Referenzobjekt benutzen, um mit einem weiteren Aufruf von **IdentifyObject** den Layer dieses Knotens an derselben Position zu erfragen.

Sollten Sie mit einer Entwicklungsumgebung arbeiten, die immer eine Referenz auf ein Objekt erfordert, benutzen Sie bitte die Methode **IdentifyObjectAt**, weil dort der Parameter **referenceObject** nicht erforderlich ist.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇒ referenceObject	Object	Bezugsobjekt, auf das sich die Identifizierung bezieht
⇐ identifiedObject	Object	Erkanntes Objekt
⇐ identifiedObjectType	VcObjectTypeEnum	Typ des erkannten Objekts
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeCalendarGrid 18 vcObjTypeCurve 12 vcObjTypeDateLine 9 vcObjTypeGroup 7 vcObjTypeGroupInDiagram 11 vcObjTypeGroupInTable 7 vcObjTypeHistogram 13 vcObjTypeLayer 8 vcObjTypeLinkCollection 3 vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14 vcObjTypeTable 4 vcObjTypeTableCaption 5 vcObjTypeTimeScale 6	Objekttyp Box Objekttyp Kalendergitter Objekttyp Kurve Objekttyp Stichtaglinie Objekttyp Gruppe Objekttyp Gruppe im Knotenbereich Objekttyp Gruppe im Tabellenbereich Objekttyp Histogramm Objekttyp Layer Objekttyp LinkCollection Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Knoten im Tabellenbereich kein Objekt Objekttyp Werteskala Objekttyp Summenbalken Objekttyp Tabelle Objekttyp Tabellenüberschrift Objekttyp Zeitskala
Rückgabewert	Boolean	Objekt identifiziert/Objekt nicht identifiziert

Code-Beispiel

```
Private Sub VcGantt1_DragDrop(Source As Control, X As Single, Y As Single)
    Dim label As Label
```

832 API-Referenz: VcGantt

```
Dim identifiedObj As Object
Dim referenceObj As Object
Dim identifiedObjType As Long
Dim xPix, yPix As Long
Dim colorValue As String

xPix = X / Screen.TwipsPerPixelX
yPix = Y / Screen.TwipsPerPixelY

Set referenceObj = Nothing

Call VcGantt1.IdentifyObject(xPix, yPix, referenceObj, identifiedObj, _
    identifiedObjType)

Select Case identifiedObjType
    Case vcObjTypeNodeInDiagram
        Dim identifiedLayer As Object
        Dim identifiedSubObjType As Long
        Call VcGantt1.IdentifyObject(xPix, yPix, identifiedObj, _
            identifiedLayer, identifiedSubObjType)
        If identifiedSubObjType = VcGanttLib.VcObjectTypeEnum. _
            vcObjTypeLayer Then
            Dim node As VcNode
            Dim layer As VcLayer
            Set node = identifiedObj

            Set layer = identifiedLayer
            MsgBox ("The Node " + node.DataField(0) + "; Layer " + _
                layer.Name + " was identified in diagram
area")
        Else
            MsgBox ("The Node " + identifiedObj.DataField(0) + " was _
                identified in diagram area; no layer
identified")
        End If
    Case vcObjTypeNodeInTable
        MsgBox ("The Node " + identifiedObj.DataField(0) + " was _
            identified via the table")
    Case vcObjTypeGroup
        Dim barGroup As VcGroup
        Set barGroup = identifiedObj
        MsgBox ("The Group " + barGroup.Name + " was identified")
    Case vcObjTypeLinkCollection
        Dim linkCltn As VcLinkCollection
        Dim link As VcLink
        Set linkCltn = identifiedObj
        Set link = linkCltn.FirstLink
        While Not link Is Nothing
            MsgBox ("The Link " + link.AllData + " was identified")
            Set link = linkCltn.NextLink
        Wend
    Case vcObjTypeTable
        MsgBox ("The Table was hit")
    Case vcObjTypeTableCaption
        MsgBox ("The TableCaption was hit")
    Case vcObjTypeTimeScale
        MsgBox ("The Timescale " + identifiedObj.Name + " was identified")
    Case Else
        MsgBox ("No object identified.")
End Select
End Sub
```

IdentifyObjectAt

Methode von VcGantt

Mit dieser Methode können Sie ein beliebiges Objekt in VARCHART XGantt identifizieren. Der Typ des Objekts wird zurückgegeben. Wenn Sie mit dieser Methode einen Knoten identifiziert haben, können Sie ihn als Referenzobjekt benutzen, um mit einem Aufruf von **IdentifyLayerAt** den Layer dieses Knotens an derselben Position zu erfragen. Zum Identifizieren einer Kurve im Histogramm muss die Methode **IdentifyObject** verwendet werden.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **IdentifyObjectAtAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇐ identifiedObject	Object	Erkanntes Objekt
⇐ identifiedObjectType	VcObjectTypeEnum	Typ des erkannten Objekts
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeCalendarGrid 18 vcObjTypeCurve 12 vcObjTypeDateLine 9 vcObjTypeGroup 7 vcObjTypeGroupInDiagram 11 vcObjTypeGroupInTable 7 vcObjTypeHistogram 13 vcObjTypeLayer 8 vcObjTypeLinkCollection 3 vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14 vcObjTypeTable 4 vcObjTypeTableCaption 5 vcObjTypeTimeScale 6	Objekttyp Box Objekttyp Kalendergitter Objekttyp Kurve Objekttyp Stichtaglinie Objekttyp Gruppe Objekttyp Gruppe im Knotenbereich Objekttyp Gruppe im Tabellenbereich Objekttyp Histogramm Objekttyp Layer Objekttyp LinkCollection Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendsbereich Objekttyp Knoten im Tabellenbereich kein Objekt Objekttyp Werteskala Objekttyp Summenbalken Objekttyp Tabelle Objekttyp Tabellenüberschrift Objekttyp Zeitskala
Rückgabewert	Boolean	Objekt identifiziert/Objekt nicht identifiziert

Code-Beispiel

```
Private Sub VcGantt1_OnMouseMove(ByVal button As Integer, ByVal Shift As Integer, ByVal x As Long, ByVal y As Long)
```

```
    Dim identifiedObject As Object
    Dim identifiedObjectType As VcObjectTypeEnum
```

```

Dim node As VcNode

Call VcGantt1.IdentifyObjectAt(x, y, identifiedObject, identifiedObjectType)

Select Case identifiedObjectType
    Case VcObjectTypeEnum.vcObjTypeNodeInDiagram,
VcObjectTypeEnum.vcObjTypeNodeInTable
        Set node = identifiedObject
        Label1.Caption = node.DataField(1)
    Case Else
        Label1.Caption = ""
End Select

End Sub

```

IdentifyObjectAtAsVariant

Methode von VcGantt

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyObjectAt**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↩) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

InsertLinkRecord

Methode von VcGantt

Mit dieser Methode werden die Daten einer Verbindung zwischen zwei Knoten geladen. Die Daten werden als CSV-String oder als Datenfeld gemäß der im Dialog **Datentabellen verwalten** in der **Relations**-Tabelle vereinbarten Struktur übergeben. Die Methode **EndLoading** sollte am Ende des kompletten Ladevorgangs (Verbindungen und Knoten) einmal aufgerufen werden.

	Datentyp	Beschreibung
Parameter: ⇒ linkRecordContent	Object	Inhalt des Verbindungsdatensatzes
Rückgabewert	VcLink	Verbindung

Code-Beispiel

```

VcGantt1.InsertNodeRecord ("A100;Activity 1;12.09.14;17.09.14;5;Planning")
VcGantt1.InsertNodeRecord ("A105;Activity 5;13.09.14;18.09.14;7;Testing")

VcGantt1.InsertLinkRecord ("A100;A105;FS;0")

VcGantt1.EndLoading

' or:

```

```

Dim linkRecord As Variant

linkRecord = Array("A100", "A105", "FS", 0)

VcGantt1.InsertNodeRecord ("A100;Activity 1;12.09.14;17.09.14;5;Planning")
VcGantt1.InsertNodeRecord ("A105;Activity 5;13.09.14;18.09.14;7;Testing")

VcGantt1.InsertLinkRecord (linkRecord)

VcGantt1.EndLoading

```

InsertNodeRecord

Methode von VcGantt

Mit dieser Methode werden die Daten eines Knotens geladen. Die Daten werden als CSV-String oder Datenfeld gemäß der im Dialog **Datentabellen verwalten** in der **Maindata**-Tabelle vereinbarten Struktur übergeben. Die Methode **EndLoading** sollte am Ende des kompletten Ladevorgangs (Verbindungen und Knoten) einmal aufgerufen werden.

	Datentyp	Beschreibung
Parameter: ⇒ nodeRecordContent	Data field	Inhalt des Knotendatensatzes
Rückgabewert	VcNode	Knoten

Code-Beispiel

```

Dim nodeRecord As String
' data format: "Number;Name;Start date;Finish date;Group code;Group name"
nodeRecord = "A100;Activity 1;12.09.14;17.09.14;5;Planning"
VcGantt1.InsertNodeRecord (nodeRecord)

VcGantt1.EndLoading

' or

Dim nodeRecord As Variant

nodeRecord = Array("A100", "Activity 1", "12.09.14", "17.09.14", 5, "Planning")

VcGantt1.InsertNodeRecord (nodeRecord)

VcGantt1.EndLoading

```

MakeARGB

Methode von VcGantt

Mit dieser Methode können Sie aus den vier Einzelwerten einer Farbe einen ARGB-Wert bilden.

	Datentyp	Beschreibung
Parameter: ⇒ alpha	Integer	Alpha-Wert
	Mögliche Werte:	Datenfeldindex
	⇒ red	Rot-Wert
	Mögliche Werte:	Datenfeldindex
⇒ green	Integer	Grün-Wert
	Mögliche Werte:	Datenfeldindex
⇒ blue	Integer	Blau-Wert
	Mögliche Werte:	Datenfeldindex
Rückgabewert	Long	Zurückgegebener ARGB-Wert

Code-Beispiel

```

Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = FF
red = A0
green = 34
blue = AB
argb = VcGantt1.MakeARGB(alpha, red, green, blue)

```

Open**Methode von VcGantt**

Mit dieser Methode werden aus der angegebenen Datei die Datensätze der Datentabellen eingelesen, die zu einem früheren Zeitpunkt mit der Methode SaveAsEx (...) im CSV-Format gespeichert wurden. CSV-Dateien können sowohl im ANSI- als auch in Unicode-Zeichensatz gelesen und geschrieben werden. Beim Einlesen erfolgt eine automatische Erkennung der Codierung.

Die Zuordnung der Datensätze zu den einzelnen Datentabellen wird jeweils durch eine entsprechende Identifizierungszeile sichergestellt.

```
**** Name der Tabelle ****
```

Beispiel:

```
**** Maindata ****
```

```

1;Node 1;07.05.2007;;5
2;Node 2;14.05.2007;;5
3;Node 3;21.05.2007;;5
**** Relations ****
1;1;2
2;2;3

```

Alle Datensätze von Tabellen, die nicht existieren, werden beim Einlesen ignoriert.

Der Inhalt der Datentabellen wird vollständig ersetzt.

	Datentyp	Beschreibung
Parameter: ⇒ fileName	String Mögliche Werte:	Name der zu öffnenden Datei Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Datei erfolgreich (True) / nicht erfolgreich (False) geöffnet

Code-Beispiel

```
VcGantt1.Open "C:\Data\project1.csv"
```

OptimizeTimeScaleStartEnd

Methode von VcGantt

Mit dieser Methode werden Anfang und Ende der Zeitskala in Abhängigkeit von den dargestellten Vorgängen so gesetzt, dass alle Vorgänge zwischen Anfang und Ende der Zeitskala liegen. Mit dem Parameter **NoOfUnits** können Sie festlegen, um wie viele Zeiteinheiten die Skala früher beginnen soll als der früheste Anfang aller Vorgänge und später enden soll als das späteste Ende aller Vorgänge. Die Zeiteinheit selbst wird auf der Eigenschaftenseite **Allgemeines** eingestellt.

	Datentyp	Beschreibung
Parameter: ⇒ noOfUnits	Integer Mögliche Werte:	Anzahl der Zeiteinheiten Datenfeldindex

Rückgabewert	Boolean	Zeitskala erfolgreich (True) / nicht erfolgreich (False) optimiert. Der Rückgabewert ist false, wenn sowohl TimeScaleStart als auch TimeScaleEnd nicht verändert wurden. Wenn keine Vorgänge vorliegen, ist der Rückgabewert stets false, weil keine Datumsänderungen stattfinden. Die angegebene Anzahl der Zeiteinheiten ist dabei ohne Bedeutung.
---------------------	---------	---

Code-Beispiel

```
VcGantt1.OptimizeTimeScaleStartEnd (5)
```

PageLayout

Methode von VcGantt

Mit dieser Methode wird das Dialogfeld **Seite einrichten** aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Dialogfeld erfolgreich (True) / nicht erfolgreich (False) geöffnet

Code-Beispiel

```
VcGantt1.PageLayout
```

PrintDirectEx

Methode von VcGantt

Mit dieser Methode können Sie das Diagramm direkt ausdrucken, ohne dass zuvor ein Dialogfeld erscheint. Der Rückgabewert soll bei nicht erfolgreichem Druck Aufschluss über die Ursache liefern. Dies kann z.B. auch ein Eintrag in einer Logdatei sein.

	Datentyp	Beschreibung
Rückgabewert	PrintResultStatusEnum	Mögliche Werte: vcPrintingSucceeded 0: Druck verlief erfolgreich vcNoPrinterInstalled 1 Es wurde weder der über VcPrinter.PrinterName angegebene noch ein im Windows-Betriebssystem als Standarddrucker gekennzeichnete Drucker gefunden. vcPrintingAbortedByUser 2: Der Druck wurde durch den Anwender abgebrochen. vcPrintingAbortedByDriver 3: Der Druck wurde durch den Windows-Druckertreiber abgebrochen. vcUnprintablePageLayout 4: Es konnte nicht gedruckt werden, weil das Seitenlayout zusammen mit den Druckereigenschaften wie Papiergröße und Ränder zu einem nicht druckbaren Layout führten.

Code-Beispiel

```
PrintStatusResultEnum status = VcGantt1.PrintDirectEx()
If status <> vcPrintingSucceeded Then
    Debug.Print "Printing failed: " & status & vbCrLf
End If
```

PrinterSetup**Methode von VcGantt**

Mit dieser Methode wird das Windows-Dialogfeld **Drucker einrichten** aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Dialogfeld erfolgreich (True) / nicht erfolgreich (False) aufgerufen

Code-Beispiel

```
VcGantt1.PrinterSetup
```

PrintIt

Methode von VcGantt

Mit dieser Methode wird der Ausdruck der Grafik ausgelöst. Es wird das Windows-Dialogfeld **Drucken** geöffnet, dabei werden die unter **PageLayout** aktuell eingestellten Parameter verwendet.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ausdruck erfolgreich (True) / nicht erfolgreich (False) ausgelöst

Code-Beispiel

```
VcGantt1.PrintIt
```

PrintPreview

Methode von VcGantt

Mit dieser Methode wird die Druckvorschau aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Dialogfeld erfolgreich (True) / nicht erfolgreich (False) aufgerufen

Code-Beispiel

```
VcGantt1.PrintPreview
```

PrintToFile

Methode von VcGantt

Mit dieser Methode können Sie das Diagramm direkt in eine Datei drucken. Ob dies gelingt, hängt vom Druckertreiber ab, da viele PDF-Druckertreiber keine Dateinamen akzeptieren.

	Datentyp	Beschreibung
Parameter: ⇒ fileName	String	Dateiname
	Mögliche Werte:	Name der Farbzustimmungstabelle
Rückgabewert	Void	

Code-Beispiel

```
VcGantt1.PrintToFile
```

RecalculateAllStructureCodes

Methode von VcGantt

Mit dieser Methode können Sie den Struktur-Code der Knotenhierarchie für alle Knoten neu berechnen lassen. Eine Neuberechnung erfolgt automatisch nach jeder Änderung. Um die automatische Neuberechnung für bestimmte Aktionen zu unterdrücken, können Sie Anweisungen zwischen `VcGantt.SuspendUpdate(true)` und `VcGantt.SuspendUpdate(false)` "klammern".

	Datentyp	Beschreibung
Rückgabewert	Void	

Reset

Methode von VcGantt

Mit dieser Methode können Sie - je nach eingestelltem Wert von `resetAction` - Objekte (Knoten, Verbindungen, Kalender etc.) aus dem Diagramm löschen bzw. den zur Entwurfszeit auf den Eigenschaftenseiten eingestellten Zustand wiederherstellen.

	Datentyp	Beschreibung
Parameter: ⇒ <code>resetAction</code>	ResetActionEnum Mögliche Werte: <code>vcEmptyAllDataTables</code> 4 <code>vcReloadConfiguration</code> 2 <code>vcRemoveGroups</code> 0 <code>vcRemoveNodes</code> 1	Objekte, die gelöscht oder neu initialisiert werden Der komplette Inhalt sämtlicher Datentabellen wird gelöscht, die Datentabellen selbst bleiben bestehen. Komplette Neuinitialisierung. Alle Einstellungen und erzeugten Objekte verfallen. Alle Gruppen und abhängigen Objekte und damit auch alle Knoten und Links werden entfernt. Alle Knoten und abhängigen Objekte und damit auch alle Links, sofern vorhanden, werden entfernt.
Rückgabewert	Boolean	Objekte im Diagramm erfolgreich gelöscht (True)

Code-Beispiel

```
VcGantt1.Reset(vcRemoveNodes) = True
```

SaveAsEx

Methode von VcGantt

Mit dieser Methode werden die Datensätze aller Datentabellen im CSV-Format gespeichert. Dabei wird die auf der Eigenschaftenseite **Objekte** unter **Datentabellen** festgelegte Struktur verwendet. Datentabellen, die keine Datensätze enthalten, werden nicht gespeichert. Ist kein Name angegeben, wird die zuletzt bei **Open** angegebene Datei überschrieben (entspricht der üblichen **Save**-Funktion).

	Datentyp	Beschreibung
Parameter: ⇒ fileName ⇒ encoding	String Mögliche Werte:	Name der zu speichernden Datei Name der Farbzordnungstabelle
	EncodingEnum Mögliche Werte: vcANSIEncoding 1 vcUnicodeEncoding 2	Art der Kodierung Wird eine Datei in der ANSI-Kodierung gespeichert, so geschieht dies in Abhängigkeit von den lokalen Einstellungen des Windows-Betriebssystems, d.h. die Datei enthält Zeichen, die nur in der aktuell eingestellten Sprachversion auch wieder korrekt eingelesen werden können. Wird eine Datei in Unicode-Kodierung gespeichert, ist sie unabhängig von irgendwelchen Einstellungen. Dies Verfahren sollte, wenn möglich, bevorzugt werden. Eine in Unicode-Kodierung gespeicherte Datei erfordert jedoch in Visual Basic 6 eine spezielle Behandlung, wenn sie dort unabhängig von der VARCHART Komponente eingelesen werden soll.
Rückgabewert	Boolean	Speicherung war erfolgreich (True)/nicht erfolgreich (False)

Code-Beispiel

```
VcGantt1.SaveAsEx "C:\Data\project1.csv" , vcANSIEncoding
```

Schedule

Methode von VcGantt

Mit dieser Methode können Sie eine Vorwärts- und Rückwärtsberechnung des aktuellen Projekts durchführen. Bei alleiniger Übergabe des Starttermins wird zunächst eine Vorwärtsberechnung, dann eine Rückwärtsberechnung durchgeführt. Bei alleiniger Übergabe des Endtermins wird zunächst eine Rückwärtsberechnung, dann eine Vorwärtsberechnung durchgeführt. Es können auch beide Termine übergeben werden, die Vorgänge erhalten dann entsprechende Pufferzeiten. Das Fehlen beider Daten führt zu einer

Fehlermeldung. Falls ein Zyklus der Knoten und Verbindungen festgestellt wird, werden diese automatisch markiert.

	Datentyp	Beschreibung
Parameter: ⇒ startDate	Date/Time	Startdatum
Rückgabewert	Boolean	Vorwärtsrechnung erfolgreich (True) / nicht erfolgreich (False)

Code-Beispiel

```
VcGantt1.ScheduleProject("21.06.14", "")
```

ScrollComponentStartTo

Methode von VcGantt

Mit dieser Methode können Sie ein grafisches Grundelement des VARCHART-Windows-Forms-Steuerelements (Zeitskala, Diagramm, Histogramm, Tabelle, Tabellenüberschrift usw.) in beliebiger Richtung auf den angegebenen Scrollwert (die Startkoordinate) in 1/100 mm scrollen.

	Datentyp	Beschreibung
Parameter: ⇐ component	ComponentTypeEnum Mögliche Werte: vcAdditionalListComponent 1 vcBottomListComponent 14 vcBottomRightListComponent 17 vcBottomTimeScaleComponent 15 vcDiagramComponent 4 vcHistogramComponent 8 vcHistogramVerScaleComponent 7 vcLegendComponent 10 vcListComponent 0 vcListTitleComponent 2 vcRightListComponent 5 vcRightListTitleComponent 16 vcTimeScaleComponent 3 vcTopTitleComponent 11	Typ des grafischen Elements zusätzliche Tabelle untere Titelleiste Tabellenbereich unten rechts untere Zeitskala Diagramm Histogramm numerische Skala (vertikale Histogramm-skala) Legende (zur Zeit ohne Funktion; Rückgabewerte 00) Tabelle Tabellenüberschrift rechte Tabelle Tabellenüberschrift der rechten Tabelle obere Zeitskala obere Titelleiste
⇐ scrollOrientation	ScrollOrientationEnum Mögliche Werte: vcHorizontal 1 vcVertical 2	Richtung, in die gescrollt wird horizontales Scrollen vertikales Scrollen
Rückgabewert	Boolean	Gewünschter Scrollwert wird/wird nicht zurückgegeben

ScrollToDate

Methode von VcGantt

Mit dieser Methode können Sie in der Zeitskala zu einem bestimmten Datum scrollen. Der Parameter **gapAsNoOfTimeUnits** gibt an, wie viele Zeiteinheiten der Abstand zwischen dem angegebenen Datum und dem linken (**vcLeftAligned**) bzw. rechten (**vcRightAligned**) Rand der Zeitskala betragen soll, wenn Sie durch den Parameter **horAlignment** bestimmt haben, dass das angegebene Datum am linken bzw. rechten Rand des sichtbaren Bereiches der Zeitskala stehen soll. Die Zeiteinheit kann auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Sind Nicht-Arbeitszeiten kollabiert, werden die kollabierten (unsichtbaren) Zeiten zwar korrekt eingerechnet, aber optisch nicht dargestellt, was zu scheinbaren Abweichungen vom eingestellten Ergebnis führen kann.

	Datentyp	Beschreibung
Parameter:		
⇒ date	Date/Time	Datum
⇒ horAlignment	HorizontalAlignmentEnum	horizontale Ausrichtung
	Mögliche Werte: vcHorCenterAligned -1 vcLeftAligned -3 vcRightAligned -2	horizontal mittig linksbündig rechtsbündig
⇒ gapAsNoOfTimeUnits	Long	Anzahl der Zeiteinheiten
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Call VcGantt1.ScrollToDate("20.10.14", vcLeftAligned, 2)
```

ScrollToGroupLine

Methode von VcGantt

Mit dieser Methode können Sie zu der Zeile einer bestimmten Gruppe scrollen und festlegen, ob diese Gruppe oben, unten oder mittig auf dem Bildschirm erscheinen soll.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Gruppe, zu der gescrollt werden soll
⇒ verAlignment	VerticalAlignmentEnum	Vertikale Ausrichtung

	Mögliche Werte: vcBottomAligned 2 vcTopAligned 1 vcVerCenterAligned -1	bündig unten bündig oben vertikal mittig
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

ScrollToNode

Methode von VcGantt

Mit dieser Methode können Sie zu einem bestimmten Knoten scrollen und festlegen, ob dieser Knoten oben, unten oder mittig auf dem Bildschirm erscheinen soll.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knoten
⇒ verAlignment	VerticalAlignmentEnum	Vertikale Ausrichtung
	Mögliche Werte: vcBottomAligned 2 vcTopAligned 1 vcVerCenterAligned -1	bündig unten bündig oben vertikal mittig
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Dim node As VcNode

Set node = VcGantt1.GetNodeByID(" 2")
VcGantt1.ScrollToNode node, vcVerCenterAligned
```

ScrollToNodeLine

Methode von VcGantt

Mit dieser Methode können Sie zu der Zeile eines bestimmten Knotens scrollen und festlegen, ob dieser Knoten oben, unten oder mittig auf dem Bildschirm erscheinen soll.

Hinweis: Bei einer optimierten Darstellung werden alle Vorgänge einer Gruppe in einer Zeile ausgegeben. Überschneiden sich Vorgänge einer Gruppe, so werden sie im expandierten Status der Gruppe automatisch untereinander angeordnet, um Überlagerungen zu vermeiden. In diesem Fall

bewirkt die Methode **ScrollToNodeLine**, dass zu der entsprechenden Gruppenzeile, in der sich der angegebene Knoten befindet gescrollt, wird. Unter Umständen steht dann der angegebene Knoten nicht mittig auf dem Bildschirm und ist ggf. nicht sofort sichtbar.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knoten
⇒ verAlignment	VerticalAlignmentEnum	vertikale Ausrichtung
	Mögliche Werte: vcBottomAligned 2 vcTopAligned 1 vcVerCenterAligned -1	bündig unten bündig oben vertikal mittig
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Imports NETRONIC.XGantt
...
Dim i As Integer
Dim objDataRecord(2) As Object
Dim dataRecordCol As VcDataRecordCollection

VcGantt1.ExtendedDataTablesEnabled = True
dataRecordCol =
VcGantt1.DataTableCollection.DataTableByName("Maindata").DataRecordCollection
    For i = 1 To 100
        objDataRecord(0) = i
        objDataRecord(1) = "Node " + i.ToString()
        dataRecordCol.Add(objDataRecord)
    Next
VcGantt1.EndLoading()
VcGantt1.ScrollToNodeLine(VcGantt1.GetNodeByID("50"),
VcVerticalAlignment.vcTopAligned)
```

ShowExportGraphicsDialog

Methode von VcGantt

Mit dieser Methode können Sie ein **Speichern unter**-Dialogfeld aufrufen, um die Darstellung abzuspeichern. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)

- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. mit eingebautem EMF)

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten lesen Sie bitte im Kapitel: **Wichtige Konzepte: Grafikformate**.

Beim Exportieren wird die Größe des exportierten Diagramms in Pixeln wie folgt berechnet:

- PNG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen.
- GIF, TIFF, BMP, JPEG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert ≤ -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.
- WMF: Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.
- EMF/EMF+: Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand verwendet.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Diagramm erfolgreich (True) / nicht erfolgreich (False) exportiert

Code-Beispiel

```
VcGantt1.ShowExportGraphicsDialog
```

SortGroups

Methode von VcGantt

Mit dieser Methode starten Sie bei einer gruppierten Darstellung eine Sortierung der Gruppen bezüglich des eingestellten Sortierparameters für Gruppen **GroupingOrderField (GroupingLevel)**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Gruppen erfolgreich sortiert/nicht erfolgreich sortiert

Code-Beispiel

```
VcGantt1.GroupingOrderField(0) = 12
VcGantt1.SortGroups
```

SortNodes

Methode von VcGantt

Mit dieser Methode starten Sie eine Sortierung der Vorgänge bezüglich der eingestellten Sortierparameter (**SortField (sortLevel)** und **SortOrder (sortLevel)**). Ist eine Gruppierung aktiv, so wirkt die Sortierung gruppenweise auf die Vorgänge.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Sortierung erfolgreich (True) / nicht erfolgreich (False)

Code-Beispiel

```
VcGantt1.SortField (0) = 3
VcGantt1.SortOrder (0) = vcAscending

VcGantt1.SortNodes
```

SuspendUpdate

Methode von VcGantt

Bei größeren Datenmengen kann es unter Umständen zu lange dauern, wenn man bei einer großen Anzahl von Knoten dieselbe Aktion durchführt. Dies kann man mit Hilfe der Methode **SuspendUpdate** beschleunigen. Klammern Sie den Code für die wiederholte Aktion wie im Code-Beispiel durch **SuspendUpdate (True)** und **SuspendUpdate (False)** ein. Dann wird das Update nicht für jeden Knoten einzeln, sondern für alle gemeinsam durchgeführt, wodurch die Performance erhöht wird.

	Datentyp	Beschreibung
Parameter: ⇒ suspendFlag	Boolean Mögliche Werte:	SuspendUpdate(True): Beginn der SuspendUpdate Methode/ SuspendUpdate(False): Ende der SuspendUpdate Methode Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
Rückgabewert	Void	

Code-Beispiel

```
VcGantt1.SuspendUpdate (True)
```

```

If updateFlag Then
    For Each node In nodeCltn
        If node.DataField(2) < "07.09.14" Then
            node.DataField(13) = "X"
            node.UpdateNode
            counter = counter + 1
        End If
    Next node
Else
    For Each node In nodeCltn
        If node.DataField(2) < "07.09.14" Then
            node.DataField(13) = ""
            node.UpdateNode
            counter = counter + 1
        End If
    Next node
End If

```

```
VcGantt1.SuspendUpdate (False)
```

UpdateLinkRecord

Methode von VcGantt

Mit dieser Methode können die Daten eines bestehenden Datensatzes einer Verbindung verändert werden. Die Verbindung wird durch den im Dialog **Datentabellen verwalten** festgelegten Primärschlüssel identifiziert. Diese

Methode kommt zur Anwendung, wenn externe Änderungen der Daten in der Grafik nachvollzogen werden sollen. Wenn die Verbindung, die aktualisiert werden soll, noch nicht existiert, so wird sie neu angelegt.

	Datentyp	Beschreibung
Parameter: ⇒ linkRecordContent	Object	Inhalt des Verbindungsdatensatzes
Rückgabewert	VcLink	Aktualisierte Verbindung

Code-Beispiel

```
VcGantt1.UpdateLinkRecord ("A100;A105;FS;0")
```

UpdateNodeRecord

Methode von VcGantt

Mit dieser Methode können die Daten eines bestehenden Datensatzes eines Knotens verändert werden. Der Datensatz wird durch den im Dialog **Datentabellen verwalten** festgelegten Primärschlüssel identifiziert. Diese Methode kommt zur Anwendung, wenn externe Änderungen der Daten in der Grafik nachvollzogen werden sollen.

	Datentyp	Beschreibung
Parameter: ⇒ nodeRecordContent	Object	Inhalt des Knotendatensatzes
Rückgabewert	VcNode	Datensatz des Knotens erfolgreich (True) / nicht erfolgreich (False) aktualisiert

Code-Beispiel

```
VcGantt1.UpdateNodeRecord ("A100;Activity 1;12.09.14;18.09.14;6;Planning")
```

UpdateRowNumberFields

Methode von VcGantt

Diese Methode aktualisiert das Feld, in dem sich die Zeilennummern der Knoten befinden. Dieses Feld kann auf der Eigenschaftenseite <bKnoten unter **Zeilennummer-Feld** eingestellt werden. Die Anwendung dieser Methode ist nur dann sinnvoll, wenn weder eine hierarchische Darstellung noch eine Gruppierung gewählt wurde.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcGantt1.UpdateRowNumberFields
VcGantt1.SaveAs ("C:\tmp\data.bar")
```

Zoom

Methode von VcGantt

Mit dieser Methode wird die augenblickliche Bildschirmdarstellung um den angegebenen Wert prozentual vergrößert (Zoomfaktor > 100) oder verkleinert (Zoomfaktor < 100).

Siehe auch die VcGantt-Methode **FitChartIntoView()** und die Eigenschaft **ZoomFactor**.

	Datentyp	Beschreibung
Parameter: ⇒ zoomFactor	Integer Mögliche Werte:	relativer Zoomfaktor {11...999}, andere Werte bleiben unberücksichtigt Datenfeldindex
Rückgabewert	Boolean	Zoomen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
VcGantt1.Zoom 120
```

Ereignisse

Error

Ereignis von VcGantt

Dieses Ereignis tritt nur dann auf, wenn ein unvorhergesehener Fehler im Code des VARCHART XGantt entdeckt wird. NETRONIC ist bemüht, jeden dieser Fehler zu beseitigen. Um sie auf einfache Art beim Kunden, z. B. in einer Datei, protokollieren zu können, werden sie nun über dieses Ereignis nach außen bekanntgegeben. Das Parameterprofil ist durch den ActiveX-

Standard vorgegeben. Dadurch sind die übergebenen Parameter teilweise konstant. Die Nummer sollte im Ereignis immer gegengeprüft werden, um bei zukünftigen Erweiterungen nicht alle Fehlerarten pauschal abzublocken.

	Datentyp	Beschreibung
Parameter:		
⇒ Description	String	Fehlerbeschreibungstext
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ Scode	Long	&h800a402f (konstant)
⇒ Source	String	Name des Controls (konstant)
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ HelpFile	String	Hilfedatei: "" (konstant)
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ HelpContext	Long	Hilfekontext: 0 (konstant)
⇐ CancelDisplay	Boolean	Beim Wert True wird kein normaler Fehler mit der Nummer 71 mehr ausgegeben, der über das 'On Error GoTo'-Konstrukt abfangbar wäre.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub VcGantt1_Error(Number As Integer, Description As String, _
    Scode As Long, Source As String, HelpFile As String, HelpContext _
    As Long, CancelDisplay As Boolean)

    Debug.Print CStr(Number) + " " + Description

End Sub
```

ErrorAsVariant

Ereignis von VcGantt

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **Error**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇐) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

KeyDown

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender eine Taste drückt, während VARCHART XGantt den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen. (Zum Interpretieren von ANSI-Zeichen verwenden Sie das KeyPress-Ereignis.)

	Datentyp	Beschreibung
Parameter:		
⇒ KeyCode	Integer	Tasten-Code wie vbKeyF1 (F1-Taste) oder vbKeyHome (POS1-Taste)
	Mögliche Werte:	Datenfeldindex
⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Private Sub VcGantt1_KeyDown(KeyCode As Integer, Shift As Integer)
    MsgBox "key pressed"
End Sub
```

KeyPress

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender eine ANSI-Taste drückt und wieder loslässt, während VARCHART XGantt den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen.

	Datentyp	Beschreibung
Parameter: ⇒ KeyAscii	Integer	Eine Ganzzahl, die den numerischen Tasten-Code einer Standard-ANSI-Taste zurückgibt. KeyAscii wird als Referenz übergeben. Wird das Argument geändert, wird ein anderes Zeichen an das Objekt gesendet. Das Ändern von KeyAscii auf 0 hebt den Tastenanschlag auf, d.h. das Objekt erhält kein Zeichen.
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Private Sub VcGantt1_KeyPress(KeyAscii As Integer)
    MsgBox "Key pressed and released."
End Sub
```

KeyUp**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender eine Taste loslässt, während VARCHART XGantt den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen. (Zum Interpretieren von ANSI-Zeichen verwenden Sie das KeyPress-Ereignis.)

	Datentyp	Beschreibung
Parameter: ⇒ KeyCode	Integer	Tasten-Code wie vbKeyF1 (F1-Taste) oder vbKeyHome (POS1-Taste)
	Mögliche Werte:	Datenfeldindex
⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Private Sub VcGantt1_KeyUp(KeyCode As Integer, Shift As Integer)
```

```

    MsgBox "key released"
End Sub

```

OLECompleteDrag

Ereignis von VcGantt

Dieses Ereignis tritt auf, wenn ein OLE Drag & Drop-Vorgang beendet wurde und die Quellkomponente darüber informiert wird, dass ein Drag&Drop-Vorgang entweder erfolgreich durchgeführt oder abgebrochen wurde.

	Datentyp	Beschreibung
Parameter: ⇨ effect	Long	Durchgeführte Aktion in der Zielkomponente
	Mögliche Werte: vcDropEffectCopy 1 vcDropEffectMove 2 vcDropEffectNone 0	Die Drop-Operation führt zum Kopieren der Daten von der Quelle zum Ziel. Die ursprünglichen Daten wurden durch die Drag-Operation nicht geändert. Die Drop-Operation führt dazu, dass Daten von der Quelle zum Ziel verschoben werden. Die Quelle sollte die Daten nach erfolgreichem Verschieben aus sich selbst entfernen. Ziel kann die Daten nicht akzeptieren.

Code-Beispiel

```

Private Sub VcGantt1_OLECompleteDrag(ByVal effect As Long)

    MsgBox effect

End Sub

```

OLEDragDrop

Ereignis von VcGantt

Dieses Ereignis tritt auf, wenn bei einem OLE Drag & Drop-Vorgang Daten auf dem Ziel abgelegt werden und die **OLEDropMode**-Eigenschaft des Ziels auf **vcOLEDropManual** gesetzt ist und Ziel- und Quellkomponente nicht identisch sind. Bei Gleichheit der beiden Komponenten erhält man stattdessen eines der beiden Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**.

	Datentyp	Beschreibung
Parameter: ⇨ data	DataObject	Übergebene Objektdaten
⇨ effect	Long	Durchzuführende Aktion

	Mögliche Werte:	
	vcDropEffectCopy	1
	vcDropEffectMove	2
	vcDropEffectNone	0
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	
	Datenfeldindex	
⇒ shift	Integer	Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen 1 , 2 und 4 entsprechen. Der Parameter Shift zeigt den Status dieser Tasten an; es können einige, alle oder keine der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise sowohl die <STRG>- als auch die <ALT>-Tasten gedrückt würden, wäre der Wert von Shift 6 .
	Mögliche Werte:	
	Datenfeldindex	
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OLEDragOver

Ereignis von VcGantt

Dieses Ereignis tritt auf, wenn Daten über ein Ziel gezogen und die **OLEDropMode**-Eigenschaft des Drop-Ziels auf **vcOLEDropManual** gesetzt ist.

	Datentyp	Beschreibung
Parameter:		
⇒ data	DataObject	Übergebene Objektdaten
⇒ effect	Long	Wert, der die Aktion angibt, die durchgeführt werden soll, wenn der Benutzer das Objekt dort ablegt. Der Wert ermöglicht der Quellkomponente, darauf zu reagieren (z.B. durch ein visuelles Signal).
	Mögliche Werte:	
	vcDropEffectCopy	1
		Die Drop-Operation führt zum Kopieren der Daten von der Quelle zum Ziel. Die ursprünglichen Daten wurden durch die Drag-Operation nicht geändert.

	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten von der Quelle zum Ziel verschoben werden. Die Quelle sollte die Daten nach erfolgtem Verschieben aus sich selbst entfernen.
	vcDropEffectNone 0	Ziel kann die Daten nicht akzeptieren.
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	Datenfeldindex
⇒ shift	Integer	Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Drop-Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen 1 , 2 und 4 entsprechen. Der Parameter shift zeigt den Status dieser Tasten an; es können einige, alle oder keines der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise die <STRG>- und die <ALT>-Tasten gedrückt werden, ist der Wert von shift "6".
	Mögliche Werte:	Datenfeldindex
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇐ state	OLEDragStateEnum	Drag-Status
	Mögliche Werte:	
	vcEnter 0	Objekt der Quellkomponente erreicht die Zielkomponente
	vcLeave 1	Objekt der Quellkomponente verlässt die Zielkomponente
	vcOver 2	Objekt der Quellkomponente veränderte seine Position innerhalb der Zielkomponente.

OLEGiveFeedback

Ereignis von VcGantt

Dieses Ereignis tritt nach jedem OLEDragOver-Ereignis auf der Quellkomponente auf. OLEGiveFeedback gibt der Quellkomponente die Möglichkeit, dem Benutzer ein "visuelles Feedback" zu geben. Dies kann z. B. die Änderung des Cursors sein, um anzuzeigen, was passiert, wenn der Benutzer das Objekt ablegt; in anderen Fällen wird ein visuelles Feedback für die ausgewählten Objekte in der Quellkomponente gegeben, um anzuzeigen, was geschehen wird.

	Datentyp	Beschreibung
Parameter: ⇨ effect	Long Mögliche Werte: vcDropEffectCopy 1 vcDropEffectMove 2 vcDropEffectNone 0	Wert, der von der Zielkomponente beim OLEDragOver-Ereignis gesetzt wird und der die Aktion angibt, die durchgeführt werden soll, wenn der Anwender das Objekt dort ablegt. Der Wert ermöglicht der Quellkomponente, darauf zu reagieren (z.B. durch ein visuelles Signal). Die Drop-Operation führt zum Kopieren der Daten von der Quelle zum Ziel. Die ursprünglichen Daten wurden durch die Drag-Operation nicht geändert. Die Drop-Operation führt dazu, dass Daten von der Quelle zum Ziel verschoben werden. Die Quelle sollte die Daten nach erfolgreichem Verschieben aus sich selbst entfernen. Ziel kann die Daten nicht akzeptieren.
⇨ defaultCursors	Boolean Mögliche Werte:	Verwendung des Standard-Cursors (True) oder eines benutzerdefinierten Cursors (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

OLESetData

Ereignis von VcGantt

Dieses Ereignis tritt bei einer Quelle auf, wenn ein Ziel die **GetData**-Methode aufruft und es in dem OLE Drag & Drop-DataObject keine Daten in dem angeforderten Format gibt.

	Datentyp	Beschreibung
Parameter:		
⇒ data	JsonObject	Übergebene Objektdaten
⇒ dataFormat	Integer	Spezifiziert das Datenformat, das von der Zielkomponente erwartet wird. Mithilfe dieses Wertes ermittelt die Quellkomponente, womit das Datenobjekt zu füllen ist.
	Mögliche Werte:	Datenfeldindex

OLEStartDrag

Ereignis von VcGantt

Dieses Ereignis tritt bei der Quelle auf, wenn eine VARCHART-XGantt-Komponente eine OLE Drag & Drop-Operation auslöst und die **OLEDrag-Mode**-Eigenschaft auf **vcOLEDragAutomatic** gesetzt ist.

Dieses Ereignis legt die Datenformate und Ablege-Effekte fest, die die Quellkomponente unterstützt. Es kann auch zum Einfügen von Daten in das DataObject-Objekt verwendet werden.

Die Quellkomponente sollte den logischen Operator **Or** auf die unterstützten Werte anwenden und das Ergebnis in den Parameter **allowedEffect** übernehmen. Die Zielkomponente kann diesen Wert zum Bestimmen der passenden Aktion (und wie das geeignete Benutzer-Feedback aussehen soll) verwenden.

Das Hinzufügen von Daten in das **DataObject** sollten Sie verschieben, bis die entsprechende Anforderung von der Zielkomponente kommt. Dadurch kann die Zielkomponente Zeit sparen, weil nicht mehrere Formate geladen werden müssen. Wenn das Ziel die **GetData**-Methode für das **DataObject** ausführt, tritt das **OLESetData**-Ereignis der Quelle auf, wenn die angeforderten Daten nicht im **DataObject** enthalten sind. Zu diesem Zeitpunkt können die Daten in das **DataObject** geladen werden, wodurch wiederum die Daten für das Ziel bereitgestellt werden.

Falls der Benutzer keine Formate in das **DataObject** lädt, wird die Drag&Drop-Operation abgebrochen.

	Datentyp	Beschreibung
Parameter:		
⇒ data	DataObject	Übergebene Objektdaten
⇒ effect	Long	Wert, der von der Zielkomponente beim OLEDragOver-Ereignis gesetzt wird und der die Aktion angibt, die durchgeführt werden soll, wenn der user das Objekt dort ablegt. Der Wert ermöglicht der Quellkomponente, darauf zu reagieren (z.B. durch ein visuelles Signal).
	Mögliche Werte:	
	vcDropEffectCopy 1	Die Drop-Operation führt zum Kopieren der Daten von der Quelle zum Ziel. Die ursprünglichen Daten wurden durch die Drag-Operation nicht geändert.
	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten von der Quelle zum Ziel verschoben werden. Die Quelle sollte die Daten nach erfolgreichem Verschieben aus sich selbst entfernen.

vcDropEffectNone 0	Ziel kann die Daten nicht akzeptieren.
--------------------	--

OnBoxCreate

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Box erzeugt.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnBoxCreateComplete**.

Durch Setzen des Rückgabestatus kann die Anlage verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ XOffset	Long	X-Position der Box
⇒ YOffset	Long	Y-Position der Box
⇒ Width	Long	Breite der Box
⇒ Height	Long	Höhe der Box
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Box wird nicht angelegt. Die Box wird angelegt.

Code-Beispiel

```
Private Sub VcGantt1_OnBoxCreate(ByVal XOffset As Long, ByVal YOffset As Long,
ByVal Width As Long, ByVal Height As Long, returnStatus As Variant)
    MsgBox "XOffset " & XOffset & " - " & "YOffset " & YOffset
End Sub
```

OnBoxCreateComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das interaktive Anlegen einer Box abgeschlossen ist. Das Box-Objekt wird als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	angelegte Box

Code-Beispiel

```
Private Sub VcGantt1_OnBoxCreateComplete(ByVal box As VcGanttLib.VcBox)
    box.LineColor = RGB(255, 0, 0)
End Sub
```

OnBoxLClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Box klickt. Das getroffene VcBox-Objekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter mitgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	Getroffene Box
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnBoxLClick(ByVal box As VcGanttLib.VcBox, _
    ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    Text1.Text = box.FieldText(1)
End Sub
```

OnBoxLDbClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Box doppelklickt. Das getroffene VcBox-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	getroffene Box
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnBoxLDbClick(ByVal box As VcGanttLib.VcBox, _
```

```

        ByVal x As Long, ByVal y As Long, returnStatus As Variant)
    box.FieldText(0) = Text1.Text
End Sub

```

OnBoxModify

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn eine Box interaktiv verändert wurde. Das veränderte VcBox-Objekt und der Modifikationstyp werden als Parameter mitgegeben.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnBoxModifyComplete**.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	Veränderte Box
⇒ modificationType	BoxModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcBMTAnchoringModified 16 vcBMTAnything 1 vcBMTNothing 0 vcBMTSizeModified 8 vcBMTTextModified 4 vcBMTXYOffsetModified 2	Verankerung der Box geändert beliebige Veränderung keine Veränderung Größe der Box verändert Text der Box verändert Offset verändert
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Veränderung wird rückgängig gemacht. Die Veränderung wird durchgeführt.

Code-Beispiel

```

Private Sub VcGantt1_OnBoxModify(ByVal box As VcGanttLib.VcBox, _
    ByVal modificationType As _
    VcGanttLib.BoxModificationTypeEnum, _
    returnStatus As Variant)

    Select Case modificationType
        Case vcBMTAnything: MsgBox "Box modification"
        Case vcBMTXYOffsetModified: MsgBox "Offset changed"
        Case vcBMTTextModified: MsgBox "Box field text changed"
    End Select

End Sub

```

OnBoxModifyCompleteEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Modifizierung der Box abgeschlossen ist. Das veränderte VcBox-Objekt und der Modifikationstyp werden als Parameter mitgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ modificationType	BoxModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcBMTAnchoringModified 16 vcBMTAnything 1 vcBMTNothing 0 vcBMTSizeModified 8 vcBMTTextModified 4 vcBMTXYOffsetModified 2	Verankerung der Box geändert beliebige Veränderung keine Veränderung Größe der Box verändert Text der Box verändert Offset verändert

Code-Beispiel

```
Private Sub VcGantt1_OnBoxModifyCompleteEx(ByVal box As _
    VcGanttLib.VcBox)

    MsgBox "The box has been modified."

End Sub
```

OnBoxRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Box klickt. Das getroffene Boxobjekt wird zusammen mit der Position (x,y-Koordinaten) des Cursors als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	Getroffene Box
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnBoxRClick(ByVal box As VcGanttLib.VcBox, _
    ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuBoxPopup

End Sub
```

OnCalendarGridRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf ein Kalendergitter mit der rechten Maustaste klickt. Das getroffene Kalendergitter-Objekt und die Mausposition (x,y-Koordinaten) werden als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

Dieses Ereignis wird nur ausgelöst, wenn das Kalendergitter identifizierbar ist, d.h. wenn die Kalendergitter-Eigenschaft **Identifiable** auf **True** gesetzt wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ dateLine	VcDateLine	Kalendergitter
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnCalendarGridRClick(ByVal group As
VcGanttLib.VcCalendarGrid, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuCalendarGridPopup

    ' Suppress built-in context menu
    returnStatus = vcRetStatNoPopup

End Sub
```

OnCurveLClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Histogrammkurve klickt und bevor die Kurve markiert wird. Durch Setzen des VcReturnStatus auf **vcRetStatFalse** kann das Markieren der Kurve verhindert werden. Trotzdem können die Kurvenwerte verändert werden. Eine Möglichkeit zur Unterdrückung gibt es zur Zeit nicht. Das getroffene Kurvenobjekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Kurve wird nicht markiert. Die Kurve wird markiert.

Code-Beispiel

```
Private Sub VcGantt1_OnCurveLClick(ByVal curve As _
                                   VcGanttLib.VcCurve, ByVal x As Long, _
                                   ByVal y As Long, returnStatus As Variant)
    curve.LineColor = vbRed
End Sub
```

OnCurveLDbClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf eine Histogrammkurve mit der linken Maustaste doppelklickt. Das getroffene VcCurve-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnCurveLDbClick(ByVal curve As
                                VcGanttLib.VcCurve, ByVal x As Long, _
                                ByVal y As Long, returnStatus As Variant)
    Call MsgBox("x: " & x & vbCrLf & "y: " & y)
End Sub
```

OnCurveModifyComplete**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn die Modifizierung der Kurve abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Veränderte Kurve

OnCurveModifyEx**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn eine Histogrammkurve interaktiv verändert wurde. Das gilt sowohl für das interaktive Verändern von über die API gesetzten Histogrammkurven, als auch für das Verändern von layergenerierten Histogrammkurven durch das Verschieben von Knoten. Die veränderte Kurve, der Anfang und das Ende des Bereiches, in dem sie verändert wurde, sowie der Wert, um den der Kurvenpunkt in y-Richtung verändert wurde, werden als Parameter zurückgegeben. Die Art der veränderten Kurve kann über die VcCurve-Eigenschaft **CurveSource** erfragt werden.

Hinweis: Für jeden veränderten Layer, der zu einer Veränderung einer layergenerierten Kurve beiträgt, kommt das Ereignis **OnCurveModifyEx** je zweimal (einmal für den Anfangsbereich und einmal für den Endbereich der Veränderung).

Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	veränderte Kurve

⇒ date1	Date/Time	Anfang des Bereiches, in dem die Kurve verändert wurde
⇒ date2	Date/Time	Ende des Bereiches, in dem die Kurve verändert wurde
⇒ increment	Long	Wert, um den die Kurve in y-Richtung verändert wurde
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnCurveModifyEx(ByVal curve As _
                                   VcGanttLib.VcCurve, ByVal date1 As Date, _
                                   ByVal date2 As Date, ByVal increment As Long, _
                                   returnStatus As Variant)

    Select Case curve.CurveSource
        Case 1
            MsgBox "The curve is calculated from layers." & vbCrLf _
                & "Increment: " & increment & vbCrLf _
                & "Changed start date: " & date1 & vbCrLf _
                & "Changed end date: " & date2
        Case 3
            MsgBox "Curve set via API." & vbCrLf _
                & "Increment: " & increment & vbCrLf _
                & "Changed start date: " & date1 & vbCrLf _
                & "Changed end date: " & date2
    End Select
End Sub
```

OnCurveModifyEx2

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn eine Histogrammkurve interaktiv verändert wurde. Das gilt sowohl für das interaktive Verändern von über die API gesetzten Histogrammkurven, als auch für das Verändern von layergenerierten Histogrammkurven durch das Verschieben von Knoten. Die veränderte Kurve, der Anfang und das Ende des Bereiches, in dem sie verändert wurde, sowie der Wert, um den der Kurvenpunkt in y-Richtung verändert wurde, werden als Parameter zurückgegeben. Die Art der veränderten Kurve kann über die VcCurve-Eigenschaft **CurveSource** erfragt werden.

Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

Hinweis: Für jeden veränderten Layer, der zu einer Veränderung einer layergenerierten Kurve beiträgt, kommt das Ereignis **OnCurveModifyEx** je zweimal (einmal für den Anfangsbereich und einmal für den Endbereich der Veränderung).

Hinweis: Im Vergleich zum Ereignis **OnCurveModifyEx** kann hier der Parameter **increment** als Fließkommazahl übergeben werden.

	Datentyp	Beschreibung

Code-Beispiel

```
Private Sub VcGantt1_OnCurveModifyEx(ByVal curve As _
    VcGanttLib.VcCurve, ByVal date1 As Date, _
    ByVal date2 As Date, ByVal increment As Double,
    _
    returnStatus As Variant)
    Select Case curve.CurveSource
        Case 1
            MsgBox "The curve is calculated from layers." & vbCrLf _
                & "Increment: " & increment & vbCrLf _
                & "Changed start date: " & date1 & vbCrLf _
                & "Changed end date: " & date2
        Case 3
            MsgBox "Curve set via API." & vbCrLf _
                & "Increment: " & increment & vbCrLf _
                & "Changed start date: " & date1 & vbCrLf _
                & "Changed end date: " & date2
    End Select
End Sub
```

OnCurveModifyExAsString

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn eine Histogrammkurve interaktiv verändert wurde. Das gilt sowohl für das interaktive Verändern von über die API gesetzten Histogrammkurven, als auch für das Verändern von layergenerierten Histogrammkurven durch das Verschieben von Knoten. Die veränderte Kurve, der Anfang und das Ende des Bereiches, in dem sie verändert wurde, sowie der Wert, um den der Kurvenpunkt in y-Richtung verändert wurde, werden als Parameter zurückgegeben. Die Art der veränderten Kurve kann über die VcCurve-Eigenschaft **CurveSource** erfragt werden.

Der Datumsstring hat das feste Format "DD.MM.YYYY;hh:mm:ss;".

Hinweis: Für jeden veränderten Layer, der zu einer Veränderung einer layergenerierten Kurve beiträgt, wird das Ereignis **OnCurveModifyExAsString** je zweimal aufgerufen (einmal für den Anfangsbereich und einmal für den Endbereich der Veränderung).

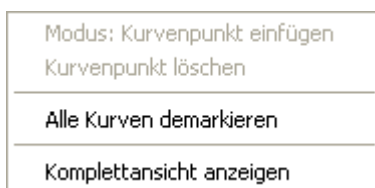
Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Änderung rückgängig gemacht.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	veränderte Kurve
⇒ date1	String	Anfang des Bereiches, in dem die Kurve verändert wurde
	Mögliche Werte:	Name der Farbzuoordnungstabelle
⇒ date2	String	Ende des Bereiches, in dem die Kurve verändert wurde
	Mögliche Werte:	Name der Farbzuoordnungstabelle
⇒ increment	Long	Wert, um den die Kurve in y-Richtung verändert wurde
⇔ returnStatus	Variant	Rückgabestatus

OnCurveRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Histogrammkurve klickt. Das getroffene Kurvenobjekt wird zusammen mit der Position (x,y-Koordinaten) des Cursors als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.



Oben: integriertes Kontextmenü

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Kurve
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte:	
	vcRetStatNoPopup 4	Das Kontextmenü wird unterdrückt.

vcRetStatOK 1	Das Kontext-Menü erscheint.
---------------	-----------------------------

Code-Beispiel

```
Private Sub VcGantt1_OnCurveRClick(ByVal curve As _
                                VcGanttLib.VcCurve, ByVal x As Long, _
                                ByVal y As Long, returnStatus As Variant)

    'Start own popup menu at the current mouse cursor position
    PopupMenu mnuCurvePopup

    ' Switch off the built-in context menu
    returnStatus = vcRetStatNoPopup

End Sub
```

OnDataRecordCreate

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv ein Objekt erzeugt hat, das einen Datensatz generiert. Das neu erzeugte Datensatz-Objekt wird als Parameter zurückgegeben, so dass eine Validierung vorgenommen werden kann.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnDataRecordCreateComplete**.

Durch Setzen des Rückgabestatus kann die Anlage verhindert werden.

Wenn eine Verbindung oder ein Knoten angelegt wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNodeCreate** und **OnLinkCreate**).

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Angelegter Datensatz
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Datensatz wird nicht angelegt. Der Datensatz wird angelegt.

Code-Beispiel

```
Private Sub VcGantt1_OnDataRecordCreate(ByVal node As VcGanttLib.VcDataRecord, _
                                        returnStatus As Variant)
```

```

'Show own "Edit" dialog for the new data record
' (EditNewDataRecord attribute must be set to off!)
On Error GoTo CancelError
frmEditDialog.Show Modal, Me

addDataRecord dataRecord.AllData

Exit Sub

CancelError:
returnStatus = vcRetStatFalse

End Sub

```

OnDataRecordCreateComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das interaktive Anlegen eines Objektes beendet ist, das einen Datensatz erzeugt. Das DataRecord-Objekt, die Form des Anlegens (hier nur **vcDataRecordCreated** und **vcDataRecordCreated-ByResourceScheduling**) und die Information, ob der angelegte Datensatz der einzige Datensatz oder der letzte einer Menge ist (derzeit immer **True**), werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

Wenn eine Verbindung oder ein Knoten angelegt wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNodeCreateComplete** und **OnLinkCreateComplete**).

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Angelegter Datensatz
⇒ creationType	CreationTypeEnum	Typ des Anlegens
	Mögliche Werte:	
	vcDataRecordCreated 6	Datensatz wurde durch Interaktion angelegt
	vcDataRecordCreatedByResourceScheduling 5	Datensatz wurde automatisch durch Ressourcenplanung angelegt
	vcLinkCreated 2	Verbindung durch das Verbinden zweier Knoten angelegt
	vcNodeCreated 1	Knoten durch "Stempeln" angelegt
⇒ isLastNodeInSeries	Boolean	True: Angelegter Datensatz ist der einzige oder der letzte Datensatz einer Menge False: Angelegter Datensatz ist nicht der einzige oder der letzte Datensatz einer Menge

	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
--	------------------------	---

Code-Beispiel

```
Private Sub VcGantt1_OnDataRecordCreateComplete(ByVal dataRecord As _
                                                VcGanttLib.VcDataRecord, ByVal creationType As
-
                                                VcGanttLib.CreationTypeEnum, _
                                                ByVal isLastDataRecordInSeries As Boolean)
    addDataRecord dataRecord.AllData
End Sub
```

OnDataRecordDelete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs ein Objekt löscht, das auf einem Datensatz-Objekt basiert. Der betroffene Datensatz wird als Parameter zurückgegeben, so dass Sie z. B. noch eine Überprüfung vornehmen und bei negativem Ergebnis dieser Prüfung die Löschung ggf. durch Setzen des Rückgabestatus verhindern können.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Gelöschter Datensatz
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Datensatz wird nicht gelöscht. Der Datensatz wird gelöscht.

Code-Beispiel

```
Private Sub VcGantt1_OnDataRecordDelete(ByVal node As VcGanttLib.VcNode, _
                                         returnStatus As Variant)
    'deny the deletion of the last data record in the chart
    If VcGantt1.DataRecordCollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last data record cannot be deleted.")
    End If
End Sub
```

OnDataRecordDeleteComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das Löschen eines Objektes, das auf einem Datensatz-Objekt basiert, beendet ist. Der Datensatz und die Information, ob der betroffene Datensatz der einzige oder der letzte Datensatz einer Menge

ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

Wenn eine Verbindung oder ein Knoten gelöscht wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNodeDeleteComplete** und **OnLinkDeleteComplete**).

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Gelöschter Datensatz
⇒ isLastNodeInSeries	Boolean	True: Gelöschter Datensatz ist der einzige oder der letzte Datensatz einer Menge False: Gelöschter Datensatz ist nicht der einzige oder der letzte Datensatz einer Menge
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

OnDataRecordModify

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn ein Objekt interaktiv verändert wurde und sich der zu Grunde liegende Datensatz verändert. Das veränderte VcDataRecord-Objekt und der Modifikationstyp werden als Parameter zurückgegeben.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnDataRecordModifyComplete**.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ dataRecord	VcBox	Veränderte Box
⇒ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcAnything 1 vcChangedGroup 16 vcEndModified 4 vcHierarchyModified 64	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Ende des Knotens wurde verändert Hierarchie der Knoten wurde verändert

⇔ returnStatus	vcModifiedByResourceScheduling 128	Änderung durch Ressourcenplanung (nur für Datensatz-Objekt)
	vcModifiedBySchedule 32	Änderung durch neue Zeitberechnung
	vcMoved 8	Objekt wurde verschoben
	vcNothing 0	keine Änderung
	vcStartModified 2	Anfang des Knotens wurde verändert
Variant		
Mögliche Werte:		
	vcRetStatFalse 0	Die Veränderung wird rückgängig gemacht.
	vcRetStatOK 1	Die Veränderung wird durchgeführt.
Rückgabestatus		

OnDataRecordModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Modifizierung des Datensatzes abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇔ dataRecord	VcDataRecord	Veränderter Datensatz

Code-Beispiel

```
Private Sub VcGantt1_OnDataRecordModifyComplete(ByVal box As _
    VcGanttLib.VcBox)

    MsgBox "The data record has been modified."

End Sub
```

OnDataRecordNotFound

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn ein abhängiger Datensatz nicht gefunden wurde. Der Index des Feldes im aktuellen Datensatz, in dem der Schlüsselwert des abhängigen Datensatzes steht, wird zurückgegeben und bietet so Informationen über den nicht gefundenen Datensatz.

	Datentyp	Beschreibung
Parameter:		
⇔ index	Long	Index des Feldes, das den Schlüssel des abhängigen Datensatzes enthält

OnDateLineModify

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn eine Stichtaglinie durch den Anwender interaktiv verschoben wurde. Die veränderte Stichtaglinie wird als Parameter zurückgegeben.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ dateLine	VcDateLine	Stichtaglinie
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Veränderung wird rückgängig gemacht. Die Veränderung wird durchgeführt.

Code-Beispiel

```
Private Sub VcGantt1_OnDateLineModify(ByVal dateLine As _
                                     VcGanttLib.VcDateLine, _
                                     returnStatus As Variant)
    MsgBox dateLine.Date
End Sub
```

OnDateLineRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Stichtaglinie klickt. Das getroffene Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ dateLine	VcDateLine	Stichtaglinie
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnDateLineRClick(ByVal dateLine As _
    VcGanttLib.VcDateLine, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant)

    MsgBox dateLine.Name

End Sub
```

OnDeleteCurvePoint**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs einen Punkt einer über die API gesetzten Histogrammkurve löscht. Die getroffene Histogrammkurve, das Datum und der y-Wert des gelöschten Kurvenpunkts werden als Parameter zurückgegeben, so dass noch eine Überprüfung vorgenommen werden kann. Durch Setzen des Rückgabestatus kann der Löschvorgang verhindert werden.

Hinweis: Mit dem Ereignis **OnDeleteCurvePointEx** können Sie auch Fließkomma-Zahlen als Y-Werte übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ pointDate	Date	Datum des gelöschten Kurvenpunkts
⇒ value	Long	Y-Wert des gelöschten Kurvenpunkts
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Kurvenpunkt wird nicht gelöscht. Der Kurvenpunkt wird gelöscht.

Code-Beispiel

```
Private Sub VcGantt1_OnDeleteCurvePoint(ByVal curve As VcGanttLib.VcCurve, _
    ByVal pointDate As Date, ByVal value As Long, _
    returnStatus As Variant)

    If MsgBox("Do you want to delete this curve point (date:" & pointDate & _
        ", y value: " & value & ")?", vbYesNo, _
        "Deleting curve point") = vbNo Then

        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnDeleteCurvePointEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs einen Punkt einer über die API gesetzten Histogrammkurve löscht. Die getroffene Histogrammkurve, das Datum und der y-Wert des gelöschten Kurvenpunkts werden als Parameter zurückgegeben, so dass noch eine Überprüfung vorgenommen werden kann. Durch Setzen des Rückgabestatus kann der Löschvorgang verhindert werden.

Hinweis: Im Vergleich zum Ereignis **OnDeleteCurvePoint** können Sie mit diesem Ereignis Fließkomma-Werte übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ pointDate	Date	Datum des gelöschten Kurvenpunkts
⇒ value	Double	Y-Wert des gelöschten Kurvenpunkts
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Kurvenpunkt wird nicht gelöscht. Der Kurvenpunkt wird gelöscht.

Code-Beispiel

```
Private Sub VcGantt1_OnDeleteCurvePointEx(ByVal curve As VcGanttLib.VcCurve, _
                                         ByVal pointDate As Date, ByVal value As Double, _
                                         _
                                         returnStatus As Variant)
    If MsgBox("Do you want to delete this curve point (date:" & pointDate & _
              ", y value: " & value & ")?", vbYesNo, _
              "Deleting curve point") = vbNo Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnDiagramLClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste im Diagrammbereich klickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten des Cursors) wird als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers

⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramLClick(ByVal x As Long, _
                                     ByVal y As Long, returnStatus As Variant)
    MsgBox "x: " & x & vbNewLine & "y: " & y
End Sub
```

OnDiagramLDbClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste im Diagrammbereich doppelklickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten des Cursors) wird als Parameter zurückgegeben.

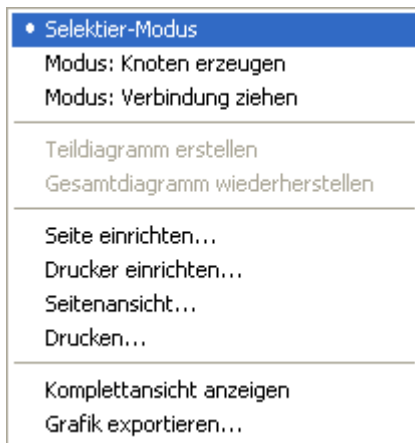
	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramLDbClick(ByVal x As Long, _
                                         ByVal y As Long, returnStatus As Variant)
    VcGantt1.Zoom (90)
End Sub
```

OnDiagramRClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste im Diagrammbereich klickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten des Mauszeigers) wird als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.



Oben: integriertes Kontextmenü

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnDiagramRClick(ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuDiagramPopup

    ' Switch off the built-in context menu
    returnStatus = vcRetStatNoPopup

End Sub
```

OnGroupDelete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Gruppe löscht. Dabei wird die Gruppe als Parameter zurückgegeben. Durch Setzen des Rückgabestatus kann der Löschvorgang verhindert werden. Es können nur Gruppen gelöscht werden, die keine Elemente enthalten.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Gelöschte Gruppe
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Gruppe wird nicht gelöscht. Die Gruppe wird gelöscht.

Code-Beispiel

```
Private Sub VcGantt1_OnGroupDelete(ByVal group As VcGanttLib.VcGroup, _
                                   returnStatus As Variant)
    If group.Name = "A" Then
        returnStatus = vcRetStatFalse
        MsgBox ("Group A cannot be deleted.")
    End If
End Sub
```

OnGroupLClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender auf eine Gruppenüberschrift in der Tabelle mit der linken Maustaste klickt. Das getroffene Group-Objekt wird zusammen mit der Position (x,y-Koordinaten des Cursors) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnGroupLClick(ByVal group As VcGanttLib.VcGroup, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)
    MsgBox group.SubGroups.Count
End Sub
```

OnGroupLDbClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender auf eine Gruppenüberschrift in der Tabelle mit der linken Maustaste doppelt klickt. Das getroffene Group-

Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnGroupLDb1Click(ByVal group As VcGanttLib.VcGroup, _
                                     ByVal x As Long, ByVal y As Long, _
                                     returnStatus As Variant)

    MsgBox group.Name

End Sub
```

OnGroupModify

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender eine Gruppe interaktiv verändert. Das getroffene Group-Objekt, die Art der Veränderung und der Rückgabestatus werden als Parameter zurückgegeben.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnGroupModifyComplete**.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ Group	VcGroup	Veränderte Gruppe
⇒ modificationType	GroupModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcGMTAnything 1 vcGMTEndModified 16 vcGMTMinusPressed 2 vcGMTMoved 32 vcGMTNothing 0 vcGMTPlusPressed 4 vcGMTStartModified 8	Änderungstyp Nicht näher bestimmt Enddatum wurde verändert Änderungstyp Minus-Symbol angeklickt Objekt wurde verschoben Änderungstyp Nichts Änderungstyp Plus-Symbol angeklickt Startdatum wurde verändert
⇔ returnStatus	Variant	Rückgabestatus

Mögliche Werte:
vcRetStatFalse 0
vcRetStatOK 1

Die Veränderung wird rückgängig gemacht.
Die Veränderung wird durchgeführt.

Code-Beispiel

```
Private Sub VcGantt1_OnGroupModify(ByVal group As VcGanttLib.VcGroup, _
                                   ByVal modificationType As _
                                   VcGanttLib.GroupModificationTypeEnum, _
                                   returnStatus As Variant)

    Select Case modificationType
        Case vcGMTNothing
            MsgBox "No modification"
        Case vcGMTAnything
            MsgBox "Any modification"
        Case vcGMTMinusPressed
            MsgBox "Collapsing group: " & group.Name
        Case vcGMTPlusPressed
            MsgBox "Expanding group: " & group.Name
    End Select
End Sub
```

OnGroupModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Modifizierung der Gruppe abgeschlossen ist.

	Datentyp	Beschreibung
Parameter: ⇒ Group	VcGroup	Veränderte Gruppe

Code-Beispiel

```
Private Sub VcGantt1_OnGroupModifyComplete(ByVal group As VcGanttLib.VcGroup)
    MsgBox "The group has been modified."
End Sub
```

OnGroupModifyEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Gruppe verändert. Die Daten der Gruppe vor und nach der Veränderung werden als Parameter zurückgegeben. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Veränderung verhindert.

Dieses Ereignis sollte nur verwendet werden, um Daten der aktuellen Gruppe auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnGroupModifyComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ oldgroup	VcGroup	Gruppe vor der Veränderung
⇒ group	VcGroup	Zu verändernde Gruppe
⇒ modificationType	GroupModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcGMTAnything 1 vcGMTEndModified 16 vcGMTMinusPressed 2 vcGMTMoved 32 vcGMTNothing 0 vcGMTPlusPressed 4 vcGMTStartModified 8	Änderungstyp Nicht näher bestimmt Enddatum wurde verändert Änderungstyp Minus-Symbol angeklickt Objekt wurde verschoben Änderungstyp Nichts Änderungstyp Plus-Symbol angeklickt Startdatum wurde verändert
⇔ returnStatus	Variant	Rückgabestatus

OnGroupRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender in der Tabelle auf eine Gruppenüberschrift mit der rechten Maustaste klickt. Das getroffene Group-Objekt und die Mausposition (x,y-Koordinaten) werden als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnGroupRClick(ByVal group As VcGanttLib.VcGroup, _
    ByVal x As Long, ByVal y As Long, _
```



```

returnStatus As Variant)

' Start own popup menu at the current mouse cursor position
PopupMenu mnuGroupPopup

' Suppress built-in context menu
returnStatus = vcRetStatNoPopup

End Sub

```

OnGroupsMark

Ereignis von VcGantt

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer Gruppen zum Markieren ausgewählt oder markierte Gruppen durch einen Klick in den leeren Diagrammbereich demarkiert hat. In der Gruppenuflistung (GroupCollection-Objekt) ist die beim letzten Markiervorgang ausgewählte Gruppe verzeichnet. Falls durch einen Klick ins leere Diagramm demarkiert wurde, ist die GroupCollection leer.

Wenn Sie den Rückgabestatus auf **vcRetStatFalse** setzen, muss das Markieren oder Demarkieren selbst übernommen werden.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnGroupsMarkComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ groupCollection	VcGroupCollection	GroupCollection, die die vom Anwender selektierten Gruppen enthält. Wenn in das Diagramm geklickt wurde, ist die GroupCollection leer.
⇒ button	Integer	Zahl, die angibt, auf welche Weise markiert wurde: 0: über die Tastatur, 1: linke Maustaste, 2: rechte Maustaste, 4: mittlere Maustaste
	Mögliche Werte:	Datenfeldindex
⇒ shift	Integer	Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen 1 , 2 und 4 entsprechen. Der Parameter shift zeigt den Status dieser Tasten an; es können einige, alle oder keine der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise die <STRG>- und die <ALT>-Tasten gedrückt werden, ist der Wert von shift "6".
	Mögliche Werte:	

↔ returnStatus	Variant	Datenfeldindex Rückgabestatus
----------------	---------	----------------------------------

Code-Beispiel

```
Private Sub VcGantt1_OnNodesMarkEx(ByVal GroupCollection As _
    VcGanttLib.VcGroupCollection, ByVal button _
    As Integer, ByVal shift As Integer, _
    returnStatus As Variant)

If MsgBox("Mark this group?", vbYesNo, "Marking groups") = _
    vbNo Then returnStatus = vcRetStatFalse

End Sub
```

OnGroupsMarkComplete**Ereignis von VcGantt**

Mit diesem Ereignis wird das Ende einer Markier- oder Demarkieroperation von Gruppen angezeigt.

	Datentyp	Beschreibung
Parameter: ↔ (no parameter)		Kein Parameter

Code-Beispiel

```
Private Sub VcGantt1_OnGroupsMarkComplete()
    MsgBox "Groups have been marked successfully."
End Sub
```

OnHelpRequested**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender in einem zur Laufzeit angezeigten Dialog die **F1**-Taste drückt. Die Applikation erhält damit die Möglichkeit, ihr eigenes Hilfesystem aufzurufen, um dialog- und anwendungsbezogene Hilfe anbieten zu können.

	Datentyp	Beschreibung
Parameter: ↔ dialogType	DialogTypeEnum	Dialog, für den Hilfe angefordert wird
	Mögliche Werte:	
	vcEditDataRecordDialog 5400	Hilfe wurde für den Datensatz bearbeiten Dialog angefordert
	vcEditTimeScaleDialog 5409	Hilfe wurde für den Zeitskala bearbeiten Dialog angefordert
	vcPageSetupDialog 4097	Hilfe wurde für den Seite einrichten Dialog angefordert

vcPrintPreviewDialog 4096

Hilfe wurde für den **Druckvorschau** Dialog
angefordert

OnHistogramLClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf ein Histogramm klickt. Das getroffene Histogramm-Objekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ histogram	VcHistogram	Getroffenes Histogramm
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnHistogramLClick(ByVal Histogram As _
    VcGanttLib.VcHistogram, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant) _

    Call MsgBox("Histogram: " & Histogram.Name & vbCrLf & _
        "x: " & x & vbCrLf & "y: " & y)

End Sub
```

OnHistogramLDbClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf ein Histogramm doppelklickt. Das getroffene Histogramm-Objekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ histogram	VcHistogram	Getroffenes Histogramm
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

⇒ returnStatus	Variant	Rückgabestatus
----------------	---------	----------------

Code-Beispiel

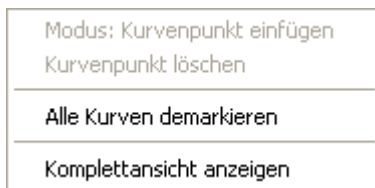
```
Private Sub VcGantt1_OnHistogramLDb1Click(ByVal Histogram As _
                                         VcGanttLib.VcHistogram, ByVal x As Long, _
                                         ByVal y As Long, returnStatus As Variant)

    MsgBox Histogram.Name

End Sub
```

OnHistogramRClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf ein Histogramm klickt. Das getroffene Histogramm-Objekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter übergeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

*Oben: integriertes Kontextmenü*

	Datentyp	Beschreibung
Parameter:		
⇒ histogram	VcHistogram	Getroffenes Histogramm
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇒ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnHistogramRClick(ByVal Histogram As _
                                         VcGanttLib.VcHistogram, ByVal x As Long, _
                                         ByVal y As Long, returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuHistogramPopup
```

```

' Suppress built-in context menu
returnStatus = vcRetStatNoPopup

End Sub

```

OnHistogramsHeight

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv das Höhenverhältnis zwischen Diagrammbereich und dem oder den Histogrammen verändert hat. Die Auflistung der Histogramme und das Diagramm / Histogramm-Höhenverhältnis werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus auf vcRetStatFalse wird die Änderung rückgängig gemacht.

	Datentyp	Beschreibung
Parameter:		
⇒ histogramCollection	VcHistogramCollection	Histogramm-Kollektion
⇒ histogramsHeightRatio	Long	Höhenverhältnis der Histogramme zum Gesamtdiagramm
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Höhe wird nicht geändert. Die Höhe wird geändert.

Code-Beispiel

```

Private Sub VcGantt1_OnHistogramsHeight(ByVal HistogramCollection As
VcGanttLib.VcHistogramCollection, ByVal histogramsHeightRatio As Long,
returnStatus As Variant)

    If histogramsHeightRatio > 30 Then
        returnStatus = vcRetStatFalse
        VcGantt1.DiagramHistogramHeightRatio = 30
    End If

End Sub

```

OnHistogramsHeightChanged

Ereignis von VcGantt

Dieses Ereignis tritt ein, nachdem das interaktiv veränderte Höhenverhältnis zwischen Diagrammbereich und dem oder den Histogrammen verändert wurde. Das Collection-Objekt der Histogramme und das Diagramm / Histogramm-Höhenverhältnis werden als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ histogramCollection	VcHistogramCollection	Histogramm-Kollektion
⇒ histogramsHeightRatio	Long	Höhenverhältnis der Histogramme zum Gesamtdiagramm

Code-Beispiel

```
Private Sub VcGantt1_OnHistogramsHeightChanged(ByVal HistogramCollection As
VcGanttLib.VcHistogramCollection, ByVal histogramsHeightRatio As Long)
```

```
    VcGantt1.FitHistogramsIntoView
```

```
End Sub
```

OnHistogramsHeightModifyEx**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender die Histogrammhöhe interaktiv verändert hat. Das Histogramm und das neue Diagramm/Histogramm-Höhenverhältnis werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

Dieses Ereignis unterscheidet sich von dem Ereignis **OnHistogramHeight** durch die Rückgabe des Parameters Diagramm/Histogramm-Höhenverhältnis als Datentyp "Double", mit dem eine höhere Genauigkeit erzielt wird. Die Verwendung dieses Ereignisses muss zuvor über die Eigenschaft **UseHigherDiagramHistogramHeightRatioPrecision** oder über die Option **Höhere Genauigkeit für die Höhenverhältnisse zwischen Diagramm und Histogramm** auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ histogramCollection	VcHistogramCollection	histogramCollection As VcHistogramCollection, histogramsHeightRatio As Double, ByRef returnStatus As Variant
⇒ histogramHeightRatio	Double	Höhenverhältnis des Gesamtdiagramms (einschließlich Histogramm) zur Höhe des Histogramms
⇔ returnStatus	Variant	Rückgabestatus

OnInsertCurvePoint

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs im Histogramm den Einfügemodus eingeschaltet und dann mit einem Klick der linken Maustaste auf eine Histogrammkurve einen Kurvenpunkt in eine über die API gesetzte Histogrammkurve eingefügt hat. Die getroffene Histogrammkurve, das Datum und der y-Wert des eingefügten Kurvenpunkts werden als Parameter zurückgegeben, so dass beispielsweise noch eine Überprüfung vorgenommen werden kann. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird das Einfügen des Kurvenpunkts wieder rückgängig gemacht.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ pointDate	Date	Datum des eingefügten Kurvenpunkts
⇒ value	Long	Y-Wert des eingefügten Kurvenpunkts
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnInsertCurvePoint(ByVal curve As VcGanttLib.VcCurve, _
                                       ByVal pointDate As Date, ByVal value As Long, _
                                       returnStatus As Variant)
    If MsgBox("Do you want to insert this curve point (date:" & pointDate & ", _
              y value: " & value & ")?", vbYesNo, _
              "Deleting curve point") = vbNo Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnInsertCurvePointEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs im Histogramm den Einfügemodus eingeschaltet und dann mit einem Klick der linken Maustaste auf eine Histogrammkurve einen Kurvenpunkt in eine über die API gesetzte Histogrammkurve eingefügt hat. Die getroffene Histogrammkurve, das Datum und der y-Wert des eingefügten Kurvenpunkts werden als Parameter zurückgegeben, so dass beispielsweise noch eine Überprüfung vorgenommen werden kann. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird das Einfügen des Kurvenpunkts wieder rückgängig gemacht.

Hinweis: Im Vergleich zum Ereignis **OnInsertCurvePointEx** kann hier der Parameter **value** als Fließkommazahl übergeben werden.

	Datentyp	Beschreibung
Parameter:		
⇒ curve	VcCurve	Getroffene Histogrammkurve
⇒ pointDate	Date	Datum des eingefügten Kurvenpunkts
⇒ value	Double	Y-Wert des eingefügten Kurvenpunkts
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnInsertCurvePoint(ByVal curve As VcGanttLib.VcCurve, _
    ByVal pointDate As Date, ByVal value As Long, _
    returnStatus As Variant)
    If MsgBox("Do you want to insert this curve point (date:" & pointDate & ", _
        y value: " & value & ")?", vbYesNo, _
        "Deleting curve point") = vbNo Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnInteractionEndComplete

Ereignis von VcGantt

Dieses Ereignis tritt im Rahmen des LiveUpdate ein, sobald eine Interaktion abgeschlossen wird.

	Datentyp	Beschreibung
Parameter:		
⇒ InInteractionMode	InInteractionModeEnum	Art der Interaktion
	Mögliche Werte:	
	vclIMCopyMoveNode 1014	Kopierter Knoten wird bewegt
	vclIMCopyNode 1007	Knoten wird kopiert
	vclIMCreateLinkChangeSuccessor 1101	Modus Verbindung erzeugen: Nachfolger wird gewechselt
	vclIMCreateNodeResizeRightX 1012	Modus Knoten erzeugen: Starttermin eines Layers
	vclIMCreateResizeObjectContainerWidthHight 1072	Größe einer Textbox verändern
	vclIMDragDropNode 1018	Knoten durch Drag and Drop verschieben
	vclIMDragDropNodeInTable 1019	Knoten in Tabelle durch Drag and Drop verschieben
	vclIMModifySectionStartDate 1061	Startdatum des Zeitskalenabschnitts verändern
	vclIMMoveCurvePointX 1052	Verschieben von Kurvenpunkt X

	vcIIMMoveCurvePointXandY 1051	Verschieben von Kurvenpunkten X und Y
	vcIIMMoveCurvePointY 1053	Verschieben von Kurvenpunkt Y
	vcIIMMoveGroupInDiagram 1100	Gruppe im Diagramm wird verschoben
	vcIIMMoveGroupInTable 1009	Gruppe in Tabelle wird verschoben
	vcIIMMoveHorValueLine 1031	Horizontales Verschieben einer Dateline
	vcIIMMoveLayer 1004	Verschieben eines Layers
	vcIIMMoveNode 1001	Verschieben eines Knotens
	vcIIMMoveNodeInRow 1002	Knoten in Zeile verschieben
	vcIIMMoveNodeInTable 1008	Knoten in Tabelle verschieben
	vcIIMMoveNodeVertical 1003	Knoten vertikal verschieben
	vcIIMMoveObjectContainer 1073	Bewegen einer Textbox
	vcIIMMoveSash 1026	Sash verschieben
	vcIIMResizeBasicUnitWidth 1062	Ändern der Grundeinheitbreite
	vcIIMResizeLeftX 1005	Starttermin eines Layers
	vcIIMResizeNumericBasicUnitWidth 1063	Ändern der numerischen Grundeinheitbreite
	vcIIMResizeObjectContainerHeight 1075	Höhe einer Textbox ändern
	vcIIMResizeObjectContainerWidth 1074	Breite einer Textbox ändern
	vcIIMResizeObjectContainerWidthHeight 1076	Höhe und Breite einer Textbox ändern
	vcIIMResizeRightX 1006	Endtermin eines Layers
	vcIIMUnknown -1	Wird i.d.R. nicht über die Eventargs zurückgegeben, kann aber z.B. genutzt werden, um eine Variable als nicht gesetzt zu kennzeichnen
	vcIIMvcIIMResizeLeftTableColumnWidth 1041	Spaltenbreite der linken Tabelle ändern
	vcIIMvcIIMResizeRightTableColumnWidth 1042	Spaltenbreite der rechten Tabelle ändern
⇒ InteractionObject	InteractionObject	von der Interaktion betroffenes Objekt
⇒ ObjectType	InteractionType	Typ des von der Interaktion betroffenen Objekts

OnInteractionModeChange

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Interaktionsmodus im Kontextmenü geändert wurde.

Durch Setzen des Rückgabewerts auf **vcRetStatFalse** wird die Änderung nicht übernommen und das Event **OnInteractionModeChangeComplete** nicht ausgelöst.

	Datentyp	Beschreibung
Parameter:		
⇒ NewInteractionMode	InteractionModeEnum	Ausgewählter Interaktionsmodus
	Mögliche Werte: vcCreateBox 36 vcCreateLink 4 vcCreateNode 2 vcDeleteLink 5 vcDeleteNode 3 vcPanning 6 vcPointer 0	Erzeugemodus für Boxen Erzeugemodus für Verbindungen Erzeugemodus für Knoten Löschmodus für Verbindungen Löschmodus für Knoten Verschiebemodus Selektiermodus
⇒ returnstatus	returnstatus	Rückgabewert

OnInteractionModeChangeComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Interaktionsmodus im Kontextmenü geändert wurde und die Änderung nicht durch Setzen des Rückgabewerts auf **vcRetStatFalse** im Ereignis **OnInteractionModeChange** verhindert wurde.

	Datentyp	Beschreibung

OnInteractionObjectChangingComplete

Ereignis von VcGantt

Dieses Ereignis tritt im Rahmen des LiveUpdate ein, wenn bei einer Interaktion initial noch kein Objekt zur Verfügung steht (z.B. Knoten- oder Boxerzeugung) und sobald intern das Objekt erzeugt wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ InInteractionMode	InInteractionModeEnum	Art der Interaktion
	Mögliche Werte: vcIIMCopyMoveNode 1014 vcIIMCopyNode 1007 vcIIMCreateLinkChangeSuccessor 1101 vcIIMCreateNodeResizeRightX 1012 vcIIMCreateResizeObjectContainerWidthHight 1072	Kopierter Knoten wird bewegt Knoten wird kopiert Modus Verbindung erzeugen: Nachfolger wird gewechselt Modus Knoten erzeugen: Starttermin eines Layers Größe einer Textbox verändern

	<code>vcIIMDragDropNode</code> 1018	Knoten durch Drag and Drop verschieben
	<code>vcIIMDragDropNodeInTable</code> 1019	Knoten in Tabelle durch Drag and Drop verschieben
	<code>vcIIMModifySectionStartDate</code> 1061	Startdatum des Zeitskalenabschnitts verändern
	<code>vcIIMMoveCurvePointX</code> 1052	Verschieben von Kurvenpunkt X
	<code>vcIIMMoveCurvePointXandY</code> 1051	Verschieben von Kurvenpunkten X und Y
	<code>vcIIMMoveCurvePointY</code> 1053	Verschieben von Kurvenpunkt Y
	<code>vcIIMMoveGroupInDiagram</code> 1100	Gruppe im Diagramm wird verschoben
	<code>vcIIMMoveGroupInTable</code> 1009	Gruppe in Tabelle wird verschoben
	<code>vcIIMMoveHorValueLine</code> 1031	Horizontales Verschieben einer Dateline
	<code>vcIIMMoveLayer</code> 1004	Verschieben eines Layers
	<code>vcIIMMoveNode</code> 1001	Verschieben eines Knotens
	<code>vcIIMMoveNodeInRow</code> 1002	Knoten in Zeile verschieben
	<code>vcIIMMoveNodeInTable</code> 1008	Knoten in Tabelle verschieben
	<code>vcIIMMoveNodeVertical</code> 1003	Knoten vertikal verschieben
	<code>vcIIMMoveObjectContainer</code> 1073	Bewegen einer Textbox
	<code>vcIIMMoveSash</code> 1026	Sash verschieben
	<code>vcIIMResizeBasicUnitWidth</code> 1062	Ändern der Grundeinheitenbreite
	<code>vcIIMResizeLeftX</code> 1005	Starttermin eines Layers
	<code>vcIIMResizeNumericBasicUnitWidth</code> 1063	Ändern der numerischen Grundeinheitenbreite
	<code>vcIIMResizeObjectContainerHeight</code> 1075	Höhe einer Textbox ändern
	<code>vcIIMResizeObjectContainerWidth</code> 1074	Breite einer Textbox ändern
	<code>vcIIMResizeObjectContainerWidthHeight</code> 1076	Höhe und Breite einer Textbox ändern
	<code>vcIIMResizeRightX</code> 1006	Endtermin eines Layers
	<code>vcIIMUnknown</code> -1	Wird i.d.R. nicht über die Eventargs zurückgegeben, kann aber z.B. genutzt werden, um eine Variable als nicht gesetzt zu kennzeichnen
	<code>vcIIMvcIIMResizeLeftTableColumnWidth</code> 1041	Spaltenbreite der linken Tabelle ändern
	<code>vcIIMvcIIMResizeRightTableColumnWidth</code> 1042	Spaltenbreite der rechten Tabelle ändern
⇒ InteractionObject	InteractionObject	von der Interaktion betroffenes Objekt
⇒ ObjectType	InteractionType	Typ des von der Interaktion betroffenen Objekts

OnInteractionStartComplete

Ereignis von VcGantt

Dieses Ereignis tritt im Rahmen des LiveUpdate ein, sobald bei einer Interaktion die linke Maustaste gedrückt wird und liefert Informationen darüber, auf welchem Objekt (Objekt und Objekttyp) der Mauszeiger steht.

	Datentyp	Beschreibung
Parameter:		
⇒ InInteractionMode	InInteractionModeEnum	Art der Interaktion
	Mögliche Werte: vclIMCopyMoveNode 1014 vclIMCopyNode 1007 vclIMCreateLinkChangeSuccessor 1101 vclIMCreateNodeResizeRightX 1012 vclIMCreateResizeObjectContainerWidthHight 1072 vclIMDragDropNode 1018 vclIMDragDropNodeInTable 1019 vclIMModifySectionStartDate 1061 vclIMMoveCurvePointX 1052 vclIMMoveCurvePointXandY 1051 vclIMMoveCurvePointY 1053 vclIMMoveGroupInDiagram 1100 vclIMMoveGroupInTable 1009 vclIMMoveHorValueLine 1031 vclIMMoveLayer 1004 vclIMMoveNode 1001 vclIMMoveNodeInRow 1002 vclIMMoveNodeInTable 1008 vclIMMoveNodeVertical 1003 vclIMMoveObjectContainer 1073 vclIMMoveSash 1026 vclIMResizeBasicUnitWidth 1062 vclIMResizeLeftX 1005 vclIMResizeNumericBasicUnitWidth 1063 vclIMResizeObjectContainerHeight 1075 vclIMResizeObjectContainerWidth 1074 vclIMResizeObjectContainerWidthHeight 1076 vclIMResizeRightX 1006	Kopierter Knoten wird bewegt Knoten wird kopiert Modus Verbindung erzeugen: Nachfolger wird gewechselt Modus Knoten erzeugen: Starttermin eines Layers Größe einer Textbox verändern Knoten durch Drag and Drop verschieben Knoten in Tabelle durch Drag and Drop verschieben Startdatum des Zeitskalenabschnitts verändern Verschieben von Kurvenpunkt X Verschieben von Kurvenpunkten X und Y Verschieben von Kurvenpunkt Y Gruppe im Diagramm wird verschoben Gruppe in Tabelle wird verschoben Horizontales Verschieben einer Dateline Verschieben eines Layers Verschieben eines Knotens Knoten in Zeile verschieben Knoten in Tabelle verschieben Knoten vertikal verschieben Bewegen einer Textbox Sash verschieben Ändern der Grundeinheitenbreite Starttermin eines Layers Ändern der numerischen Grundeinheitenbreite Höhe eine Textbox ändern Breite einer Textbox ändern Höhe und Breite einer Textbox ändern Endtermin eines Layers

	vcIIMUnKnown -1	Wird i.d.R. nicht über die Eventargs zurückgegeben, kann aber z.B. genutzt werden, um eine Variable als nicht gesetzt zu kennzeichnen
	vcIIMvcIIMResizeLeftTableColumnWidth 1041	Spaltenbreite der linken Tabelle ändern
	vcIIMvcIIMResizeRightTableColumnWidth 1042	Spaltenbreite der rechten Tabelle ändern
⇒ InteractionObject	InteractionObject	von der Interaktion betroffenes Objekt
⇒ ObjectType	InteractionType	Typ des von der Interaktion betroffenen Objekts

OnLegendViewClosed

Ereignis von VcGantt

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Legendenansicht geschlossen wird.

	Datentyp	Beschreibung
Parameter: ⇐ (no parameter)		

Code-Beispiel

```
Private Sub VcGantt1_OnLegendViewClosed()
    MsgBox "Do you want to close the legend view window?", vbOKCancel
End Sub
```

OnLinkCreate

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Verbindung zwischen zwei Knoten erzeugt hat. Das neu erzeugte Verbindungs-Objekt wird als Parameter zurückgegeben, so dass eine Validierung und ggf. ein Datenbank-Eintrag vorgenommen werden kann.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnLinkCreateComplete**.

Durch Setzen des Rückgabestatus kann die Anlage verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	Angelegte Verbindung
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Verbindung wird nicht angelegt. Die Verbindung wird angelegt.

Code-Beispiel

```
Private Sub VcGantt1_OnLinkCreate(ByVal link As VcGanttLib.VcLink, _
                                returnStatus As Variant)

    ' Show own "Edit" dialog for the new link
    On Error GoTo CancelError
    frmEditDialog.Show Modal, Me

    ' create a record in the underlying database of the application
    addDataRecord link.AllData

    Exit Sub

CancelError:
    returnStatus = vcRetStatFalse

End Sub
```

OnLinkCreateComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das interaktive Anlegen einer Verbindung zwischen zwei Knoten beendet ist. Das Verbindungs-Objekt, der Typ des Anlegens der Verbindung (hier immer **VcLinkCreated**) und die Information, ob die angelegte Verbindung die einzige Verbindung oder die letzte Verbindung einer Menge ist (derzeit immer **True**), werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	Angelegte Verbindung
⇒ creationType	CreationTypeEnum	Typ des Anlegens von Knoten bzw. Verbindungen
	Mögliche Werte: vcLinkCreated 2	Verbindung durch das Verbinden zweier Knoten angelegt
⇒ isLastLinkInSeries	Boolean	angelegte Verbindung ist/ist nicht die einzige bzw. die letzte Verbindung einer Menge
	Mögliche Werte: Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar	

Code-Beispiel

```
Private Sub VcGantt1_OnLinkCreateComplete(ByVal link As VcGanttLib.VcLink, _
                                         ByVal creationType As _
                                         VcGanttLib.CreationTypeEnum, _
                                         ByVal isLastLinkInSeries As Boolean)

    'create a record in the underlying database of the application
    addDataRecord link.AllData

End Sub
```

OnLinkDelete**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs eine Verbindung löscht. Die betroffene Verbindung wird als Parameter zurückgegeben, so dass Sie z. B. noch eine Überprüfung vornehmen und bei negativem Ergebnis dieser Prüfung die Löschung ggf. durch Setzen des Rückgabestatus verhindern können.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	Gelöschte Verbindung
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Verbindung wird nicht gelöscht. Die Verbindung wird gelöscht.

Code-Beispiel

```
Private Sub VcGantt1_OnLinkDelete(ByVal link As VcGanttLib.VcLink, _
                                   returnStatus As Variant)
    ' deny deletion of link with a certain predecessor
    If link.PredecessorNode.DataField(0) = "1" Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnLinkDeleteComplete**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn das Löschen einer Verbindung beendet ist. Die getroffene Verbindung und die Information, ob die angelegte Verbindung die einzige Verbindung oder die letzte Verbindung einer Menge ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	gelöschte Verbindung
⇒ isLastLinkInSeries	Boolean	Gelöschte Verbindung ist/ist nicht die einzige bzw. die letzte Verbindung einer Menge.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub VcGantt1_OnLinkDeleteComplete(ByVal link As VcGanttLib.VcLink, ByVal
isLastLinkInSeries As Boolean)

    MsgBox "The link " & link.AllData & " was deleted."

End Sub
```

OnLinkLClickCltn**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Verbindung oder eine Gruppe sich überlagernder Verbindungen klickt. Das getroffene LinkCollection-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCltn	VcLinkCollection	getroffenes LinkCollection-Objekt
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnLinkLClickCltn(ByVal linkCltn _
    As VcGanttLib.VcLinkCollection, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    Dim link As VcLink

    ' set certain data field of all links
    For Each link In linkCltn
        link.DataField(2) = "A"
    Next
End Sub
```


OnLinkLDblickCltn

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Verbindung oder eine Gruppe sich überlagernder Verbindungen doppelklickt. Das getroffene LinkCollection-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCltn	VcLinkCollection	getroffenes LinkCollection-Objekt
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnLinkLDblickCltn(ByVal linkCltn As _
                                         VcGanttLib.VcLinkCollection, _
                                         ByVal x As Long, ByVal y As Long, _
                                         returnStatus As Variant)

    ' Edit link in own dialog
    If linkCltn.Count = 1 Then
        On Error GoTo CancelError
        frmEditLinkDialog.setLink linkCltn.FirstLink
        frmEditLinkDialog.Show Modal, Me
    End If

    ' Deny the "Edit Link Data" dialog
    CancelError:
    returnStatus = vcRetStatFalse
End Sub
```

OnLinkRClickCltn

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Verbindung oder eine Gruppe sich überlagernder Verbindungen klickt. Das getroffene Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCltn	VcLinkCollection	Getroffenes Verbindungsauflistung

⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1		Das Kontextmenü wird unterdrückt. Das Kontext-Menü erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnLinkRClickCltn(ByVal linkCltn As _
                                VcGanttLib.VcLinkCollection, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)
    ' Start a popup menu at the current mouse cursor position
    PopupMenu mnuLinkPopup

    ' Suppress the built-in context menu
    returnStatus = vcRetStatNoPopup
End Sub
```

OnModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt immer dann auf, wenn Daten interaktiv im Chart verändert werden, also explizit nach den folgenden Ereignissen:

- OnBoxModifyComplete
- OnCurveModifyEx
- OnDeleteCurvePoint
- OnGroupModifyComplete
- OnLinkCreateComplete
- OnLinkDeleteComplete
- OnNodeCreateCompleteEx
- OnNodeDelete
- OnNodeModifyComplete

Mit diesem Ereignis ist es möglich, sich im Anwenderprogramm eine Marke zu setzen, die daran erinnert, dass beim Beenden des Programms die Daten noch gespeichert werden müssen.

	Datentyp	Beschreibung
Parameter: ⇐ (no parameter)		kein Parameter

OnMouseDown

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender eine Maustaste doppelklickt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter: ⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	Datenfeldindex
⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseDown

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf eine Maustaste klickt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	Datenfeldindex
⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseMove

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender die Maus bewegt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	Datenfeldindex

⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseUp

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender eine gedrückte Maustaste wieder loslässt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
	Mögliche Werte:	Datenfeldindex
⇒ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
	Mögliche Werte:	Datenfeldindex
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnNodeCreate

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten erzeugt. Das Knotenobjekt wird als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeCreateCompleteEx**.

Durch Setzen des Rückgabestatus kann die Anlage verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Anzulegender Knoten
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Knoten wird nicht angelegt. Der Knoten wird angelegt.

Code-Beispiel

```
Private Sub VcGantt1_OnNodeCreate(ByVal node As VcGanttLib.VcNode, _
                                returnStatus As Variant)

    'Show own "Edit" dialog for the new node
    '(EditNewNodes attribute must be set to off!)
    On Error GoTo CancelError
    frmEditDialog.Show Modal, Me

    'create a record in the underlying database of the application
    addDataRecord node.AllData

    Exit Sub

CancelError:
    returnStatus = vcRetStatFalse

End Sub
```

OnNodeCreateCompleteEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das interaktive Anlegen eines Knotens abgeschlossen ist. Das Knotenobjekt, der Typ des Knotenanlegens (hier **vcNodeCreated**) und die Information, ob der angelegte Knoten der einzige Knoten oder der letzte Knoten einer Menge ist (derzeit immer **True**), werden

als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Angelegter Knoten
⇒ creationType	CreationTypeEnum	Typ des Anlegens von Knoten bzw. Verbindungen
	Mögliche Werte: vcDataRecordCreated 6 vcDataRecordCreatedByResourceScheduling 5 vcNodeCreated 1	Datensatz wurde durch Interaktion angelegt Datensatz wurde automatisch durch Ressourcenplanung angelegt Knoten durch "Stempeln" angelegt
⇒ isLastNodeInSeries	Boolean	Angelegter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub VcGantt1_OnNodeCreateCompleteEx(ByVal node As _
    VcGanttLib.VcNode, ByVal creationType As _
    VcGanttLib.CreationTypeEnum, _
    ByVal isLastNodeInSeries As Boolean)
    'create a record in the underlying database of the application
    addDataRecord node.AllData
End Sub
```

OnNodeDelete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs einen Knoten löscht. Der betroffene Knoten wird als Parameter zurückgegeben, so dass Sie z. B. noch eine Überprüfung vornehmen und bei negativem Ergebnis dieser Prüfung die Löschung ggf. durch Setzen des Rückgabestatus verhindern können.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Gelöschter Knoten
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0	Der Knoten wird nicht gelöscht.

vcRetStatOK 1

Der Knoten wird gelöscht.

Code-Beispiel

```

Private Sub VcGantt1_OnNodeDelete(ByVal node As VcGanttLib.VcNode, _
                                returnStatus As Variant)
    'deny the deletion of the last node in the chart
    If VcGantt1.NodeCollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last node in the chart cannot be deleted.")
    End If
End Sub

```

OnNodeDeleteCompleteEx**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn das interaktive Löschen eines Knotens abgeschlossen ist. Das Knotenobjekt und die Information, ob der gelöschte Knoten der zuletzt gelöschte einer Menge ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Gelöschter Knoten
⇒ isLastNodeInSeries	Boolean	Gelöschter Knoten ist (True) /ist nicht (False) letzte Knoten einer Menge
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Private Sub VcGantt1_OnNodeDeleteCompleteEx(ByVal node As VcGanttLib.VcNode,
ByVal isLastNodeInSeries As Boolean)

    MsgBox "The node " & node.AllData & " was deleted."

End Sub

```

OnNodeLClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten (location = vcInDiagram) oder auf einen vorgangsbezogenen Tabelleneintrag (location = vcInTable) mit der linken Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Cursorposition (x,y-Koordinaten) als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Getroffener Knoten
⇒ location	LocationEnum	Lokalisierung im Diagramm
	Mögliche Werte: vcInDiagram 1 vcInTable 0	im Knotenbereich im Tabellenbereich
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnNodeLClick(ByVal node As VcGanttLib.VcNode, _
                                ByVal location As VcGanttLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)
    'change data field of the node
    node.DataField(8) = 1 - CInt(node.DataField(8))
End Sub
```

OnNodeLDbIClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten (location = vcInDiagram) oder auf einen vorgangsbezogenen Tabelleneintrag (location = vcInTable) mit der linken Maustaste doppelt klickt. Das getroffene Knotenobjekt wird zusammen mit der Cursorposition (x,y-Koordinaten) als Parameter übergeben. Durch Setzen des Rückgabestatus können Sie das Erscheinen des Dialogfelds **Vorgänge bearbeiten** unterdrücken.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Getroffener Knoten
⇒ location	LocationEnum	Lokalisierung im Diagramm
	Mögliche Werte: vcInDiagram 1 vcInTable 0	im Knotenbereich im Tabellenbereich
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Daten bearbeiten -Dialog erscheint nicht. Der Daten bearbeiten -Dialog erscheint.

Code-Beispiel

```

Private Sub VcGantt1_OnNodeLDb1Click(ByVal node As VcGanttLib.VcNode, _
                                     ByVal location As VcGanttLib.LocationEnum, _
                                     ByVal x As Long, ByVal y As Long, _
                                     returnStatus As Variant)

    ' Show own "Edit Node" dialog
    On Error GoTo CancelError
    frmEditDialog.setNode node
    frmEditDialog.Show Modal, Me

    returnStatus = vcRetStatFalse

Exit Sub

CancelError:
    returnStatus = vcRetStatFalse

End Sub

```

OnNodeModifyComplete**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn die Modifizierung des angegebenen Knotens abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Veränderter Knoten
⇒ isLastNodeInSeries	Boolean	Veränderter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Private Sub VcGantt1_OnNodeModifyComplete(ByVal node As _
                                           VcGanttLib.VcNode, ByVal _
                                           isLastNodeInSeries As Boolean)

    ' Modify a record in the underlying database of the application
    modifyDataRecord node.AllData

End Sub

```

OnNodeModifyCompleteEx**Ereignis von VcGantt**

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer den Knoten verändert hat.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	veränderter Knoten
⇒ isLastNodeInSeries	Boolean	veränderter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
⇒ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcAnything 1 vcChangedGroup 16 vcEndModified 4 vcHierarchyModified 64 vcModifiedByResourceScheduling 128 vcModifiedBySchedule 32 vcMoved 8 vcNothing 0 vcStartModified 2	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Ende des Knotens wurde verändert Hierarchie der Knoten wurde verändert Änderung durch Ressourcenplanung (nur für Datensatz-Objekt) Änderung durch neue Zeitberechnung Objekt wurde verschoben keine Änderung Anfang des Knotens wurde verändert

Code-Beispiel

```
Private Sub VcGantt1_OnNodeModifyCompleteEx(ByVal node As VcGanttLib.VcNode,
ByVal modificationType As VcGanttLib.ModificationTypeEnum, ByVal
isLastNodeInSeries As Boolean)
```

```
    Select Case modificationType
        Case ModificationTypeEnum.vcMoved
            node.MarkNode = True
    End Select
```

```
End Sub
```

OnNodeModifyEx**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten verändert. Dabei kann der Balken verlängert oder verschoben oder ein Wert im Dialogfeld **Vorgänge bearbeiten** verändert worden sein. Die Daten des Knotens vor und nach der Veränderung werden als Parameter zurückgegeben. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten nicht verändert.

Dieses Ereignis sollte nur verwendet werden, um Daten des aktuellen Knotens auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeModifyComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ oldNode	VcNode	Knoten vor der Veränderung
⇒ node	VcNode	Zu verändernder Knoten
⇒ modificationType	ModificationTypeEnum	Art der Veränderung (Auch eine Kombination der folgenden Werte ist möglich.)
	Mögliche Werte: vcAnything 1 vcChangedGroup 16 vcEndModified 4 vcHierarchyModified 64 vcModifiedByResourceScheduling 128 vcModifiedBySchedule 32 vcMoved 8 vcNothing 0 vcStartModified 2	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Ende des Knotens wurde verändert Hierarchie der Knoten wurde verändert Änderung durch Ressourcenplanung (nur für Datensatz-Objekt) Änderung durch neue Zeitberechnung Objekt wurde verschoben keine Änderung Anfang des Knotens wurde verändert
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

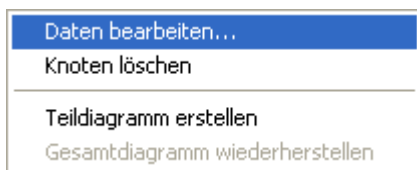
```
Private Sub VcGantt1_OnNodeModifyEx(ByVal oldNode As _
    VcGanttLib.VcNode, ByVal node As _
    VcGanttLib.VcNode, ByVal modificationType As _
    VcGanttLib.ModificationTypeEnum, returnStatus _
    As Variant)

    ' Revoke the modification if the node would change the group
    If modificationType And vcChangedGroup Then
        MsgBox "The node cannot be moved into another group."
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnNodeRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten (location = vcInDiagram) oder auf einen vorgangsbezogenen Tabelleneintrag (location = vcInTable) mit der rechten Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Cursorposition (x,y-Koordinaten) als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.



Oben: integriertes Kontextmenü

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Getroffener Knoten
⇒ location	LocationEnum	Lokalisierung im Chart
	Mögliche Werte: vcInDiagram 1 vcInTable 0	im Knotenbereich im Tabellenbereich
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontextmenü erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
                                ByVal location As VcGanttLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuNodePopup

    returnStatus = vcRetStatNoPopup
End Sub
```

OnNodeResizeStart

Ereignis von VcGantt

Dieses Ereignis wird aufgerufen, wenn der Benutzer beginnt, einen Layer an einem Knoten interaktiv zu verlängern bzw. zu verkürzen. Dies kann dazu dienen, kleinere Änderungen am XGantt vorzunehmen wie z.B. die Schrittweite knotenabhängig (TimeUnitsPerStep) anzupassen.

	Datentyp	Beschreibung
Parameter:		
⇒ layer	VcLayer	Layer, an dem gezogen wurde

OnNodesMarkComplete

Ereignis von VcGantt

Mit diesem Ereignis wird das Ende einer Markier- oder Demarkieroperation von Knoten angezeigt.

	Datentyp	Beschreibung
Parameter: ⇨ (no parameter)		Kein Parameter

Code-Beispiel

```
Private Sub VcGantt1_OnNodesMarkComplete()  
    MsgBox "Nodes have been marked successfully."  
End Sub
```

OnNodesMarkEx

Ereignis von VcGantt

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer Knoten zum Markieren ausgewählt oder markierte Knoten durch einen Klick in den leeren Diagrammbereich demarkiert hat. In der Knotenauflistung (NodeCollection-Objekt) sind die beim letzten Markiervorgang ausgewählten Knoten verzeichnet. Falls durch einen Klick ins leere Diagramm demarkiert wurde, ist die NodeCollection leer.

Die Parameter **button** und **shift** geben an, welche Steuer- und Maustasten gedrückt wurden.

Wenn Sie den Rückgabestatus auf **vcRetStatFalse** setzen, muss das Markieren oder Demarkieren selbst übernommen werden.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeMarkComplete**.

	Datentyp	Beschreibung
Parameter: ⇨ nodeCollection	VcNodeCollection	NodeCollection, die die vom Anwender selektierten Knoten enthält. Wenn in das Diagramm geklickt wurde, ist die NodeCollection leer.
⇨ button	Integer	Zahl, die angibt, auf welche Weise markiert wurde: 0: über die Tastatur, 1: linke Maustaste, 2: rechte Maustaste, 4: mittlere Maustaste
	Mögliche Werte:	

⇒ shift	Integer	Datenfeldindex Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen 1 , 2 und 4 entsprechen. Der Parameter shift zeigt den Status dieser Tasten an; es können einige, alle oder keine der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise die <STRG>- und die <ALT>-Tasten gedrückt werden, ist der Wert von shift "6".
⇔ returnStatus	Variant Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Datenfeldindex Rückgabestatus Die Markierung muss händig erfolgen. Die Markierung erfolgt automatisch.

Code-Beispiel

```
Private Sub VcGantt1_OnNodesMarkEx(ByVal NodeCollection As _
    VcGanttLib.VcNodeCollection, ByVal button _
    As Integer, ByVal shift As Integer, _
    returnStatus As Variant)

    If MsgBox("Mark this node?", vbYesNo, "Marking nodes") = _
        vbNo Then returnStatus = vcRetStatFalse

End Sub
```

OnNumericScaleLClick**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf die numerische Skala klickt. Das getroffene NumericScale-Objekt wird zusammen mit der Cursorposition (x,y-Koordinaten) des Cursors als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ numericScale	VcNumericScale	Getroffene Werteskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnNumericScaleLClick(ByVal numericScale As _
    VcGanttLib.VcNumericScale, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant)
```

```

        numericScale.BackgroundColor = RGB(222, 211, 33)
End Sub

```

OnNumericScaleLDbClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf die numerische Skala doppelt klickt. Das getroffene NumericScale-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ numericScale	VcNumericScale	Getroffene Werteskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

Private Sub VcGantt1_OnNumericScaleLDbClick(ByVal numericScale As _
                                           VcGanttLib.VcNumericScale, ByVal x As Long, _
                                           ByVal y As Long, returnStatus As Variant)

    numericScale.MajorTicks = Text1.Text
End Sub

```

OnNumericScaleRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf die numerische Skala klickt. Das getroffene NumericScale-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können an der entsprechenden Position Ihr eigenes gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ numericScale	VcNumericScale	Getroffene Werteskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Mögliche Werte:

vcRetStatNoPopup 4
vcRetStatOK 1

Das Kontextmenü wird unterdrückt.
Das Kontextmenü erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnNumericScaleRClick(ByVal numericScale As _  
                                         VcGanttLib.VcNumericScale, ByVal x As Long, _  
                                         ByVal y As Long, returnStatus As Variant)  
  
    If MsgBox("Change unit label of numeric scale?", vbYesNo) = vbYes Then  
        numericScale.UnitLabel = Text1.Text  
    End If  
  
End Sub
```

OnNumericScaleRescale

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender die numerische Skala interaktiv skaliert. Die getroffene numerische Skala und die neue Breite der Grundeinheit werden als Parameter zurückgegeben, so dass Sie die Zulässigkeit der Skalierung prüfen können. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

	Datentyp	Beschreibung
Parameter:		
⇒ numericScale	VcNumericScale	Getroffene Werteskala
⇒ newBasicUnitWidth	Long	Neue Breite der Grundeinheit
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die numerische Skala wird nicht verändert. Die numerische Skala wird verändert.

Code-Beispiel

```
Private Sub VcGantt1_OnNumericScaleRescale (ByVal numericScale As _  
                                         VcGanttLib.VcNumericScale, _  
                                         ByVal newBasicUnitWidth As Long, _  
                                         returnStatus As Variant)  
  
    Select Case newBasicUnitWidth  
    Case Is <= 1000  
        MsgBox "New basic unit width: " & newBasicUnitWidth  
    Case Is > 1000  
        MsgBox "The maximum basic unit width is 1000."  
        returnStatus = vcRetStatFalse  
    End Select  
  
End Sub
```

OnObjectDrawCompleteEx

Ereignis von VcGantt

Dieses Ereignis tritt auf, nachdem ein Objekt gezeichnet wurde. Damit können Sie durch eigenen Programmcode die durch VARCHART XGantt gezeichneten Objekte gestalterisch ergänzen.

Die ObjectDraw-Ereignisse werden nur ausgelöst, wenn dieses Verhalten für einen Objekttyp eingeschaltet wurde. Sie können sowohl Layer als auch benutzerdefinierte Skalenbeschriftungen zeichnen.

Um einen Layer zu zeichnen, setzen Sie zur Laufzeit die Eigenschaft **ObjectDrawEventsEnabled** des Objekts vom Typ **VcLayer** auf **True** oder aktivieren zur Design-Zeit auf der Eigenschaftenseite **Objekte** unter **Layer** die Option **ObjectDraw-Ereignisse** für den gewünschten Layer.

Zum Zeichnen einer benutzerdefinierten Skalenbeschriftung aktivieren Sie zur Design-Zeit im Dialog **Histogramme bearbeiten** unter **Skalenstreifen** die Option **ObjectDraw-Ereignisse** für den gewünschten Skalenstreifen.

Wenn Sie das standardmäßige Zeichnen des Layers oder der Skalenbeschriftung unterbinden und durch eigenen Programmcode ersetzen wollen, dann verwenden Sie bitte das Ereignis **OnObjectDrawEx**.

	Datentyp	Beschreibung
Parameter:		
hDC	Long	Gerätekontext
object	Object	Objekt, das gezeichnet wird
objectType	VcObjectTypeEnum	Typ des zu zeichnenden Objekts
	Mögliche Werte: vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14	Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Werteskala Objekttyp Summenbalken
subObject	Object	Unterobjekt, das kontextabhängig mitgegeben wird
subObjectType	VcObjectTypeEnum	Typ des Unterobjekts
	Mögliche Werte: vcObjTypeLayer 8	Objekttyp Layer
completeRect	VcRect	Rechteck in Gerätekoordinaten, in die das komplette Objekt gezeichnet werden soll
updateRect	VcRect	Rechteck in Gerätekoordinaten, das den Updatebereich bezeichnet. Dieser kann gleich groß wie oder kleiner als das Rechteck in completeRect sein

lineWidth	Long	Stärke einer dünnen Linie. Kann verwendet werden, um bei Zeichenbefehlen die Linienstärke der Auflösung des Gerätekontextes (Bildschirm bzw. Drucker) anzupassen.
xZoomFactor	Double	Dieser Parameter gibt den aktuellen Zoomfaktor in X-Richtung an, der eine Umrechnung von Distanzen in 1/100 Millimetern nach Pixeln und umgekehrt zulässt. Der Zoomfaktor ist jeweils auf das Ausgabegerät bezogen (also Bildschirm, Druckvorschau oder Drucker).
yZoomFactor	Double	Dieser Parameter gibt den aktuellen Zoomfaktor in Y-Richtung an, der eine Umrechnung von Distanzen in 1/100 Millimetern nach Pixeln und umgekehrt zulässt. Der Zoomfaktor ist jeweils auf das Ausgabegerät bezogen (also Bildschirm, Druckvorschau oder Drucker).

Code-Beispiel

```
Private Declare Function GetStockObject Lib "gdi32" (ByVal nIndex As Long) As Long
Private Const WHITE_BRUSH = 0
Private Const WHITE_PEN = 6
Private Declare Function FillRect Lib "user32" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Sub VcGantt1_OnObjectDrawComplete(ByVal hdc As Long, ByVal Object As Object, ByVal objectType As VcGanttLib.VcObjectTypeEnum, ByVal subObject As Object, ByVal subObjectType As VcGanttLib.VcObjectTypeEnum, ByVal completeRect As VcGanttLib.VcRect, ByVal updateRect As VcGanttLib.VcRect, ByVal xZoomFactor As Double, ByVal yZoomFactor As Double)
    Dim smallRect As RECT
    Dim hBrush As Long

    ' drawing a white square into the layer
    hBrush = GetStockObject(WHITE_BRUSH)
    smallRect.Left = completeRect.Left + 2
    smallRect.Top = completeRect.Top + 2
    smallRect.Right = completeRect.Right - 2
    smallRect.Bottom = completeRect.Bottom - 2
    FillRect hdc, smallRect, hBrush
End Sub
```

OnObjectDrawEx

Ereignis von VcGantt

Dieses Ereignis tritt auf, bevor ein Objekt gezeichnet wird. Damit können Sie das Objekt nach Ihren Wünschen durch eigenen Programmcode selber gestalten. Das nachfolgende, von der Komponente vorgesehene Zeichnen des Objekts kann durch Setzen des returnStatus verhindert werden.

Die ObjectDraw-Ereignisse werden nur ausgelöst, wenn dieses Verhalten für einen Objekttyp eingeschaltet wurde. Sie können sowohl Layer als auch benutzerdefinierte Skalenbeschriftungen zeichnen.

Um einen Layer zu zeichnen, setzen Sie zur Laufzeit die Eigenschaft **ObjectDrawEventsEnabled** des Objekts vom Typ **VcLayer** auf **True** oder aktivieren zur Design-Zeit im Dialog **Balkenaussehen festlegen** die Option **ObjectDraw-Ereignisse** für den gewünschten Layer.

Zum Zeichnen einer benutzerdefinierte Skalenbeschriftung aktivieren Sie zur Design-Zeit im Dialog **Histogramme bearbeiten** unter **Skalenstreifen** die Option **ObjectDraw-Ereignisse** für den gewünschten Skalenstreifen.

Wenn Sie einem durch VARCHART XGantt gezeichneten Layer oder einer Skalenbeschriftung noch etwas hinzufügen möchten, dann können Sie dies im Ereignis **OnObjectDrawCompleteEx** tun.

	Datentyp	Beschreibung
Parameter:		
hDC	Long	Gerätekontext
object	Object	Objekt, das gezeichnet wird
objectType	VcObjectTypeEnum	Typ des zu zeichnenden Objekts
	Mögliche Werte: vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14	Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Werteskala Objekttyp Summenbalken
subObject	Object	Unterobjekt, das kontextabhängig mitgegeben wird
subObjectType	VcObjectTypeEnum	Typ des Unterobjekts
	Mögliche Werte: vcObjTypeLayer 8	Objekttyp Layer
completeRect	VcRect	Rechteck in Gerätekoordinaten, in die das komplette Objekt gezeichnet werden soll
updateRect	VcRect	Rechteck in Gerätekoordinaten, das den Updatebereich bezeichnet. Dieser kann gleich groß wie oder kleiner als das Rechteck in completeRect sein.
lineWidth	Long	Stärke einer dünnen Linie. Kann verwendet werden, um bei Zeichenbefehlen die Linienstärke der Auflösung des Gerätekontextes (Bildschirm bzw. Drucker) anzupassen.
returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Das Objekt wird nicht gezeichnet. Das Objekt wird gezeichnet.

xZoomFactor	Double	Dieser Parameter gibt den aktuellen Zoomfaktor in X-Richtung an, der eine Umrechnung von Distanzen in 1/100 Millimetern nach Pixeln und umgekehrt zulässt. Der Zoomfaktor ist jeweils auf das Ausgabegerät bezogen (also Bildschirm, Druckvorschau oder Drucker).
yZoomFactor	Double	Dieser Parameter gibt den aktuellen Zoomfaktor in Y-Richtung an, der eine Umrechnung von Distanzen in 1/100 Millimetern nach Pixeln und umgekehrt zulässt. Der Zoomfaktor ist jeweils auf das Ausgabegerät bezogen (also Bildschirm, Druckvorschau oder Drucker).

OnOptimizeTableColumnWidth

Ereignis von VcGantt

Dieses Ereignis tritt bei einem Doppelklick auf die Trennlinie zwischen zwei Tabellenspalten ein, sofern auf der Eigenschaftenseite **Allgemeines** die Option **Tabellenspalten optimierbar** aktiviert wurde oder die Eigenschaft **AllowTableColumnWidthOptimization** gesetzt wurde. Die linke Tabellenspalte wird dann automatisch optimiert. Die Tabelle und der Index der optimierten Spalte werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Optimierung verhindert.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Tabelle
⇒ index	Integer	Index der geänderten Spalte
	Mögliche Werte:	Datenfeldindex
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Spaltenbreite wird nicht optimiert. Die Spaltenbreite wird optimiert.

Code-Beispiel

```
Private Sub VcGantt1_OnOptimizeTableColumnWidth(ByVal Table As
VcGanttLib.VcTable, ByVal Index As Integer, returnStatus As Variant)

    MsgBox "The index of the modified column is " & Index

End Sub
```

OnPreScrollComponent

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn Sie einen Scrollvorgang angefordert haben, aber bevor der integrierte Scroll-Automatismus ausgeführt wird.

Mit diesem Ereignis können Sie bei jeder interaktiven Scrollaktion Folgendes ermitteln:

1. die Komponente, die gescrollt wird (nur `vcDiagramComponent`, `vcHistogramComponent`, `vcListComponent` und `vcRightListComponent` werden als „Masterscroller“ berücksichtigt, da alle anderen Komponenten abhängig mitgescrollt werden)
2. die Bewegungsorientierung (horizontal oder vertikal)
3. die Art der Benutzeraktion.

Setzt man den Rückgabestatus auf **vcRetStatFalse**, so wird der integrierte Scroll-Automatismus unterdrückt, und Sie können in Ihrer Anwendung selbst auf das Ereignis reagieren.

Hinweis: Die tatsächliche Scrollaktion ergibt sich aus der Kombination der Parameter **orientation** und **scrollAction**, da in Windows-Programmen die up/left- und down/right-Aktionen jeweils dieselbe Nummer haben. Z. B.:

`vcScrollActionSBPageLeft = vcScrollActionSBPageUp = 2`

`vcScrollActionThumbTrackLeft = vcScrollActionThumbTrackUp = 107`

Das folgende Beispiel zeigt die Auswertung über **orientation** für **VcScrollActionSBPageLeft** und **vcScrollActionSBPageUp**, die beide den Wert 2 haben.

	Datentyp	Beschreibung
Parameter: ⇒ component	ComponentTypeEnum Mögliche Werte: vcAdditionalListComponent 1 vcBottomListTitleComponent 14 vcBottomRightListTitleComponent 17 vcBottomTimeScaleComponent 15 vcDiagramComponent 4 vcHistogramComponent 8	Komponententyp zusätzliche Tabelle untere Titelleiste Tabellenbereich unten rechts untere Zeitskala Diagramm Histogramm

	vcHistogramVerScaleComponent 7	numerische Skala (vertikale Histogramm-skala)
	vcLegendComponent 10	Legende (zur Zeit ohne Funktion; Rückgabewerte 00)
	vcListComponent 0	Tabelle
	vcListTitleComponent 2	Tabellenüberschrift
	vcRightListComponent 5	rechte Tabelle
	vcRightListTitleComponent 16	Tabellenüberschrift der rechten Tabelle
	vcTimeScaleComponent 3	obere Zeitskala
	vcTopTitleComponent 11	obere Titelleiste
⇒ Orientation	ScrollOrientationEnum	Scrollrichtung
	Mögliche Werte:	
	vcHorizontal 1	horizontales Scrollen
	vcVertical 2	vertikales Scrollen
⇒ scrollAction	ScrollActionEnum	Art des Scrollvorgangs
	Mögliche Werte:	
	vcScrollActionAutoscrollDown 102	Die Ansicht verschob sich automatisch nach unten.
	vcScrollActionAutoscrollLeft 101	Die Ansicht verschob sich automatisch nach links.
	vcScrollActionAutoscrollRight 102	Die Ansicht verschob sich automatisch nach rechts.
	vcScrollActionAutoscrollUp 101	Die Ansicht verschob sich automatisch nach oben.
	vcScrollActionMouseWheelDown 106	Bei gedrücktem Mausrad wurde die Maus nach unten geschoben.
	vcScrollActionMouseWheelLeft 105	Bei gedrücktem Mausrad wurde die Maus nach links geschoben.
	vcScrollActionMouseWheelRight 106	Bei gedrücktem Mausrad wurde die Maus nach rechts geschoben.
	vcScrollActionMouseWheelUp 105	Bei gedrücktem Mausrad wurde die Maus nach oben geschoben.
	vcScrollActionSBLineDown 1	Die Ansicht verschob sich zur unteren Begrenzungslinie.
	vcScrollActionSBLineLeft 0	Die Ansicht verschob sich zur linken Begrenzungslinie.
	vcScrollActionSBLineRight 1	Die Ansicht verschob sich zur rechten Begrenzungslinie.
	vcScrollActionSBLineUp 0	Die Ansicht verschob sich zur oberen Begrenzungslinie.
	vcScrollActionSBNothing -1	Die Ansicht wurde nicht verschoben.
	vcScrollActionSBPageDown 3	Die Ansicht wurde um eine Seite nach unten versetzt.
	vcScrollActionSBPageLeft 2	Die Ansicht wurde um eine Seite nach links versetzt.
	vcScrollActionSBPageRight 3	Die Ansicht wurde um eine Seite nach rechts versetzt.
	vcScrollActionSBPageUp 2	Die Ansicht wurde um eine Seite nach oben versetzt.
	vcScrollActionSBThumbPosition 4	Die Versetzung um einen Schritt ist abgeschlossen.
	vcScrollActionSBThumbTrack 5	Die Ansicht wurde um einen Schritt versetzt.
	vcScrollActionScrollEnd 104	Scrollen über die Taste Ende oder über das Kontextmenü an das Diagrammende (rechts unten)
	vcScrollActionScrollHome 103	Scrollen über die Taste Pos 1 oder über das Kontextmenü an den Diagrammanfang (links oben)
	vcScrollActionThumbTrackDown 108	Thumb (Balken im Scrollbar) nach unten geschoben
	vcScrollActionThumbTrackLeft 107	Thumb (Balken im Scrollbar) nach links geschoben

	vcScrollActionThumbTrackRight 108	Thumb (Balken im Scrollbar) nach rechts geschoben
	vcScrollActionThumbTrackUp 107	Thumb (Balken im Scrollbar) nach oben geschoben
⇒ delta	Long	Länge des Scrollvorgangs (in Pixeln)
⇒ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

If orientation = vcHorizontal and scrollAction = vcScrollActionSBPageLeft _
    Then MsgBox "Scrolled left" _
ElseIf orientation = vcHorizontal _
    and scrollAction = vcScrollActionSBPageRight _
    Then MsgBox "Scrolled right" _
ElseIf orientation = vcVertical and scrollAction = vcScrollActionSBPageUp _
    Then MsgBox "Scrolled up" _
End If

```

OnPreScrollDiagramHor

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn Sie einen Scrollvorgang angefordert haben, aber bevor der integrierte Scroll-Automatismus ausgeführt wird. Das ursprüngliche Anfangs- und Enddatum des sichtbaren Diagrammbereichs werden zurückgegeben. Anhand des Parameters **scrollAction** erhalten Sie Informationen darüber, welcher Art der ausgelöste Scrollvorgang ist. Setzt man den Rückgabestatus auf **vcRetStatFalse**, so wird der integrierte Scroll-Automatismus unterdrückt, und Sie können in Ihrer Anwendung selbst auf das Ereignis reagieren.

	Datentyp	Beschreibung
Parameter:		
⇒ curStartDate	Date/Time	aktuelles Anfangsdatum des sichtbaren Diagrammbereiches
⇒ curEndDate	Date/Time	aktuelles Enddatum des sichtbaren Diagrammbereiches
⇒ scrollAction	ScrollActionEnum	Art des Scrollvorgangs
	Mögliche Werte:	
	vcScrollActionAutoscrollDown 102	Die Ansicht verschob sich automatisch nach unten.
	vcScrollActionAutoscrollLeft 101	Die Ansicht verschob sich automatisch nach links.
	vcScrollActionAutoscrollRight 102	Die Ansicht verschob sich automatisch nach rechts.
	vcScrollActionAutoscrollUp 101	Die Ansicht verschob sich automatisch nach oben.
	vcScrollActionMouseWheelDown 106	Bei gedrücktem MauseRad wurde die Maus nach unten geschoben.
	vcScrollActionMouseWheelLeft 105	Bei gedrücktem MauseRad wurde die Maus nach links geschoben.

	vcScrollActionMouseWheelRight 106	Bei gedrücktem Mausrad wurde die Maus nach rechts geschoben.
	vcScrollActionMouseWheelUp 105	Bei gedrücktem Mausrad wurde die Maus nach oben geschoben.
	vcScrollActionSBLineDown 1	Die Ansicht verschob sich zur unteren Begrenzungslinie.
	vcScrollActionSBLineLeft 0	Die Ansicht verschob sich zur linken Begrenzungslinie.
	vcScrollActionSBLineRight 1	Die Ansicht verschob sich zur rechten Begrenzungslinie.
	vcScrollActionSBLineUp 0	Die Ansicht verschob sich zur oberen Begrenzungslinie.
	vcScrollActionSBNothing -1	Die Ansicht wurde nicht verschoben.
	vcScrollActionSBPageDown 3	Die Ansicht wurde um eine Seite nach unten versetzt.
	vcScrollActionSBPageLeft 2	Die Ansicht wurde um eine Seite nach links versetzt.
	vcScrollActionSBPageRight 3	Die Ansicht wurde um eine Seite nach rechts versetzt.
	vcScrollActionSBPageUp 2	Die Ansicht wurde um eine Seite nach oben versetzt.
	vcScrollActionSBThumbPosition 4	Die Versetzung um einen Schritt ist abgeschlossen.
	vcScrollActionSBThumbTrack 5	Die Ansicht wurde um einen Schritt versetzt.
	vcScrollActionScrollEnd 104	Scrollen über die Taste Ende oder über das Kontextmenü an das Diagrammende (rechts unten)
	vcScrollActionScrollHome 103	Scrollen über die Taste Pos 1 oder über das Kontextmenü an den Diagrammanfang (links oben)
	vcScrollActionThumbTrackDown 108	Thumb (Balken im Scrollbar) nach unten geschoben
	vcScrollActionThumbTrackLeft 107	Thumb (Balken im Scrollbar) nach links geschoben
	vcScrollActionThumbTrackRight 108	Thumb (Balken im Scrollbar) nach rechts geschoben
	vcScrollActionThumbTrackUp 107	Thumb (Balken im Scrollbar) nach oben geschoben
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnPreScrollDiagramHor(ByVal curStartDate _
                                         As Date, ByVal curEndDate As Date, _
                                         ByVal scrollAction As _
                                         VcGanttLib.ScrollActionEnum, _
                                         returnStatus As Variant)

    If curStartDate > "01.01.2014" Then
        returnStatus = vcRetStatFalse
    End If

End Sub
```

OnResourceSchedulingProgress

Ereignis von VcGantt

Bei der Ressourcenplanung informiert dieses Ereignis über den Fortschritt des Planungsprozesses. Es wird die Anzahl der bereits geplanten

Arbeitsaufträge und die Gesamtzahl der Aufträge bekannt gegeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Planung abgebrochen.

	Datentyp	Beschreibung
Parameter:		
⇒ ScheduledJobCount	Long	Anzahl der eingeplanten Arbeitsaufträge
⇒ TotalJobCount	Long	Gesamtzahl der Arbeitsaufträge
⇐ ReturnStatus	Object	Rückgabestatus vcRetStatFalse: Der Planungsprozess wird abgebrochen vcRetStatDefault: Der Planungsprozess wird weitergeführt

Code-Beispiel

```
Private Sub VcGantt1_OnResourceSchedulingProgress(ByVal scheduledJobCount As Long, ByVal totalJobCount As Long, returnStatus As Variant)
    MsgBox scheduledJobCount & " of " & totalJobCount
End Sub
```

OnResourceSchedulingWarning

Ereignis von VcGantt

Dieses Ereignis wird ausgelöst, wenn die Ressourcenplanung (s. Methode **process** im Objekt VcResourceScheduler2) Inkonsistenzen in den Datensätzen findet. Dieses Ereignis ermöglicht, bestimmte Fehler in der Datendefinition herauszufinden. Sie haben die Möglichkeit, die Ressourcenplanung abzubrechen, indem Sie den Rückgabestatus setzen.

	Datentyp	Beschreibung
Parameter:		
⇐ warningType	ResSchedWarningTypeEnum	Art der Warnung
	Mögliche Werte: vcResSchedAssignment-LoadPerItemIsZero 23 vcResSchedAssignment-NoDataRecords 0 vcResSchedAssignment-NoOperationID 3 vcResSchedAssignment-NoResourceID 1	Beim angegebenen Zuweisungsdatensatz wird der Inhalt des Datenfeldes LoadOrConsumptionPerItem zur Zahl 0 ausgewertet, was dazu führt, dass die Zuweisung bei der Planung ignoriert wird. Es sind keine Zuweisungsdatensätze vorhanden; der Parameter dataRecord ist Nothing . Im mitübergebenen Zuweisungsdatensatz ist das Datenfeld für die Operationsdatensatz-ID leer, dadurch wird die Zuweisung im weiteren Ablauf ignoriert. Im mitübergebenen Zuweisungsdatensatz sind alle Datenfelder für Ressourcendatensatz-IDs leer, dadurch wird die Zuweisung im weiteren Ablauf ignoriert.

vcResSchedAssignment- OperationNotFound 4	Im mitübergebenen Zuweisungsdatensatz wurde der Operationsdatensatz zur dort angegebenen Operationsdatensatz-ID nicht gefunden, dadurch wird die Zuweisung im weiteren Ablauf ignoriert.
vcResSchedAssignment- ResourceNotFound 2	Im mitübergebenen Zuweisungsdatensatz wurde der Ressourcendatensatz zur dort angegebenen Ressourcendatensatz-ID nicht gefunden, dadurch wird die Zuweisung im weiteren Ablauf ignoriert.
vcResSchedAssignment- TimingResourceMultiple 5	Der mitübergebene Zuweisungsdatensatz stellt eine unerlaubte zweite oder weitere Zuweisung einer Operation zu einer Ressource vom Typ vcResSchedTiming dar. Dadurch wird die Zuweisung im weiteren Ablauf ignoriert.
vcResSchedOperation- LoadPerItemIsZero 24	Beim angegebenen Operationsdatensatz wird der Inhalt des Datenfeldes für LoadPerItem zur Zahl 0 ausgewertet, was dazu führt, dass die Operation bei der Planung ignoriert wird.
vcResSchedOperation- NoTaskID 6	Im mitübergebenen Operationsdatensatz ist das Datenfeld für die Auftragsdatensatz-ID leer, dadurch wird die Operation im weiteren Ablauf ignoriert.
vcResSchedOperation- OverlapQuantityOutOf- Range 19	Die Warnung tritt auf, wenn die Überlappungs- menge (OverlapQuantity) einer Operation größer ist als die Menge des zugehörigen Auftrags. Dies führt in der Folge dazu, dass der Auftrag nicht eingeplant werden kann.
vcResSchedOperation- StartLockDateOutOf- Range 15	Die Warnung tritt auf, wenn das StartLockDate einer Operation nicht zwischen Freigabe- und Fälligkeitsdatum des Auftrags (ReleaseDate und DueDate) liegt. Dies führt in der Folge dazu, dass der Auftrag nicht eingeplant werden kann.
vcResSchedOperation- TaskNotFound 7	Im mitübergebenen Operationsdatensatz wurde der Auftragsdatensatz zur dort angegebenen Auftragsdatensatz-ID nicht gefunden, dadurch wird die Operation im weiteren Ablauf ignoriert.
vcResSchedOperation- WorkInProgressOutOf- Range 20	Die Warnung tritt auf, wenn die bereits fertig- gestellte Menge in dem mitgegebenen Operationsdatensatz größer ist als die Arbeits- menge des zugehörigen Auftrags. Dies führt in der Folge dazu, dass der Auftrag nicht eingeplant werden kann.
vcResSchedResource- CalendarNotFound 22	Die Warnung tritt auf, wenn zum Namen eines Kalenders, der im Ressourcen-Datenfeld mit dem Index aus der Eigenschaft ResourceCalendar- NameFieldIndex gesetzt ist, kein Kalenderobjekt vorhanden ist.
vcResSchedResource- GroupResourceNot- Found 10	Im mitübergebenen Ressourcendatensatz wurde der Ressourcendatensatz zur dort angegebenen Gruppenressourcendatensatz-ID nicht gefunden, dadurch kann die Ressource keiner Gruppe zugeordnet werden.
vcResSchedResource- HistogramNotFound 21	Die Warnung tritt auf, wenn das namensgleiche Histogramm der Ressource nicht vorhanden ist.
vcResSchedResource- InputCurvelsCompletely- Zero 12	Alle Werte der Eingabekurve zum mitübergebenen Ressourcendatensatz sind Null. Eingabekurven sind bei Ressourcen vom Typ vcResSchedTiming und vcResSchedWork Kapazitätskurven und bei Ressourcen vom Typ vcResSchedMaterial sind sie Lieferkurven.
vcResSchedResource- InputCurveNotFound 11	Die Eingabekurve zum mitübergebenen Ressourcendatensatz konnte nicht gefunden werden. Eingabekurven sind bei Ressourcen vom Typ vcResSchedTiming und vcResSchedWork Kapazitätskurven und bei Ressourcen vom Typ vcResSchedMaterial sind sie Lieferkurven.

⇐ dataRecord ⇐ returnStatus	vcResSchedResource-OutputCurveNotFound 13	Die Ausgabekurve zum mitübergebenen Ressourcendatensatz konnte nicht gefunden werden. Ausgabekurven sind bei Ressourcen vom Typ vcResSchedTiming und vcResSchedWork Belegungskurven und bei Ressourcen vom Typ vcResSchedMaterial sind sie Lagerbestandskurven.
	vcResSchedResource-OutputCurveOfFalse-Type 14	Die Ausgabekurve zum mitübergebenen Ressourcendatensatz kann nicht benutzt werden, weil sie nicht vom Typ vcSetCurve ist (s. Methode CurveType beim Objekt VcCurve). Ausgabekurven sind bei Ressourcen vom Typ vcResSchedTiming und vcResSchedWork Belegungskurven und bei Ressourcen vom Typ vcResSchedMaterial sind sie Lagerbestandskurven.
	vcResSchedTaskCapacity-BeyondLimit 25	Diese Warnung tritt auf, wenn in einem Auftrag mindestens eine Operation vorhanden ist, deren Kapazitätsanforderung über einem internen Limit liegt. Die Kapazitätsanforderung ergibt sich aus der Auftragsmenge beim Auftragsdatensatz, dem LoadPerItem bei der Operation sowie gegebenenfalls einem Effizienzfaktor bei der zu belegenden Ressource. Das Limit liegt zur Zeit bei 100000.
	vcResSchedTaskDueDate-EarlierThanReleaseDate 9	Im mitübergebenen Auftragsdatensatz liegt das Fälligkeitsdatum früher als das Freigabedatum, dadurch wird der Auftrag im weiteren Ablauf ignoriert.
	vcResSchedTaskDueDate-EqualToReleaseDate 18	Die Warnung tritt auf, wenn das Freigabedatum des Auftrags (ReleaseDate) mit dem Fälligkeitsdatum (DueDate) identisch ist. Dies führt in der Folge dazu, dass der Auftrag nicht eingeplant werden kann.
	vcResSchedTaskDueDate-OutOfRange 17	Die Warnung tritt auf, wenn das Fälligkeitsdatum (DueDate) eines Auftrags nicht zwischen PlanningStartDate und PlanningEndDate oder im per Default gesetzten sichtbaren Bereich liegt. Falls auch das Freigabedatum außerhalb des erlaubten Zeitraumes liegt, kann der Auftrag nicht eingeplant werden.
	vcResSchedTaskQuantity-IsZero 8	Im mitübergebenen Auftragsdatensatz ist die Auftragsmenge Null, dadurch wird der Auftrag im weiteren Ablauf ignoriert.
	vcResSchedTaskRelease-DateOutOfRange 16	Die Warnung tritt auf, wenn das Freigabedatum (ReleaseDate) eines Auftrags nicht zwischen PlanningStartDate und PlanningEndDate oder den per Default gesetzten Daten liegt. Falls auch das Fälligkeitsdatum außerhalb des erlaubten Zeitraumes liegt, kann der Auftrag nicht eingeplant werden.
	VcDataRecord	Datensatz, auf den sich die Warnung bezieht
	Variant	Rückgabestatus
		vcRetStatFalse: Der Planungsprozess wird abgebrochen
		vcRetStatDefault: Der Planungsprozess wird weitergeführt
	Mögliche Werte:	
	vcRetStatFalse 0	Die Ressourcenplanung wird abgebrochen.
	vcRetStatOK 1	Die Ressourcenplanung wird fortgeführt.

Code-Beispiel

```

Private Sub VcGantt1_OnResourceSchedulingWarning(ByVal warningType As
VcGanttLib.ResourceSchedulingWarningTypeEnum, ByVal DataRecord As
VcGanttLib.VcDataRecord, returnStatus As Variant)

    Select Case warningType
    Case
ResourceSchedulingWarningTypeEnum.vcResSchedTaskDueDateEarlierThanReleaseDate
        MsgBox "Please change the due date " & DataRecord.DataField(11)
    End Select

End Sub

```

OnScrollComponent**Ereignis von VcGantt**

Mit diesem Ereignis können Sie bei jeder interaktiven Scrollaktion Folgendes ermitteln:

1. die Komponente, die gescrollt wird (nur vcDiagramComponent, vcHistogramComponent, vcListComponent und vcRightListComponent werden als „Masterscroller“ berücksichtigt, da alle anderen Komponenten abhängig mitgescrollt werden)
2. die Bewegungsorientierung (horizontal oder vertikal)
3. die Art der Benutzeraktion.

Hinweis: Die tatsächliche Scrollaktion ergibt sich aus der Kombination der Parameter **orientation** und **scrollAction**, da in Windows-Programmen die up/left- und down/right-Aktionen jeweils dieselbe Nummer haben. Z. B.:

vcScrollActionSBPageLeft = vcScrollActionSBPageUp = 2

vcScrollActionThumbTrackLeft = vcScrollActionThumbTrackUp = 107

Das folgende Beispiel zeigt die Auswertung über **orientation** für **VcScrollActionSBPageLeft** und **vcScrollActionSBPageUp**, die beide den Wert 2 haben.

	Datentyp	Beschreibung
Parameter: ⇒ component	ComponentTypeEnum Mögliche Werte: vcAdditionalListComponent 1 vcBottomListTitleComponent 14 vcBottomRightListTitleComponent 17	Komponententyp zusätzliche Tabelle untere Titelleiste Tabellenbereich unten rechts

	vcBottomTimeScaleComponent 15	untere Zeitskala
	vcDiagramComponent 4	Diagramm
	vcHistogramComponent 8	Histogramm
	vcHistogramVerScaleComponent 7	numerische Skala (vertikale Histogramm- skala)
	vcLegendComponent 10	Legende (zur Zeit ohne Funktion; Rückgabewerte 00)
	vcListComponent 0	Tabelle
	vcListTitleComponent 2	Tabellenüberschrift
	vcRightListComponent 5	rechte Tabelle
	vcRightListTitleComponent 16	Tabellenüberschrift der rechten Tabelle
	vcTimeScaleComponent 3	obere Zeitskala
	vcTopTitleComponent 11	obere Titelleiste
⇒ Orientation	ScrollOrientationEnum	Scrollrichtung
	Mögliche Werte:	
	vcHorizontal 1	horizontales Scrollen
	vcVertical 2	vertikales Scrollen
⇒ scrollAction	ScrollActionEnum	Art des Scrollvorgangs
	Mögliche Werte:	
	vcScrollActionAutoscrollDown 102	Die Ansicht verschob sich automatisch nach unten.
	vcScrollActionAutoscrollLeft 101	Die Ansicht verschob sich automatisch nach links.
	vcScrollActionAutoscrollRight 102	Die Ansicht verschob sich automatisch nach rechts.
	vcScrollActionAutoscrollUp 101	Die Ansicht verschob sich automatisch nach oben.
	vcScrollActionMouseWheelDown 106	Bei gedrücktem Mousrad wurde die Maus nach unten geschoben.
	vcScrollActionMouseWheelLeft 105	Bei gedrücktem Mousrad wurde die Maus nach links geschoben.
	vcScrollActionMouseWheelRight 106	Bei gedrücktem Mousrad wurde die Maus nach rechts geschoben.
	vcScrollActionMouseWheelUp 105	Bei gedrücktem Mousrad wurde die Maus nach oben geschoben.
	vcScrollActionSBLineDown 1	Die Ansicht verschob sich zur unteren Begrenzungslinie.
	vcScrollActionSBLineLeft 0	Die Ansicht verschob sich zur linken Begrenzungslinie.
	vcScrollActionSBLineRight 1	Die Ansicht verschob sich zur rechten Begrenzungslinie.
	vcScrollActionSBLineUp 0	Die Ansicht verschob sich zur oberen Begrenzungslinie.
	vcScrollActionSBNothing -1	Die Ansicht wurde nicht verschoben.
	vcScrollActionSBPageDown 3	Die Ansicht wurde um eine Seite nach unten versetzt.
	vcScrollActionSBPageLeft 2	Die Ansicht wurde um eine Seite nach links versetzt.
	vcScrollActionSBPageRight 3	Die Ansicht wurde um eine Seite nach rechts versetzt.
	vcScrollActionSBPageUp 2	Die Ansicht wurde um eine Seite nach oben versetzt.
	vcScrollActionSBThumbPosition 4	Die Versetzung um einen Schritt ist abgeschlossen.
	vcScrollActionSBThumbTrack 5	Die Ansicht wurde um einen Schritt versetzt.
	vcScrollActionScrollEnd 104	Scrollen über die Taste Ende oder über das Kontextmenü an das Diagrammende (rechts unten)
	vcScrollActionScrollHome 103	Scrollen über die Taste Pos 1 oder über das Kontextmenü an den Diagrammanfang (links oben)

vcScrollActionThumbTrackDown 108	Thumb (Balken im Scrollbar) nach unten geschoben
vcScrollActionThumbTrackLeft 107	Thumb (Balken im Scrollbar) nach links geschoben
vcScrollActionThumbTrackRight 108	Thumb (Balken im Scrollbar) nach rechts geschoben
vcScrollActionThumbTrackUp 107	Thumb (Balken im Scrollbar) nach oben geschoben

Code-Beispiel

```

If orientation = vcHorizontal and scrollAction = vcScrollActionSBPageLeft _
    Then MsgBox "Scolled left"
ElseIf orientation = vcHorizontal and _
    scrollAction = vcScrollActionSBPageRight _
    Then MsgBox "Scrolled right"
ElseIf orientation = vcVertical and scrollAction = vcScrollActionSBPageUp _
    Then MsgBox "Scrolled up"
End If

```

OnScrollDiagramHor

Ereignis von VcGantt

Dieses Ereignis tritt ein, nachdem der Scrollvorgang ausgeführt wurde. Das neue Anfangs- und Enddatum des sichtbaren Diagrammbereichs werden übergeben. Anhand des Parameters **scrollAction** erhalten Sie Informationen darüber, welcher Art der ausgeführte Scrollvorgang war.

	Datentyp	Beschreibung
Parameter:		
⇒ newStartDate	Date/Time	Neues Anfangsdatum des sichtbaren Diagrammbereiches
⇒ newEndDate	Date/Time	Neues Enddatum des sichtbaren Diagrammbereiches
⇒ scrollAction	ScrollActionEnum	Art des Scrollvorgangs
	Mögliche Werte:	
	vcScrollActionAutoscrollDown 102	Die Ansicht verschob sich automatisch nach unten.
	vcScrollActionAutoscrollLeft 101	Die Ansicht verschob sich automatisch nach links.
	vcScrollActionAutoscrollRight 102	Die Ansicht verschob sich automatisch nach rechts.
	vcScrollActionAutoscrollUp 101	Die Ansicht verschob sich automatisch nach oben.
	vcScrollActionMouseWheelDown 106	Bei gedrücktem Mausrad wurde die Maus nach unten geschoben.
	vcScrollActionMouseWheelLeft 105	Bei gedrücktem Mausrad wurde die Maus nach links geschoben.
	vcScrollActionMouseWheelRight 106	Bei gedrücktem Mausrad wurde die Maus nach rechts geschoben.
	vcScrollActionMouseWheelUp 105	Bei gedrücktem Mausrad wurde die Maus nach oben geschoben.
	vcScrollActionSBLineDown 1	Die Ansicht verschob sich zur unteren Begrenzungslinie.

vcScrollActionSBLineLeft 0	Die Ansicht verschob sich zur linken Begrenzungslinie.
vcScrollActionSBLineRight 1	Die Ansicht verschob sich zur rechten Begrenzungslinie.
vcScrollActionSBLineUp 0	Die Ansicht verschob sich zur oberen Begrenzungslinie.
vcScrollActionSBNothing -1	Die Ansicht wurde nicht verschoben.
vcScrollActionSBPageDown 3	Die Ansicht wurde um eine Seite nach unten versetzt.
vcScrollActionSBPageLeft 2	Die Ansicht wurde um eine Seite nach links versetzt.
vcScrollActionSBPageRight 3	Die Ansicht wurde um eine Seite nach rechts versetzt.
vcScrollActionSBPageUp 2	Die Ansicht wurde um eine Seite nach oben versetzt.
vcScrollActionSBThumbPosition 4	Die Versetzung um einen Schritt ist abgeschlossen.
vcScrollActionSBThumbTrack 5	Die Ansicht wurde um einen Schritt versetzt.
vcScrollActionScrollEnd 104	Scrollen über die Taste Ende oder über das Kontextmenü an das Diagrammende (rechts unten)
vcScrollActionScrollHome 103	Scrollen über die Taste Pos 1 oder über das Kontextmenü an den Diagrammanfang (links oben)
vcScrollActionThumbTrackDown 108	Thumb (Balken im Scrollbar) nach unten geschoben
vcScrollActionThumbTrackLeft 107	Thumb (Balken im Scrollbar) nach links geschoben
vcScrollActionThumbTrackRight 108	Thumb (Balken im Scrollbar) nach rechts geschoben
vcScrollActionThumbTrackUp 107	Thumb (Balken im Scrollbar) nach oben geschoben

Code-Beispiel

```
Private Sub VcGantt1_OnScrollDiagramHor(ByVal newStartDate _
                                         As Date, ByVal newEndDate As Date, _
                                         ByVal scrollAction As _
                                         VcGanttLib.ScrollActionEnum)

    If newEndDate > "01.01.2014" Then
        Select Case MsgBox("scrolling to: " & newEndDate, vbOKCancel)
            Case vbOK
                Call VcGantt1.ScrollToDate(newEndDate, vcHorCenterAligned, 0)
            Case vbCancel
                returnStatus = vcRetStatFalse
        End Select
    End If
End Sub
```

OnSelectField

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Zelle einer Tabelle oder das Feld einer Box selektiert wird. Die Selektion kann durch Setzen des Rückgabestatus verhindert werden.

	Datentyp	Beschreibung
Parameter:		
editObject	Object editiertes Objekt	
editObjectType	VcObjectTypeEnum Objekttyp	
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeCalendarGrid 18 vcObjTypeCurve 12 vcObjTypeDateLine 9 vcObjTypeGroup 7 vcObjTypeGroupInDiagram 11 vcObjTypeGroupInTable 7 vcObjTypeHistogram 13 vcObjTypeLayer 8 vcObjTypeLinkCollection 3 vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14 vcObjTypeTable 4 vcObjTypeTableCaption 5 vcObjTypeTimeScale 6	Objekttyp Box Objekttyp Kalendergitter Objekttyp Kurve Objekttyp Stichtaglinie Objekttyp Gruppe Objekttyp Gruppe im Knotenbereich Objekttyp Gruppe im Tabellenbereich Objekttyp Histogramm Objekttyp Layer Objekttyp LinkCollection Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Knoten im Tabellenbereich kein Objekt Objekttyp Werteskala Objekttyp Summenbalken Objekttyp Tabelle Objekttyp Tabellenüberschrift Objekttyp Zeitskala
fieldIndex	Long Feldindex	
objRectComplete	VcRect komplettes Rechteck des getroffenen Objekts	
objRectVisible	VcRect sichtbares Rechteck des getroffenen Objekts	
fldRectComplete	VcRect komplettes Rechteck des getroffenen Feldes	
fldRectVisible	VcRect sichtbares Rechteck des getroffenen Feldes	
returnStatus	Variant	
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Das Feld wird nicht ausgewählt. Das Feld wird ausgewählt.

OnShowCurveNameInMenu

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Namen von per API definierten Histogrammkurven in einem Kontextmenü angezeigt werden. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** werden die Namen der Histogrammkurven nicht im Kontextmenü angezeigt.

	Datentyp	Beschreibung
Parameter:		
⇒ Histogram	VcHistogram	Getroffenes Histogramm
⇒ curveName	String	Name der Histogrammkurve
	Mögliche Werte:	Name der Farbzuoordnungstabelle
⇔ returnStatus	Variant	Rückgabewert

Code-Beispiel

```
Private Sub VcGantt1_OnShowCurveNameInMenu(ByVal Histogram As _
                                         VcGanttLib.VcHistogram, _
                                         ByVal curveName As String, _
                                         returnStatus As Variant)

    returnStatus = retStatFalse
End Sub
```

OnShowDate**Ereignis von VcGantt**

Dieses Ereignis tritt bei jeder Mausbewegung im Diagramm- oder Zeitskalenbereich ein. Es wird das Datum der Mausposition als Parameter zurückgeben.

	Datentyp	Beschreibung
Parameter:		
⇒ dateVal	Date/Time	Datum

Code-Beispiel

```
Private Sub VcGantt1_OnShowDate(ByVal dateVal As Date)

    Text1.Text = dateVal
End Sub
```

OnShowInPlaceEditor**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der im Programm implementierte Editor gestartet wird.

Das Ereignis wird erst aktiviert, wenn die Eigenschaft **InPlaceEditing-Allowed** auf True gesetzt ist.

Durch setzen des Rückgabestatus auf **False** kann dieses Ereignis verhindert werden, so dass an den übergebenen Koordinaten ein eigener Editor aufgerufen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ editObject	Object	editiertes Objekt
⇒ editObjectType	VcObjectTypeEnum	Objekttyp
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0	Objekttyp Box Objekttyp Knoten im Legendenbereich Objekttyp Knoten im Tabellenbereich kein Objekt
⇒ fieldIndex	Long	Feldindex
⇒ objRectComplete	VcRect	komplettes Rechteck des getroffenen Objekts
⇒ objRectVisible	VcRect	sichtbares Rechteck des getroffenen Objekts
⇒ fldRectComplete	VcRect	komplettes Rechteck des getroffenen Feldes
⇒ fldRectVisible	VcRect	sichtbares Rechteck des getroffenen Feldes
returnStatus	Variant	
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die integrierte Editor wird nicht gestartet. Der integrierte Editor wird gestartet.

Code-Beispiel

```
Private Sub VcGantt1_OnShowInPlaceEditor(ByVal editObject As Object, _
    ByVal editObjectType As VcGanttLib.VcObjectTypeEnum, _
    ByVal fieldIndex As Long, ByVal objRectComplete As _
    VcGanttLib.VcRect, ByVal objRectVisible As _
    VcGanttLib.VcRect, ByVal fldRectComplete As _
    VcGanttLib.VcRect, ByVal fldRectVisible As _
    VcGanttLib.VcRect, returnStatus As Variant)

    Dim oldScaleMode As Long

    If editObjectType = vcObjTypeNodeInTable Then
        returnStatus = vcRetStatFalse

        Set myEditObject = editObject
        myEditObjectType = editObjectType
        myEditObjectFieldIndex = fieldIndex
        oldScaleMode = Me.ScaleMode
        Me.ScaleMode = vbPixels

        Select Case fieldIndex
            Case 1 'Name
                Text1.Left = fldRectVisible.Left + VcGantt1.Left
                Text1.Top = fldRectVisible.Top + VcGantt1.Top
                Text1.Width = fldRectVisible.Width
                Text1.Height = fldRectVisible.Height
                Text1.Text = editObject.DataField(fieldIndex)
                Text1.Visible = True
                Text1.SetFocus

            Case 2, 3 'Start or End
                MonthView1.Left = fldRectVisible.Left + VcGantt1.Left
```

```

        MonthView1.Top = fldRectVisible.Top + VcGantt1.Top
        MonthView1.Value = editObject.DataField(fieldIndex)
        MonthView1.Visible = True
        MonthView1.SetFocus

    Case 13 'Employee
        Combo1.Left = fldRectVisible.Left + VcGantt1.Left
        Combo1.Top = fldRectVisible.Top + VcGantt1.Top
        Combo1.Width = fldRectVisible.Width
        Combo1.Text = editObject.DataField(fieldIndex)
        Combo1.Visible = True
        Combo1.SetFocus

End Select

Me.ScaleMode = oldScaleMode

End If

End Sub

```

OnStatusLineText

Ereignis von VcGantt

Dieses Ereignis tritt immer dann ein, wenn eine Information von allgemeiner Bedeutung in der Statuszeile bereitgestellt wird. Das kann ein funktionaler Hinweis sein (z. B. beim Laden), aber auch Informationen über einen Knoten, auf den die Maus gerade zeigt.

	Datentyp	Beschreibung
Parameter:		
⇒ text	String	Informationstext
	Mögliche Werte:	Name der Farbzusordnungstabelle
⇒ paneNo	Integer	Dummy
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```

Private Sub VcGantt1_OnStatusLineText(ByVal Text As String, _
                                       ByVal paneNo As Integer)
    Text1.Text = Text
End Sub

```

OnSupplyTextEntry

Ereignis von VcGantt

Dieses Ereignis tritt nur auf, wenn Sie die VcGantt-Eigenschaft **EnableSupplyTextEntryEvent** auf **True** gesetzt haben. Das Ereignis tritt auf, wenn ein Text ausgegeben werden soll. Sie können hier alle vorgegebenen Texte

durch eigene Texte ersetzen, z. B. um sie in unterschiedliche Sprachen zu übersetzen. Das betrifft die Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen und Monats- und Tagesnamen.

	Datentyp	Beschreibung
Parameter: ⇒ controllIndex	TextEntryIndexEnum	Textkonstante, deren Inhalt ersetzt werden soll
	Mögliche Werte: vcTXECtxmenArrowMode 2116	Text im Kontextmenü: Selektier-Modus
	vcTXECtxmenBarGroupSepLine 2111	Nicht mehr verwendete Konstante, an der API noch sichtbar
	vcTXECtxmenCancelGrouping 2108	Nicht mehr verwendete Konstante, an der API noch sichtbar
	vcTXECtxmenCreateBoxMode 2135	Text im Kontextmenü: Modus: Box erzeugen
	vcTXECtxmenCreateLinkMode 2118	Text im Kontextmenü: Verbindung ziehen
	vcTXECtxmenCreateNodeMode 2117	Text im Kontextmenü: Knoten erzeugen
	vcTXECtxmenDateLineGrid 2106	Text im Kontextmenü: Gitternetzlinien
	vcTXECtxmenDeleteCurvePoint 2131	Text im Kontextmenü: Kurvenpunkt löschen
	vcTXECtxmenDeleteLink 2102	Text im Kontextmenü: Verbindung löschen
	vcTXECtxmenDeleteNode 2101	Text im Kontextmenü: Knoten löschen
	vcTXECtxmenEditGroup 2160	Text im Kontextmenü der Gruppe : Gruppensdaten bearbeiten
	vcTXECtxmenEditLink 2154	Text im Kontextmenü: Verbindung bearbeiten
	vcTXECtxmenEditNode 2100	Text im Kontextmenü: Daten bearbeiten
	vcTXECtxmenFilePrint 2122	Text im Kontextmenü: Drucken
	vcTXECtxmenFilePrintPreview 2121	Text im Kontextmenü: Druckvorschau
	vcTXECtxmenFilePrintSetup 2120	Text im Kontextmenü: Drucker einrichten
	vcTXECtxmenFullDiagram 2156	Text im Kontextmenü: Gesamtdiagramm wiederherstellen
	vcTXECtxmenGraphicExport 2123	Text im Kontextmenü: Grafik exportieren
	vcTXECtxmenGroupCollapse 2114	Text im Kontextmenü: Gruppe kollabieren
	vcTXECtxmenGroupCollapseRowsBelow 2129	Text im Kontextmenü: Zeilen der Gruppe kollabieren
	vcTXECtxmenGroupDelete 2115	Text im Kontextmenü: Gruppe löschen
	vcTXECtxmenGrouped 2107	Nicht mehr verwendete Konstante, an der API noch sichtbar
	vcTXECtxmenGroupExpand 2113	Text im Kontextmenü: Gruppe expandieren
	vcTXECtxmenGroupExpandRowsBelow 2128	Text im Kontextmenü: Zeilen der Gruppe expandieren
	vcTXECtxmenGroupingOptions 2109	Nicht mehr verwendete Konstante, an der API noch sichtbar
	vcTXECtxmenGroupNodesBelow 2126	Nicht mehr verwendete Konstante, an der API noch sichtbar
	vcTXECtxmenGroupNodesInOneRow 2127	Text im Kontextmenü für Gruppen : Alle Knoten in einer Zeile
	vcTXECtxmenGroupNodesOptimized 2124	Text im Kontextmenü für Gruppen : Knoten optimiert

vcTXEctxmenGroupNodesOverlaid 2125	Text im Kontextmenü für Gruppen: Knoten überlappend
vcTXEctxmenGroupOutlineIndent 2134	Text im Kontextmenü: Tiefer stufen
vcTXEctxmenGroupOutlineOutdent 2133	Text im Kontextmenü: Höher stufen
vcTXEctxmenGroupSortingOptions 2110	Text im Kontextmenü: Sortieroptionen für Gruppen
vcTXEctxmenInsertCurvePointMode 2130	Text im Kontextmenü: Kurvenpunkt einfügen
vcTXEctxmenInvertSelection 2103	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEctxmenPageLayout 2119	Text im Kontextmenü: Seite einrichten
vcTXEctxmenReOptimizeNodesInGroup 2136	Text im Kontextmenü: Knoten erneut optimieren
vcTXEctxmenShowLegendView 2158	Text im Kontextmenü Legendenansicht anzeigen
vcTXEctxmenShowWorldView 2157	Text im Kontextmenü Komplettansicht anzeigen
vcTXEctxmenSubDiagram 2155	Text im Kontextmenü Teildigramm erstellen
vcTXEctxmenTimeScaleEditor 2104	Text im Kontextmenü: Zeitskala bearbeiten
vcTXEctxmenToggleDateLine 2105	Text im Kontextmenü: Stichtaglinie
vcTXEctxmenUnmarkAllCurves 2136	Text im Kontextmenü des Histogramms : Alle Kurven demarkieren
vcTXEDlgLegArrangement 2046	Text im Dialog Legendenattribute: Anordnung
vcTXEDlgLegBottomMargin 2052	Text im Dialog Legendenattribute: Unterer Rand :
vcTXEDlgLegFixedToColumns 2048	Text im Dialog Legendenattribute: nach Spaltenanzahl
vcTXEDlgLegFixedToRows 2047	Text im Dialog Legendenattribute: nach Zeilenanzahl
vcTXEDlgLegFixedToRowsAndColumns 2049	Text im Dialog Legendenattribute: nach Zeilen- und Spaltenanzahl
vcTXEDlgLegIdcancel 2042	Schaltfläche im Dialog Legendenattribute: Abbrechen
vcTXEDlgLegIdd 2040	Dialog Legendenattribute Beschriftung der Titelzeile
vcTXEDlgLegIdok 2041	Schaltflächentext im Dialog Legendenattribute: OK
vcTXEDlgLegLegendElements 2045	Text im Dialog Legendenattribute: Legendenelemente
vcTXEDlgLegLegendFont 2053	Schaltfläche im Dialog Legendenattribute: Schriftart... für Legende
vcTXEDlgLegLegendTitleFont 2044	Schaltfläche im Dialog Legendenattribute: Schriftart... für Legendentitel
vcTXEDlgLegLegendTitleVisible 2043	Text im Dialog Legendenattribute: Legendentitel sichtbar
vcTXEDlgLegMargins 2050	Text im Dialog Legendenattribute: Ränder
vcTXEDlgLegTopMargin 2051	Text im Dialog Legendenattribute: Oberer Rand :
vcTXEDlgNedCaptionPrefix 2024	Dialog Vorgänge bearbeiten ,: Text für Beschriftungszeile: "Knoten"
vcTXEDlgNedIdapply 2027	Dialog Vorgänge bearbeiten , "Übernehmen"-Schaltfläche
vcTXEDlgNedIdcancel 2016	Text im Dialog Vorgänge bearbeiten: Abbrechen
vcTXEDlgNedIdclose 2029	Dialog Vorgänge bearbeiten: Schließen -Schaltfläche
vcTXEDlgNedIdd 2014	Überschrift des Dialogs Vorgänge bearbeiten

vcTXEDlgNedIdhelp 2028	Dialog Vorgänge bearbeiten : Hilfe-Schaltfläche
vcTXEDlgNedIdok 2015	Text im Dialog Vorgänge bearbeiten : OK
vcTXEDlgNedNamesColStr 2018	Text im Dialog Vorgänge bearbeiten : Datenfelder
vcTXEDlgNedTTGotoFirst 2032	Dialog Vorgänge bearbeiten : Tooltiptext Ersten ausgewählten Vorgang anzeigen
vcTXEDlgNedTTGotoLast 2035	Dialog Vorgänge bearbeiten , Tooltipp "Letzten ausgewählten Vorgang anzeigen"
vcTXEDlgNedTTGotoNext 2034	Dialog Vorgänge bearbeiten , Tooltiptext Nächsten ausgewählten Vorgang anzeigen
vcTXEDlgNedTTGotoPrev 2033	Dialog Vorgänge bearbeiten : Tooltiptext Vorherigen ausgewählten Vorgang anzeigen
vcTXEDlgNedValuesColStr 2019	Text im Dialog Vorgänge bearbeiten : Werte
vcTXEDlgTscEndDate 2012	Text im Dialog Zeitskala bearbeiten : Ende
vcTXEDlgTscIdcancel 2010	Dialog Zeitskala bearbeiten : Schaltflächentext Abbrechen
vcTXEDlgTscIdd 2008	Dialog Zeitskala bearbeiten : Text der Titelleiste
vcTXEDlgTscIdok 2009	Dialog Zeitskala bearbeiten : Schaltflächentext OK
vcTXEDlgTscScale 2013	Text im Dialog Zeitskala bearbeiten : Skala
vcTXEDlgTscStartDate 2011	Text im Dialog Zeitskala bearbeiten : Anfang
vcTXEErrTxtCannotMoveToEmptyRow 2735	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtEndNotEarlierThanNextSect 2734	Meldungstext: "Enddatum ""%s"" ist nicht früher als Enddatum des nächsten Zeitskalenabschnitts.\n\nDas alte Datum wird wieder eingefügt."
vcTXEErrTxtEndNotLaterThanStart 2732	Meldungstext: "Enddatum ""%s"" ist nicht später als das Startdatum.\n\nDas alte Datum wird wieder eingefügt."
vcTXEErrTxtEntryTooLong 2730	Meldungstext: "Eintrag ist zu lang, %s Zeichen sind möglich."
vcTXEErrTxtSpinNoButton 2727	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinNumberFormatFloat 2724	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinNumberFormatInt 2723	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinNumberMissing 2722	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinNumberTooHigh 2725	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinNumberTooLow 2726	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinUnitInsert 2720	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinUnitNotInsert 2721	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinWrongFormatString 2728	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinWrongUnitInserted 2718	Nicht mehr verwendete Konstante, an der API noch sichtbar
vcTXEErrTxtSpinWrongUnitNotInserted 2719	Nicht mehr verwendete Konstante, an der API noch sichtbar

vcTXEErrTxtStartNotEarlierThanEnd 2731	Meldungstext: "Startdatum ""%s"" ist nicht früher als das Enddatum.\n\nDas alte Datum wird wieder eingefügt."
vcTXEErrTxtStartNotLaterThanPrevSect 2733	Meldungstext: "Startdatum ""%s"" ist nicht später als der Start des vorangehenden Zeitskalenabschnitts.\n\nDas alte Datum wird wieder eingefügt."
vcTXEErrTxtWrongLongInteger 2729	Meldungstext: "Eintrag ist kein Integer oder zu lang."
vcTXEInfWndChangeEndDate 2615	Tooltipp-Text: Ende verschieben
vcTXEInfWndChangeSectionStartDate 2614	Tooltipp-Text: Startdatum des Zeitskalenabschnitts verändern.
vcTXEInfWndChangeStartDate 2618	Tooltipp-Text: Anfang verschieben
vcTXEInfWndCopyActivity 2619	Tooltipp-Text: Vorgang kopieren
vcTXEInfWndCreateActivity 2611	Tooltipp-Text: Knoten neu anlegen
vcTXEInfWndDate 2620	Tooltipp-Text: Datum
vcTXEInfWndDateValue 12620	Tooltipp-Text: Datum
vcTXEInfWndDayPI 2604	Tooltipp-Text: Tag
vcTXEInfWndDaySi 2603	Tooltipp-Text: Tag
vcTXEInfWndDuration 2602	Tooltipp-Text: Dauer
vcTXEInfWndDurationValue 12602	Tooltipp-Text: Dauer
vcTXEInfWndEnd 2601	Tooltipp-Text: Ende
vcTXEInfWndEndValue 12601	Tooltipp-Text: Enddatum
vcTXEInfWndHourPI 2606	Tooltipp-Text: Stunden
vcTXEInfWndHourSi 2605	Tooltipp-Text: Stunde
vcTXEInfWndMinPI 2608	Tooltipp-Text: Minuten
vcTXEInfWndMinSi 2607	Tooltipp-Text: Minuten
vcTXEInfWndMoveActivity 2612	Tooltipp-Text: Vorgang verschieben
vcTXEInfWndMoveLayer 2613	Tooltipp-Text: Layer verschieben
vcTXEInfWndResizeBUW 2616	Tooltipp-Text: Auflösung des Skalenabschnitts verändern
vcTXEInfWndResizeNumericBUW 2617	Tooltipp-Text: Auflösung der Skala verändern
vcTXEInfWndSecPI 2610	Tooltipp-Text: Sekunden
vcTXEInfWndSecSi 2609	Tooltipp-Text: Sekunde
vcTXEInfWndStart 12600	Tooltipp-Text: Startdatum der Stichtaglinie
vcTXEInfWndStart 2600	Tooltipp-Text: Anfang
vcTXEPrctBtAll 2306	Schaltflächen-Text des Druckvorschau -Dialogs: Übersicht
vcTXEPrctBtApply 2318	Schaltflächen-Text im Seite einrichten -Dialog: Anwenden
vcTXEPrctBtCancel 2302	Schaltflächen-Text im Druck-Info -Fenster: Abbrechen
vcTXEPrctBtClose 2303	Schaltflächen-Text des Druckvorschau -Dialogs: Schließen
vcTXEPrctBtFitToPage 2308	Schaltflächen-Text des Druckvorschau -Dialogs: Einpassen
vcTXEPrctBtNext 2305	Schaltflächen-Text des Druckvorschau -Dialogs: Weiter
vcTXEPrctBtOk 2301	Schaltflächen-Text des Seite einrichten -Dialogs: OK
vcTXEPrctBtPageLayout 2311	Schaltflächen-Text des Druckvorschau -Dialogs: Seite einrichten
vcTXEPrctBtPreviewZoomFactorItems 2321	Einträge für die Kombobox Zoomfaktor des Druckvorschau -Dialogs: Auto 75% 100% 150% 200%
vcTXEPrctBtPrevious 2304	Schaltflächen-Text des Druckvorschau -Dialogs: Vorher
vcTXEPrctBtPrint 2313	Schaltflächen-Text des Druckvorschau -Dialogs: Drucken

vcTXEPrctBtPrinterSetup 2312	Schaltflächen-Text des Druckvorschau -Dialogs: Drucker einrichten
vcTXEPrctBtSingle 2307	Schaltflächen-Text des Druckvorschau -Dialogs: Einzelseite
vcTXEPrctBtZoomPrint 2319	Schaltflächen-Text des Druckvorschau -Dialogs: Ausschnitt drucken...
vcTXEPrctDtAddCuttingMarks 2514	Text des Seite einrichten -Dialogs: Zuschnittmarken
vcTXEPrctDtAdjustTimescale 2560	Text des Seite einrichten -Dialogs: Zeitskala an Breite der Seite anpassen
vcTXEPrctDtAdoptTableWidthOfView 2591	Text des Seite einrichten -Dialogs: Aussehen von Bildschirmansicht übernehmen
vcTXEPrctDtAlignment 2526	Text des Seite einrichten -Dialogs: Ausrichtung
vcTXEPrctDtAlignmentItems 2583	Text des Seite einrichten -Dialogs: Oben links Oben Oben rechts Links Mittig Rechts Unten links Unten Unten rechts
vcTXEPrctDtBottom 2521	Text des Seite einrichten -Dialogs: Unten
vcTXEPrctDtCm 2530	Text des Seite einrichten -Dialogs: cm
vcTXEPrctDtCombinedFitToPage 2574	Text des Seite einrichten -Dialogs: Zoomen mit horizontaler Anpassung
vcTXEPrctDtCurrentValues 2581	Text des Seite einrichten -Dialogs: Aktuell
vcTXEPrctDtEnableDiagram 2559	Text des Seite einrichten -Dialogs: Diagramm anzeigen
vcTXEPrctDtEnableTable 2558	Text des Seite einrichten -Dialogs: Tabelle anzeigen
vcTXEPrctDtExportPage 2568	Text des Seite einrichten -Dialogs: Anpassen an Seitenzahl
vcTXEPrctDtFitToPage 2508	Text des Seite einrichten -Dialogs: Form A Form B Form C
vcTXEPrctDtFoldingMarksItems 2577	Text des Seite einrichten -Dialogs: "&Faltmarkierungen (DIN 824)
vcTXEPrctDtFoldingMarksText 2576	Text des Seite einrichten -Dialogs: Fußzeile
vcTXEPrctDtFooterGroup 2584	Text des Seite einrichten -Dialogs: Rahmen außen
vcTXEPrctDtFrameOutside 2515	Text des Seite einrichten -Dialogs: Zoll
vcTXEPrctDtInch 2588	Text des Seite einrichten -Dialogs: Links
vcTXEPrctDtLeft 2520	Text des Seite einrichten -Dialogs: Mindestgrößen für die Seitenränder
vcTXEPrctDtMargins 2529	Text des Seite einrichten -Dialogs: Seiten
vcTXEPrctDtMaxPages 2580	Text Aus Dialog
vcTXEPrctDtOff 2557	Text des Seite einrichten -Dialogs: Seitenaufteilung
vcTXEPrctDtOptions 2528	Text des Seite einrichten -Dialogs: Text
vcTXEPrctDtPageDescription 2562	Fenstertitel des Seite einrichten -Dialogs
vcTXEPrctDtPageLayout 2532	Text des Seite einrichten -Dialogs: Zeile.Spalte Spalte.Zeile Seite/Anzahl
vcTXEPrctDtPageNumberingItems 2582	Text des Seite einrichten -Dialogs: Seitennummerierung
vcTXEPrctDtPageNumbers 2518	Text des Seite einrichten -Dialogs: Seiten mit Leerraum auff&üllen
vcTXEPrctDtPagePadding 2585	

vcTXEPrctDtPagePreview 2533	Fenstertitel des Dialogs Druckvorschau
vcTXEPrctDtPagesMaxHeight 2511	Text des Seite einrichten -Dialogs: Maximale Höhe
vcTXEPrctDtPagesMaxWidth 2510	Text des Seite einrichten -Dialogs: Maximale Breite
vcTXEPrctDtPercent 2509	Text des Seite einrichten -Dialogs: %
vcTXEPrctDtPrintDate 2564	Text des Seite einrichten -Dialogs: &Druckdatum
vcTXEPrctDtPrintingPage 2556	Text im Druck-Info-Fenster: Seite %1 von %2 wird gedruckt auf
vcTXEPrctDtReduceExpand 2507	Text des Seite einrichten -Dialogs: Zoomfaktor
vcTXEPrctDtRepeatTable 2565	Text des Seite einrichten -Dialogs für Balkenpläne: Titel/ Tabelle/ Zeitskala /Legende wiederholen
vcTXEPrctDtRight 2522	Text des Seite einrichten -Dialogs: Rechts
vcTXEPrctDtScaling 2527	Text des Seite einrichten -Dialogs: Skalierung
vcTXEPrctDtScalingMode 2578	Text des Seite einrichten -Dialogs: &Modus:
vcTXEPrctDtStatusBarCurrentValues 2586	Statuszeilentext des Druckvorschau- Dialogs: %1 Seiten in %2 Zeilen und %3 Spalten
vcTXEPrctDtStatusBarSelectedPage 2587	Statuszeilentext des Druckvorschau- Dialogs: Seite %1 selektiert (in Zeile %2, Spalte %3)
vcTXEPrctDtTableColumnRange 2575	Text des Seite einrichten -Dialogs: Tabellenspalten (1-5;7)
vcTXEPrctDtTimeColumnEnd 2590	Text des Seite einrichten -Dialogs: Zeitskalenende:
vcTXEPrctDtTimeColumnStart 2589	Text des Seite einrichten -Dialogs: Zeitskalenstart:
vcTXEPrctDtTop 2519	Text des Seite einrichten -Dialogs: Oben
vcTXEPrctDtZoomFactor 2579	Text des Seite einrichten - Dialogs: &Zoomfaktor:
vcTXEPrctMtAdjustBottomAndTopMargin 2437	Meldungstext: Der untere Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. l.r.n Außerdem wird der obere Rand auf %2 cm reduziert.
vcTXEPrctMtAdjustLeftAndRightMargin 2434	Meldungstext: Der linke Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. l.r.n Außerdem wird der rechte Rand auf %2 cm reduziert.
vcTXEPrctMtAdjustRightAndLeftMargin 2435	Meldungstext: Der rechte Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. l.r.n Außerdem wird der linke Rand auf %2 cm reduziert.
vcTXEPrctMtAdjustTopAndBottomMargin 2436	Meldungstext: Der obere Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. l.r.n Außerdem wird der untere Rand auf %2 cm reduziert.
vcTXEPrctMtBottomMargin 2409	Meldungstext: Der untere Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.
vcTXEPrctMtIncompatibleVcVersion 2414	Meldungstext: VcVersion inkompatibel

	vcTXEPrctMtLeftMargin 2406	Meldungstext: Der linke Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.
	vcTXEPrctMtPrinterNotInstalled 2411	Meldungstext: Kein Drucker installiert
	vcTXEPrctMtPrintingNotPossible 2402	Meldungstext: Drucken z. Zt nicht möglich
	vcTXEPrctMtRightMargin 2408	Meldungstext: Der rechte Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.
	vcTXEPrctMtSelectPaperSize 2413	Meldungstext: Gewählte Blattgröße zu klein
	vcTXEPrctMtTopMargin 2407	Meldungstext: Der obere Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.
	vcTXEPrctMtValueOutOfRange 2404	Meldungstext: Außerhalb des Wertebereichs %1 bis %2
	vcTXEPrctMtWillBeAdjustedTo 2410	Meldungstext: Wird korrigiert auf
	vcTXERelTypeLongFF 3001	Text im Dialog Verbindungen bearbeiten: Ende-Ende (FF)
	vcTXERelTypeLongFS 3000	Text im Dialog Verbindungen bearbeiten: Ende-Anfang (FS)
	vcTXERelTypeLongSF 3003	Text im Dialog Verbindungen bearbeiten: Anfang-Ende (SF)
	vcTXERelTypeLongSS 3002	Text im Dialog Verbindungen bearbeiten: Anfang-Anfang (SS)
	vcTXERibAM 2225	Text im Skalenstreifen für vormittags
	vcTXERibCW 2223	Text im Skalenstreifen für Kalenderwoche
	vcTXERibDay0 2212	Text im Skalenstreifen für Montag
	vcTXERibDay1 2213	Text im Skalenstreifen für Dienstag
	vcTXERibDay2 2214	Text im Skalenstreifen für Mittwoch
	vcTXERibDay3 2215	Text im Skalenstreifen für Donnerstag
	vcTXERibDay4 2216	Text im Skalenstreifen für Freitag
	vcTXERibDay5 2217	Text im Skalenstreifen für Samstag
	vcTXERibDay6 2218	Text im Skalenstreifen für Sonntag
	vcTXERibMon0 2200	Text im Skalenstreifen für Januar
	vcTXERibMon1 2201	Text im Skalenstreifen für Februar
	vcTXERibMon10 2210	Text im Skalenstreifen für November
	vcTXERibMon11 2211	Text im Skalenstreifen für Dezember
	vcTXERibMon2 2202	Text im Skalenstreifen für März
	vcTXERibMon3 2203	Text im Skalenstreifen für April
	vcTXERibMon4 2204	Text im Skalenstreifen für Mai
	vcTXERibMon5 2205	Text im Skalenstreifen für Juni
	vcTXERibMon6 2206	Text im Skalenstreifen für Juli
	vcTXERibMon7 2207	Text im Skalenstreifen für August
	vcTXERibMon8 2208	Text im Skalenstreifen für September
	vcTXERibMon9 2209	Text im Skalenstreifen für Oktober
	vcTXERiboClock 2224	Text im Skalenstreifen für Uhr
	vcTXERibPM 2226	Text im Skalenstreifen für nachmittags
	vcTXERibQuar0 2219	Text im Skalenstreifen für 1. Quartal
	vcTXERibQuar1 2220	Text im Skalenstreifen für 2. Quartal
	vcTXERibQuar2 2221	Text im Skalenstreifen für 3. Quartal
	vcTXERibQuar3 2222	Text im Skalenstreifen für 4. Quartal
⇌ textEntry	String	Texteintrag, der den Standardtext ersetzt
	Mögliche Werte:	
		Name der Farbzuoordnungstabelle
⇌ returnStatus	Variant	Rückgabestatus

• Selektier-Modus	vcTXECtxmenArrowMode
Modus: Knoten erzeugen	vcTXECtxmenCreateNodeMode
Modus: Verbindung ziehen	vcTXECtxmenCreateLinkMode
Modus: Box erzeugen	vcTXECtxmenCreateBoxMode
Teildiagramm erstellen	vcTXECtxmenSubDiagram
Gesamtdiagramm wiederherstellen	vcTXECtxmenFullDiagram
Seite einrichten...	vcTXECtxmenPageLayout
Drucker einrichten...	vcTXECtxmenFilePrintSetup
Druckvorschau...	vcTXECtxmenFilePrintPreview
Drucken...	vcTXECtxmenFilePrint
Komplettansicht anzeigen	vcTXECtxmenShowWorldView
Legendenansicht anzeigen	vcTXECtxmenShowLegendView
Grafik exportieren...	vcTXECtxmenGraphicExport

Konstanten des Kontextmenüs für das Diagramm

Daten bearbeiten...	vcTXECtxmenEditNode
Knoten löschen	vcTXECtxmenDeleteNode
Teildiagramm erstellen	vcTXECtxmenSubDiagram
Gesamtdiagramm wiederherstellen	vcTXECtxmenFullDiagram
Höher stufen	vcTXECtxmenGroupOutlineOutdent
Tiefer stufen	vcTXECtxmenGroupOutlineIndent

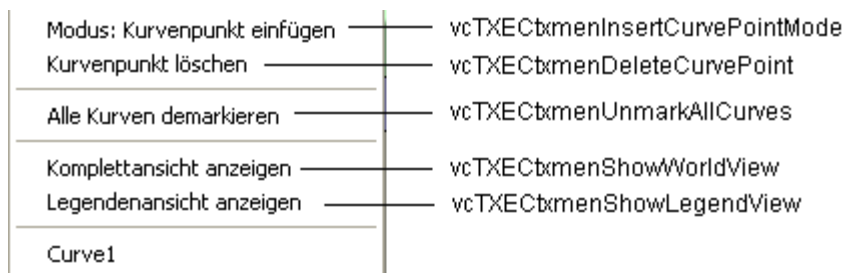
Konstanten des Kontextmenüs für Knoten

Gruppe kollabieren	vcTXECtxmenGroupCollapse
Zeilen der Gruppe kollabieren	vcTXECtxmenGroupCollapseRowsBelow
Alle Knoten in einer Zeile	vcTXECtxmenGroupNodesInOneRow
Knoten überlappend	vcTXECtxmenGroupNodesOverlaid
Gruppe löschen	vcTXECtxmenGroupDelete
Gruppendaten bearbeiten...	vcTXECtxmenEditGroup

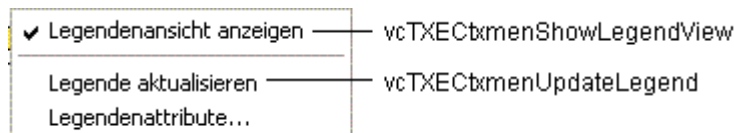
Konstanten des Kontextmenüs für Gruppen bei expandierten Gruppen

Gruppe expandieren	vcTXECtxmenGroupExpand
Zeilen der Gruppe expandieren	vcTXECtxmenGroupExpandRowsBelow
Alle Knoten in einer Zeile	
Knoten optimiert	vcTXECtxmenGroupNodesOptimized
Gruppe löschen	
Gruppendaten bearbeiten...	

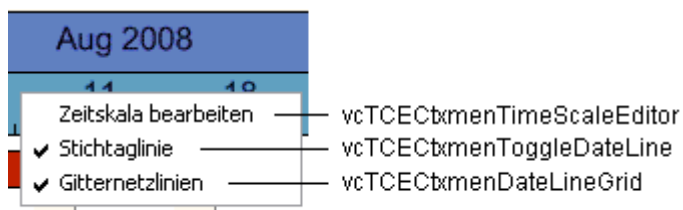
Konstanten des Kontextmenüs für Gruppen bei kollabierten Gruppen



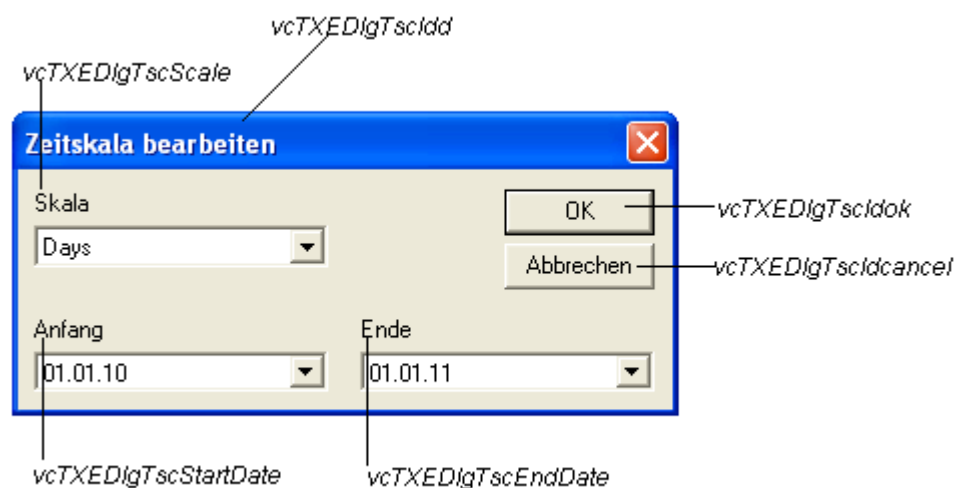
Konstanten des Kontextmenüs für Histogramme



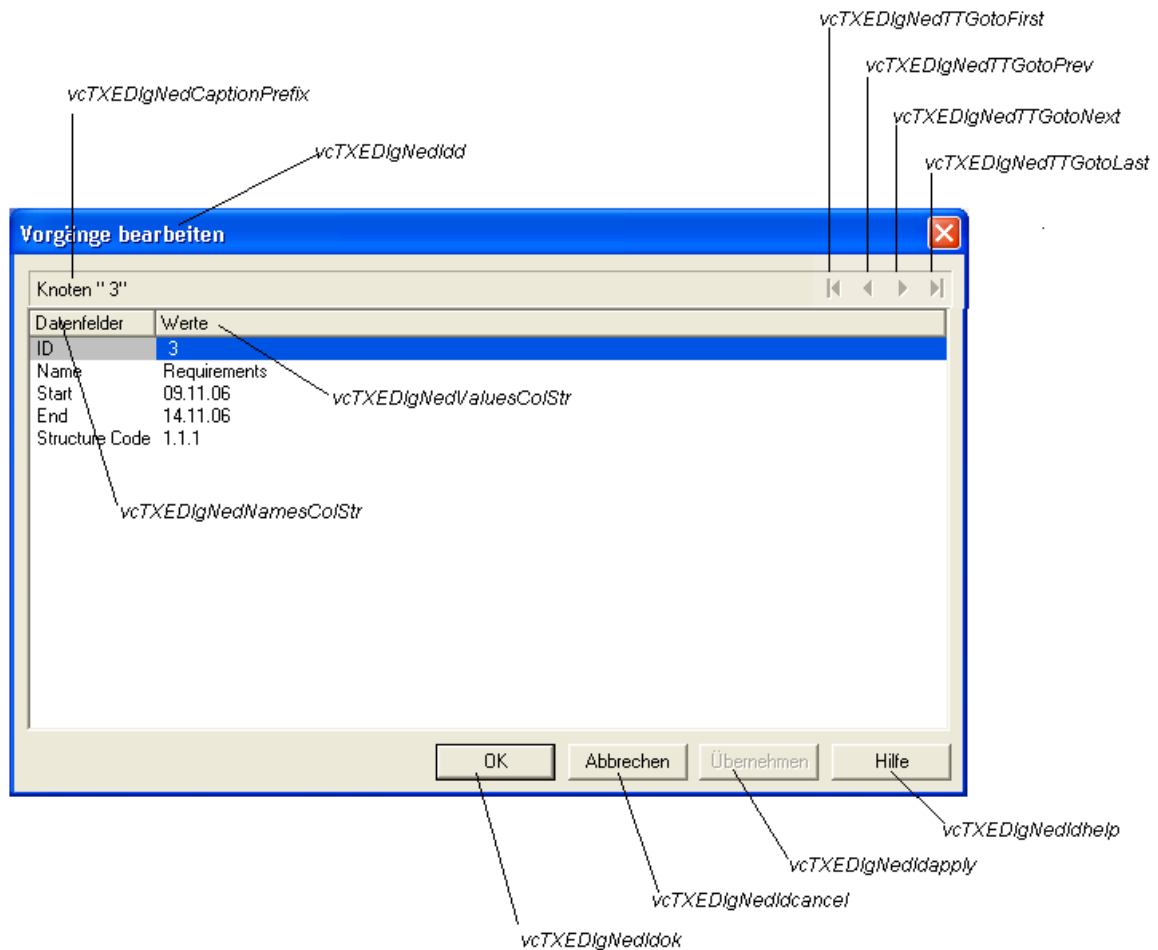
Konstanten des Kontextmenüs für die Legende



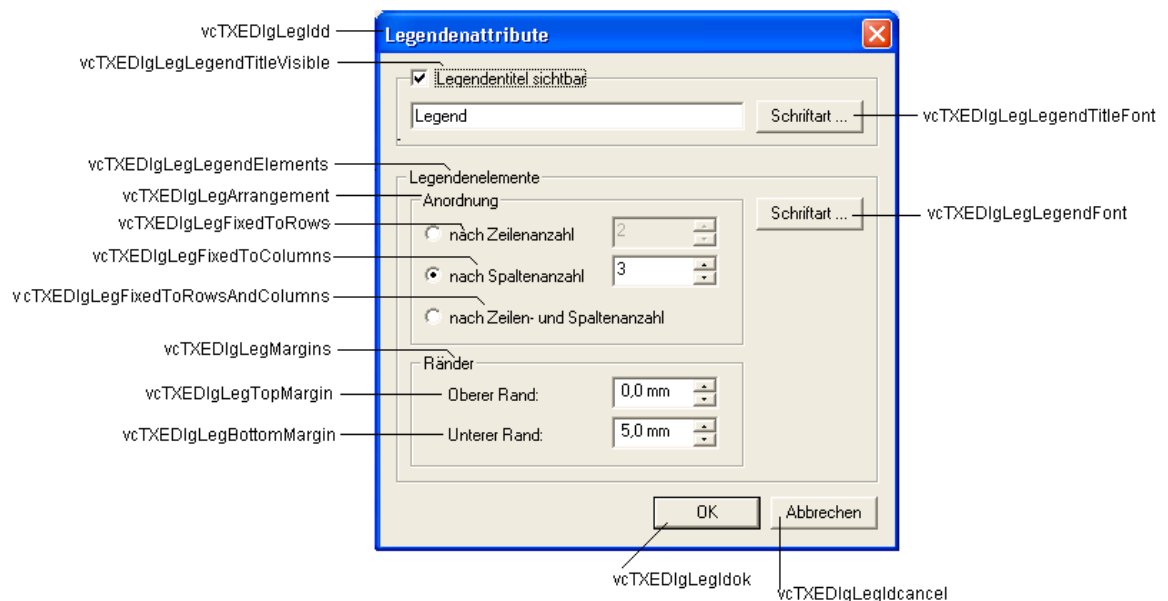
Konstanten des Kontextmenüs für die Zeitskala



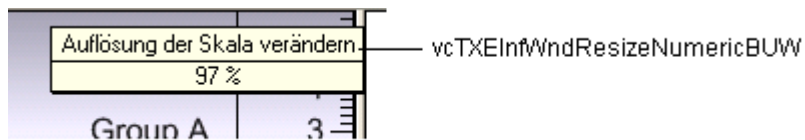
Konstanten des Dialogs **Zeitskala bearbeiten**



Konstanten der Dialoge **Vorgänge bearbeiten**, **Verbindung bearbeiten** und **Gruppierung bearbeiten**, hier am Beispiel des Dialogs **Vorgänge bearbeiten** dargestellt



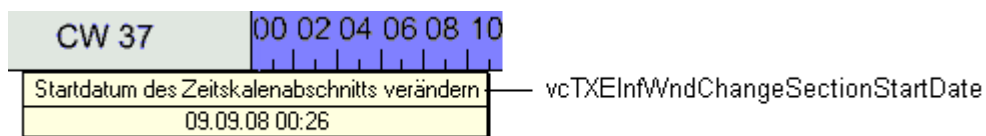
Konstanten des Dialogs **Legendenattribute**



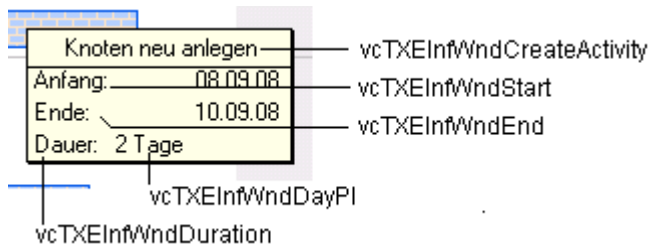
Konstante des Tooltip-Textes, der erscheint, wenn die Größe der Grundeinheit (Basic Unit Width) der **numerischen Skala im Histogramm** verändert wird.



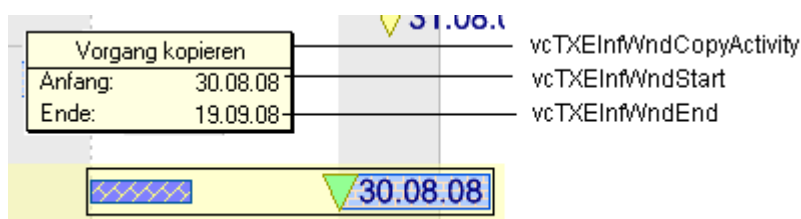
Konstanten der Tooltip-Textes, der erscheint, wenn die **Auflösung des Zeitskalenabschnitts** verändert wird



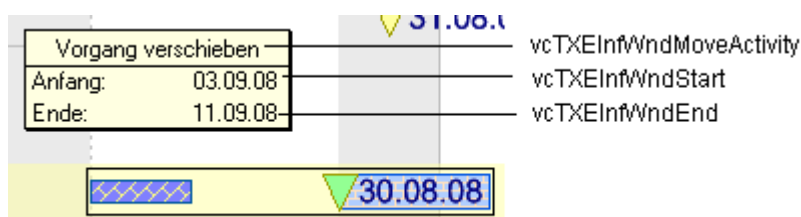
Konstanten des Tooltip-Textes, der erscheint, wenn das **Anfangsdatum eines Zeitskalenabschnitts** verändert wird



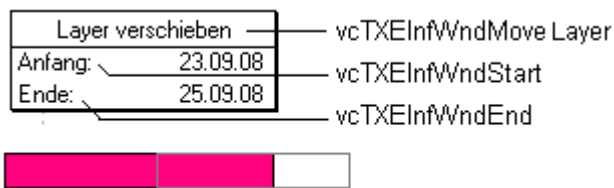
Konstanten des Tooltip-Textes, der erscheint, wenn ein **neuer Knoten** angelegt wird



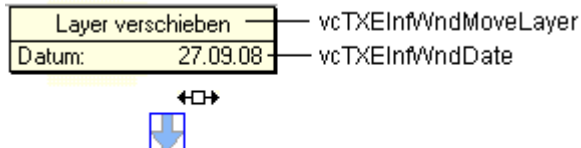
Konstanten des Tooltip-Textes, der erscheint, wenn ein **Knoten kopiert** wird



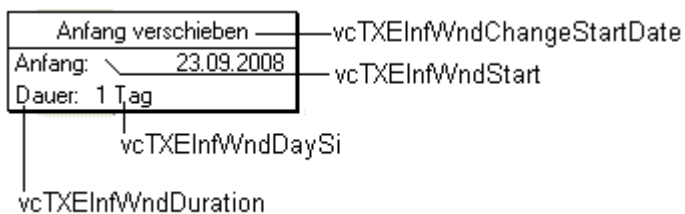
Konstanten des Tooltip-Textes, der beim **Verschieben eines Knotens** erscheint



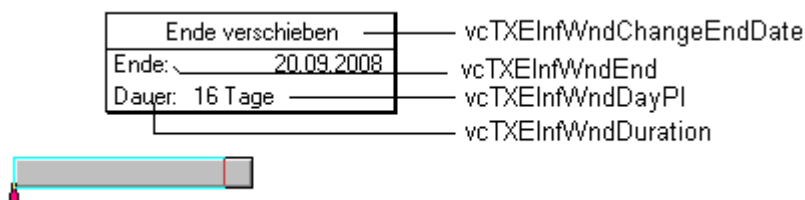
Konstanten des Tooltip-Textes, der beim **Verschieben eines Layers** erscheint



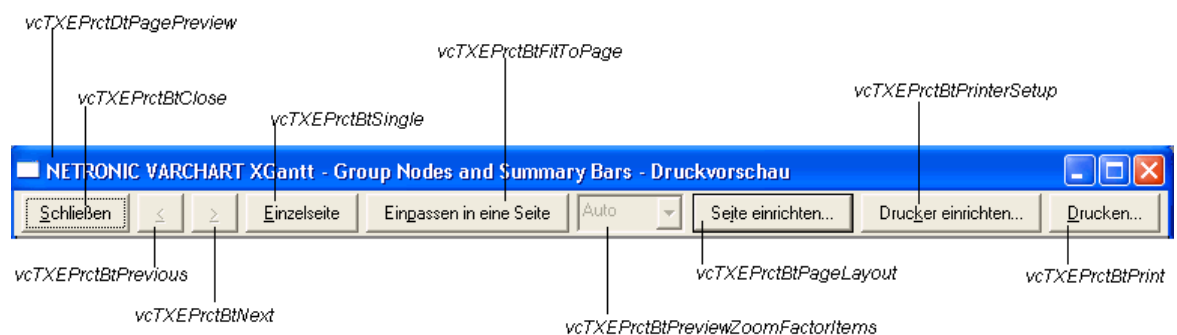
Konstanten des Tooltip-Textes, der beim **Verschieben eines Symbolayers** erscheint



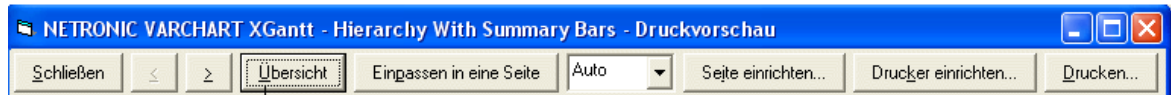
Konstanten des Tooltip-Textes, der erscheint, wenn der **Anfang eines Knotens** verschoben wird



Konstanten des Tooltip-Textes, der erscheint, wenn das **Ende eines Knotens** verschoben wird



Konstanten der Tastenbeschriftungen in der **Druckvorschau im Übersichtsmodus**



vcTXEPrctBtAll

Konstanten der Tastenbeschriftungen in der **Druckvorschau im Einzelansichtsmodus**



vcTXEPrctBtZoomPrint

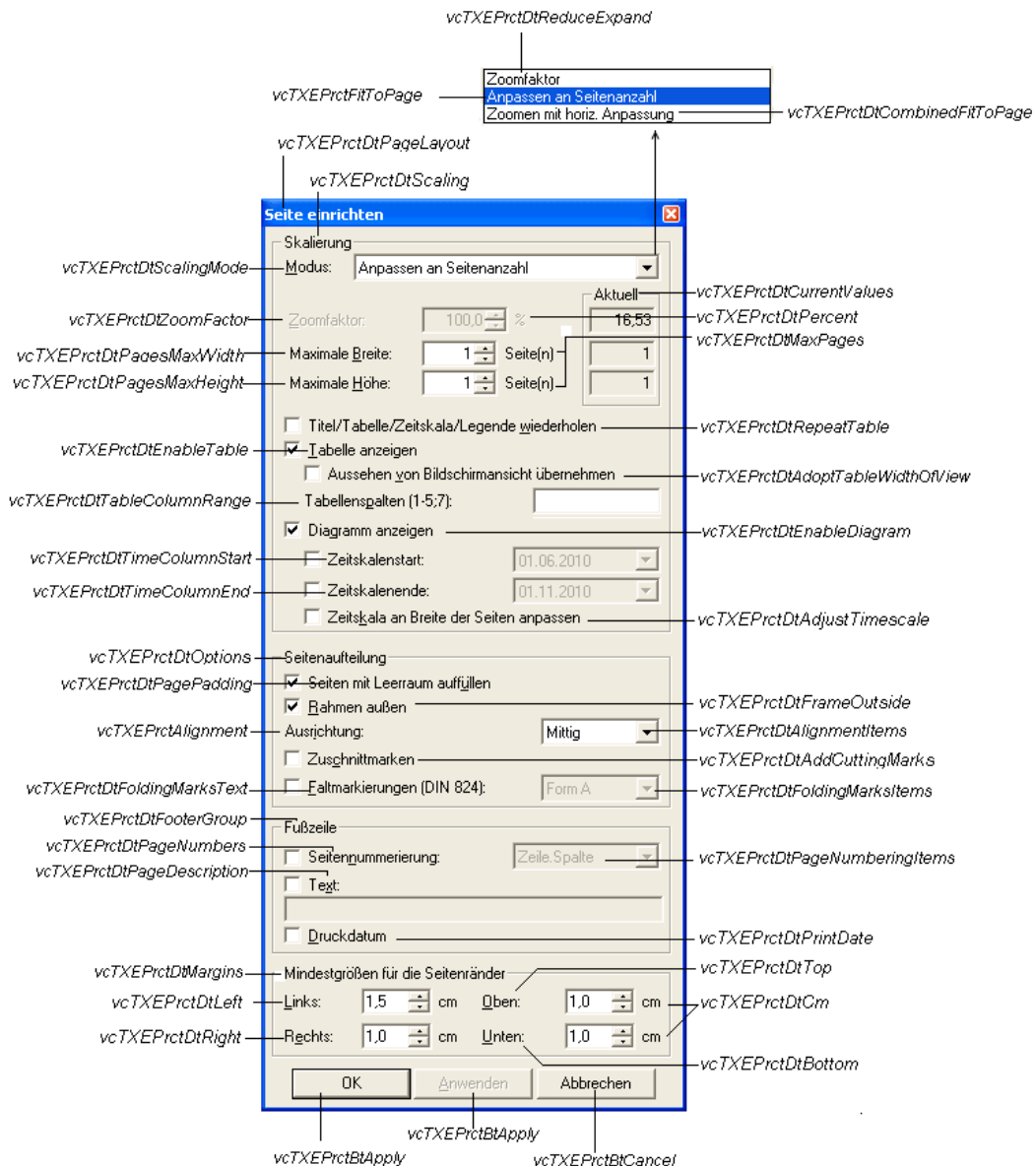
Konstanten der Tastenbeschriftungen in der **Druckvorschau im Einzelansichtsmodus** bei interaktiv ausgewählten Ausschnitt



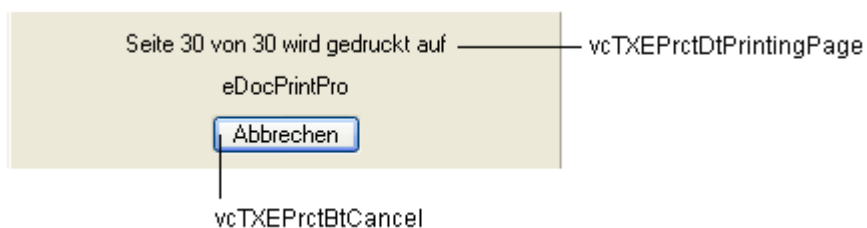
vcTXEPrctDtStatusBarSelectedPage

vcTXEPrctDtStatusBarCurrentValues

Konstanten der Statuszeile im Dialog **Druckvorschau**



Konstanten des Dialogs **Seite einrichten**



Konstanten der Infobox **Drucken**

Code-Beispiel

```
Private Sub VcGantt1_OnSupplyTextEntry(ByVal controlIndex As _
                                     VcGanttLib.TextEntryIndexEnum, _
                                     TextEntry As String, _
                                     returnStatus As Variant)
    Select Case controlIndex
    Case vcTXEPrctBtNext
        TextEntry = "Next page"
    Case vcTXEPrctBtPrevious
        TextEntry = "Previous page"
    Case vcTXEErrTxtInvalidNodePosition
        TextEntry = " Invalid Node Position !"
    End Select
End Sub
```

OnSupplyTextEntryAsVariant

Ereignis von VcGantt

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **OnSupplyTextEntry**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↩) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

OnTableCaptionLClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender in den Titelbereich der Tabelle mit der linken Maustaste klickt. Das getroffene Tabellen-Objekt wird zusammen mit der Spaltennummer und der Cursorposition (x,y-Koordinaten) als Parameter zurückgegeben. Wenn die Darstellung weder gruppiert noch hierarchisch ist, so werden die Vorgänge nach der getroffenen Tabellenspalte sortiert.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Getroffene Tabelle
⇒ columnNumber	Long	Index der getroffenen Tabellenspalte
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnTableCaptionLClick(ByVal Table As _
```

```

VcGanttLib.VcTable, ByVal columnNumber _
As Long, ByVal x As Long, ByVal y As Long, _
returnStatus As Variant)

For i = 1 To 2
    VcGantt1.Table.FormatCollection.FirstFormat.FieldBackgroundColor(i) = _
        RGB(222, 211, 33)
Next i

End Sub

```

OnTableCaptionLDbIcClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender in den Titelbereich der Tabelle mit der linken Maustaste doppelt klickt. Das getroffene Table-Objekt wird zusammen mit der Spaltennummer und der Cursorposition (x,y-Koordinaten) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Getroffene Tabelle
⇒ columnNumber	Long	Index der getroffenen Tabellenspalte
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

Private Sub VcGantt1_OnTableCaptionLDbIcClick(ByVal Table As _
    VcGanttLib.VcTable, ByVal columnNumber _
    As Long, ByVal x As Long, _
    ByVal y As Long, _
    returnStatus As Variant)

    VcGantt1.Table.Visible = False

End Sub

```

OnTableCaptionRcClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender in den Titelbereich der Tabelle mit der rechten Maustaste klickt. Das getroffene Table-Objekt wird zusammen mit der Spaltennummer und der Cursorposition (x,y-Koordinaten) als Parameter zurückgegeben. Durch das Setzen des Rückgabestatus können Sie an der übergebenen Position ein eignes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Getroffene Tabelle
⇒ columnNumber	Long	Index der getroffenen Tabellenspalte
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menü erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnTableCaptionRClick(ByVal Table As _
                                         VcGanttLib.VcTable, ByVal columnNumber _
                                         As Long, ByVal x As Long, _
                                         ByVal y As Long, _
                                         returnStatus As Variant)

    'start a popup menu at the current cursor position
    PopupMenu mnuTableCaptionPopup

End Sub
```

OnTableColumnWidth

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender die Breite einer Tabellenspalte interaktiv verändert. Die Tabelle, der Index der geänderten Spalte und die neue Spaltenbreite (in 1/100 mm) werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Tabelle
⇒ index	Integer	Index der geänderten Spalte
	Mögliche Werte:	Datenfeldindex
⇒ currentWidth	Long	Neue Spaltenbreite
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Spaltenbreite wird nicht geändert. Die Spaltenbreite wird geändert.

Code-Beispiel

```

Private Sub VcGantt1_OnTableColumnWidth(ByVal Table As _
                                         VcGanttLib.VcTable, ByVal index As Integer, _
                                         ByVal currentWidth As Long, _
                                         returnStatus As Variant)

    If currentWidth > 5000 Then
        returnStatus = vcRetStatFalse
        VcGantt1.Table.ColumnWidth(index) = 5000
    End If
End Sub

```

OnTableColumnWidthModifyComplete**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender die Breite einer Tabellenspalte interaktiv verändert hat. Die Tabelle, der Index der geänderten Spalte und die neue Spaltenbreite (in 1/100 mm) werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Tabelle
⇒ index	Integer	Index der geänderten Spalte
	Mögliche Werte:	Datenfeldindex
⇒ currentWidth	Long	Neue Spaltenbreite
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Spaltenbreite wird nicht geändert. Die Spaltenbreite wird geändert.

OnTableWidth**Ereignis von VcGantt**

Dieses Ereignis tritt ein, wenn der Anwender die Tabellenbreite interaktiv verändert hat. Die Tabelle und das neue Tabelle/Diagramm-Breitenverhältnis werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Tabelle

⇒ tableWidthRatio	Long	Breitenverhältnis der Tabelle zum Gesamtdiagramm (einschließlich Tabelle)
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Tabellenbreite wird nicht geändert. Die Tabellenbreite wird geändert.

Code-Beispiel

```
Private Sub VcGantt1_OnTableWidth(ByVal Table As VcGanttLib.VcTable, _  
                                ByVal tableWidthRatio As Long, _  
                                returnStatus As Variant)  
  
    If tableWidthRatio > 30 Then  
        returnStatus = vcRetStatFalse  
        VcGantt1.TableDiagramWidthRatio = 30  
    End If  
  
End Sub
```

OnTableWidthModifyEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender die Tabellenbreite interaktiv verändert hat. Die Tabelle und das neue Tabelle/Diagramm-Breitenverhältnis werden als Parameter zurückgegeben. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

Dieses Ereignis unterscheidet sich von dem Ereignis **OnTableWidth** durch die Rückgabe des Parameters Tabelle/Diagramm-Breitenverhältnis als Datentyp "Double", mit dem eine höhere Genauigkeit erzielt wird. Die Verwendung dieses Ereignisses muss zuvor über die Eigenschaft **UseHigherTableDiagramWidthRatioPrecision** oder über die Option **Höhere Genauigkeit für die Breitenverhältnisse zwischen Tabelle(n) und Diagramm** auf der Eigenschaftenseite **Allgemeines** aktiviert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ table	VcTable	Tabelle
⇒ tableWidthRatio	Double	Breitenverhältnis der Tabelle zum Gesamtdiagramm (einschließlich Tabelle)
⇔ returnStatus	Variant	Rückgabestatus

OnTimeScaleChangeComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn das Zoomen der Zeitskala abgeschlossen ist.

	Datentyp	Beschreibung
Parameter: ⇒ timeScale	VcTimeScale	Veränderte Zeitskala

OnTimeScaleEndModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Änderung des Enddatums der Zeitskala abgeschlossen ist.

	Datentyp	Beschreibung
Parameter: ⇒ newEndDate	Date	Neues Enddatum

OnTimeScaleLClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf die Zeitskala klickt. Das getroffene TimeScale-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter: ⇒ timeScale	VcTimeScale	Getroffene Zeitskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleLClick(ByVal timeScale As _
    VcGanttLib.VcTimeScale, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant) _

VcGantt1.TimeScaleCollection.Active.BackgroundColor = RGB(225, 50, 10)

End Sub
```


OnTimeScaleLDblick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf die Zeitskala doppelt klickt. Das getroffene TimeScale-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Das integrierte Dialogfeld kann durch Setzen des Rückgabestatus unterdrückt werden.

	Datentyp	Beschreibung
Parameter:		
⇒ timeScale	VcTimeScale	Getroffene Zeitskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Der Zeitskala bearbeiten Dialog erscheint nicht. Der Zeitskala bearbeiten Dialog erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleLDblick(ByVal timeScale As _
                                         VcGanttLib.VcTimeScale, ByVal x As Long, _
                                         ByVal y As Long, returnStatus As Variant)

    ' show own "Edit Timescale" dialog
    On Error GoTo CancelError
    frmEditDialog.setTimescale timeScale
    frmEditDialog.Show Modal, Me

    returnStatus = vcRetStatFalse

Exit Sub

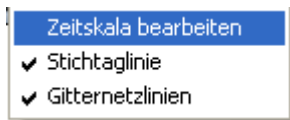
CancelError:
    returnStatus = vcRetStatFalse

End Sub
```

OnTimeScaleRClick

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf die Zeitskala klickt. Das getroffene TimeScale-Objekt wird zusammen mit der Mausposition (x,y-Koordinaten) als Parameter zurückgegeben. Durch das Setzen des Rückgabestatus können Sie an der übergebenen Position ein eignes Kontextmenü anzeigen.



Oben: integriertes Kontextmenü

	Datentyp	Beschreibung
Parameter:		
⇒ timeScale	VcTimeScale	Getroffene Zeitskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatNoPopup 4 vcRetStatOK 1	Das Kontextmenü wird unterdrückt. Das Kontext-Menu erscheint.

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleRClick(ByVal timeScale As _
                                     VcGanttLib.VcTimeScale, _
                                     ByVal x As Long, ByVal y As Long, _
                                     returnStatus As Variant)

    ' Start own popup menu at the current cursor position
    PopupMenu mnuTimescalePopup

    returnStatus = vcRetStatNoPopup
End Sub
```

OnTimeScaleSectionRescaleCompleteEx

Ereignis von VcGantt

Mit diesem Ereignis wird das Ende einer Veränderung der Auflösung in einem Zeitskalenabschnitt angezeigt. Das Zeitskalenobjekt, der Abschnittsindex und die neue Breite der Grundeinheit (basic unit width) werden übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ timeScale	VcTimeScale	Zeitskala
⇒ sectionIndex	Integer	Index des Skalenabschnitts
	Mögliche Werte: Datenfeldindex	
⇒ newBasicUnitWidth	Long	Neue Breite der Grundeinheit

OnTimeScaleSectionRescaleEx

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender einen Abschnitt der Zeitskala interaktiv skaliert. Die Zeitskala, die Nummer des Zeitskalenabschnitts und die neue Grundeinheit (**BasicUnitWidth**) werden als Parameter zurückgegeben, so dass Sie beispielsweise die Zulässigkeit der Skalierung prüfen können. Durch Setzen des Rückgabestatus können Sie die Änderung verhindern.

	Datentyp	Beschreibung
Parameter:		
⇒ timeScale	VcTimeScale	Zeitskala
⇒ sectionIndex	Integer	Index des Skalenabschnitts
	Mögliche Werte:	Datenfeldindex
⇒ newBasicUnitWidth	Long	Neue Breite der Grundeinheit
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleSectionRescaleEx(ByVal timeScale As _
    VcGanttLib.VcTimeScale, _
    ByVal sectionIndex As Integer, _
    ByVal newBasicUnitWidth As Long, _
    returnStatus As Variant)

    If newBasicUnitWidth <= 1000 Then
        MsgBox "New basic unit width: " & newBasicUnitWidth
    Else
        MsgBox "The maximum basic unit width is 1000."
        returnStatus = vcRetStatFalse
    End If

End Sub
```

OnTimeScaleSectionStartModify

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender den Anfang eines Zeitskalenabschnitts interaktiv verändert. Die Zeitskala, die Nummer des Zeitskalenabschnitts und das neue Anfangsdatum werden als Parameter zurückgegeben. Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnTimeScaleSectionStartModifyComplete**.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ timeScale	VcTimeScale	Zeitskala
⇒ sectionIndex	Integer	Index des Zeitskalenabschnitts
	Mögliche Werte:	Datenfeldindex
⇒ newStartDate	Date/Time	Datum
⇒ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	Die Veränderung wird rückgängig gemacht. Die Veränderung wird durchgeführt.

Code-Beispiel

```
Private Sub VcGantt1_OnTimeScaleSectionStartModify(ByVal timeScale As _
    VcGanttLib.VcTimeScale, _
    ByVal sectionIndex As Integer, _
    ByVal newStartDate As Date, _
    returnStatus As Variant)

    If MsgBox("Do you want to change the start of section No. " & sectionIndex _
        & " to " & newStartDate & "?", vbOKCancel) _
        = vbCancel Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnTimeScaleStartModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn die Änderung des Startdatums der Zeitskala abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇒ newStartDate	Date	Neues Startdatum

OnToolTipText

Ereignis von VcGantt

Dieses Ereignis tritt nur auf, wenn Sie die VcGantt-Eigenschaft **Show-ToolTip** auf **True** gesetzt haben bzw. auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **Tooltip anzeigen** aktiviert haben. Sie können mit Hilfe dieses Ereignisses Daten des getroffenen Objektes als

Tooltip anzeigen. Das Ereignis tritt auf, sobald der Cursor auf ein VcGantt-Objekt bewegt wird. Es gibt das getroffene Objekt, den Objekttyp und die Koordinaten des Cursors zurück. Durch Setzen des Returnstatus auf **vcRetStatFalse** können Sie den Tooltip an dieser Stelle unterdrücken.

Bei einem Kalendergitter wird ein Tooltip-Text nur dann angefordert, wenn das Gitter identifizierbar ist, d.h. die Kalendergitter-Eigenschaft **Identifiable** auf **True** gesetzt wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ hitObject	Object	Getroffenes Objekt
⇒ hitObjectType	VcObjectTypeEnum	Typ des getroffenen Objektes
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeCalendarGrid 18 vcObjTypeCurve 12 vcObjTypeDateLine 9 vcObjTypeGroup 7 vcObjTypeGroupInDiagram 11 vcObjTypeGroupInTable 7 vcObjTypeHistogram 13 vcObjTypeLayer 8 vcObjTypeLinkCollection 3 vcObjTypeNodeInDiagram 2 vcObjTypeNodeInLegend 17 vcObjTypeNodeInTable 1 vcObjTypeNone 0 vcObjTypeNumericScale 10 vcObjTypeSummaryNode 14 vcObjTypeTable 4 vcObjTypeTableCaption 5 vcObjTypeTimeScale 6	Objekttyp Box Objekttyp Kalendergitter Objekttyp Kurve Objekttyp Stichtaglinie Objekttyp Gruppe Objekttyp Gruppe im Knotenbereich Objekttyp Gruppe im Tabellenbereich Objekttyp Histogramm Objekttyp Layer Objekttyp LinkCollection Objekttyp Knoten im Knotenbereich Objekttyp Knoten im Legendenbereich Objekttyp Knoten im Tabellenbereich kein Objekt Objekttyp Werteskala Objekttyp Summenbalken Objekttyp Tabelle Objekttyp Tabellenüberschrift Objekttyp Zeitskala
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers
⇐ ToolTipText	String	Tooltip Text, kann maximal 1024 Zeichen lang sein
	Mögliche Werte:	Name der Farbzuoordnungstabelle
⇐ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcGantt1_OnToolTipText(ByVal hitObject As Object, _
    ByVal hitObjectType As _
    VcGanttLib.VcObjectTypeEnum, ByVal x As Long, _
    ByVal y As Long, ToolTipText As String, _
    returnStatus As Variant)

    Select Case hitObjectType
        Case vcObjTypeNodeInDiagram
            ToolTipText = hitObject.DataField(1)

        Case Else
```

```
End Select
End Sub
```

OnToolTipTextAsVariant

Ereignis von VcGantt

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **OnToolTipText**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch $\leftrightarrow$) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

Code-Beispiel

```
Private Sub VcGantt1_OnToolTipTextAsVariant(ByVal hitObject As Object, ByVal
hitObjectType As VcGanttLib.VcObjectTypeEnum, ByVal x As Long, ByVal y As Long,
ToolTipText As Variant, returnStatus As Variant)
    Select Case hitObjectType
        Case VcObjectTypeEnum.vcObjTypeNodeInDiagram
            ToolTipText = hitObject.DataField(0)
    End Select
End Sub
```

OnViewComponentsSizeModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt auf, wenn zur Laufzeit die Größe eines grafischen Grundelements des VARCHART-ActiveX-Steuerelements (Zeitskala, Diagramm, Histogramm, Tabelle, Tabellenüberschrift usw.) verändert wurde. Sie können nun über die Schnittstelle darauf reagieren, indem Sie die Position und Größe aller grafischen Grundelemente abfragen. Dabei ist folgendes zu beachten:

1. Die Position bezieht sich auf den Nullpunkt des VARCHART-XGantt-Steuerelements.
2. Die Werte werden in Pixel zurückgegeben.

	Datentyp	Beschreibung
Parameter: $\Rightarrow$ (no parameter)		

Code-Beispiel

```
Private Sub VcGantt1_OnViewComponentsSizeModifyComplete()
    Dim x As Long
    Dim y As Long
    Dim width As Long
```

```

Dim height As Long
Dim scMod As Long
scMod = ScaleMode
ScaleMode = vbPixels

VcGantt1.GetViewComponentSize vcHistogramVerScaleComponent, x, y, _
    width, height

' plus 6 because of the sash
Text1.Top = VcGantt1.Top + y + 6
Text1.Left = VcGantt1.Left + x
' minus 25 because of the numeric scale
Text1.Width = width - 25
' minus 6 because of the sash
Text1.Height = height - 6
ScaleMode = scMod
End Sub

```

OnWorldViewClosed

Ereignis von VcGantt

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Komplettansicht geschlossen wird.

	Datentyp	Beschreibung
Parameter: ⇐ (no parameter)		

Code-Beispiel

```

Private Sub VcGantt1_OnWorldViewClosed()
    MsgBox "Do you want to close the worldview window?", vbOKCancel
End Sub

```

OnZoomFactorModifyComplete

Ereignis von VcGantt

Dieses Ereignis tritt ein, wenn der Anwender auf markierte Objekte gezoomt hat. Sie können stufenlos zoomen, indem Sie bei gedrückter Strg-Taste das Mausrad drehen. In bestimmten Schritten können Sie zoomen, indem Sie bei gedrückter **Strg**-Taste die **Plus**- bzw. **Minus**-Tasten des Ziffernblocks der Tastatur drücken.

	Datentyp	Beschreibung
Parameter: ⇐ (no parameter)		

Code-Beispiel

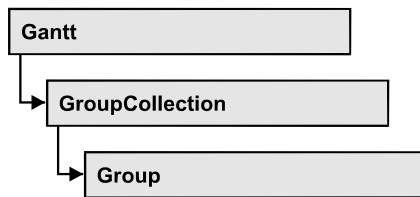
```

Private Sub VcGantt1_OnZoomFactorModifyComplete()
    MsgBox "Zoomfactor: " & VcGantt1.ZoomFactor

```

End Sub

7.38 VcGroup



Eine Gruppe enthält alle Knoten, die im Gruppierfeld denselben Wert haben. Dieser Wert kann als Gruppenname abgefragt werden. Auf die Knoten, die eine Gruppe bilden, können Sie über die Eigenschaft **NodeCollection** zugreifen.

Eigenschaften

- BodyCollapsed
- DataField
- GroupingLevel
- GroupInvisible
- ID
- MarkGroup
- Name
- NodeCollection
- NodesInHeader
- NodesOverlaid
- RowsBelowCollapsed
- SubGroups
- SuperGroup
- Visible

Methoden

- DataRecord
- DeleteGroup
- RelatedDataRecord
- ReOptimizeNodes
- UpdateGroup

Eigenschaften

BodyCollapsed

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob eine Gruppe kollabiert (True) oder expandiert (False) ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppe kollabiert/expandiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub VcGantt1_OnGroupLClick(ByVal group As VcGanttLib.VcGroup, _
                                   ByVal x As Long, ByVal y As Long, _
                                   returnStatus As Variant)

    If body.Collapsed = False Then
        body.Collapsed = True
    Else
        body.Collapsed = False
    End If

End Sub
```

DataField

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie den Inhalt eines Datenfeldes für den Datensatz der Gruppe festlegen oder erfragen. Der Gruppendatensatz ist eine Kopie des Datensatzes des ersten Vorganges, der in die Gruppe eingefügt wurde. Das Datenfeld wird über seinen Index angesprochen. Anschließend aktualisieren Sie die Gruppe über die Methode **UpdateGroup**.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Datenfelds
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Void	

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set groupCltn = VcGantt1.groupCollection

For Each group In groupCltn
    Set nodeCltn = group.nodeCollection

    For Each node In nodeCltn
        If node.DataField(3) > group.DataField(3) Then
            group.DataField(3) = node.DataField(3)
        End If
    Next node

    group.UpdateGroup
Next group
```

GroupingLevel

Nur-Lese-Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie die Gruppierenebene der Gruppe bei mehrstufiger Gruppierung erfragen. Es sind maximal 25 Gruppierungsebenen möglich.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Gruppierenebene der Gruppe
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim group As VcGroup
Dim superGroup As VcGroup

Dim nodeCollection As VcNodeCollection
Dim node As VcNode

Set nodeCollection = VcGantt1.nodeCollection
Set node = nodeCollection.FirstNode
Set group = node.SuperGroup

If group.GroupingLevel > 0 Then
    Set superGroup = group.superGroup
End If
```

GroupInvisible

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob diese Gruppe angezeigt wird. Der Standardwert ist der für die Gruppierungsebene festgelegte Wert.

	Datentyp	Beschreibung

ID

Nur-Lese-Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie die ID einer Gruppe erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Gruppen-ID Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim groupName As String
Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup
groupID = group.ID
MsgBox group.ID
```

MarkGroup

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob eine Gruppe markiert ist.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcNode
Set nodeCltn = VcGantt1.nodeCollection
nodeCltn.SelectNodes (vcSelected)
```

968 API-Referenz: VcGroup

```
For Each node In nodeCltn
    Group.MarkGroup = False
Next node
```

Name

Nur-Lese-Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie den Namen der Gruppe (= Wert des Gruppierfeldes GroupField) erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Gruppenname
	Mögliche Werte:	Name der Farbzustimmungstabelle

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim groupName As String

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup

groupName = group.Name
```

NodeCollection

Nur-Lese-Eigenschaft von VcGroup

Über diese Eigenschaft haben Sie Zugriff auf alle Knoten, die zu einer Gruppe gehören.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	NodeCollection Objekt

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim nodeCltn As VcNodeCollection

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup
Set nodeCltn = group.NodeCollection
```

NodesInHeader

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Knotenobjekte der Gruppe alle in derselben Zeile liegen (True) oder nicht (False).

	Datentyp	Beschreibung

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName("A")

group.NodesInHeader = True
```

NodesOverlaid

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das Knotenlayout optimiert ist (False) oder ob Knoten überlappen (True).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Das Knotenlayout ist/ist nicht optimiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

RowsBelowCollapsed

Eigenschaft von VcGroup

Diese Eigenschaft wirkt nur auf der 1. bis (n-1). Ebene bei mehrstufiger Gruppierung (n Ebenen). Wenn für die Gruppe alle Knoten in einer Zeile dargestellt werden, bewirkt diese Eigenschaft (True), dass nur die vorhandenen Untergruppen in der gewählten Gruppe ausgeblendet werden. Wenn Sie stattdessen diese Gruppe mit der Eigenschaft **Collapsed** kollabieren, so werden zusätzlich die Vorgänge in der Gruppe, die zu keiner Untergruppe gehören, überlappend dargestellt.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zeilen unterhalb der obersten sind/sind nicht kollabiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName("A")

group.RowsBelowCollapsed = True
```

SubGroups**Nur-Lese-Eigenschaft von VcGroup**

Mit dieser Eigenschaft können Sie bei mehrstufiger Gruppierung Untergruppen erfragen, die in einer Group-Auflistung zurückgeliefert werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroupCollection	GoupCollection-Objekt, das die Untergruppen enthält

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim subGroupCltn As VcGroupCollection

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName("A")

Set subGroupCltn = group.SubGroups
```

SuperGroup**Nur-Lese-Eigenschaft von VcGroup**

Mit dieser Eigenschaft können Sie bei mehrstufiger Gruppierung die Obergruppe der aktuellen Gruppe erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroup	Obergruppe

Code-Beispiel

```

Dim group As VcGroup
Dim superGroup As VcGroup

Dim nodeCollection As VcNodeCollection
Dim node As VcNode

Set nodeCollection = VcGantt1.NodeCollection
Set node = nodeCollection.FirstNode
Set group = node.SuperGroup

If group.GroupingLevel > 0 Then
    Set superGroup = group.SuperGroup
End If

```

Visible**Eigenschaft von VcGroup**

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die aktuelle Gruppe sichtbar (True) oder unsichtbar (False) sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppe sichtbar/unsichtbar
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName("A")

group.Visible = False

```

Methoden**DataRecord****Methode von VcGroup**

Mit dieser Eigenschaft können Sie die Gruppe als Datensatzobjekt erfragen. Über die Eigenschaften des Datensatzobjektes haben Sie auch Zugriff auf die entsprechende Datentabelle und Tabellenauflistung.

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Zurückgegebener Datensatz

DeleteGroup

Methode von VcGroup

Mit dieser Methode können Sie eine Gruppe löschen. Das Löschen einer Gruppe ist nur möglich, wenn kein Knoten mehr in der Gruppe ist. Andernfalls müssen Sie zuvor alle Vorgänge aus der Gruppe löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Gruppe erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName ("A")
Set nodeCltn = group.NodeCollection

For Each node In nodeCltn
    node.DeleteNode
Next node

group.DeleteGroup
```

RelatedDataRecord

Methode von VcGroup

Mit dieser Methode können Sie einen Datensatz aus einer verknüpften Tabelle erfragen, der dem Datensatz der Gruppentabelle zugeordnet ist. Der im Parameter übergebene Index bezeichnet das Feld im Datensatz, in dem der Schlüssel des zugeordneten Datensatzes steht.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Datenfeldes, das den Schlüssel enthält
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcDataRecord	Zurückgegebener zugeordneter Datensatz

ReOptimizeNodes

Methode von VcGroup

Ist die Eigenschaft **VcGantt.GroupOptimizationOnInteractionsEnabled** ausgeschaltet und die Knoten der Gruppe sind grundsätzlich optimiert dargestellt, dann kann hierüber nach einer vorangegangenen Interaktion eine erneute Optimierung der Gruppe erreicht werden.

	Datentyp	Beschreibung
Rückgabewert	Void	

UpdateGroup

Methode von VcGroup

Mit dieser Methode aktualisieren Sie eine Gruppe nach der Veränderung eines Datenfeldes mit der Eigenschaft **DataField**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Gruppe erfolgreich (True) / nicht erfolgreich (False) aktualisiert

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

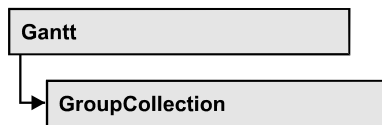
Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName("A")
Set nodeCltn = group.NodeCollection

group.DataField(3) = nodeCltn.FirstNode.DataField(3)

For Each node In nodeCltn
    If node.DataField(3) > group.DataField(3) Then
        group.DataField(3) = node.DataField(3)
    End If
Next node

group.UpdateGroup
```

7.39 VcGroupCollection



In einem Objekt vom Typ `VcGroupCollection` sind die bestehenden Gruppen zusammengefasst, sofern Knoten gruppiert worden sind. Über **For Each group In GroupCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Gruppen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaft **GroupByName**. Die Anzahl der im Auflistungsobjekt vorhandenen Gruppen kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `FirstGroup`
- `GroupByName`
- `NextGroup`
- `SelectGroups`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcGroupCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Gruppenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim group As VcGroup

For Each group In VcGantt1.GroupCollection
    Debug.Print group.Name
Next
```

Count**Nur-Lese-Eigenschaft von VcGroupCollection**

Mit dieser Eigenschaft kann die Anzahl der Gruppen in der Group-Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Gruppen

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim numberOfGroups As Integer

Set groupCltn = VcGantt1.GroupCollection
numberOfGroups = groupCltn.Count
```

Methoden**FirstGroup****Methode von VcGroupCollection**

Mit dieser Methode können Sie auf den Initialwert, d. h. die erste Gruppe der Gruppenuflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextGroup** über die nachfolgenden Gruppen zu iterieren. Existiert keine Gruppe in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcGroup	erste Gruppe der GroupCollection

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
```

```
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup
```

GroupByName

Methode von VcGroupCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Gruppe zugreifen. Existiert keine Gruppe unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇒ Rückgabewert	VcGroup	Gruppe
⇒ groupName	String	Gruppenbezeichnung
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcGroup	Gruppe

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.GroupByName ("Group A")
```

NextGroup

Methode von VcGroupCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Gruppen der Group-Auflistung zugreifen, nachdem Sie mit der Methode **FirstGroup** den Initialwert erfasst haben. Sind alle Gruppen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcGroup	Nachfolgende Gruppe

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup

While Not group Is Nothing
```

```

        List1.AddItem group.Name
        Set group = groupCltn.NextGroup
    Wend

```

SelectGroups

Methode von VcGroupCollection

Mit dieser Methode können Sie festlegen, welche Gruppen in der Gruppen-Auflistung verfügbar sein sollen.

	Datentyp	Beschreibung
Parameter: ⇒ groupSelType	GroupSelectionTypeEnum Mögliche Werte: vcAllGroups 0 vcCollapsedGroups 1 vcExpandedGroups 2 vcInvisibleGroups 5 vcSelectedGroups 3 vcVisibleGroups 4	Auszuwählender Gruppentyp alle Gruppen ausgewählt kollabierte Gruppen ausgewählt expandierte Gruppen ausgewählt unsichtbare Gruppen ausgewählt ausgewählte Gruppen ausgewählt sichtbare Gruppen ausgewählt
Rückgabewert	Long	Anzahl der ausgewählten Gruppen

Code-Beispiel

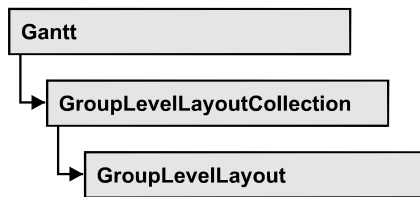
```

Dim groupCltn As VcGroupCollection

Set groupCltn = VcGantt1.GroupCollection
groupCltn.SelectGroups (vcAllGroups)

```

7.40 VcGroupLevelLayout



Ein Objekt vom Typ VcGroupLevelLayout legt Inhalt und Erscheinungsbild von Gruppierungsebenen fest. Dazu dienen der Name der Gruppierungsebene, Ebenennummer, das Feld, nach dem gruppiert wird, Sortierung und Sortierreihenfolge sowie verschiedene Optionen zum Design von Kalender- und Liniengittern sowie Trennlinien.

Eigenschaften

- AllowVerticalGroupMovementViaDiagram
- AllowVerticalGroupMovementViaTable
- AutoCollapseGroups
- AutoExpandTargetGroup
- BodiesCollapsed
- BodiesCollapsedDataFieldIndex
- BodiesCollapsedMapName
- CalendarGridName
- CalendarGridsWithChildGroups
- CalendarNameDataFieldIndex
- DateLineGridName
- DateLineGridsWithChildGroups
- DateLineName
- DateLinesWithChildGroups
- GroupDataFieldIndex
- GroupsInvisible
- GroupsInvisibleCollapsedMapName
- GroupsInvisibleDataFieldIndex
- Level
- ModificationsAllowed
- Name
- NodesInHeaders
- NodesInHeadersDataFieldIndex
- NodesInHeadersMapName
- NodesOverlaid
- OptimizedNodesSortDataFieldIndex

- OptimizedNodesSortOrder
- OverlaidNodesSortDataFieldIndex
- OverlaidNodesSortOrder
- PagebreakMode
- RestoreAutoCollapsedGroups
- RestoreAutoExpandedGroups
- RowBackColorAsARGB
- RowBackColorDataFieldIndex
- RowBackColorMapName
- RowPattern
- RowPatternColorAsARGB
- RowPatternColorDataFieldIndex
- RowPatternColorMapName
- RowPatternDataFieldIndex
- RowPatternMapName
- SeparationLineColor
- SeparationLineColorDataFieldIndex
- SeparationLineColorMapName
- SeparationLineThickness
- SeparationLineType
- ShowCalendarGrids
- ShowDateLineGrids
- ShowDateLines
- ShowGroupNodes
- ShowSeparationLines
- ShowSeparationLinesAtTop
- SortDataFieldIndex
- SortOrder
- Specification
- SummaryBarsVisible
- Visible

Eigenschaften

AllowVerticalGroupMovementViaDiagram

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das vertikale Verschieben von Gruppen im Diagramm zugelassen ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vertikales Verschieben von Gruppen im Diagramm zugelassen/nicht zugelassen Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AllowVerticalGroupMovementViaTable

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das vertikale Verschieben von Gruppen in der Tabelle zugelassen ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vertikales Verschieben von Gruppen in Tabelle zugelassen/nicht zugelassen
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AutoCollapseGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Gruppierungsebenen-Layout die Gruppen bei Interaktionen automatisch kollabiert werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppen werden/werden nicht bei Interaktionen automatisch kollabiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AutoExpandTargetGroup

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Gruppierungsebenen-Layout Zielgruppen bei Interaktionen automatisch expandiert werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zielgruppen werden/werden nicht bei Interaktionen automatisch expandiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

BodiesCollapsed

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob alle Gruppen der Gruppierungsebene kollabiert (True) oder expandiert (False) sind.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppen kollabiert/expandiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

BodiesCollapsedDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes für die kollabierten Rümpfen dieser Gruppierungsebene setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Gruppierungsebene mit den kollabierten Rümpfen enthält.

BodiesCollapsedMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle für den kollabierten Gruppenkörper auf dieser Gruppenebene setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **BodiesCollapsedDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Name Zuordnungstabelle

CalendarGridName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen des Kalendergitters für dieses Gruppierenebenenlayout erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Parameter: Rückgabewert	String	Name des Kalendergitters
	Mögliche Werte:	Name der Farbzusordnungstabelle
Eigenschaftswert	String	Name des Kalendergitters
	Mögliche Werte:	Name der Farbzusordnungstabelle

CalendarGridsWithChildGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen, ob auch für Untergruppen ein Kalendergitter angezeigt wird. Die Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Kalendergitter für Untergruppen wird/wird nicht angezeigt Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

CalendarNameDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das den Namen des für den Kalender der Gruppenebene zu verwendenden Kalenders enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Namen des zu verwendenden Kalenders enthält

DateLineGridName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen des Terminliniengitters für dieses Gruppierungsebenenlayout erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Name des Terminliniengitters Name der Farbzuoordnungstabelle

DateLineGridsWithChildGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen, ob auch für Untergruppen ein Terminliniengitter angezeigt wird. Die Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Terminliniengitter für Untergruppen wird/wird nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

DateLineName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen der Stichtalinie für dieses Gruppierungsebenenlayout erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Stichtaglinie
	Mögliche Werte:	Name der Farbzuoordnungstabelle

DateLinesWithChildGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Stichtaglinien gruppenübergreifend angezeigt werden sollen. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Stichtaglinie für Untergruppen wird/wird nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

GroupDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index, der für die Gruppierung dieses VcGrouLevelLayout-Objekt verwendet wird, festlegen oder erfragen

	Datentyp	Beschreibung
Eigenschaftswert	Long	Gruppierungsindex dieses VcGroupLevelLayout-Objektes

GroupsInvisible

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Gruppen dieser Gruppierungsebene angezeigt werden. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung

GroupsInvisibleCollapsedMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle für die nicht sichtbaren Gruppen auf dieser Gruppenebene setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **BodiesCollapsedDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

GroupsInvisibleDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes für die nicht sichtbaren Gruppen dieser Gruppierungsebene setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung

Level

Nur-Lese-Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Gruppierenebene dieses Gruppierenebenenlayouts erfragen. Es sind maximal 25 Gruppierungsebenen möglich.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Gruppierenebene des Gruppierenebenenlayouts Datenfeldindex

ModificationsAllowed

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft stellen Sie ein, ob der Anwender expandierte Gruppen dieser Ebene kollabieren oder kollabierte Gruppen wieder expandieren kann. Der Anwender kann dann die Gruppen durch einen Doppelklick auf die Gruppenüberschrift im Tabellenteil oder durch einen Klick auf das Minus- bzw. Plus-Zeichen neben der Gruppenüberschrift oder über das Kontextmenü für Gruppen kollabieren bzw. expandieren.

Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Änderungen erlaubt (True)/ nicht erlaubt (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGroupLevelLayout.ModificationsAllowed(0) = False
```

Name

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen des Gruppierungsebenenlayouts erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Gruppierungsebene
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim groupName As String

Set groupCltn = VcGantt1.GroupCollection
Set group = groupCltn.FirstGroup

groupName = group.Name
```

NodesInHeaders

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Knotenobjekte der Gruppe auf dieser Gruppierungsebene alle in derselben Zeile liegen (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Alle Knoten befinden sich/befinden sich nicht in derselben Zeile
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

NodesInHeadersDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes für die Knoten in Kopfzeilen dieser Gruppierungsebene setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

NodesInHeadersMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle für Knoten in Kopfzeilen auf dieser Gruppenebene setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **NodesInHeadersDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Name der Zuordnungstabelle

NodesOverlaid

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob auf dieser Gruppierungsebene das Knotenlayout optimiert ist (False) oder ob Knoten überlappen (True).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Das Knotenlayout ist/ist nicht optimiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
group.LevelLayout.NodesOverlaid = True
```

OptimizedNodesSortDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, das das Sortierkriterium (die Zeichnungspriorität) für die Darstellung mehrere Knoten in einer Zeile enthält. Das Setzen dieser Eigenschaft ist nur sinnvoll, wenn die Eigenschaft **NodesArranged-**

Optimized auf **True** gesetzt ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das Sortierkriterium enthält

OptimizedNodesSortOrder

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Sortierrichtung für das Sortierkriterium festlegen oder erfragen, das durch die Eigenschaft **OptimizedNodesSortDataFieldIndex** ausgewählt wurde. Das Setzen dieser Eigenschaft ist nur sinnvoll, wenn die Eigenschaft **NodesArranged-Optimized** auf **True** gesetzt ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	SortOrderEnum Mögliche Werte: vcAscending 1 vcDescending 2	Sortierrichtung Standardwert: vcAscending aufsteigende Abfolge absteigende Abfolge

OverlaidNodesSortDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, das das Sortierkriterium (die Zeichnungspriorität) für die Darstellung mehrere Knoten in einer Zeile enthält. Das Setzen dieser Eigenschaft ist nur sinnvoll, wenn die Eigenschaft **NodesArranged-Optimized** auf **False** gesetzt ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das Sortierkriterium enthält

OverlaidNodesSortOrder

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Sortierrichtung für das Sortierkriterium festlegen oder erfragen, das durch die Eigenschaft **OverlaidNodesSortDataFieldIndex** ausgewählt wurde. Das Setzen dieser Eigenschaft ist nur sinnvoll, wenn die Eigenschaft **NodesArrangedOptimized** auf **False** gesetzt ist. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	SortOrderEnum Mögliche Werte: vcAscending 1 vcDescending 2	Sortierrichtung Standardwert: vcAscending aufsteigende Abfolge absteigende Abfolge

PagebreakMode

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob und wann Seitenumbrüche nach Gruppen durchgeführt werden. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	PagebreakModeEnum Mögliche Werte: vcPagebreakAfterEachGroup 1 vcPagebreakNone 0 vcPagebreakOnPageFull 2	Art des Seitenumbruchs Standardwert: vcPagebreakNone Seitenumbruch nach jeder Gruppe Kein Seitenumbruch Seitenumbruch, wenn Folgegruppe nicht mehr komplett auf die Seite passt

RestoreAutoCollapsedGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Gruppierungsebenen-Layout automatisch kollabierte Gruppen bei Interaktionen automatisch wieder hergestellt werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Automatisch kollabierte Gruppen werden/werden nicht bei Interaktionen automatisch wieder hergestellt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

RestoreAutoExpandedGroups

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Gruppierungsebenen-Layout automatisch expandierte Gruppen bei Interaktionen automatisch wieder hergestellt werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Automatisch expandierte Gruppen werden/werden nicht bei Interaktionen automatisch wieder hergestellt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

RowBackColorAsARGB

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Hintergrundfarbe der Gruppentitelzeile für diese Gruppierungsebene setzen oder erfragen. Die Standard-Farbe ist weiß.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255})

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout

Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.FirstGroupLevelLayout

groupLevelLayout.RowBackColor = RGB(128, 128, 128)
```

RowBackColorDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowBackColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowBackColorMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuhordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **RowBackColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **RowBackColor** ausgegeben.

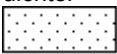
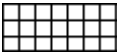
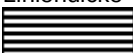
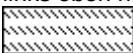
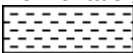
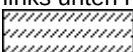
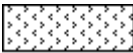
	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

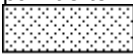
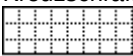
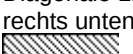
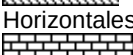
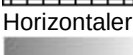
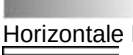
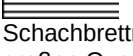
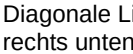
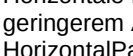
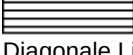
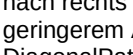
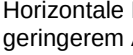
RowPattern

Nur-Lese-Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie für den Hintergrund des Gruppentitelzeile dieser Gruppierungsebene ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte:	

vc05PercentPattern...	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter
vc90PercentPattern 01 - 11	
vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe
	
vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben
	
vcCrossPattern 6	Kreuzschraffur
	
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke
	
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke
	
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke
	
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
	
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
	
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
	
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
	
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
	
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
	
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
	
vcDivotPattern 2036	Grassoden-Muster
	

vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern 
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschildel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 

vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

RowPatternColorAsARGB

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Musterfarbe der Gruppentitelzeile dieser Gruppierungsebene festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getRowBackColorAsARGB**.

Wenn in der Eigenschaft **RowPatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

Code-Beispiel

```
Dim groupLevelLayout As VcGroupLevelLayout

Set groupLevelLayout =
VcGantt1.GroupLevelLayoutCollection.GroupLevelLayoutByIndex(0)
groupLevelLayout.RowPatternColorAsARGB = &h88FF0A06
```

RowPatternColorDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowPatternColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowPatternColorMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuoordnungstabelle und ein Datenfeldindex in der Eigenschaft **RowPatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Gruppentitelzeile aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **RowPatternColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuoordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

RowPatternDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowPatternMapName** benötigt wird. Wenn Sie hier -1 angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowPatternMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Musterzuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn der Name einer Farbzuoordnungstabelle und ein Datenfeldindex in der Eigenschaft **RowPatternDataFieldIndex** angegeben sind, wird das Muster des Layers

aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **RowPattern** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

SeparationLineColor

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Farbe der Trennlinien für die Gruppierungsebenen festlegen.

Sie können diese Eigenschaft auch im Dialog **Gruppierung**, Bereich **Gruppierung**, Feld **Trennlinie** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	Farbwert {0...255},{0...255},{0...255})

SeparationLineColorDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuhordnungstabelle in der Eigenschaft **SeparationLineColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

SeparationLineColorMapName

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle für die Trennlinienfarbe setzen oder erfragen. Wird hier "" oder bei der

Eigenschaft **SeparationLineColorDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhnungstabelle
	Mögliche Werte:	Name der Farbzuhnungstabelle

SeparationLineThickness

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Liniendicke einer Trennlinie zwischen Gruppierungsebenen erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Liniendicke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Liniendicke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Liniendicke
		LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm

SeparationLineType

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie die Linienart einer Trennlinie zwischen Gruppierungsebenen erfragen oder festlegen.

Sie können diese Eigenschaft auch im Dialog **Gruppierung**, Bereich **Gruppierung**, Feld **Trennlinie** festlegen.

Eigenschaftswert	Datentyp	Beschreibung
	LineTypeEnum	Typ der Trennlinien der Hierarchieebenen
	Mögliche Werte:	
	vcDashed 4	Linientyp gestrichelt
	vcDashedDotted 5	Linientyp gestrichelt-gepunktet
	vcDotted 3	Linientyp gepunktet
	vcLineType0 100	Linientyp 0 _____
	vcLineType1 101	Linientyp 1 _ _ _ _ _
	vcLineType10 110	Linientyp 10 _
	vcLineType11 111	Linientyp 11 _
	vcLineType12 112	Linientyp 12 _
	vcLineType13 113	Linientyp 13 _
	vcLineType14 114	Linientyp 14 _
	vcLineType15 115	Linientyp 15 _
	vcLineType16 116	Linientyp 16 _
	vcLineType17 117	Linientyp 17 _
	vcLineType18 118	Linientyp 18 _
	vcLineType2 102	Linientyp 2 _
	vcLineType3 103	Linientyp 3 _
	vcLineType4 104	Linientyp 4 _
	vcLineType5 105	Linientyp 5 _
	vcLineType6 106	Linientyp 6 _
	vcLineType7 107	Linientyp 7 _
	vcLineType8 108	Linientyp 8 _
	vcLineType9 109	Linientyp 9 _
	vcNone 1	Kein Linientyp
	vcNotSet -1	Kein Linientyp zugewiesen
	vcSolid 2	Linientyp durchgezogen

ShowCalendarGrids

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob arbeitsfreie Zeiten farblich und/oder durch ein Muster hervorgehoben werden sollen. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Arbeitsfreie Zeiten werden/werden nicht hervorgehoben
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
End Sub
```

ShowDateLineGrids

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob vertikale Rasterlinien angezeigt werden. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rasterlinien werden/werden nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
End Sub
```

ShowDateLines

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Stichtaglinien angezeigt werden sollen. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Stichtaglinien werden/werden nicht angezeigt Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowGroupNodes

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Knoten dieser Gruppierungsebene angezeigt werden. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Gruppenknoten sind/sind nicht sichtbar Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowSeparationLines

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob bei eingestellter Gruppierung Trennlinien zwischen den Gruppierungsebenen dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung

ShowSeparationLinesAtTop

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob bei eingestellter Gruppierung Trennlinien oberhalb der Gruppen dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Gruppenweise** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Trennlinien oben werden angezeigt/nicht angezeigt Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

SortDataFieldIndex

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, nach dem Gruppen auf dieser Gruppierungsebene sortiert werden sollen. Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ sortlevel	Integer Mögliche Werte:	Sortierebene Datenfeldindex
Eigenschaftswert	Long	Index des Datenfeldes, das das Sortierkriterium enthält

SortOrder

Eigenschaft von VcGroupLevelLayout

Mit dieser Eigenschaft legen Sie fest, ob die Gruppen auf- oder absteigend sortiert werden. Das Feld, nach dem die Gruppen sortiert werden, legen Sie mit der Eigenschaft **SortDataFieldIndex** fest. Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ sortLevel	Integer Mögliche Werte:	Sortierebene Datenfeldindex
Eigenschaftswert	SortOrderEnum Mögliche Werte: vcAscending 1 vcDescending 2	Sortierrichtung Standardwert: vcAscending aufsteigende Abfolge absteigende Abfolge

Code-Beispiel

```
VcGantt1.VcGroupLevelLayout.SortOrderField (0) = 12
VcGantt1.VcGroupLevelLayout (0) = vcAscending
VcGantt1.VcGroupLevelLayout
```

Specification**Nur-Lese-Eigenschaft von VcGroupLevelLayout**

Mit dieser Eigenschaft können Sie die Spezifikation des Layouts der Gruppierungsebene auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Layouts der Gruppierungsebene mit der Methode **VcGroupLevelLayout-Collection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Spezifikation des Layouts der Gruppierungsebene Name der Farbzuoordnungstabelle

SummaryBarsVisible**Eigenschaft von VcGroupLevelLayout**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Summenbalken angezeigt werden oder nicht.

Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Summenbalken sichtbar (True)/ unsichtbar (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGroupLevelLayout.SummaryBarsVisible (-1) = False
```

Visible**Eigenschaft von VcGroupLevelLayout**

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die aktuelle Gruppierenebene sichtbar (True) oder unsichtbar (False) sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppierenebene sichtbar/unsichtbar
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

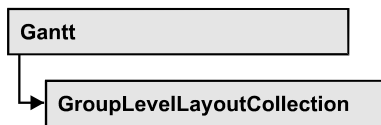
Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout

Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.GroupLevelLayoutByName("A")

groupLevelLayout.Visible = False
```

7.41 VcGroupLevelLayoutCollection



In einem Objekt des Typs `VcGroupLevelLayoutCollection` sind alle verfügbaren Layout-Objekte der Gruppierungsebenen zusammengefasst, sofern Knoten gruppiert worden sind. Über **For Each groupLevelLayout In GroupLevelLayoutCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Layout-Objekte zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **GroupLevelLayoutByName** und **GroupLevelLayoutByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Layouts kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Layouts.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstGroupLevelLayout`
- `GroupLevelLayoutByIndex`
- `GroupLevelLayoutByName`
- `NextGroupLevelLayout`
- `Remove`
- `Update`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcGroupLevelLayoutCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle ent-

haltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim groupLevelLayout As VcGroupLevelLayout

For Each groupLevelLayout In VcGantt1.GroupLevelLayout
    Debug.Print groupLevelLayout.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcGroupLevelLayoutCollection

Mit dieser Eigenschaft kann die Anzahl der Gruppierungsebenenlayouts in der Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der GroupLevelLayouts

Code-Beispiel

```
Dim groupLevelLayoutCltn As Vc GroupLevelLayoutCollection
Dim numberOfGroupLevelLayouts As Long
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
numberOfGroupLevelLayouts = groupLevelLayoutCltn.Count
```

Methoden

Add

Methode von VcGroupLevelLayoutCollection

Mit dieser Methode können Sie ein neues Gruppierungsebenenlayout in der GroupLevelLayoutCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Gruppierungsebenenlayoutobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ groupLevelLayoutName	String Mögliche Werte:	Name des GroupLevelLayouts Name der Farbzordnungstabelle
Rückgabewert	VcGroupLevelLayout	Neues GroupLevelLayout-Objekt

Code-Beispiel

```
Set newGroupLevelLayout =
VcGantt1.GroupLevelLayoutCollection.Add("GroupingLevel1")
```

AddBySpecification**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie das Layout einer Gruppierungsebene über eine Spezifikation erzeugen. Dies dient der Persistenz von Layout-Objekten der Gruppierungsebenen. Die Spezifikation eines layouts kann erfragt und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Layout mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	GroupLevelLayout-Spezifikation Name der Farbzordnungstabelle
Rückgabewert	VcGroupLevelLayout	Neues GroupLevelLayout-Objekt

Copy**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie ein Gruppierungsebenenlayout kopieren. Wenn das Gruppierungsebenenlayout mit dem angegebenen Namen existiert und der Name des neuen Gruppierungsebenenlayouts noch nicht verwendet wird, wird das neue Gruppierungsebenenlayoutobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ groupLevelLayoutName	String	Name des zu kopierenden GroupLevelLayouts
	Mögliche Werte:	Name der Farbzuoordnungstabelle
⇒ newGroupLevelLayoutName	String	Name des neuen GroupLevelLayouts
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	VcGroupLevelLayout	GroupLevelLayoutobjekt

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.Copy("CurrentGroupLevelLayout",
"NewGroupLevelLayout")
```

FirstGroupLevelLayout**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie auf das erste Gruppierungsebenenlayout der Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextGroupLevelLayout** über die nachfolgenden Gruppierungsebenenlayouts zu iterieren. Existiert kein Gruppierungsebenenlayout in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcGroupLevelLayout	Erstes GroupLevelLayout

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
groupLevelLayoutCltn.SelectgroupLevelLayouts (vcAnyGroupLevelLayout)
Set groupLevelLayout = groupLevelLayoutCltn.FirstGroupLevelLayout
```

GroupLevelLayoutByIndex**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie auf ein bestimmtes Gruppierungsebenenlayout über seinen Index zugreifen. Existiert kein Gruppierungsebenenlayout unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des GroupLevelLayouts Datenfeldindex
Rückgabewert	VcGroupLevelLayout	Ermitteltes GroupLevelLayoutobjekt

Code-Beispiel

```

Dim groupLevelLayoutCltn As VcGroupLevelLayout
Dim dateLine As VcDateLine
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayout
Set groupLevelLayout = groupLevelLayoutCltn.GroupLevelLayoutByIndex(2)
MsgBox groupLevelLayout.Name

```

GroupLevelLayoutByName**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie unter Verwendung des Namens des Gruppierungsebenenlayouts auf ein bestimmtes Gruppierungsebenenlayout zugreifen. Existiert kein GroupLevelLayout-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ groupLevelLayoutName	String Mögliche Werte:	Name des GroupLevelLayouts Name der Farbzusordnungstabelle
Rückgabewert	VcGroupLevelLayout	GroupLevelLayout

Code-Beispiel

```

Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.GroupLevelLayoutByName("Grouping level A")

```

NextGroupLevelLayout**Methode von VcGroupLevelLayoutCollection**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Gruppierungsebenenlayouts der Auflistung zugreifen, nachdem Sie mit der Methode **FirstGroupLevelLayout** den Initialwert erfasst haben. Sind alle

Levels durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcGroupLevelLayout	Nachfolgendes GroupLevelLayout

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.FirstGroupLevelLayout
While Not groupLevelLayout Is Nothing
Listbox.AddItem groupLevelLayout.Name
Set groupLevelLayout = groupLevelLayoutCltn.NextGroupLevelLayout
Wend
```

Remove

Methode von VcGroupLevelLayoutCollection

Mit dieser Methode können Sie ein Gruppierungsebenenlayout löschen. Wenn das Gruppierungsebenenlayout noch in irgendeinem anderen Objekt benutzt wird, kann sie nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ groupLevelLayoutName	String	Name des GroupLevelLayouts
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	GroupLevelLayout gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim groupLevelLayoutCltn As VcGroupLevelLayoutCollection
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayoutCltn = VcGantt1.GroupLevelLayoutCollection
Set groupLevelLayout = groupLevelLayoutCltn.FormatByIndex(1)
groupLevelLayoutCltn.Remove (groupLevelLayout.Name)
```

Update

Methode von VcGroupLevelLayoutCollection

Mit dieser Methode können Sie die Darstellung aller Objekte, die durch die verwendeten Gruppierungsebenenlayouts bestimmt werden, aktualisieren. Wenn Sie diese Methode nicht aufrufen, werden die Änderungen der

1012 API-Referenz: VcGroupLevelLayoutCollection

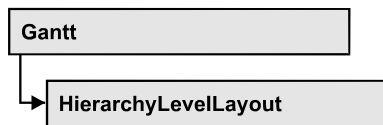
Gruppierungsebenenlayouts zur Laufzeit nicht ausgeführt. Sie sollten diese Methode erst am Ende des Codes zur Festlegung der Gruppierungsebenenlayouts und der Level-Auflistung verwenden, damit die Aktualisierung nicht schon ausgeführt wird, bevor alle Festlegungen der Gruppierungsebenenlayouts ausgeführt worden sind.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim groupLevelLayout As VcGroupLevelLayout
Set groupLevelLayout =
VcGantt1.GroupLevelLayout.Collection.GroupLevelLayoutByName("Grouping Level 3")
groupLevelLayout.Update
```

7.42 VcHierarchyLevelLayout



Ein Objekt vom Typ **VcHierarchyLevelLayout** legt Inhalt und Erscheinungsbild einer hierarchischen Anordnung von Knoten fest.

Eigenschaften

- AutoCollapseGroups
- AutoExpandTargetGroup
- BodiesCollapsed
- BodiesCollapsedDataFieldIndex
- BodiesCollapsedMapName
- HierarchyDataFieldIndex
- LevelMaximumForPagebreaks
- NodeSeparationLinesVisible
- NodesInHeaders
- NodesInHeadersDataFieldIndex
- NodesInHeadersMapName
- NodesOverlaid
- PagebreakMode
- RestoreAutoCollapsedGroups
- RestoreAutoExpandedGroups
- SeparationLineColor
- SeparationLineThickness
- SeparationLineType
- ShowSeparationLines
- SummaryBarsVisible

Eigenschaften

AutoCollapseGroups

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Hierarchieebenen-Layout die Gruppen bei Interaktionen automatisch kollabiert werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppen werden/werden nicht bei Interaktionen automatisch kollabiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

AutoExpandTargetGroup

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Hierarchieebenen-Layout Zielgruppen bei Interaktionen automatisch expandiert werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zielgruppen werden/werden nicht bei Interaktionen automatisch expandiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

BodiesCollapsed

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob alle Gruppen kollabiert sind (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	alle Gruppen sind/sind nicht kollabiert Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

BodiesCollapsedDataFieldIndex

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes für die kollabierten Rümpfen dieser Hierarchieebene setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Gruppierungsebene mit den kollabierten Rümpfen enthält.

BodiesCollapsedMapName

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle für den kollabierten Gruppenkörper auf dieser Hierarchieebene setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **BodiesCollapsedDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Name Zuordnungstabelle

HierarchyDataFieldIndex

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, nach dem Vorgänge in diesem **VcGroupLevelLayout-Objekt** hierarchisch angeordnet werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, nach dem Vorgänge hierarchisch angeordnet werden sollen

LevelMaximumForPagebreaks

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, bis zu welcher Hierarchieebene Seitenumbrüche ausgeführt werden.

Ist der Standardwert -1 gesetzt, werden für alle Ebenen Seitenumbrüche durchgeführt.

	Datentyp	Beschreibung

NodeSeparationLinesVisible

Nur-Lese-Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Trennlinien zwischen Knoten angezeigt werden oder nicht.

Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Knoten** festgelegt werden.

	Datentyp	Beschreibung

Code-Beispiel

```
VcHierarchyLevelLayout.NodeSeparationLinesVisible (-1) = False
```

NodesInHeaders

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Knotenobjekte der Gruppe auf dieser Hierarchieebene alle in derselben Zeile liegen (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Alle Knoten befinden sich/befinden sich nicht in derselben Zeile
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim hierarchyLevelLayoutCltn As VcHierarchyLevelLayoutCollection
Dim hierarchyLevelLayout As VcHierarchyLevelLayout

Se thierarchyLevelLayoutCltn = VcGantt1.HierarchyLevelLayoutCollection
Set hierarchyLevelLayout =
hierarchyLevelLayoutCltn.HierarchyLevelLayoutByName("3")

hierarchyLevelLayout.AllNodesInOneRow = True
```

NodesInHeadersDataFieldIndex**Eigenschaft von VcHierarchyLevelLayout**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes für die Knoten in Kopfzeilen dieser Hierarchieebene setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	inout

NodesInHeadersMapName**Eigenschaft von VcHierarchyLevelLayout**

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle für Knoten in Kopfzeilen auf dieser Gruppenebene setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **NodesInHeadersDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	in

NodesOverlaid

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob auf dieser Hierarchieebene das Knotenlayout optimiert ist (False) oder ob Knoten überlappen (True).

	Datentyp	Beschreibung
Parameter: ↔ Rückgabewert	Boolean Mögliche Werte:	Das Knotenlayout ist/ist nicht optimal Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
Eigenschaftswert	Long	out

Code-Beispiel

```
group.LevelLayout.NodesArrangedOptimized = True
```

PagebreakMode

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob und wann Seitenumbrüche nach Gruppen durchgeführt werden. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	PagebreakModeEnum Mögliche Werte: vcPagebreakAfterEachGroup 1 vcPagebreakNone 0 vcPagebreakOnPageFull 2	Art des Seitenumbruchs Standardwert: vcPagebreakNone Seitenumbruch nach jeder Gruppe Kein Seitenumbruch Seitenumbruch, wenn Folgegruppe nicht mehr komplett auf die Seite passt

RestoreAutoCollapsedGroups

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Hierarchieebenen-Layout automatisch kollabierte Gruppen bei Interaktionen automatisch wieder hergestellt werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Automatisch kollabierte Gruppen werden/werden nicht bei Interaktionen automatisch wieder hergestellt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

RestoreAutoExpandedGroups

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob im Hierarchieebenen-Layout automatisch expandierte Gruppen bei Interaktionen automatisch wieder hergestellt werden sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Automatisch expandierte Gruppen werden/werden nicht bei Interaktionen automatisch wieder hergestellt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

SeparationLineColor

Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie die Farbe der Trennlinien für die Hierarchieebenen festlegen.

Sie können diese Eigenschaft auch im **Hierarchie**-Bereich des Dialogs **Gruppierung** festlegen, indem Sie bei **Trennlinie** auf  klicken.

	Datentyp	Beschreibung
Eigenschaftswert	Color	Farbwert ({0...255},{0...255},{0...255})

Code-Beispiel

```
VcHierarchyLevelLayout.SeparationLineColor = RGB(255, 204, 204)
```


SeparationLineThickness

Nur-Lese-Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie die Linienstärke einer Trennlinie zwischen Hierarchieebenen erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Sie können diese Eigenschaft auch im **Hierarchie**-Bereich des Dialogs **Gruppierung** festlegen, indem Sie bei **Trennlinie** auf  klicken.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm

SeparationLineType

Nur-Lese-Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie die Linienart einer Trennlinie zwischen Hierarchieebenen erfragen oder festlegen.

Sie können diese Eigenschaft auch im **Hierarchie**-Bereich des Dialogs **Gruppierung** festlegen, indem Sie bei **Trennlinie** auf  klicken.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Typ der Trennlinien der Hierarchieebenen
	Mögliche Werte:	
	vcDashed 4	Linientyp gestrichelt
	vcDashedDotted 5	Linientyp gestrichelt-gepunktet
	vcDotted 3	Linientyp gepunktet
	vcLineType0 100	Linientyp 0 
	vcLineType1 101	Linientyp 1 
	vcLineType10 110	Linientyp 10 
	vcLineType11 111	Linientyp 11 
	vcLineType12 112	Linientyp 12 
	vcLineType13 113	Linientyp 13 
	vcLineType14 114	Linientyp 14 
	vcLineType15 115	Linientyp 15 
	vcLineType16 116	Linientyp 16 
	vcLineType17 117	Linientyp 17 
	vcLineType18 118	Linientyp 18 
	vcLineType2 102	Linientyp 2 
	vcLineType3 103	Linientyp 3 
	vcLineType4 104	Linientyp 4 
	vcLineType5 105	Linientyp 5 
	vcLineType6 106	Linientyp 6 
	vcLineType7 107	Linientyp 7 
	vcLineType8 108	Linientyp 8 
	vcLineType9 109	Linientyp 9 
	vcNone 1	Kein Linientyp
	vcNotSet -1	Kein Linientyp zugewiesen
	vcSolid 2	Linientyp durchgezogen

ShowSeparationLines

Nur-Lese-Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob bei eingestellter Hierarchie Trennlinien zwischen den Hierarchieebenen dargestellt werden sollen oder nicht.

Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Hierarchie** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Trennlinien werden angezeigt/nicht angezeigt Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcHierarchyLevelLayout.ShowSeparationLines = True
```

SummaryBarsVisible

Nur-Lese-Eigenschaft von VcHierarchyLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Summenbalken angezeigt werden oder nicht.

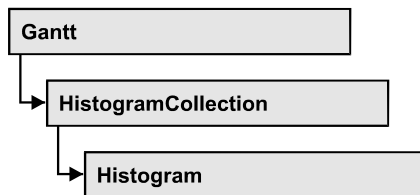
Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Hierarchie** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Summenbalken sichtbar (True)/ unsichtbar (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcHierarchyLevelLayout.SummaryBarsVisible (-1) = False
```

7.43 VcHistogram



Ein Objekt vom Typ **VcHistogram** ist ein Element des Objektes **VcHistogramCollection** und bezeichnet ein Histogramm, das sinnvollerweise mit Auslastungs- und Kapazitätskurven zu dem darüber liegenden Gantt-Diagramm bestückt wird. Sie können hier eine Skala anlegen und Kurven definieren, die aus verschiedenen Quellen generiert werden können.

Eigenschaften

- CalendarName
- CurveCollection
- Name
- NominalScaleMaximum
- NominalScaleMinimum
- NumericScaleCollection
- RowBackColorAsARGB
- RowPattern
- RowPatternColorAsARGB
- ShowCalendarGrids
- Visible

Methoden

- FitRangeIntoView
- GetActualScaleValues
- GetActualScaleValuesAsVariant
- GetCurrentYValues
- GetCurrentYValuesAsVariant
- PutInOrderAfter
- ScrollToValue

Eigenschaften

CalendarName

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie dem Histogramm einen Kalender zuweisen. Der Kalender beinhaltet das abzubildende Zeitmuster für die Gitterlinien und wird mit Hilfe seines Namens spezifiziert.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Kalendernamen übergibt
	Mögliche Werte:	Name der Farbuordnungstabelle

CurveCollection

Nur-Lese-Eigenschaft von VcHistogram

Mit dieser Eigenschaft haben Sie Zugriff auf die KurvenAuflistung, in der alle zur Verfügung stehenden Kurven enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcCurveCollection	CurveCollection-Objekt

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram
Dim curveCltn As VcCurveCollection

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("Histogram_1")
Set curveCltn = histogram.CurveCollection
```

Name

Nur-Lese-Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie den Namen eines Histogramms erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Histogramms
	Mögliche Werte:	

	Name der Farbzusordnungstabelle
--	---------------------------------

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.Active
MsgBox histogram.Name
```

NominalScaleMaximum**Eigenschaft von VcHistogram**

Mit dieser Eigenschaft können Sie den Maximalwert der numerischen Skala des Histogramms beim Starten des Programms setzen. Falls die y-Werte der Histogrammkurven diesen Maximalwert überschreiten, wird die numerische Skala an die y-Werte der Histogrammkurven angepaßt.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Maximaler y-Wert

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("Histogram_1")

histogram.NominalScaleMaximum (20)
```

NominalScaleMinimum**Eigenschaft von VcHistogram**

Mit dieser Eigenschaft können Sie den Minimalwert der numerischen Skala des Histogramms setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Minimaler y-Wert

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("Histogram_1")

histogram.NominalScaleMinimum (2)
```

NumericScaleCollection

Nur-Lese-Eigenschaft von VcHistogram

Diese Eigenschaft ermöglicht den Zugriff auf die NumericScale-Auflistung, in der alle zur Verfügung stehenden numerischen Skalen enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcNumericScaleCollection	NumericScaleCollection-Objekt

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram
Dim numericScaleCltn As VcNumericScaleCollection

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("Histogram_1")
Set numericScaleCltn = histogram.NumericScaleCollection
```

RowBackColorAsARGB

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie die Hintergrundfarbe für das Histogramm setzen oder erfragen. Diese Option können Sie auch im Dialog "Histogramme verwalten festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	System.Drawing.Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}}

Code-Beispiel

```
VcHistogram.RowBackColor = RGB(255, 0, 0)
```

RowPattern

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie für den Hintergrund des Histogramms ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcFillPattern	Mustertyp

RowPatternColorAsARGB

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie die Musterfarbe des Histogramms festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Tipp:> Der Hintergrund im Bereich der numerischen Skala wird erst dann sichtbar, wenn deren Streifenhintergrund transparent ist, ansonsten bleibt immer der Streifenhintergrund sichtbar.

	Datentyp	Beschreibung
Eigenschaftswert	System.Drawing.Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}

ShowCalendarGrids

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob arbeitsfreie Zeiten farblich und/oder durch ein Muster hervorgehoben werden sollen. Sie können diese Eigenschaft auch im Dialog **Histogramme verwalten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Arbeitsfreie Zeiten werden/werden nicht hervorgehoben
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Visible

Eigenschaft von VcHistogram

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Histogramm sichtbar ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Histogramm sichtbar (True)/ nicht sichtbar (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.Active
histogram.Visible = True
```

Methoden

FitRangeIntoView

Methode von VcHistogram

Mit dieser Methode wird ein bestimmter Teilbereich der numerischen Skala in das Histogramm eingepasst und damit sichtbar gemacht. Die Skalierung der Skala wird dabei entsprechend verändert. Mit den Parametern **startValue** und **endValue** werden Anfang und Ende des Bereiches festgelegt. Der Parameter **gapAsNoOfTimeUnits** wird nicht verwendet. Um passende Bereichsgrenzen aus den vorhandenen Kurven abzuleiten s. **GetCurrentYValues(...)**

Für das Einpassen von Histogrammen in ein Fenster s. **VcGantt.Fit-HistogramsIntoView**.

	Datentyp	Beschreibung
Parameter:		
⇒ startValue	Long	Anfangswert des einzupassenden Bereichs.
⇒ endValue	Long	Endwert des einzupassenden Bereichs.
⇒ gapAsNoOfTimeUnits	Long	Parameter wird nicht verwendet
Rückgabewert	Boolean	Bereich wurde erfolgreich (True) /nicht erfolgreich (False) eingepasst.

GetActualScaleValues

Methode von VcHistogram

Mit dieser Methode können Sie die tatsächlichen minimalen und maximalen Werte der numerischen Skala im Histogramm ermitteln.

	Datentyp	Beschreibung
Parameter:		
↔ minimumValue	Long	Kleinsten Y-Wert der numerischen Skala
↔ maximumValue	Long	Größten Y-Wert der numerischen Skala
Rückgabewert	Boolean	Extremwerte wurden erfolgreich (True) / nicht erfolgreich (False) ermittelt.

GetActualScaleValuesAsVariant

Methode von VcHistogram

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetActualScaleValues**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ↔) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

GetCurrentYValues

Methode von VcHistogram

Mit dieser Methode können Sie den minimalen und maximalen Y-Wert aller Kurven des Histogramms ermitteln. Diese Abfrage kann dazu dienen, z. B. den sichtbaren Bereich der numerischen Skala entsprechend anzupassen (s. **FitRangeIntoView**)

	Datentyp	Beschreibung
Parameter:		
↔ minValue	Long	Kleinsten Y-Wert aller Kurven.
↔ maxValue	Long	Größten Y-Wert aller Kurven.
Rückgabewert	Boolean	Extremwerte wurden erfolgreich (True) / nicht erfolgreich (False) ermittelt.

GetCurrentYValuesAsVariant

Methode von VcHistogram

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetCurrentYValues**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch $\Leftrightarrow$) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

PutInOrderAfter

Methode von VcHistogram

Mit dieser Methode können Sie das Histogramm in der Auflistung aller Histogramme hinter das durch den Namen angegebene setzen. Wenn als Name "" übergeben wird, wird das Histogramm an die erste Stelle gesetzt. Die Reihenfolge der Histogramme in der Auflistung entscheidet darüber, in welcher Reihenfolge sie dargestellt werden.

	Datentyp	Beschreibung
Parameter: $\Rightarrow$ refName	String	Name des Histogramms, hinter das das aktuelle Histogramm eingeordnet werden soll.
	Mögliche Werte:	Name der Farbzuoordnungstabelle
Rückgabewert	Void	

Code-Beispiel

```
Dim histgrCltn As VcHistogramCollection
Dim histgr1 As VcHistogram
Dim histgr2 As VcHistogram

histgrCltn = VcGantt1.HistogramCollection()
histgr1 = histgrCltn.Add("histgr1")
histgr2 = histgrCltn.Add("histgr2")
histgr1.PutInOrderAfter("histgr2")
histgrCltn.Update()
```

ScrollToValue

Methode von VcHistogram

Mit dieser Methode können Sie zu einem bestimmten y-Wert im Histogramm scrollen und festlegen, ob dieser Wert oben, unten oder mittig auf dem Bildschirm erscheinen soll.

	Datentyp	Beschreibung
Parameter: ⇒ value ⇒ verAlignment	Long VerticalAlignmentEnum Mögliche Werte: vcBottomAligned 2 vcTopAligned 1 vcVerCenterAligned -1	y-Wert, zu dem gescrollt werden soll vertikale Ausrichtung bündig unten bündig oben vertikal mittig
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

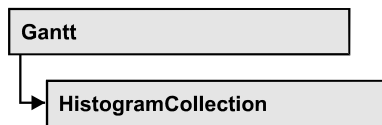
```

Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("HISTOGRAM_1")
histogram.ScrollToValue 7, vcCenterAligned

```

7.44 VcHistogramCollection



In einem Objekt des Typs VcHistogramCollection sind alle Histogramme automatisch zusammengefasst. Über **For Each histogram In HistogramCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Histogramme zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaft **HistogramByName**. Die Anzahl der im Auflistungsobjekt vorhandenen Histogramme kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- _NewEnum
- Active
- Count

Methoden

- CreateHistogram
- DeleteHistogram
- FirstHistogram
- HistogramByIndex
- HistogramByName
- NextHistogram

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcHistogramCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Histogrammobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim histogram As VcHistogram

For Each histogram In VcGantt1.HistogramCollection
    Debug.Print histogram.Name
Next
```

Active**Eigenschaft von VcHistogramCollection**

Mit dieser Eigenschaft können sie das aktuell verwendete Histogramm setzen oder erfragen.

Das verwendete Histogramm kann allerdings **Nothing** sein, wenn bis dahin keine Benutzeraktivität (z. B. Markierung einer Kurve) stattgefunden hat.

	Datentyp	Beschreibung
Eigenschaftswert	VcHistogram	Aktuell verwendetes Histogramm

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.Active
```

Count**Nur-Lese-Eigenschaft von VcHistogramCollection**

Mit dieser Eigenschaft kann die Anzahl der Histogramme in der Histogrammauflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Histogramme

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim numberOfHistograms As Long

Set histogramCltn = VcGantt1.HistogramCollection
numberOfHistograms = histogramCltn.Count
```

Methoden

CreateHistogram

Methode von VcHistogramCollection

Mit dieser Methode können Sie ein Histogrammobjekt anlegen. Dieses Histogramm ist eine Kopie des letzten zuvor angelegten und enthält somit auch dessen Kurven. Es gehört automatisch der Histogrammauflistung an.

	Datentyp	Beschreibung
Parameter: ⇒ histogramName	String Mögliche Werte:	Name des Histogramms Name der Farbzuoordnungstabelle
Rückgabewert	VcHistogram	Angelegtes Histogramm

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogram = histogramCltn.CreateHistogram ("Histogram2")
```

DeleteHistogram

Methode von VcHistogramCollection

Mit dieser Methode können Sie ein Histogramm aus der Histogrammauflistung entfernen.

	Datentyp	Beschreibung
Parameter: ⇒ histogramName	String Mögliche Werte:	Name des Histogramms Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	Histogramm erfolgreich (True) / nicht erfolgreich (False) gelöscht.

Code-Beispiel

```
Dim histogramCltn As VcHistogramCollection
Dim Deleted As Boolean

Set histogramCltn = VcGantt1.HistogramCollection
Deleted = histogramCltn.DeleteHistogram (String "name")
```

FirstHistogram

Methode von VcHistogramCollection

Mit dieser Methode können Sie auf das erste Histogramm der Histogrammauflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextHistogram** über die nachfolgenden Histogramme zu iterieren. Existiert kein Histogramm in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcHistogram	Erstes Histogramm

Code-Beispiel

```
Dim HistogramCltn As VcHistogramCollection
Dim Histogram As VcHistogram

Set HistogramCltn = VcGantt1.HistogramCollection
Set Histogram = HistogramCltn.FirstHistogram
```

HistogramByIndex

Methode von VcHistogramCollection

Mit dieser Methode können Sie auf ein einzelnes Histogramm über seinen Index zugreifen. Existiert kein Histogramm unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Histogramms Datenfeldindex
Rückgabewert	VcHistogram	Ermitteltes Histogrammobjekt

HistogramByName

Methode von VcHistogramCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Histogramm zugreifen. Existiert kein Histogramm unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ histogramName	String	Name des Histogramms
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcHistogram	Histogramm

Code-Beispiel

```

Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.HistogramByName("Histogram2")

```

NextHistogram**Methode von VcHistogramCollection**

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Histogramme der Histogramm-Auflistung zugreifen, nachdem Sie mit der Methode **FirstHistogram** den Initialwert erfasst haben. Sind alle Histogramme durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcHistogram	Nachfolgendes Histogramm

Code-Beispiel

```

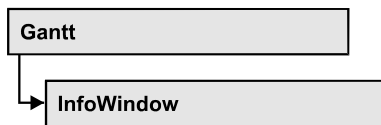
Dim histogramCltn As VcHistogramCollection
Dim histogram As VcHistogram

Set histogramCltn = VcGantt1.HistogramCollection
Set histogram = histogramCltn.FirstHistogram

While Not histogram Is Nothing
    List1.AddItem histogram.Name
    Set histogram = histogramCltn.NextHistogram
Wend

```

7.45 VcInfoWindow



Einem Objekt vom Typ VcInfoWindow bezeichnet das Informationsfenster eines Knotens, das in einem Balkenplan beim Anlegen oder Verändern des Knotens erscheint.

Eigenschaften

- OutputFormatForCenterDate
- OutputFormatForDuration
- OutputFormatForEndDate
- OutputFormatForStartDate
- ReferenceDate
- UseReferenceDate
- Visible

Eigenschaften

OutputFormatForCenterDate

Eigenschaft von VcInfoWindow

Mit dieser Eigenschaft können Sie das Ausgabeformat des in der Mitte eines Layers befindlichen Datums (z.B. Symbol-Layer) im Informationsfenster eines Knotens einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- | | |
|------|---|
| D: | Wochentagsname erster Buchstabe |
| TD: | Wochentagsname (kann über das Ereignis OnSupplyTextEntry angepasst werden) |
| DD: | Tagesnummer zweistellig: 01-31 (nicht anpassbar) |
| DDD: | die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar) |
| M: | erster Buchstabe des Monatsnamens (nicht anpassbar) |
| TM: | Monatsname (kann über das Ereignis OnSupplyTextEntry angepasst werden) |

MM:	zweistellige Monatsnummer: 01-12
MMM:	erste drei Buchstaben des Monatsnamens (nicht anpassbar)
YY:	zweistellige Jahreszahl
YYYY:	vierstellige Jahreszahl
WW:	zweistellige Nummer der Kalenderwoche: 01-53
TW:	Text für "Kalenderwoche" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
Q:	einstellige Quartalsnummer: 1-4
TQ:	Quartalsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
hh:	Stunde zweistellig im 24-Stunden-Format: 00-23
HH:	Stunde zweistellig im 12-Stunden-Format: 01-12
Th:	Text für "Uhr" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
mm:	Minute zweistellig: 00-59
ss:	Sekunde zweistellig: 00-59
TS:	kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TL:	langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TT:	Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Code für das zu verwendende Format enthält; bei einer leeren Zeichenkette wird das Output-Format des Gantt-Objektes verwendet (s. VcGantt.DateOutputFormat).
	Mögliche Werte:	

	Name der Farbzuoordnungstabelle
--	---------------------------------

OutputFormatForDuration

Eigenschaft von VcInfoWindow

Mit dieser Eigenschaft können Sie das Ausgabeformat der Dauer im Informationsfenster eines Knotens einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

hh: Stunde zweistellig im 24-Stunden-Format: 00-23

mm: Minute zweistellig: 00-59

ss: Sekunde zweistellig: 00-59

xC/XC: *Um dieses Format nutzen zu können, muss eine spezielle Einstellung in der Konfigurationsdatei vorgenommen werden. Bitte kontaktieren Sie bei Bedarf NETRONIC.* Sie können die Dauer als maximal zehngliedrige, einfache Aufwärtszählung gestalten, z.B. "07:16:00". Dieses Beispiel zeigt eine Dauer von 7 Stunden, 16 Minuten und 0 Sekunden. Die Notation dazu ist: **xC22:C11:C00**. Es sollen je 2 Stellen für die Glieder 2...0 vorgesehen werden. Die Trennzeichen sind variabel und könnten durch andere ersetzt werden. "x" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, aber kein "+"Symbol, falls er positiv ist. "X" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, und ebenso ein "+"Symbol, falls er positiv ist.

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Code für das zu verwendende Format enthält; bei einer leeren Zeichenkette wird das Output-Format des Gantt-Objektes verwendet (s. VcGantt.DateOutputFormat).
	Mögliche Werte:	Name der Farbzuoordnungstabelle

OutputFormatForEndDate

Eigenschaft von VcInfoWindow

Sie Mit dieser Eigenschaft können Sie das Ausgabeformat eines Layer-Enddatums im Informationsfenster eines Knotens einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM: Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- MM: zweistellige Monatsnummer: 01-12
- MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)
- YY: zweistellige Jahreszahl
- YYYY: vierstellige Jahreszahl
- WW: zweistellige Nummer der Kalenderwoche: 01-53
- TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- Q: einstellige Quartalsnummer: 1-4
- TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- hh: Stunde zweistellig im 24-Stunden-Format: 00-23
- HH: Stunde zweistellig im 12-Stunden-Format: 01-12

Th:	Text für "Uhr" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
mm:	Minute zweistellig: 00-59
ss:	Sekunde zweistellig: 00-59
TS:	kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TL:	langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TT:	Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Code für das zu verwendende Format enthält; bei einer leeren Zeichenkette wird das Output-Format des Gantt-Objektes verwendet (s. VcGantt.DateOutputFormat).
	Mögliche Werte:	Name der Farbzuoordnungstabelle

OutputFormatForStartDate

Eigenschaft von VcInfoWindow

Sie Mit dieser Eigenschaft können Sie das Ausgabeformat des Layer-Startdatums im Informationsfenster eines Knotens einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

D:	Wochentagsname erster Buchstabe
TD:	Wochentagsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
DD:	Tagesnummer zweistellig: 01-31 (nicht anpassbar)

- DDD. die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM. Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- MM: zweistellige Monatsnummer: 01-12
- MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)
- YY: zweistellige Jahreszahl
- YYYY. vierstellige Jahreszahl
- WW: zweistellige Nummer der Kalenderwoche: 01-53
- TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- Q: einstellige Quartalsnummer: 1-4
- TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- hh: Stunde zweistellig im 24-Stunden-Format: 00-23
- HH: Stunde zweistellig im 12-Stunden-Format: 01-12
- Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- mm: Minute zweistellig: 00-59
- ss: Sekunde zweistellig: 00-59
- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Code für das zu verwendende Format enthält; bei einer leeren Zeichenkette wird das Output-Format des Gantt-Objektes verwendet (s. VcGantt.DateOutputFormat).
	Mögliche Werte:	Name der Farbzuoordnungstabelle

ReferenceDate

Eigenschaft von VcInfoWindow

Mit dieser Eigenschaft können Sie ein Referenzdatum für das Informationsfenster erfragen oder festlegen. Für die tatsächliche Verwendung dieses Datums muss die Eigenschaft **UseReferenceDate** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Referenzdatum

UseReferenceDate

Eigenschaft von VcInfoWindow

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Informationsfenster ein Referenzdatum verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Informationsfenster verwendet (True) / verwendet kein (False) Referenzdatum Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Visible

Eigenschaft von VcInfoWindow

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Informationsfenster bei Interaktionen mit dem Knoten sichtbar sein soll.

1044 API-Referenz: VcInfoWindow

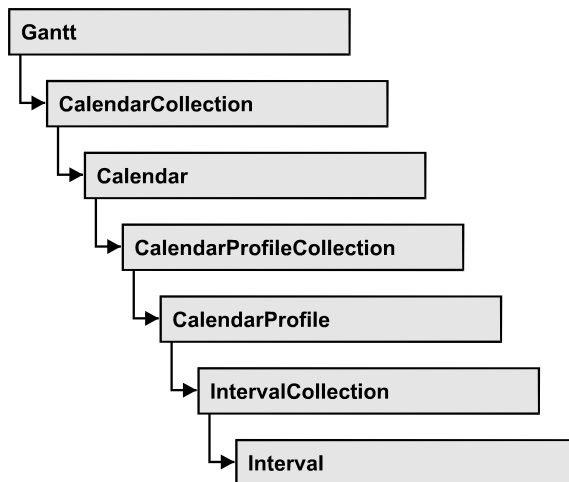
	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Informationsfenster sichtbar/unsichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcGantt1.BoxCollection
Set box = boxCltn.FirstBox
box.Visible = False
```

7.46 VcInterval



Ein ODas Objekt **VcInterval** bietet die Möglichkeit, Zeitintervalle zu definieren, die als Arbeitszeit oder Nicht-Arbeitszeit interpretiert werden. Die Unterscheidung zwischen den beiden Ausprägungen erfolgt durch die speziellen Setzungen **<WORK>** und **<NONWORK>** bei der Eigenschaft **CalendarProfileName**. Ein Intervall kann sich durch die Eigenschaft **CalendarProfileName** auch auf andere bereits definierte Kalenderprofile beziehen.

Je nach vorliegendem Intervalltyp (**vcCalendarInterval**, **vcDayProfileInterval**, **vcWeekProfileInterval**, **vcYearProfileInterval** oder **vcShiftProfileInterval**), der nicht explizit gesetzt wird, sondern sich aus dem Verwendungszusammenhang ergibt, sind nur bestimmte Eigenschaften des Objektes wirksam.

Die folgende Tabelle listet auf, welche Eigenschaften bei den einzelnen Intervalltypen einsetzbar sind:

vcCalendar-Interval	vcYearProfile-Interval	vcWeekProfile-Interval	vcDayProfile-Interval	vcShift-Interval
StartDateTime	StartMonth	StartWeekday	StartTime	Duration
EndDateTime	EndMonth	EndWeekday	EndTime	TimeUnit
	DayInEndMonth			
	DayInStartMonth			

Ein **CalendarInterval** beschreibt eine einmalige Zeitspanne innerhalb eines genau spezifizierten Zeitraums. Beispiel: 5.5.2010 11:30 Uhr bis 15.9.2010 17:00 Uhr.

Ein **YearProfileInterval** erlaubt die Vereinbarung eines jährlich wiederkehrenden Tages oder einer wiederkehrenden Zeitspanne. Beispiel: 1. 5. oder 24.12-26.12.

Ein **WeekProfileInterval** bezieht sich auf einzelne oder mehrere zusammenhängende Tage einer Woche. Beispiel: Samstag oder Montag bis Freitag

Ein **DayProfileProfileInterval** bezieht sich auf die Angabe von Zeiten innerhalb eines Tages. Beispiel: 8.00 bis 17.00 Uhr

Ein **ShiftProfile** beschreibt eine Zeitspanne, die eine bestimmte Dauer in der angegebenen Einheit **vcDay**, **vcHours**, **vcMinute** oder **vcSeconds** ohne Terminbezug darstellt. Beispiel: 4 Stunden

Eigenschaften

- BackColorAsARGB
- CalendarProfileName
- DayInEndMonth
- DayInStartMonth
- Duration
- EndDateTime
- EndMonth
- EndTime
- EndWeekday
- LineColor
- LineThickness
- LineType
- Name
- Pattern
- PatternColorAsARGB
- Specification
- StartDateTime
- StartMonth
- StartTime
- StartWeekday
- Text
- TimeUnit
- Type
- UseGraphicalAttributes

Methoden

- PutInOrderAfter

Eigenschaften

BackColorAsARGB

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Hintergrundfarbe für das Kalendergitter des Intervalls erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Die Hintergrundfarbe kann auch im Dialog **Intervalle verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}) Standardwert: &hFFD8D8D8 (grau)

CalendarProfileName

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie dem Intervall ein Kalenderprofil zuweisen oder das verwendete erfragen. Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalenderprofils
	Mögliche Werte:	Name der Farbzuoordnungstabelle

DayInEndMonth

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den Tag des letzten Monats des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcYearProfile**).

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Tag im letzten Monat
	Mögliche Werte:	Datenfeldindex

DayInStartMonth

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den Tag des ersten Monats des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcYearProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Tag im ersten Monat
	Mögliche Werte:	Datenfeldindex

Duration

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Dauer des Intervalls setzen oder erfragen. Dies gilt nur für Kalenderprofile vom Typ **vcShiftProfile**. Die Dauer kann auch im Dialog **Intervalle verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Dauer des Intervalls

EndTime

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Enddatum und -zeit des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcCalendar**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	Date	Enddatum und -zeit des Intervalls

EndMonth

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Endmonat des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcYearProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	MonthEnum Mögliche Werte: vcApril 4 vcAugust 8 vcDecember 12 vcFebruary 2 vcJanuary 1 vcJuly 7 vcJune 6 vcMarch 3 vcMay 5 vcNovember 11 vcOktober 10 vcSeptember 9	Endmonat des Intervalls April August Dezember Februar Januar Juli Juni März Mai November Oktober September

EndTime

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Endezeit des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcDayProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	Date	Endezeit des Intervalls

EndWeekday

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie letzten Wochentag des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcWeekProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	WeekdayEnum	Letzter Wochentag des Intervalls
	Mögliche Werte: vcFriday 5 vcMonday 1 vcSaturday 6 vcSunday 7 vcThursday 4 vcTuesday 2 vcWednesday 3	Wochentag Freitag Wochentag Montag Wochentag Samstag Wochentag Sonntag Wochentag Donnerstag Wochentag Dienstag Wochentag Mittwoch

LineColor

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Linienfarbe für das Kalendergitter eines Intervalls erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Intervalle verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255}

LineThickness

Nur-Lese-Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Linienstärke für das Kalendergitter eines Intervalls erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Sie können diese Eigenschaft auch im Dialog **Intervalle** verwalten festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

LineType

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den Linientyp für das Kalendergitter des Intervalls erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Intervalle verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp Standardwert: vcSolid
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0

vcLineType1 101	Linientyp 1
vcLineType10 110	Linientyp 10
vcLineType11 111	Linientyp 11
vcLineType12 112	Linientyp 12
vcLineType13 113	Linientyp 13
vcLineType14 114	Linientyp 14
vcLineType15 115	Linientyp 15
vcLineType16 116	Linientyp 16
vcLineType17 117	Linientyp 17
vcLineType18 118	Linientyp 18
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3
vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Name

Nur-Lese-Eigenschaft von VcInterval

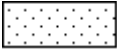
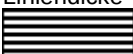
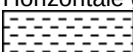
Mit dieser Eigenschaft können Sie den Namen des Intervalls erfragen. Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

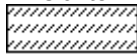
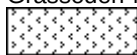
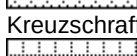
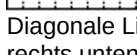
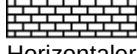
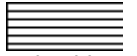
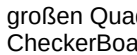
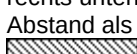
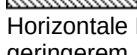
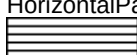
	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Intervalls
	Mögliche Werte:	Name der Farbzuoordnungstabelle

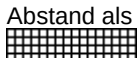
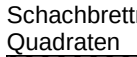
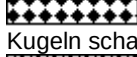
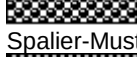
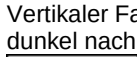
Pattern

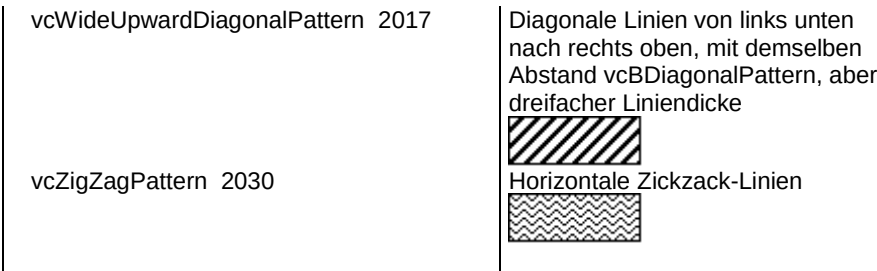
Nur-Lese-Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie das Muster für das Kalendergitter des Intervalls festlegen oder erfragen. Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
	vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
	vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
	vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 

vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern 

vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 
vcVerticalGradientPattern 62	Vertikaler Farbverlauf 
vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 



PatternColorAsARGB

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Musterfarbe für das Kalendergitter des Intervalls festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Die Musterfarbe kann auch im Dialog **Intervalle vewalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}

Specification

Nur-Lese-Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Spezifikation dieses Intervalls erfragen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Intervalls mit der Methode **VcIntervalCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Intervalls
	Mögliche Werte:	

	Name der Farbzuoordnungstabelle
--	---------------------------------

StartDateTime

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Startdatum und -zeit des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcCalendar**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	Date	Startdatum und -zeit des Intervalls

StartMonth

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Startmonat des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcYearProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	MonthEnum	Startmonat des Intervalls
	Mögliche Werte: vcApril 4 vcAugust 8 vcDecember 12 vcFebruary 2 vcJanuary 1 vcJuly 7 vcJune 6 vcMarch 3 vcMay 5 vcNovember 11 vcOktober 10 vcSeptember 9	April August Dezember Februar Januar Juli Juni März Mai November Oktober September

StartTime

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie Startzeit des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcDayProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	Date	Startzeit des Intervalls

StartWeekday

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den ersten Wochentag des Intervalls setzen oder erfragen (nur für Profile vom Typ **vcWeekProfile**). Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	WeekdayEnum	Erster Wochentag des Intervalls
	Mögliche Werte: vcFriday 5 vcMonday 1 vcSaturday 6 vcSunday 7 vcThursday 4 vcTuesday 2 vcWednesday 3	Wochentag Freitag Wochentag Montag Wochentag Samstag Wochentag Sonntag Wochentag Donnerstag Wochentag Dienstag Wochentag Mittwoch

Text

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den Text festlegen oder erfragen, der im Zeitstreifen dieses Intervalls erscheinen soll. Dies gilt nur für Kalenderprofile vom Typ **vcShiftProfile**. Der Text kann auch im Dialog **Intervalle verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Text für den Zeitstreifen
	Mögliche Werte:	Name der Farbzuoordnungstabelle

TimeUnit

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie die Zeiteinheit für das Intervall setzen oder erfragen. Dies gilt nur für Kalenderprofile vom Typ **vcVariableProfile**. Die Zeiteinheit kann auch im Dialog **Intervalle verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	TimeUnitEnum Mögliche Werte: vcDay 5 vcHour 6 vcMinute 7 vcSecond 8	Zeiteinheit Standardwert: vcDay Zeiteinheit Tag Zeiteinheit Stunde Zeiteinheit Minute Zeiteinheit Sekunde

Type

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie den Typ des Intervalls erfragen. Diese Option kann auch über die Eigenschaft **VcInterval.Duration** gesetzt werden

	Datentyp	Beschreibung
Eigenschaftswert	IntervalTypeEnum Mögliche Werte: vcCalendarInterval 139 vcDayProfileInterval 4 vcVariableProfileInterval 5 vcWeekProfileInterval 3 vcYearProfileInterval 2	Typ des Intervalls

UseGraphicalAttributes

Eigenschaft von VcInterval

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob die für dieses Intervall gesetzten grafischen Attribute verwendet werden oder nicht. Sollen sie verwendet werden, muss die Eigenschaft **VcCalendarGrid.UseGraphicalAttributesOfIntervals** auf **True** gesetzt werden.

Die Option kann auch im Dialog **Intervalle verwalten** (zu erreichen über den Dialog **Kalenderprofile verwalten**) eingestellt werden.

	Datentyp	Beschreibung
Parameter: Rückgabewert	Boolean Mögliche Werte:	Grafische Attribute des Intervalls werden(True)/werden nicht dargestellt (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
Eigenschaftswert	Boolean Mögliche Werte:	Graphische Attribute werden (True)/werden nicht (False) verwendet Standardwert: True Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Methoden

PutInOrderAfter

Methode von VcInterval

Mit dieser Methode können Sie dieses Intervall in der Auflistung aller Intervalle hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Intervall an die erste Stelle gesetzt. Die Reihenfolge der Intervalle in der Auflistung entscheidet darüber, in welcher Reihenfolge sie in Kalendern angewendet werden.

	Datentyp	Beschreibung
Parameter: refNameParam	String Mögliche Werte:	Name des Intervalls, hinter das das aktuelle Intervall gesetzt werden soll. Name der Farbzuoordnungstabelle
Rückgabewert	Void	

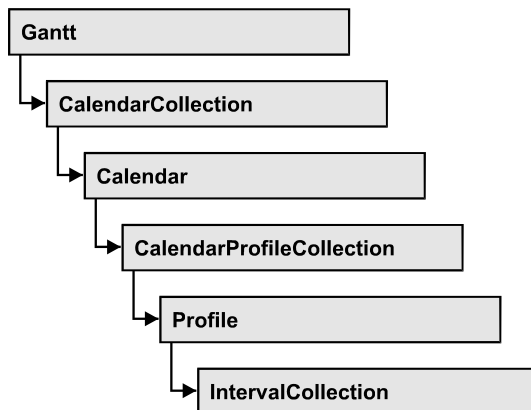
Code-Beispiel

```
Dim intvlCltn As VcIntervalCollection
Dim intvl1 As VcInterval
Dim intvl2 As VcInterval

intvlCltn = VcGantt1.IntervalCollection()
intvl1 = intvlCltn.Add("intvl1")
intvl2 = intvlCltn.Add("intvl2")
intvl1.PutInOrderAfter("intvl2")
```

```
intvlCltn.Update()
```

7.47 VcIntervalCollection



Im VcInterval-Auflistungsobjekt sind alle verfügbaren Intervalle zusammengefasst. Über **For Each interval In IntervalCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Formate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **IntervalByName** und **IntervalByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Intervalle kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Intervallen.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstInterval
- IntervalByIndex
- IntervalByName
- NextInterval
- Remove
- Update

Eigenschaften

NewEnum

Eigenschaft von VcIntervalCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Intervallobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Count

Nur-Lese-Eigenschaft von VcIntervalCollection

Mit dieser Eigenschaft können Sie die Anzahl der Intervallobjekte in der Intervall-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Intervallobjekte

Methoden

Add

Methode von VcIntervalCollection

Mit dieser Methode können Sie ein neues Intervall in der IntervalCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Intervallobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ intervalName	String Mögliche Werte:	Name des Intervalls Name der Farbzuoordnungstabelle
Rückgabewert	VcInterval	Neues Intervallobjekt

AddBySpecification

Methode von VcIntervalCollection

Mit dieser Methode können Sie ein Intervall über eine Intervall-Spezifikation erzeugen. Dies dient der Persistenz von Intervallobjekten. Die Spezifikation eines Intervallobjektes kann erfragt (siehe VcInterval-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Intervall mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Intervallspezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcInterval	Neues Intervallobjekt

Copy

Methode von VcIntervalCollection

Mit dieser Methode können Sie ein Intervall kopieren. Wenn das Intervall mit dem angegebenen Namen existiert und der Name des neuen Intervalls noch nicht verwendet wird, wird das neue Intervallobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ intervalName	String Mögliche Werte:	Name des zu kopierenden Intervalls Name der Farbzuoordnungstabelle

⇒ newIntervalName	String Mögliche Werte:	Name des neuen Intervalls Name der Farbzuoordnungstabelle
Rückgabewert	VcInterval	Intervallobjekt

FirstInterval

Methode von VcIntervalCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. das erste Intervall der Intervall-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextIntervall** über die nachfolgenden Intervalle zu iterieren. Existiert kein Intervall in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcInterval	Erstes Intervallobjekt

IntervalByIndex

Methode von VcIntervalCollection

Mit dieser Methode können Sie auf ein einzelnes Intervall über seinen Index zugreifen. Existiert kein Intervall unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ Index	Integer Mögliche Werte:	Index des Intervalls Datenfeldindex
Rückgabewert	VcInterval	Ermitteltes Intervallobjekt

IntervalByName

Methode von VcIntervalCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Intervall zugreifen. Existiert kein Intervall unter dem

angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ intervalName	String Mögliche Werte:	Name des Intervallobjekts Name der Farbzuoordnungstabelle
Rückgabewert	VcInterval	Zurückgegebenes Interval-Objekt

NextInterval

Methode von VcIntervalCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Intervalle der Intervall-Auflistung zugreifen, nachdem Sie mit der Methode **FirstInterval** den Initialwert erfasst haben. Sind alle Intervalle durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcInterval	Nachfolgendes Intervallobjekt

Remove

Methode von VcIntervalCollection

Mit dieser Methode können Sie ein Intervall löschen. Wenn das Intervall noch in irgendeinem anderen Objekt verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ intervalName	String Mögliche Werte:	Name des Intervalls Name der Farbzuoordnungstabelle
Rückgabewert	Boolean	Intervall gelöscht (True)/nicht gelöscht (False)

Update

Methode von VcIntervalCollection

Mit dieser Methode können Sie eine Intervall-Collection aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

7.48 VcLayer



Ein Layer definiert die grafische Darstellung von einem Termin (Symbollayer) oder einem Terminpaar (Rechtecklayer oder Linienlayer) in einem Knoten. Ein Layer kann durch viele Attribute - wie Form, Farbe, Höhe, Offset sowie Art und Inhalt von Beschriftungsfeldern - konfiguriert werden.

Eigenschaften

- BackColorAsARGB
- BackColorDataFieldIndex
- BackColorMapName
- CompletionDataFieldIndex
- DurationDataFieldIndex
- EndDataFieldIndex
- EndSnapTarget
- FilterName
- GraphicsFileName
- GraphicsFileNameDataFieldIndex
- GraphicsFileNameMapName
- Height
- HeightDataFieldIndex
- HeightMapName
- HorizontalOffset
- LabelSizeDependence
- LayerFormat
- LayerShape
- LegendText
- LineColor
- LineColorDataFieldIndex
- LineColorMapName
- LineThickness
- LineType
- MaximumEndDataFieldIndex
- MinimumStartDataFieldIndex

- Moveable
- Name
- NonWorkInterval
- NonWorkIntervalBackColorAsARGB
- NonWorkIntervalBackColorDataFieldIndex
- NonWorkIntervalBackColorMapName
- NonWorkIntervalLineColor
- NonWorkIntervalLineColorDataFieldIndex
- NonWorkIntervalLineColorMapName
- NonWorkIntervalLineThickness
- NonWorkIntervalLineType
- NonWorkIntervalPattern
- NonWorkIntervalPatternColorAsARGB
- NonWorkIntervalPatternColorDataFieldIndex
- NonWorkIntervalPatternColorDataFieldIndex
- NonWorkIntervalPatternDataFieldIndex
- NonWorkIntervalPatternMapName
- NonWorkIntervalShape
- ObjectDrawEventsEnabled
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- Sizeable
- Specification
- StartDataFieldIndex
- StartSnapTarget
- ThreeDEffect
- UsedAsOverlapLayer
- VerticalOffset
- VerticalOffsetDataFieldIndex
- VerticalOffsetMapName
- Visible
- VisibleInLegend

Methoden

- CalculateCurrentWidth
- PutInOrderAfter

Eigenschaften

BackColorAsARGB

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Layers festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getPatternColorAsARGB**.

Wenn in der Eigenschaft **BackColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim layer As VcLayer
```

```
Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.BackColorAsARGB = &h88FF0A06
```

BackColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **BackColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
```

```

Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maingroup")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Red")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapColor")

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcRectangleLayer
layer.BackColorMapName = "MapColor"
layer.BackColorDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

BackColorMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer FarbZuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **BackColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **BackColorAsARGB** ausgegeben.

Falls die Zuordnungstabelle transparente Farben (ARGB-Werte) enthält, eine Eigenschaft aber nur RGB-Werte verwenden kann, stellt VARCHART XGantt die bezeichnete Farbe als vollständig deckend dar.

	Datentyp	Beschreibung

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

```

1072 API-Referenz: VcLayer

```
Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Red")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapColor")

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcRectangleLayer
layer.BackColorMapName = "MapColor"
layer.BackColorDataFieldIndex = 5
VcGantt1.LayerCollection.Update
```

CompletionDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie das Datenfeld setzen oder erfragen, das den prozentualen Fertigstellungsgrad des Layers enthält.

Der durch den Layer dargestellte Endtermin wird aus dem Start-Termin-Feld, dem Endtermin-Feld bzw. der Dauer und dem Fertigstellungsgrad berechnet. Die dem Vorgang zu Grunde liegenden Daten bleiben unverändert.

Für Symbol- und Bitmaplayer ist diese Eigenschaft nicht verfügbar.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Fertigstellungsgrad enthält

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection

For Each layer In layerCltn
    layer.CompletionDataFieldIndex = 10
Next
```

DurationDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie das Datenfeld setzen oder erfragen, das die Dauer des Layers enthält.

Die Einheit der Dauer wird in Abhängigkeit von der Einstellung der Zeiteinheit auf der Eigenschaftenseite **Allgemeines** interpretiert.

Für Symbol- und Bitmaplayer ist diese Eigenschaft nicht verfügbar.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Dauer enthält

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection

For Each layer In layerCltn
    layer.DurationDataFieldIndex = 4
Next
```

EndDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie das Datenfeld setzen oder erfragen, das den Endtermin des Layers enthält, z. B. Frühestes Ende, Spätestes Ende, Geplantes Ende.

Ein Rechteck- bzw. Linien-Layer wird durch die Angabe eines Start-Termin-Feldes und eines End-Termin-Feldes oder eines Start-Termin-Feldes und einer Dauer definiert. Sind sowohl das End-Termin-Feld als auch die Dauer spezifiziert, so hat die Dauer Vorrang vor dem End-Termin-Feld. Bei Interaktionen wird dann aber nicht nur das Dauer-Feld aktualisiert, sondern auch das End-Termin-Feld.

Für Symbol- und Bitmaplayer ist diese Eigenschaft nicht verfügbar.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes, das den Endtermin enthält
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection

For Each layer In layerCltn
    layer.DurationDataFieldIndex = 3
Next
```

EndSnapTarget

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Endtermin dieses Layers als Einrastziel definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Endtermin des Layers wird/wird nicht als Einrastziel definiert. Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

FilterName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Filter setzen oder erfragen, der bestimmt, unter welcher Bedingung dieser Layer verwendet werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Filtername Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection

For Each layer In layerCltn
    layer.FilterName = "Milestone"
Next
```

GraphicsFileName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Grafikdatei setzen oder erfragen, deren Inhalt in dem Layer ausgegeben wird. Der Name muss eine gültige Grafikdatei bezeichnen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. mit eingebautem EMF)

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

Für die Anzeige der Grafikdatei muss unabhängig vom hier gesetzten Format die Eigenschaft **LayerShape** auf **vcBitmapLayer** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei
	Mögliche Werte:	Name der Farbzustimmungstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
```


1076 API-Referenz: VcLayer

```
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Pic1.bmp")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapGraphic")

map.Type = vcGraphicsFileMap
Set mapEntry = map.CreateEntry
mapEntry.GraphicsFileName = "c:\Pic1.bmp"
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcBitmapLayer
layer.GraphicsFileName = "c:\Pic1.bmp"
layer.GraphicsFileNameMapName = "MapGraphic"
layer.GraphicsFileNameDataFieldIndex = 5
VcGantt1.LayerCollection.Update
```

GraphicsFileNameDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **GraphicsFileNameMapName** benötigt wird. Ist ein gültiger Datenfeldindex angegeben und keine Zuordnungstabelle, dann wird der Grafikdateiname direkt aus dem Inhalt des angegebenen Datenfelds entnommen. Damit die Grafik angezeigt wird, muss die Eigenschaft **LayerShape** auf **vcBitmapLayer** gesetzt sein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Pic1.bmp")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
```

```

Set map = mapCltn.Add("MapGraphic")

map.Type = vcGraphicsFileMap
Set mapEntry = map.CreateEntry
mapEntry.GraphicsFileName = "c:\Pic1.bmp"
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcBitmapLayer
layer.GraphicsFileName = "c:\Pic1.bmp"
layer.GraphicsFileNameMapName = "MapGraphic"
layer.GraphicsFileNameDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

GraphicsFileNameMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Zuordnungstabelle vom Typ **vcGraphicsFileMap** oder "" setzen oder erfragen. Wenn ein Name und zusätzlich ein Datenfeldindex in der Eigenschaft **GraphicsFileNameDataFieldIndex** angegeben ist, wird eine Grafik aus der Zuordnungstabelle angezeigt. Trifft kein Datenfeldeintrag zu, wird die Grafik aus der Eigenschaft **GraphicsFileName** ausgegeben.

Damit die Grafik angezeigt wird, muss die Eigenschaft **LayerShape** auf **vcBitmapLayer** gesetzt sein.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafik-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Pic1.bmp")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapGraphic")

map.Type = vcGraphicsFileMap
Set mapEntry = map.CreateEntry
mapEntry.GraphicsFileName = "c:\Pic1.bmp"
mapCltn.Update

```

```

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcBitmapLayer
layer.GraphicsFileName = "c:\Pic1.bmp"
layer.GraphicsFileNameMapName = "MapGraphic"
layer.GraphicsFileNameDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

Height

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Höhe des Layers festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe in 1/100 mm

HeightDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **HeightMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

Die Setzung dieser Eigenschaft wird erst wirksam, nachdem die Layer-Auflistung mit der Methode **VcLayerCollection.Update()** aktualisiert wurde.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

Code-Beispiel

```

Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim layer As VcLayer

Set layer = VcGantt1.LayerCollection.FirstLayer()
Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcMillimeterMap)
Set map = mapCltn.FirstMap
layer.HeightMapName = map.Name
layer.HeightDataFieldIndex = VcGantt1.DetectFieldIndex("Maindata",
"LayerHeight")
VcGantt1.LayerCollection.Update

```

HeightMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Millimeter-Zuordnungstabelle (Typ `VcMillimeterMap`) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Millimeter-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **HeightDataFieldIndex** angegeben ist, wird die Höhe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Höhe aus der Eigenschaft **Height** ausgegeben.

Die Setzung dieser Eigenschaft wird erst wirksam, nachdem die Layer-Auflistung mit der Methode **VcLayerCollection.Update()** aktualisiert wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Millimeter-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim layer As VcLayer

Set layer = VcGantt1.LayerCollection.FirstLayer()
Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcMillimeterMap)
Set map = mapCltn.FirstMap
layer.HeightMapName = map.Name
layer.HeightDataFieldIndex = VcGantt1.DetectFieldIndex("Maingroup",
"LayerHeight")
VcGantt1.LayerCollection.Update
```

HorizontalOffset

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den horizontalen Offset des Layers festlegen oder erfragen. Dies gilt nur für Symbol- und Bitmap-Layern, bei allen anderen Layerformen hat die Einstellung des Offsets keine Auswirkung.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	horizontaler Offset in %
	Mögliche Werte:	-50 ... 50

LabelSizeDependence

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob und wie die Größe des Beschriftungsfeldes von der Größe des Layers abhängig sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	LabelSizeDependenceEnum	Abhängigkeit des Beschriftungsfeldes von der Layer-Größe
	Mögliche Werte: vcFixedToBar 1 vcTextHeightAndWidthIndependent 79 vcTextHeightIndependent 39 vcTextWidthIndependent 40	durch Layergröße beschränkt unabhängig von Texthöhe und -breite unabhängig von Texthöhe unabhängig von Textbreite

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
Set layer = layerCltn.LayerByName("Start-End")
layer.LabelSizeDependence = vcFixedToBar
```

LayerFormat

Nur-Lese-Eigenschaft von VcLayer

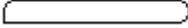
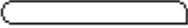
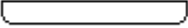
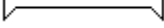
Mit dieser Methode können Sie das Format des Layers erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcLayerFormat	Layerformat

LayerShape

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Form des Layers festlegen oder erfragen. Bei den unten abgebildeten Symbolen sind schwarze Flächen durch zugewiesene Farben ersetzbar (s. (s. **BackColorAsARGB**, **Pattern** und **PatternColor**)).

Eigenschaftswert	Datentyp	Beschreibung
	LayerShapeEnum	Layerform
	Mögliche Werte:	
	vcAllRoundedRectangleLayer 61441	Alle Ecken gerundet 
	vcBitmapLayer 103007	Layerform Bitmap <Bitmap-Layer>
	vcInvisibleSymbolLayer 101000	Layer unsichtbar <unsichtbares Symt>
	vcLineLayer 2	Layerform Linie 
	vcNAndSERoundedRectangleLayer 45057	Linke und rechte obere Ecke sowie rechte untere Ecke gerundet 
	vcNAndSWRoundedRectangleLayer 28673	Linke und rechte obere Ecke sowie linke untere Ecke gerundet 
	vcNEAndSERoundedRectangleLayer 40961	Obere und untere rechte Ecke gerundet 
	vcNEAndSRoundedRectangleLayer 57345	Linke und rechte untere Ecke sowie obere rechte Ecke gerundet 
	vcNEAndSWRoundedRectangleLayer 24577	Linke untere und rechte obere Ecke gerundet 
	vcNERoundedRectangleLayer 8193	Obere rechte Ecke gerundet 
	vcNRoundedRectangleLayer 12289	Obere rechte und obere linke Ecke gerundet 
	vcNWAndSERoundedRectangleLayer 36865	Obere linke und untere rechte Ecke gerundet 
	vcNWAndSRoundedRectangleLayer 52349	Untere linke und rechte Ecke sowie obere linke Ecke gerundet 
	vcNWAndSWRoundedRectangleLayer 20481	Untere und obere linke Ecke gerundet 
	vcNWRoundedRectangleLayer 4097	Obere linke Ecke gerundet 
	vcRectangleLayer 1	Rechtecklayer 
	vcRectangleTriangleLeft 1938	Rechtecklayer, Spitze links 
	vcRectangleTriangleLeftRight 1939	Rechtecklayer, Spitze beidseitig!!! Datei E:\Manuals\Generierung\Handbuch\Bilder\div\Layer\vcRectangleTriangleLeftRight.png nicht gefunden!!!
	vcRectangleTriangleRight 1939	Rechtecklayer, Spitze rechts!!! Datei E:\Manuals\Generierung\Handbuch\Bilder\div\Layer\vcRectangleTriangleRight.png nicht gefunden!!!
	vcSERoundedRectangleLayer 32769	Untere rechte Ecke gerundet 
	vcSRoundedRectangleLayer 49153	Untere rechte und untere linke Ecke gerundet 
	vcSummaryBar1 1858	Summenbalken 

1082 API-Referenz: VcLayer

vcSummaryBar2 1859	Summenbalken
vcSummaryBar3 1860	Summenbalken
vcSummaryBar4 1861	Summenbalken
vcSWRoundedRectangleLayer 16385	Untere linke Ecke gerundet
vcSymbolLayer1 101001	Pfeil nach unten
vcSymbolLayer10 101010	Quadrat
vcSymbolLayer11 101032	Kreis
vcSymbolLayer12 101033	Pfeil nach unten in Kreis
vcSymbolLayer13 101034	Dreieck im Kreis, Spitze unten
vcSymbolLayer14 101035	Spitze rechte Klammer im Kreis
vcSymbolLayer15 101036	Schmales Dreieck im Kreis, Spitze oben
vcSymbolLayer16 101037	Dreieck im Kreis, Spitze nach rechts
vcSymbolLayer17 101038	Dreieck im Kreis, Spitze links
vcSymbolLayer18 101039	Auf die Spitze gestelltes Quadrat im Kreis
vcSymbolLayer19 101040	Zwei schmale Dreiecke im Kreis, horizontal, Spitzen
vcSymbolLayer2 101002	Dreieck, Spitze nach unten
vcSymbolLayer20 101041	Schmales Dreieck im Kreis, Spitze nach unten
vcSymbolLayer21 101042	Quadrat im Kreis
vcSymbolLayer22 103001	Kreis
vcSymbolLayer23 102031	Pfeil nach oben
vcSymbolLayer24 102034	Dreieck, Spitze nach oben
vcSymbolLayer25 102016	Linke spitze Klammer
vcSymbolLayer26 102051	Pfeil nach oben im Kreis

vcSymbolLayer27 102054	Dreieck im Kreis, Spitze nach oben
vcSymbolLayer3 101003	Rechte spitze Klammer
vcSymbolLayer4 101004	Schmales Dreieck, Spitze nach oben
vcSymbolLayer5 101005	Dreieck, Spitze nach rechts
vcSymbolLayer6 101006	Dreieck, Spitze nach links
vcSymbolLayer7 101007	Auf die Spitze gestelltes Quadrat
vcSymbolLayer8 101008	Zwei schmale Dreiecke, horizontal, Spitzen zur Mitte
vcSymbolLayer9 101009	Schmales Dreieck, Spitze nach unten
vcTriangleBottomLeftLayer 1566	Dreieck, Spitze nach links
vcTriangleBottomRightLayer 1564	Dreieck, Spitze nach rechts

LegendText

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie einem Layer einen Text zuweisen oder erfragen, der in der Legende für den jeweiligen Layer angezeigt wird. Steht hier "", dann wird der Layername (Eigenschaft **Name**) angezeigt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendentext des Layers
	Mögliche Werte:	Standardwert: " " (content of the property Name) Name der Farbzuoordnungstabelle

LineColor

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Farbe der (Rand)linie des Layers festlegen oder erfragen.

	Datentyp	Beschreibung

LineColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuoordnungstabelle in der Eigenschaft **LineColorMapName** benötigt wird. Wenn Sie hier -1 angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

LineColorMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle für die Linienfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **LineColorDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

LineThickness

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Stärke der (Rand)linie des Layers festlegen oder erfragen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert
	Mögliche Werte:	Datenfeldindex

LineType

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Typ der (Rand)linie des Layers festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp ({0...255},{0...255},{0...255})
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100 vcLineType1 101 vcLineType10 110 vcLineType11 111	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0 Linientyp 1 ----- Linientyp 10 ----- Linientyp 11 -----

vcLineType12 112	Linientyp 12 -----
vcLineType13 113	Linientyp 13 -----
vcLineType14 114	Linientyp 14 -----
vcLineType15 115	Linientyp 15 -----
vcLineType16 116	Linientyp 16 -----
vcLineType17 117	Linientyp 17 -----
vcLineType18 118	Linientyp 18 -----
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3 -----
vcLineType4 104	Linientyp 4 -----
vcLineType5 105	Linientyp 5 -----
vcLineType6 106	Linientyp 6 -----
vcLineType7 107	Linientyp 7 -----
vcLineType8 108	Linientyp 8 -----
vcLineType9 109	Linientyp 9 -----
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

MaximumEndDataFieldIndex

Eigenschaft von VcLayer

Wird diese Eigenschaft auf einen gültigen Feldindex gesetzt, dann gilt das im angesprochenen Datenfeld hinterlegte Datum mit Uhrzeit als obere Grenze für die Endzeit des Layers beim interaktiven Verschieben des Layers bzw. Knotens.

Diese Eigenschaft kann auch im Dialog **Layer-Bearbeiten** festgelegt werden.

	Datentyp	Beschreibung

MinimumStartDataFieldIndex

Eigenschaft von VcLayer

Wird diese Eigenschaft auf einen gültigen Feldindex gesetzt, dann gilt das im angesprochenen Datenfeld hinterlegte Datum mit Uhrzeit als untere Grenze für die Startzeit des Layers beim interaktiven Verschieben des Layers bzw. Knotens.

Diese Eigenschaft kann auch im Dialog **Layer-Bearbeiten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex für frühestes Startzeit

Moveable

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Layer interaktiv verschiebbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	verschiebbar (True)/ nicht verschiebbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim layer1 As VcLayer

Set layer1 = VcGantt1.Layer.LayerByName("layer1")
If chkMoveable.Value = vbUnchecked Then
    layer1.Moveable = False
Else
    layer1.Moveable = True
End If
```

Name

Nur-Lese-Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen eines Layer-Objekts erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Layers
	Mögliche Werte:	Name der Farbzordnungstabelle

Code-Beispiel

```

Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection

For Each layer In layerCltn
    List1.AddItem layer.Name
Next layer

```

NonWorkInterval**Eigenschaft von VcLayer**

Mit dieser Eigenschaft können Sie den Namen einer FarbZuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColorAsARGB** ausgegeben.

Falls die Zuordnungstabelle transparente Farben (ARGB-Werte) enthält, eine Eigenschaft aber nur RGB-Werte verwenden kann, stellt VARCHART XGantt die bezeichnete Farbe als vollständig deckend dar.

	Datentyp	Beschreibung

NonWorkIntervalBackColorAsARGB**Eigenschaft von VcLayer**

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Layers festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines

RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getPatternColorAsARGB**.

Wenn in der Eigenschaft **BackColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}

Code-Beispiel

```
Dim layer As VcLayer

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.BackColorAsARGB = &h88FF0A06
```

NonWorkIntervalBackColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **BackColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Red")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapColor")
```

```

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcRectangleLayer
layer.BackColorMapName = "MapColor"
layer.BackColorDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

NonWorkIntervalBackColorMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer FarbZuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuhordnungs-tabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **BackColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **BackColorAsARGB** ausgegeben.

Falls die Zuordnungstabelle transparente Farben (ARGB-Werte) enthält, eine Eigenschaft aber nur RGB-Werte verwenden kann, stellt VARCHART XGantt die bezeichnete Farbe als vollständig deckend dar.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Red")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapColor")

```

```

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.LayerShape = vcRectangleLayer
layer.BackColorMapName = "MapColor"
layer.BackColorDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

NonWorkIntervalLineColor

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Farbe der (Rand)linie des Layers festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

NonWorkIntervalLineColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuoordnungstabelle in der Eigenschaft **LineColorMapName** benötigt wird. Wenn Sie hier -1 angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

NonWorkIntervalLineColorMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle für die Linienfarbe setzen oder erfragen. Wird hier "" oder bei der

Eigenschaft **LineColorDataFieldIndex** -1 angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuordnungstabelle
	Mögliche Werte:	Name der Farbzuordnungstabelle

NonWorkIntervalLineThickness

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Stärke der (Rand)linie des Layers festlegen oder erfragen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke
	Mögliche Werte:	LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert

Datenfeldindex

NonWorkIntervalLineType

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Typ der (Rand)linie des Layers festlegen oder erfragen.

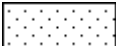
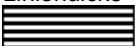
	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp ({0...255},{0...255},{0...255})
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100 vcLineType1 101 vcLineType10 110 vcLineType11 111 vcLineType12 112 vcLineType13 113 vcLineType14 114 vcLineType15 115 vcLineType16 116 vcLineType17 117 vcLineType18 118 vcLineType2 102 vcLineType3 103 vcLineType4 104 vcLineType5 105 vcLineType6 106 vcLineType7 107 vcLineType8 108 vcLineType9 109 vcNone 1	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0 Linientyp 1 Linientyp 10 Linientyp 11 Linientyp 12 Linientyp 13 Linientyp 14 Linientyp 15 Linientyp 16 Linientyp 17 Linientyp 18 Linientyp 2 Linientyp 3 Linientyp 4 Linientyp 5 Linientyp 6 Linientyp 7 Linientyp 8 Linientyp 9 Kein Linientyp

vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

NonWorkIntervalPattern

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie das Muster des Layers festlegen oder erfragen. Wenn in der Eigenschaft **PatternMapName** eine Zuordnungstabelle angegeben ist, steuert diese das Muster in Abhängigkeit von den Daten.

Eigenschaftswert	Datentyp	Beschreibung
	FillPatternEnum	Mustertyp Standardwert: Wie im Dialog definiert
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 

vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
vcDivotPattern 2036	Grassoden-Muster
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern

vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel

vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

NonWorkIntervalPatternColorAsARGB

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Musterfarbe des Layers festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getBackColorAsARGB**.

Wenn in der Eigenschaft **PatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},0...255},{0...255},{0...255}

Code-Beispiel

```
Dim layer As VcLayer

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.PatternColorAsARGB = &h88FF0A06
```

NonWorkIntervalPatternColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

NonWorkIntervalPatternColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

NonWorkIntervalPatternDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;;8;Horizontal")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapPattern")

map.Type = vcPatternMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Diagonal"
mapEntry.Pattern = vcBDDiagonalPattern
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Horizontal"
mapEntry.Pattern = vcHorizontalPattern
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.PatternMapName = "MapPattern"
layer.PatternDataFieldIndex = 5
VcGantt1.LayerCollection.Update

```

NonWorkIntervalPatternMapName**Eigenschaft von VcLayer**

Mit dieser Eigenschaft können Sie den Namen einer Musterzuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn der Name einer Farbzuhordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternDataFieldIndex** angegeben sind, wird das Muster des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **Pattern** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Code-Beispiel

```

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim layer As VcLayer
Dim mapCltn As VcMapCollection
Dim map As VcMap

```



```
Dim mapEntry As VcMapEntry

Set dataTable = VcGantt1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection
Set dataRec1 = dataRecCltn.Add("1;Node 1;01.01.14;;8;Horizontal")
VcGantt1.EndLoading

Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapPattern")

map.Type = vcPatternMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Diagonal"
mapEntry.Pattern = vcBDiagonalPattern
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Horizontal"
mapEntry.Pattern = vcHorizontalPattern
mapCltn.Update

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.PatternMapName = "MapPattern"
layer.PatternDataFieldIndex = 5
VcGantt1.LayerCollection.Update
```

NonWorkIntervalShape

Eigenschaft von VcLayer

Mit dieser Eigenschaft legen Sie die Form für die Darstellung arbeitsfreier Zeiten in Rechtecklayern fest. Die Auswahl kann auch im Dialog **Layer bearbeiten** getroffen werden.

	Datentyp	Beschreibung
Eigenschaftswert	NonWorkIntervalShapeEnum Mögliche Werte: vcEmptyArea 2 vcLine 1 vcNo 0 vcRectangle 112	Form für arbeitsfreie Zeiten in Rechtecklayern arbeitsfreie Zeiten werden als leerer Bereich dargestellt arbeitsfreie Zeiten werden als Linie dargestellt arbeitsfreie Zeiten werden nicht dargestellt arbeitsfreie Zeiten werden als Rechteck dargestellt

ObjectDrawEventsEnabled

Eigenschaft von VcLayer

Wenn diese Eigenschaft auf **True** gesetzt wird, können bei Knoten, die mit diesem Layer gezeichnet werden oder bei Skalenbezeichnungen, die Ereignisse **OnObjectDrawEx** und **OnObjectDrawCompleteEx** auftreten.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	ObjectDraw-Ereignisse eingeschaltet (True) / ausgeschaltet (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

PatternColorAsARGB

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Musterfarbe des Layers festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getBackColorAsARGB**.

Wenn in der Eigenschaft **PatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim layer As VcLayer

Set layer = VcGantt1.LayerCollection.LayerByIndex(0)
layer.PatternColorAsARGB = &h88FF0A06
```

PatternColorDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung

PatternColorMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer FarbZuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuhordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColorAsARGB** ausgegeben.

Falls die Zuordnungstabelle transparente Farben (ARGB-Werte) enthält, eine Eigenschaft aber nur RGB-Werte verwenden kann, stellt VARCHART XGantt die bezeichnete Farbe als vollständig deckend dar.

	Datentyp	Beschreibung

Sizeable

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Layerbreite interaktiv veränderbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	LayerSizeabilityEnum	Art der Veränderbarkeit der Layerbreite Standardwert: True

Code-Beispiel

```
Dim layer1 As VcLayer

Set layer1 = VcGantt1.Layer.LayerByName("layer1")
If chkSizeable.Value = vbUnchecked Then
    layer1.Sizeable = vcSizeableNone
Else
    layer1.Sizeable = vcSizeableLeftRight
End If
```

Specification

Nur-Lese-Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie die Spezifikation dieses Layers auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung eines Layers mit der Methode **VcLayerCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Layerspezifikation
	Mögliche Werte:	Name der Farbzuoordnungstabelle

StartDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie das Datenfeld setzen oder erfragen, das den Anfangstermin des Layers enthält, z. B. Frühester Anfang, Spätester Anfang, Geplanter Start.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes, das den Anfangstermin enthält
	Mögliche Werte:	Datenfeldindex

StartSnapTarget

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Starttermin dieses Layers als Einrastziel definiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Starttermin des Layers wird/wird nicht als Einrastziel definiert.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar

Gruppen sichtbar/unsichtbar

ThreeDEffect

Eigenschaft von VcLayer

Mit dieser Eigenschaft wird festgelegt oder erfragt, ob der Layer einen 3D-Effekt erhält oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	3D-Effekt eingeschaltet (True)/ausgeschaltet (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

UsedAsOverlapLayer

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Layer als Overlap-Layer verwendet werden soll. Overlap-Layer treten in Erscheinung, wenn eine Überlappung (und damit die Verdeckung) von Knoten anzuzeigen ist und werden mit dem Ausmaß der Überlappung größer oder kleiner. Vgl. dazu auch die Eigenschaften **VcGantt.OverlapLayerEnabled** und **VcGantt.OverlapLayerName**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	True: Layer wird als überlappungslayer verwendet; False: Layer wird nicht als Überlappungslayer verwendet. Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim layer1 As VcLayer

Set layer1 = VcGantt1.Layer.LayerByName("layer1")
If chkOverlapLayer.Value = vbUnchecked Then
    layer1.UsedAsOverlapLayer = False
Else
    layer1.UsedAsOverlapLayer = True
End If
```

End If

VerticalOffset

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den vertikalen Offset des Layers festlegen oder erfragen. Wenn in der Eigenschaft **VerticalOffsetMapName** eine Zuordnungstabelle angegeben ist, steuert diese den vertikalen Offset in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Long	vertikaler Offset in 1/100 mm

VerticalOffsetDataFieldIndex

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **VerticalOffsetMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

VerticalOffsetMapName

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie den Namen einer Millimeter-Zuordnungstabelle (Typ vcMillimeterMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Millimeter-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **VerticalOffsetDataFieldIndex** angegeben ist, wird der vertikale Offset des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird der vertikale Offset aus der Eigenschaft **VerticalOffset** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Millimeter-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

Visible

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Layer sichtbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Layer sichtbar/nicht sichtbar Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
Set layer = layerCltn.LayerByName("Start-End")
layer.Visible = False
```

VisibleInLegend

Eigenschaft von VcLayer

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Layer-Objekt in der Legende sichtbar ist. Diese Eigenschaft kann auch im Dialog **Balkenaussehen verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Layer in Legende sichtbar (True)/nicht sichtbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.layerCollection
Set layer = layerCltn.layerByName("Standard")

layer.VisibleInLegend = False

```

Methoden

CalculateCurrentWidth

Methode von VcLayer

Mit dieser Methode können Sie die aktuelle Breite des Layers in 1/100 mm ermitteln, der zur angegebenen Layerdefinition am angegebenen Knoten gehört. Wenn kein Layer an diesem Knoten zur angegebenen Layerdefinition sichtbar ist z.B. aufgrund von Filterbedingungen, dann wird **-1** zurückgegeben.

	Datentyp	Beschreibung
Parameter: ⇒ node	VcNode	Knoten, in dessen Layer-Definition der Layer gesucht wird.
Rückgabewert	Long	Breite des Layers in 1/100 mm

PutInOrderAfter

Methode von VcLayer

Mit dieser Methode können Sie den Layer in der Auflistung aller Layer hinter den durch den Namen angegebenen setzen. Wenn als Name "" übergeben wird, wird der Layer an die erste Stelle gesetzt. Die Reihenfolge der Layer in der Auflistung entscheidet darüber, in welcher Reihenfolge sie dargestellt werden.

	Datentyp	Beschreibung
Parameter: ⇒ refName	String	Name des Layers, hinter den der aktuelle Layer eingeordnet werden soll.
	Mögliche Werte:	Name der Farbzusordnungstabelle

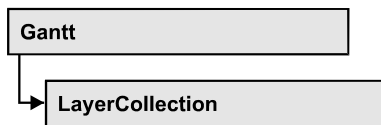
Rückgabewert	Void	
---------------------	------	--

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer1 As VcLayer
Dim layer2 As VcLayer

layerCltn = VcGantt1.LayerCollection()
layer1 = layerCltn.Add("layer1")
layer2 = layerCltn.Add("layer2")
layer1.PutInOrderAfter("layer2")
layerCltn.Update()
```

7.49 VcLayerCollection



In einem Objekt vom Typ `VcLayerCollection` sind automatisch alle verfügbaren Layer zusammengefasst. Über **For Each layer In LayerCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Layer zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **LayerByName** und **LayerByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Layer kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Layern.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstLayer`
- `LayerByIndex`
- `LayerByName`
- `NextLayer`
- `Remove`
- `Update`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcLayerCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Layer-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim layer As VcBoxLayer

For Each layer In VcGantt1.LayerCollection
    Debug.Print layer.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcLayerCollection

Mit dieser Eigenschaft kann die Anzahl der Layer in der Layer-Auflistung abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Layer

Code-Beispiel

```
Dim numberOfLayers As Long

numberOfLayers = VcGantt1.LayerCollection.Count
```

Methoden

Add

Methode von VcLayerCollection

Mit dieser Methode können Sie einen neuen Layer in der LayerCollection anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Layerobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ LayerName	String	Name des Layers
Mögliche Werte:		

		Name der Farbzuoordnungstabelle
Rückgabewert	VcLayer	Neues Layerobjekt

Code-Beispiel

```
Set newLayer = VcGantt1.LayerCollection.Add("test1")
```

AddBySpecification**Methode von VcLayerCollection**

Mit dieser Methode können Sie einen Layer über eine Layer-Spezifikation erzeugen. Dies dient der Persistenz von Layerobjekten. Die Spezifikation eines Layers kann erfragt (siehe VcLayer-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann der gleiche Layer mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Layerspezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcLayer	Neues Layerobjekt

Copy**Methode von VcLayerCollection**

Mit dieser Methode können Sie einen Layer kopieren. Wenn der Layer mit dem angegebenen Namen existiert und der Name des neuen Layers noch nicht verwendet wird, wird das neue Layerobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ LayerName	String Mögliche Werte:	Name des zu kopierenden Layers Name der Farbzuoordnungstabelle
⇒ newLayerName	String Mögliche Werte:	Name des neuen Layers Name der Farbzuoordnungstabelle

Rückgabewert	VcLayer	Layerobjekt
---------------------	---------	-------------

FirstLayer

Methode von VcLayerCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Layer der Layer-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextLayer** über die nachfolgenden Layer zu iterieren. Existiert kein Layer in der Layer-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLayer	Erster Layer

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
Set layer = layerCltn.FirstLayer
```

LayerByIndex

Methode von VcLayerCollection

Mit dieser Methode können Sie auf einen einzelnen Layer über seinen Index zugreifen. Existiert kein Layer unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Layers
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcLayer	Ermitteltes Layerobjekt

LayerByName

Methode von VcLayerCollection

Mit dieser Methode können Sie unter Verwendung des Layernamens auf einen bestimmten Layer zugreifen. Existiert kein Layer unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇨ layerName	String	Name des Layers
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcLayer	Layer

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
Set layer = layerCltn.LayerByName("Start-End")
```

NextLayer

Methode von VcLayerCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Layer der LayerCollection zugreifen, nachdem Sie mit der Methode **FirstLayer** den Initialwert erfasst haben. Sind alle Layer durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLayer	Nachfolgender Layer

Code-Beispiel

```
Dim layerCltn As VcLayerCollection
Dim layer As VcLayer

Set layerCltn = VcGantt1.LayerCollection
Set layer = layerCltn.FirstLayer
While Not layer Is Nothing
    list1.AddItem layer.Name
    Set layer = layerCltn.NextLayer
Wend
```

Remove

Methode von VcLayerCollection

Mit dieser Methode können Sie einen Layer löschen. Wenn der Layer noch in irgendeinem anderen Objekt benutzt wird, kann er nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇨ LayerName	String Mögliche Werte:	Name des Layers Name der Farbzordnungstabelle
Rückgabewert	Boolean	Layer gelöscht (True)/nicht gelöscht (False)

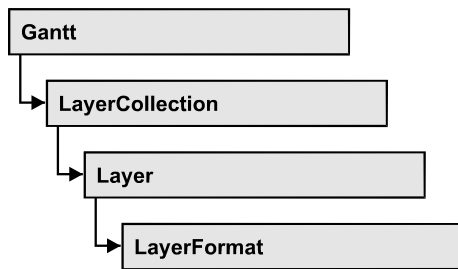
Update

Methode von VcLayerCollection

Mit dieser Methode können Sie eine LayerCollection aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

7.50 VcLayerFormat



Ein Layerformat definiert die Beschriftung eines Layers.

Eigenschaften

- _NewEnum
- FormatField
- FormatFieldCount

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcLayerFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Layer-Formatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```

Dim formatField As VcLayerFormatField

For Each formatField In format

```



```
Debug.Print formatField.Index  
Next
```

FormatField

Nur-Lese-Eigenschaft von VcLayerFormat

Mit dieser Eigenschaft können Sie ein VcLayerFormatField-Objekt per Index ansprechen. Der Index muss im Bereich von 0 bis .FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht wie in den bisherigen Feldeigenschaften von 1 bis .FormatFieldCount.

	Datentyp	Beschreibung
Parameter: Index	Integer Mögliche Werte:	Index des Layerformatfeldes 0FormatFieldCount-1 Datenfeldindex
Eigenschaftswert	VcLayerFormatField	Layerformatfeld

FormatFieldCount

Nur-Lese-Eigenschaft von VcLayerFormat

Diese Eigenschaft gibt die Anzahl der Felder eines Layerformats an.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Anzahl der Felder im Layerformat Datenfeldindex

Methoden

CopyFormatField

Methode von VcLayerFormat

Mit dieser Methode können Sie ein Layerformatfeld kopieren. Das neue VcLayerFormatField-Objekt wird zurückgegeben. Es erhält den nächsten, noch nicht vergebenen Index.

	Datentyp	Beschreibung
Parameter: ⇒ position	FormatFieldPositionEnum Mögliche Werte: vcAbove 1 vcBelow 3 vcLeftOf 0 vcOutsideAbove 9 vcOutsideBelow 11 vcOutsideLeftOf 8 vcOutsideRightOf 12 vcRightOf 4	Position des neuen Layerformatfeldes oberhalb unterhalb links von außerhalb, oberhalb außerhalb, unterhalb außerhalb, links von außerhalb, rechts von rechts von
⇒ refIndex	Integer Mögliche Werte:	Index des Referenz-Layerformatfeldes Datenfeldindex
Rückgabewert	VcLayerFormatField	Layerformatfeld-Objekt

RemoveFormatField

Methode von VcLayerFormat

Mit dieser Methode können Sie ein Layerformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Layerformatfelder neu festgesetzt, so dass sie wieder fortlaufend numeriert sind.

	Datentyp	Beschreibung
Parameter: ⇒ Index	Integer Mögliche Werte:	Index des zu löschenden Layerformatfeldes Datenfeldindex

7.51 VcLayerFormatField

Ein Objekt vom Typ VcLayerFormatField stellt ein Layerformatfeld, also ein Feld eines VcLayerFormat-Objekts, dar. Ein Layerformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Layerformat untergebracht ist.

Eigenschaften

- Alignment
- BottomMargin
- BottomMargin
- ConstantText
- FormatName
- Index
- LeftMargin
- LeftMargin
- MinimumWidth
- Priority
- RightMargin
- RightMargin
- SuppressTruncatedText
- TextDataFieldIndex
- TextFont
- TextFontColor
- TextFontColorDataFieldIndex
- TextFontColorMapName
- TextFontDataFieldIndex
- TextFontMapName
- TextLineCount
- TextLineCountDataFieldIndex
- TextLineCountMapName
- TopMargin
- TopMargin

Methoden

- CalculateLineCount

Eigenschaften

Alignment

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Layerformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum	Ausrichtung des Feldinhalts
	Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

BottomMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite des unteren Randes des Layerformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des unteren Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex
Eigenschaftswert	Integer	Breite des unteren Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex

BottomMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des unteren Randes des Layerformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Layerformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des unteren Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex
Eigenschaftswert	Integer	Breite des unteren Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex

ConstantText

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Layerformatfeld ausgeben, wenn die Eigenschaft **TextDataFieldIndex** auf **-1** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text
	Mögliche Werte:	Name der Farbzusordnungstabelle

FormatName

Nur-Lese-Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Namen des Layerformats erfragen, zu dem dieses Layerformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Layerformats
	Mögliche Werte:	Name der Farbzordnungstabelle

Index

Nur-Lese-Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Index des Layerformatfelds im zugehörigen Layerformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Layerformatfeldes
	Mögliche Werte:	Datenfeldindex

LeftMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite des linken Randes des Layerformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des linken Randes des Layerformatfeldes in mm 0...9
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Integer	Breite des linken Randes des Layerformatfeldes in mm 0...9
	Mögliche Werte:	Datenfeldindex

LeftMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des linken Randes des Layerformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Layerformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des linken Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex
Eigenschaftswert	Integer	Breite des linken Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex

MinimumWidth

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Layerformatfeldes in mm erfragen oder festlegen, wenn die Größe des Beschriftungsfeldes dies zulässt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Breite des Layerformatfeldes in mm 0 ... 99 Mögliche Werte: Datenfeldindex

Priority

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Priorität des Layerformatfeldes erfragen oder festlegen. Über die Priorität können Sie die Aufteilung des

vorhandenen Platzes beeinflussen. Felder mit höherer Priorität erhalten bevorzugt den Platz, den sie benötigen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Priorität des Layerformatfeldes
	Mögliche Werte:	Datenfeldindex

RightMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite des rechten Randes des Layerformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des rechten Randes des Layerformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex
Eigenschaftswert	Integer	Breite des rechten Randes des Layerformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

RightMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des rechten Randes des Layerformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Layerformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer Mögliche Werte:	Breite des rechten Randes des Layerformatfeldes in mm 0...9 Datenfeldindex
Eigenschaftswert	Integer Mögliche Werte:	Breite des rechten Randes des Layerformatfeldes in mm 0...9 Datenfeldindex

SuppressTruncatedText

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob ein Text, der nicht ganz in das Layerformatfeld paßt, unterdrückt oder abgeschnitten werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Eigenschaft in Kraft (True)/ nicht in Kraft (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

TextDataFieldIndex

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Index des Datenfelds, dessen Inhalt in dem Layerformatfeld dargestellt werden soll, erfragen oder setzen. Falls der Index den Wert **-1** besitzt, wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Index des Datenfeldes Datenfeldindex

TextFont

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Layerformatfeldes festlegen oder erfragen. Wenn in der Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Schriftart in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Layerformatfelds

TextFontColor

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Layerformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Layerformatfelds Standardwert: -1

TextFontColorDataFieldIndex

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Schriftfarbe-Zuordnungstabelle in Verbindung mit der Eigenschaft **TextFontColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextFontColorMapName

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzordnungstabelle (Typ vcColorMap) für die Schriftfarbe setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzordnungstabelle und zusätzlich ein Datenfeldindex in Verbindung mit der Eigenschaft **TextFontColorDataFieldIndex** angegeben ist, wird die Schriftfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **TextFontColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schriftfarben-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzordnungstabelle

TextFontDataFieldIndex

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der Verbindung mit der Eigenschaft **TextFontMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextFontMapName

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Namen einer Schrift-Zuordnungstabelle (Typ vcFontMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Schrift-Zuordnungstabelle und zusätzlich ein Datenfeldindex in Verbindung mit der

Eigenschaft **TextFontDataFieldIndex** angegeben ist, wird die Schriftart aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Schriftart aus der Eigenschaft **TextFont** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schrift-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzordnungstabelle

TextLineCount

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Zeilenanzahl erfragen oder setzen, wenn die Größe des Beschriftungsfeldes dies zulässt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeilenzahl
	Mögliche Werte:	Datenfeldindex

TextLineCountDataFieldIndex

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer numerischen Zuordnungstabelle für die Anzahl der Textzeilen in Verbindung mit der Eigenschaft **TextLineCountMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextLineCountMapName

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie den Namen einer numerischen Zuordnungstabelle für die die Anzahl von Textzeilen setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn der Name einer Zuordnungstabelle und zusätzlich ein Datenfeldindex in Verbindung mit der Eigenschaft **TextLineCountDataFieldIndex** angegeben ist, wird die Anzahl der Textzeilen aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Anzahl der Textzeilen aus der Eigenschaft **TextLineCount** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der numerischen Zuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

TopMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite des oberen Randes des Layerformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des oberen Randes des Layerformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex
Eigenschaftswert	Integer	Breite des oberen Randes des Layerformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

TopMargin

Eigenschaft von VcLayerFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des oberen Randes des Layerformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Layerformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	Breite des oberen Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex
Eigenschaftswert	Integer	Breite des oberen Randes des Layerformatfeldes in mm 0...9 Mögliche Werte: Datenfeldindex

Methoden

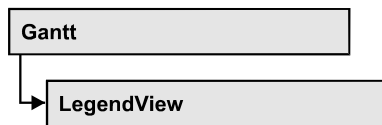
CalculateLineCount

Methode von VcLayerFormatField

Gilt nur für außenliegende Felder eines Layers: Mit dieser Methode können Sie die Anzahl der Zeilen im Layerformatfeld des angegebenen Knotens ermitteln, die der Text bei der aktuellen Größe des Feldes und der Schriftgröße einnehmen wird. Bei innenliegenden Feldern wird -1 zurückgegeben. Das Ergebnis dieser Methode kann zur Weiterverarbeitung in ein Datenfeld des Knotens geschrieben werden, um die Anzahl angezeigter Zeilen im gleichen Formatfeld zu steuern (S. Dialog **Layerformat bearbeiten** -> **Zeilenanzahl**).

	Datentyp	Beschreibung
Parameter: ⇒ node	VcNode	Knoten
Rückgabewert	Long	Anzahl der Textzeilen

7.52 VcLegendView



Ein Objekt vom Typ **VcLegendView** bezeichnet das Legendenansicht-Fenster.

Eigenschaften

- Border
- BorderColor
- Height
- HeightActualValue
- Left
- LeftActualValue
- ParentHWnd
- ScrollBarMode
- Top
- TopActualValue
- Visible
- Width
- WidthActualValue
- WindowMode

Methoden

- Update

Eigenschaften

Border

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann gesetzt oder erfragt werden, ob die Legendenansicht einen Rahmen besitzt (nicht im Modus **vcPopupWindow**). Die Rahmenfarbe ist **Color.Black**. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rahmen um die Legendenansicht (True)/kein Rahmen um die Legendenansicht (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.LegendView.Mode = vcNotFixed
VcGantt1.LegendView.Border = True
```

BorderColor**Eigenschaft von VcLegendView**

Diese Eigenschaft setzt/liefert die des ggf. sichtbaren Rahmens.

	Datentyp	Beschreibung
Eigenschaftswert	Color RGB {{0...255},{0...255},{0...255}}	RGB-Farbwerte {{0...255},{0...255},{0...255}}

Height**Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann die vertikale Ausdehnung der Legendenansicht erfragt werden. In den Modi **vcFixedAtTop**, **vcFixedAtBottom**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe der Legendenansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.Height = 100
```

HeightActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellte vertikale Ausdehnung der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Höhe der Legendenansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.HeightActualValue = 300
```

Left

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die linke Position der Legendenansicht erfragt werden. In den Modi **vcLVNotFixed** und **vcLVPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linke Position der Legendenansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.Left = 200
```

LeftActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellte linke Position der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche linke Position der Legendenansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.LeftActualValue = 150
```

ParentHwnd

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann im Modus **vcLVNotFixed** das Hwnd-Handle des Vaterfensters festgelegt werden, wenn die Legendenansicht beispielsweise in einem selbst implementierten Rahmenfenster erscheinen soll. Standardmäßig steht dies auf dem Hwnd-Handle des Vaterfensters des VARCHART-ActiveX-Hauptfensters. Diese Eigenschaft kann nur zur Laufzeit verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_HANDLE	Zugriffsnummer

Code-Beispiel

```
MsgBox (VcGantt1.legendview.ParentHWND)
```

ScrollBarMode

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann der Scrollbarmodus der Legendenansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LegendViewScrollBarModeEnum	Scrollbarmodus Standardwert: NoScrollBar
	Mögliche Werte: vcAutomaticScrollBar 3 vcHorizontalScrollBar 1 vcNoScrollBar 0 vcVerticalScrollBar 2	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig. Anzeige einer horizontalen Bildlaufleiste, wenn nötig. Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt. Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcGantt1.LegendView.ScrollBarMode = vcAutomaticScrollBar
```

Top

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die obere Position der Legendenansicht erfragt werden. In den Modi **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Obere Position der Legendenansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.Top = 20
```

TopActualValue**Nur-Lese-Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann die tatsächlich dargestellte obere Position der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche obere Position der Legendenansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.TopActualValue = 40
```

Visible**Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann festgelegt oder erfragt werden, ob die Legendenansicht sichtbar ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Legendenansicht sichtbar (True)/unsichtbar (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.LegendView.Visible = True
```

Width**Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann die horizontale Ausdehnung der Legendenansicht erfragt werden. In den Modi **vcFixedAtLeft**, **vcFixedAtRight**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann diese Eigenschaft außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Horizontale Ausdehnung der Legendenansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.Width = 200
```

WidthActualValue**Nur-Lese-Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann die tatsächlich dargestellt horizontale Ausdehnung der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche horizontale Ausdehnung der Legendenansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.WidthActualValue = 600
```

WindowMode**Eigenschaft von VcLegendView**

Mit dieser Eigenschaft kann der Modus der Legendenansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LegendViewWindowModeEnum	Modus der Legendenansicht Standardwert: vcPopupWindow
	Mögliche Werte:	
	vcFixedAtBottom 4	Die Legendenansicht wird unten im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
	vcFixedAtLeft 1	Die Legendenansicht wird links im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
	vcFixedAtRight 2	Die Legendenansicht wird rechts im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
	vcFixedAtTop 3	Die Legendenansicht wird oben im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
	vcNotFixed 5	Die Legendenansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des VARCHART ActiveX und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft VcWorldView.ParentHwnd geändert werden.
	vcPopupWindow 6	Die Legendenansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Es kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die Schließen -Schaltfläche in der Titelleiste ausgeschaltet werden.

Code-Beispiel

```
VcGantt1.LegendView.WindowMode = vcNotFixed
```

Methoden

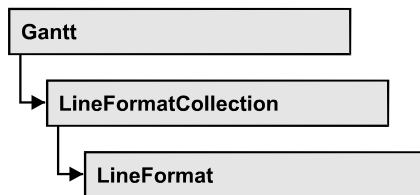
Update

Methode von VcLegendView

Mit dieser Methode wird die Legende aktualisiert.

	Datentyp	Beschreibung

7.53 VcLineFormat



Ein Objekt vom Typ VcLineFormat legt Inhalt und Erscheinungsbild von Linien fest, z.B. im Terminliniengitter.

Eigenschaften

- _NewEnum
- FormatField
- FormatFieldCount
- Name
- Specification

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcLineFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Linienformatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim formatField As VcLineFormatField
```

```

For Each formatField In format
    Debug.Print formatField.Index
Next

```

FormatField

Nur-Lese-Eigenschaft von VcLineFormat

Mit dieser Eigenschaft können Sie ein VcLineFormatField-Objekt per Index holen. Der Index muss im Bereich von 0 bis FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht wie in den bisherigen Feldeigenschaften von 1 bis .FormatFieldCount!

	Datentyp	Beschreibung
Parameter: index	Integer Mögliche Werte:	Index des Linienformatfeldes 0FormatFieldCount-1 Datenfeldindex
Eigenschaftswert	VcNodeFormatField	Knotenformatfeld

FormatFieldCount

Nur-Lese-Eigenschaft von VcLineFormat

Mit dieser Eigenschaft können Sie die Anzahl der Formatfelder dieses Linienformats erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Anzahl der Felder im Linienformat Datenfeldindex

Code-Beispiel

```

Dim format As VcLineFormat
Dim numberOfColumns As Integer

Set format = VcGantt1.Line.LineFormatCollection.FormatByName("StandardList")
numberOfColumns = FormatFieldCount

```

Name

Eigenschaft von VcLineFormat

Mit dieser Eigenschaft können Sie den Namen des Linienformats erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Linienformatname
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim format As VcLineFormat
Dim formatName As String

Set format = VcGantt1.Line.LineFormatCollection.FirstFormat
formatName = format.Name
```

Specification

Nur-Lese-Eigenschaft von VcLineFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Linienformats erfragen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Linienformats mit der Methode **VcLineFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Linienformats
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Methoden

CopyFormatField

Methode von VcLineFormat

Mit dieser Methode können Sie ein Linienformatfeld kopieren. Das neue VcLineFormatField-Objekt wird zurückgegeben. Es erhält den nächsten, noch nicht vergebenen Index.

	Datentyp	Beschreibung
Parameter: ⇒ position	FormatFieldPositionEnum Mögliche Werte: vcAbove 1 vcBelow 3 vcLeftOf 0 vcOutsideAbove 9 vcOutsideBelow 11 vcOutsideLeftOf 8 vcOutsideRightOf 12 vcRightOf 4	Position des neuen Linienformatfeldes oberhalb unterhalb links von außerhalb, oberhalb außerhalb, unterhalb außerhalb, links von außerhalb, rechts von rechts von
⇒ refIndex	Integer Mögliche Werte:	Index des Referenz-Linienformatfeldes Datenfeldindex
Rückgabewert	VcLineFormatField	Linienformatfeld-Objekt

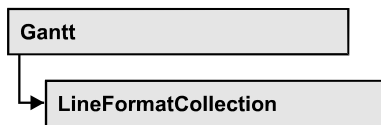
RemoveFormatField

Methode von VcLineFormat

Mit dieser Methode können Sie ein Linienformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Knotenformatfelder neu festgesetzt, so dass sie wieder fortlaufend nummeriert sind.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des zu löschenden Linienformatfeldes Datenfeldindex
Rückgabewert	Void	

7.54 VcLineFormatCollection



In einem Objekt vom Typ `VcLineFormatCollection` sind alle verfügbaren Linienformate zusammengefasst. Über **For Each lineFormat In LineFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Formate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **FormatByName** und **FormatByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Linienformate kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Linienformaten.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstFormat`
- `FormatByIndex`
- `FormatByName`
- `NextFormat`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcLineFormatCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Linienformatobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode

GetEnumerator angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim format As VcLineFormat

For Each format In VcGantt1.LineFormatCollection
    Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcLineFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Linienformatobjekte in der LinienFormat-Auflistung abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Linienformate

Code-Beispiel

```
Dim lineFormatCltn As VcLineFormatCollection
Dim numberOfLineformats As Long

Set lineFormatCltn = VcGantt1.LineFormatCollection
Dim numberOfLineformats = lineFormatCltn.Count
```

Methoden

Add

Methode von VcLineFormatCollection

Mit dieser Methode können Sie ein neues Linienformat in der Linienformat-Auflistung anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Linienformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String	Name des Linienformats
	Mögliche Werte:	

		Name der Farbzuoordnungstabelle
Rückgabewert	VcLineFormat	Neues Linienformatobjekt

Code-Beispiel

```
Set newLineFormat = VcGantt1.LineFormatCollection.Add("boxFormat1")
```

AddBySpecification**Methode von VcLineFormatCollection**

Mit dieser Methode können Sie ein Linienformat über eine Linienformat-Spezifikation erzeugen. Dies dient der Persistenz von Linienformatobjekten. Die Spezifikation eines Linienformats kann erfragt (siehe VcLinienFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Linienformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ formatSpecification	String Mögliche Werte:	Linienformatspezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcLineFormat	Neues Lineformatobjekt

Copy**Methode von VcLineFormatCollection**

Mit dieser Methode können Sie ein Linienformat kopieren. Wenn das Linienformat mit dem angegebenen Namen existiert und der Name des neuen Linienformats noch nicht verwendet wird, wird das neue Linienformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String Mögliche Werte:	Name des zu kopierenden Linienformats Name der Farbzuoordnungstabelle
⇒ newFormatName	String Mögliche Werte:	Name des neuen Linienformats

		Name der Farbzuoordnungstabelle
Rückgabewert	VcLineFormat	Linienformatobjekt

Code-Beispiel

```
Dim lineFormatCltn As VcLineFormatCollection
Dim lineFormat As VcLineFormat

Set lineFormatCltn = VcGantt1.LineFormatCollection
Set lineFormat = lineFormatCltn.Copy("CurrentLineFormat", "NewLineFormat")
```

FirstFormat**Methode von VcLineFormatCollection**

Mit dieser Methode können Sie auf das erste Linienformat der Linienformat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Linienformate zu iterieren. Existiert kein Linienformat in der Linienformat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLineFormat	erstes Linienformat

Code-Beispiel

```
Dim format As VcLineFormat

Set format = VcGantt1.LineFormatCollection.FirstFormat
```

FormatByIndex**Methode von VcLineFormatCollection**

Mit dieser Methode können Sie auf ein einzelnes Linienformat über seinen Index zugreifen. Existiert kein Linienformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Linienformats
	Mögliche Werte:	Datenfeldindex
Rückgabewert	VcLineFormat	Ermitteltes Linienformat-Objekt

Code-Beispiel

```
Dim lineFormatCltn As VcLineFormatCollection
```



```
Dim format As VcLineFormat

Set lineFormatCltn = VcGantt1.LineFormatCollection
Set format = lineFormatCltn.FormatByIndex(2)
```

FormatByName

Methode von VcLineFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Linienformat zugreifen. Existiert kein Linienformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ formatName	String Mögliche Werte:	Name des Linienformats Name der Farbzuoordnungstabelle
Rückgabewert	VcLineFormat	Linienformat

Code-Beispiel

```
Dim formatCltn As VcLineFormatCollection
Dim format As VcLineFormat

Set formatCltn = VcGantt1.LineFormatCollection
Set format = formatCltn.FormatByName("Standard")
```

NextFormat

Methode von VcLineFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Linienformate der BoxFormat-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFormat** den Initialwert erfasst haben. Sind alle Formate durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLineFormat	Nachfolgendes Linienformat

Code-Beispiel

```
Dim formatCltn As VcLineFormatCollection
Dim format As VcLineFormat

Set formatCltn = VcGantt1.LineFormatCollection
Set format = formatCltn.FirstFormat

While Not format Is Nothing
```

```

List1.AddItem format.Name
Set format = formatCltn.NextFormat
Wend

```

Remove

Methode von VcLineFormatCollection

Mit dieser Methode können Sie ein Linienformat löschen. Wenn das Linienformat noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird **False** zurückgegeben, ansonsten **True**.

	Datentyp	Beschreibung
Parameter: ⇒ FormatName	String Mögliche Werte:	Name des Linienformats Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Linienformat gelöscht (True) / nicht gelöscht (False)

Code-Beispiel

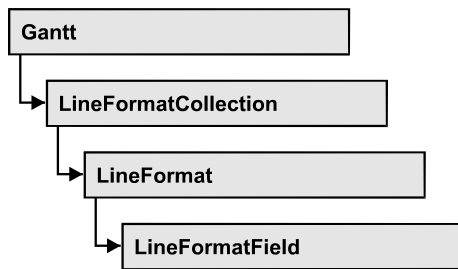
```

Dim lineFormatCltn As VcLineFormatCollection
Dim lineFormat As VcLineFormat

Set lineFormatCltn = VcGantt1.LineFormatCollection
Set lineFormat = lineFormatCltn.FormatByIndex(1)
lineFormatCltn.Remove (lineFormat.Name)

```

7.55 VcLineFormatField



Ein Objekt vom Typ **VcLineFormatField** stellt ein Linienformatfeld, also ein Feld eines VcLineFormat-Objekts dar. Ein Linienformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Linienformat untergebracht ist.

Eigenschaften

- Alignment
- ConstantText
- DateOutputFormat
- FormatName
- Index
- PatternBackgroundColorAsARGB
- PatternBackgroundColorDataFieldIndex
- PatternBackgroundColorMapName
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternEx
- PatternExDataFieldIndex
- PatternExMapName
- TextDataFieldIndex
- TextFont
- TextFontColor
- TextFontColorDataFieldIndex
- TextFontColorMapName
- TextFontDataFieldIndex
- TextFontMapName
- TextLineCount

Eigenschaften

Alignment

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Linienformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum	Ausrichtung des Feldinhalts
	Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

ConstantText

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Linienformatfeld ausgeben, falls der Typ des Linienformatfeldes auf **vcFFTText** und falls die Eigenschaft **TextDataFieldIndex** auf -1 gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text
	Mögliche Werte:	Name der Farbzusordnungstabelle

DateOutputFormat

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie das Ausgabeformat von Terminen einstellen oder erfragen. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

D:	Wochentagsname erster Buchstabe
TD:	Wochentagsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
DD:	Tagesnummer zweistellig: 01-31 (nicht anpassbar)
DDD:	die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
M:	erster Buchstabe des Monatsnamens (nicht anpassbar)
TM:	Monatsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
MM:	zweistellige Monatsnummer: 01-12
MMM:	erste drei Buchstaben des Monatsnamens (nicht anpassbar)
YY:	zweistellige Jahreszahl
YYYY:	vierstellige Jahreszahl
WW:	zweistellige Nummer der Kalenderwoche: 01-53
TW:	Text für "Kalenderwoche" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
Q:	einstellige Quartalsnummer: 1-4
TQ:	Quartalsname (kann über das Ereignis OnSupplyTextEntry angepasst werden)
hh:	Stunde zweistellig im 24-Stunden-Format: 00-23
HH:	Stunde zweistellig im 12-Stunden-Format: 01-12
Th:	Text für "Uhr" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis OnSupplyTextEntry angepasst werden)
mm:	Minute zweistellig: 00-59
ss:	Sekunde zweistellig: 00-59
TS:	kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TL:	langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
TT:	Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datum {DMYhms:;}
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcLineFormatField.DateOutputFormat = "DD.MM.YY"
```

FormatName

Nur-Lese-Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Namen des Linienformats erfragen, zu dem dieses Linienformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Linienformat-Objektes
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Index

Nur-Lese-Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Index des Linienformatfeldes im zugehörigen Linienformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Tabellenformatfeldes
	Mögliche Werte:	Datenfeldindex

PatternBackgroundColorAsARGB

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Linienformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert -1, wenn das Feld die Hintergrundfarbe des Linienformats besitzen soll.

Wenn in der Eigenschaft **BackColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Hintergrundfarbe des Tabellenformatfelds Standardwert: -1

PatternBackgroundColorDataFieldIndex

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternBackgroundColorMapName** benötigt wird. Wenn Sie hier -1 angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

PatternBackgroundColorMapName

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzusordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzusordnungs-

tabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **Pattern-BackgroundColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **BackColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

PatternColorAsARGB

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Linienformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Boxformats besitzen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Linienformatfeldes

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcGantt1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.PatternColor = RGB(0, 255, 0)
```

PatternColorDataFieldIndex

Nur-Lese-Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

PatternColorMapName

Eigenschaft von VcLineFormatField

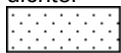
Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuoordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Kalendergitters aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColor** ausgegeben.

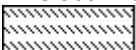
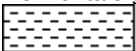
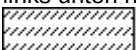
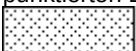
	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuoordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

PatternEx

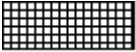
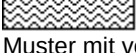
Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Linienformatfeldes ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Standardwert: Wie im Dialog definiert Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 

vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
vcCrossPattern 6	Kreuzschraffur 
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 

vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster

vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 
vcVerticalGradientPattern 62	Vertikaler Farbverlauf 
vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke 
vcZigZagPattern 2030	Horizontale Zickzack-Linien 

PatternExDataFieldIndex

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternExMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

PatternExMapName

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Namen einer Muster-Zuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Muster-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **PatternExDataFieldIndex** angegeben ist, wird das Muster aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **PatternEx** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

TextDataFieldIndex

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTText** den Index des Datenfelds, dessen Inhalt in dem Linienformatfeld dargestellt werden soll, erfragen oder setzen. Wenn Sie den Index auf **-1** setzen, wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes
	Mögliche Werte:	

	Datenfeldindex
--	----------------

TextFont

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Linienformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftart in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Tabellenformatfelds

TextFontColor

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Linienformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Tabellenformatfelds Standardwert: -1

TextFontColorDataFieldIndex

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Zuordnungstabelle für Schriftfarben von der Eigenschaft **TextFontColorMapName** benötigt wird. Wenn Sie hier -1 angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	

	Datenfeldindex
--	----------------

TextFontColorMapName

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle (Typ vcColorMap) für die Schriftfarbe setzen oder erfragen, falls der Typ des Formatfeldes auf **vcFFTTText** gesetzt wurde. Wird für den Namen "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn der Name einer Farbzuoordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **TextFontColorDataFieldIndex** angegeben ist, wird die Schriftfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Eintrag der Zuordnungstabelle zu, wird die Hintergrundfarbe verwendet, die über die Eigenschaft **TextFontColor** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schriftfarben-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

TextFontDataFieldIndex

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Schrift-Zuordnungstabelle in der Eigenschaft **TextFontMapName** benötigt wird. Wenn Sie hier -1 angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextFontMapName

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie den Namen einer Schrift-Zuordnungstabelle (Typ vcFontMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Schrift-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **TextFontDataFieldIndex** angegeben ist, wird die Schriftart aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Schriftart aus der Eigenschaft **TextFont** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schrift-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

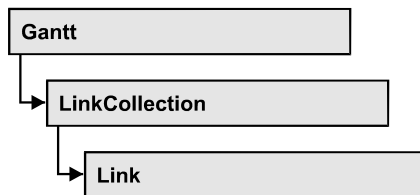
TextLineCount

Eigenschaft von VcLineFormatField

Mit dieser Eigenschaft können Sie die Zeilenanzahl erfragen oder setzen, wenn die Größe des Beschriftungsfeldes mehrere zulässt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeilenzahl
	Mögliche Werte:	Datenfeldindex

7.56 VcLink



Ein Verbindungsobjekt stellt die logische und grafische Verbindung zwischen zwei Knoten dar. Ob Verbindungen dargestellt werden, kann auf der Eigenschaftenseite **Verbindungen** im Kontrollkästchen **Verbindungen anzeigen** festgelegt werden. Auch wenn sie nicht angezeigt werden, werden sie für die Terminrechnung (Schedule) verwendet.

Eigenschaften

- AllData
- DataField
- ID
- PredecessorNode
- SuccessorNode

Methoden

- DataRecord
- DeleteLink
- RelatedDataRecord
- UpdateLink

Eigenschaften

AllData

Eigenschaft von VcLink

Mit dieser Eigenschaft können alle Daten für die Verbindung gesetzt oder erfragt werden. Beim Setzen ist ein CSV-String (Semikolon als Trennzeichen) oder ein Datenfeld erlaubt, beim Erfragen wird ein String zurückgegeben. (siehe auch **InsertLinkRecord**.)

	Datentyp	Beschreibung
Eigenschaftswert	data field/string	Alle Daten der Verbindung

Code-Beispiel

```

Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim allDataOfLink As String

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink

allDataOfLink = link.AllData

```

DataField**Eigenschaft von VcLink**

Mit dieser Eigenschaft kann ein einzelnes Datenfeld einer Verbindung gesetzt oder erfragt werden. Die Werte, die den Vorgänger- oder Nachfolgerknoten identifizieren, dürfen nicht verändert werden.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Datenfeldes Datenfeldindex
Eigenschaftswert	Variant	Inhalt des Datenfelds

Code-Beispiel

```

Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim message As String

Set linkCltn = VcGantt1.LinkCollection
For Each link in linkCltn
    message = "Delete link from " & link.DataField(1)
                & " to " & link.DataField(2) & " ?"
    If MsgBox(message, vbOKCancel, "Delete Link") = vbOK Then
        link.DeleteLink
    End If
Next link

```

ID**Nur-Lese-Eigenschaft von VcLink**

Mit dieser Eigenschaft können Sie die ID einer Verbindung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Verbindungs-ID
	Mögliche Werte:	Name der Farbzuoordnungstabelle

PredecessorNode

Nur-Lese-Eigenschaft von VcLink

Mit dieser Eigenschaft kann der Vorgängerknoten einer Verbindung identifiziert werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcNode	Vorgängerknoten

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim node As VcNode
Dim nodeName As String

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink
Set node = link.PredecessorNode
nodeName = node.DataField(1)
```

SuccessorNode

Nur-Lese-Eigenschaft von VcLink

Mit dieser Eigenschaft kann der Nachfolgerknoten einer Verbindung identifiziert werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcNode	Nachfolgerknoten

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim node As VcNode
Dim nodeName As String

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink
Set node = link.SuccessorNode
nodeName = node.DataField(1)
```

Methoden

DataRecord

Methode von VcLink

Mit dieser Eigenschaft können Sie die Verbindung als Datensatzobjekt erfragen. Über die Eigenschaften des Datensatzobjektes haben Sie auch Zugriff auf die entsprechende Datentabelle und TabellenAuflistung.

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Zurückgegebener Datensatz

DeleteLink

Methode von VcLink

Mit dieser Methode können Sie eine Verbindung löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Verbindung erfolgreich /nicht erfolgreich gelöscht

Code-Beispiel

```
Private Sub VcGantt1_OnLinkRClick(ByVal link As VcGanttLib.VcLink, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    Dim message As String

    message = "Delete Link: " & link.AllData

    If MsgBox(message, vbOKCancel, "Delete Link") = vbOK Then
        link.DeleteLink
    End If

    returnStatus = vcRetStatNoPopup
End Sub
```

RelatedDataRecord

Methode von VcLink

Mit dieser Eigenschaft können Sie einen Datensatz aus einer verknüpften Tabelle erfragen, der dem Datensatz der Verbindungsdatentabelle zugeordnet ist. Der im Parameter übergebene Index bezeichnet das Feld im Datensatz, in dem der Schlüssel des zugeordneten Datensatzes steht.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Datenfeldes, das den Schlüssel enthält Datenfeldindex
Rückgabewert	VcDataRecord	Zurückgegebener zugeordneter Datensatz

UpdateLink

Methode von VcLink

Wenn ein Datenfeld einer Verbindung mit der Eigenschaft **DataField** verändert wurde, so kann mit der Methode **UpdateLink** die Darstellung aktualisiert werden.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Verbindung erfolgreich /nicht erfolgreich aktualisiert

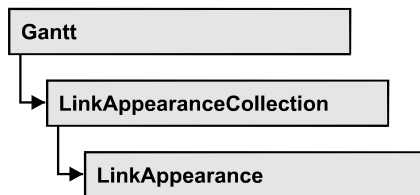
Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink

link.DataField(2) = "10"
link.UpdateLink
```

7.57 VcLinkAppearance



Ein LinkAppearance-Objekt bestimmt das Aussehen aller Verbindungen, deren Daten die Filterbedingungen erfüllen, die dem LinkAppearance-Objekt zugeordnet sind. Verschiedene LinkAppearance-Objekte können auf der Eigenschaftenseite **Verbindungen** in der Tabelle voreingestellt werden.

Eigenschaften

- FilterName
- LineColor
- LineThickness
- LineType
- Name
- PredecessorLayerName
- PrePortSymbol
- RoutingType
- Specification
- SuccessorLayerName
- SuccPortSymbol
- Visible

Methoden

- PutInOrderAfter

Eigenschaften

FilterName

Nur-Lese-Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie den Filter erfragen, der für ein bestimmtes Verbindungsaussehen verwendet wird. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Verbindungen** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Filterobjekt

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim filterOfLinkApp As String

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Blue")
filterOfLinkApp = linkAppearance.Filter
```

LineColor**Eigenschaft von VcLinkAppearance**

Mit dieser Eigenschaft können Sie die Farbe eines LinkAppearance-Objekts erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialogfeld **Linienart**, das Sie über die Eigenschaftenseite **Verbindungen** erreichen, festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Blue")

linkAppearance.LineColor = RGB(0, 0, 255)
```

LineThickness**Eigenschaft von VcLinkAppearance**

Mit dieser Eigenschaft können Sie die Linienstärke eines LinkAppearance-Objekts erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie auf der Eigenschaftenseite definiert

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Standard")

linkAppearance.LineThickness = 4
```

LineType

Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie den Linientyp eines LinkAppearance-Objekts erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialogfeld **Linienart**, das Sie über die Eigenschaftenseite **Verbindungen** erreichen, festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3	Linientyp Standardwert: vcSolid Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet

1172 API-Referenz: VcLinkAppearance

vcLineType0 100	Linientyp 0
vcLineType1 101	Linientyp 1
vcLineType10 110	Linientyp 10
vcLineType11 111	Linientyp 11
vcLineType12 112	Linientyp 12
vcLineType13 113	Linientyp 13
vcLineType14 114	Linientyp 14
vcLineType15 115	Linientyp 15
vcLineType16 116	Linientyp 16
vcLineType17 117	Linientyp 17
vcLineType18 118	Linientyp 18
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3
vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Blue")

linkAppearance.LineType = 5
```

Name

Nur-Lese-Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie den Namen der LinkAppearance erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```

Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

nameLinkApp = linkAppearance.name

```

PredecessorLayerName**Eigenschaft von VcLinkAppearance**

Mit dieser Eigenschaft können Sie vorgeben oder erfragen, von welchem Layer des jeweiligen Vorgängerknotens eine Verbindung, die mit dieser VcLinkAppearance verwendet wird, abgehen soll. Bei Eingabe von "" (Default) geht die Verbindung vom ersten sichtbaren Layer des jeweiligen Vorgängerknotens ab.

Sie können diese Eigenschaft auch im Dialogfeld **Linienart**, das Sie über die Eigenschaftenseite **Verbindungen** erreichen, festlegen

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Layer-Namen übergibt
	Mögliche Werte:	Name der Farbzuoordnungstabelle

PrePortSymbol**Eigenschaft von VcLinkAppearance**

Mit dieser Eigenschaft können Sie einer Verbindung ein Portsymbol, das die Einmündung zum Vorgängerknoten kennzeichnet, zuweisen oder erfragen.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LinkPredecessorSymbolEnum	Symbol am Vorgängerknoten Standardwert: vcLPSNone
	Mögliche Werte:	
	vcLPSArrow 64	Vorgänger-Portsymbol Pfeil 
	vcLPSDoubleArrow 65	Vorgänger-Portsymbol Doppelpfeil 
	vcLPSDoubleSemiCircle 97	Vorgänger-Portsymbol Doppelhalbkreis 
	vcLPSFilledArrow 72	Vorgänger-Portsymbol Gefüllter Pfeil 
	vcLPSFilledDoubleArrow 88	Vorgänger-Portsymbol Gefüllter Doppelpfeil 
	vcLPSFilledDoubleSemiCircle 120	Vorgänger-Portsymbol Gefüllter Doppelhalbkreis 
	vcLPSFilledSemiCircle 104	Vorgänger-Portsymbol Gefüllter Halbkreis 
	vcLPSNone 0 vcLPSSemiCircle 96	Vorgänger-Portsymbol Keines Vorgänger-Portsymbol Halbkreis 

Code-Beispiel

```

Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

linkAppearance.PrePortSymbol = vcLPSFilledDoubleSemiCircle

```

RoutingType

Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Verbindungen im Diagramm nur waagrecht und senkrecht (und damit orthogonal) verlaufen dürfen, oder ob sie auch schräg (d.h. durch Objekte hindurchschneidend) verlaufen dürfen.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LinkRoutingTypeEnum	Routentyp Standardwert: vcLRTOrthogonal
	Mögliche Werte: vcLRTNotSet -1	Ein in der Liste der LinkAppearance-Objekte weiter vorne stehender Verbindungsführungstyp wird verwendet.
	vcLRTOrthogonal 1	Verbindungen verlaufen nur waagerecht und senkrecht und zeigen eine orthogonale Form.
	vcLRTOrthogonalDistinguishable 2	Verbindungen verlaufen nur waagerecht und senkrecht und zeigen eine orthogonale Form. Durch die Darstellung von entsprechenden Schrägen bei Knicken kann man die Verbindungen besser voneinander unterscheiden sowie ihre Richtung besser verfolgen. Das Diagramm wird dabei entsprechend der Anzahl der entstehenden horizontalen Linienstücke höher.
	vcLRTStraightLined 4	Verbindungen zeigen eine gerade Form und können auch schräg verlaufen.

Specification

Nur-Lese-Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie die Spezifikation dieses Verbindungsaussehens auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines

Verbindungsaussiehens mit der Methode **VcLinkAppearanceCollection.Add-BySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Verbindungsaussiehens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

SuccessorLayerName

Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie vorgeben oder erfragen, zu welchem Layer des jeweiligen Vorgängerknotens eine Verbindung, die mit dieser VcLinkAppearance verwendet wird, führen soll. Bei Eingabe von "" (Default) führt die Verbindung zum ersten sichtbaren Layer des jeweiligenNachfolgerknotes.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die den Layer-Namen übergibt
	Mögliche Werte:	Name der Farbzuoordnungstabelle

SuccPortSymbol

Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie einer Verbindung ein Portsymbol, das die Einmündung zum Nachfolgerknoten kennzeichnet, zuweisen oder erfragen.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LinkSuccessorSymbolEnum	Symbol am Nachfolgerknoten Standardwert: vcLSSNone
	Mögliche Werte:	

vcLSSArrow 32	Nachfolger-Portsymbol Pfeil →
vcLSSDoubleArrow 33	Nachfolger-Portsymbol Doppelpfeil →→
vcLSSFilledArrow 40	Nachfolger-Portsymbol Gefüllter Pfeil →
vcLSSFilledDoubleArrow 56	Nachfolger-Portsymbol Gefüllter Doppelpfeil →→
vcLSSNone 0	Nachfolger-Portsymbol Keines

Code-Beispiel

```

Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

linkAppearance.SuccPortSymbol = vcLSSFilledDoubleArrow

```

Visible**Eigenschaft von VcLinkAppearance**

Mit der Eigenschaft Visible können Sie erfragen oder festlegen, ob die Verbindungen - nicht jedoch die Phantomlinien für Links bei Interaktionen - sichtbar sein sollen oder nicht.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Verbindungen** festgelegt werden, gilt dort jedoch auch für die Phantomlinien.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```

Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

linkAppearance.Visible = False

```

Methoden

PutInOrderAfter

Methode von VcLinkAppearance

Mit dieser Methode können Sie dieses Verbindungsausssehen in der Auflistung aller Verbindungsausssehen hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Verbindungsausssehen an die erste Stelle gesetzt. Die Reihenfolge der Verbindungsausssehen in der Auflistung entscheidet darüber, in welcher Reihenfolge sie auf die Verbindungen angewendet werden.

	Datentyp	Beschreibung
Parameter: refLinkAppearanceName	String Mögliche Werte:	Name des Verbindungsausssehens, hinter das das aktuelle Verbindungsausssehen gesetzt werden soll. Name der Farbzusordnungstabelle
Rückgabewert	Void	

Code-Beispiel

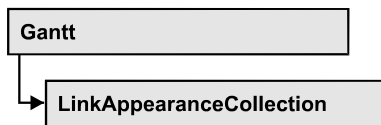
```

Dim linkAppCltn As VcLinkAppearanceCollection
Dim linkApp1 As VcLinkAppearance
Dim linkApp2 As VcLinkAppearance

linkAppCltn = VcGantt1.LinkAppearanceCollection()
linkApp1 = linkAppCltn.Add("linkApp1")
linkApp2 = linkAppCltn.Add("linkApp2")
linkApp1.PutInOrderAfter("linkApp2")
linkAppCltn.Update()

```

7.58 VcLinkAppearanceCollection



In einem Objekt vom Typ **VcLinkAppearanceCollection** sind automatisch alle definierten LinkAppearance-Objekte zusammengefasst. Über **For Each linkAppearance In LinkAppearanceCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Verbindungsaussen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **LinkAppearanceByName** und **LinkAppearanceByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Verbindungsaussen kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstLinkAppearance
- LinkAppearanceByIndex
- LinkAppearanceByName
- NextLinkAppearance
- Remove
- Update

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcLinkAppearanceCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Link-Appearance-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In**

collection angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim linkApp As VcLinkAppearance

For Each linkApp In VcGantt1.LinkAppearanceCollection
    Debug.Print linkApp.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcLinkAppearanceCollection

Mit dieser Eigenschaft kann die Anzahl der LinkAppearance-Objekte in der LinkAppearance-Auflistung abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der LinkAppearance-Objekte

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim numberOfLinkAppearances As Integer

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection

numberOfLinkAppearances = linkAppearanceCltn.Count
```

Methoden

Add

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie ein neues Verbindungsausssehen in der LinkAppearanceCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcLinkAppearance-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Bei dem neuen Verbindungsausssehen sind standardmäßig alle Eigenschaften auf transparent gesetzt.

	Datentyp	Beschreibung
Parameter: ⇒ newName	String Mögliche Werte:	Name des LinkAppearance-Objekts Name der Farbzuoordnungstabelle
Rückgabewert	VcLinkAppearance	Neues LinkAppearance-Objekt

Code-Beispiel

```
Set newLinkAppearance = VcGantt1.LinkAppearanceCollection.Add("linkapp1")
```

AddBySpecification**Methode von VcLinkAppearanceCollection**

Mit dieser Methode können Sie ein Verbindungsaussehen über eine Verbindungsaussehen-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Verbindungsaussehen-Objekten. Die Spezifikation eines Verbindungsaussehens kann erfragt (siehe VcLinkAppearance-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Verbindungsaussehen mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ linkAppearanceSpecification	String Mögliche Werte:	Verbindungsaussehen-Spezifikation Name der Farbzuoordnungstabelle
Rückgabewert	VcLinkAppearance	Neues Verbindungsaussehen-Objekt

Copy**Methode von VcLinkAppearanceCollection**

Mit dieser Methode können Sie ein Verbindungsaussehen kopieren. Wenn das Verbindungsaussehen mit dem angegebenen Namen existiert und der Name des neuen Verbindungsaussehens noch nicht verwendet wird, wird das neue Verbindungsaussehen-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ fromName	String	Name des zu kopierenden Verbindungsaussehens
	Mögliche Werte:	Name der Farbzuzuordnungstabelle
⇒ newName	String	Name des neuen Verbindungsaussehens
	Mögliche Werte:	Name der Farbzuzuordnungstabelle
Rückgabewert	VcLinkAppearance	LinkAppearance-Objekt

FirstLinkAppearance

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. das erste LinkAppearance-Objekt der LinkAppearance-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextLinkAppearance** über die nachfolgenden Objekte zu iterieren. Existiert kein LinkAppearance-Objekt in der LinkAppearance-Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLinkAppearance	Erstes LinkAppearance-Objekt

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance
```

LinkAppearanceByIndex

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie auf ein einzelnes LinkAppearance-Objekt über seinen Index zugreifen. Existiert kein LinkAppearance-Objekt unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	Integer	Index des LinkAppearance-Objektes

LinkAppearanceByName

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes LinkAppearance-Objekt zugreifen. Existiert kein LinkAppearance-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇨ linkAppearanceName	String	Name des LinkAppearance-Objektes
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	VcLinkAppearance	LinkAppearance-Objekt

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Standard")
```

NextLinkAppearance

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden LinkAppearance-Objekte der LinkAppearanceCollection zugreifen, nachdem Sie mit der Methode **FirstLinkAppearance** den Initialwert erfasst haben. Sind alle LinkAppearance-Objekte durchlaufen, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLinkAppearance	Nachfolgendes LinkAppearance-Objekt

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcGantt1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

While Not linkAppearance Is Nothing
    linkAppearance.Visible = False
    Listbox.AddItem ("Name:" & linkAppearance.name)
    Set linkAppearance = linkAppearanceCltn.NextLinkAppearance
Wend
```

Remove

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie ein LinkAppearance-Objekt löschen. Wenn das Objekt noch irgendwo verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird **False** zurückgegeben, sonst **True**.

	Datentyp	Beschreibung
Parameter: ⇒ name	String Mögliche Werte:	Name des Verbindungsaussehens Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Verbindungsaussehen gelöscht (True)/nicht gelöscht (False)

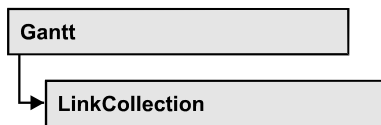
Update

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie eine Auflistung von Verbindungsaussehen aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)

7.59 VcLinkCollection



In einem Objekt vom Typ **VcLinkCollection** sind automatisch alle definierten Verbindungsobjekte zusammengefasst. Über **For Each link In LinkCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Verbindungen zugreifen. Die Anzahl der im Auflistungsobjekt vorhandenen Verbindungen kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- _NewEnum
- Count

Methoden

- FirstLink
- NextLink
- SelectLinks

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcLinkCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Verbindungsobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim link As VcLink

For Each link In VcGantt1.LinkCollection
    Debug.Print link.Name
Next
```

Count**Nur-Lese-Eigenschaft von VcLinkCollection**

Mit dieser Eigenschaft kann die Anzahl der Verbindungen in der Link-Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Verbindungen

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim numberLinks As Integer

Set linkCltn = VcGantt1.LinkCollection
numberLinks = linkCltn.Count
```

Methoden**FirstLink****Methode von VcLinkCollection**

Mit dieser Methode können Sie auf die erste Verbindung der Link-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextLink** über die nachfolgenden Verbindungen zu iterieren. Existiert keine Verbindung in der Link-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLink	Erste Verbindung

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink
```

NextLink

Methode von VcLinkCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Verbindungen der Link-Auflistung zugreifen, nachdem Sie mit der Methode **FirstLink** den Initialwert erfasst haben. Sind alle Verbindungen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLink	Nachfolgendes Verbindung

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim stringLink As String

Set linkCltn = VcGantt1.LinkCollection
Set link = linkCltn.FirstLink

While Not link Is Nothing
    stringLink = link.AllData
    Listbox.AddItem (stringLink)
    Set link = linkCltn.NextLink
Wend
```

SelectLinks

Methode von VcLinkCollection

Mit dieser Methode können Sie festlegen, welche Verbindungen in der Link-Auflistung verfügbar sein sollen.

	Datentyp	Beschreibung
Parameter: ⇒ selectionType	SelectionTypeEnum Mögliche Werte: vcAll 0 vcAllLinksCausingCycles 7 vcAllLinksInCycles 6 vcAllVisible 1 vcSelected 2	Auszuwählende Verbindungen Alle Objekte im Diagramm werden ausgewählt. Wird diese Selektion gewählt, dann befinden sich in der LinkCollection alle Verbindungen, die tatsächlich Zyklen verursachen, d.h. würde diese minimale Anzahl Verbindungen gelöscht, gäbe es keine Zyklen mehr. Wird dieser Selektionstyp gewählt, dann befinden sich in der LinkCollection alle Verbindungen, die Zyklen bilden. Zyklen sind geschlossene Ketten von Knoten und Verbindungen. Alle sichtbaren Objekte werden ausgewählt. Alle markierten Objekte werden ausgewählt.
Rückgabewert	Long	Anzahl ausgewählter Verbindungen

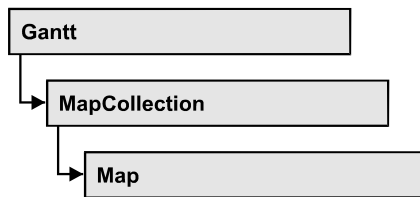
1188 API-Referenz: VcLinkCollection

Code-Beispiel

```
Dim linkCollection As VcLinkCollection

Set linkCollection = VcGantt1.LinkCollection
linkCollection.SelectGroups (vcAllMarked)
```

7.60 VcMap



Eine Zuordnungstabelle (Map) legt über Datenfeldeinträge bestimmte Eigenschaften von Knoten, beispielsweise die Hintergrundfarbe, datenfeldabhängig fest.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Über **For Each mapentry In Map** können Sie in einer Schleife auf alle Einträge der Tabelle zugreifen.

Eigenschaften

- _NewEnum
- ConsiderFilterEntries
- Count
- Name
- Specification
- Type

Methoden

- CreateEntry
- DeleteEntry
- FirstMapEntry
- GetMapEntry
- NextMapEntry

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcMap

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In**

collection angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim map As VcMap

For Each map in VcGantt1.Map
    Debug.Print.map.Name
Next
```

ConsiderFilterEntries

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob bei der Zuordnung von Datenfeldeinträgen zu einer Zuordnungstabelle Filter berücksichtigt werden, um so auch Wertebereiche als Schlüsselwerte angeben zu können.

	Datentyp	Beschreibung

Count

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft kann die Anzahl der Einträge in der Zuordnungstabelle (Map) abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Mapeinträge

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim numberOfEntries As Long

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
numberOfEntries = map.count
```

Name

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie den Namen der Zuordnungstabelle (Map) abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapName As String

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap
mapName = map.Name
```

Specification

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie die Spezifikation dieser Zuordnungstabelle auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung einer Zuordnungstabelle mit der Methode **VcMapCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation der Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

Type

Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie den Typ der Zuordnungstabelle (Map) abfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	MapTypeEnum Mögliche Werte: vcAnyMap 0 vcColorMap 1 vcFontMap 8 vcGraphicsFileMap 7 vcMillimeterMap 9 vcNumberMap 10 vcPatternMap 3 vcTextMap 6	Typ der Zuordnungstabelle Beliebig (nur zum Selektieren verwendet) Farben Schriften Grafikdatei Millimeter Zahlen Schraffuren Text

Code-Beispiel

```
Dim mapCollection As VcMapCollection
Dim map As VcMap

Set mapCollection = VcGantt1.MapCollection
mapCollection.SelectMaps (vcAnyMap)
Set map = mapCollection.MapByName("Map1")
map.Type = vcPatternMap
```

Methoden

CreateEntry

Methode von VcMap

Mit dieser Methode kann ein neuer Eintrag (= eine neue Zeile) für die Zuordnungstabelle (Map) erzeugt werden. Damit der neue Eintrag in der Zuordnungstabelle wirksam wird, sollte anschließend die Methode **MapCollection.Update()** aufgerufen werden.

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Map-Eintrag

Code-Beispiel

```
Set mapCltn = VcGantt1.MapCollection
Set map = mapCltn.Add("MapColor")

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update
```

DeleteEntry

Methode von VcMap

Mit dieser Methode kann ein Eintrag (= eine Zeile) der Zuordnungstabelle (Map) gelöscht werden. Damit die Löschung in der Zuordnungstabelle wirksam wird, sollte anschließend die Methode **MapCollection.Update()** aufgerufen werden.

	Datentyp	Beschreibung
Parameter: ⇒ mapEntry	VcMapEntry	Eintrag der Map
Rückgabewert	Boolean	Map-Eintrag erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

map.DeleteEntry mapEntry
mapCltn.Update
```

FirstMapEntry

Methode von VcMap

Mit dieser Methode können Sie auf den ersten Eintrag der Zuordnungstabelle (Map) zugreifen, um anschließend in einer Schleife mit der Methode **NextMapEntry** über die nachfolgenden Einträge zu iterieren. Existiert kein Eintrag im Map-Objekt, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Erster Map-Eintrag

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)

Set map = mapCltn.FirstMap
Set mapEntry = map.FirstMapEntry
```

GetMapEntry

Methode von VcMap

Diese Methode liefert den entsprechenden Eintrag der Zuordnungstabelle (Map) zu dem im Datenfeld angegebenen Wert.

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Eintrag der Zuordnungstabelle entsprechend Feldinhalt

NextMapEntry

Methode von VcMap

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Einträge (Zeilen) des Map-Objekts zugreifen, nachdem Sie mit der Methode **FirstMapEntry** den Initialwert erfasst haben. Sind alle Einträge durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Nachfolgender Eintrag der Zuordnungstabelle

Code-Beispiel

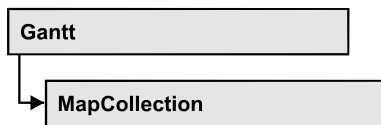
```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)

Set map = mapCltn.FirstMap
Set mapEntry = map.FirstMapEntry

While Not mapEntry Is Nothing
    List1.AddItem (mapEntry.Legend)
    Set mapEntry = map.NextMapEntry
Wend
```

7.61 VcMapCollection



In einem Objekt des Typs VcMapCollection sind die Zuordnungstabellen (Maps) zusammengefasst, die dem Auflistungsobjekt über die Methode **SelectMaps** zugewiesen wurden. Über **For Each map In MapCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Zuordnungstabellen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **MapByName** und **MapByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Zuordnungstabellen kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Zuordnungstabellen.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstMap
- MapByIndex
- MapByName
- NextMap
- Remove
- SelectMaps
- Update

Eigenschaften

NewEnum

Nur-Lese-Eigenschaft von VcMapCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim map As VcMap

For Each map In VcGantt1.MapCollection
    Debug.Print map.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcMapCollection

Mit dieser Eigenschaft kann die Anzahl der Zuordnungstabellen (Maps) in der Map-Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Maps

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim numberOfMaps As Long

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
numberOfMaps = mapCltn.Count
```

Methoden

Add

Methode von VcMapCollection

Mit dieser Methode können Sie eine neue Zuordnungstabelle in der MapCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Zuordnungstabellenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ mapName	String Mögliche Werte:	Name der Zuordnungstabelle Name der Farbzusordnungstabelle
Rückgabewert	VcMap	Neues Zuordnungstabellenobjekt

Code-Beispiel

```
Set newMap = VcGantt1.MapCollection.Add("map1")
```

AddBySpecification

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle über eine Zuordnungstabellen-Spezifikation erzeugen. Dies dient der Persistenz von Zuordnungstabellen-Objekten. Die Spezifikation einer Zuordnungstabelle kann erfragt werden (siehe VcMap-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Zuordnungstabelle mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Zuordnungstabellenspezifikation Name der Farbzusordnungstabelle
Rückgabewert	VcMap	Neues Zuordnungstabellenobjekt

Copy

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle kopieren. Wenn die Zuordnungstabelle mit dem angegebenen Namen existiert und der Name der neuen Zuordnungstabelle noch nicht verwendet wird, wird das neue Zuordnungstabellenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ mapName	String	Name der zu kopierenden Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle
⇒ newMapName	String	Name der neuen Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle
Rückgabewert	VcMap	Zuordnungstabellenobjekt

FirstMap

Methode von VcMapCollection

Mit dieser Methode können Sie auf die erste Zuordnungstabelle (Map) der Map-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextMap** über die nachfolgenden Zuordnungstabellen zu iterieren. Existiert keine Zuordnungstabelle in der Map-Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**). Zuvor muss mit der Methode **SelectMaps** die gewünschte Map-Auswahl getroffen worden sein.

	Datentyp	Beschreibung
Rückgabewert	VcMap	Erste Map

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap
```

MapByIndex

Methode von VcMapCollection

Mit dieser Methode können Sie auf eine einzelne Zuordnungstabelle über ihren Index zugreifen. Existiert keine Zuordnungstabelle unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Zuordnungstabelle Datenfeldindex
Rückgabewert	VcMap	Ermitteltes Zuordnungstabellenobjekt

MapByName

Methode von VcMapCollection

Mit dieser Methode können Sie unter Verwendung des Namens der Zuordnungstabelle (Map) auf eine bestimmte Zuordnungstabelle zugreifen. Zuvor muss mit der Methode **SelectMaps** die gewünschte Auswahl der Zuordnungstabelle getroffen worden sein. Existiert kein Map-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ mapName	String Mögliche Werte:	Name der Map Name der Farbzusordnungstabelle
Rückgabewert	VcMap	Map

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
```

```
Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.MapByName("Map_1")
```

NextMap

Methode von VcMapCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Zuordnungstabellen (Maps) der Map-Auflistung zugreifen, nachdem Sie mit der Methode **FirstMap** den Initialwert erfasst haben. Sind alle Maps durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMap	Nachfolgende Zuordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap

While Not map Is Nothing
    List1.AddItem map.Name
    Set map = mapCltn.NextMap
Wend
```

Remove

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle löschen. Wenn die Zuordnungstabelle noch in irgendeinem anderen Objekt benutzt wird, kann sie nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ mapName	String	Name der Zuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Boolean	Zuordnungstabelle gelöscht (True)/nicht gelöscht (False)

SelectMaps

Methode von VcMapCollection

Mit dieser Methode können Sie steuern, welche Typen von Zuordnungstabellen (Maps) in Ihre Map-Auflistung aufgenommen werden.

	Datentyp	Beschreibung
Parameter: ⇒ selectionType	MapTypeEnum Mögliche Werte: vcAnyMap 0 vcColorMap 1 vcFontMap 8 vcGraphicsFileMap 7 vcMillimeterMap 9 vcNumberMap 10 vcPatternMap 3 vcTextMap 6	Auszuwählender Map-Typ Beliebig (nur zum Selektieren verwendet) Farben Schriften Grafikdatei Millimeter Zahlen Schraffuren Text
Rückgabewert	Long	Anzahl der Ausgewählten Maps

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
```

Update

Methode von VcMapCollection

Mit dieser Methode können Sie die Darstellung aller Objekte, die durch die verwendeten Zuordnungstabellen (Maps) bestimmt werden, aktualisieren. Wenn Sie diese Methode nicht aufrufen, werden die Änderungen der Zuordnungstabellen (Maps) zur Laufzeit nicht ausgeführt. Sie sollten diese Methode erst am Ende des Codes zur Festlegung der Zuordnungstabellen und der Map-Auflistung verwenden, damit die Aktualisierung nicht schon ausgeführt wird, bevor alle Festlegungen der Zuordnungstabellen ausgeführt worden sind.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
```

1202 API-Referenz: VcMapCollection

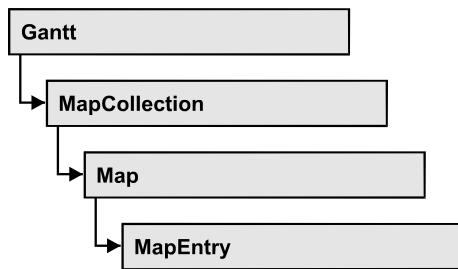
```
Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

While Not mapEntry.DataFieldValue = "A"
    Set mapEntry = map.NextMapEntry
Wend

mapEntry.Color = RGB(0, 0, 0)

mapCltn.Update
```

7.62 VcMapEntry



Ein Objekt vom Typ VcMapEntry ist ein Eintrag einer Zuordnungstabelle (Map) und damit das Element einer Zuordnungstabelle. Ein Zuordnungstabelleneintrag enthält die Kombination aus einem Datenfeldinhalt des Vorgangsdatensatzes sowie bestimmten Attributen (z. B. Farbe, Grafikdatei, Schriftattribute).

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie noch mehr Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

Eigenschaften

- ColorAsARGB
- DataFieldValue
- FontBody
- FontName
- FontSize
- GraphicsFileName
- Legend
- Millimeter
- Number
- Pattern

Eigenschaften

ColorAsARGB

Eigenschaft von VcMapEntry

für Farben-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Farbe für den Zuordnungstabelleneintrag erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen

Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim colorOfMapEntry As OLE_COLOR

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcColorMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

colorOfMapEntry = mapEntry.Color
```

DataFieldValue

Eigenschaft von VcMapEntry

Mit dieser Eigenschaft können Sie den Inhalt des Datenfeldes des Zuordnungstabelleneintrags erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Inhalt des Datenfelds
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim dataFieldValue As String

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

dataFieldValue = mapEntry.DataFieldValue
```

FontBody

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie den Schriftgrad für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	FontBodyEnum	Schriftgrad
	Mögliche Werte: vcBold 2 vcBoldItalic 4 vcItalic 3 vcRegular 1	fett fett und kursiv kursiv normal

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontBodyOfMapEntry As FontBodyEnum

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontBodyOfMapEntry = vcBold
```

FontName

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Schriftart für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Schriftart
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontNameOfMapEntry As String

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontNameOfMapEntry = "Arial"
```

FontSize

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Schriftgröße für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Schriftgröße

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontSizeOfMapEntry As Long

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontSizeOfMapEntry = 12
```

GraphicsFileName

Eigenschaft von VcMapEntry

Für Grafikdateien-Zuordnungstabellen: Mit dieser Eigenschaft können Sie den Namen der Grafikdatei des Zuordnungstabelleneintrags erfragen oder festlegen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. WMF mit eingebautem EMF)

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcGraphicsFileMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

mapEntry.GraphicsFileName = AppPath & "\picture1.bmp"
```

Legend

Eigenschaft von VcMapEntry

Mit dieser Eigenschaft können Sie den Legendentext für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendentext
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim LegendOfMapEntry As String

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

LegendOfMapEntry = "1. activity"
```

Millimeter

Eigenschaft von VcMapEntry

für Millimeter-Zuordnungstabellen: Mit dieser Eigenschaft können Sie den Millimeter-Wert des Zuordnungstabelleneintrags erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Wert in 1/100

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim MillimeterOfMapEntry As Long

Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcMillimeterMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

MillimeterOfMapEntry = 3
```

Number

Eigenschaft von VcMapEntry

für numerische Zuordnungstabellen: Mit dieser Eigenschaft können Sie einen numerische Wert in der Zuordnungstabelle erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Numerischer Wert

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim MillimeterOfMapEntry As Long

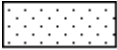
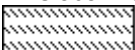
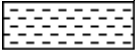
Set mapCltn = VcGantt1.MapCollection
mapCltn.SelectMaps (vcMillimeterMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

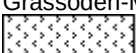
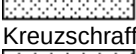
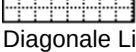
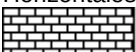
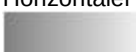
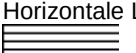
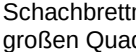
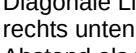
MillimeterOfMapEntry = 3
```

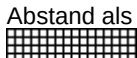
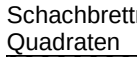
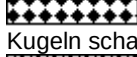
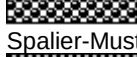
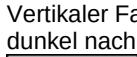
Pattern

Eigenschaft von VcMapEntry

für *Schraffuren-Zuordnungstabellen (vcPatternMap)*: Mit dieser Eigenschaft können Sie den Schraffurtyp des Zuordnungstabelleneintrags erfragen oder festlegen.

Eigenschaftswert	Datentyp	Beschreibung
	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
	vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
	vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
	vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 

vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern 

vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 
vcVerticalGradientPattern 62	Vertikaler Farbverlauf 
vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 

1212 API-Referenz: VcMapEntry

vcWideUpwardDiagonalPattern 2017

vcZigZagPattern 2030

Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke



Horizontale Zickzack-Linien



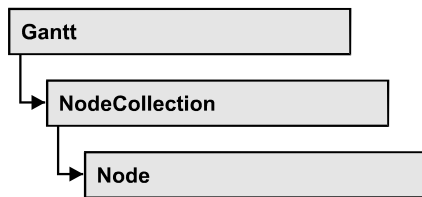
Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim pattern As FillPatternEnum

Set mapCltn = VcGantt1.mapCollection
mapCltn.SelectMaps (vcPatternMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

pattern = vcBDiagonalPattern
```

7.63 VcNode



Ein Knoten ist ein Grundelement eines Balkendiagramms. Knoten lassen sich über Verbindungen zu einer Struktur verknüpfen. Das Aussehen eines Knotens wird über die Layer bestimmt, deren Filter auf den Knoten zutreffen. Eingefügt werden Knoten über die VcGantt-Methoden **InsertNodeRecord** bzw. **Open** oder interaktiv.

Eigenschaften

- AllData
- DataField
- ID
- IncomingLinks
- MarkNode
- MoveMode
- OutgoingLinks
- SnapTargetMode
- SuperGroup
- UpdateBehaviorName

Methoden

- DataRecord
- DeleteNode
- GetPositionInView
- GetPositionInViewAsVariant
- NodeRowInView
- OutlineIndent
- OutlineOutdent
- RelatedDataRecord
- SetPositionInView
- UpdateNode

Eigenschaften

AllData

Eigenschaft von VcNode

Mit dieser Eigenschaft können alle Daten auf einmal für den Knoten gesetzt oder erfragt werden. Beim Setzen ist ein CSV-String (Semikolon als Trennzeichen) oder ein Variant erlaubt, der in einem Feld (Array) alle Datenfelder des Knotens erhält. Beim Erfragen wird ein String zurückgegeben. (Siehe auch **InsertNodeRecord**.)

	Datentyp	Beschreibung
Eigenschaftswert	String/data field	Alle Daten des Datensatzes

Code-Beispiel

```
Private Sub VcGantt1_OnNodeModify(ByVal node As VcGanttLib.VcNode, _
                                   ByVal modificationType As _
                                   VcGanttLib.ModificationTypeEnum, _
                                   returnStatus As Variant)

    Dim allDataOfNode As String

    returnStatus = vcRetStatFalse

    allDataOfNode = node.AllData
    MsgBox allDataOfNode

End Sub
```

DataField

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie einem Datenfeld des Knotens einen Wert zuweisen oder einen gesetzten Wert erfragen. Wenn ein Knoten durch diese Methode einen neuen Wert erhalten hat, muss anschließend die Methode **UpdateNode** aufgerufen werden.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Datenfeldes
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

```

Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    If MsgBox("Delete Node: " & node.dataField(0), vbYesNo, "Delete Node") = _
        vbYes Then node.DeleteNode

    returnStatus = vcRetStatNoPopup
End Sub

```

ID**Nur-Lese-Eigenschaft von VcNode**

Mit dieser Eigenschaft können Sie die ID eines Knotens erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Knoten-ID
	Mögliche Werte:	Name der Farbzuoordnungstabelle

IncomingLinks**Nur-Lese-Eigenschaft von VcNode**

Mit dieser Eigenschaft haben Sie Zugriff auf alle Verbindungen, die in einen Knoten hineinführen.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkCollection	LinkCollection-Objekt

Code-Beispiel

```

Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    Dim incomingLinks As VcLinkCollection
    Dim link As VcLink
    Dim predecessorNode As VcNode

    Set incomingLinks = node.IncomingLinks

    For Each link In incomingLinks
        Set predecessorNode = link.PredecessorNode
        predecessorNode.MarkNode = True
    Next link

    returnStatus = vcRetStatNoPopup

```

End Sub

MarkNode

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob ein Knoten markiert ist. Die gesetzte Markierung ist nur dann sichtbar, wenn auf der Eigenschaftenseite **Knoten** unter **Markierung** nicht **Ohne** ausgewählt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knoten markiert/nicht markiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode
Dim predecessor As VcNode
Dim linkCltn As VcLinkCollection
Dim link As VcLink

Set nodeCltn = VcGantt1.nodeCollection
nodeCltn.SelectNodes (vcAll)

For Each node In nodeCltn
    Set linkCltn = node.IncomingLinks
    For Each link In linkCltn
        Set predecessor = link.predecessorNode
        predecessor.MarkNode = True
    Next link
Next node
```

MoveMode

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie einstellen oder erfragen, in welche Richtung(en) ein Knoten interaktiv verschoben werden darf.

	Datentyp	Beschreibung
Eigenschaftswert	NodeMoveModeEnum	Verschiebemodus
	Mögliche Werte: vcNodeMoveModeAutomaticXOrY 4 vcNodeMoveModeNoMove 0	interaktive Verschiebung nach Beginn des Ziehens entweder nur in X- oder Y-Richtung möglich. Die Richtung des Ziehens gibt automatisch vor, ob die X- oder die Y-Richtung ausgewählt wird. keine interaktive Verschiebung möglich

vcNodeMoveModeX 1	interaktive Verschiebung in x-Richtung möglich
vcNodeMoveModeXY 3	interaktive Verschiebung in x- und y-Richtung möglich
vcNodeMoveModeY 2	interaktive Verschiebung in y-Richtung

Code-Beispiel

```

Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcGantt1.NodeCollection
nodeCltn.SelectNodes (vcAll)

For Each node In nodeCltn
    If node.DataField(2) < Now Then node.MoveMode = vcNodeMoveModeNoMove
Next node

```

OutgoingLinks**Nur-Lese-Eigenschaft von VcNode**

Mit dieser Eigenschaft haben Sie Zugriff auf alle Verbindungen, die von einem Knoten ausgehen.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkCollection	LinkCollection-Objekt

Code-Beispiel

```

Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    Dim outgoingLinks As VcLinkCollection
    Dim link As VcLink
    Dim successorNode As VcNode

    Set outgoingLinks = node.outgoingLinks

    For Each link In outgoingLinks
        Set successorNode = link.successorNode
        successorNode.MarkNode = True
    Next link

    returnStatus = vcRetStatNoPopup
End Sub

```

SnapTargetMode**Eigenschaft von VcNode**

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob dieser Knoten automatisch oder manuell als mögliches Einrstziel selektiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	NodeSnapTargetModeEnum	Selektionsmodus für den Knoten beim Verschieben mit Einrastzielen Standardwert: vcNSTMAutomatically
	Mögliche Werte: vcNSTMAutomatically 1	Knoten wird automatisch als Einrastziel ausgewählt, wenn VcGantt.SnapTargetNodesSelectionMode auf vcAutomatically gesetzt wurde. Keine Auswirkung bei vcUserSelection . Nur die zuvor ausgewählte Knotenmenge wird auf den Wert der Eigenschaft überprüft.
	vcNSTMYesOnUserSelection 2	Knoten wird als Einrastziel ausgewählt, wenn VcGantt.SnapTargetNodesSelectionMode auf vcUserSelection gesetzt wurde. Keine Auswirkung bei vcAutomatically . ALLE vorhandenen Knoten werden auf den gesetzten Wert überprüft.
	vcNSTMNo 0	Knoten wird nicht als Einrastziel ausgewählt.

SuperGroup

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie bei gruppierter Darstellung die Gruppe erfragen, der der aktuelle Knoten angehört.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroup	Gruppe, der der Knoten angehört

Code-Beispiel

```
Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As VcGanttLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    returnStatus = vcRetStatNoPopup
    Dim group As VcGroup
    Set group = node.SuperGroup
    Label1.Caption = "Group: " & group.Name

End Sub
```

UpdateBehaviorName

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzustimmungstabelle

Methoden

DataRecord

Methode von VcNode

Mit dieser Eigenschaft können Sie den Knoten als Datensatzobjekt erfragen. Über die Eigenschaften des Datensatzobjektes haben Sie auch Zugriff auf die entsprechende Datentabelle und Tabellenauflistung.

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Zurückgegebener Datensatz

DeleteNode

Methode von VcNode

Mit dieser Methode können Sie einen Knoten löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich (true) / nicht erfolgreich (false) gelöscht

Code-Beispiel

```
Private Sub VcGantt1_OnNodeRClick(ByVal node As VcGanttLib.VcNode, _
    ByVal location As _
    VcGanttLib.LocationEnum, ByVal x As Long, _
    ByVal y As Long, returnStatus As Variant)

    If MsgBox("Delete Node: " & node.DataField(0), vbYesNo, "Delete Node") = _
        vbYes Then node.DeleteNode
    returnStatus = vcRetStatNoPopup
End Sub
```


GetPositionInView

Methode von VcNode

Mit dieser Methode können Sie erfragen, an welcher Position im sichtbaren Bereich des Diagramms der Knoten liegt.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetPositionInViewAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
ViewReferencePoint	ViewReferencePointEnum	Referenzpunkt (des Diagramms)
	Mögliche Werte: vcVRPBottomCenter 28 vcVRPBottomLeft 27 vcVRPBottomRight 29 vcVRPCenterCenter 25 vcVRPCenterLeft 24 vcVRPCenterRight 26 vcVRPTopCenter 22 vcVRPTopLeft 21 vcVRPTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts
NodeReferencePoint	NodeReferencePointEnum	Knotenreferenzpunkt
	Mögliche Werte: vcNRPBottomCenter 28 vcNRPBottomLeft 27 vcNRPBottomRight 29 vcNRPCenterCenter 25 vcNRPCenterLeft 24 vcNRPCenterRight 26 vcNRPTopCenter 22 vcNRPTopLeft 21 vcNRPTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts
⇐ XOffset	Long	x-Wert des Offsets (= des Abstands des Knotenreferenzpunkts vom Referenzpunkt) in Pixeln
⇐ YOffset	Long	y-Wert des Offsets in Pixeln
Rückgabewert	Void	

GetPositionInViewAsVariant

Methode von VcNode

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetPositionInView**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference

(gekennzeichnet durch ↩) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

NodeRowInView

Methode von VcNode

Mit dieser Methode können Sie erfragen, ob die Zeile, in der sich der Knoten befindet, im sichtbaren Bereich des Diagramms ist.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Zeile befindet sich/befindet sich nicht im sichtbaren Bereich des Diagramms

Code-Beispiel

```
Dim node As VcNode

Set node = VcGantt1.GetNodeByID("15")
If Not node.NodeRowInView Then _
    Call VcGantt1.ScrollToNodeLine(node, vcVerCenterAligned)
```

OutlineIndent

Methode von VcNode

Mit dieser Methode kann man im hierarchisch gruppierten Diagramm Knoten tiefer stufen. Dabei wird der Knoten im Tabellenteil nach rechts gerückt. Er bleibt aber in seiner bisherigen Zeile stehen. Diese Methode entspricht dem Menüpunkt **Tiefer stufen** im Kontextmenü für Knoten.

Der Rückgabewert gibt an, ob die Methode erfolgreich war. Das Tieferstufen ist beispielsweise für Knoten, die bereits auf der tiefsten Ebene liegen, nicht möglich.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Methode erfolgreich (True)/ Methode nicht erfolgreich (False)

Code-Beispiel

```
Private Sub CommandOutlineIndent_Click()
    Dim anode As VcNode
    Set anode = getSelectedNode 'Function returns selected node
    MsgBox anode.OutlineIndent
    Set anode = Nothing
End Sub
```

OutlineOutdent

Methode von VcNode

Mit dieser Methode kann man im hierarchisch gruppierten Diagramm Knoten höher stufen. Dabei wird der Knoten im Tabellenteil nach links gerückt. Er bleibt aber in seiner bisherigen Zeile stehen. Diese Methode entspricht dem Menüpunkt **Höher stufen** im Kontextmenü für Knoten.

Der Rückgabewert gibt an, ob die Methode erfolgreich war. Das Höherstufen ist beispielsweise für Knoten, die bereits auf der höchsten Ebene liegen, nicht möglich.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Methode erfolgreich (True)/ Methode nicht erfolgreich (False)

Code-Beispiel

```
Private Sub CommandOutlineOutdent_Click()
    Dim anode As VcNode
    Set anode = getSelectedNode 'Function returns selected node
    MsgBox anode.OutlineOutdent
    Set anode = Nothing
End Sub
```

RelatedDataRecord

Methode von VcNode

Mit dieser Eigenschaft können Sie einen Datensatz aus einer verknüpften Tabelle erfragen, der dem Datensatz der Knotentabellenebene zugeordnet ist. Der im Parameter übergebene Index bezeichnet das Feld im Datensatz, in dem der Schlüssel des zugeordneten Datensatzes steht.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Datenfeldes, das den Schlüssel enthält Datenfeldindex
Rückgabewert	VcDataRecord	Zurückgegebener zugeordneter Datensatz

SetPositionInView

Methode von VcNode

Mit dieser Methode können Sie festlegen, dass so gescrollt werden soll, dass der Knoten an einer bestimmten Position im sichtbaren Bereich des Diagramms liegt. Diese Position können Sie angeben, indem Sie den Abstandsvektor (x,y) zwischen einem bestimmten Knotenreferenzpunkt und einem bestimmten Referenzpunkt des Diagramms angeben.

	Datentyp	Beschreibung
Parameter:		
ViewReferencePoint	ViewReferencePointEnum	Referenzpunkt (des Diagramms)
	Mögliche Werte: vcVRPBottomCenter 28 vcVRPBottomLeft 27 vcVRPBottomRight 29 vcVRPCenterCenter 25 vcVRPCenterLeft 24 vcVRPCenterRight 26 vcVRPTopCenter 22 vcVRPTopLeft 21 vcVRPTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts
NodeReferencePoint	NodeReferencePointEnum	Knotenreferenzpunkt
	Mögliche Werte: vcNRPBottomCenter 28 vcNRPBottomLeft 27 vcNRPBottomRight 29 vcNRPCenterCenter 25 vcNRPCenterLeft 24 vcNRPCenterRight 26 vcNRPTopCenter 22 vcNRPTopLeft 21 vcNRPTopRight 23	unten mittig unten links unten rechts mittig mittig mittig links mittig rechts oben mittig oben links oben rechts
⇨ XOffset	Long	X-Wert des Offsets (= des Abstands des Knotenreferenzpunkts vom Referenzpunkt) in Pixeln
⇨ YOffset	Long	Y-Wert des Offsets in Pixeln
Rückgabewert	Void	

Code-Beispiel

```
' scroll the diagram so that the vector between the bottom right corner of the
node and the bottom right corner of the diagram is (-10, -10)
```

```
mySelNode.SetPositionInView vcVRPBottomRight, vcNRPBottomRight, -10, -10
```

UpdateNode

Methode von VcNode

Nachdem Sie ein oder mehrere Datenfelder eines Knotens mit der Eigenschaft **DataField** verändert haben, aktualisieren Sie die Grafik mit **UpdateNode**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich (true) / nicht erfolgreich (false) aktualisiert

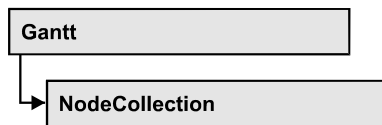
Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcGantt1.NodeCollection
Set node = nodeCltn.FirstNode

node.DataField(12) = "Group A"
node.UpdateNode
```

7.64 VcNodeCollection



Ein Objekt vom Typ `VcNodeCollection` beinhaltet alle im Diagramm vorhandenen Knoten. Mit der Methode **SelectNodes** können Sie eine Untermenge dieser Knoten selektieren. Über **For Each node In NodeCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Knoten zugreifen. Die Anzahl der im Auflistungsobjekt vorhandenen Knoten kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `FirstNode`
- `NextNode`
- `SelectNodes`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcNodeCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim node As VcNode

For Each node In VcGantt1.NodeCollection
    Debug.Print node.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcNodeCollection

Mit dieser Eigenschaft können Sie die Anzahl der Knoten in der KnotenAuflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Knoten im NodeCollection-Objekt

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection

Set nodeCltn = VcGantt1.NodeCollection
MsgBox "Number of nodes: " & nodeCltn.Count
```

Methoden

FirstNode

Methode von VcNodeCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Knoten der Knoten-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextNode** über die nachfolgenden Knoten zu iterieren. Existiert kein Knoten in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNode	Erster Knoten

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcGantt1.NodeCollection
Set node = nodeCltn.FirstNode
```

NextNode

Methode von VcNodeCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Knoten der Node-Auflistung zugreifen, nachdem Sie mit der Methode **FirstNode** den Initialwert erfasst haben. Sind alle Knoten durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNode	Nachfolgender Knoten

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcGantt1.NodeCollection
Set node = nodeCltn.FirstNode

While Not node Is Nothing
    node.MarkNode = False
    Set node = nodeCltn.NextNode
Wend
```

SelectNodes

Methode von VcNodeCollection

Mit dieser Methode können Sie steuern, welche Knoten in die Knotenaufliistung aufgenommen werden.

	Datentyp	Beschreibung
Parameter: ⇒ selType	SelectionTypeEnum Mögliche Werte: vcAll 0 vcAllLinksCausingCycles 7 vcAllLinksInCycles 6 vcAllVisible 1 vcSelected 2	auszuwählende Knoten Alle Objekte im Diagramm werden ausgewählt. Wird diese Selektion gewählt, dann befinden sich in der LinkCollection alle Verbindungen, die tatsächlich Zyklen verursachen, d.h. würde diese minimale Anzahl Verbindungen gelöscht, gäbe es keine Zyklen mehr. Wird dieser Selektionstyp gewählt, dann befinden sich in der LinkCollection alle Verbindungen, die Zyklen bilden. Zyklen sind geschlossene Ketten von Knoten und Verbindungen. Alle sichtbaren Objekte werden ausgewählt. Alle markierten Objekte werden ausgewählt.
Rückgabewert	Long	Anzahl ausgewählter Knoten

Code-Beispiel

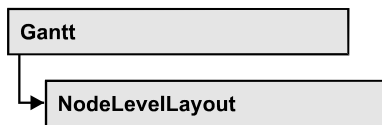
```
Dim nodeCltn As VcNodeCollection
```


1228 API-Referenz: VcNodeCollection

```
Dim node As VcNode
```

```
Set nodeCltn = VcGantt1.NodeCollection  
nodeCltn.SelectNodes vcSelected
```

7.65 VcNodeLevelLayout



Ein Objekt vom Typ VcNodeLevelLayout legt die Art der Sortierung von Knoten sowie das Erscheinungsbild der Knotenzeilen fest:

Eigenschaften

- CalendarGridName
- DateLineName
- RowBackColorAsARGB
- RowBackColorDataFieldIndex
- RowBackColorMapName
- RowPattern
- RowPatternColorAsARGB
- RowPatternColorDataFieldIndex
- RowPatternColorMapName
- RowPatternDataFieldIndex
- RowPatternMapName
- SeparationLineColor
- SeparationLineInterval
- SeparationLineThickness
- SeparationLineType
- ShowCalendarGrids
- ShowDateLines
- ShowSeparationLines
- ShowSeparationLinesAtTop
- SortDataFieldIndex
- SortOrder

Eigenschaften

CalendarGridName

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Namen des Kalendergitters für die Vorgänge setzen oder erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Knoten** eingestellt werden.

	Datentyp	Beschreibung

DateLineName

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Namen der Stichtalinie für dieses Knotenebenenlayout erfragen. Diese Eigenschaft kann auch im Dialog **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Stichtaglinie
	Mögliche Werte:	Name der Farbzuoordnungstabelle

RowBackColorAsARGB

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie die Hintergrundfarbe der Knotenzeile für diese Gruppierungsebene setzen oder erfragen. Die Standard-Farbe ist weiß.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

RowBackColorDataFieldIndex

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowBackColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowBackColorMapName

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuhordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **RowBackColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **RowBackColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	Name der Farbzuhordnungstabelle

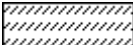
RowPattern

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie für den Hintergrund der Knotenzeilen dieser Gruppierungsebene ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte:	

1232 API-Referenz: VcNodeLevelLayout

vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
vcCrossPattern 6	Kreuzschraffur 
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 

vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschildel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster

1234 API-Referenz: VcNodeLevellLayout

vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

RowPatternColorAsARGB

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie die Musterfarbe der Knotenzeilen dieser Gruppierungsebene festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

S. auch **set/getRowBackColorAsARGB**.

Wenn in der Eigenschaft **RowPatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

RowPatternColorDataFieldIndex

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowPatternColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowPatternColorMapName

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuoordnungstabelle und ein Datenfeldindex in der Eigenschaft **RowPatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Gruppentitelzeile aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **RowPatternColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuoordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

RowPatternDataFieldIndex

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **RowPatternMapName** benötigt wird. Wenn Sie hier -1 angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

RowPatternMapName

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Namen einer Musterzuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn der Name einer Farbzuoordnungstabelle und ein Datenfeldindex in der Eigenschaft **RowPatternDataFieldIndex** angegeben sind, wird das Muster des Layers

aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **RowPattern** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

SeparationLineColor

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie die Farbe der Trennlinien für die Gruppierungsebenen festlegen.

Sie können diese Eigenschaft auch im Dialog **Gruppierung**, Bereich **Knoten**, Feld **Trennlinie** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	Farbwert ({0...255},{0...255},{0...255})

SeparationLineInterval

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie setzen oder erfragen nach wie vielen Vorgängen eine Trennlinie gezogen werden soll.

	Datentyp	Beschreibung

SeparationLineThickness

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie die Linienstärke einer Trennlinie zwischen Vorgängen erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Liniestärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Liniestärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Liniestärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm

SeparationLineType

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie die Linienart einer Trennlinie zwischen Vorgängen erfragen oder festlegen.

Sie können diese Eigenschaft auch im Dialog **Gruppierung**, Bereich **Knoten**, Feld **Trennlinie** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineType0 100	Typ der Trennlinien der Hierarchieebenen Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0

vcLineType1 101	Linientyp 1 -----
vcLineType10 110	Linientyp 10 -----
vcLineType11 111	Linientyp 11 -----
vcLineType12 112	Linientyp 12 -----
vcLineType13 113	Linientyp 13 -----
vcLineType14 114	Linientyp 14 -----
vcLineType15 115	Linientyp 15 -----
vcLineType16 116	Linientyp 16 -----
vcLineType17 117	Linientyp 17 -----
vcLineType18 118	Linientyp 18 -----
vcLineType2 102	Linientyp 2 -----
vcLineType3 103	Linientyp 3 -----
vcLineType4 104	Linientyp 4 -----
vcLineType5 105	Linientyp 5 -----
vcLineType6 106	Linientyp 6 -----
vcLineType7 107	Linientyp 7 -----
vcLineType8 108	Linientyp 8 -----
vcLineType9 109	Linientyp 9 -----
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp durchgezogen

ShowCalendarGrids

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob Kalendergitter dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Arbeitsfreie Zeiten werden/werden nicht hervorgehoben
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowDateLines

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Stichtaglinien angezeigt werden sollen. Sie können diese Eigenschaft auch im Dialog **Gruppierung** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Stichtaglinien werden/werden nicht angezeigt Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ShowSeparationLines

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob Trennlinien zwischen den Vorgängen dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Knoten** festgelegt werden.

	Datentyp	Beschreibung

ShowSeparationLinesAtTop

Nur-Lese-Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob Trennlinien zwischen Vorgängen oberhalb der Vorgänge dargestellt werden sollen oder nicht. Diese Eigenschaft kann auch im Dialog **Gruppierung**, Bereich **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	Trennlinien oben werden angezeigt/nicht angezeigt Gruppe unsichtbar/sichtbar

Gruppen sichtbar/unsichtbar

SortDataFieldIndex

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, nach dem Knoten in diesem VcGroupLevelLayout-Objekt sortiert werden sollen.

	Datentyp	Beschreibung
Parameter: ⇒ sortlevel	Integer Mögliche Werte:	Sortierebene Datenfeldindex
Eigenschaftswert	Long	Feld, nach dem sortiert wird

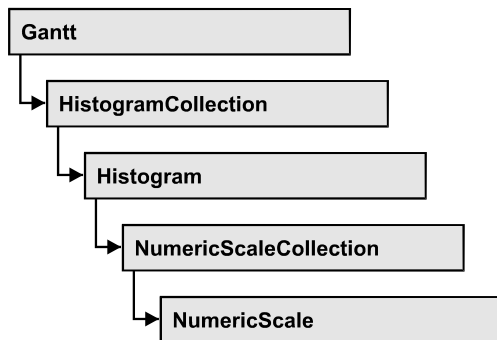
SortOrder

Eigenschaft von VcNodeLevelLayout

Mit dieser Eigenschaft legen Sie fest, ob die Vorgänge auf- oder absteigend sortiert werden. Das Feld, nach dem die Vorgänge sortiert werden, legen Sie mit der Eigenschaft **SortDataFieldIndex** fest. Diese Eigenschaft können Sie auch im Dialog **Gruppierung** festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ sortLevel	Integer Mögliche Werte:	Sortierebene Datenfeldindex
Eigenschaftswert	SortOrderEnum Mögliche Werte: vcAscending 1 vcDescending 2	aufsteigende oder absteigende Sortierreihenfolge aufsteigende Abfolge absteigende Abfolge

7.66 VcNumericScale



Ein Objekt vom Typ VcNumericScale stellt die Skala der vertikalen Achse von Histogrammen dar.

Eigenschaften

- DoubleOutputFormat
- Font
- FontColor
- Histogram
- LineColor
- MajorTicks
- MajorTicksEx
- MinorTicks
- MinorTicksEx
- Name
- PatternBackgroundColorAsARGB
- PatternColorAsARGB
- PatternEx
- ThreeDEffect
- TickColor
- Title
- Unit
- UnitEx
- UnitLabel
- UnitWidth
- UpdateBehaviorName

Eigenschaften

DoubleOutputFormat

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie das Ausgabeformat von Zahlen als double-Wert in der numerischen Skala einstellen oder erfragen. Das Format kann über folgende Zeichen dargestellt werden:

- Text
- I
- D

sowie die Trennzeichen Komma und Punkt. Dabei steht **Text** für einen beliebigen Text, **I** für die Ziffern vor dem Dezimaltrenner und **D** für je eine Ziffer hinter dem Dezimaltrenner. Die erlaubte, generelle Reihenfolge ist **Text I D Text**, wobei Komma und Punkt an beliebiger Stelle eingefügt werden können. Als Beispiel sei die Zahl -284901,3458 gegeben. Über das Format **I,DDDD ppm** wird sie als **-284901,3458 ppm** dargestellt. Über das Format **\$I,III.DD** wird sie als **\$-284,901.35** dargestellt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenfolge, die das Double-Format beschreibt, z.B. "I,DDDD ppm".
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.DoubleOutputFormat = "I,DDDD ppm"
```

Font

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie alle Schriftattribute der numerischen Skala erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute der numerischen Skala

Code-Beispiel

```
Dim font As StdFont
font = VcGantt1.NumericScaleCollection.Active.Font
```

FontColor

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Schriftfarbe der numerischen Skala erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte Standardwert: RGB (0,0,0)

Code-Beispiel

```
Dim histogram As VcHistogram
Dim numericScale As VcNumericScale

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set numericScale = histogram.NumericScaleCollection.Active

numericScale.FontColor = RGB(10, 10, 180)
```

Histogram

Nur-Lese-Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen, zu welchem Histogramm eine numerische Skala gehört.

	Datentyp	Beschreibung
Eigenschaftswert	VcHistogram	Histogrammobjekt

Code-Beispiel

```
Private Sub VcGantt1_OnNumericScaleLClick(ByVal numericScale As
VcGanttLib.VcNumericScale, ByVal x As Long, ByVal y As Long, returnStatus As
Variant)

    MsgBox "Clicked on numeric scale of the histogram " &
numericScale.histogram.Name

End Sub
```

LineColor

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Farbe aller Randlinien der numerischen Skalen von Histogrammen festlegen oder erfragen.

Gesetzt wird die Randlinienfarbe für **alle** numerischen Skalen, zurückgegeben wird die Linienfarbe der **ersten numerischen Skala**.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255}

MajorTicks

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Einheiten eine Hauptmarkierung erscheint. Eine Hauptmarkierung ist eine beschriftete Markierung auf der numerischen Achse. Die Anzahl der Einheiten können Sie auch im Dialog **Histogramm bearbeiten** festlegen. S. auch **set/getMinorTicks** und **set/getMajorTicksEx**

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Einheiten zwischen zwei Hauptmarkierungen {1...32767}
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.NumericScaleCollection
Set numericScale = numericScaleCltn.Active
numericScale.MajorTicks = 4
```

MajorTicksEx

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Einheiten eine Hauptmarkierung erscheint. Im Vergleich zur Eigenschaft

MajorTicks können hier auch Fließkomma-Zahlen eingesetzt werden. Die Anzahl der Einheiten können Sie auch im Dialog **Histogramm bearbeiten** festlegen.

S. auch **set/getMinorTicks**.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Anzahl der Einheiten zwischen zwei Hauptmarkierungen

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.NumericScaleCollection
Set numericScale = numericScaleCltn.Active
numericScale.MajorTicksEx = 100 000
```

MinorTicks

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Einheiten eine Nebenmarkierung (ein kleinerer, unbeschrifteter Teilstrich) erscheint. Die Anzahl der Einheiten können Sie im Dialog **Histogramm bearbeiten** festlegen. S. auch **set/getMinorTicksEx** und **set/getMajorTicks**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Einheiten zwischen zwei Nebenmarkierungen
		{1...32767}
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.NumericScaleCollection
Set numericScale = numericScaleCltn.Active
numericScale.MinorTicks = 2
```

MinorTicksEx

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Einheiten eine Nebenmarkierung (ein kleinerer, unbeschrifteter Teilstrich) erscheint. Im Vergleich zur Eigenschaft **set/getMinorTicks** können hier Fließkomma-Zahlen gesetzt werden. Die Anzahl der Einheiten können Sie im Dialog **Histogramm bearbeiten** festlegen. S. auch **set/getMajorTicks**

	Datentyp	Beschreibung
Eigenschaftswert	Double	Anzahl der Einheiten zwischen zwei Nebenmarkierungen {32 767...999 999}

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.NumericScaleCollection
Set numericScale = numericScaleCltn.Active
numericScale.MinorTicks = 2
```

Name

Nur-Lese-Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie den Namen einer numerischen Skala eines Histogramms erfragen. Sie können diesen Namen im Dialog **Histogramm bearbeiten** auswählen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der numerischen Skala
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim histogram As VcHistogram
Dim numericScale As VcNumericScale

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set numericScale = histogram.NumericScaleCollection.Active

MsgBox "Active num. Scale: " & numericScale.Name
```

PatternBackgroundColorAsARGB

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Hintergrundfarbe der numerischen Skala erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Hintergrundfarbe des Boxformatfelds Standardwert: -1

PatternColorAsARGB

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Musterfarbe der numerischen Skala erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

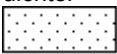
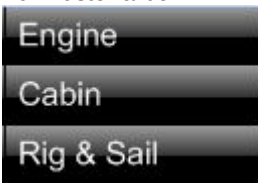
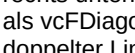
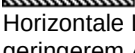
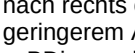
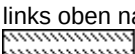
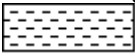
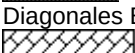
	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Linienformatfeldes

PatternEx

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie für den Hintergrund der numerischen Skala ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum Mögliche Werte:	Mustertyp

vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
vcCrossPattern 6	Kreuzschraffur 
vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 

1250 API-Referenz: VcNumericScale

vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster

vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

ThreeDEffect

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die numerische Skala mit 3D-Effekt dargestellt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	3D-Effekt eingeschaltet (True)/ausgeschaltet (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim histogram As VcHistogram
Dim numericScale As VcNumericScale

Set histogram = VcGantt1.HistogramCollection.HistogramByName("HISTOGRAM_1")
Set numericScale = histogram.NumericScaleCollection.Active

numericScale.ThreeDEffect = True
```

TickColor

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Markierungsfarbe aller numerischer Skalenstreifen von Histogrammen festlegen oder erfragen.

Gesetzt wird die Randlinienfarbe für **alle** numerischen Skalenstreifen, zurückgegeben wird die Linienfarbe des **ersten numerischen Skalaenstreifens**>.

	Datentyp	Beschreibung
Eigenschaftswert	Color RGB ({0...255},{0...255},{0...255})	RGB-Farbwerte Standardwert: 0,0,0

Title

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie einen Titel für die numerische Skala erfragen oder festlegen. Der Skalenstreifen, der den Titel zeigt, muss vom Typ **textuell** sein. Skalen und Skalenstreifen werden über den Dialog

Histogramm bearbeiten angelegt, der von der Eigenschaftenseite **Layout** aus aufrufbar ist.

	Datentyp	Beschreibung
Parameter: ⇒ Position	NumericAnnotationPositionEnum Mögliche Werte: vc10PercentFromTop 4 vc30PercentFromTop 3 vc50PercentFromTop 2 vc70PercentFromTop 1 vc90PercentFromTop 0	Position des Titels der numerischen Skala 10% der Skalenlänge vom oberen Rand entfernt 30% der Skalenlänge vom oberen Rand entfernt 50% der Skalenlänge vom oberen Rand entfernt 70% der Skalenlänge vom oberen Rand entfernt 90% der Skalenlänge vom oberen Rand entfernt
Eigenschaftswert	String Mögliche Werte:	Titel der numerischen Skala Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim activeNumericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection._
                        HistogramByName("HISTOGRAM_1"). _
                        NumericScaleCollection
Set activeNumericScale = numericScaleCltn.Active

activeNumericScale.Title(vc90PercentFromTop) = "10 % occupied"
activeNumericScale.Title(vc70PercentFromTop) = "30 % occupied"
```

Unit

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Einheiten der numerischen Skala erfragen oder festlegen. Die Einheit ist eine ganze Zahl. S. auch **set/getUnit-Width**. Diese Eigenschaft kann auch im Dialog **Histogramm bearbeiten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Einheit Datenfeldindex

UnitEx

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Grundeinheit der numerischen Skala als double-Wert einstellen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Zahlenwert im Double-Format {-999 999 .. 999 999}

Code-Beispiel

```
Private Sub CommandReduceRibbonUnit_Click()
    Dim numCol As VcNumericScaleCollection
    Dim numScale As VcNumericScale
    Set numCol =
VcGantt1.HistogramCollection.FirstHistogram.NumericScaleCollection
    Set numScale = numCol.FirstNumericScale
    numScale.UnitEx = numScale.UnitEx / 2#
    TextScaleUnit.Text = numScale.UnitEx
End Sub
```

UnitLabel

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Beschriftung der Einheiten der numerischen Skala erfragen oder festlegen. Diese Beschriftung erscheint mittig am oberen Rand der numerischen Skala.

	Datentyp	Beschreibung
Eigenschaftswert	String	Bezeichnung der Einheit
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim activeNumericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection. _
    HistogramByName("HISTOGRAM_1"). _
    NumericScaleCollection

Set activeNumericScale = numericScaleCltn.Active
activeNumericScale.UnitLabel = "Hours"
```

UnitWidth

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie die Breite der auf der numerischen Skala verwendeten Einheiten erfragen oder festlegen (in 1/100 mm). Siehe auch **set/getUnit**. Diese Eigenschaft kann auch im Dialog **Histogramm bearbeiten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Breite der Einheiten (in 1/100 mm)

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim activeNumericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection._
                        HistogramByName("HISTOGRAM_1")._
                        NumericScaleCollection

Set activeNumericScale = numericScaleCltn.Active
activeNumericScale.UnitWidth = 200
```

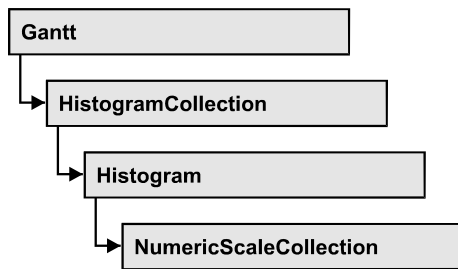
UpdateBehaviorName

Eigenschaft von VcNumericScale

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

7.67 VcNumericScaleCollection



Ein Objekt des Typs VcNumericScaleCollection beinhaltet alle bestehenden numerischen Skalen. Über **For Each numericScale In NumericScaleCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle numerischen Skalen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **NumericScaleByName** und **NumericScaleByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Skalen kann über die Eigenschaft **Count** erfragt werden. Die aktuelle numerische Skala kann durch die Eigenschaft **Active** gesetzt oder erfragt werden.

Eigenschaften

- _NewEnum
- Active
- Count

Methoden

- FirstNumericScale
- NextNumericScale
- NumericScaleByIndex
- NumericScaleByName

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNumericScaleCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Skalenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim numscale As VcNumericScale

For Each numscale In VcGantt1.NumericScaleCollection
    Debug.Print numscale.Name
Next
```

Active

Eigenschaft von VcNumericScaleCollection

Mit dieser Eigenschaft kann die aktuell für die Darstellung verwendete numerische Skala erfragt oder gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcNumericScale	Aktuell verwendete numerische Skala

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim activeNumericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection._
                        HistogramByName("HISTOGRAM_1")._
                        NumericScaleCollection
Set activeNumericScale = numericScaleCltn.Active
```

Count

Eigenschaft von VcNumericScaleCollection

Mit dieser Eigenschaft kann die Anzahl der numerischen Skalen in der NumericScale-Auflistung abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der numerischen Skalen

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numberOfNumericScales As Long

Set numericScaleCltn = VcGantt1.HistogramCollection._
                        HistogramByName("HISTOGRAM_1")._
```

NumericScaleCollection

```
numberOfNumericScales = numericScaleCltn.Count
```

Methoden

FirstNumericScale

Methode von VcNumericScaleCollection

Mit dieser Methode können Sie auf die erste numerische Skala der NumericScale-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextNumericScale** über die nachfolgenden numerischen Skalen zu iterieren. Existiert keine numerische Skala in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNumericScale	erste numerische Skala

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection._
    HistogramByName("HISTOGRAM_1"). _
    NumericScaleCollection
Set numericScale = numericScaleCltn.FirstNumericScale
```

NextNumericScale

Methode von VcNumericScaleCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden numerischen Skalen der NumericScale-Auflistung zugreifen, nachdem Sie mit der Methode **FirstNumericScale** den Initialwert erfasst haben. Sind alle numerischen Skalen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNumericScale	Nachfolgende numerische Skala

Code-Beispiel

```
Dim numericScaleCltn As VcNumericScaleCollection
Dim numericScale As VcNumericScale

Set numericScaleCltn = VcGantt1.HistogramCollection. _
```

```

HistogramByName("HISTOGRAM_1"). _
NumericScaleCollection
Set numericScale = numericScaleCltn.FirstNumericScale
While Not numericScale Is Nothing
    List1.AddItem numericScale.Name
    Set numericScale = numericScaleCltn.NextNumericScale
Wend

```

NumericScaleByIndex

Methode von VcNumericScaleCollection

Mit dieser Methode können Sie auf eine einzelne numerische Skalen über ihren Index zugreifen. Existiert keine numerische Skala unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der numerischen Skala Datenfeldindex
Rückgabewert	VcNumericScale	Ermitteltes Skalenobjekt

NumericScaleByName

Methode von VcNumericScaleCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte numerische Skala zugreifen. Existiert keine numerische Skala unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ numericScaleName	String Mögliche Werte:	Name der numerischen Skala Name der Farbzusordnungstabelle
Rückgabewert	VcNumericScale	Numerische Skala

Code-Beispiel

```

Dim numericScaleCltn As VcNumericScaleCollection

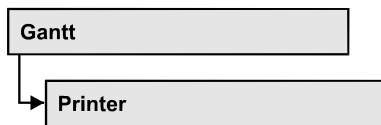
Set numericScaleCltn = VcGantt1.HistogramCollection._
HistogramByName("HISTOGRAM_1"). _

```


1260 API-Referenz: VcNumericScaleCollection

```
numericScaleCltn.Active = numericScaleCltn._  
    NumericScaleCollection.  
    NumericScaleByName("STEP1")
```

7.68 VcPrinter



Das VcPrinter-Objekt stellt Ihnen Eigenschaften zur Verfügung, die die Seitengestaltung und den Druckvorgang betreffen. Sie können die Randbreiten der Seiten einstellen sowie Seitenrahmen, Seitenzahlen, Seitenbeschriftung, Schnittmarkierungen und Druckdatum setzen. Weiterhin können Sie die Anzahl der Seiten, auf die das Diagramm verteilt werden soll, sowie Vergrößerungsfaktor, Druckausrichtung, Hoch- bzw. Querformat, Papierformat und Farbmodus festlegen.

Eigenschaften

- AbsoluteBottomMarginInCM
- AbsoluteBottomMarginInInches
- AbsoluteLeftMarginInCM
- AbsoluteLeftMarginInInches
- AbsoluteRightMarginInCM
- AbsoluteRightMarginInInches
- AbsoluteTopMarginInCM
- AbsoluteTopMarginInInches
- Alignment
- AllBorderBoxesShownOnCombinedControls
- CombiningControlsEnabled
- CurrentHorizontalPagesCount
- CurrentVerticalPagesCount
- CurrentZoomFactor
- CuttingMarks
- DateFormat
- DefaultPrinterName
- DiagramEnabled
- DocumentName
- FoldingMarksType
- MarginsShownInInches
- MaxHorizontalPagesCount
- MaxVerticalPagesCount
- Orientation
- PageDescription

- PageDescriptionString
- PageFrame
- PageNumberMode
- PageNumbers
- PagePaddingEnabled
- PaperSize
- PrintDate
- PrinterName
- ReOptimizeNodesInGroupsEnabled
- RepeatTableTimeScale
- ScalingMode
- StartUpSinglePage
- TableColumnRanges
- TableWidthAdoptionFromViewOnScreen
- TimeColumnEndDate
- TimeColumnStartDate
- TimeScaleAdjustment
- ZoomFactorAsDouble

Eigenschaften

AbsoluteBottomMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des unteren Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des unteren Seitenrandes in cm

Code-Beispiel

```
VcGantt1.Printer.AbsoluteBottomMarginInCM = 1.5
```

AbsoluteBottomMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des unteren Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des unteren Seitenrandes in Zoll

Code-Beispiel

```
VcGantt1.Printer.AbsoluteBottomMarginInches = 0.5
```

AbsoluteLeftMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des linken Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des linken Seitenrandes in cm

Code-Beispiel

```
VcGantt1.Printer.AbsoluteLeftMarginInCM = 1.5
```

AbsoluteLeftMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des linken Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des linken Seitenrandes in Zoll

Code-Beispiel

```
VcGantt1.Printer.AbsoluteLeftMarginInInches = 0.5
```

AbsoluteRightMarginInCM**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie die absolute Breite des rechten Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des rechten Seitenrandes in cm

Code-Beispiel

```
VcGantt1.Printer.AbsoluteRightMarginInCM = 1.5
```

AbsoluteRightMarginInInches**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie die absolute Breite des rechten Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des rechten Seitenrandes in Zoll

Code-Beispiel

```
VcGantt1.Printer.AbsoluteRightMarginInInches = 0.5
```

AbsoluteTopMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des oberen Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des oberen Seitenrandes in cm

Code-Beispiel

```
VcGantt1.Printer.AbsoluteTopMarginInCM = 1.5
```

AbsoluteTopMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des oberen Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des oberen Seitenrandes in Zoll

Code-Beispiel

```
VcGantt1.Printer.AbsoluteTopMarginInInches = 0.5
```

Alignment

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die Ausrichtung des Ausdrucks auf einer Seite setzen oder erfragen. Sie hat nur dann eine Auswirkung, wenn die Gesamtgrafik auf einer einzigen Seite dargestellt wird, oder wenn die Eigenschaft **RepeatTableTimeScale** eingeschaltet ist. In allen anderen Fällen wird die Gesamtgrafik zentriert ausgerichtet.

	Datentyp	Beschreibung
Eigenschaftswert	PrinterAlignmentEnum	Ausrichtung des Ausdrucks auf der Seite Standardwert: vcPCenterCenter
	Mögliche Werte:	
	vcPBottomCenter 28	vertikale Ausrichtung: unten, horizontale Ausrichtung: mittig
	vcPBottomLeft 27	vertikale Ausrichtung: unten, horizontale Ausrichtung: links
	vcPBottomRight 29	vertikale Ausrichtung: unten, horizontale Ausrichtung: rechts
	vcPCenterCenter 25	vertikale Ausrichtung: mittig, horizontale Ausrichtung: mittig
	vcPCenterLeft 24	vertikale Ausrichtung: mittig, horizontale Ausrichtung: links
	vcPCenterRight 26	vertikale Ausrichtung: mittig, horizontale Ausrichtung: rechts
	vcPTopCenter 22	vertikale Ausrichtung: oben, horizontale Ausrichtung: mittig
	vcPTopLeft 21	vertikale Ausrichtung: oben, horizontale Ausrichtung: links
	vcPTopRight 23	vertikale Ausrichtung: oben, horizontale Ausrichtung: rechts

Code-Beispiel

```
VcGantt1.Printer.Alignment = vcPTopLeft
```

AllBorderBoxesShownOnCombinedControls**Eigenschaft von VcPrinter**

Wenn diese Eigenschaft auf "True" gesetzt wird, werden alle Borderboxen auch beim kombinierten Druck mehrerer XGantt-Diagramme mit ausgedruckt. Ist "False" eingestellt, bleiben die definierten Borderboxen unberücksichtigt. Siehe auch die Objekte **VcBorderArea** und **VcBorderBox**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Borderboxen werden (True)/werden nicht (False) beim kombinierten Druck berücksichtigt
	Mögliche Werte:	
		Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.AllBorderBoxesShownOnCombinedControls = True
```

CombiningControlsEnabled

Eigenschaft von VcPrinter

Wenn diese Eigenschaft auf **True** gesetzt ist, werden alle XGantt-Steuererelemente von einer Form beim Exportieren sowie beim Drucken und in der Druckvorschau nach ihrer relativen vertikalen Position untereinander angeordnet. Dadurch können mehrere Diagramme auf einmal ausgegeben werden.

Hinweis: Beim kombinierten Druck werden die Eigenschaften **RepeatTableTimeScale** und **TimeScaleAdjustment** nicht berücksichtigt und ihr Wert als "False" angenommen. Auch die Eigenschaft **VcPrinter.FoldingMarksType** wird nicht berücksichtigt und ihr Wert als "vcFMTNone" angenommen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	XGantt-Steuererelemente einer Form werden (True) / werden nicht (False) untereinander angeordnet Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.CombiningControlsEnabled = True
```

CurrentHorizontalPagesCount

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die tatsächliche Anzahl der Seiten des Ausdrucks in der Breite ermitteln. Siehe auch die Eigenschaften **CurrentVerticalPagesCount** und **MaxHorizontalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Anzahl Seiten in horizontaler Richtung

CurrentVerticalPagesCount

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die tatsächliche Anzahl der Seiten des Ausdrucks in der Höhe ermitteln. Siehe auch die Eigenschaften **CurrentHorizontalPagesCount** und **MaxVerticalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Anzahl Seiten in vertikaler Richtung

CurrentZoomFactor

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Zoomfaktor in Prozent für den Skalierungsmodus **vcFitToPageCounterfragen** (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Tatsächlicher Zoomfaktor

CuttingMarks

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Schnittmarkierungen auf eine Seite gedruckt werden sollen (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Schnittmarken werden (True) / werden nicht (False) gedruckt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.CuttingMarks = True
```

DateFormat

Eigenschaft von VcPrinter

Mit dieser Eigenschaft kann das Datumsformat bestimmt werden, das in den DatePicker-Dialogelementen des Dialogs **Seite einrichten** verwendet werden soll. Der leere String steht dabei für das Standard-Datumsformat TS. Folgende Kürzel stehen zur Verfügung:

D:	Wochentagsname erster Buchstabe
TD:	Wochentagsname (kann über das Ereignis VcTextEntrySupplying angepasst werden)
DD:	Tagesnummer zweistellig: 01-31 (nicht anpassbar)
DDD:	die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
M:	erster Buchstabe des Monatsnamens (nicht anpassbar)
TM:	Monatsname (kann über das Ereignis VcTextEntrySupplying angepasst werden)
MM:	zweistellige Monatsnummer: 01-12
MMM:	erste drei Buchstaben des Monatsnamens (nicht anpassbar)
YY:	zweistellige Jahreszahl
YYYY:	vierstellige Jahreszahl
WW:	zweistellige Nummer der Kalenderwoche: 01-53
TW:	Text für "Kalenderwoche" (kann über das Ereignis VcTextEntrySupplying angepasst werden)
Q:	einstellige Quartalsnummer: 1-4
TQ:	Quartalsname (kann über das Ereignis VcTextEntrySupplying angepasst werden)
hh:	Stunde zweistellig im 24-Stunden-Format: 00-23
HH:	Stunde zweistellig im 12-Stunden-Format: 01-12
Th:	Text für "o' clock" (kann über das Ereignis VcTextEntrySupplying angepasst werden)
TH:	"am" oder "pm" (kann über das Ereignis VcTextEntrySupplying angepasst werden)
mm:	Minute zweistellig: 00-59
ss:	Sekunde zweistellig: 00-59

- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat für Seite einrichten Dialog
	Mögliche Werte:	Name der Farbzordnungstabelle

DefaultPrinterName

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Namen des aktuellen Standard-Systemdruckers erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des aktuellen Standard-Systemdruckers
	Mögliche Werte:	Name der Farbzordnungstabelle

DiagramEnabled

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen, ob das Diagramm (Zeitskala und Balken) mit ausgedruckt werden soll oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Diagramm wird (True) / wird nicht (False) gedruckt Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.DiagramEnabled = True
```

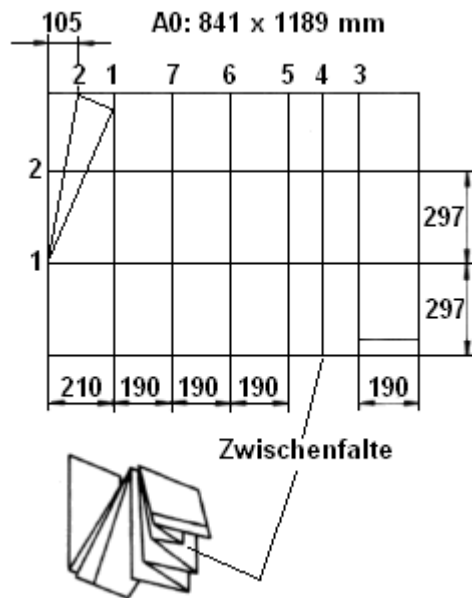
DocumentName**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie den Namen des Dokumentes bestimmen oder auslesen. Der Dokumentenname wird beim Drucker in der Liste der zu druckenden Dokumente angezeigt und hat bei speziellen Druckertreibern wie z. B. einigen, die PDF-Dateien erzeugen, besondere Funktionen.

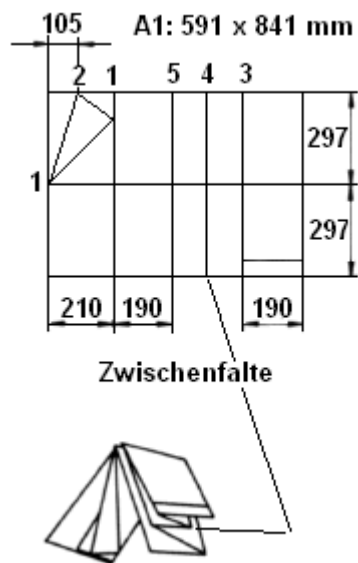
	Datentyp	Beschreibung
Eigenschaftswert	String	Dokumentenname Standardwert: " "
	Mögliche Werte:	Name der Farbzusordnungstabelle

FoldingMarksType**Nur-Lese-Eigenschaft von VcPrinter**

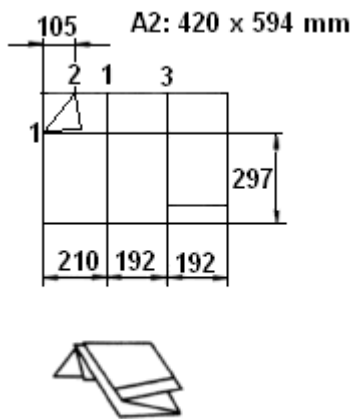
Mit dieser Eigenschaft können Sie folgende Faltmarkierungen nach DIN 824 für den Ausdruck festlegen oder erfragen. Diese ermöglichen das standardisierte Falten für DIN-A-Blattgrößen:



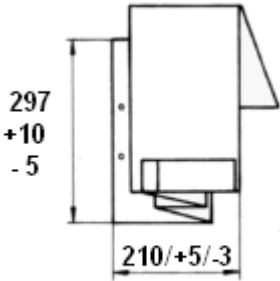
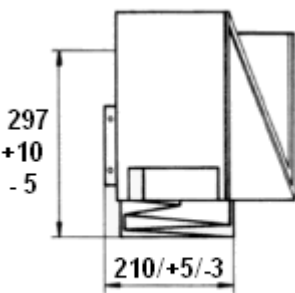
Faltung des DIN-A-0 Formats

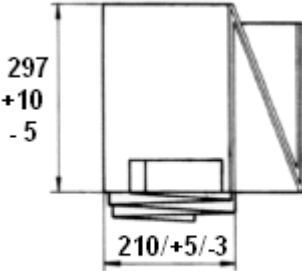


Faltung des DIN-A-1 Formats



Faltung des DIN-A-2 Formats

Eigenschaftswert	Datentyp	Beschreibung
	FoldingMarksTypeEnum	Faltmarkierungen Standardwert: vcFMTNone
	Mögliche Werte: vcFMTDIN824FormA 65	Ausgabe von Faltmarkierungen nach DIN824-A: Die gefaltete Zeichnung kann gelocht und ohne Heftstreifen abgeheftet werden. 
	vcFMTDIN824FormB 66	Faltung nach DIN 824-A Ausgabe von Faltmarkierungen nach DIN824-B: Die gefaltete Zeichnung kann gelocht und mit Heftstreifen abgeheftet werden.  Faltung nach DIN 824-B

vcFMTDIN824FormC 67	Ausgabe von Faltmarkierungen nach DIN824-C: Die gefaltete Zeichnung wird nicht gelocht, sondern in eine Sichthülle gelegt.
	
vcFMTNone 0	Keine Ausgabe von Faltmarkierungen

MarginsShownInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob im Dialog **Seite einrichten** die Maßeinheit für Seitenränder in Zoll ein- und ausgegeben wird (gegenwärtig nur zur Laufzeit möglich).

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Maßeinheit der Seitenränder im Dialog Seite einrichten in Zoll (True)/ in cm (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

MaxHorizontalPagesCount

Eigenschaft von VcPrinter

Diese Eigenschaft dient der Festlegung oder Erfragung der horizontalen Seitenzahl beim Drucken und für die Druckvorschau. Die Festlegung ist nur wirksam, wenn Sie in der Eigenschaft **ScalingMode** den Wert **vcFitToPageCount** oder **vcZoomWithHorizontalFit** festgelegt haben. S.

auch Eigenschaft **MaxVerticalPagesCount** und **CurrentHorizontalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Maximale Anzahl Seiten in horizontaler Richtung Standardwert: 1

Code-Beispiel

```
VcGantt1.Printer.MaxHorizontalPagesCount = 4
```

MaxVerticalPagesCount

Eigenschaft von VcPrinter

Diese Eigenschaft dient der Festlegung oder Erfragung der vertikalen Seitenzahl beim Drucken und für die Druckvorschau. Diese Festlegung ist nur wirksam, wenn Sie in der Eigenschaft **ScalingMode** den Wert **vcFitToPageCount** festgelegt haben. S. auch Eigenschaft **MaxHorizontalPagesCount** und **CurrentVerticalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Maximale Anzahl Seiten in vertikaler Richtung Standardwert: 1

Code-Beispiel

```
VcGantt1.Printer.MaxVerticalPagesCount = 4
```

Orientation

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen, ob die einzelnen Seiten des Ausdrucks im Hoch- oder Querformat verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	OrientationEnum	Ausrichtung Standardwert: VcPortrait
	Mögliche Werte: vcLandscape 42 vcPortrait 41	Querformat Hochformat

Code-Beispiel

```
VcGantt1.Printer.Orientation = vcLandScape
```


PageDescription

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Seitenbeschriftung für die linke untere Ecke jeder Seite erscheinen soll (True) oder nicht (False). Den Inhalt der Seitenbeschriftung legen Sie über die Eigenschaft **PageDescriptionString** fest.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Seitenbeschriftung wird (True) / wird nicht (False) gedruckt Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.PageDescription = True
```

PageDescriptionString

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie eine Seitenbeschriftung für die linke untere Ecke jeder Seite festlegen oder erfragen. Die Ausgabe der Seitenbeschriftung können Sie mit der Eigenschaft **PageDescription** steuern. Für die Seitennummerierung können Sie folgende Platzhalter angeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{PAGE} = fortlaufende Seitennummer

{NUMPAGES} = Gesamtanzahl der Seiten

{ROW} = Zeilenposition des Ausschnitts im Gesamtdiagramm

{COLUMN} = Spaltenposition des Ausschnitts im Gesamtdiagramm

	Datentyp	Beschreibung
Eigenschaftswert	String	Seitenbeschriftung Standardwert: Leere Zeichenkette ""
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.Printer.PageDescriptionString = "VARCHART chart"
```

PageFrame**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob um den Ausdruck ein Rahmen gezogen werden soll (True) oder nicht (False). Wenn die Eigenschaft **RepeatTableTimeScale** eingeschaltet ist, so wird der Rahmen um jede Einzelseite gezogen, anderenfalls wird er um die gesamte Grafik gezogen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Seitenrahmen wird dargestellt (True) / wurde nicht (False) dargestellt Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.PageFrame = True
```

PageNumberMode**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, wie die Seitennummerierung ausgegeben werden soll: "Seite N von M Seiten" oder "x.y" (Zeilennummer/Spaltennummer).

	Datentyp	Beschreibung
Eigenschaftswert	pageNumberModeEnum	Art der Seitennummerierung Standardwert: vcPRowColumn
	Mögliche Werte: vcPageNOFM 1597 vcPRowColumn 1596	"Seite N von M Seiten" "x.y" (Zeilennummer.Spaltennummer)

Code-Beispiel

```
Dim printer As VcPrinter

Set printer = VcGantt1.printer

With printer
    .Orientation = vcLandscape
    .PageNumberMode = vcPageNOFM
End With
```

```

        .PageNumbers = True
        .FitToPage = False
    End With

VcGantt1.PrintPreview

```

PageNumbers

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Seitenzahl in der linken unteren Ecke einer Seite erscheinen soll (True) oder nicht (False). Die Art der Nummerierung können Sie mit Hilfe der Eigenschaft **PageNumberMode** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Seitenzahlen werden (True) / werden nicht (False) ausgegeben Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.PageNumbers = True
```

PagePaddingEnabled

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob zwischen dem Diagramm und den Boxen für Titel und Legende so viel Platz gelassen wird, dass die Boxen auf jeder Druckseite immer in voller Breite gedruckt werden können und fest am Blattrand positioniert sind. Ist die Eigenschaft auf **False** gesetzt, werden die Boxen ohne Zwischenraum am Diagramm gedruckt und können dann je nach Diagramm auf den verschiedenen Druckseiten in der Breite variieren.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zwischenraum zwischen Diagramm und Boxen für Legende/Titel wird (True) / wird nicht (False) ausgegeben Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.PagePaddingEnabled = True
```

PaperSize**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie die zu verwendende Papiergröße festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	PaperSizeEnum	Papiergröße
	Mögliche Werte:	
	vcDIN_A2 66	DIN A2
	vcDIN_A3 8	DIN A3
	vcDIN_A4 9	DIN A4
	vcISO_C 24	ISO C
	vcISO_D 25	ISO D
	vcISO_E 26	ISO E
	vcUS_LEGAL 5	US LEGAL
	vcUS_LETTER 1	US LETTER

Code-Beispiel

```
VcGantt1.Printer.PaperSize = vcDIN_A3
```

PrintDate**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Druckdatum in der linken unteren Ecke jeder Seite erscheinen soll (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Druckdatum wird/wird nicht ausgegeben
	Mögliche Werte:	
		Gruppe unsichtbar/sichtbar
		Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.PrintDate = True
```

PrinterName

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Namen des aktuell ausgewählten Druckers auslesen oder setzen. Dies kann zum Speichern und Wiederherstellen des Zustands des Printer-Objekts verwendet werden.

Wenn man beim Setzen der Eigenschaft einen leeren String übergibt, wird der im System eingestellte Standarddrucker benutzt.

Hinweis: Bitte beachten Sie, dass bei Netzwerkdruckern der Druckername in UNC-Notation angegeben werden muss, bspw. "\\server01\printer5".

	Datentyp	Beschreibung
Eigenschaftswert	String	Druckername
	Mögliche Werte:	Name der Farbzuoordnungstabelle

ReOptimizeNodesInGroupsEnabled

Eigenschaft von VcPrinter

Wenn die Eigenschaft **TimeScaleAdjustment** aktiviert ist, können hierüber optimiert dargestellte Gruppen für die Druckvorschau bzw. den Druck automatisch neu optimiert werden. Dies ist nur notwendig, wenn einzelne Layer einen außenliegenden Text aufweisen. Die automatische Optimierung ist äusserst zeitaufwändig und kann zu stark verlängerten Reaktionszeiten in der Druckvorschau führen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Bei gleichzeitig aktivierter Eigenschaft TimeScaleAdjustment : Optimierte Gruppen werden (True)/werden nicht (False) automatisch für Druck oder Druckvorschau neu optimiert
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.ReOptimizeNodesInGroupsEnabled = True
```

RepeatTableTimeScale

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Titel, Legende, Tabelle und Zeitskala auf jeder Seite erscheinen sollen (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Titel, Legende, Tabelle und Zeitskala werden auf jeder Seite wiederholt (True)/ Titel, Legende, Tabelle und Zeitskala werden nur einmal ausgegeben und ggf. beim Seitenumbruch durchtrennt (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.RepeatTableTimeScale = True
```

ScalingMode

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Skalierungsmodus beim Drucken setzen oder erfragen. Wenn der Skalierungsmodus auf **vcZoomFactor** gesetzt ist, bestimmt der Wert in der Eigenschaft **ZoomFactor** die Größe des Ausdrucks. Wenn er auf **vcFitToPageCount** gesetzt ist, sind die Werte in **MaxHorizontalPagesCount** und **MaxVerticalPagesCount** ausschlaggebend. Bei **vcZoomWithHorizontalFit** sind dies **ZoomFactor** und **MaxHorizontalPagesCount**, das heißt ein fester Zoomfaktor mit einer festen Anzahl an Druckseiten in der Breite, die im Druck durch Stauchen oder Strecken der Zeitskala eingehalten wird. Bei den ersten beiden Skalierungsmodi kann das vollständige Füllen der Druckseiten über die Eigenschaft **AdjustTimeScale** erreicht werden.

	Datentyp	Beschreibung
Eigenschaftswert	ScalingModeEnum	Skalierungsmodus für den Ausdruck
	Mögliche Werte:	
	vcFitToPageCount 1	Skalierungsmodus "Anpassung an Seitenzahl"
	vcZoomFactor 0	Skalierungsmodus "Zoomfaktor".
	vcZoomWithHorizontalFit 2	Skalierungsmodus "Kombiniertes Anpassen"

StartUpSinglePage

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, wie die Seitenansicht beim Aufruf aussehen soll: als Gesamtansicht über alle Blätter des Diagramms oder als Darstellung der ersten Seite.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	beim Aufruf der Seitenansicht: Darstellung der ersten Seite (True)/ Gesamtansicht über alle Blätter des Diagramms (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim printer As VcPrinter

Set printer = VcGantt1.printer

With printer
    .Orientation = vcLandscape
    .StartUpSinglePage = True
    .FitToPage = False
End With

VcGantt1.PrintPreview
```

TableColumnRanges

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie Spalten der Tabelle auswählen, die im Druck angezeigt werden sollen. Dazu können Sie wie z.B. bei Microsoft Word die anzuzeigenden Spalten einzeln oder in Bereichen durch Komma oder Semikolon getrennt angeben. Beispiel: "1;5-7;3" selektiert die Spalten 1, 3 und 5 bis 7. Die Angabe "0", ein einfaches Komma oder ein Semikolon führen dazu, dass keine Spalten angezeigt werden. Der Standardwert -1 bewirkt, dass alle Spalten ausgedruckt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Anzahl der zu druckenden Tabellenspalten Standardwert: leere Zeichenkette
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
VcGantt1.Printer.TableColumnRange = "1;5-7;3"
```

TableWidthAdoptionFromViewOnScreen**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob bei der Druckvorschau und beim Druck die aktuell auf dem Bildschirm sichtbare Breite der Tabelle übernommen wird.

Diese Eigenschaft kann auch zur Laufzeit im Dialog "Seite einrichten" eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	die aktuell auf dem Bildschirm sichtbare Breite der Tabelle wird (True) / wird nicht (False) für Druckvorschau und Druck übernommen Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.Printer.TableWidthAdoptionFromViewOnScreen = True
```

TimeColumnEndDate**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft kann der Endtermin des zu druckenden Zeitraums festgelegt werden. Dabei ist es nur möglich, den Zeitraum gegenüber dem am Bildschirm gezeigten einzuschränken, d.h. nur ein früheres Enddatum als das über die Eigenschaft **VcGantt.TimeScaleEnd** gesetzte führt zu einer Änderung des Ausdrucks.

Diese Eigenschaft kann auch zur Laufzeit im Dialog **Seite einrichten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	DateTime	Enddatum des Zeitraumes für den Ausdruck

TimeColumnStartDate

Eigenschaft von VcPrinter

Mit dieser Eigenschaft kann der Starttermin des zu druckenden Zeitraums festgelegt oder erfragt werden. Dabei ist es nur möglich, den Zeitraum gegenüber dem am Bildschirm gezeigten einzuschränken, d.h. nur ein späteres Startdatum als das über die Eigenschaft **VcGantt.TimeScaleStart** gesetzte, führt zu einer Änderung des Ausdrucks.

Diese Eigenschaft kann auch zur Laufzeit im Dialog **Seite einrichten** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	DateTime	Startdatum des Zeitraumes für den Ausdruck

TimeScaleAdjustment

Eigenschaft von VcPrinter

Mit dieser Eigenschaft kann der Platz auf den Druckseiten besser ausgenutzt werden:

- Bei einer Skalierung über eine feste Anzahl Seiten: Der Zoomfaktor wird so berechnet, dass die eingestellte Anzahl Seiten in der Höhe voll bedruckt werden. Gleichzeitig wird die Zeitskala so gestaucht oder gestreckt, dass die eingestellte Anzahl Seiten in der Breite voll ausgenutzt wird.
- Bei einer Skalierung über einen Zoomfaktor: Die Zeitskala wird so gestaucht oder gestreckt, dass die eingestellte Anzahl Seiten in der Breite voll ausgenutzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	TimeScaleAdjustment	Zeitskala wird angepasst Standardwert: False

Code-Beispiel

```
VcGantt1.Printer.TimeScaleAdjustment = True
```

ZoomFactorAsDouble

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Zoomfaktor in Prozent für den Skalierungsmodus **vcZoomFactor** bzw. **vcZoomWithHorizontalFit** setzen oder erfragen (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Zoomfaktor für das Diagramm Standardwert: 100

Code-Beispiel

```
VcGantt1.Printer.ZoomFactorAsDouble = 150
```

7.69 VcRect

Rect

Ein Objekt vom Typ **VcRect** bezeichnet ein Rechteck-Objekt und wird nur im Ereignis VcGantt.OnShowInPlaceEditor übergeben.

Eigenschaften

- Bottom
- Height
- Left
- Right
- Top
- Width

Eigenschaften

Bottom

Eigenschaft von VcRect

Diese Eigenschaft gibt die untere Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des unteren Rands des Rechtecks

Height

Nur-Lese-Eigenschaft von VcRect

Diese Eigenschaft gibt die Höhe des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe des Rechtecks

Left

Eigenschaft von VcRect

Diese Eigenschaft gibt die linke Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des linken Rands des Rechtecks

Code-Beispiel

```
Private Sub VcGantt1_OnShowInPlaceEditor(ByVal editObject As Object, _
    ByVal editObjectType As _
    VcGanttLib.VcObjectTypeEnum, _
    ByVal fieldIndex As Long, ByVal objRectComplete As _
    VcGanttLib.VcRect, ByVal objRectVisible As _
    VcGanttLib.VcRect, ByVal fldRectComplete As _
    VcGanttLib.VcRect, ByVal fldRectVisible As _
    VcGanttLib.VcRect, returnStatus As Variant)

    Dim oldScaleMode As Long

    If editObjectType = vcObjTypeNodeInTable Then
        returnStatus = vcRetStatFalse

        Set myEditObject = editObject
        myEditObjectType = editObjectType
        myEditObjectFieldIndex = fieldIndex

        oldScaleMode = Me.ScaleMode
        Me.ScaleMode = vbPixels

        Select Case fieldIndex
            Case 1 'Name
                Text1.Left = fldRectVisible.Left + VcGantt1.Left
                Text1.Top = fldRectVisible.Top + VcGantt1.Top
                Text1.Width = fldRectVisible.Width
                Text1.Height = fldRectVisible.Height

                Text1.Text = editObject.DataField(fieldIndex)
                Text1.Visible = True
                Text1.SetFocus

            Case 2, 3 'Start or End
                MonthView1.Left = fldRectVisible.Left + VcGantt1.Left
                MonthView1.Top = fldRectVisible.Top + VcGantt1.Top

                MonthView1.Value = editObject.DataField(fieldIndex)
                MonthView1.Visible = True
                MonthView1.SetFocus

            Case 13 'Employee
                Combo1.Left = fldRectVisible.Left + VcGantt1.Left
                Combo1.Top = fldRectVisible.Top + VcGantt1.Top
                Combo1.Width = fldRectVisible.Width

                Combo1.Text = editObject.DataField(fieldIndex)
                Combo1.Visible = True
                Combo1.SetFocus

        End Select

        Me.ScaleMode = oldScaleMode
    End Sub
```

```
End If
```

```
End Sub
```

Right

Eigenschaft von VcRect

Diese Eigenschaft gibt die rechte Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des rechten Rands des Rechtecks

Top

Eigenschaft von VcRect

Diese Eigenschaft gibt die obere Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des oberen Rands des Rechtecks

Code-Beispiel

```
MonthView1.Top = fldRectVisible.Top + VcGantt1.Top
```

Width

Nur-Lese-Eigenschaft von VcRect

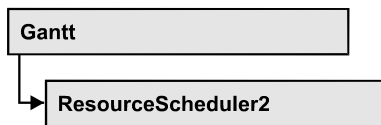
Diese Eigenschaft gibt die Breite des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Breite des Rechtecks

Code-Beispiel

```
Text1.Width = fldRectVisible.Width
```

7.70 VcResourceScheduler2



Der neue ResourceScheduler2 stellt eine wesentliche Erweiterung des ResourceScheduler1 aus Version 3.1 dar. Es werden die für eine Ressourcenplanung benötigten Objektarten nun in eigenen Datentabellen erwartet, was erst mit VARCHART XGantt 4.0 möglich wird. Im ResourceScheduler1 waren dagegen Aufgaben, Operationen, Zuweisungen und Ressourcen implizit in der Maindata-Tabelle zu definieren.

Zur Verfügung stehen die folgenden Objektarten, die jeweils in einer eigenen Datentabelle erwartet werden (bei Ressourcen können es bis zu 25 sein):

- **Aufgaben** (Tasks): Diese Objekte bilden eine Gruppierung von Operationen (s.u.) und halten elementare Eigenschaften wie das Freigabe- und das Fälligkeitsdatum sowie Priorität und Aufgabenmenge.
- **Operationen**: Diese Objekte sind über Zuweisungen (siehe unten) auf Ressourcen (siehe unten) einplanbar und erhalten durch die Ressourcenplanung als Ergebnis Start- und Enddatum des Ausführungszeitraums. Operationen besitzen eine definierte Reihenfolge in der zugehörigen Aufgabe und können als bereits begonnen gekennzeichnet werden. Außerdem ist es möglich, mehrere einander ausschließende Operationsfolgen (sogenannte Routen) als Alternativen vorzugeben. Alle Operationen der von der Ressourcenplanung gewählten Route werden gemeinsam eingeplant.
- **Ressourcen**: Diese Objekte besitzen als Hauptmerkmal eine Kapazitätskurve und nach der Ausführung einer Ressourcenplanung auch eine Belegungskurve. Außerdem können Ressourcen für eine auf ihr eingeplante Operation zeitdauerbestimmend sein, wovon augenblicklich einer Operation immer eine zugewiesen sein muss, um die Operation einplanen zu können. Daneben können einer Operation beliebig viele weitere Arbeits- und Material-Ressourcen zugewiesen sein. Weiteres Wesensmerkmal einer zeitdauerbestimmenden Ressource ist die Fähigkeit zur ein- oder mehrstufigen Gruppierung, wobei eine solche Ressource auch mehreren Gruppen gleichzeitig angehören kann.
- **Zuweisungen** (Assignments): Diese Objekte sind die eigentliche Verknüpfung zwischen Operationen und Ressourcen, wobei hier ein Übersetzungsfaktor für die Aufgabenmenge angegeben werden kann. Im Falle von zeitdauerbestimmenden Ressourcengruppen werden

Zuweisungen von der Ressourcenplanung entsprechend als solche gekennzeichnet und konkrete Ersatzzuweisungen zu der ausgewählten Einzelressource erzeugt, so dass die Einplanung nachvollziehbar und im XGantt darstellbar wird.

- **Verbindungen (Links):** Diese Objekte beschreiben die Reihenfolge von Aufgaben, das heißt, vorangehende Aufgaben müssen beendet sein, bevor nachfolgende Aufgaben begonnen werden dürfen.

Eigenschaften

- AssignmentDataTableName
- AssignmentIsResultFieldIndex
- AssignmentIsVisibleFieldIndex
- AssignmentLoadOrConsumptionPerItemFieldIndex
- AssignmentMaximumLoadFieldIndex
- AssignmentMinimumLoadFieldIndex
- AssignmentMinimumMaximumLoadType
- AssignmentOperationIDFieldIndex
- AssignmentResourceIDFieldIndex
- AssignmentResourceSelectionStrategyFieldIndex
- BaseCalendarUsageForSupplementTimes
- BaseTimeUnit
- BaseTimeUnitsPerStep
- DataRecordEventsEnabled
- DefaultOperationMaximumInterruptionTime
- DefaultResourceCalendarName
- FullUsageOfPlanningUnitsEnabled
- LinkDataTableName
- LinkDurationFieldIndex
- LinkPredecessorOperationIDFieldIndex
- LinkPredecessorTaskIDFieldIndex
- LinkSuccessorOperationIDFieldIndex
- LinkSuccessorTaskIDFieldIndex
- OperationDataTableName
- OperationLoadPerItemFieldIndex
- OperationMaximumInterruptionTimeFieldIndex
- OperationMinimumSupplementTimeFieldIndex
- OperationOverlapQuantityFieldIndex
- OperationPostLoadFieldIndex
- OperationPostOffsetFieldIndex

- OperationPreparationLoadFieldIndex
- OperationPreparationOffsetFieldIndex
- OperationResultEndDateFieldIndex
- OperationResultPostEndDateFieldIndex
- OperationResultPreparationStartDateFieldIndex
- OperationResultProcessingTimeFieldIndex
- OperationResultSelectedTimingResourceIDFieldIndex
- OperationResultStartDateFieldIndex
- OperationResultStatusFieldIndex
- OperationRouteFieldIndex
- OperationSequenceNumberFieldIndex
- OperationStartLockDateFieldIndex
- OperationTaskIDFieldIndex
- OperationWorkInProgressFieldIndex
- PlanningEndDate
- PlanningStartDate
- PlanningStrategy
- ResourceCalendarNameFieldIndex
- ResourceCapacityType
- ResourceCapacityTypeFieldIndex
- ResourceConstraintTypeFieldIndex
- ResourceDataTableName
- ResourceEfficiencyFieldIndex
- ResourceGroupDataTableName
- ResourceGroupIDFieldIndex
- ResourceNameFieldIndex
- ResourceResultLoadCurveNamePrefix
- ResourceResultStockCurveNamePrefix
- ResourceSelectionStrategy
- ResourceType
- ResultProcessingStepCount
- TaskDataTableName
- TaskDueDateFieldIndex
- TaskPlanningStrategyFieldIndex
- TaskPriorityFieldIndex
- TaskQuantityFieldIndex
- TaskReleaseDateFieldIndex
- TaskResultEndDateFieldIndex
- TaskResultPostEndDateFieldIndex
- TaskResultPreparationStartDateFieldIndex

- TaskResultProcessingStepFieldIndex
- TaskResultProcessingTimeFieldIndex
- TaskResultRouteFieldIndex
- TaskResultStartDateFieldIndex
- ToleranceTimeOnASAPDueDates
- ToleranceTimeOnJITReleaseDates
- ToleranceTimeOnStartLockDates
- WorkInProgressType
- WritingDebugFilesEnabled

Methoden

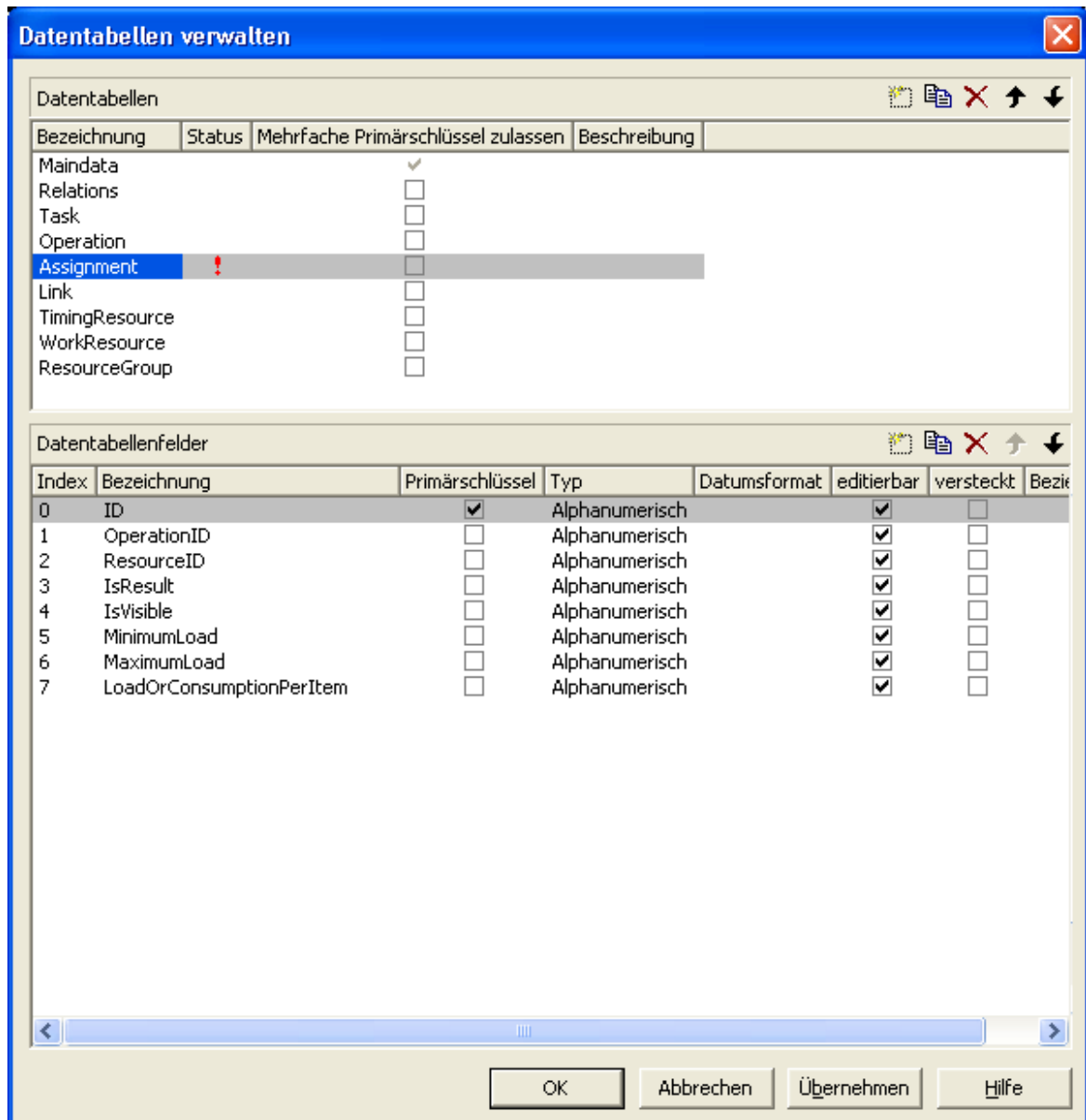
- DetermineIDOfFirstOperationByTaskID
- DetermineIDOfLastOperationByTaskID
- Process

Eigenschaften

AssignmentDataTableName

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können sie den Namen der Zuweisungstabelle setzen oder erfragen. Sie ist die Datentabelle für die Zuweisungen von Operationen zu Ressourcen, deren Name obligat zu setzen ist.



	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Zuweisungsdatentabelle
	Mögliche Werte:	Standardwert: Leere Zeichenkette Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentDataTableName = "Assignment"
```

AssignmentIsResultFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Zuweisungstabelle setzen oder erfragen, in welches VARCHART XGantt einträgt, ob der Datensatz als Ergebnis der Ressourcenplanung erzeugt wurde. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 3. Das Setzen dieser Eigenschaft ist optional. Die Ressourcenplanung erzeugt Zuweisungen nur, wenn in den vorgegebenen Zuweisungen beim Start Zuweisungen auf Ressourcengruppen vorhanden sind. Bei Gruppen wird eine Zuweisung auf die aus der Gruppe ermittelte Ressource generiert und bei dieser das entsprechende Feld auf den Wert 1 gesetzt. In alle von der Anwendung vorgegebene Zuweisungen sollte in diesem Feld nichts oder den Wert 0 eingetragen sein.

Zudem ermöglicht die Benutzung eines solchen Feldes das mehrmalige Aufrufen mit dem jeweils gleichen Ergebnis, was der Anwendung das manuelle Zurückändern der Zuweisungen auf den Ursprungszustand erspart. Die Ressourcenplanung verwendet erzeugte Zuweisungen immer wieder, um unnötiges Löschen und Erzeugen von Datensätzen zu vermeiden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme der Werte zur Kennzeichnung der von der Ressourcenplanung erzeugten Datensätze vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentIsResultFieldIndex = 3
```

AssignmentIsVisibleFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfeldes der Zuweisungstabelle, in welches die Ressourcenplanung einträgt, ob die Zuweisung im Gantt-Graphen sichtbar gemacht werden sollten. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 4. Das Feld ist nützlich, wenn Sie z. B. vor dem Planungsdurchlauf im Gantt-

Graphen die Zuweisungen auf die Ressourcengruppen anzeigen möchten und nach dem Durchlauf die Zuweisungen auf die ermittelten Einzelressourcen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme der Werte zur Sichtbarkeit vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentIsVisibleFieldIndex = 4
```

AssignmentLoadOrConsumptionPerItemFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfeldes der Zuweisungstabelle, in dem ein Wert pro Stück (s. Eigenschaft **Task-QuantityFieldIndex**) festgelegt werden kann. Ein Wert kann nur für Zuweisungen an Arbeits- oder Materialressourcen gesetzt werden. Enthält der Index den Wert -1, wird 1 angenommen. Steht im Datenfeld des Datensatzes kein gültiger Wert, wird 0 angenommen. Wenn das Datenfeld vom Typ **String** ist, können Sie im Datenfeld auch eine Fließkommazahl angeben.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme des Wertes vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

AssignmentMaximumLoadFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Zuweisungstabelle, das für den maximalen Auslastungsgrenzwert einer

Ressource vorgesehen ist. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 6.

Ein derartiger Höchstwert kann nur für Zuweisungen an zeitdauerbestimmende Ressourcen gesetzt werden. Das Datenfeld selbst enthält relative Werte in % von {0...100}, wobei sowohl ein leeres Feld als auch der Wert 0 ebenfalls als 100 interpretiert werden.

Werte zwischen 1 und 99 im Datenfeld setzen die Eigenschaften **FullUsageOfPlanningUnitsEnabled** und **OperationMaximumInterruptionTimeFieldIndex** außer Kraft.

Beachten Sie bitte auch **AssignmentMinimumLoadFieldIndex**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme des Höchstwertes zur Begrenzung der Ressourcenauslastung vorgesehen ist. {1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentMaximumLoadFieldIndex = 6
```

AssignmentMinimumLoadFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Zuweisungstabelle, das für den minimalen Auslastungsgrenzwert einer Ressource vorgesehen ist. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 5. Nur zeitdauerbestimmenden Ressourcen kann ein derartiger Tiefstwert zugewiesen werden.

Das Datenfeld selbst enthält relative Werte in % von {0...100}. Bitte beachten Sie auch **AssignmentMaximumLoadFieldIndex**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme des Mindestwertes zur Begrenzung der Ressourcenauslastung vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentMinimumLoadFieldIndex = 5
```

AssignmentMinimumMaximumLoadType

Nur-Lese-Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie bestimmen oder erfragen, ob die Werte, die in den Datenfeldern mit den über die Eigenschaften **AssignmentMinimumLoadFieldIndex** und **AssignmentMaximumLoadFieldIndex** gesetzten Indizes bereitgestellt werden, relativ zur Ressourcenkapazität sind oder absolut. Absolute Werte sind z. B. hilfreich, wenn die zugewiesene Ressource ein Team mit einer nicht konstanten Anzahl Personen ist und die Zuweisung nicht das ganze Team belegen soll.

	Datentyp	Beschreibung
Eigenschaftswert	ResourceSchedulingMinimumMaximumLoadTypeEnum	Feldwerte absolut/relativ zur Ressourcenkapazität
	Mögliche Werte: vcResSchedAbsoluteValues 2 vcResSchedPercentageValues 0	Datenfeldwerte absolut zur Ressourcenkapazität Datenfeldwerte relativ zur Ressourcenkapazität

AssignmentOperationIDFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index des Datenfeldes in der Zuweisungstabelle setzen oder erfragen, das die ID einer Operation enthält. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 1. Diese Eigenschaft darf für einen Planungsdurchlauf nicht auf -1 gesetzt sein.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme der Operation-ID vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentOperationIDFieldIndex = 1
```

AssignmentResourceIDFieldIndex**Eigenschaft von VcResourceScheduler2**

Mit dieser indizierten Eigenschaft können Sie den Index eines Datenfeldes in der Zuweisungstabelle setzen oder erfragen, das IDs für Ressourcen enthält. In der Abbildung zu **AssignmentDataTableName** ist der Index beispielsweise 2.

Der als Parameter übergebene Ressourcentabellenindex bezeichnet eine von 25 möglichen Ressourcentabellen. Die davon verwendeten Tabellen werden über die indizierte Eigenschaft **ResourceDataTableName** definiert.

	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index einer Ressourcendatentabelle entsprechend der Zuweisung durch die Eigenschaft ResourceDataTableName {0...24}
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme von Ressourcen-IDs vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.AssignmentResourceIDFieldIndex(0) = 2
```

AssignmentResourceSelectionStrategyFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Zuweisungstabelle, das für die jeweilige Zuweisung zu einer Ressourcen-Gruppe eine eigene Ressourcenauswahlstrategie bestimmt. Ist dieses Datenfeld an einer Ressource leer oder ist die Eigenschaft auf -1 gesetzt, dann gilt defaultmäßig der Wert der allgemeinen Eigenschaft **ResourceSelectionStrategy** (siehe dort).

Das Datenfeld kann folgende Werte enthalten:

0: entspricht vcResSchedRSSequential

1: entspricht vcResSchedRSLeastLoaded

2: entspricht vcResSchedRSMostLoaded

3: entspricht vcResSchedRSHighestEfficiency

7: entspricht vcResSchedRSFirstAvailable

Bei den Werten 1 und 2 (LeastLoaded bzw. MostLoaded) wird bei der Einplanung der Ressourcen eine laufende Summe der Belegungen aufaddiert, anhand derer die Entscheidung für die mindest- bzw. meistbelegte Ressource getroffen wird. Daher ist es nicht ausgeschlossen, dass bei stark differierenden Planungszeiträumen von Aufgaben oder bei Nutzung beider Planungsstrategien das Ergebnis nicht zufriedenstellend ausfällt.

Bei dem Wert 7 (FirstAvailable) ist die Entscheidung nur von der zuerst verfügbaren zeitdauergebenden Ressource abhängig. Weitere Zuweisungen der gleichen Operation werden nicht berücksichtigt, so dass bei Verwendung von Material- und Arbeitsressourcen ein nicht zufriedenstellendes Ergebnis entstehen kann.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Zuweisungsdatentabelle, das für die Aufnahme der Daten zur Ressourcenauswahlstrategie vorgesehen ist. {-1...AnzahlDatenfelderInZuweisungsdatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Zuweisungsdatentabelle zugeordnet ist. Standardwert: -1

BaseCalendarUsageForSupplementTimes

Eigenschaft von VcResourceScheduler2

Wenn diese Eigenschaft auf **false** gesetzt ist, wird für die Mindest-Ergänzungszeiten (die indirekt über die Eigenschaft **VcResourceScheduler2.OperationMinimumSupplementTimeFieldIndex** definiert werden) kein Kalender verwendet, d.h. die angegebene Zeit wird direkt verwendet (Praxisbeispiel: geeignet für die Trocknung produzierter Teile). Wenn diese Eigenschaft auf **true** gesetzt ist, wird der Basiskalender des Gantt-Objektes verwendet, d.h. die Ergänzungszeit wird als Arbeitszeit abgearbeitet, die im Basiskalender definiert wurde (Praxisbeispiel: geeignet für einen Transport produzierter Teile).

Bitte sehen Sie ergänzend auch **VcResourceScheduler2.OperationMinimumSupplementTimeFieldIndex**.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	True: Der Basiskalender des Gantt-Objekts wird verwendet. False: Die angegebene Zeit wird direkt verwendet. Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ResourceScheduler2.BaseCalendarUsageForSupplementTimes = True
```

BaseTimeUnit

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie eine von der Eigenschaft **VcGantt.TimeUnit** abweichende Basiszeiteinheit für die Ressourcenplanung festlegen oder erfragen. Werte in den Kapazitäts-, Belegungs- und Lagerbestandskurven sind immer auf die hier angegebene Basiszeiteinheit bezogen.

	Datentyp	Beschreibung
Eigenschaftswert	TimeUnitEnum	Zeiteinheit Standardwert: Wert, der zur Designzeit über vcGantt.TimeUnit festgelegt wurde. Wenn keine Veränderungen vorgenommen wurde, ist der Wert vcDay .
	Mögliche Werte: vcDay 5 vcHour 6 vcMinute 7 vcSecond 8	Zeiteinheit Tag Zeiteinheit Stunde Zeiteinheit Minute Zeiteinheit Sekunde

Code-Beispiel

```
VcResourceScheduler2.BaseTimeUnit = VcTimeUnit.vcMinute
VcResourceScheduler2.BaseTimeUnitsPerStep = 15
```

BaseTimeUnitsPerStep

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie die Schrittweite für die Ressourcenplanung festlegen oder erfragen. Je höher dieser Wert ist, umso schneller, aber auch umso grober wird die Berechnung. Die Zahl bezeichnet ein Vielfaches der über **VcResourceScheduler2.BaseTimeUnit** festgelegten Grundeinheit.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Zeiteinheiten pro Schritt Standardwert: 1
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
VcGantt1.ResourceScheduler2.BaseTimeUnitsPerStep = 4
```

DataRecordEventsEnabled

Eigenschaft von VcResourceScheduler2

Wenn diese Eigenschaft auf **True** gesetzt wird, dann werden die Ereignisse aufgerufen, die Datenveränderungen während der Process-Methoden anzeigen, das heißt, bei VcGantt.OnDataRecordModify, VcGantt.OnDataRecordModifyComplete, VcGantt.OnDataRecordCreate, VcGantt.OnDataRecordCreateComplete, VcGantt.OnDataRecordDelete und VcGantt.OnDataRecordDeleteComplete.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	True: Ereignisse werden ausgelöst. False: Ereignisse werden nicht ausgelöst. Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ResourceScheduler2.DataRecordEventsEnabled = True
```

DefaultOperationMaximumInterruptionTime

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie einen Wert für die maximale Zeitdauer einer Unterbrechung der Verarbeitung von Operationen setzen oder erfragen. Der Wert ist eine Zahl, die in Basiszeiteinheiten angegeben wird (s. Eigenschaft **BaseTimeUnit**). Sie wird wirksam, wenn die Eigenschaft **OperationMaximumInterruptionTimeFieldIndex** auf -1 gesetzt ist oder der aus einem Datensatz gelesene Wert der Operationstabelle 0 ist oder das Feld leer ist. Wird der Wert auf 0 gesetzt, ist keine Unterbrechung erlaubt.

Diese Eigenschaft wird durch die Herabsetzung der maximalen Auslastung auf weniger als 100% (s. Eigenschaft **AssignmentMaximumLoadFieldIndex**) außer Kraft gesetzt.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Basiszeiteinheiten Standardwert: 0

Code-Beispiel

```
VcGantt1.ResourceScheduler2.DefaultOperationMaximumInterruptionTime = 1
```

DefaultResourceCalendarName

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Namen für einen Standardkalender vorgeben, der verwendet wird, wenn über die Eigenschaften **VcResourceScheduler2.ResourceCalendarNameFieldIndex** und **VcResourceScheduler2.ResourceNameFieldIndex** kein ressourcenspezifischer Kalender gefunden wurde. Wird die Eigenschaft nicht gesetzt, greift die Ressource auf den Standard-Kalender des XGantt-Objektes zurück (siehe **VcCalendarCollection.Active**)..

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Kalenders Standardwert: Leere Zeichenkette
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.DefaultResourceCalendarName = ""
```

FullUsageOfPlanningUnitsEnabled

Eigenschaft von VcResourceScheduler2

Wenn diese Eigenschaft auf **True** gesetzt wird, dann kann in der ersten und/oder letzten Planungszeiteinheit der Belegung einer Ressource durch eine Aufgabe eine weitere Aufgabe enden bzw. beginnen. Auf diese Weise können vorhandene Restkapazitäten genutzt werden. Bei **False** werden Restkapazitäten dagegen nicht genutzt.

Die Eigenschaft beeinflusst nur den Beginn der ersten Operation einer Aufgabe. Auf die Operationen innerhalb einer Aufgabe hat diese Eigenschaft keine Auswirkung.

Diese Eigenschaft wird durch die Herabsetzung der maximalen Auslastung auf weniger als 100% (s. Eigenschaft **AssignmentMaximumLoadFieldIndex**) außer Kraft gesetzt.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	True: Vorhandene Restkapazitäten werden genutzt. False: Vorhandene Restkapazitäten werden nicht genutzt. Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

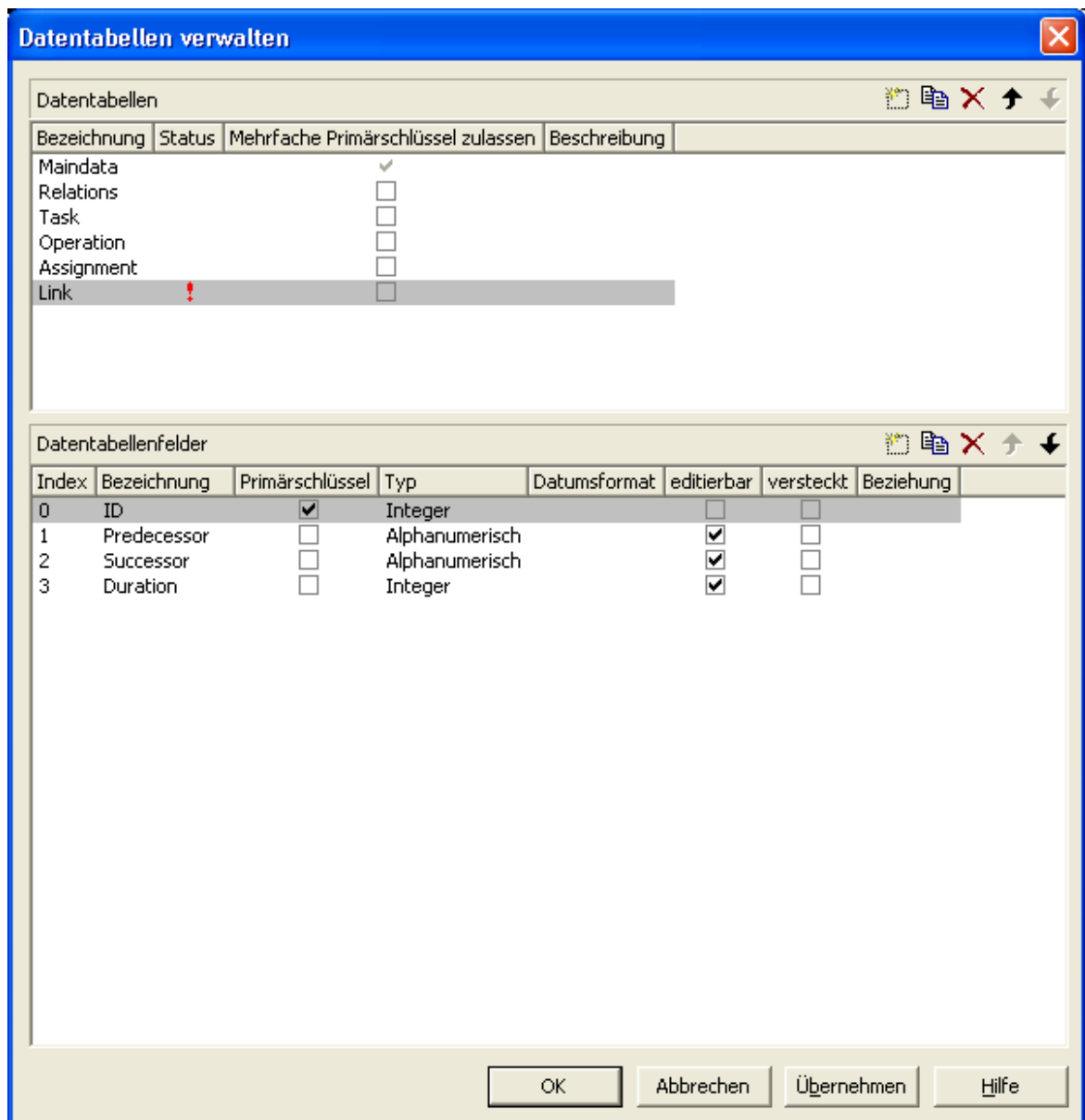
Code-Beispiel

```
VcGantt1.ResourceScheduler2.FullUsageOfPlanningUnitsEnabled = True
```

LinkDataTableName

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Namen der Verbindungsdatentabelle setzen oder erfragen. Wenn Sie diesen Namen nicht setzen, werden Verbindungen bei der Ressourcenplanung nicht berücksichtigt.



	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Verbindungsdatentabelle Standardwert: Leere Zeichenkette
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkDataTableName = "Link"
```

LinkDurationFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index des Datenfeldes in der Verbindungstabelle setzen oder erfragen, in dem ein minimaler zeitlicher Abstand zwischen Vorgänger und Nachfolger abgelegt werden kann. Dieser zeitliche Abstand kann auch negativ sein. Einheiten: Basiszeiteinheiten (s. Methode `BaseTimeUnit`). In der Abbildung zu **LinkDataTableName** ist der Index beispielsweise 3.

Als Begrenzung kann der Nachfolger bei Planungsstrategie ASAP nicht früher anfangen als der Vorgänger und bei Planungsstrategie JIT der Vorgänger nicht später aufhören als der Nachfolger.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Verbindungsdatentabelle, das für die Aufnahme der Dauer vorgesehen ist. {1...AnzahlDatenfelderInVerbindungstabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Verbindungstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkDurationFieldIndex = 3
```

LinkPredecessorOperationIDFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Verbindungstabelle setzen oder erfragen, das die ID der Vorgängeroperation enthält. Da das ResourceScheduling-Modul nur Aufgaben (Tasks) miteinander verbinden kann, erleichtert diese Eigenschaft die Verwendung der Verbindungen in XGantt, die zur Zeit nur zwischen Operationen dargestellt werden. Intern wird daher immer eine Verbindung zwischen den Tasks der durch die ID angegebenen Operationen erzeugt.

Bei Verwendung einer Verbindungstabelle ist diese Eigenschaft verbindlich auf einen Wert ungleich -1 zu setzen, wenn nicht die `VcResourceScheduler2`-Eigenschaft **LinkPredecessorTaskIDFieldIndex** verwendet wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Verbindungsdatentabelle, das für die Aufnahme von IDs von Vorgängeroperationen vorgesehen ist. {-1...AnzahlDatenfelderInVerbindungstabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Verbindungstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkPredecessorOperationIDFieldIndex = 1
```

LinkPredecessorTaskIDFieldIndex**Eigenschaft von VcResourceScheduler2**

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Verbindungstabelle setzen oder erfragen, das die ID des Vorgängerauftrags enthält. In der Abbildung zu **LinkDataTableName** ist der Index beispielsweise 1.

Bei Verwendung einer Verbindungstabelle ist diese Eigenschaft verbindlich auf einen Wert ungleich -1 zu setzen, wenn nicht die VcResourceScheduler2-Eigenschaft **LinkPredecessorOperationIDFieldIndex** verwendet wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Verbindungsdatentabelle, das für die Aufnahme von IDs von Vorgängeraufgaben vorgesehen ist. {-1...AnzahlDatenfelderInVerbindungstabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Verbindungstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkPredecessorTaskIDFieldIndex = 1
```

LinkSuccessorOperationIDFieldIndex**Eigenschaft von VcResourceScheduler2**

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Verbindungstabelle setzen oder erfragen, das die ID der Nachfolgeroperation enthält. Da das ResourceScheduling-Modul nur Aufgaben (Tasks) miteinander verbinden kann, erleichtert diese Eigenschaft die Verwendung der Verbindungen in XGantt, die zur Zeit nur zwischen Operationen

dargestellt werden. Intern wird daher immer eine Verbindung zwischen den Tasks der durch die ID angegebenen Operationen erzeugt.

Bei Verwendung einer Verbindungstabelle ist diese Eigenschaft verbindlich auf einen Wert ungleich -1 zu setzen, wenn nicht die VcResourceScheduler2-Eigenschaft **LinkSuccessorTaskIDFieldIndex** verwendet wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Verbindungsdatentabelle, das für die Aufnahme von IDs von Nachfolgeoperationen vorgesehen ist. {-1...AnzahlDatenfelderInVerbindungstabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Verbindungstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkSuccessorOperationIDFieldIndex = 1
```

LinkSuccessorTaskIDFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Verbindungstabelle setzen oder erfragen, das die ID des Nachfolgerauftrags enthält. In der Abbildung zu **LinkDataTableName** ist der Index beispielsweise 2.

Bei Verwendung einer Verbindungstabelle ist diese Eigenschaft verbindlich auf einen Wert ungleich -1 zu setzen, wenn nicht die VcResourceScheduler2-Eigenschaft **LinkSuccessorOperationIDFieldIndex** verwendet wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Verbindungsdatentabelle, das für die Aufnahme von IDs von Nachfolgeraufgaben vorgesehen ist. {-1...AnzahlDatenfelderInVerbindungstabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Verbindungstabelle zugeordnet ist. Standardwert: -1

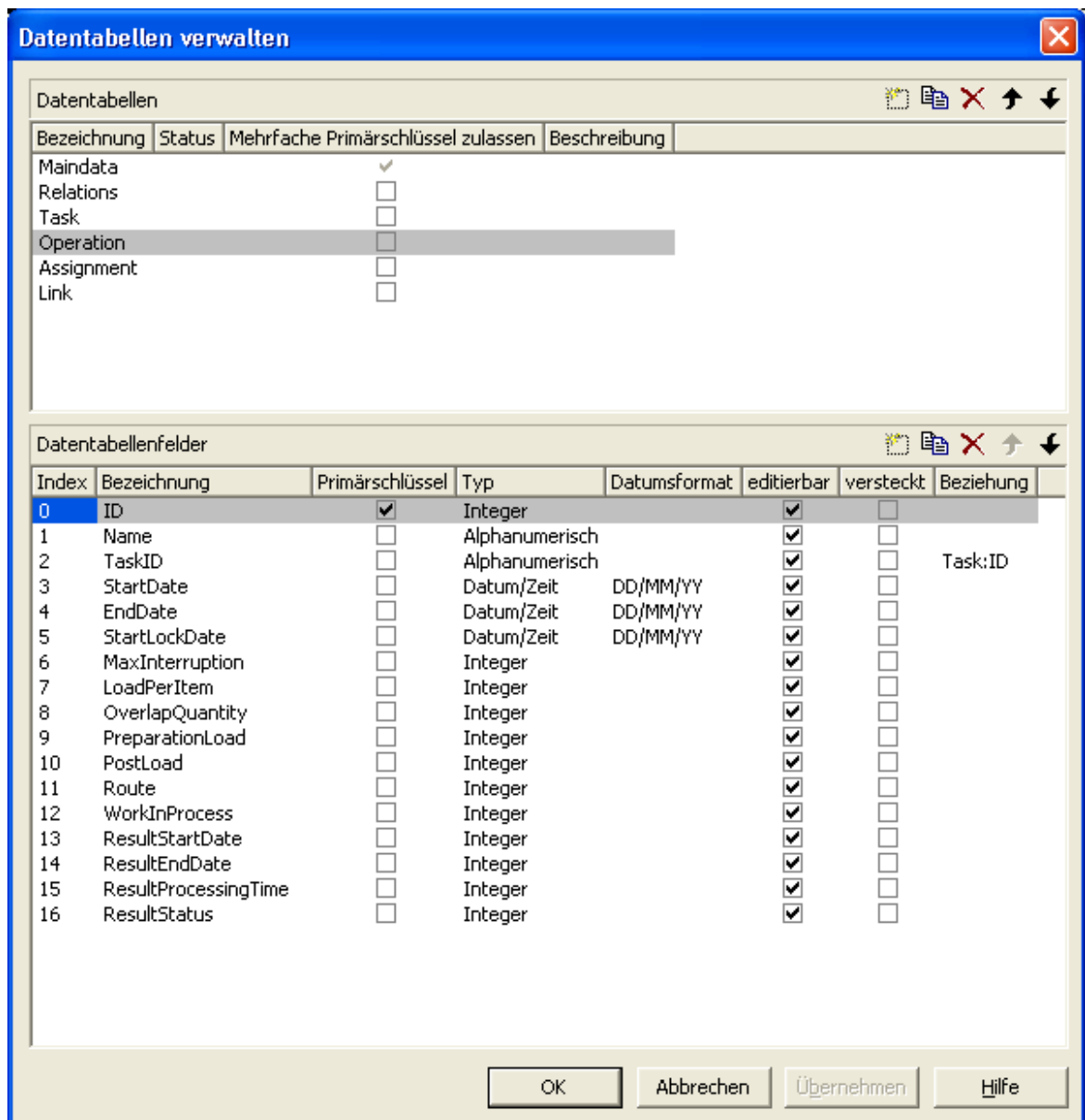
Code-Beispiel

```
VcGantt1.ResourceScheduler2.LinkSuccessorTaskIDFieldIndex = 2
```

OperationDataTableName

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Namen der Operationstabelle setzen oder erfragen. Sie ist die Datentabelle für Operationen, deren Name obligat zu setzen ist.



	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Operationstabelle Standardwert: Leere Zeichenkette
	Mögliche Werte:	Name der Farbzuordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationDataTableName = "Operation"
```

OperationLoadPerItemFieldIndex**Eigenschaft von VcResourceScheduler2**

Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfelds in der Operations-Datentabelle, das die Belegung der zeitdauergebenden Ressourcen pro Stück bezeichnet. Für die Gesamtbelegung auf der zeitdauergebenden Ressource wird dieser Wert mit der in der Aufgabe angegebenen Anzahl multipliziert. Wenn in einem Datenfeld ein ungültiger Wert steht oder diese Eigenschaft auf -1 gesetzt ist, dann wird der Wert zu 0 angenommen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der Auslastung vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationLoadPerItemFieldIndex = 10
```

OperationMaximumInterruptionTimeFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Operationstabelle, das für einen Wert vorgesehen ist, der die maximale Zeitdauer bezeichnet, die diese Operation während ihrer Verarbeitungszeit unterbrochen werden darf. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 9.

Eine "Unterbrechung" ist eine aktionsfreie Zeit auf einer voll belegten und einer Operation zugewiesenen Ressource. Sie unterscheidet sich von einer "Pause" dadurch, dass sie nicht durch eine vordefinierte Nicht-Arbeitszeit verursacht wird.

Der Feldinhalt ist eine Zahl, die in Basiszeiteinheiten angegeben wird (s. Eigenschaft **BaseTimeUnit**).

Ist diese Eigenschaft auf -1 gesetzt oder der Wert im jeweiligen Datensatz leer oder 0, dann wird der Wert verwendet, der durch die Eigenschaft

DefaultOperationMaximumInterruptionTime gesetzt wurde. Ist auch diese 0, dann ist keine Unterbrechung erlaubt. Ist der Wert im jeweiligen Datensatz < 0 , dann ist ebenfalls keine Unterbrechung erlaubt, auch wenn die Eigenschaft **DefaultOperationMaximumInterruptionTime** einen Wert ungleich 0 hat.

Diese Eigenschaft wird durch die Herabsetzung der maximalen Auslastung auf weniger als 100% (s. Eigenschaft **AssignmentMaximumLoadFieldIndex**) außer Kraft gesetzt.

	Datentyp	Beschreibung
Parameter: timeSpan	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Fertigstellungsgrades vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist.
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der maximalen Unterbrechungszeit vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationMaximumInterruptionTimeFieldIndex = 9
```

OperationMinimumSupplementTimeFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem eine Mindest-Ergänzungszeit für jede Operation definiert ist. Für diese Zeitspanne wird auf den der Operation zugewiesenen Ressourcen keine Belegung vorgenommen, das heißt, diese Zeitspanne kann z. B. für Warte- oder Transportzeiten genutzt werden, die unabhängig von den zugewiesenen Ressourcen auftreten.

Der Feldinhalt ist eine Zahl, die in Basiszeiteinheiten angegeben wird (s. Eigenschaft **BaseTimeUnit**). In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 7.

Bitte sehen Sie ergänzend auch **OperationMaximumSupplementTimeFieldIndex** sowie **OperationPreparationLoadFieldIndex** und **OperationPostLoadFieldIndex**

BaseCalendarUsageForSupplementTimes.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der Mindest-Ergänzungszeit vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationMinimumSupplementTimeFieldIndex = 7
```

OperationOverlapQuantityFieldIndex

Eigenschaft von VcResourceScheduler2

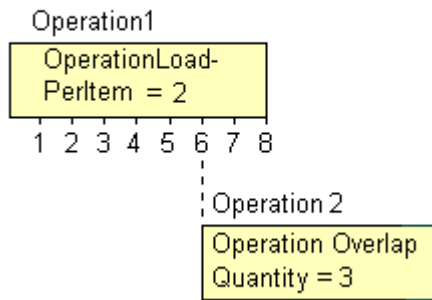
Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfeldes der Operationstabelle, in dem die 'Überlappungsmenge' einer Operation angegeben werden kann.

Überlappungen können nur in Aufgaben auftreten, die nach der Strategie ASAP eingeplant worden sind. Mit dieser Eigenschaft können Sie überlappende Ressourcenbelegungen aufeinanderfolgender Operationen erreichen. Dies ist sinnvoll, wenn die nachfolgende Operation nicht auf die vollständige Fertigstellung der vorhergehenden warten muss.

Die im Datenfeld angegebene Menge bezieht sich auf die fertiggestellte Menge der Aufgabe (s. Eigenschaft **TaskQuantityFieldIndex**), ab der die nachfolgende Operation überlappend gestartet werden kann. Es handelt sich um die Angabe einer Mindestmenge, d.h. nach wievielen Einheiten die Überlappung frühestens starten darf, bei optional späterem Start.

Im Beispiel unten ist der Overlap-Wert=3, der sich auf die Aufgabenmenge 4 bezieht. Wenn von diesen 4 Einheiten 3 durch die Operation1 abgearbeitet worden sind, darf Operation2 starten. Wenn Sie einen Mengenfaktor (OperationLoadPerItem) für die Operation1 gesetzt haben (im Bild ist dieser =2), wird dieser mit dem Overlap-Wert multipliziert: $3 \cdot 2 = 6$. Daher startet Operation2 erst, wenn Operation1 den Wert 6 erreicht hat:

Task Quantity = 4



Beispiel-Szenario: Es sollen 4 Kerzenleuchter mit jeweils 3 Kerzen produziert werden. Je 2 Leuchter mit 6 Kerzen werden zusammen verpackt. Nachdem über Operation1 final 6 Kerzen produziert worden sind, kann die Operation2 mit der Verpackung beginnen.

Wird in dem Index nichts oder ein Wert = 0 angegeben, dann findet keine Überlappung mit der Vorgängeroperation statt; bei -1 startet die Operation gleichzeitig mit der Vorgängeroperation.

Ist gleichzeitig eine Vorbereitungszeit definiert, wird diese in der Überlappung berücksichtigt, so dass dann für die Überlappungsmenge ggf. diese Vorbereitungszeit geteilt durch die Auslastung pro Stück der Operation (s. Eigenschaft **OperationLoadPerItemFieldIndex**) hinzuaddiert werden muss. Diese Eigenschaft sollte nicht zusammen mit der Eigenschaft **ResourceEfficiencyFieldIndex** verwendet werden. Auch kann es Probleme bei der gleichzeitigen Verwendung von **AssignmentMaximumLoadFieldIndex** geben.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der "Überlappungsmenge" vorgesehen ist. {1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationOverlapQuantityFieldIndex = 11
```

OperationPostLoadFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem eine Nachlaufzeit für jede Operation definiert ist. Für diese Zeitspanne wird auf den der Operation zugewiesenen Ressourcen eine Belegung vorgenommen.

Der Feldinhalt ist eine Zahl, die die erforderliche Kapazität darstellt. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 13.

Bitte sehen Sie dazu auch **OperationPreparationLoadFieldIndex**, **OperationMaximumInterruptionTimeFieldIndex** und **OperationMinimumSupplementTimeFieldIndex**.

Ressourcenunabhängige Nachlaufzeiten können über die Eigenschaft **OperationPostOffsetFieldIndex** definiert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der Nachlaufzeit vorgesehen ist. {1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationPostLoadFieldIndex = 13
```

OperationPostOffsetFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem eine Nachlaufzeit für jede Operation definiert ist. Enthält dieses Feld positive ganze Zahlen (in der Einheit der gültigen Basiszeiteinheit), so ist die Nachlaufzeit der Operationen von den Ressourcen unabhängig.

Wenn der Index den Wert -1 enthält, gibt es keine Nachlaufzeiten. Dies gilt ebenso für Operationen, bei denen der Index auf ein Datenfeld verweist, das bei der jeweiligen Operation keine gültige Zahl oder eine 0 enthält.

Bitte sehen Sie dazu auch **OperationPreparationOffsetFieldIndex**.

Ressourcenabhängige Nachlaufzeiten können über die Eigenschaft **OperationPostLoadFieldIndex** definiert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, mit dem bestimmt wird, ob die Nachlaufzeit einer Operation ressourcenunabhängig sein soll. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass keine Nachlaufzeiten existieren. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationPostOffsetFieldIndex = 8
```

OperationPreparationLoadFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem eine Vorbereitungszeit für jede Operation definiert ist. Für diese Zeitspanne wird auf den der Operation zugewiesenen Ressourcen eine Belegung vorgenommen.

Der Feldinhalt ist eine Zahl, die die erforderliche Kapazität darstellt. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 12.

Bitte sehen Sie dazu auch **OperationPostLoadFieldIndex**, **OperationMaximumInterruptionTimeFieldIndex** und **OperationMinimumSupplementTimeFieldIndex**.

Ressourcenunabhängige Nachlaufzeiten können über die Eigenschaft **OperationPreparationOffsetFieldIndex** definiert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der Vorbereitungszeit vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationPreparationLoadFieldIndex = 12
```


OperationPreparationOffsetFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem eine Vorlaufzeit für jede Operation definiert ist. Enthält dieses Feld positive ganze Zahlen (in der Einheit der gültigen Basiszeiteinheit), so ist die Vorlaufzeit der Operationen von den Ressourcen unabhängig.

Wenn der Index den Wert -1 enthält, gibt es keine Vorlaufzeiten. Dies gilt ebenso für Operationen, bei denen der Index auf ein Datenfeld verweist, das bei der jeweiligen Operation keine gültige Zahl oder eine 0 enthält.

Bitte sehen Sie dazu auch **OperationPostOffsetFieldIndex**.

Ressourcenabhängige Nachlaufzeiten können über die Eigenschaft **OperationPreparationLoadFieldIndex** definiert werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, mit dem bestimmt wird, ob die Vorlaufzeit einer Operation ressourcenunabhängig sein soll. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass keine Vorlaufzeiten existieren. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationPreparationOffsetFieldIndex = 8
```

OperationResultEndDateFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfeldes der Operationstabelle, in das das berechnete Enddatum für die jeweilige Operation eingetragen wird. In der Abbildung zu **OperationDataTable-Name** ist der Index beispielsweise 17.

Um sinnvolle Ergebnisse aus der Ressourcenplanung zu erhalten, müssen mindestens zwei der drei Eigenschaften **OperationResultStartDateFieldIndex**, **OperationResultProcessingTimeFieldIndex** und **OperationResult-EndDateFieldIndex** ungleich -1 sein.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Enddatums vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultEndDateFieldIndex = 17
```

OperationResultPostEndDateFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in das nach einer erfolgten Planung die Endezeit der Nachlaufphase einer Operation abgelegt wird. Wenn die Nachlaufzeit 0 ist, dann ist dieses Datum identisch mit dem Wert in dem Datenfeld, das über die Eigenschaft **OperationResult-EndDateFieldIndex** adressiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Enddatums der Nachlaufphase vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultPostEndDateFieldIndex = 15
```

OperationResultPreparationStartDateFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in das nach einer erfolgten Planung die Startzeit der Vorbereitungsphase einer Operation abgelegt wird. Wenn die Vorbereitungszeit 0 ist, dann ist dieses Datum identisch mit dem Wert in dem Datenfeld, das über die Eigenschaft **OperationResultStartDateFieldIndex** adressiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Startdatums der Vorbereitungsphase vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultPreparationStartDateFieldIndex = 10
```

OperationResultProcessingTimeFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft setzen oder erfragen Sie den Index eines Datenfeldes der Operationstabelle, in das die berechneten Verarbeitungszeitraum für die jeweilige Operation eingetragen wird. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 18.

Um sinnvolle Ergebnisse aus der Ressourcenplanung zu erhalten, müssen mindestens zwei der drei Eigenschaften **OperationResultStartDateFieldIndex**, **OperationResultProcessingTimeFieldIndex** und **OperationResultEndDateFieldIndex** ungleich -1 sein.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Verarbeitungszeitraums vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultProcessingTimeFieldIndex = 18
```

OperationResultSelectedTimingResourceIDFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in das nach einer erfolgten Planung die vom Modul ausgewählte ID der zeitdauerbestimmenden Ressource abgelegt wird. Auf diese Art kann in der

Tabelle oder einer Layerbeschriftung eine zugewiesene Ressource einfach grafisch dargestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	-Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der ID der zeitdauerbestimmenden Ressource vorgesehen ist, die vom Modul bei der Berechnung ausgewählt wird. {0...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist.

Code-Beispiel

```
VcGantt1.OperationResultSelectedTimingResourceFieldIndex = 8
```

OperationResultStartDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in das das berechnete Startdatum für die jeweilige Operation eingetragen wird. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 16.

Um sinnvolle Ergebnisse aus der Ressourcenplanung zu erhalten, müssen mindestens zwei der drei Eigenschaften **OperationResultStartDateFieldIndex**, **OperationResultProcessingTimeFieldIndex** und **OperationResultEndDateFieldIndex** ungleich -1 sein.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Startdatums vorgesehen ist. {-1...AnzahlDatensätzeInOperationTabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultStartDateFieldIndex = 16
```

OperationResultStatusFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, das einen Fehler- oder einen Warnstatus aufnimmt. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 19.

Mögliche Werte, die von der Ressourcenplanung hier eingetragen werden:

0: die Operation wurde fehlerfrei eingeplant

1: die Operation wurde nicht eingeplant, weil die Ressourcenplanung eine andere Route der zugehörigen Aufgabe ausgewählt hat. Der Fall kann nur auftreten, wenn die Eigenschaft **OperationRouteFieldIndex** auf einen Wert ungleich -1 gesetzt wurde.

1000: die Operation wurde nicht eingeplant

1001: die Operation wurde nicht eingeplant und ist die Operation einer Aufgabe, die verursacht hat, dass die gesamte Aufgabe nicht einplanbar war. Die Gründe dafür können vielfältig sein.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Fehlerstatus vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationstabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationResultStatusFieldIndex = 19
```

OperationRouteFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in der Operationstabelle setzen oder erfragen, über dessen Werte Operationen einer bestimmten Route zugewiesen werden. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 14.

Operationen mit gleichem Inhalt in diesem Feld gehören der gleichen Route an. Der Inhalt des Felds bezeichnet damit den Namen einer Route.

Mit Routen können Alternativen für die Ausführung einer Aufgabe (Task) beschrieben werden, das heißt, die Ressourcenplanung prüft alle Routen für die Ausführung einer Aufgabe und wählt eine Route aus. So können Sie mehrere alternative Operationsketten für eine Aufgabe definieren. Nicht mehr als 10 Routen können pro Aufgabe definiert werden. Die Route wird in der Reihenfolge des Auftretens der Routen in den Operationen ausgewählt.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Namens der Route vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationRouteFieldIndex = 14
```

OperationSequenceNumberFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index des Datenfeldes in der Operationstabelle setzen oder erfragen, über dessen Werte die Reihenfolge der Operationen innerhalb einer Aufgabe (Task) festgelegt wird. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 6.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der Reihenfolge vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationSequenceNumberFieldIndex = 6
```

OperationStartLockDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem für jede Operation im Falle der Planungsstrategie ASAP (s. Eigenschaft **PlanningStrategy**) ein Startdatum hinterlegt werden kann. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 5.

Wenn das Datenfeld einen gültigen Datumseintrag enthält, dann wird die Aufgabe an dieses Startdatum gebunden und durch die Ressourcenplanung nicht mehr verschoben, was insbesondere für bereits begonnene Aufgaben sinnvoll ist (siehe in diesem Zusammenhang bitte auch die Eigenschaft **OperationWorkInProgressFieldIndex**).

Sie können über die Eigenschaft **ToleranceTimeOnStartLockDates** eine Toleranz festlegen, um die eine Operation später anfangen darf, als über das gebundenes Startdatum festgelegt. Beachten Sie auch, dass Aufgaben mit Operationen, die ein gebundenes Startdatum haben, nicht automatisch bevorzugt eingeplant werden. Wenn Sie dies wünschen, dann müssen Sie die Prioritäten der Aufgaben selbst berechnen (s. Eigenschaft **TaskPriorityFieldIndex**).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des feststehenden Startdatums vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationStartLockDateFieldIndex = 5
```

OperationTaskIDFieldIndex

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Index des Datenfeldes in der Operationstabelle setzen oder erfragen, in dem die ID der zugehörigen Aufgabe abgelegt ist. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 2.

Für die Einplanung der Operation muss diese Eigenschaft auf einen Wert ungleich -1 gesetzt werden. Mithilfe dieses Datenfelds können einer Aufgabe beliebig viele Operationen zugewiesen werden. Die Reihenfolge, in der die Operationen einer Aufgabe eingeplant werden, hängt vom Wert in dem Datenfeld ab, dessen Index über die Eigenschaft **OperationSequenceNumberDataFieldIndex** eingestellt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme der ID der Aufgabe vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationTaskIDFieldIndex = 2
```

OperationWorkInProcessFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Operationstabelle, in dem der Fertigstellungsgrad einer Operation enthalten ist. In der Abbildung zu **OperationDataTableName** ist der Index beispielsweise 15.

Falls der Datenfeldindex auf -1 gesetzt ist oder im Datenfeld kein gültiger Wert gefunden wird, wird 0% angenommen ("nicht angefangen").

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Operationsdatentabelle, das für die Aufnahme des Fertigstellungsgrades vorgesehen ist. {-1...AnzahlDatenfelderInOperationsdatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Operationsdatentabelle zugeordnet ist. Standardwert: -1

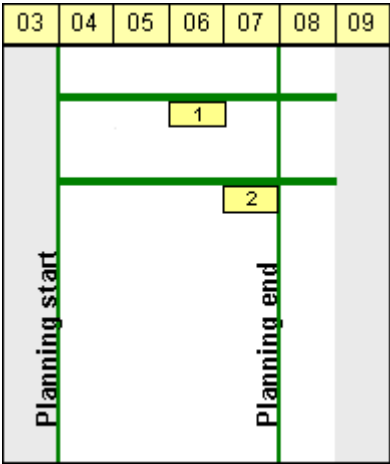
Code-Beispiel

```
VcGantt1.ResourceScheduler2.OperationWorkInProcessFieldIndex = 15
```


PlanningEndDate

Eigenschaft von VcResourceScheduler2

Mit Hilfe dieser Eigenschaft setzen oder erfragen Sie das Enddatum des Planungszeitraums für die Ressourcenplanung. Wenn Sie kein Enddatum setzen, wird das Ende der Zeitskala verwendet (gesetzt durch die Eigenschaft **VcGantt.TimeScaleEnd**). Den Anfang des Zeitraums setzen Sie mit der Eigenschaft **PlanningStartDate**.



Begrenzter Planungszeitraum

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Enddatum der Planungsperiode Standardwert: DateTime.MinValue

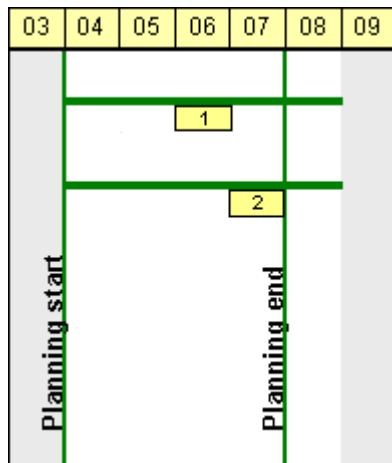
Code-Beispiel

```
VcGantt1.ResourceScheduler2.PlanningEndDate = VcGantt1.TimeScaleEnd
```

PlanningStartDate

Eigenschaft von VcResourceScheduler2

Mit Hilfe dieser Eigenschaft setzen oder erfragen Sie das Startdatum des Planungszeitraums für die Ressourcenplanung. Wenn Sie kein Startdatum setzen, wird der Beginn der Zeitskala verwendet (gesetzt durch die Eigenschaft **VcGantt.TimeScaleStart**). Das Ende des Zeitraums setzen Sie mit der Eigenschaft **PlanningEndDate**.



Begrenzter Planungszeitraum

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Startdatum der Planungsperiode Standardwert: DateTime.MinValue

Code-Beispiel

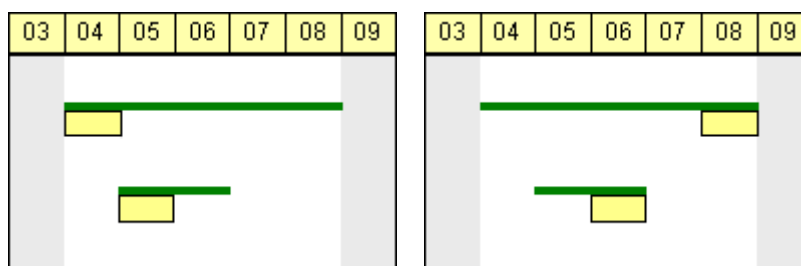
```
VcGantt1.ResourceScheduler2.PlanningStartDate = VcGantt1.TimeScaleStart
```

PlanningStrategy

Eigenschaft von VcResourceScheduler2

Diese Eigenschaft beschreibt die Planungsstrategie für alle Arbeitsaufträge.

Es gibt zwei mögliche Planungsstrategien: Bei der einen ist die Zielsetzung, Aufgaben so früh wie möglich (as soon as possible, ASAP) abzuarbeiten, um Ressourcen für Folgeaufträge bereit zu halten und einen hohen Durchsatz anzustreben. Die zweite Strategie strebt an, Aufgaben termingerecht (just in time, JIT) abzuarbeiten, um z. B. den Lagerungsaufwand gering zu halten.



Bei der ASAP-Strategie wird das Startdatum möglichst früh gelegt (Bild links), bei der JIT-Strategie wird das Enddatum möglichst spät eingeplant (Bild rechts). Die langen, linienförmigen Balken zeigen den möglichen

Ausführungszeitraum der Aufgabe an; die kurzen, rechteckigen stellen die tatsächlich eingeplante Ausführungszeit dar. ASAP-Aufgaben erscheinen daher nach Möglichkeit am Anfang, JIT-Aufgaben am Ende innerhalb des möglichen Ausführungszeitraums.

Abweichend von der hier eingestellten Planungsstrategie kann durch die Zuweisung eines Datenfeldes bei **TaskPlanningStrategyFieldIndex** für jeden Arbeitsauftrag eine individuelle Einstellung vorgenommen werden. Letztere überschreibt die allgemein gesetzte Strategie.

	Datentyp	Beschreibung
Eigenschaftswert	ResourceSchedulingPlanningStrategy Mögliche Werte: vcResSchedPSASAP 1- vcResSchedPSJIT 0	Planungsstrategie Standardwert: vcResSchedPSASAP So bald wie möglich (as soon as possible) Gerade rechtzeitig (just in time)

Code-Beispiel

```
VcGantt1.ResourceScheduler2.PlanningStrategy = vcResSchedPSASAP
```

ResourceCalendarNameFieldIndex

Eigenschaft von VcResourceScheduler2

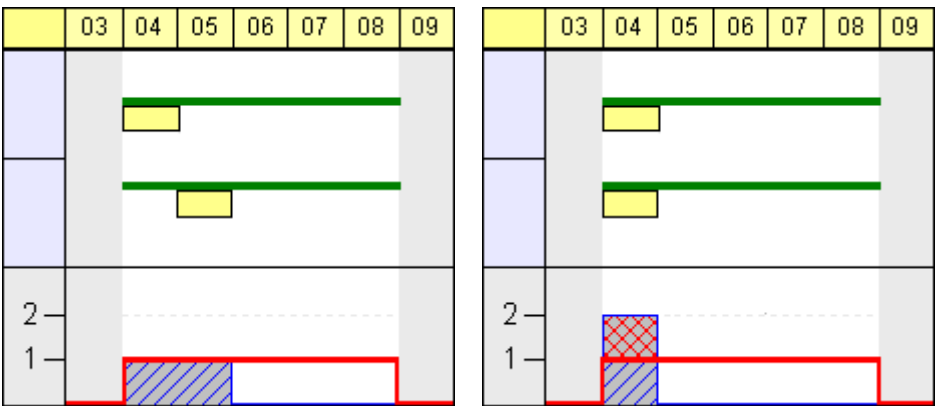
Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Ressourcentabelle, das für die jeweilige Ressource des Typs **TimingResource** oder **WorkResource** den Namen eines Kalenders bestimmt. Ist dieses Datenfeld an einer Ressource leer, enthält einen ungültigen Namen oder ist die Eigenschaft auf -1 gesetzt, dann wird ersatzweise die Bezeichnung der Ressource als Kalendernamen benutzt, die mittelbar aus der Eigenschaft **ResourceNameFieldIndex** gewonnen wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendaten-tabelle, das für eine Resource vom Typ TimingResource oder WorkResource einen Kalendernamen bestimmt.

ResourceCapacityType

Eigenschaft von VcResourceScheduler2

Diese Eigenschaft beschreibt den Kapazitätstyp für alle Ressourcen, falls dieser nicht vorgangsweise durch **ResourceCapacityTypeFieldIndex** gesetzt wird.



Bei begrenzter Kapazität werden die Aufgaben nacheinander eingeplant (links); bei unbegrenzter Kapazität können sie zeitgleich eingeplant werden (rechts).

	Datentyp	Beschreibung
Eigenschaftswert	ResourceCapacityType	Kapazitätstypen Standardwert: vcResSchedCTFinite
	Mögliche Werte: vcResSchedCTFinite -1 vcResSchedCTInfinite 0	Begrenzte Kapazitäten Unbegrenzte Kapazitäten

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceCapacityType = vcResSchedCTFinite
```

ResourceCapacityTypeFieldIndex

Eigenschaft von VcResourceScheduler2

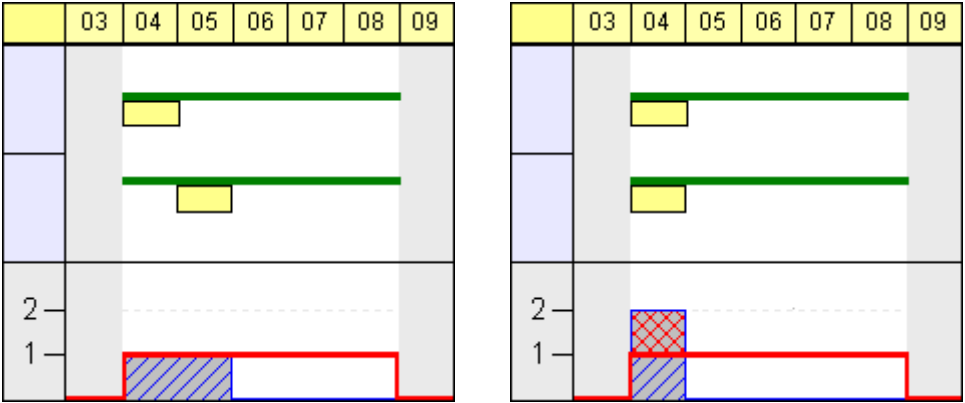
Mit dieser Eigenschaft können Sie den Index eines Datenfeldes in einer Ressourcentabelle setzen oder erfragen, in dem der Kapazitätstyp für eine einzelne zeitdauerbestimmende Ressource enthalten ist. In der Abbildung zu **ResourceDataTableName** ist der Index beispielsweise 2.

Der als Parameter übergebene Ressourcentabellenindex bezeichnet eine von 25 möglichen Ressourcentabellen, die verwendet werden soll und über die indizierte Eigenschaft **ResourceDataTableName** definiert wird.

Definierte Werte der Datenfeldinhalte:

1 begrenzte Kapazität

2 unbegrenzte Kapazität



Bei begrenzter Kapazität werden die Arbeitsaufträge nacheinander eingeplant (links); bei unbegrenzter Kapazität können sie zeitgleich eingeplant werden (rechts).

Mit der Eigenschaft **ResourceCapacityType** können Sie den Kapazitätstyp für alle Datensätze einer Ressourcentabelle vorgeben. Diese Vorgabe wird durch die Einzelsetzung hier überschrieben.

Wenn eine Ressource mehreren Gruppen zugehörig ist, muss sie in jeder Gruppe den gleichen Kapazitätstyp besitzen.

	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index der Ressourcendatentabelle. {0...24}
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendaten-tabelle, das für die Aufnahme des Kapazitätstyps vorgesehen ist. {-1...AnzahlDatenfelderInRessourcendatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Ressourcendatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceCapacityTypeFieldIndex(0) = 1
```

ResourceConstraintTypeFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Ressourcentabelle, in dem eine Bedingung oder Beschränkung einer einzelnen Arbeits- und Materialressource enthalten ist.

Unter den 25 möglichen Ressourcentabellen wird die betroffene durch den als Parameter übergebenen Index adressiert.

Erlaubt sind als Typen die Inhalte 0,1 oder 3 oder ein Leerwert. Die Werte "" oder "1" oder wenn kein Feld vorhanden ist, bedeuten, dass die vorgegebene Kapazität der Resource bindend ist (sogen. "harte" Resource).

Der Wert "0" bedeutet, dass die vorgegebene Kapazität der Resource bei Bedarf ignoriert wird, da sie dann in unbegrenzter Kapazität zur Verfügung steht (sogen. "weiche" Resource).

Der Wert "3" bedeutet, dass die Resource "hart" ist, aber arbeitsfreie Zeiten berücksichtigt werden, die nicht als Unterbrechung bei der Einplanung der Operation gelten sollen.

	Datentyp	Beschreibung
Parameter: index	Short	Index der Ressourcendatentabelle {0...24}
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendatentabelle, das für die Aufnahme einer Bedingung vorgesehen ist. {-1...AnzahlDatenfelderInRessourcendatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Ressourcendatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceConstraintTypeFieldIndex(0) = 1
```

ResourceDataTableName

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie die Namen von bis zu 25 Ressourcentabellen setzen oder erfragen. Der Name der Tabelle mit dem Index 0 ist

obligat zu setzen. Die Indizes sind von 0 an aufwärts fortlaufend zu besetzen. Für jede hier angegebene Ressourcentabelle ist auch ein Feld in der Zuweisungstabelle vorzusehen und mit der Eigenschaft **Assignment-ResourceIDFieldIndex** bekannt zu geben.

Datentabellen verwalten

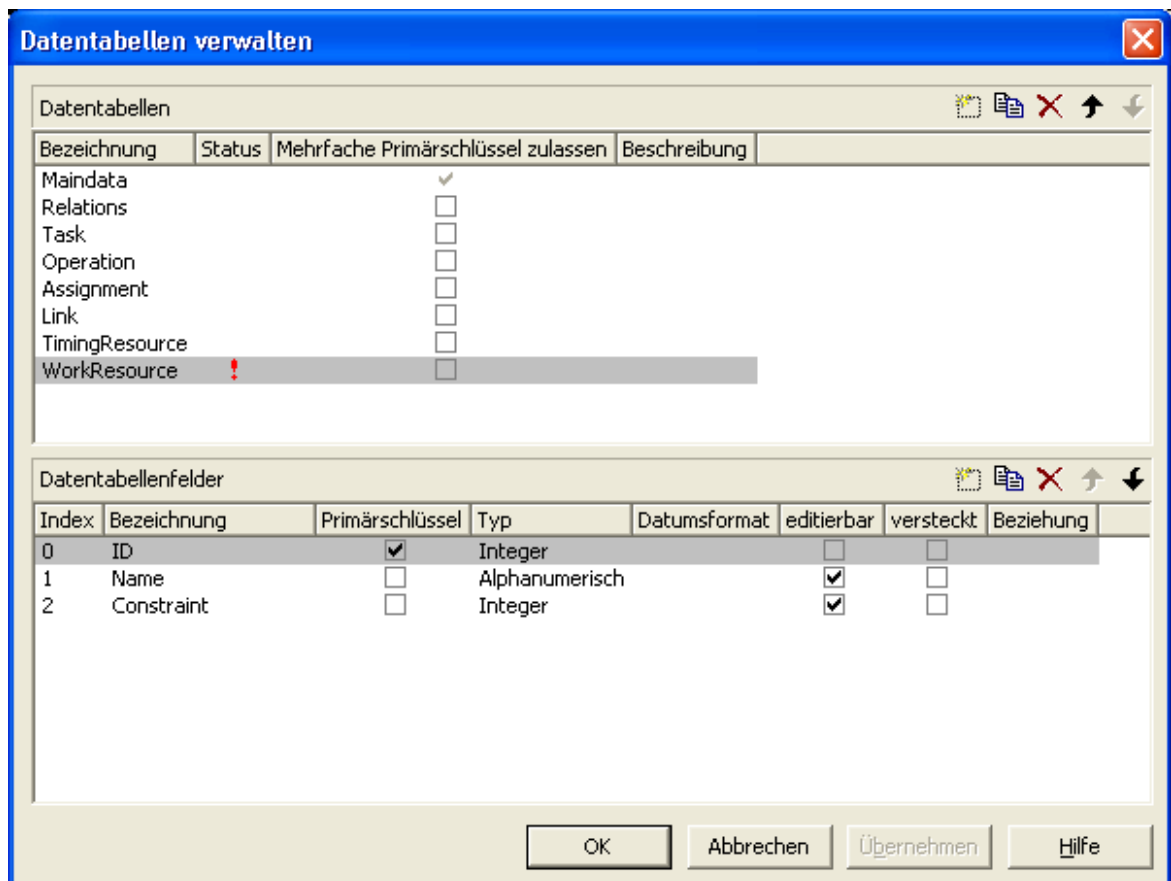
Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata		☒	
Relations		☐	
Task		☐	
Operation		☐	
Assignment		☐	
Link		☐	
TimingResource		☐	
WorkResource		☐	

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt	Beziehung
0	ID	☐	Integer		☐	☐	
1	Name	☐	Alphanumerisch		☒	☐	
2	CapacityType	☐	Integer		☒	☐	
3	ResourceGroupID	☐	Integer		☒	☐	

OK Abbrechen Übernehmen Hilfe



	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index der der Ressourcendatentabelle. {0...24}
Eigenschaftswert	String Mögliche Werte:	Name der Datentabelle Standardwert: Empty string Name der Farbzusordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceDataTableName(1) = "Timing Resource"
```

ResourceEfficiencyFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Ressourcentabelle, das für die jeweilige Ressource des Typs **TimingResource** eine Effizienz in Prozent angibt. Ist dieses Datenfeld an einer Ressource leer oder die Eigenschaft auf -1 gesetzt, dann ist die Effizienz standardmäßig 100. Wenn hingegen ein Wert angegeben wird, dann

wird die Gesamtsumme der Belegungen durch eine Zuweisung vor der Einplanung auf dieser Ressource mit dem gegebenen Effizienzwert multipliziert, d.h. bei einer Effizienz unter 100 Prozent dauert eine zugewiesene Operation auf dieser Ressource länger als im Default, während bei Werten über 100 Prozent eine zugewiesene Operation hier schneller abgearbeitet wird als im Default. Interessant ist dies besonders bei der Definition von Ressourcengruppen (s. auch **ResourceSelectionStrategy**), wobei dann die verfügbare Ressource mit der höchsten Effizienz bevorzugt werden kann.

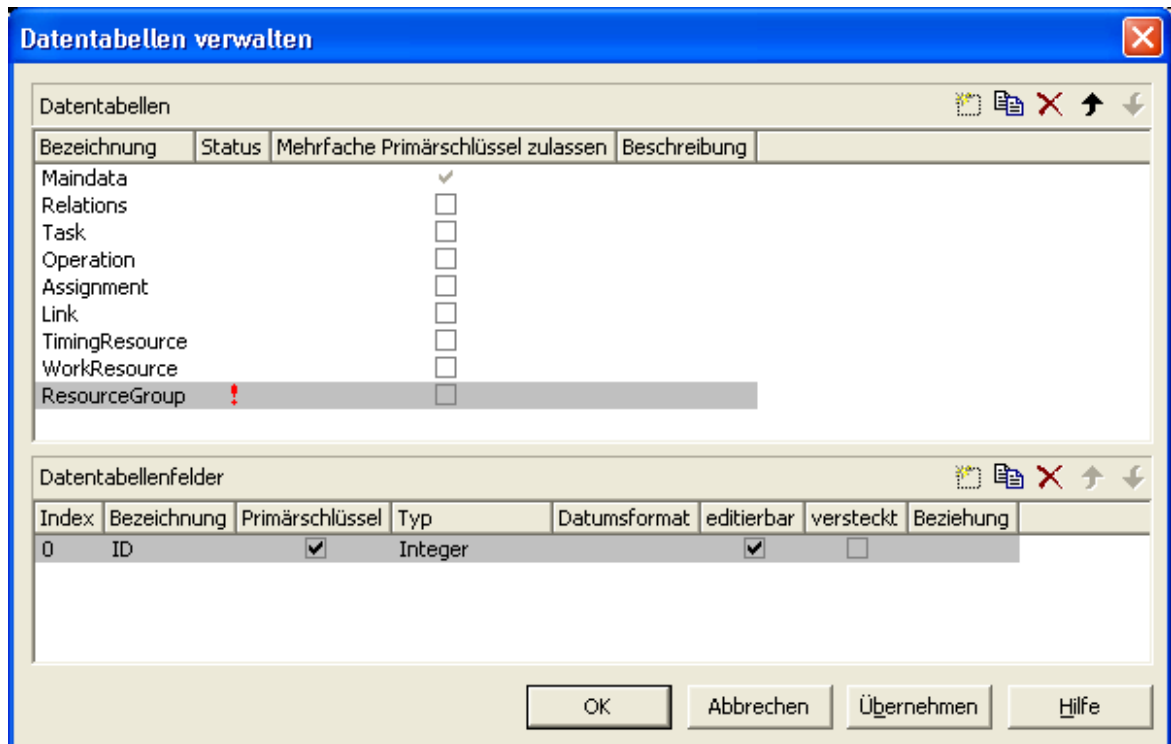
Üblicherweise bewegt sich die Effizienz im Bereich zwischen 1 und 100. Werte über 1.000 werden automatisch auf 1.000 begrenzt. Die Effizienz sollte nicht so hoch gesetzt werden, dass die Belegung einer Ressource durch eine Operation den Wert von 1 unterschreitet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendaten-tabelle, das die Effizienz einer Resource vom Typ Timing Resource in Prozent angibt.

ResourceGroupDataTableName

Eigenschaft von VcResourceScheduler2

Diese indizierte Eigenschaft bestimmt, in welcher Datentabelle die Ressourcengruppen gefunden werden, deren IDs in den Feldern enthalten sind, die durch die Eigenschaft **ResourceGroupIDFieldIndex** adressiert werden. Das heißt, für jeden Feldindex, den Sie über die Eigenschaft **ResourceGroupIDFieldIndex** ungleich -1 setzen, müssen Sie über diese Eigenschaft den Namen einer Datentabelle angeben, wobei die gleichen Datentabellen benutzt werden, die auch in der indizierten Eigenschaft **ResourceDataTableName** angegeben werden. Der als Parameter übergebene Ressourcentabellenindex bezeichnet eine von 25 möglichen Ressourcentabellen, die per indizierter Eigenschaft **ResourceDataTableName** zugewiesen werden.



	Datentyp	Beschreibung
Parameter: resourceGroupTableIndex	Short	Index der der Ressourcengruppendatentabelle. {0...24}
Eigenschaftswert	String Mögliche Werte:	Name einer Ressourcengruppendatentabelle Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceGroupDataTableName(1) = "Printer Resource"
```

ResourceGroupIDFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in einer Ressourcentabelle, das für die ID einer Gruppen-Ressource vorgesehen ist. Die Ressource wird durch das Setzen der ID in dem bezeichneten Feld als einer Gruppe zugehörig beschrieben. In der Abbildung zu **ResourceGroupDataTableName** ist der Index beispielsweise 0. Wird der Feldindex auf -1 gesetzt oder ist das entsprechende Datenfeld der Resource leer, gehört die Ressource keiner Gruppe an. Diese Eigenschaft darf nur bei einer zeitdauerbestimmenden Ressource gesetzt werden (s. Eigenschaft **ResourceType**).

Der als Parameter übergebene Ressourcentabellenindex bezeichnet eine von 25 möglichen Ressourcentabellen, die über die indizierte Eigenschaft **ResourceDataTableName** zugewiesen werden.

	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index der Ressourcendatentabelle {0...24}
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendatentabelle, das für die Aufnahme einer GruppenID vorgesehen ist. {-1...AnzahlDatenfelderInRessourcendatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Ressourcendatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceGroupIDFieldIndex(0) = 1
```

ResourceNameFieldIndex

Eigenschaft von VcResourceScheduler2

Der als Eigenschaftswert übergebene Index bezeichnet ein Datenfeld in der Ressourcentabelle, das für den Namen einer Ressource vorgesehen ist. In der Abbildung zu **ResourceDataTableName** ist der Index beispielsweise 1. Der Ressourcename dient der Ermittlung von Histogramm-, Kurven- und Kalendernamen. Außerdem wird er benutzt, um bei Verwendung von Ressourcengruppen die Möglichkeit zu erhalten, eine Ressource gleichzeitig mehreren Gruppen zuordnen zu können. Dazu muss der Ressourcename in mehreren Datensätzen identisch angegeben werden, aber mit verschiedenen IDs der Gruppen-Ressourcen. Wird kein Feldindex angegeben, wird für die Ermittlung von Histogramm-, Kurven- und Kalendernamen die Ressourcen-ID benutzt.

Der als Parameter übergebene Ressourcentabellenindex bezeichnet eine von 25 möglichen Ressourcentabellen, deren Verwendung über die indizierte Eigenschaft **ResourceDataTableName** gesetzt wird.

	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index der Ressourcendatentabelle. {0...24}
Eigenschaftswert	Long	Index des Datenfeldes in der Ressourcendaten-tabelle, das für die Aufnahme des Namens vorgesehen ist. {-1...AnzahlDatenfelderInRessourcendatentabelle - 1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Ressourcendatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceNameFieldIndex(0) = 1
```

ResourceResultLoadCurveNamePrefix**Eigenschaft von VcResourceScheduler2**

Präfix für den Namen einer Kurve, die nach der Ressourcenplanung die Auslastung je zeitdauerbestimmender Ressource und je Arbeitsressource enthalten soll.

Diese Kurven müssen vor dem Aufruf der Methode **Process** bereits definiert sein, damit sie im Histogramm dargestellt werden können. Für den Rest des Kurvennamens wird der Ressourcenname oder die Ressourcen-ID verwendet (s. Eigenschaft **ResourceNameFieldIndex**). Wird eine Kurve nicht gefunden, gehen die Ergebnisse der Auslastung für die jeweilige Ressource verloren.

Bei den Kurven muss außerdem die Eigenschaft **CurveSource** auf **vcSet-Curve** gesetzt sein, das heißt, sie müssen mittels **VcCurve.SetValues** über die API bestückbar sein.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die das Präfix enthält Standardwert: "Load_"
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceResultLoadCurveNamePrefix = "LoadCurve_"
```

ResourceResultStockCurveNamePrefix

Eigenschaft von VcResourceScheduler2

Präfix für den Namen einer Kurve, die nach der Ressourcenplanung den Lagerbestand je Materialressource enthalten soll. Diese Kurven müssen vor dem Aufruf der Methode **Process** bereits definiert sein, damit sie im Histogramm dargestellt werden können. Der Ressourcenname oder die Ressourcen-ID wird für den Rest des Namens verwendet.

Wird eine Kurve nicht gefunden, gehen die Ergebnisse des Lagerbestands für die jeweilige Ressource verloren. Der Lagerbestand ergibt sich aus der Kumulierung von Materialzuflüssen (also durch die Lieferkurve, die von der Applikation vor der Durchführung der Ressourcenplanung bestückt wird) und dem Verbrauch durch die der Ressource zugewiesenen Operationen über die Zeit.

Bei den Kurven muss außerdem die Eigenschaft **CurveSource** auf **vcSetCurve** gesetzt sein, das heißt, sie müssen mittels **VcCurve.SetValues** über die API bestückbar sein.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenkette, die das Präfix enthält Standardwert: "Stock_"
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceResultStockCurveNamePrefix = 1
```

ResourceSelectionStrategy

Eigenschaft von VcResourceScheduler2

Diese Eigenschaft beschreibt die Auswahlstrategie für Ressourcen, die von der Ressourcenplanung aus einer Gruppe ausgewählt werden (gilt daher nur für zeitdauerbestimmende Ressourcen).

	Datentyp	Beschreibung
Eigenschaftswert	VcResourceSchedulingResource- SelectionStrategyEnum	Auswahltypen Standardwert: VcResSchedRSSequential
	Mögliche Werte:	

vcResSchedRSFirstAvailable 6	Die zum Zeitpunkt der Planung einer Operation am frühesten verfügbare Ressource aus der Gruppe, die die benötigte freie Kapazität aufweist, wird ausgewählt. Bei Verwendung dieser Konstanten ist die Entscheidung nur von der zuerst verfügbaren zeitdauergebenden Ressource abhängig. Weitere Zuweisungen der gleichen Operation werden nicht berücksichtigt, so dass bei Verwendung von Material- und Arbeitsressourcen ein nicht zufriedenstellendes Ergebnis entstehen kann.
vcResSchedRSHighestEfficiency 2	Die zum Zeitpunkt der Planung einer Operation effizienteste Ressource aus der Gruppe, die die benötigte freie Kapazität aufweist, wird ausgewählt (nur sinnvoll, wenn die Eigenschaft ResourceEfficiencyFieldIndex genutzt wird).
vcResSchedRSLeastLoaded 0	Die zum Zeitpunkt der Planung einer Operation am geringsten ausgelastete Ressource aus der Gruppe, die die benötigte freie Kapazität aufweist, wird zuerst belegt. Diese Strategie ist sinnvoll, wenn Ressourcen möglichst gleichmäßig ausgelastet werden sollen. Bei Verwendung dieser Konstanten wird bei der Einplanung der Ressourcen eine laufende Summe der Belegungen aufaddiert, anhand derer die Entscheidung für die am geringsten belegte Ressource getroffen wird. Daher ist es nicht ausgeschlossen, dass bei stark differierenden Planungszeiträumen von Aufgaben oder bei Nutzung beider Planungsstrategien das Ergebnis nicht zufriedenstellend ausfällt.
vcResSchedRSMostLoaded 1	Die zum Zeitpunkt der Planung einer Operation am höchsten ausgelastete Ressource aus der Gruppe, die die benötigte freie Kapazität aufweist, wird zuerst belegt. Diese Strategie ist sinnvoll, wenn möglichst wenige Ressourcen möglichst hoch ausgelastet werden sollen. Bei Verwendung dieser Konstanten wird bei der Einplanung der Ressourcen eine laufende Summe der Belegungen aufaddiert, anhand derer die Entscheidung für die am höchsten belegte Ressource getroffen wird. Daher ist es nicht ausgeschlossen, dass bei stark differierenden Planungszeiträumen von Aufgaben oder bei Nutzung beider Planungsstrategien das Ergebnis nicht zufriedenstellend ausfällt.
vcResSchedRSSequential -1	Es wird versucht, die Ressourcen in vorgegebener Reihenfolge zu benutzen.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceSelectionStrategy = vcResSchedRSLeastLoaded
```

ResourceType

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie den Typ einer Ressourcentabelle festlegen oder erfragen. Der übergebene Index gibt an, von welcher der 25 möglichen Datentabellen der Typ festgelegt werden soll. Es gibt drei Ressourcentypen:

1. Zeitbestimmende Ressourcen

Eine Operation wird genau einer Ressource dieses Typs zugewiesen. Sie bestimmt die Verarbeitungsdauer der Operation. Als Kapazitätstypen sind finit und infinit erlaubt (s. Eigenschaft **ResourceCapacityTypeFieldIndex**). Ressourcen dieses Typs können gruppiert werden (s. Eigenschaften **ResourceGroupDataTableName** und **ResourceGroupIDFieldIndex**). Zudem ist eine Beschränkung der Belegung möglich (s. Eigenschaften **AssignmentMinimumLoadFieldIndex** und **AssignmentMaximumLoadFieldIndex**). Die zeitbestimmende Ressource benötigt Kapazitätskurven als indirekte Ressourceninformation und verwendet Belegungskurven als Ergebnis (s. Eigenschaften **ResourceNameFieldIndex** und **ResourceResultLoadCurvePrefix**).

2. Arbeits-Ressourcen

Dieser Ressourcentyp ist durch zwei Merkmale gekennzeichnet. Zum einen kann eine Operation mehreren Ressourcen dieses Typs zugewiesen werden. Genau wie die zeitbestimmende Ressource benötigt auch die Arbeits-Ressource Kapazitätskurven als indirekte Ressourceninformation und verwendet Belegungskurven als Ergebnis (s. Eigenschaften **ResourceNameFieldIndex** und **ResourceResultLoadCurvePrefix**).

3. Materialressourcen

Auch die Materialressource ist durch zwei Merkmale gekennzeichnet. Eine Operation kann mehreren Ressourcen dieses Typs zugewiesen werden. Anders als die anderen beiden Ressourcentypen benötigt die Materialressource Lieferkurven als indirekte Ressourceninformation und verwendet Lagerbestandskurven als Ergebnis (s. Eigenschaften **ResourceNameFieldIndex** und **ResourceResultStockCurvePrefix**).

	Datentyp	Beschreibung
Parameter: resourceTableIndex	Short	Index der Ressourcendatentabelle. {0...24}.
Eigenschaftswert	VcResourceSchedulingResource- TypeEnum Mögliche Werte: vcResSchedMaterial -1 vcResSchedTiming 1 vcResSchedWork 0	Typ der Ressourcendatentabelle Standardwert: vcTiming Der Ressourcentyp ist "Material". Der Ressourcentyp ist "zeitdauerbestimmend". Der Ressourcentyp ist "Arbeit".

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ResourceType(0) = vcResSchedTiming
```

ResultProcessingStepCount**Eigenschaft von VcResourceScheduler2**

Nach einem Berechnungsdurchlauf kann mit dieser Eigenschaft die Anzahl der im Durchlauf geplanten Tasks erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Tasks Standardwert: 0

Code-Beispiel

```
Dim i As Integer  
i = VcGantt1.ResourceScheduler2.ResultProcessingStepCount
```

TaskDataTableName**Eigenschaft von VcResourceScheduler2**

Mit dieser Eigenschaft können Sie den Namen der Aufgabentabelle setzen oder erfragen. Die Eigenschaft muss mit einem gültigen Tabellennamen besetzt werden.

Datentabellen verwalten

Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata	☒		
Relations	☐		
Task	☐		
Operation	☐		
Assignment	☐		
Link	☐		
TimingResource	☐		
WorkResource	☐		
ResourceGroup	☐		

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt	Beziehung
0	ID	☒	Integer		☒	☐	
1	Name	☐	Alphanumerisch		☒	☐	
2	Release date	☐	Datum/Zeit	DD/MM/YY	☒	☐	
3	Due date	☐	Datum/Zeit	DD/MM/YY	☒	☐	
4	Quantity	☐	Integer		☒	☐	
5	Priority	☐	Integer		☒	☐	
6	PlanningStrategy	☐	Integer		☒	☐	
7	ResultStartDate	☐	Datum/Zeit	DD/MM/YY	☒	☐	
8	ResultEndDate	☐	Datum/Zeit	DD/MM/YY	☒	☐	
9	ResultRoute	☐	Integer		☒	☐	
10	ResultProcessingTime	☐	Integer		☒	☐	
11	ResultProcessingStep	☐	Integer		☒	☐	

OK Abbrechen Übernehmen Hilfe

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Aufgabendatentabelle
	Mögliche Werte:	Standardwert: Empty string Name der Farbzuoordnungstabelle

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskDataTableName = "Task"
```

TaskDueDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem das Fälligkeitsdatum steht, an dem ein Arbeitsauftrag fertiggestellt sein muss. Falls kein gültiger Wert in dem Datenfeld gefunden wird, wird der von der Eigenschaft **TimeScaleEnd** im VcGantt-Objekt gesetzte Wert angenommen. Falls der entsprechende Arbeitsauftrag eingeplant werden soll, darf diese Eigenschaft während der Ressourcenplanung nicht auf -1 gesetzt sein. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 3.

Fälligkeitsdaten können über die Eigenschaft **ToleranceTimeOnASAPDueDates** pauschal mit einer Toleranz versehen werden. Bitte beachten Sie, dass Aufgaben mit vergleichsweise nahem Fälligkeitsdatum oder kurzem Zeitraum zwischen Freigabe- und Fälligkeitsdatum nicht automatisch bevorzugt eingeplant werden. Wenn Sie dies wünschen, dann müssen Sie die Prioritäten der Aufgaben selbst berechnen (s. Eigenschaft **TaskPriorityFieldIndex**).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme des Fälligkeitsdatums vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskDueDateFieldIndex = 3
```

TaskPlanningStrategyFieldIndex

Eigenschaft von VcResourceScheduler2

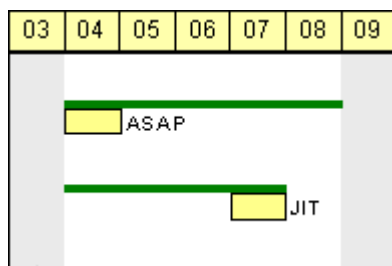
Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem individuell eine Planungsstrategie für eine Aufgabe gesetzt werden kann.

Falls ein Wert nicht gesetzt oder < 1 oder > 2 gesetzt wird, wird der gesetzte Wert aus der Eigenschaft **PlanningStrategy** verwendet. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 6.

Definierte Werte der Datenfeldinhalte {1...2}:

1 - so bald wie möglich (ASAP: as soon as possible)

2 - gerade rechtzeitig (JIT: just in time)



Bei der ASAP-Strategie wird die Aufgabe möglichst früh, bei der JIT-Strategie möglichst spät eingeplant. Die langen, linienförmigen Balken zeigen den möglichen Ausführungszeitraum der Aufgabe an; die kurzen, rechteckigen stellen die tatsächlich eingeplante Ausführungszeit dar. ASAP-Aufgaben erscheinen daher nach Möglichkeit links, JIT-Aufgaben rechts innerhalb des möglichen Ausführungszeitraums.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme der Daten zur Planungsstrategie vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

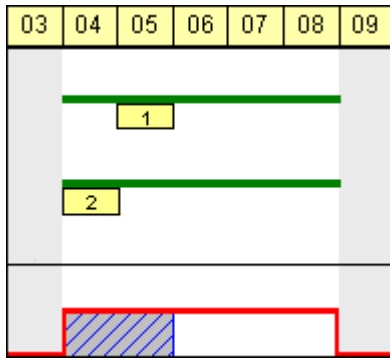
Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskPlanningStrategyFieldIndex = 6
```

TaskPriorityFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem für die Aufgabe eine Priorität hinterlegt werden kann.

Je höher der Wert für die Priorität ist, desto eher wird die Aufgabe in der Reihenfolge des Planungsprozesses berücksichtigt.



Eine Aufgabe mit höherer Priorität (2) wird vor einer Aufgabe mit niedrigerer Priorität (1) eingeplant.

Zur Beachtung: Bei über Verbindungen verknüpften Aufgaben sollte die Priorität mit Bedacht gewählt werden, d.h. bei ASAP sollten üblicherweise Vorgänger maximal die gleiche Priorität haben wie ihre Nachfolger, bei JIT sollten Vorgänger mindestens die gleiche Priorität haben wie ihre Nachfolger. Über die Priorität lassen sich Aufgaben gruppieren. So kann beispielsweise auf einer Maschine bei Gruppierung gleichartiger Tasks auf Vorbereitung und Reinigung zwischen diesen gleichartigen Tasks verzichtet werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme der Priorität vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskPriorityFieldIndex = 5
```

TaskQuantityFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem die abzuarbeitende Arbeitsmenge für die Aufgabe steht. Diese Eigenschaft muss auf einen Wert ungleich -1 gesetzt sein.

Die Arbeitsmenge bestimmt indirekt die Dauer, die die Aufgabe zu ihrer Fertigstellung benötigt. Dabei kann die Dauer auch durch die Effizienz bei den Ressourcen (s. **ResourceEfficiencyFieldIndex**) sowie durch Faktoren

bei Operationen (s. **OperationLoadPerItemFieldIndex**) und bei Zuweisungen (s. **AssignmentLoadOrConsumptionPerItemFieldIndex**) beeinflusst werden.

Falls kein gültiger Wert in dem Datenfeld gefunden wird, wird eine Arbeitsmenge von 1 angenommen. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 4.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme der Auftragsmenge vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskQuantityFieldIndex = 4
```

TaskReleaseDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem das Freigabedatum steht, ab dem ein Arbeitsauftrag eingeplant werden darf. Diese Eigenschaft muss auf einen Wert ungleich -1 gesetzt sein

Falls kein gültiger Wert in dem Datenfeld gefunden wird, wird der von der Eigenschaft **TimeScaleStart** im VcGantt-Objekt gesetzte Wert angenommen. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 2.

Freigeabedaten können über die Eigenschaft **ToleranceTimeOnJITReleaseDates** pauschal mit einer Toleranz versehen werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme des Freigabedatums vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskReleaseDateFieldIndex = 2
```

TaskResultEndDateFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem das ermittelte Enddatum der spätesten zugehörigen, eingeplanten Operation abgelegt wird. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 8.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabendatentabelle, das für die Aufnahme des Enddatums vorgesehen ist. {-1...AnzahlDatenfelderInAufgabendatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabendatentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultEndDateFieldIndex = 8
```

TaskResultPostEndDateFieldIndex**Eigenschaft von VcResourceScheduler2**

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in das nach einer erfolgten Planung die Endezeit der Nachlaufphase einer Aufgabe abgelegt wird. Wenn die Nachlaufzeit 0 ist, dann ist dieses Datum identisch mit dem Wert in dem Datenfeld, das über die Eigenschaft **TaskResultEndDateFieldIndex** adressiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabendatentabelle, das für die Aufnahme des Enddatums der Nachlaufphase vorgesehen ist. {-1...AnzahlDatenfelderInAufgabendatentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabendatentabelle zugeordnet ist.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultPostEndDateFieldIndex = 15
```

TaskResultPreparationStartDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in das nach einer erfolgten Planung die Startzeit der Vorbereitungsphase einer Aufgabe abgelegt wird. Wenn die Vorbereitungszeit 0 ist, dann ist dieses Datum identisch mit dem Wert in dem Datenfeld, das über die Eigenschaft **TaskResultStartDateFieldIndex** adressiert wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme des Startdatums der Vorbereitungsphase vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultPreparationStartDateFieldIndex = 10
```

TaskResultProcessingStepFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem die beim vorangegangenen Planungsdurchlauf festgelegte Abfolgenummer der Aufgabe steht. Dieser Wert ist hilfreich um zu erkennen, welche Aufgabe als erste aufgrund von Ressourcenengpässen nicht eingeplant werden konnte.

Die als erste eingeplante Aufgabe erhält die Nummer 0, die folgenden Aufgaben fortlaufend aufsteigende Nummern. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 11.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme der Abfolgenummer vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultProcessingStepFieldIndex = 11
```

TaskResultProcessingTimeFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem die beim vorangegangenen Planungsprozess ermittelte Gesamtverarbeitungszeit aller zu der Aufgabe gehörigen, eingeplanten Operationen abgelegt wird. Es ist die Zeitspanne zwischen dem Startdatum der ersten Operation und dem Enddatum der letzten Operation. Die Angabe erfolgt in der eingestellten BaseTimeUnit. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 10.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme der Verarbeitungszeit vorgesehen ist. {1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultProcessingTimeFieldIndex = 10
```

TaskResultRouteFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem der Name einer Route steht, die während der Planung für die Aufgabe durch die Ressourcenplanung ausgewählt wurde.

Diese Eigenschaft sollte nicht auf -1 gesetzt sein, wenn die Eigenschaft **OperationRouteFieldIndex** verwendet wird. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 9.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme des Namens der Route vorgesehen ist. {1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultRouteFieldIndex = 9
```


TaskResultStartDateFieldIndex

Eigenschaft von VcResourceScheduler2

Der Index bezeichnet ein Datenfeld in der Aufgabentabelle, in dem das ermittelte Startdatum der frühesten zugehörigen, eingeplanten Operation abgelegt wird. In der Abbildung zu **TaskDataTableName** ist der Index beispielsweise 7.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes in der Aufgabentabelle, das für die Aufnahme des Startdatums vorgesehen ist. {-1...AnzahlDatenfelderInAufgabentabelle -1}. Der Wert -1 bedeutet, dass dieser Eigenschaft kein Datenfeld der Aufgabentabelle zugeordnet ist. Standardwert: -1

Code-Beispiel

```
VcGantt1.ResourceScheduler2.TaskResultStartDateFieldIndex = 7
```

ToleranceTimeOnASAPDueDates

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie eine Toleranz in Basiszeiteinheiten (s. Eigenschaft **BaseTimeUnit**) setzen oder erfragen, die sich auf die Fälligkeitsdaten von Aufträgen bezieht und nur bei der Planungsstrategie "As Soon As Possible" (ASAP) wirksam ist.

Beim Planungsprozess werden die Fälligkeitsdaten der Aufgaben um die angegebene Anzahl von Einheiten später gesetzt, womit sich der erlaubte Planungszeitraum für die Aufgabe verlängert. Diese Eigenschaft ist nützlich, um z.B. herauszufinden, ob bei Verlängerung des Planungszeitraums alle Operationen der Aufgaben eingeplant werden können und erspart, die Fälligkeitsdaten der Aufgaben individuell zu ändern.

Sehen Sie bitte auch **ToleranceTimeOnJITReleaseDates**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Basiszeiteinheiten {>=0} Standardwert: 0

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ToleranceTimeOnASAPDueDates = 1
```

ToleranceTimeOnJITReleaseDates

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie eine Toleranz in Basiszeiteinheiten (s. Eigenschaft **BaseTimeUnit**) setzen oder erfragen, die sich auf die Freigabedaten von Aufträgen bezieht und nur bei der Planungsstrategie "Just In Time" (JIT) wirksam ist.

Beim Planungsprozess werden die Freigabedaten der Aufgaben um die angegebene Anzahl von Einheiten früher gesetzt, womit sich der erlaubte Planungszeitraum für die Aufgabe verlängert. Diese Eigenschaft ist nützlich, um z.B. herauszufinden, ob bei Verlängerung des Planungszeitraums alle Operationen der Aufgaben eingeplant werden können und erspart, die Freigabedaten der Aufgaben individuell zu ändern.

Sehen Sie bitte auch **ToleranceTimeOnASAPDueDates**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Basiszeiteinheiten {>=0} Standardwert: 0

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ToleranceTimeOnJITReleaseDates = 1
```

ToleranceTimeOnStartLockDates

Eigenschaft von VcResourceScheduler2

Mit dieser Eigenschaft können Sie eine Toleranz in Basiszeiteinheiten (s. Eigenschaft **BaseTimeUnit**) für ein festes Startdatum einer Operation (s. **OperationStartLockDateFieldIndex**) setzen oder erfragen. Eine Operation kann dadurch um die spezifizierte Anzahl der Einheiten später eingeplant werden als vorgegeben, wenn die zu belegenden Ressourcen zum festgelegten Startdatum nicht verfügbar sind.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Basiszeiteinheiten {>=0} Standardwert: 0

Code-Beispiel

```
VcGantt1.ResourceScheduler2.ToleranceTimeOnStartLockDates = 1
```

WorkInProcessType

Eigenschaft von VcResourceScheduler2

Diese Eigenschaft setzt die Einheit, in der der Fertigstellungsgrad angegeben wird (s. **OperationWorkInProcessFieldIndex**).

	Datentyp	Beschreibung
Eigenschaftswert	ResourceSchedulingWorkInProcessTypeEnum Mögliche Werte: vcResSchedWIPCompleted 0 vcResSchedWIPPercentage -1 vcResSchedWIPRemaining 1	Einheit des Fertigstellungsgrades Standardwert: vcResSchedWIPPercentage Einheit in fertiggestellten Arbeitsmengen Einheit in Prozentzahlen (0...100) Einheit in noch zu fertigenden Arbeitsmengen

Code-Beispiel

```
VcGantt1.ResourceScheduler2.WorkInProcessType = vcResSchedWIPCompleted
```

WritingDebugFilesEnabled

Eigenschaft von VcResourceScheduler2

Wenn diese Eigenschaft auf **True** gesetzt wird, werden in das augenblicklich für die Applikation aktuelle Verzeichnis Debug-Dateien geschrieben, die für Analysezwecke wichtig sein können.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	True: Debug-Dateien können in das aktuelle Verzeichnis geschrieben werden. False: Debug-Dateien können nicht in das aktuelle Verzeichnis geschrieben werden. Standardwert: False Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.ResourceScheduler2.WritingDebugFilesEnabled = True
```

Methoden

DetermineIDOfFirstOperationByTaskID

Methode von VcResourceScheduler2

Diese Methode legt über die angegebene TaskID die ID der ersten Operation einer Aufgabe (Task) fest und hilft so dem Entwickler bei der Aktualisierung des Datenfelds einer Verbindung, das die ID der ersten Operation einer Aufgabe enthält.

Weitere Informationen dazu finden Sie auch bei der Beschreibung der beiden VcResourceScheduler-Eigenschaften **LinkPredecessorOperationIDFieldIndex** und **LinkSuccessorOperationIDFieldIndex**.

	Datentyp	Beschreibung
Parameter: ⇒ taskID	String Mögliche Werte:	ID einer Aufgabe aus der entsprechenden Datentabelle, die zuvor über die VcResourceScheduler2-Eigenschaft TaskDataTableName gesetzt wurde. Name der Farbzuoordnungstabelle
Rückgabewert	String	ID der ersten Operation aus der entsprechenden Datentabelle, die zuvor über die VcResourceScheduler2-Eigenschaft OperationDataTableName gesetzt wurde.

DetermineIDOfLastOperationByTaskID

Methode von VcResourceScheduler2

Diese Methode legt über die angegebene TaskID die ID der letzten Operation einer Aufgabe (Task) fest und hilft so dem Entwickler bei der Aktualisierung des Datenfelds einer Verbindung, das die ID der letzten Operation einer Aufgabe enthält.

Weitere Informationen dazu finden Sie auch bei der Beschreibung der beiden VcResourceScheduler-Eigenschaften **LinkPredecessorOperationIDFieldIndex** und **LinkSuccessorOperationIDFieldIndex**.

	Datentyp	Beschreibung
Parameter: taskID	String Mögliche Werte:	ID einer Aufgabe aus der entsprechenden Datentabelle, die zuvor über die VcResourceScheduler2-Eigenschaft TaskDataTableName gesetzt wurde. Name der Farbzuoordnungstabelle
Rückgabewert	String	ID der letzten Operation aus der entsprechenden Datentabelle, die zuvor über die VcResourceScheduler2-Eigenschaft OperationDataTableName gesetzt wurde.

Process

Methode von VcResourceScheduler2

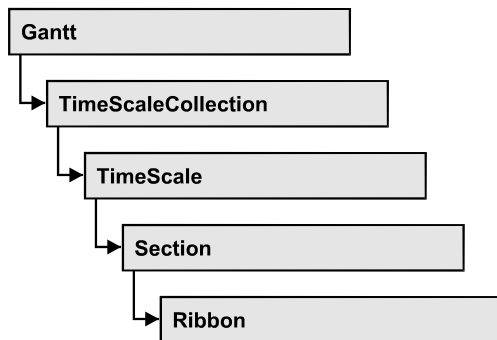
Mit dieser Methode können Sie die Ressourcenplanung starten, nachdem Sie alle gewünschten Eigenschaften gesetzt haben. Für Fortschrittsmeldungen s. **OnResourceSchedulingProgress**. Zudem werden Warnungen über **ResourceSchedulingWarning** ausgegeben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	True: Die Ressourcenplanung wurde fehlerfrei durchgeführt. False: Die Ressourcenplanung wurde entweder nicht fehlerfrei durchgeführt oder abgebrochen. Fehler sind - falls so eingestellt - in dem Datenfeld, das über die Eigenschaft OperationResultStatus-FieldIndex adressiert wird, für jeden Arbeitsauftrag erfragbar.

Code-Beispiel

```
VcGantt1.ResourceScheduler2.Process
```

7.71 VcRibbon



Ein Objekt des Typs VcRibbon stellt einen definierten Zeitstreifen in einer Zeitskala mit gleichen Einheiten und einheitlicher Skalierung dar. Bei diesem Objekt können Eigenschaften wie Hintergrundfarbe, Trennlinien, Schrifttyp, -farbe, -größe, -ausrichtung u.a. eingestellt werden.

Eigenschaften

- CalendarName
- DateOutputFormat
- Font
- FontColor
- MajorTicks
- MinorTicks
- ObserveDST
- PatternBackgroundColorAsARGB
- PatternColorAsARGB
- PatternEx
- Position
- ReferenceDate
- TextAlignment
- TickColor
- TickPosition
- Type
- UnitSeparation
- UseReferenceDate

Eigenschaften

CalendarName

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie den Kalendernamen erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Kalendername
	Mögliche Werte:	Name der Farbzuoordnungstabelle

DateOutputFormat

Eigenschaft von VcRibbon

Mit dieser Eigenschaft stellen Sie das Ausgabeformat von Terminen für den Zeitstreifen ein. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

- D: Wochentagsname erster Buchstabe
- TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- DD: Tagesnummer zweistellig: 01-31 (nicht anpassbar)
- DDD: die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)
- M: erster Buchstabe des Monatsnamens (nicht anpassbar)
- TM: Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- MM: zweistellige Monatsnummer: 01-12
- MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)
- YY: zweistellige Jahreszahl
- YYYY: vierstellige Jahreszahl
- WW: zweistellige Nummer der Kalenderwoche: 01-53
- TW: Text für "Kalenderwoche" (kann über das Ereignis

OnSupplyTextEntry angepasst werden)

- Q: einstellige Quartalsnummer: 1-4
- TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- hh: Stunde zweistellig im 24-Stunden-Format: 00-23
- HH: Stunde zweistellig im 12-Stunden-Format: 01-12
- Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)
- mm: Minute zweistellig: 00-59
- ss: Sekunde zweistellig: 00-59
- TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert
- xC/XC: Sie können das Datum als maximal zehngliedrige, einfache Aufwärtszählung ab einem bestimmten Referenzdatum gestalten, z.B. "15:05:07:16:00". Dieses Beispiel datiert 15 Monate, 5 Tage, 7 Stunden, 16 Minuten und 0 Sekunden später als das gesetzte Referenzdatum. Die Notation dazu ist: **xC44:C33:C22:C11:C00**. Es sollen je 2 Stellen für die Glieder 4...0 vorgesehen werden, mit einem vorausgehenden "-" Symbol falls der Wert negativ ist. Die Trennzeichen sind variabel und könnten durch andere ersetzt werden. "x" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, aber kein "+"Symbol, falls er positiv ist. "X" bedeutet: Es soll ein vorausgehendes -Symbol gesetzt werden, falls der Wert negativ ist, und ebenso ein "+"Symbol, falls er positiv ist. Im Dialog **Zeitskalenabschnitt ... bearbeiten** sollten die Felder **Referenzdatum verwenden** und **Hauptmarkierung am Referenzdatum ausrichten** aktiviert sein, ebenso sollte **Serielle Beschriftungen** auf **Keine** gesetzt sein. Das Referenzdatum wird zur Laufzeit über die Methode **VcRibbon.setReferenceDate** gesetzt und überschreibt die Werte des Dialogs.

Hinweis: Bitte setzen Sie allen Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: :, /, - und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat {DMYhms:; /} oder {01234CXx}
	Mögliche Werte:	Name der Farbzustimmungstabelle

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon(0, 0)
ribbon.DateOutputFormat = "DD.MMM.YYYY hh:mm:ss"
```

Font

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Schriftattribute des Zeitstreifens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute des Zeitstreifens

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon
Dim newFont As New StdFont

newFont.Name = "Times New Roman"
newFont.Italic = True
newFont.Bold = True
newFont.Size = 12

Set timeScale = VcGantt1.timeScaleCollection.Active
Set ribbon = timeScale.ribbon(0, 0)
Set ribbon.Font = newFont
```

FontColor

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Schriftfarbe des Zeitstreifens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon(0, 0)
ribbon.FontColor = RGB(240, 130, 220)
```

MajorTicks

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Zeiteinheiten eine Hauptmarkierung erscheint. Die Zeiteinheit hängt vom Typ des verwendeten Zeitstreifens ab. Die Hauptmarkierungen werden beschriftet, wenn der Platz dafür ausreicht. Diese Eigenschaft können Sie auch im Dialog **Zeitskalenabschnitt bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Einheiten zwischen zwei Hauptmarkierungen
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon (0, 1)
ribbon.MajorTicks = 7
```

MinorTicks

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie erfragen oder festlegen, nach wie vielen Zeiteinheiten eine Nebenmarkierung erscheint. Die Zeiteinheit hängt vom

Typ des verwendeten Zeitstreifens ab. Die Nebenmarkierungen werden nicht beschriftet. Diese Eigenschaft können Sie auch im Dialog **Zeitskalenabschnitt bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Anzahl Einheiten zwischen zwei Nebenmarkierungen Datenfeldindex

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon (0, 1)
ribbon.MinorTicks = 1
```

ObserveDST

Nur-Lese-Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob für diesen Zeitstreifen die Sommerzeit in der Darstellung berücksichtigt wird oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	RibbonObserveDSTEnum Mögliche Werte: vcGODDefault 9999 vcRODNo 0 vcRODYes 1	Sommerzeit wird/wird nicht berücksichtigt. Standardeinstellung aus der .INI-Datei wird verwendet Sommerzeit wird nicht berücksichtigt Sommerzeit wird berücksichtigt

PatternBackgroundColorAsARGB

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Zeitstreifens erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil (ARGB-Wert) im Zahlenbereich von 0..255. Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon(0, 0)
ribbon.PatternBackgroundColorAsARGB = &h88FF0A06
```

PatternColorAsARGB**Eigenschaft von VcRibbon**

Mit dieser Eigenschaft können Sie die Musterfarbe des Zeitstreifens erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Hintergrundfarbe des Zeitstreifens Datenfeldindex

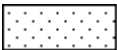
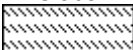
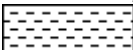
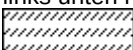
Code-Beispiel

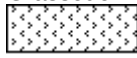
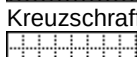
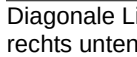
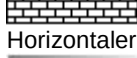
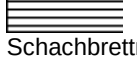
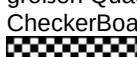
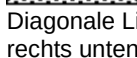
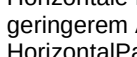
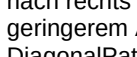
```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon(0, 0)
ribbon.PatternColorAsARGB = &h88FF0A06
```

PatternEx**Eigenschaft von VcRibbon**

Mit dieser Eigenschaft können Sie das Muster des Zeitstreifens erfragen oder festlegen.

Eigenschaftswert	Datentyp	Beschreibung
	FillPatternEnum	Mustertyp
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
	vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
	vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
	vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
	vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
	vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
	vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 

vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmall-CheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vc-HorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vc-HorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern 
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 

vcShinglePattern 2039	Diagonales Dachschildel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel
vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

Position

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Position des Zeitstreifens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	RibbonPositionEnum Mögliche Werte: vcRPBottom 2 vcRPNone 0 vcRPTop 1	Position des Zeitstreifens unten keine oben

ReferenceDate

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie das Referenzdatum erfragen oder festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Boolean Mögliche Werte:	Zeitstreifen verwendet (True) / verwendet kein (False) Referenzdatum Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar
Eigenschaftswert	Date	Referenzdatum

TextAlignment

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Ausrichtung der Beschriftung der Hauptmarkierungen des Zeitstreifens festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	HorRibbonTextAlignmentEnum Mögliche Werte: vcRTATickAligned 1039 vcRTHorCenterAligned -1	Art der Textausrichtung Text an der Hauptmarkierung ausgerichtet Text horizontal mittig zwischen zwei Hauptmarkierungen ausgerichtet

1364 API-Referenz: VcRibbon

vcRTLeftAligned -3
vcRTRightAligned -2

Text linksbündig zwischen zwei Hauptmarkierungen ausgerichtet
Text rechtsbündig zwischen zwei Hauptmarkierungen ausgerichtet

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon (0, 0)
ribbon.TextAlignment = vcRTLeftAligned
```

TickColor

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Farbe der Markierungen erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color RGB ({0...255},{0...255},{0...255})	RGB-Farbwerte Standardwert: 0,0,0

TickPosition

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie die Position der Markierungen erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	RibbonTickPositionEnum Mögliche Werte: vcTPAbove 1044 vcTPBelow 1045	Position der Markierungen oberhalb unterhalb

Type

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie den Typ des Zeitstreifens festlegen oder erfragen. Zur Verfügung stehen die unten aufgeführten Einheitenstreifen.

	Datentyp	Beschreibung
Eigenschaftswert	RibbonTypeEnum	Typ des Zeitstreifens
	Mögliche Werte: vcDayRibbon 5 vcFiscalQuarterRibbon 3002 vcFiscalYearRibbon 3001 vcHourRibbon 6 vcMinuteRibbon 7 vcMonthRibbon 3 vcQuarterRibbon 10 vcSecondRibbon 9 vcShiftRibbon 8 vcWeekRibbon 4 vcYearRibbon 1	Zeitstreifen mit Einheit Tag Zeitstreifen mit Einheit Fiskalisches Quartal Zeitstreifen mit Einheit Fiskalisches Jahr Zeitstreifen mit Einheit Stunde Zeitstreifen mit Einheit Minute Zeitstreifen mit Einheit Monat Zeitstreifen mit Einheit Quartal Zeitstreifen mit Einheit Sekunde Zeitstreifen mit Schichten als Einheit Zeitstreifen mit Einheit Woche Zeitstreifen mit Einheit Jahr

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon(0, 0)
ribbon.Type = vcWeekRibbon
```

UnitSeparation**Eigenschaft von VcRibbon**

Mit dieser Eigenschaft können Sie das Aussehen der Hauptmarkierungen (Major Ticks) des Zeitstreifens festlegen oder erfragen. Zur Verfügung stehen Vollstrich, Teilstrich und kein Strich.

	Datentyp	Beschreibung
Eigenschaftswert	UnitSeparationEnum	Aussehen der Hauptmarkierung
	Mögliche Werte: vcUSFullLine 4 vcUSNone 1 vcUSTick 1035	Trennstrich durchgezogen kein Trennstrich Teiltrennstrich

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon (0, 1)
ribbon.UnitSeparation = vcUSTick
```

UseReferenceDate

Eigenschaft von VcRibbon

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob der Zeitstreifen ein Referenzdatum verwendet.

	Datentyp	Beschreibung

7.72 VcScheduler

Scheduler

Ein Objekt vom Typ **VcScheduler** bezeichnet ein Rechenmodul, mit dem Sie einfache Projektdaten wie frühestmögliches Ende, frühestmöglicher Anfang (bei Rückwärtsrechnung), freie oder Gesamt-Pufferzeit eines Projektes berechnen können.

Eigenschaften

- ActualEndDateDataFieldIndex
- ActualStartDateDataFieldIndex
- AutomaticSchedulingEnabled
- DurationDataFieldIndex
- EarlyEndDateDataFieldIndex
- EarlyStartDateDataFieldIndex
- EndDateForAutomaticScheduling
- EndDateNotLaterThanDataFieldIndex
- FreeFloatDataFieldIndex
- LateEndDateDataFieldIndex
- LateStartDateDataFieldIndex
- LinkDurationDataFieldIndex
- ScheduledProjectEndDate
- ScheduledProjectStartDate
- ScheduleSuccessorsOnlyEnabled
- StartDateForAutomaticScheduling
- StartDateNotEarlierThanDataFieldIndex
- TotalFloatDataFieldIndex

Methoden

- ScheduleProject

Eigenschaften

ActualEndDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das aktuelle Enddatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das aktuelle Enddatum enthält

ActualStartDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das aktuelle Anfangsdatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das aktuelle Startdatum enthält

AutomaticSchedulingEnabled

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die Zeitberechnung automatisch erfolgt.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Automatische Zeitberechnung Ist an- (True) oder abgeschaltet (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

DurationDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das die Dauer des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Vorgangsdauer enthält

EarlyEndDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das berechnete früheste Enddatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das frühestmögliche Enddatum eines Vorgangs enthält

EarlyStartDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das berechnete früheste Anfangsdatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das frühestmögliche Startdatum eines Vorgangs enthält

EndDateForAutomaticScheduling

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie für den Fall, dass die automatische Zeitberechnung durchgeführt wird, das Enddatum des Projektes setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Gewünschtes Enddatum für automatische Zeitrechnung

EndDateNotLaterThanDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das gewünschte späteste Enddatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das gewünschte späteste Enddatum enthält

FreeFloatDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das den berechneten freien Puffer des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den freien Puffer enthält

LateEndDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das berechnete späteste Enddatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das spätestmögliche Enddatum enthält

LateStartDateDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das das berechnete späteste Anfangsdatum des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das spätestmögliche Startdatum enthält

LinkDurationDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, in dem ein minimaler zeitlicher Abstand zwischen Vorgänger und Nachfolger abgelegt werden kann. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den minimalen zeitlichen Abstand zwischen Vorgänger und Nachfolger enthält

ScheduledProjectEndDate

Nur-Lese-Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie, nachdem mit der Methode **VcScheduler.-ScheduleProject** die Projektdaten berechnet wurden, das **Früheste Ende** des Projektes erfragen, wenn bei **VcScheduler.ScheduleProject** ein Anfangsdatum vorgegeben wurde.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Index des Datenfeldes, das das geplante Enddatum des Projektes enthält

ScheduledProjectStartDate

Nur-Lese-Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie, nachdem mit der Methode **VcScheduler.-ScheduleProject** die Projektdaten berechnet wurden, den **Spätesten Anfang** des Projektes erfragen, wenn bei **VcScheduler.ScheduleProject** ein Enddatum vorgegeben wurde.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Index des Datenfeldes, das das geplante Startdatum des Projektes enthält

ScheduleSuccessorsOnlyEnabled

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob bei der Zeitrechnung nur die Vorgänge, die Vorgänger besitzen, berechnet werden. Ein "Projektstart" beim Aufruf der Zeitrechnung wird hier also ignoriert.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Berechnung nur der Knoten mit Vorgängern ist an/abgeschaltet
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

StartDateForAutomaticScheduling

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie für den Fall, dass die automatische Zeitberechnung durchgeführt wird, das Startdatum des Projektes setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Date	Gewünschtes Startdatum für automatische Zeitrechnung

StartDateNotEarlierThanDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können Sie den Index des Datenfeldes setzen oder erfragen, das das gewünschte früheste Startdatum eines Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das das gewünschte früheste Startdatum enthält

TotalFloatDataFieldIndex

Eigenschaft von VcScheduler

Mit dieser Eigenschaft können sie den Index des Datenfeldes setzen oder erfragen, das den berechneten Gesamtpuffer des Vorgangs enthält. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Gesamtpuffer enthält

Methoden

ScheduleProject

Methode von VcScheduler

Mit dieser Methode können Sie die Daten (frühester/spätester Anfang, frühestes/spätestes Ende, Freier Puffer, Gesamtpuffer) eines Projektes berechnen lassen, wobei Sie das gewünschte Anfangs- und Enddatum mit dieser Methode setzen. Bei alleiniger Übergabe des Starttermins wird das Projektende, bei alleiniger Übergabe des Endtermins wird der Projektstart berechnet. Es können auch beide Termine übergeben werden, die Vorgänge erhalten dann entsprechende Pufferzeiten. (Dies geht allerdings nur, wenn die Termine zueinander passen, d.h. der Endtermin sollte beispielsweise nicht innerhalb der Projektzeit liegen.) Das Fehlen beider Daten führt zu einer Fehlermeldung. Falls ein Zyklus der Knoten und Verbindungen festgestellt wird, werden diese automatisch markiert.

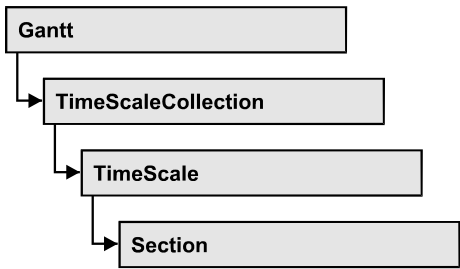
Die Ergebnisse werden in Feldern gespeichert, die Sie mit den Eigenschaften **EarlyStartDateDataFieldIndex**, **LateStartDateDataFieldIndex**, **EarlyEndDateDataFieldIndex**, **LateEndDateDataFieldIndex**, **FreeFloatDataFieldIndex** und **TotalFloatDataFieldIndex** festlegen.

	Datentyp	Beschreibung
Parameter:		
⇒ startDate	Date	Gewünschtes Anfangsdatum
⇒ endDate	Date	Gewünschtes Enddatum
Rückgabewert	Boolean	Die Projektdaten wurden erfolgreich berechnet (True) / nicht berechnet (False)

Code-Beispiel

```
VcScheduler.ScheduleProject (3.5.2012,1.10.2012)
```

7.73 VcSection



Ein Objekt vom Typ VcSection stellt einen Abschnitt einer Zeitskala dar.

Eigenschaften

- CalendarGridEx
- Collapse
- DateLineGrid
- LineColor
- LineColor
- Ribbon
- StartDate
- Unit
- UnitWidth
- UnitWidthEx

Eigenschaften

CalendarGridEx

Nur-Lese-Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie ein im Zeitskalenabschnitt verwendetes Kalenderliniengitter erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ gridIndex	Integer Mögliche Werte:	Index der Kalenderliniengitters Datenfeldindex
Eigenschaftswert	VcCalendarGrid	Kalenderliniengitter-Objekt

Collapse

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die arbeitsfreien Bereiche in diesem Zeitskalenabschnitt kollabiert werden sollen. Sie können diese Eigenschaft auch im Unterdialog **Zeitskalenabschnitt bearbeiten** von **Zeitskala festlegen** setzen. Dieser Dialog ist zu erreichen über die Eigenschaftenseite **Objekte**, Schaltfläche **Zeitskalen...**

Hinweis: Wenn Sie den Eigenschaftswert zur Laufzeit verändern, beachten Sie bitte, dass sich der sichtbare Zeitskalenbereich verschiebt. Möchten Sie sicherstellen, dass Sie links stets das gleiche Bezugsdatum haben, rufen Sie bitte folgende Methode auf:

```
Set_NonWorkIntervalsCollapsed(vcGantt1, true);
```

```
private static void Set_NonWorkIntervalsCollapsed(VcGantt gantt, bool collapse)
```

```
{
```

```
    DateTime dt_left = new DateTime();
```

```
    DateTime dt_right = new DateTime();
```

```
    gantt.GetCurrentViewDates(ref dt_left, ref dt_right);
```

```
gantt.TimeScaleCollection.Active.get_Section(0).NonWorkIntervalsCollapsed  
= collapse;
```

```
gantt.ScrollToDate(dt_left, VcHorizontalAlignment.vcLeftAligned, 0);
```

```
}
```

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Arbeitsfreie Bereiche werden/werden nicht kollabiert.
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
```

```

Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScaleCltn = VcGantt1.TimeScaleCollection
Set timeScale = timeScaleCltn.Active
Set section = timeScale.Section(1)
section.Collapse = True

```

DateLineGrid

Nur-Lese-Eigenschaft von VcSection

Mit dieser Eigenschaft haben Sie Zugriff auf das VcDateLineGrid-Objekt, mit dem Sie Zeitabschnitte (Tage, Wochen, Monate, etc.) durch vertikale Linien markieren können.

	Datentyp	Beschreibung
Parameter: ⇒ gridIndex	Integer Mögliche Werte:	Index der Terminliniengitters Datenfeldindex
Eigenschaftswert	VcDateLineGrid	Terminliniengitter-Objekt

Code-Beispiel

```

Dim timeScale As VcTimeScale
Dim section As VcSection
Dim dateLineGrid As VcDateLineGrid

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
Set dateLineGrid = section.DateLineGrid (0)

```

LineColor

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie die Farbe der (Rand)linie **n aller** Zeitskalenabschnitte gemeinsam festlegen oder erfragen. Die Eigenschaft gibt die Linienfarbe des ersten Zeitskalenabschnitts zurück,. Es ist nicht möglich, für jeden SAbschnitt eine eigene Farbe zu definieren.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer Mögliche Werte:	RGB-Farbwerte ({0...255},{0...255},{0...255}) Datenfeldindex

Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})
-------------------------	-------	--

LineColor

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie die Farbe der (Rand)linie n **aller** Zeitskalenabschnitte gemeinsam festlegen oder erfragen. Die Eigenschaft gibt die Linienfarbe des ersten Zeitskalenabschnitts zurück,. Es ist nicht möglich, für jeden SAbschnitt eine eigene Farbe zu definieren.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Integer	RGB-Farbwerte ({0...255},{0...255},{0...255})
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Ribbon

Eigenschaft von VcSection

Mit dieser Eigenschaft haben Sie Zugriff auf die einzelnen Zeitstreifen eines Zeitskalenabschnitts.

	Datentyp	Beschreibung
Parameter: ⇒ ribbonIndex	Integer	Index des Zeitskalenstreifens
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	VcRibbon	Zeitstreifen-Objekt

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim section As VcSection
Dim ribbon As VcRibbon
```

```

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
Set ribbon = section.ribbon(0)

```

StartDate

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie das Anfangsdatum eines Zeitskalenabschnitts erfragen oder festlegen. Das Anfangsdatum des ersten Zeitskalenabschnitts (Section 0) wird durch den Anfang der Zeitskala festgelegt und darf hier nicht geändert, sondern nur erfragt werden. Sie dürfen auch kein Anfangsdatum wählen, das außerhalb des Bereichs der Zeitskala liegt.

	Datentyp	Beschreibung
Eigenschaftswert	Date/Time	Anfangsdatum des Zeitskalenabschnittes

Code-Beispiel

```

Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(1)
section.StartDate = "21.06.14"

```

Unit

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie die einem Zeitskalenabschnitt zu Grunde liegende Zeiteinheit erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	TimeUnitEnum	Zeiteinheit des Zeitskalenabschnitts
	Mögliche Werte: vcDay 5 vcHour 6 vcMinute 7 vcSecond 8	Zeiteinheit Tag Zeiteinheit Stunde Zeiteinheit Minute Zeiteinheit Sekunde

Code-Beispiel

```

Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
section.Unit = vcHour

```


UnitWidth

Eigenschaft von VcSection

Mit dieser Eigenschaft können Sie die Breite pro Zeiteinheit in 1/100 Millimetern für einen Skalenabschnitt erfragen oder festlegen. Sie können diese Eigenschaft auch im Dialog **Zeitskala festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Breite der Einheiten (in 1/100 mm; muss > 0 sein)

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
section.Unit = vcDay
section.UnitWidth = 660
```

UnitWidthEx

Eigenschaft von VcSection

Diese Eigenschaft unterscheidet sich von **UnitWidth** nur durch den Datentyp **Double** und ist damit genauer als **UnitWidth**.

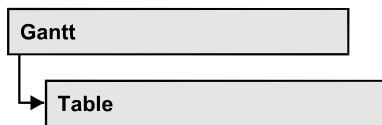
	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite der Einheiten (1/100 mm)

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
section.Unit = vcDay
section.UnitWidthEx = 660
```

7.74 VcTable



Ein Objekt des Typs VcTable kontrolliert die grafische Aufbereitung des Tabellenteils der Darstellung: die Spaltenüberschriften, Spaltenbreiten und die verfügbaren Formate.

Eigenschaften

- ColumnTitle
- ColumnWidth
- Name
- NoOfColumns
- Position
- TableFormatCollection
- UpdateBehaviorName
- Visible

Methoden

- IdentifyFormatField
- OptimizeColumnWidth

Eigenschaften

ColumnTitle

Eigenschaft von VcTable

Mit dieser Eigenschaft können Sie für jede Spalte der Tabelle die gewünschte Überschrift setzen. Diese Eigenschaft können Sie auch im Dialog **Tabelle bearbeiten** einstellen.

	Datentyp	Beschreibung
Parameter: ⇒ colNumber	Integer	Nummer der Tabellenspalte
	Mögliche Werte:	Datenfeldindex

Eigenschaftswert	String	Spaltentitel
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim table As VcTable

Set table = VcGantt1.Table
table.ColumnTitle(1) = "ID"
```

ColumnWidth**Eigenschaft von VcTable**

Mit dieser Eigenschaft können Sie für jede Spalte der Tabelle die gewünschte Breite setzen. Diese Eigenschaft können Sie auch im Dialog **Tabelle bearbeiten** einstellen.

	Datentyp	Beschreibung
Parameter: ⇒ colNumber	Integer	Nummer der Tabellenspalte
	Mögliche Werte:	Datenfeldindex
Eigenschaftswert	Long	Spaltenbreite in 1/100 mm

Code-Beispiel

```
Dim table As VcTable

Set table = VcGantt1.Table
table.ColumnWidth(1) = 1200
```

Name**Eigenschaft von VcTable**

Mit dieser Eigenschaft können Sie der Tabelle einen Namen zuweisen oder erfragen. Diese Eigenschaft können Sie auch im Dialog **Tabelle bearbeiten** einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Tabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

NoOfColumns

Nur-Lese-Eigenschaft von VcTable

Mit dieser Eigenschaft können Sie die Anzahl der Spalten der Tabelle erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Tabellenspalten
	Mögliche Werte:	Datenfeldindex

Position

Nur-Lese-Eigenschaft von VcTable

Mit dieser Eigenschaft können Sie erfragen, ob die Tabelle rechts oder links vom Diagramm dargestellt wird.

	Datentyp	Beschreibung
Eigenschaftswert	TablePositionEnum	Position der Tabelle
	Mögliche Werte: vcLeftTable 0 vcRightTable 1	Tabelle links vom Diagramm Tabelle rechts vom Diagramm

Code-Beispiel

```
Dim table As VcTable

Set table = VcGantt1.Table
MsgBox (table.Position)
```

TableFormatCollection

Nur-Lese-Eigenschaft von VcTable

Mit dieser Eigenschaft haben Sie Zugriff auf die TableFormat-Auflistung, in dem alle zur Verfügung stehenden Tabellenformate enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcTableFormatCollection	TableFormatCollection-Objekt

Code-Beispiel

```
Dim table As VcTable
Dim formatCltn As VcTableFormatCollection

Set table = VcGantt1.Table
```

```
Set formatCltn = table.TableFormatCollection
```

UpdateBehaviorName

Eigenschaft von VcTable

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzustimmungstabelle

Visible

Eigenschaft von VcTable

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Tabelle sichtbar ist oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Tabelle sichtbar/unsichtbar
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub chkBoxTable_Click()

    Dim table As VcTable

    Set table = VcGantt1.table

    If chkBoxTable.Value = vbChecked Then
        table.Visible = True
    Else
        table.Visible = False
    End If

End Sub
```

Methoden

IdentifyFormatField

Methode von VcTable

Mit dieser Methode können Sie den Index des an der bezeichneten Position befindlichen Formatfeldes erfragen. Falls sich an der bezeichneten Position ein Feld befindet, wird **True** zurückgegeben, ansonsten **False**.

	Datentyp	Beschreibung
Parameter: ⇒ x ⇒ y ⇐ format ⇐ formatFieldIndex	Long Long VcTableFormat Integer Mögliche Werte:	X-Koordinate der Position Y-Koordinate der Position Identifiziertes Format Format-Feldindex Datenfeldindex
Rückgabewert	Boolean	Ein Formatfeld befindet sich/befindet sich nicht an der angegebenen Position

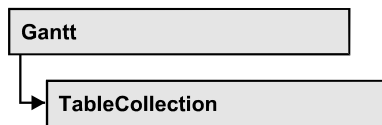
OptimizeColumnWidth

Methode von VcTable

Mit dieser Methode können Sie die optimierte Breite einer Spalte berechnen. Dabei entscheidet die Länge des längsten Textes in dieser Spalte über die Spaltenbreite. Die Angabe von ColumnNo = 0 bewirkt, dass alle Spalten optimiert werden.

	Datentyp	Beschreibung
Parameter: ⇒ columnNo	Integer Mögliche Werte:	Spaltennummer Datenfeldindex
Rückgabewert	Void	

7.75 VcTableCollection



Ein Objekt des Typs VcTableCollection beinhaltet alle bestehenden Tabellen. Über **For Each table In TableCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Tabellen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **TableByName** und **TableByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Tabellen kann über die Eigenschaft **Count** erfragt werden. Die aktuelle Tabelle kann durch die Eigenschaft **Active** gesetzt oder erfragt werden.

Eigenschaften

- _NewEnum
- Active
- Count

Methoden

- FirstTable
- NextTable
- TableByIndex
- TableByName

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcTableCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Tabellenobjekte iterieren. In VB wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode GetEnumerator angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim table As VcTable

For Each table In VcGantt1.TableCollection
    Debug.Print table.Name
Next
```

Active**Eigenschaft von VcTableCollection**

Mit dieser Eigenschaft kann die aktuell für die Darstellung verwendete Tabelle erfragt oder gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcTable	Aktuell verwendete Tabelle

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim activeTable As VcTable

Set tableCltn = VcGantt1.TableCollection.
                TableByName("TABLE_1"). _
                TableCollection
Set activeTable = tableCltn.Active
```

Count**Nur-Lese-Eigenschaft von VcTableCollection**

Mit dieser Eigenschaft kann die Anzahl der Tabellen in der TabellenAuflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Tabellen

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim numberTables As Integer

Set tableCltn = VcGantt1.TableCollection
numberTables = tableCltn.Count
```


Methoden

FirstTable

Methode von VcTableCollection

Mit dieser Methode können Sie auf die erste Tabelle der Tabellenauflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextTable** über die nachfolgenden Tabellen zu iterieren. Existiert keine Tabelle in der Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTable	Erste Tabelle

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim table As VcTable

Set tableCltn = VcGantt1.TableCollection
Set table = tableCltn.FirstTable
```

NextTable

Methode von VcTableCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Tabellen der Table-Auflistung zugreifen, nachdem Sie mit der Methode **FirstTable** den Initialwert erfasst haben. Sind alle Tabellen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTable	Nachfolgende Tabelle

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim table As VcTable
Dim stringTable As String

Set tableCltn = VcGantt1.TableCollection
Set table = tableCltn.FirstTable

While Not table Is Nothing
    stringTable = table.AllData
    Listbox.AddItem (stringTable)
    Set table = tableCltn.NextTable
Wend
```

TableByIndex

Methode von VcTableCollection

Mit dieser Methode können Sie auf eine einzelne Tabellen über ihren Index zugreifen. Existiert keine Tabelle unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Tabelle Datenfeldindex
Rückgabewert	VcTable	Ermitteltes Tabellenobjekt

TableByName

Methode von VcTableCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Tabellenobjekt zugreifen. Existiert kein Tabellenobjekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

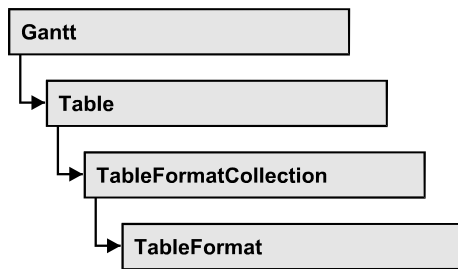
	Datentyp	Beschreibung
Parameter: ⇒ tableName	String Mögliche Werte:	Name der Tabelle Name der Farbzusordnungstabelle
Rückgabewert	VcTable	Zurückgegebene Datentabelle

Code-Beispiel

```
Dim tableCltn As VcTableCollection
Dim table As VcTable

Set tableCltn = VcGantt1.TableCollection
Set table = tableCltn.TableByName("Standard")
```

7.76 VcTableFormat



Ein Objekt vom Typ VcTableFormat legt Inhalt und Erscheinungsbild einer Zeile der Tabelle fest. In einer Zeile der Tabelle stehen entweder die Daten eines Vorgangs oder eine Gruppenüberschrift. In einem Tabellenformat kann man für jedes Tabellenfeld (definiert durch die Angabe der Spalte) separat festlegen, welches Datenfeld es enthalten soll, sowie Schriftattribute (Schriftart und -farbe), Hintergrundfarbe, horizontale Ausrichtung und Ränder einstellen.

Verfügbare Tabellenformate:

- StandardList (für normale Vorgänge)
- ListFormat2 (alternatives Tabellenformat für normale Vorgänge, über Filter zuweisbar)
- ListFormat3 (alternatives Tabellenformat für normale Vorgänge, über Filter zuweisbar)
- Subtitle (für Gruppenüberschriften, wenn die Gruppe expandiert ist)
- Subtitle_n (für Gruppenüberschriften bei mehrstufiger Gruppierung, wenn die Gruppe expandiert ist)
- Collapsed (für Gruppenüberschriften, wenn die Gruppe kollabiert ist)
- Collapsed_n (für Gruppenüberschriften bei mehrstufiger Gruppierung, wenn die Gruppe kollabiert ist)
- Hierarchy (für Summenvorgänge bei hierarchischer Darstellung)
- HierarchyCollapsed (für kollabierte Summenvorgänge bei hierarchischer Darstellung)

Eigenschaften

- _NewEnum
- CollapseColumn
- FieldsSeparatedByLines
- FilterName

- FormatField
- FormatFieldCount
- IndentColumn
- IndentWidth
- Name
- SeparationLineColor
- ThreeDEffect

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcTableFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Tabellenformatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim formatField As VcTableFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

CollapseColumn

Eigenschaft von VcTableFormat

Mit dieser Eigenschaft können Sie festlegen, ob und in welcher Spalte, die mehrere Zeilen enthält, +/- zum Ein- oder Ausblenden der weiteren Zeilen angezeigt werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzeige von +/- in Spalte eingeschaltet.
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
' Display of +/- in the fifth column
VcGantt1.Table.TableFormatCollection.FormatByName("Hierarchy").CollapseColumn =
5
VcGantt1.Table.TableFormatCollection.FormatByName("HierarchyCollapsed").Collapse
Column = 5
```

FieldsSeparatedByLines**Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft können Sie festlegen, ob die Tabellenfelder durch Linien getrennt werden (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Tabellenfelder werden durch Linien getrennt (True)/ nicht getrennt (False).
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim format As VcTableFormat

Set format = VcGantt1.Table.TableFormatCollection.FormatByName("StandardList")
format.FieldsSeparatedByLines = True
```

FilterName**Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft können Sie den Filter setzen oder erfragen, der bestimmt, unter welcher Bedingung (d. h. für welche Vorgänge) dieses Tabellenformat in der Tabelle verwendet werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	String	Filtername
	Mögliche Werte:	Name der Farbzusordnungstabelle

Code-Beispiel

```
Dim format As VcTableFormat

Set format = VcGantt1.Table.TableFormatCollection.FormatByName("ListFormat2")
format.FilterName = "Code1NotA"
```

FormatField**Nur-Lese-Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft können Sie ein VcTableFormatField-Objekt per Index holen. Der Index muss im Bereich von 0 bis .FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht wie in den bisherigen Feldeigenschaften von 1 bis .FormatFieldCount!

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Tabellenformatfeldes 0FormatFieldCount-1 Datenfeldindex
Eigenschaftswert	VcTableFormatField	Tabellenformatfeld

FormatFieldCount**Nur-Lese-Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft können Sie die Anzahl der Tabellenspalten dieses Tabellenformats erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer Mögliche Werte:	Anzahl der Tabellenspalten Datenfeldindex

Code-Beispiel

```
Dim format As VcTableFormat
Dim numberOfColumns As Integer

Set format = VcGantt1.Table.TableFormatCollection.FormatByName("StandardList")
numberOfColumns = FormatFieldCount
```

IndentColumn

Eigenschaft von VcTableFormat

Mit dieser Eigenschaft können Sie festlegen, welche Spalte einen Einzug bekommen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Nummer der Spalte mit Einzug
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
' Second column is indented
VcGantt1.Table.TableFormatCollection.FormatByName("StandardList").IndentColumn
= 2
```

IndentWidth

Eigenschaft von VcTableFormat

Mit dieser Eigenschaft können Sie die Größe des Text-Einzuges (in mm) einer Tabellenspalte festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Grösse des Einzugs

Code-Beispiel

```
' Second column is indented by 100 mm
VcGantt1.Table.TableFormatCollection.FormatByName("StandardList").IndentColumn
= 2
VcGantt1.Table.TableFormatCollection.FormatByName("StandardList").IndentWidth =
100
```

Name

Eigenschaft von VcTableFormat

Mit dieser Eigenschaft können Sie den Namen des Tabellenformats erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Tabellenformatname
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim format As VcTableFormat
Dim formatName As String

Set format = VcGantt1.Table.TableFormatCollection.FirstFormat
formatName = format.Name
```

SeparationLineColor**Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft können Sie die Farbe der Trennlinien für die Tabellenfelder festlegen. Die Standardfarbe ist weiß.

	Datentyp	Beschreibung
Eigenschaftswert	Color RGB	Farbwert {0...255},{0...255},{0...255} Standardwert: RGB(0,0,0)

Code-Beispiel

```
VcTableFormat.SeparationLineColor = RGB(255, 204, 204)
```

ThreeDEffect**Eigenschaft von VcTableFormat**

Mit dieser Eigenschaft wird festgelegt oder erfragt, ob das Tabellenformat einen 3D-Effekt erhält oder nicht.

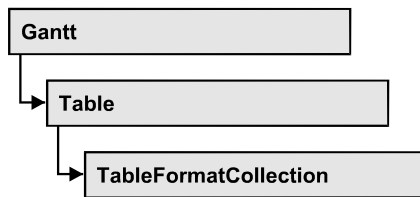
	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	3D-Effekt eingeschaltet (True)/ausgeschaltet (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim format As VcTableFormat

Set format = VcGantt1.Table.TableFormatCollection.FormatByName("StandardList")
format.ThreeDEffect = True
```


7.77 VcTableFormatCollection



In einem Objekt vom Typ `VcTableFormatCollection` sind alle verfügbaren Tabellenformate enthalten. Über **For Each tableFormat In TableFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Formate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **FormatByName** und **FormatByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Tabellenformate kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `FirstFormat`
- `FormatByIndex`
- `FormatByName`
- `NextFormat`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcTableFormatCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Tabellenformat-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim format As VcTableFormat

For Each format In VcGantt1.Table.TableFormatCollection
    Debug.Print format.Name
Next
```

Count**Nur-Lese-Eigenschaft von VcTableFormatCollection**

Mit dieser Eigenschaft können Sie die Anzahl der Tabellenformatobjekte in der TableFormat-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Tabellenformate

Code-Beispiel

```
Dim formatCltn As VcTableFormatCollection
Dim numberOfFormats As Long

Set formatCltn = VcGantt1.TableFormatCollection
numberOfFormats = formatCltn.Count
```

Methoden**FirstFormat****Methode von VcTableFormatCollection**

Mit dieser Methode können Sie auf den Initialwert, d. h. das erste Tabellenformat der TableFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Tabellenformate zu iterieren. Existiert kein Tabellenformat in der TableFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTableFormat	erstes Tabellenformat

Code-Beispiel

```
Dim format As VcTableFormat
```

```
Set format = VcGantt1.Table.TableFormatCollection.FirstFormat
```

FormatByIndex

Methode von VcTableFormatCollection

Mit dieser Methode können Sie auf ein einzelnes Tabellenformat über seinen Index zugreifen. Existiert kein Tabellenformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index des Tabellenformats Datenfeldindex
Rückgabewert	VcTableFormat	Ermitteltes Tabellenformatobjekt

FormatByName

Methode von VcTableFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Tabellenformat zugreifen. Existiert kein Tabellenformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ formatName	String Mögliche Werte:	Name des Tabellenformats Name der Farbzusordnungstabelle
Rückgabewert	VcTableFormat	Tabellenformat

Code-Beispiel

```
Dim format As VcTableFormat
```

```
Set format = VcGantt1.Table.TableFormatCollection.FormatByName("StandardList")
```

NextFormat

Methode von VcTableFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Tabellenformate der TableFormat-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFormat** den Initialwert erfasst haben. Sind alle Tabellenformate durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTableFormat	Nachfolgendes Tabellenformat

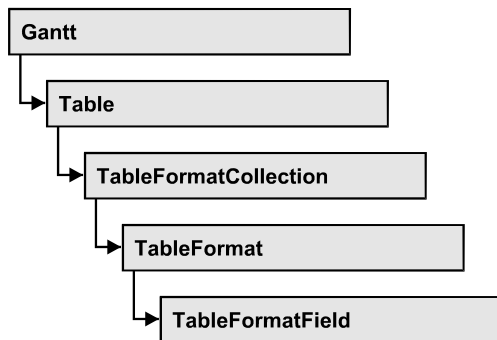
Code-Beispiel

```
Dim formatCltn As VcTableFormatCollection
Dim format As VcTableFormat

Set formatCltn = VcGantt1.Table.TableFormatCollection
Set format = formatCltn.FirstFormat

While Not format Is Nothing
    ListBox.AddItem format.Name
    Set format = formatCltn.NextFormat
Wend
```

7.78 VcTableFormatField



Ein Objekt vom Typ **VcTableFormatField** stellt ein Tabellenformatfeld, also ein Feld eines **VcTableFormat**-Objekts dar. Ein Tabellenformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Tabellenformat untergebracht ist. Für eine Tabelle können maximal 100 Formatfelder definiert werden.

Eigenschaften

- Alignment
- BottomMargin
- CombiField
- ConstantText
- FormatName
- GraphicsFileName
- GraphicsFileNameDataFieldIndex
- GraphicsFileNameMapName
- GraphicsHeight
- Index
- LeftMargin
- MaximumTextLineCount
- MinimumTextLineCount
- MultiState
- PatternBackgroundColorAsARGB
- PatternBackgroundColorDataFieldIndex
- PatternBackgroundColorMapName
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternEx
- PatternExDataFieldIndex

- PatternExMapName
- RightMargin
- TextDataFieldIndex
- TextFont
- TextFontColor
- TextFontColorDataFieldIndex
- TextFontColorMapName
- TextFontDataFieldIndex
- TextFontMapName
- TopMargin
- Type

Eigenschaften

Alignment

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Tabellenformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum	Ausrichtung des Feldinhalts
	Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

BottomMargin

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des unteren Randes des Tabellenformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Tabellenformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des unteren Randes des Tabellenformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

CombiField

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Tabellenfeld ein Kombifeld ist. (Vgl. Dialog **Tabellenformat bearbeiten**).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kombifeld (True) / kein Kombifeld (False)
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

ConstantText

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Tabellenformatfeld ausgeben, falls der Typ des Tabellenformatfeldes auf **vcFFTText** und falls die Eigenschaft **TextDataFieldIndex** auf **-1** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text
	Mögliche Werte:	Name der Farbzusordnungstabelle

FormatName

Nur-Lese-Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen des Tabellenformats erfragen, zu dem dieses Tabellenformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Tabellenformat-Objektes
	Mögliche Werte:	Name der Farbzordnungstabelle

GraphicsFileName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Namen einer Grafikdatei setzen oder erfragen, deren Inhalt in dem Tabellenformatfeld ausgegeben wird. Der Name muss eine gültige Grafikdatei bezeichnen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile)
- *.WMF mit eingebautem EMF

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei
	Mögliche Werte:	Name der Farbzordnungstabelle

GraphicsFileNameDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Datenfeldindex festlegen oder erfragen, der in der Eigenschaft **GraphicsFileNameMapName** benötigt wird. Beim Wert **-1** wird in dem Tabellenformatfeld die Grafikdatei ausgegeben, die im entsprechenden Tabellenformat angegeben ist. Ist ein gültiger Datenfeldindex angegeben und keine Zuordnungstabelle, dann wird der Grafikdateiname direkt aus dem Inhalt des angegebenen Datenfelds entnommen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes
	Mögliche Werte:	Datenfeldindex

GraphicsFileNameMapName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Namen einer Zuordnungstabelle vom Typ **vcGraphicsFileMap** oder "" setzen oder erfragen. Wenn ein Name und zusätzlich ein Datenfeldindex in der Eigenschaft **GraphicsFileNameDataFieldIndex** angegeben ist, wird eine Grafik aus der Zuordnungstabelle angezeigt. Trifft kein Datenfeldeintrag zu, wird die Grafik aus der Eigenschaft **GraphicsFileName** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafik-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzordnungstabelle

GraphicsHeight

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** die Höhe der Grafik im Tabellenformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe der Grafik in mm
	Mögliche Werte:	0 ... 99 Datenfeldindex

Index

Nur-Lese-Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Index des Tabellenformatfelds im zugehörigen Tabellenformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Tabellenformatfeldes
	Mögliche Werte:	Datenfeldindex

LeftMargin

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des linken Randes des Tabellenformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Tabellenformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des linken Randes des Tabellenformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

MaximumTextLineCount

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die maximale Anzahl der Zeilen in dem Tabellenformatfeld setzen oder erfragen, falls das Tabellenformatfeld vom Typ **vcFFTText** ist. Bitte sehen Sie auch die Eigenschaft **MinimumTextLineCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	maximale Zeilenzahl
	Mögliche Werte:	0...9 Datenfeldindex

MinimumTextLineCount

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die minimale Anzahl der Zeilen in dem Tabellenformatfeld setzen oder erfragen, falls der Typ des Tabellenformatfeldes auf **vcFFTText** gesetzt wurde. Ist in einem Knoten mehr Text vorhanden, als in die minimale Anzahl der Zeilen hineinpasst, wird dieses Feld für diesen Knoten dynamisch bis zur maximalen angegebenen Anzahl der Zeilen ausgedehnt. Bitte sehen Sie auch Eigenschaft **MaximumTextLineCount**. Wenn Sie dieser Eigenschaft einen Wert zuweisen, sollten Sie anschließend auch erneut der Eigenschaft **MaximumTextLineCount** den gewünschten Wert setzen, sonst könnte es vorkommen, dass das Maximum durch das Minimum überschrieben wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Zeilenzahl
	Mögliche Werte:	0...9 Datenfeldindex

MultiState

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das Tabellenformatfeld ein Multi-State-Feld ist. MultiState-Felder werden verwendet, um z.B. beim Anklicken eines Tabellenfeldes eine zyklische Abfolge von Zuständen wie auch der zugehörigen Datenfelder auszulösen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	MultiState-Feld (True) / kein MultiState-Feld (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

PatternBackgroundColorAsARGB

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Tabellenformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Tabellenformats besitzen soll.

Wenn in der Eigenschaft **BackColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Hintergrundfarbe des Tabellenformatfelds Standardwert: -1

PatternBackgroundColorDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternBackgroundColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

PatternBackgroundColorMapName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Farbzuhnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **PatternBackgroundColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **BackColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhnungstabelle
	Mögliche Werte:	Name der Farbzuhnungstabelle

PatternColorAsARGB

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Tabellenformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Tabellenformats besitzen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Tabellenformatfeldes

Code-Beispiel

```
Dim tableFormatCltn As VcTableFormatCollection
Dim tableFormatField As VcTableFormatField

Set tableFormatCltn = VcGantt1.TableFormatCollection
Set tableFormatField = tableFormatCltn.FirstFormat.formatField(0)
tableFormatField.PatternColor = RGB(0, 255, 0)
```

PatternColorDataFieldIndex

Nur-Lese-Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

PatternColorMapName

Nur-Lese-Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuhordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Kalendergitters aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColor** ausgegeben.

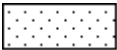
	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhordnungstabelle
	Mögliche Werte:	

	Name der Farbzuoordnungstabelle
--	---------------------------------

PatternEx

Nur-Lese-Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Tabellenformatfeldes ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp Standardwert: Wie im Dialog definiert
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppelter Liniendicke 
	vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 

vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster
vcDivotPattern 2036	Grassoden-Muster
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf
vcHorizontalPattern 3	Horizontale Linien
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmallCheckerBoardPattern
vcLargeConfettiPattern 2029	Konfetti-Muster, groß
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern

vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vcHorizontalPattern
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß
vcPlaidPattern 2035	Schottenstoff-Muster
vcShinglePattern 2039	Diagonales Dachschindel-Muster
vcSmallCheckerBoardPattern 2043	Schachbrettmuster
vcSmallConfettiPattern 2028	Konfetti-Muster
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet
vcTrellisPattern 2040	Spalier-Muster
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell
vcVerticalGradientPattern 62	Vertikaler Farbverlauf
vcVerticalPattern 2	Vertikale Linien
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel

vcWavePattern 2031	Horizontales Wellenmuster
vcWeavePattern 2034	Muster mit verwobenen Streifen
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke
vcZigZagPattern 2030	Horizontale Zickzack-Linien

PatternExDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternExMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

PatternExMapName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen einer Muster-Zuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Muster-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **PatternExDataFieldIndex** angegeben ist, wird das Muster aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **PatternEx** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle
	Mögliche Werte:	Name der Farbzusordnungstabelle

RightMargin

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des rechten Randes des Tabellenformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Tabellenformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des rechten Randes des Tabellenformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

TextDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTTText** den Index des Datenfeldes, dessen Inhalt in dem Tabellenformatfeld dargestellt werden soll, erfragen oder setzen. Bei **-1** wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes
	Mögliche Werte:	Datenfeldindex

TextFont

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Tabellenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftart in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Tabellenformatfelds

TextFontColor

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Tabellenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Tabellenformatfelds Standardwert: -1

TextFontColorDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Schriftfarbe-Zuordnungstabelle in der Eigenschaft **TextFontColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextFontColorMapName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzuoordnungstabelle (Typ vcColorMap) für die Schriftfarbe setzen oder erfragen, falls der Typ des Formatfeldes auf **vcFFTText** gesetzt wurde. Wird für den Namen "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn der Name einer Farbzuoordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **TextFontColorDataFieldIndex** angegeben ist, wird die Schriftfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein Eintrag der Zuordnungstabelle zu, wird die Hintergrundfarbe verwendet, die über die Eigenschaft **TextFontColor** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schriftfarben-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzuoordnungstabelle

TextFontDataFieldIndex

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Schrift-Zuordnungstabelle in der Eigenschaft **TextFontMapName** benötigt wird. Wenn Sie hier -1 angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex
	Mögliche Werte:	Datenfeldindex

TextFontMapName

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Namen einer Schrift-Zuordnungstabelle (Typ vcFontMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Schrift-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft

TextFontDataFieldIndex angegeben ist, wird die Schriftart aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Schriftart aus der Eigenschaft **TextFont** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schrift-Zuordnungstabelle
	Mögliche Werte:	Name der Farbzordnungstabelle

TopMargin

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie die Breite (in mm) des oberen Randes des Tabellenformatfeldes festlegen oder erfragen. Sie kann ebenfalls in dem Dialog **Tabellenformat bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des oberen Randes des Tabellenformatfeldes in mm
	Mögliche Werte:	0...9 Datenfeldindex

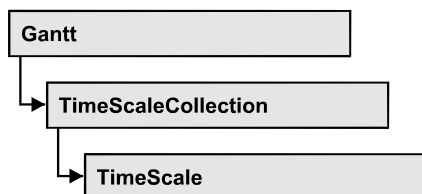
Type

Eigenschaft von VcTableFormatField

Mit dieser Eigenschaft können Sie den Typ des Tabellenformatfeldes erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldTypeEnum	Typ des Tabellenformatfeldes
	Mögliche Werte: vcFFTGraphics 64 vcFFTText 36	Grafik Text

7.79 VcTimeScale



Das VcTimeScale-Objekt ist die Zeitskala oberhalb des Diagrammbereichs, in dem die Knoten dargestellt werden. Zum dargestellten Zeitbereich können Sie aus verschiedenen Zeitskalen die passende wählen. Die Farbe und diverse Schriftattribute können beliebig gesetzt werden. Beim Zeitskala-Objekt werden auch das (vertikale) Zeitraster sowie ggf. die Kennzeichnung der Wochenenden eingeschaltet.

Eigenschaften

- BackgroundColor
- Font
- FontColor
- Name
- Ribbon
- Section
- ShowCalendarGrids
- ShowDateGrids
- ThreeDEffect
- UpdateBehaviorName

Eigenschaften

BackgroundColor

Eigenschaft von VcTimeScale

Mit dieser Eigenschaft können Sie die Hintergrundfarbe der Zeitskala erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte

Code-Beispiel

```
Dim timeScale As VcTimeScale

Set timeScale = VcGantt1.TimeScaleCollection.Active
timeScale.BackgroundColor = RGB(200, 100, 150)
```

Font**Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie alle Schriftattribute der Zeitskala erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute der Zeitskala

Code-Beispiel

```
Dim font As StdFont
font = VcGantt1.TimeScaleCollection.Active.Font
```

FontColor**Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie die Schriftfarbe der Zeitskala erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte

Code-Beispiel

```
Dim timeScale As VcTimeScale

Set timeScale = VcGantt1.TimeScaleCollection.Active
timeScale.FontColor = RGB(10, 220, 220)
```

Name**Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie den Namen einer Zeitskala setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name
	Mögliche Werte:	

	Name der Farbzusordnungstabelle
--	---------------------------------

Code-Beispiel

```
Dim timeScale As VcTimeScale

Set timeScale = VcGantt1.TimeScaleCollection.Active
MsgBox "Active Timescale: " & timeScale.Name
```

Ribbon**Nur-Lese-Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie auf einen Zeitstreifen der Zeitskala zugreifen.

	Datentyp	Beschreibung
Parameter: ⇒ sectionIndex Mögliche Werte: ⇒ ribbonIndex Mögliche Werte:	Integer Integer	Index des Zeitskalenabschnitts Datenfeldindex Index des Zeitstreifens Datenfeldindex
Eigenschaftswert	VcRibbon	Zeitstreifen-Objekt

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim ribbon As VcRibbon

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set ribbon = timeScale.Ribbon (0, 0)
```

Section**Nur-Lese-Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie auf einen Skalenabschnitt der Zeitskala zugreifen.

	Datentyp	Beschreibung
Parameter: ⇒ sectionIndex Mögliche Werte:	Integer	Index des Skalenabschnitts Datenfeldindex
Eigenschaftswert	VcSection	Skalenabschnittsobjekt

Code-Beispiel

```
Dim timeScale As VcTimeScale
Dim section As VcSection

Set timeScale = VcGantt1.TimeScaleCollection.Active
Set section = timeScale.Section(0)
```

ShowCalendarGrids**Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob arbeitsfreie Zeiten durch graue Schattierungen markiert werden sollen. Sie können diese Eigenschaft auch im Dialog **Zeitskala festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Arbeitsfreie Zeiten werden/werden nicht grau hinterlegt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Private Sub chkBoxCalGrids_Click()

    Dim timescale As VcTimeScale

    Set timescale = VcGantt1.TimeScaleCollection.Active

    If chkBoxCalGrids = vbChecked Then
        timescale.ShowCalendarGrids = True
    Else
        timescale.ShowCalendarGrids = False
    End If

End Sub
```

ShowDateGrids**Eigenschaft von VcTimeScale**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob vertikale Rasterlinien angezeigt werden. Sie können diese Eigenschaft auch im Dialog **Zeitskala festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rasterlinien werden/werden nicht angezeigt
	Mögliche Werte:	Gruppe unsichtbar/sichtbar

Code-Beispiel

```
Private Sub chkBoxDateGrid_Click()  
  
    Dim timescale As VcTimeScale  
  
    Set timescale = VcGantt1.TimeScaleCollection.Active  
  
    If chkBoxCalGrids = vbChecked Then  
        timescale.ShowDateGrids = True  
    Else  
        timescale.ShowDateGrids = False  
    End If  
  
End Sub
```

ThreeDEffect

Eigenschaft von VcTimeScale

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Zeitskala mit 3D-Effekt dargestellt wird. Sie können diese Eigenschaft auch im Dialog **Zeitskala festlegen** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean Mögliche Werte:	3D-Effekt eingeschaltet (True)/ausgeschaltet (False) Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim timescale As VcTimeScale  
  
Set timescale = VcGantt1.TimeScaleCollection.Active  
timescale.ThreeDEffect = False
```

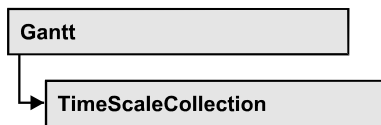
UpdateBehaviorName

Eigenschaft von VcTimeScale

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String Mögliche Werte:	Name des Aktualisierungsverhaltens Name der Farbzuoordnungstabelle

7.80 VcTimeScaleCollection



Ein Objekt des Typs `VcTimeScaleCollection` beinhaltet alle bestehenden Zeitskalen. Über **For Each timeScale In TimeScaleCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Zeitskalen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **TimeScaleByName** und **TimeScaleByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Zeitskalen kann über die Eigenschaft **Count** erfragt werden. Die aktuelle Zeitskala kann durch die Eigenschaft **Active** gesetzt oder erfragt werden.

Eigenschaften

- `_NewEnum`
- `Active`
- `Count`

Methoden

- `FirstTimeScale`
- `NextTimeScale`
- `TimeScaleByIndex`
- `TimeScaleByName`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcTimeScaleCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zeitskalenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim timescale As VcTimeScale

For Each timescale In VcGantt1.TimeScaleCollection
    Debug.Print timescale.Name
Next
```

Active**Eigenschaft von VcTimeScaleCollection**

Mit dieser Eigenschaft kann die aktuell für die Darstellung verwendete Zeitskala erfragt oder gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcTimeScale	Aktuell dargestellte Zeitskala

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
Dim timeScale As VcTimeScale

Set timeScaleCltn = VcGantt1.TimeScaleCollection
Set timeScale = timeScaleCltn.Active
```

Count**Nur-Lese-Eigenschaft von VcTimeScaleCollection**

Mit dieser Eigenschaft kann die Anzahl der Zeitskalen in der TimeScale-Auflistung abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Zeitskalen

Code-Beispiel

```
Dim numberOfTimeScales As Long

numberOfTimeScales = VcGantt1.TimeScaleCollection.Count
```

Methoden

FirstTimeScale

Methode von VcTimeScaleCollection

Mit dieser Methode können Sie auf die erste Zeitskala der TimeScale-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextTimeScale** über die nachfolgenden Zeitskalen zu iterieren. Existiert keine Zeitskala in der Zeitskalen-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTimeScale	Erste Zeitskala

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
Dim timeScale As VcTimeScale

Set timeScaleCltn = VcGantt1.TimeScaleCollection
Set timeScale = timeScaleCltn.FirstTimeScale
```

NextTimeScale

Methode von VcTimeScaleCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Zeitskalen der TimeScale-Auflistung zugreifen, nachdem Sie mit der Methode **FirstTimeScale** den Initialwert erfasst haben. Sind alle Zeitskalen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcTimeScale	Nachfolgende Zeitskala

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
Dim timeScale As VcTimeScale

Set timeScaleCltn = VcGantt1.TimeScaleCollection
Set timeScale = timeScaleCltn.FirstTimeScale

While Not timeScale Is Nothing
    List1.AddItem timeScale.Name
    Set timeScale = timeScaleCltn.NextTimeScale
Wend
```

TimeScaleByIndex

Methode von VcTimeScaleCollection

Mit dieser Methode können Sie auf eine einzelne Zeitskala über ihren Index zugreifen. Existiert keine Zeitskala unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Zeitskala Datenfeldindex
Rückgabewert	VcTimeScale	Ermitteltes Zeitskalenobjekt

TimeScaleByName

Methode von VcTimeScaleCollection

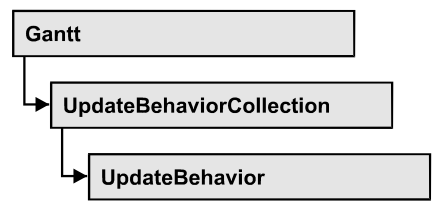
Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Zeitskala zugreifen. Existiert keine Zeitskala unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ timeScaleName	String Mögliche Werte:	Name der Zeitskala Name der Farbzusordnungstabelle
Rückgabewert	VcTimeScale	Zeitskala

Code-Beispiel

```
Dim timeScaleCltn As VcTimeScaleCollection
Set timeScaleCltn = VcGantt1.TimeScaleCollection
timeScaleCltn.Active = timeScaleCltn.TimeScaleByName("Days")
```


7.81 VcUpdateBehavior



Ein Objekt vom Typ **VcUpdateBehavior** beinhaltet eine Sammlung von Eigenschaften und Methoden, die das Aktualisierungsverhalten der Objekte am Bildschirm steuern, denen es zugewiesen wurde.

Eigenschaften

- IsEditable
- Name
- Specification

Methoden

- Context
- PutInOrderAfter

Eigenschaften

IsEditable

Eigenschaft von VcUpdateBehavior

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das spezifizierte Aktualisierungsverhalten zur Laufzeit editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Aktualisierungsverhalten editierbar (True) / nicht editierbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim updBeh As VcUpdateBehavior
Set updBeh = VcUpdateBehavior.UpdateBehaviorByName("Start")
```

```
updBeh.IsEditable = False
```

Name

Eigenschaft von VcUpdateBehavior

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.UpdateBehaviorCollection

For Each updBeh In updBehCltn
    ComboBox1.AddItem updBeh.Name
Next updBeh
```

Specification

Eigenschaft von VcUpdateBehavior

Mit dieser Eigenschaft können Sie die Spezifikation dieses Aktualisierungsverhaltens auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung eines Aktualisierungsverhaltens mit der Methode **VcUpdateBehaviorCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Code-Beispiel

```
Dim updateBehaviorCltn As VcUpdateBehaviorCollection
Dim updateBehavior As VcUpdateBehavior

updateBehaviorCltn = VcGantt1.UpdateBehaviorCollection
updateBehavior = updateBehaviorCltn.FirstUpdateBehavior
MsgBox(updateBehavior.Specification)
```

Methoden

Context

Methode von VcUpdateBehavior

Mit dieser Methode können Sie die Kontext-Setzungen des Aktualisierungsverhaltens erfragen.

	Datentyp	Beschreibung

PutInOrderAfter

Methode von VcUpdateBehavior

Mit dieser Methode können Sie dieses Aktualisierungsverhalten in der Auflistung aller Aktualisierungsverhalten hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Aktualisierungsverhalten an die erste Stelle gesetzt. Die Reihenfolge der Aktualisierungsverhalten in der Auflistung entscheidet darüber, in welcher Reihenfolge sie auf die zugewiesenen Objekte angewendet werden.

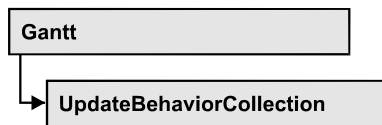
	Datentyp	Beschreibung
Parameter: refUpdateBehaviorName	String	Name des Aktualisierungsverhaltens, hinter das das aktuelle Aktualisierungsverhalten gesetzt werden soll.
	Mögliche Werte:	Name der Farbzusordnungstabelle
Rückgabewert	Void	

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh1 As VcUpdateBehavior
Dim updBeh2 As VcUpdateBehavior

updBehCltn = VcGantt1.UpdateBehaviorCollection()
updBeh1 = updBehCltn.Add("updBeh1")
updBeh2 = updBehCltn.Add("updBeh2")
updBeh1.PutInOrderAfter("updBeh2")
updBehCltn.Update()
```

7.82 VcUpdateBehaviorCollection



Im VcUpdateBehavior-Auflistungsobjekt sind alle verfügbaren Aktualisierungsverhalten zusammengefasst. Über **For Each UpdateBehavior In UpdateBehaviorCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Verhalten zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **UpdateBehaviorByName** und **UpdateBehaviorByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Aktualisierungsverhalten kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add...**, **Copy...** und **Remove...** ermöglichen das Hinzufügen, Kopieren und Löschen von Aktualisierungsverhalten.

Eigenschaften

- _NewEnum
- Active
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstUpdateBehavior
- NextUpdateBehavior
- Remove
- UpdateBehaviorByIndex
- UpdateBehaviorByName

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcUpdateBehaviorCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Aktualisierungsobjekte iterieren. In Visual Basic wird diese

Eigenschaft nie angezeigt, sondern über den Befehl **For Each element In collection** angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim updBeh As VcUpdateBehavior

For Each updBeh In VcGantt1.UpdateBehaviorCollection
    Debug.Print updBeh.Name
Next
```

Active

Eigenschaft von VcUpdateBehaviorCollection

Mit dieser Eigenschaft können sie das aktuell verwendete Aktualisierungsverhalten setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcUpdateBehavior	Aktuell aktives Aktualisierungsverhalten

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.UpdateBehaviorCollection
Set updBeh = UpdateBehaviorCltn.Active
```

Count

Nur-Lese-Eigenschaft von VcUpdateBehaviorCollection

Mit dieser Eigenschaft können Sie die Anzahl der Aktualisierungsverhalten in der Auflistung abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Aktualisierungsverhalten

Code-Beispiel

```
Dim updBeh As VcUpdateBehavior
Dim updBehCltn As VcUpdateBehaviorCollection
Dim numberOfUpdateBehaviors As Long
```

```

Set updBeh =
VcGantt1.UpdateBehaviorCollection.UpdateBehaviorByName("UPDBEHAVIOR_1")
Set updBehCltn = updBeh.UpdateBehaviorCollection

numberOfUpdateBehaviors = updBehCltn.Count

```

Methoden

Add

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie ein neues Aktualisierungsverhalten in der UpdateBehaviorCollection anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Aktualisierungsobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ updateBehaviorName	String Mögliche Werte:	Name des Aktualisierungsverhaltens Name der Farbzuoordnungstabelle
Rückgabewert	VcUpdateBehavior	Neues Aktualisierungsverhalten

Code-Beispiel

```
Set newUpdateBehavior = VcGantt1.UpdateBehaviorCollection.Add("test1")
```

AddBySpecification

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie ein Aktualisierungsverhalten über eine Spezifikation erzeugen. Dies dient der Persistenz von Aktualisierungsobjekten. Die Spezifikation eines Aktualisierungsverhaltens kann erfragt (siehe VcUpdateBehavior-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Aktualisierungsverhalten mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ Specification	String Mögliche Werte:	Spezifikation des Aktualisierungsverhaltens

1434 API-Referenz: VcUpdateBehaviorCollection

		Name der Farbzuoordnungstabelle
Rückgabewert	VcUpdateBehavior	Neues Aktualisierungsverhalten

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection  
  
updBehCltn = VcGantt1.UpdateBehaviorCollection  
updBehCltn.AddBySpecification(textSpecification)
```

Copy

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie ein Aktualisierungsverhalten kopieren. Wenn das Aktualisierungsverhalten mit dem angegebenen Namen existiert und der Name des neuen Aktualisierungsverhalten noch nicht verwendet wird, wird das neue Aktualisierungsobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ updateBehaviorName	String Mögliche Werte:	Name des zu kopierenden Aktualisierungsverhaltens Name der Farbzuoordnungstabelle
⇒ newUpdateBehaviorName	String Mögliche Werte:	Name des neuen Aktualisierungsverhaltens Name der Farbzuoordnungstabelle
Rückgabewert	VcUpdateBehavior	Aktualisierungsverhalten

Code-Beispiel

```
Dim updBeh As VcUpdateBehavior  
Dim updBehCltn As VcUpdateBehaviorCollection  
Dim updBeh As VcUpdateBehavior  
  
Set updBeh = VcGantt1.UpdateBehaviorCollection.FirstUpdateBehavior  
Set updBehCltn = histogram.UpdateBehaviorCollection  
Set updBeh = updBehCltn.Copy("CurrentUpdateBehavior", "NewUpdateBehavior")
```

FirstUpdateBehavior

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie auf das erste Aktualisierungsverhalten der UpdateBehavior-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextUpdateBehavior** über die nachfolgenden Aktualisierungs-

verhalten zu iterieren. Existiert kein Aktualisierungsverhalten in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcUpdateBehavior	Erstes Aktualisierungsverhalten

Code-Beispiel

```
Dim updBeh As VcUpdateBehavior
Set updBeh = VcGantt1.UpdateBehaviorCollection.FirstUpdateBehavior
```

NextUpdateBehavior

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Aktualisierungsverhalten der UpdateBehavior-Auflistung zugreifen, nachdem Sie mit der Methode **FirstUpdateBehavior** den Initialwert erfasst haben. Sind alle Aktualisierungsverhalten durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcUpdateBehavior	Nachfolgendes Aktualisierungsverhalten

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.UpdateBehaviorCollection
Set updBeh = updBehCltn.FirstUpdateBehavior

While Not updBeh Is Nothing
    List1.AddItem updBeh.Name
    Set updBeh = updBehCltn.NextUpdateBehavior
Wend
```

Remove

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie ein Aktualisierungsverhalten löschen. Wenn das Aktualisierungsverhalten noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

1436 API-Referenz: VcUpdateBehaviorCollection

	Datentyp	Beschreibung
Parameter: ⇒ updateBehaviorName	String Mögliche Werte:	Name des Aktualisierungsverhaltens Name der Farbzuordnungstabelle
Rückgabewert	Boolean	Aktualisierungsverhalten gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim histogram As VcHistogram
Dim updBehCltn As VcUpdateBehaviorCollection

Set histogram = VcGantt1.HistogramCollection.FirstHistogram
Set updBehCltn = histogram.UpdateBehaviorCollection
updBehCltn.Remove ("CurrentUpdateBehavior")
```

UpdateBehaviorByIndex

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie auf ein einzelnes Aktualisierungsverhalten über seinen Index zugreifen. Existiert kein Aktualisierungsverhalten unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer Mögliche Werte:	Index der Aktualisierungsverhaltens Datenfeldindex
Rückgabewert	VcUpdateBehavior	Ermitteltes Aktualisierungsverhalten

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.UpdateBehaviorCollection
Set updBeh = updBehCltn.UpdateBehaviorByIndex(2)
```

UpdateBehaviorByName

Methode von VcUpdateBehaviorCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Aktualisierungsverhalten zugreifen. Existiert kein Aktualisie-

rungsverhalten unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

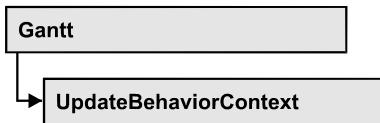
	Datentyp	Beschreibung
Parameter: ⇒ updateBehaviorName	String Mögliche Werte:	Name des Aktualisierungsverhaltens Name der Farbzordnungstabelle
Rückgabewert	VcUpdateBehavior	Aktualisierungsverhalten

Code-Beispiel

```
Dim updBehCltn As VcUpdateBehaviorCollection
Dim updBeh As VcUpdateBehavior

Set updBehCltn = VcGantt1.UpdateBehaviorCollection
Set updBeh = updBehCltn.UpdateBehaviorByName("Standard")
```

7.83 VcUpdateBehaviorContext



Ein Objekt vom Typ **VcUpdateBehaviorContext** beschreibt den Kontext des Aktualisierungsverhaltens, d.h. das Verhalten anderer Objekte, die vom LiveUpdate betroffen sind und das vom Benutzer einstellbar ist.

Eigenschaften

- DelayTime
- IsEditable
- Type
- UpdateMode

Eigenschaften

DelayTime

Eigenschaft von VcUpdateBehaviorContext

Mit dieser Eigenschaft können Sie die Verzögerungszeit setzen, nach der während des Verschiebvorgangs des Mauszeigers die Aktualisierung durch das Live-Update erfolgen soll. Die Setzung dieser Eigenschaft ist nur sinnvoll, wenn die Eigenschaft **UpdateMode** auf **Verharren während des Verschiebens** gesetzt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Millisekunden Standardwert: 500
	Mögliche Werte:	Datenfeldindex

Code-Beispiel

```
Dim updBehCtx As VcUpdateBehaviorContext
Dim delTim As Integer

delTim = VcGantt1.updBehCtx.DelayTime
```

IsEditable

Eigenschaft von VcUpdateBehaviorContext

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob der spezifizierte Kontext des Aktualisierungsverhaltens zur Laufzeit editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kontext des Aktualisierungsverhaltens editierbar (True) / nicht editierbar (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
Dim updBehCtx As VcUpdateBehaviorContext
updBehCtx.IsEditable = False
```

Type

Nur-Lese-Eigenschaft von VcUpdateBehaviorContext

Mit dieser Eigenschaft können Sie einzelne Bereiche (Kontext-Typen) erfragen, die vom Live-Update betroffen sind und auf die die Eigenschaften **Editable**, **UpdateMode** und **DelayTime** anwendbar sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcUpdateBehaviorContextType	Mögliche Aktualisierungsbereiche (Typen):
	Mögliche Werte:	
	vcHistogramsLayerSourceCurvesCalculations 1101	Kurvenberechnung aus Layerdaten in Histogrammen
	vcLinksChangeSuccessorNode 402	Verbindungen wechseln den Nachfolgerknoten
	vcNodeLevelLayoutsChangeNodesSortingOrder 201	Knotenebenen-Layouts ändern die Sortierrichtung für Knoten
	vcNodesAutoScheduling 105	Automatische Zeiteinplanung für Knoten
	vcNodesChangeDatesDuration 101	Knoten ändern ihre Termine oder ihre Dauer
	vcNodesFiltering 102	Knoten werden gefiltert
	vcNodesGrouping 104	Knoten werden gruppiert
	vcNumericScalesChangeUnitWidth 1201	Numerische Skalen ändern die Breite der Einheit
	vcSashesChangePosition 1501	Trennbalken ändern ihre Position
	vcTablesChangeColumnWidth 901	Tabellen ändern die Breite ihrer Spalten

vcTimeScalesChangeSectionStartDate 1002	Zeitskalen ändern das Anfangsdatum eines Zeitabschnitts Zeitskalen ändern die Breite der Einheit
vcTimeScalesChangeUnitWidth 1001	

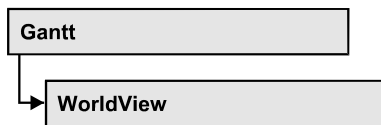
UpdateMode

Eigenschaft von VcUpdateBehaviorContext

Mit dieser Eigenschaft können Sie bei einem selbst generierten Aktualisierungsverhalten einstellen oder erfragen, bei welcher Aktion des Cursors die Aktualisierung durch das Live-Update erfolgen soll. Ist diese Eigenschaft auf **Verharren während des Verschiebens** gesetzt, kann über die Eigenschaft **DelayTime** die gewünschte Verzögerungszeit eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	VcUpdateMode	Mögliche Aktionen des Cursors: Standardwert: vcOnMouseMove
	Mögliche Werte: vcOnMouseMove 1 vcOnMouseUp 0 vcOnPauseWhileMouseMoving 2	Anzeige beim Bewegen des Mauszeigers Anzeige wenn die linke Maustaste losgelassen wird Anzeige, wenn beim Bewegen des Mauszeigers angehalten wird

7.84 VcWorldView



Ein Objekt vom Typ **VcWorldView** bezeichnet das Komplettansicht-Fenster.

Eigenschaften

- Border
- BorderColor
- Height
- HeightActualValue
- Left
- LeftActualValue
- MarkingColor
- Mode
- ParentHWnd
- ScrollBarMode
- Top
- TopActualValue
- UpdateBehaviorName
- Visible
- Width
- WidthActualValue

Eigenschaften

Border

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann gesetzt oder erfragt werden, ob die Komplettansicht einen Rahmen besitzt (nicht im Modus **vcPopupWindow**). Die Rahmenfarbe ist **Color.Black**. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rahmen um die Komplettansicht (True)/kein Rahmen um die Komplettansicht (False) Standardwert: True
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.WorldView.Mode = vcNotFixed
VcGantt1.WorldView.Border = True
```

BorderColor**Eigenschaft von VcWorldView**

Diese Eigenschaft setzt/liefert die des ggf. sichtbaren Rahmens.

	Datentyp	Beschreibung
Eigenschaftswert	Color RGB {{0...255},{0...255},{0...255}}	RGB-Farbwerte {{0...255},{0...255},{0...255}}

Height**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann die vertikale Ausdehnung der Komplettansicht erfragt werden. In den Modi **vcFixedAtBottom**, **vcFixedAtTop**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe der Komplettansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.WorldView.Height = 100
```

HeightActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte vertikale Ausdehnung der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Höhe der Komplettansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.Height = 300
```

Left

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die linke Position der Zusätzliche Ansichten erfragt werden. In den Modi **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linke Position der Komplettansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.WorldView.Left = 200
```

LeftActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte linke Position der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche linke Position der Komplettansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.LeftActualValue = 150
```

MarkingColor

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die Farbe der Linie des Rechtecks erfragt oder gesetzt werden, das in der **Zusätzliche Ansichten** den aktuell gewählten Ausschnitt anzeigt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte Standardwert: RGB(0, 0, 255)

Code-Beispiel

```
VcGantt1.WorldView.MarkingColor = RGB(255, 0, 0)
```

Mode**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann der Modus der Gesamtansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	WorldViewModeEnum	Modus der Gesamtansicht Standardwert: vcPopupWindow
	Mögliche Werte: vcFixedAtBottom 4	Die Komplettansicht wird unten im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Höhe verändert werden, während Position und Breite vorgegeben sind.
	vcFixedAtLeft 1	Die Komplettansicht wird links im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
	vcFixedAtRight 2	Die Komplettansicht wird rechts im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
	vcFixedAtTop 3	Die Komplettansicht wird oben im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
	vcNotFixed 5	Die Komplettansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des Steuerelements und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Bezugssystem der Koordinaten ist das Vaterfenster. Das Kindfenster ist ohne eigenen Fensterrahmen und kann vom Benutzer nicht interaktiv verschoben werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft VcWorldView.ParentHwnd geändert werden.

vcPopupWindow 6

Die Komplettansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Das Bezugssystem der Koordinaten ist der Bildschirm. Das Fenster kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die **Schließen**-Schaltfläche in der Titelleiste ausgeschaltet werden.

Code-Beispiel

```
VcGantt1.WorldView.Mode = vcFixedAtBottom
```

ParentHWnd**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann im Modus **vcNotFixed** das HWnd-Handle des Vaterfensters festgelegt werden, wenn die Komplettansicht beispielsweise in einem selbst implementierten Rahmenfenster erscheinen soll. Standardmäßig steht dies auf dem HWnd-Handle des Vaterfensters des VARCHART-ActiveX-Hauptfensters. Diese Eigenschaft kann nur zur Laufzeit verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_HANDLE	Zugriffsnummer

Code-Beispiel

```
MsgBox (VcGantt1.worldview.ParentHWnd)
```

ScrollBarMode**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann der Scrollbarmodus der Komplettansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	WorldViewScrollBarModeEnum	Scrollbarmodus Standardwert: NoScrollBar
	Mögliche Werte: vcAutomaticScrollBar 3 vcHorizontalScrollBar 1 vcNoScrollBar 0 vcVerticalScrollBar 2	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig. Anzeige einer horizontalen Bildlaufleiste, wenn nötig. Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt. Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcGantt1.WorldView.ScrollBarMode = vcAutomaticScrollBar
```

Top**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann die obere Position der Komplettansicht erfragt werden. In den Positionen **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Obere Position der Komplettansicht

Code-Beispiel

```
VcGantt1.WorldView.Top = 20
```

TopActualValue**Nur-Lese-Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann die tatsächlich dargestellte obere Position der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche obere Position der Komplettansicht Standardwert: 0

Code-Beispiel

```
VcGantt1.LegendView.TopActualValue = 40
```

UpdateBehaviorName**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens
	Mögliche Werte:	Name der Farbzuoordnungstabelle

Visible**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann festgelegt oder erfragt werden, ob die Komplettansicht sichtbar ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Komplettansicht sichtbar (True)/unsichtbar (False) Standardwert: False
	Mögliche Werte:	Gruppe unsichtbar/sichtbar Gruppen sichtbar/unsichtbar

Code-Beispiel

```
VcGantt1.WorldView.Visible = True
```

Width**Eigenschaft von VcWorldView**

Mit dieser Eigenschaft kann die horizontale Ausdehnung der Komplettansicht erfragt werden. In den Positionen **vcFixedAtLeft**, **vcFixedAtRight**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann diese Eigenschaft außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Horizontale Ausdehnung der Komplettansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.WorldView.Width = 200
```

WidthActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte horizontale Ausdehnung der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche horizontale Ausdehnung der Komplettansicht {0, ...} Standardwert: 100

Code-Beispiel

```
VcGantt1.LegendView.WidthActualValue = 600
```


8 Index

_NewEnum

Eigenschaft von

DataObjectFiles 467
VcBoxCollection 496
VcBoxFormat 503
VcBoxFormatCollection 508
VcCalendarCollection 533
VcCalendarGridCollection 558
VcCalendarProfileCollection 569
VcCurveCollection 607
VcDataDefinitionTable 614, 619
VcDataRecordCollection 630
VcDataTableCollection 639
VcDataTableFieldCollection 652
VcDateLineCollection 669
VcDateLineGridCollection 689
VcFilter 702
VcFilterCollection 708
VcGroupCollection 974
VcGroupLevelLayoutCollection 1006
VcHistogramCollection 1032
VcIntervalCollection 1063
VcLayerCollection 1109
VcLayerFormat 1115
VcLineFormat 1140
VcLineFormatCollection 1144
VcLinkAppearanceCollection 1179
VcLinkCollection 1185
VcMap 1189
VcMapCollection 1196
VcNodeCollection 1225

VcNumericScaleCollection 1256
VcTableCollection 1386
VcTableFormat 1391
VcTableFormatCollection 1396
VcTimeScaleCollection 1424
VcUpdateBehaviorCollection 1431

A

AboutBox

Methode von

VcGantt 809

About-Box 809

AbsoluteBottomMarginInCM

Eigenschaft von

VcPrinter 1262

AbsoluteBottomMarginInInches

Eigenschaft von

VcPrinter 1263

AbsoluteLeftMarginInCM

Eigenschaft von

VcPrinter 1263

AbsoluteLeftMarginInInches

Eigenschaft von

VcPrinter 1263

AbsoluteRightMarginInCM

Eigenschaft von

VcPrinter 1264

AbsoluteRightMarginInInches

Eigenschaft von

VcPrinter 1264

AbsoluteTopMarginInCM

Eigenschaft von

VcPrinter 1265

AbsoluteTopMarginInInches

Eigenschaft von

VcPrinter 1265

Active

Eigenschaft von

VcCalendarCollection 534

VcHistogramCollection 1033

VcNumericScaleCollection 1257

VcTableCollection 1387

VcTimeScaleCollection 1425

VcUpdateBehaviorCollection 1432

ActiveNodeFilter

Eigenschaft von

VcGantt 728

ActualEndDateDataFieldIndex

Eigenschaft von

VcScheduler 1368

ActualStartDateDataFieldIndex

Eigenschaft von

VcScheduler 1368

Add

Methode von

DataObjectFiles 469

VcBoxCollection 497

VcBoxFormatCollection 509

VcCalendarCollection 535

VcCalendarGridCollection 559

VcCalendarProfileCollection 569

VcCurveCollection 608

VcDataRecordCollection 631

VcDataTableCollection 640

VcDataTableFieldCollection 653

VcDateLineCollection 670

VcDateLineGridCollection 690

VcFilterCollection 710

VcGroupLevelLayoutCollection
1007

VcIntervalCollection 1063

VcLayerCollection 1110

VcLineFormatCollection 1145

VcLinkAppearanceCollection 1180

VcMapCollection 1197

VcUpdateBehaviorCollection 1433

AddBySpecification

Methode von

VcBoxCollection 498

VcBoxFormatCollection 510

VcCalendarCollection 535

VcCalendarGridCollection 560

VcCalendarProfileCollection 570

VcCurveCollection 608

VcDateLineCollection 671

VcDateLineGridCollection 691

VcFilterCollection 710

VcGroupLevelLayoutCollection
1008

VcIntervalCollection 1064

VcLayerCollection 1111

VcLineFormatCollection 1146

VcLinkAppearanceCollection 1181

VcMapCollection 1197

VcUpdateBehaviorCollection 1433

AddDuration

Methode von

VcCalendar 528

Addend

Eigenschaft von

VcCurve 575

AddSubCondition

Methode von

VcFilter 705

AdjustToReferenceDate

Eigenschaft von

VcDateLineGrid 677

Aktualisierungsverhalten

- editierbar 1428
- Kontext 1430
- Name 1429
- Reihenfolge 1430
- Specification 1429
- Aktueller Scrollwert**
 - grafisches Grundelement 821, 843
- Alignment**
 - Eigenschaft von
 - VcBoundingBox 472
 - VcBoxFormatField 514
 - VcLayerFormatField 1119
 - VcLineFormatField 1151
 - VcPrinter 1265
 - VcTableFormatField 1401
- AllBorderBoxesShownOnCombinedControls**
 - Eigenschaft von
 - VcPrinter 1266
- AllData**
 - Eigenschaft von
 - VcDataRecord 624
 - VcLink 1164
 - VcNode 1214
- AllowMultipleBoxMarking**
 - Eigenschaft von
 - VcGantt 728
- AllowNewBoxes**
 - Eigenschaft von
 - VcGantt 729
- AllowNewNodes**
 - Eigenschaft von
 - VcGantt 729
- AllowNumericScaleRescale**
 - Eigenschaft von
 - VcGantt 730
- AllowPanningMode**
 - Eigenschaft von
 - VcGantt 730
- AllowSelectionViaRubberRect**
 - Eigenschaft von
 - VcGantt 731
- AllowTableColumnWidthOptimization**
 - Eigenschaft von
 - VcGantt 731
- AllowTimescaleRescale**
 - Eigenschaft von
 - VcGantt 732
- AllowVerticalGroupMovementViaDiagram**
 - Eigenschaft von
 - VcGroupLevelLayout 980
- AllowVerticalGroupMovementViaTable**
 - Eigenschaft von
 - VcGroupLevelLayout 980
- AllowVerticalNodeMovement**
 - Eigenschaft von
 - VcGantt 732
- AllowVerticalNodeMovementViaTable**
 - Eigenschaft von
 - VcGantt 732
- AlwaysCurrentDate**
 - Eigenschaft von
 - VcDateLine 658
- AnchoringInteractionsAllowed**
 - Eigenschaft von
 - VcBox 481
- AnchoringLineVisible**
 - Eigenschaft von
 - VcBox 481
- AnchorToNode**
 - Methode von
 - VcBox 491

AnnotationAtBottom

Eigenschaft von
VcDateLineGrid 677

AnnotationAtCenter

Eigenschaft von
VcDateLineGrid 678

AnnotationAtTop

Eigenschaft von
VcDateLineGrid 678

Arbeitsfreie Zeiten

kennzeichnen 38

arbeitsfreie Zeiten anzeigen 789

Arbeitsperiode

Ende 529

Arbeitszeitelemente

berechnen 528

Arrangement

Eigenschaft von
VcGantt 733

ArrowKeyMode

Eigenschaft von
VcGantt 733

ArrowKeyStepSizeMultiplier

Eigenschaft von
VcGantt 735

AssignCalendarToNodes

Eigenschaft von
VcGantt 735

AssignmentDataTableName

Eigenschaft von
VcResourceScheduler2 1292

AssignmentIsResultFieldIndex

Eigenschaft von
VcResourceScheduler2 1294

AssignmentIsVisibleFieldIndex

Eigenschaft von
VcResourceScheduler2 1294

AssignmentLoadOrConsumptionPerItemFieldIndex

Eigenschaft von
VcResourceScheduler2 1295

AssignmentMaximumLoadFieldIndex

Eigenschaft von
VcResourceScheduler2 1295

AssignmentMinimumLoadFieldIndex

Eigenschaft von
VcResourceScheduler2 1296

AssignmentMinimumMaximumLoadType

Eigenschaft von
VcResourceScheduler2 1297

AssignmentOperationIDFieldIndex

Eigenschaft von
VcResourceScheduler2 1297

AssignmentResourceIDFieldIndex

Eigenschaft von
VcResourceScheduler2 1298

AssignmentResourceSelectionStrategyFieldIndex

Eigenschaft von
VcResourceScheduler2 1299

Auflistung Aktualisierungsverhalten

aktives Aktualisierungsverhalten
1432
Anzahl 1432
Enumerator 1432
hinzufügen 1433
hinzufügen über Spezifikation 1433
Iteration, erste 1435
Iteration, nächste 1435
Kopie 1434
löschen 1435
Zugriff über Index 1436
Zugriff über Name 1437

Ausgabeformat double 748

**Ausgabeformat double bei Num.
Skala 1243**

auslesen 1280

Auslieferung 16

AutoCollapseGroups

Eigenschaft von

VcGroupLevelLayout 980

VcHierarchyLevelLayout 1014

AutoExpandTargetGroup

Eigenschaft von

VcGroupLevelLayout 981

VcHierarchyLevelLayout 1014

AutomaticSchedulingEnabled

Eigenschaft von

VcScheduler 1368

Autoschedule 254

B

BackColorAsARGB

Eigenschaft von

VcCalendarGrid 540

VcInterval 1047

VcLayer 1070

BackColorDataFieldIndex

Eigenschaft von

VcCalendarGrid 541

VcLayer 1070

BackColorMapName

Eigenschaft von

VcCalendarGrid 541

VcLayer 1071

BackgroundColor

Eigenschaft von

VcTimeScale 1418

Balken

Balkenaussehen festlegen 262

in sichtbaren Bereich bringen 445

BarSeparationGroupBy

Eigenschaft von

VcGantt 736

BaseCalendarUsageForSupplementTimes

Eigenschaft von

VcResourceScheduler2 1300

BaseTimeUnit

Eigenschaft von

VcResourceScheduler2 1301

BaseTimeUnitsPerStep

Eigenschaft von

VcResourceScheduler2 1301

Benutzerkonten

Problem mit Control 442

Bildschirmausschnitt

verschieben 229

BodiesCollapsed

Eigenschaft von

VcGroupLevelLayout 981

VcHierarchyLevelLayout 1014

BodiesCollapsedDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 982

VcHierarchyLevelLayout 1015

BodiesCollapsedMapName

Eigenschaft von

VcGroupLevelLayout 982

VcHierarchyLevelLayout 1015

BodyCollapsed

Eigenschaft von

VcGroup 965

Border

Eigenschaft von

VcLegendView 1131

VcWorldView 1441

BorderArea

Eigenschaft von

VcGantt 737

siehe auch

VcBorderArea 471

BorderBox

Ausrichtung 472

Methode von

VcBorderArea 471

siehe auch

VcBorderBox 472

BorderColor

Eigenschaft von

VcLegendView 1132

VcWorldView 1442

Borland Delphi 441

Bottom

Eigenschaft von

VcRect 1286

BottomMargin

Eigenschaft von

VcLayerFormatField 1119, 1120

VcTableFormatField 1401

Box

Farbe der Randlinie 483

Größe ändern 489

ID des verbundenen Knotens 487

interaktiv anlegen erlaubt 729

interaktiv erzeugen 860, 861

Kontextmenü wird/wird nicht
angezeigt 740

Linienstärke 484

markieren 485

Mehrfachmarkierung zulassen 224

mit Knoten verankern 491

Offset 493, 494

Priorität 488

Referenzpunkt 488

sichtbar 490, 1043

siehe auch

VcBox 480

Spezifikation 489

Typ der Randlinie 484

über Index 498

Ursprungspunkt 487

Verankern mit Knoten 481

Verankerungslinie anzeigen 481

verschiebbar 486

Box Format

über Index 511

BoxByIndex

Methode von

VcBoxCollection 498

BoxByName

Methode von

VcBoxCollection 499

BoxCollection

Eigenschaft von

VcGantt 737

siehe auch

VcBoxCollection 496

Boxen 69

Ausdehnung 492

Boxen bearbeiten 316

Boxen verwalten 312

Boxformat bearbeiten 320

Boxformate verwalten 318

Name 486

Name UpdateBehavior 490, 595,
666, 1218, 1255, 1384, 1422

neue zulassen 224

Offset 493

Offset in Pixel umrechnen 492

Pixel in Offset umrechnen 494

Textfeld 482

BoxFormat

siehe auch

VcBoxFormat 503

BoxFormatCollection*Eigenschaft von*

VcGantt 737

siehe auch

VcBoxFormatCollection 508

Boxformate

Name 505

Boxformatfeld

Ausrichtung 514

Füllmuster 519

Hintergrundfarbe 518

Höhe Grafik 515

Index 516

maximale Zeilenzahl 516

Mindestbreite 518

minimale Zeilenzahl 517

Musterfarbe 519, 1155

Name des Formats 515

Schriftart 523

Schriftfarbe 523

Typ 523

BoxFormatField

siehe auch

VcBoxFormatField 514

Breite pro Zeiteinheit 204**Breitenverhältnis**

Tabelle/komplettes Diagramm 246

Breitenverhältniss Tabelle/Diagramm

genauere Methode 247

Browser 12, 20**C****CalcDuration***Methode von*

VcCalendar 528

CalculateCurrentWidth*Methode von*

VcLayer 1107

CalculateLineCount*Methode von*

VcLayerFormatField 1129

Calendar

siehe auch

VcCalendar 525

CalendarByIndex*Methode von*

VcCalendarCollection 536

CalendarByName*Methode von*

VcCalendarCollection 536

CalendarCollection*Eigenschaft von*

VcGantt 738

siehe auch

VcCalendarCollection 533

CalendarGrid

siehe auch

VcCalendarGrid 539

CalendarGridByIndex*Methode von*

VcCalendarGridCollection 560

CalendarGridByName*Methode von*

VcCalendarGridCollection 561

CalendarGridCollection*Eigenschaft von*

VcGantt 738

siehe auch

VcCalendarGridCollection 558

CalendarGridEx*Eigenschaft von*

VcSection 1375

CalendarGridName

Eigenschaft von

VcGroupLevelLayout 982

VcNodeLevelLayout 1230

CalendarGridsWithChildGroups

Eigenschaft von

VcGroupLevelLayout 983

CalendarName

Eigenschaft von

VcCalendarGrid 541

VcHistogram 1024

VcRibbon 1354

CalendarNameDataFieldIndex

Eigenschaft von

VcCalendarGrid 542

VcGroupLevelLayout 983

CalendarNameMapName

Eigenschaft von

VcCalendarGrid 542

CalendarProfile

siehe auch

VcCalendarProfile 565

CalendarProfileByIndex

Methode von

VcCalendarProfileCollection 570

CalendarProfileByName

Methode von

VcCalendarProfileCollection 571

CalendarProfileCollection

Eigenschaft von

VcCalendar 526

VcGantt 738

siehe auch

VcCalendarProfileCollection 568

CalendarProfileName

Eigenschaft von

VcInterval 1047

Clear

Methode von

DataObject 462

DataObjectFiles 469

VcCalendar 529

VcCurve 597

VcGantt 809

ClearAll

Methode von

VcGantt 810

Collapse

Eigenschaft von

VcSection 1376

CollapseColumn

Eigenschaft von

VcTableFormat 1391

ColorAsARGB

Eigenschaft von

VcMapEntry 1203

ColumnTitle

Eigenschaft von

VcTable 1381

ColumnWidth

Eigenschaft von

VcTable 1382

CombiField

Eigenschaft von

VcTableFormatField 1402

CombiningControlsEnabled

Eigenschaft von

VcPrinter 1267

ComparisonValueAsString

Eigenschaft von

VcFilterSubCondition 714

CompletionDataFieldIndex

Eigenschaft von

- VcLayer 1072
- ConfigurationName**
 - Eigenschaft von
 - VcGantt 739
- ConnectionOperator**
 - Eigenschaft von
 - VcFilterSubCondition 715
- ConsiderFilterEntries**
 - Eigenschaft von
 - VcMap 1190
- ConsiderLinkRelationTypesOnNodeDragging**
 - Eigenschaft von
 - VcGantt 740
- ConstantText**
 - Eigenschaft von
 - VcLayerFormatField 1120
 - VcLineFormatField 1151
 - VcTableFormatField 1402
- Context**
 - Methode von
 - VcUpdateBehavior 1430
- ContextMenuForBoxesEnabled**
 - Eigenschaft von
 - VcGantt 740
- ConvertDistance**
 - Methode von
 - VcGantt 810
- Copy**
 - Methode von
 - VcBoxCollection 499
 - VcBoxFormatCollection 510
 - VcCalendarCollection 537
 - VcCalendarGridCollection 561
 - VcCalendarProfileCollection 571
 - VcCurveCollection 609
 - VcDataTableCollection 641
 - VcDataTableFieldCollection 654
 - VcDateLineCollection 671
 - VcDateLineGridCollection 691
 - VcFilterCollection 711
 - VcGroupLevelLayoutCollection 1008
 - VcIntervalCollection 1064
 - VcLayerCollection 1111
 - VcLineFormatCollection 1146
 - VcLinkAppearanceCollection 1181
 - VcMapCollection 1198
 - VcUpdateBehaviorCollection 1434
- CopyFormatField**
 - Methode von
 - VcBoxFormat 506
 - VcLayerFormat 1117
 - VcLineFormat 1143
- CopySubCondition**
 - Methode von
 - VcFilter 706
- Count**
 - Eigenschaft von
 - DataObjectFiles 468
 - VcBoxCollection 497
 - VcBoxFormatCollection 509
 - VcCalendarCollection 535
 - VcCalendarGridCollection 559
 - VcCalendarProfileCollection 569
 - VcCurveCollection 607
 - VcDataDefinitionTable 615, 620
 - VcDataRecordCollection 631
 - VcDataTableCollection 640
 - VcDataTableFieldCollection 653
 - VcDateLineCollection 670
 - VcDateLineGridCollection 690
 - VcFilterCollection 709
 - VcGroupCollection 975

- VcGroupLevelLayoutCollection* 1007
- VcHistogramCollection* 1033
- VcIntervalCollection* 1063
- VcLayerCollection* 1110
- VcLineFormatCollection* 1145
- VcLinkAppearanceCollection* 1180
- VcLinkCollection* 1186
- VcMap* 1190
- VcMapCollection* 1196
- VcNodeCollection* 1226
- VcNumericScaleCollection* 1257
- VcTableCollection* 1387
- VcTableFormatCollection* 1397
- VcTimeScaleCollection* 1425
- VcUpdateBehaviorCollection* 1432
- CreateDataField**
 - Methode von
 - VcDataDefinitionTable* 615, 620
- CreateEntry**
 - Methode von
 - VcMap* 1192
- CreateHistogram**
 - Methode von
 - VcHistogramCollection* 1034
- CtrlCXVProcessing**
 - Eigenschaft von
 - VcGantt* 741
- CurrentHorizontalPagesCount**
 - Eigenschaft von
 - VcPrinter* 1267
- CurrentVersion**
 - Eigenschaft von
 - VcGantt* 741
- CurrentVerticalPagesCount**
 - Eigenschaft von
 - VcPrinter* 1268
- CurrentZoomFactor**
 - Eigenschaft von
 - VcPrinter* 1268
- Curve**
 - siehe auch
 - VcCurve* 574
- CurveByIndex**
 - Methode von
 - VcCurveCollection* 609
- CurveByName**
 - Methode von
 - VcCurveCollection* 610
- CurveCollection**
 - Eigenschaft von
 - VcHistogram* 1024
 - siehe auch
 - VcCurveCollection* 606
- CurveSource**
 - Eigenschaft von
 - VcCurve* 576
- CurveType**
 - Eigenschaft von
 - VcCurve* 576
- CuttingMarks**
 - Eigenschaft von
 - VcPrinter* 1268

D

- DataDefinition**
 - Eigenschaft von
 - VcGantt* 742
 - siehe auch
 - VcDataDefinition* 613
- DataDefinitionTable**
 - Eigenschaft von
 - VcFilter* 702
 - siehe auch

VcDataDefinitionTable 614, 619

DataField

Eigenschaft von

VcDataRecord 625

VcGroup 965

VcLink 1165

VcNode 1214

DataFieldID

Eigenschaft von

VcField 700

DataFieldIndex

Eigenschaft von

VcFilterSubCondition 716

DataFieldValue

Eigenschaft von

VcMapEntry 1204

DataObject 460

Clear 462

DropEndDate 461

DropStartDate 461

Files 461

GetData 462

GetFormat 464

SetData 465

DataObjectFiles 467

_NewEnum 467

Add 469

Clear 469

Count 468

Item 468

Remove 469

DataRecord

Methode von

VcGroup 971

VcLink 1167

VcNode 1219

siehe auch

VcDataRecord 624

DataRecordByID

Methode von

VcDataRecordCollection 632

DataRecordCollection

Eigenschaft von

VcDataTable 636

siehe auch

VcDataRecordCollection 630

DataRecordEventsEnabled

Eigenschaft von

VcResourceScheduler2 1302

DataTable

siehe auch

VcDataTable 636

DataTableByIndex

Methode von

VcDataTableCollection 642

DataTableByName

Methode von

VcDataTableCollection 642

DataTableCollection

Eigenschaft von

VcGantt 742

siehe auch

VcDataTableCollection 639

DataTableField

siehe auch

VcDataTableField 645

DataTableFieldByIndex

Methode von

VcDataTableFieldCollection 655

DataTableFieldByName

Methode von

VcDataTableFieldCollection 655

DataTableFieldCollection

Eigenschaft von

- VcDataTable* 637
 - siehe auch
 - VcDataTableFieldCollection* 652
- DataTableName**
 - Eigenschaft von*
 - VcDataRecord* 626
 - VcDataTableField* 646
- Date**
 - Eigenschaft von*
 - VcDateLine* 658
- DateDataFieldIndex**
 - Eigenschaft von*
 - VcDateLine* 659
- DateFormat**
 - Eigenschaft von*
 - VcDataTableField* 646
 - VcDefinitionField* 696
 - VcPrinter* 1269
- Dateiliste 461**
- Dateinamen**
 - Anzahl 468
 - entfernen 469
 - hinzufügen 469
 - Index 468
 - löschen 469
- DateLine**
 - siehe auch
 - VcDateLine* 657
- DateLineByIndex**
 - Methode von*
 - VcDateLineCollection* 672
- DateLineByName**
 - Methode von*
 - VcDateLineCollection* 673
- DateLineCollection**
 - Eigenschaft von*
 - VcGantt* 742
- siehe auch
 - VcDateLineCollection* 669
- DateLineGrid**
 - Eigenschaft von*
 - VcSection* 1377
 - siehe auch
 - VcDateLineGrid* 676
- DateLineGridByIndex**
 - Methode von*
 - VcDateLineGridCollection* 692
- DateLineGridByName**
 - Methode von*
 - VcDateLineGridCollection* 692
- DateLineGridCollection**
 - Eigenschaft von*
 - VcGantt* 743
 - siehe auch
 - VcDateLineGridCollection* 689
- DateLineGridName**
 - Eigenschaft von*
 - VcGroupLevelLayout* 983
- DateLineGridsWithChildGroups**
 - Eigenschaft von*
 - VcGroupLevelLayout* 984
- DateLineName**
 - Eigenschaft von*
 - VcGroupLevelLayout* 984
 - VcNodeLevelLayout* 1230
- DateLinesWithChildGroups**
 - Eigenschaft von*
 - VcGroupLevelLayout* 984
- Daten**
 - aus Datei einlesen 837
 - bearbeiten 398, 403, 431
 - speichern 842
 - zum DataObject hinzufügen 465
- Datenanbindung 30**

Datenänderung

ohne Auswertung 747

Datendatei

temporär 221

Datendefinitionstabellen 613

Anzahl der Felder 615, 620

Datumsformat eines Feldes 697

Feld zur Laufzeit hinzufügenr 615, 620

Index eines Feldes 698

Name eines Feldes 698

Typ eines Feldes 699

Zugriff auf Feld über Index 616, 621

Zugriff auf Feld über Namen 617, 621

Datenfeld

editierbar 697

für Tooltiptext 233, 776

Identifikation 700

versteckt 698

Datenfelder

Knoten 398

Datensatz

abhängiger Datensatz nicht gefunden 875

aktualisieren 629

Aktualisierung 635

alle Daten 625

Anzahl in Collection 631

aus Collection entfernen 634

datenbasiertes Objekt 627

Datenfeld 625

eindeutige ID 633

Ereignis Veränderung 874

Ereignis Veränderung abgeschlossen 874

erzeugen 871

ID 626

Iteration, Enumerator Objekt 631

Iteration, Erstwert 633

Iteration, Folgewert 634

löschen 627, 872, 873

Name der zugehörigen Tabelle 626

über ID 633

zu Collection hinzufügen 632

zugeordneter Datensatz 628

Datentabelle

Aktualisierung 644

Anzahl in Collection 640

Auflistungsobjekt 742

Beschreibung 637

Datensatz-Collection 636

Erweiterte Datentabellen setzen 752

für Knoten 232

innerhalb der Collection kopieren 641

Iteration, Enumerator Objekt 640

Iteration, Erstwert 643

Iteration, Folgewert 643

Name 812

Name 638

Tabellendatenfeld-Collection 637

über Index 642

über Name 642

zu Collection hinzufügen 640

Datentabellen 73

erweiterte Nutzung 221

verwalten 259

Datentabellenfeld

Datentyp 650

Datumsformat 646

editierbar 647

Index 812

Index 648

Index des Bezugsfeldes 649

Name 811

- Name 648
- Primärschlüssel 649
- versteckt 647
- zugehöriger Tabellename 646
- DateOutputFormat**
 - Eigenschaft von
 - VcGantt 743
 - VcLineFormatField 1151
 - VcRibbon 1354
- DatesWithHourAndMinute**
 - Eigenschaft von
 - VcFilter 703
- Datum**
 - des linken bzw. des rechten Randes
der aktuellen Darstellung 821,
822, 823
 - zu einer X-Koordinate 823
- Datumsausgabeformat 218, 744**
- DayInEndMonth**
 - Eigenschaft von
 - VcInterval 1048
- DayInStartMonth**
 - Eigenschaft von
 - VcInterval 1048
- DefaultOperationMaximumInterruptionTime**
 - Eigenschaft von
 - VcResourceScheduler2 1302
- DefaultPrinterName**
 - Eigenschaft von
 - VcPrinter 1270
- DefaultResourceCalendarName**
 - Eigenschaft von
 - VcResourceScheduler2 1303
- DefinitionField**
 - siehe auch
 - VcDefinitionField 696
- DefinitionTable**
 - Eigenschaft von
 - VcDataDefinition 613
- DelayTime**
 - Eigenschaft von
 - VcUpdateBehaviorContext 1438
- DeleteDataRecord**
 - Methode von
 - VcDataRecord 627
- DeleteEntry**
 - Methode von
 - VcMap 1193
- DeleteGroup**
 - Methode von
 - VcGroup 972
- DeleteHistogram**
 - Methode von
 - VcHistogramCollection 1034
- DeleteLink**
 - Methode von
 - VcLink 1167
- DeleteLinkRecord**
 - Methode von
 - VcGantt 811
- DeleteNode**
 - Methode von
 - VcNode 1219
- DeleteNodeRecord**
 - Methode von
 - VcGantt 811
- DeletePoint**
 - Methode von
 - VcCurve 597
- DeletePointAsVariant**
 - Methode von
 - VcCurve 598
- Description**
 - Eigenschaft von

- VcDataTable* 637
- DetectDataTableFieldName**
 - Methode von
 - VcGantt* 811
- DetectDataTableName**
 - Methode von
 - VcGantt* 812
- DetectFieldIndex**
 - Methode von
 - VcGantt* 812
- DetermineIDOfFirstOperationByTaskID**
 - Methode von
 - VcResourceScheduler2* 1351
- DetermineIDOfLastOperationByTaskID**
 - Methode von
 - VcResourceScheduler2* 1351
- DiagramAlternatingRowBackColor**
 - Eigenschaft von
 - VcGantt* 745
- DiagramBackColor**
 - Eigenschaft von
 - VcGantt* 745
- DiagramEnabled**
 - Eigenschaft von
 - VcPrinter* 1270
- DiagramHistogramHeightRatio**
 - Eigenschaft von
 - VcGantt* 746
- DiagramHistogramHeightRatioEx**
 - Eigenschaft von
 - VcGantt* 746
- Diagramm**
 - Ausrichtung 418
 - exportieren 66, 429
 - Hintergrundfarbe 745, 806, 807
 - in Datei speichern 817
 - leeren 809
 - speichern 848
 - Tabelle und GanttGraph anzeigen/nicht anzeigen 747
 - Titel/Tabelle/Zeitskala wiederholen 416
 - vollständig leeren 810
 - zweite Hintergrundfarbe für alternierende Streifen 745
- Diagrammhintergrundfarbe** 248
- DiagramVisible**
 - Eigenschaft von
 - VcGantt* 747
- Dialogfeld**
 - Aktualisierungsverhalte bearbeiten 257
 - Aktualisierungsverhalte verwalten 256
 - Druckvorschau 421
 - Gruppen bearbeiten 403
 - Gruppierung 290
 - Linienformate verwalten 283, 285
 - Seite einrichten 415
 - Verbindung bearbeiten 400
 - Vorgänge bearbeiten 398
 - Zuordnung einstellen 311
- Dialogfelder**
 - Schriftattribute 747
- DialogFont**
 - Eigenschaft von
 - VcGantt* 747
- DirectDataWritingModeEnabled**
 - Eigenschaft von
 - VcGantt* 747
- DocumentName**
 - Eigenschaft von
 - VcPrinter* 1271

Doppelklick

auf Gruppe 404, 405

auf Zeitskala 410

**Double Ausgabeformat Grundeinheit
Num. Skala 1254**

Doubleausgabeformat 220

DoubleOutputFormat

Eigenschaft von

VcGantt 748

VcNumericScale 1243

DropEndDate

Eigenschaft von

DataObject 461

DropStartDate

Eigenschaft von

DataObject 461

Druckdatum 420

Drucken 65, 429

Absolute Breite des linken
Seitenrandes in cm 1263

Absolute Breite des linken
Seitenrandes in Zoll 1263

Absolute Breite des rechten
Seitenrandes in cm 1264

Absolute Breite des rechten
Seitenrandes in Zoll 1264

Absolute Höhe des oberen
Seitenrandes in cm 1265

Absolute Höhe des oberen
Seitenrandes in Zoll 1265

Absolute Höhe des unteren
Seitenrandes in cm 1262

Absolute Höhe des unteren
Seitenrandes in Zoll 1263

aktueller Drucker 1270

an Seitenzahlvorgabe anpassen 416

Anzahl der Tabellenspalten 1282

Art der Seitennummerierung 1277

auslösen 840

Ausrichtung 1265

Datumsformat für Dialog Seite
einrichten 1270

Diagramm 1270

Diagramm drucken 416, 417

direkt 839

Dokumentenname 1271

Druckdatum 1279

Drucker einrichten 839

Druckereigenschaften setzen bzw.
ausfragen 784

Druckernamen festlegen 1280

Druckvorschau 840

Enddatum des gedruckten
Zeitraumes 417

Enddatum des zu druckenden
Zeitraumes 1283

Faltmarkierungen 1273

Gruppen automatisch neu optimieren
1280

Hoch- oder Querformat 1275

in eine Datei 840

max. Anzahl von Seiten (horizontal)
1275

max. Anzahl von Seiten (vertikal)
1275

mehrere Diagramme auf einmal 1267

Papiergröße 1279

Probleme 450

Rahmen 1277

Schnittmarkierungen 1268

Seitenbeschriftung 1276

Seitennummern 1278

Skalierungsmodus 1281

Startdatum des gedruckten
Zeitraumes 417

Startdatum des zu druckenden
Zeitraumes 1284

Tabellenbreite 1283

Tabellenbreite wie auf dem Bildschirm 416

Tabellenspalten 417

tatsächlicher Zoomfaktor 1268

Titel, Legende, Tabelle und Zeitskala auf jeder Seite wiederholen 1281

Zoomen mit horizontaler Anpassung 416

Zoomfaktor 416, 1285

Drucker einrichten 428

Druckvorschau 421, 429, 1282

DumpConfiguration

Methode von

VcGantt 813

Duration

Eigenschaft von

VcInterval 1048

DurationDataFieldIndex

Eigenschaft von

VcLayer 1073

VcScheduler 1369

E

EarlyEndDateDataFieldIndex

Eigenschaft von

VcScheduler 1369

EarlyStartDateDataFieldIndex

Eigenschaft von

VcScheduler 1369

Editable

Eigenschaft von

VcDataTableField 647

VcDefinitionField 697

EditGroup

Methode von

VcGantt 814

Editieren

in Gruppenfeldern der Tabelle 762

in Gruppenfeldern des Diagramms 761

in Knotenfeldern der Tabelle 763

in Knotenfeldern des Diagramms 762

EditLink

Methode von

VcGantt 814

EditNewNode

Eigenschaft von

VcGantt 749

EditNode

Methode von

VcGantt 814

Eigenschaften

_NewEnum

DataObjectFiles 467

VcBoxCollection 496

VcBoxFormat 503

VcBoxFormatCollection 508

VcCalendarCollection 533

VcCalendarGridCollection 558

VcCalendarProfileCollection 569

VcCurveCollection 607

VcDataDefinitionTable 614, 619

VcDataRecordCollection 630

VcDataTableCollection 639

VcDataTableFieldCollection 652

VcDateLineCollection 669

VcDateLineGridCollection 689

VcFilter 702

VcFilterCollection 708

VcGroupCollection 974

VcGroupLevelLayoutCollection 1006

VcHistogramCollection 1032

VcIntervalCollection 1063

- VcLayerCollection* 1109
- VcLayerFormat* 1115
- VcLineFormat* 1140
- VcLineFormatCollection* 1144
- VcLinkAppearanceCollection* 1179
- VcLinkCollection* 1185
- VcMap* 1189
- VcMapCollection* 1196
- VcNodeCollection* 1225
- VcNumericScaleCollection* 1256
- VcTableCollection* 1386
- VcTableFormat* 1391
- VcTableFormatCollection* 1396
- VcTimeScaleCollection* 1424
- VcUpdateBehaviorCollection* 1431
- AbsoluteBottomMarginInCM*
 - VcPrinter* 1262
- AbsoluteBottomMarginInInches*
 - VcPrinter* 1263
- AbsoluteLeftMarginInCM*
 - VcPrinter* 1263
- AbsoluteLeftMarginInInches*
 - VcPrinter* 1263
- AbsoluteRightMarginInCM*
 - VcPrinter* 1264
- AbsoluteRightMarginInInches*
 - VcPrinter* 1264
- AbsoluteTopMarginInCM*
 - VcPrinter* 1265
- AbsoluteTopMarginInInches*
 - VcPrinter* 1265
- Active*
 - VcCalendarCollection* 534
 - VcHistogramCollection* 1033
 - VcNumericScaleCollection* 1257
 - VcTableCollection* 1387
 - VcTimeScaleCollection* 1425
 - VcUpdateBehaviorCollection* 1432
- ActiveNodeFilter*
 - VcGantt* 728
- ActualEndDateDataFieldIndex*
 - VcScheduler* 1368
- ActualStartDateDataFieldIndex*
 - VcScheduler* 1368
- Addend*
 - VcCurve* 575
- AdjustToReferenceDate*
 - VcDateLineGrid* 677
- Alignment*
 - VcBoundingBox* 472
 - VcBoxFormatField* 514
 - VcLayerFormatField* 1119
 - VcLineFormatField* 1151
 - VcPrinter* 1265
 - VcTableFormatField* 1401
- AllBorderBoxesShownOnCombinedControls*
 - VcPrinter* 1266
- AllData*
 - VcDataRecord* 624
 - VcLink* 1164
 - VcNode* 1214
- AllowMultipleBoxMarking*
 - VcGantt* 728
- AllowNewBoxes*
 - VcGantt* 729
- AllowNewNodes*
 - VcGantt* 729
- AllowNumericScaleRescale*
 - VcGantt* 730
- AllowPanningMode*
 - VcGantt* 730
- AllowSelectionViaRubberRect*
 - VcGantt* 731

- AllowTableColumnWidthOptimization*
 - VcGantt* 731
- AllowTimescaleRescale*
 - VcGantt* 732
- AllowVerticalGroupMovementViaDiagram*
 - VcGroupLevelLayout* 980
- AllowVerticalGroupMovementViaTable*
 - VcGroupLevelLayout* 980
- AllowVerticalNodeMovement*
 - VcGantt* 732
- AllowVerticalNodeMovementViaTable*
 - VcGantt* 732
- AlwaysCurrentDate*
 - VcDateLine* 658
- AnchoringInteractionsAllowed*
 - VcBox* 481
- AnchoringLineVisible*
 - VcBox* 481
- AnnotationAtBottom*
 - VcDateLineGrid* 677
- AnnotationAtCenter*
 - VcDateLineGrid* 678
- AnnotationAtTop*
 - VcDateLineGrid* 678
- Arrangement*
 - VcGantt* 733
- ArrowKeyMode*
 - VcGantt* 733
- ArrowKeyStepSizeMultiplier*
 - VcGantt* 735
- AssignCalendarToNodes*
 - VcGantt* 735
- AssignmentDataTableName*
 - VcResourceScheduler2* 1292
- AssignmentIsResultFieldIndex*
 - VcResourceScheduler2* 1294
- AssignmentIsVisibleFieldIndex*
 - VcResourceScheduler2* 1294
- AssignmentLoadOrConsumptionPerItemFieldIndex*
 - VcResourceScheduler2* 1295
- AssignmentMaximumLoadFieldIndex*
 - VcResourceScheduler2* 1295
- AssignmentMinimumLoadFieldIndex*
 - VcResourceScheduler2* 1296
- AssignmentMinimumMaximumLoadType*
 - VcResourceScheduler2* 1297
- AssignmentOperationIDFieldIndex*
 - VcResourceScheduler2* 1297
- AssignmentResourceIDFieldIndex*
 - VcResourceScheduler2* 1298
- AssignmentResourceSelectionStrategyFieldIndex*
 - VcResourceScheduler2* 1299
- AutoCollapseGroups*
 - VcGroupLevelLayout* 980
 - VcHierarchyLevelLayout* 1014
- AutoExpandTargetGroup*
 - VcGroupLevelLayout* 981
 - VcHierarchyLevelLayout* 1014
- AutomaticSchedulingEnabled*
 - VcScheduler* 1368
- BackColorAsARGB*
 - VcCalendarGrid* 540
 - VcInterval* 1047
 - VcLayer* 1070
- BackColorDataFieldIndex*
 - VcCalendarGrid* 541
 - VcLayer* 1070
- BackColorMapName*
 - VcCalendarGrid* 541
 - VcLayer* 1071

- BackgroundColor*
 - VcTimeScale* 1418
- BarSeparationGroupBy*
 - VcGantt* 736
- BaseCalendarUsageForSupplementTimes*
 - VcResourceScheduler2* 1300
- BaseTimeUnit*
 - VcResourceScheduler2* 1301
- BaseTimeUnitsPerStep*
 - VcResourceScheduler2* 1301
- BodiesCollapsed*
 - VcGroupLevelLayout* 981
 - VcHierarchyLevelLayout* 1014
- BodiesCollapsedDataFieldIndex*
 - VcGroupLevelLayout* 982
 - VcHierarchyLevelLayout* 1015
- BodiesCollapsedMapName*
 - VcGroupLevelLayout* 982
 - VcHierarchyLevelLayout* 1015
- BodyCollapsed*
 - VcGroup* 965
- Border*
 - VcLegendView* 1131
 - VcWorldView* 1441
- BorderArea*
 - VcGantt* 737
- BorderColor*
 - VcLegendView* 1132
 - VcWorldView* 1442
- Bottom*
 - VcRect* 1286
- BottomMargin*
 - VcLayerFormatField* 1119, 1120
 - VcTableFormatField* 1401
- BoxCollection*
 - VcGantt* 737
- BoxFormatCollection*
 - VcGantt* 737
- CalendarCollection*
 - VcGantt* 738
- CalendarGridCollection*
 - VcGantt* 738
- CalendarGridEx*
 - VcSection* 1375
- CalendarGridName*
 - VcGroupLevelLayout* 982
 - VcNodeLevelLayout* 1230
- CalendarGridsWithChildGroups*
 - VcGroupLevelLayout* 983
- CalendarName*
 - VcCalendarGrid* 541
 - VcHistogram* 1024
 - VcRibbon* 1354
- CalendarNameDataFieldIndex*
 - VcCalendarGrid* 542
 - VcGroupLevelLayout* 983
- CalendarNameMapName*
 - VcCalendarGrid* 542
- CalendarProfileCollection*
 - VcCalendar* 526
 - VcGantt* 738
- CalendarProfileName*
 - VcInterval* 1047
- Collapse*
 - VcSection* 1376
- CollapseColumn*
 - VcTableFormat* 1391
- ColorAsARGB*
 - VcMapEntry* 1203
- ColumnTitle*
 - VcTable* 1381
- ColumnWidth*
 - VcTable* 1382

- CombiField*
 - VcTableFormatField* 1402
- CombiningControlsEnabled*
 - VcPrinter* 1267
- ComparisonValueAsString*
 - VcFilterSubCondition* 714
- CompletionDataFieldIndex*
 - VcLayer* 1072
- ConfigurationName*
 - VcGantt* 739
- ConnectionOperator*
 - VcFilterSubCondition* 715
- ConsiderFilterEntries*
 - VcMap* 1190
- ConsiderLinkRelationTypesOnNodeDragging*
 - VcGantt* 740
- ConstantText*
 - VcLayerFormatField* 1120
 - VcLineFormatField* 1151
 - VcTableFormatField* 1402
- ContextMenuForBoxesEnabled*
 - VcGantt* 740
- Count*
 - DataObjectFiles* 468
 - VcBoxCollection* 497
 - VcBoxFormatCollection* 509
 - VcCalendarCollection* 535
 - VcCalendarGridCollection* 559
 - VcCalendarProfileCollection* 569
 - VcCurveCollection* 607
 - VcDataDefinitionTable* 615, 620
 - VcDataRecordCollection* 631
 - VcDataTableCollection* 640
 - VcDataTableFieldCollection* 653
 - VcDateLineCollection* 670
 - VcDateLineGridCollection* 690
 - VcFilterCollection* 709
 - VcGroupCollection* 975
 - VcGroupLevelLayoutCollection* 1007
 - VcHistogramCollection* 1033
 - VcIntervalCollection* 1063
 - VcLayerCollection* 1110
 - VcLineFormatCollection* 1145
 - VcLinkAppearanceCollection* 1180
 - VcLinkCollection* 1186
 - VcMap* 1190
 - VcMapCollection* 1196
 - VcNodeCollection* 1226
 - VcNumericScaleCollection* 1257
 - VcTableCollection* 1387
 - VcTableFormatCollection* 1397
 - VcTimeScaleCollection* 1425
 - VcUpdateBehaviorCollection* 1432
- CtrlCXVProcessing*
 - VcGantt* 741
- CurrentHorizontalPagesCount*
 - VcPrinter* 1267
- CurrentVersion*
 - VcGantt* 741
- CurrentVerticalPagesCount*
 - VcPrinter* 1268
- CurrentZoomFactor*
 - VcPrinter* 1268
- CurveCollection*
 - VcHistogram* 1024
- CurveSource*
 - VcCurve* 576
- CurveType*
 - VcCurve* 576
- CuttingMarks*
 - VcPrinter* 1268
- DataDefinition*

- VcGantt* 742
- DataDefinitionTable*
 - VcFilter* 702
- DataField*
 - VcDataRecord* 625
 - VcGroup* 965
 - VcLink* 1165
 - VcNode* 1214
- DataFieldID*
 - VcField* 700
- DataFieldIndex*
 - VcFilterSubCondition* 716
- DataFieldValue*
 - VcMapEntry* 1204
- DataRecordCollection*
 - VcDataTable* 636
- DataRecordEventsEnabled*
 - VcResourceScheduler2* 1302
- DataTableCollection*
 - VcGantt* 742
- DataTableFieldCollection*
 - VcDataTable* 637
- DataTableName*
 - VcDataRecord* 626
 - VcDataTableField* 646
- Date*
 - VcDateLine* 658
- DateDataFieldIndex*
 - VcDateLine* 659
- DateFormat*
 - VcDataTableField* 646
 - VcDefinitionField* 696
 - VcPrinter* 1269
- DateLineCollection*
 - VcGantt* 742
- DateLineGrid*
 - VcSection* 1377
- DateLineGridCollection*
 - VcGantt* 743
- DateLineGridName*
 - VcGroupLevelLayout* 983
- DateLineGridsWithChildGroups*
 - VcGroupLevelLayout* 984
- DateLineName*
 - VcGroupLevelLayout* 984
 - VcNodeLevelLayout* 1230
- DateLinesWithChildGroups*
 - VcGroupLevelLayout* 984
- DateOutputFormat*
 - VcGantt* 743
 - VcLineFormatField* 1151
 - VcRibbon* 1354
- DatesWithHourAndMinute*
 - VcFilter* 703
- DayInEndMonth*
 - VcInterval* 1048
- DayInStartMonth*
 - VcInterval* 1048
- DefaultOperationMaximumInterruptionTime*
 - VcResourceScheduler2* 1302
- DefaultPrinterName*
 - VcPrinter* 1270
- DefaultResourceCalendarName*
 - VcResourceScheduler2* 1303
- DefinitionTable*
 - VcDataDefinition* 613
- DelayTime*
 - VcUpdateBehaviorContext* 1438
- Description*
 - VcDataTable* 637
- DiagramAlternatingRowBackColor*
 - VcGantt* 745
- DiagramBackColor*

- VcGantt 745
- DiagramEnabled
 - VcPrinter 1270
- DiagramHistogramHeightRatio
 - VcGantt 746
- DiagramHistogramHeightRatioEx
 - VcGantt 746
- DiagramVisible
 - VcGantt 747
- DialogFont
 - VcGantt 747
- DirectDataWritingModeEnabled
 - VcGantt 747
- DocumentName
 - VcPrinter 1271
- DoubleOutputFormat
 - VcGantt 748
 - VcNumericScale 1243
- DropEndDate
 - DataObject 461
- DropStartDate
 - DataObject 461
- Duration
 - VcInterval 1048
- DurationDataFieldIndex
 - VcLayer 1073
 - VcScheduler 1369
- EarlyEndDateDataFieldIndex
 - VcScheduler 1369
- EarlyStartDateDataFieldIndex
 - VcScheduler 1369
- Editable
 - VcDataTableField 647
 - VcDefinitionField 697
- EditNewNode
 - VcGantt 749
- Enabled
 - VcGantt 749
- EnableSupplyTextEntryEvent
 - VcGantt 749
- EndDataFieldIndex
 - VcLayer 1073
- EndDateForAutomaticScheduling
 - VcScheduler 1370
- EndDateNotLaterThanDataFieldIndex
 - VcScheduler 1370
- EndTime
 - VcInterval 1049
- EndMonth
 - VcInterval 1049
- EndSnapTarget
 - VcCalendarGrid 542
 - VcLayer 1074
- EndTime
 - VcInterval 1049
- EndWeekday
 - VcInterval 1050
- EventReturnStatus
 - VcGantt 750
- EventsSecurityCheck
 - VcGantt 751
- EventText
 - VcGantt 751
- ExtendedDataTables
 - VcGantt 751
- ExtendedEditingBehavior
 - VcGantt 752
- FieldsSeparatedByLines
 - VcBoxFormat 504
 - VcTableFormat 1392
- FieldText
 - VcBox 482
- FilePath
 - VcGantt 752

- Files*
 - DataObject* 461
- Fill2Color*
 - VcCurve* 577
- Fill2Pattern*
 - VcCurve* 578
- Fill2ReferenceName*
 - VcCurve* 581
- FillColor*
 - VcCurve* 582
- FillPattern*
 - VcCurve* 582
- FillReferenceName*
 - VcCurve* 585
- FilterCollection*
 - VcGantt* 753
- FilterName*
 - VcCurve* 586
 - VcFilterSubCondition* 716
 - VcLayer* 1074
 - VcLinkAppearance* 1169
 - VcTableFormat* 1392
- FoldingMarksType*
 - VcPrinter* 1271
- Font*
 - VcDateLine* 659
 - VcNumericScale* 1243
 - VcRibbon* 1356
 - VcTimeScale* 1419
- FontAntiAliasingEnabled*
 - VcGantt* 754
- FontBody*
 - VcMapEntry* 1205
- FontColor*
 - VcDateLine* 659
 - VcNumericScale* 1244
 - VcRibbon* 1357
 - VcTimeScale* 1419
- FontName*
 - VcMapEntry* 1205
- FontSize*
 - VcMapEntry* 1206
- FormatField*
 - VcBoxFormat* 504
 - VcLayerFormat* 1116
 - VcLineFormat* 1141
 - VcTableFormat* 1393
- FormatFieldCount*
 - VcBoxFormat* 505
 - VcLayerFormat* 1116
 - VcLineFormat* 1141
 - VcTableFormat* 1393
- FormatName*
 - VcBox* 482
 - VcBoxFormatField* 515
 - VcDateLineGrid* 678
 - VcLayerFormatField* 1120
 - VcLineFormatField* 1153
 - VcTableFormatField* 1402
- FreeFloatDataFieldIndex*
 - VcScheduler* 1370
- FullUsageOfPlanningUnitsEnabled*
 - VcResourceScheduler2* 1303
- GraphicsFileName*
 - VcBorderBox* 473
 - VcLayer* 1075
 - VcMapEntry* 1206
 - VcTableFormatField* 1403
- GraphicsFileNameDataFieldIndex*
 - VcLayer* 1076
 - VcTableFormatField* 1404
- GraphicsFileNameMapName*
 - VcLayer* 1077
 - VcTableFormatField* 1404

- GraphicsHeight*
 - VcBoxFormatField* 515
 - VcTableFormatField* 1405
- GroupCollection*
 - VcGantt* 754
- GroupDataFieldIndex*
 - VcGroupLevelLayout* 985
- GroupingField*
 - VcGantt* 755
- GroupingLevel*
 - VcGroup* 966
- GroupingModificationsAllowed*
 - VcGantt* 755
- GroupingOrderField*
 - VcGantt* 756
- GroupingSortOrder*
 - VcGantt* 756
- GroupInvisible*
 - VcGroup* 967
- GroupLevelLayoutCollection*
 - VcGantt* 757
- GroupOptimizationOnInteractionsEnabled*
 - VcGantt* 757
- GroupsInvisible*
 - VcGroupLevelLayout* 985
- GroupsInvisibleCollapsedMapName*
 - VcGroupLevelLayout* 985
- GroupsInvisibleDataFieldIndex*
 - VcGroupLevelLayout* 985
- Height*
 - VcLayer* 1078
 - VcLegendView* 1132
 - VcRect* 1286
 - VcWorldView* 1442
- HeightActualValue*
 - VcLegendView* 1133
- VcWorldView* 1443
- HeightDataFieldIndex*
 - VcLayer* 1078
- HeightMapName*
 - VcLayer* 1079
- Hidden*
 - VcDataTableField* 647
 - VcDefinitionField* 698
- HierarchyDataFieldIndex*
 - VcGantt* 758
 - VcHierarchyLevelLayout* 1015
- HierarchyLevelLayout*
 - VcGantt* 759
- Histogram*
 - VcCurve* 587
 - VcNumericScale* 1244
- HistogramCollection*
 - VcGantt* 759
- HistogramSeparationLineColor*
 - VcGantt* 759
- HorAlignment*
 - VcDateLineGrid* 679
- HorizontalOffset*
 - VcLayer* 1079
- hWnd*
 - VcGantt* 760
- ID*
 - VcDataRecord* 626
 - VcDefinitionField* 698
 - VcGroup* 967
 - VcLink* 1165
 - VcNode* 1215
- Identifiable*
 - VcCalendarGrid* 543
 - VcDateLine* 660
- IncomingLinks*
 - VcNode* 1215

*IndentColumn**VcTableFormat* 1394*IndentWidth**VcTableFormat* 1394*Index**VcBoxFormatField* 516*VcDataTableField* 648*VcFilterSubCondition* 717*VcLayerFormatField* 1121*VcLineFormatField* 1153*VcTableFormatField* 1405*InfoWindow**VcGantt* 760*InInteractionEventsEnabled**VcGantt* 760*InPlaceEditingOnGroupsInDiagramEnabled**VcGantt* 761*InPlaceEditingOnGroupsInTableEnabled**VcGantt* 761*InPlaceEditingOnNodesInDiagramEnabled**VcGantt* 762*InPlaceEditingOnNodesInTableEnabled**VcGantt* 763*InteractionMode**VcGantt* 764*IntervalCollection**VcCalendar* 526*VcCalendarProfile* 565*IsEditable**VcUpdateBehavior* 1428*VcUpdateBehaviorContext* 1439*Item**DataObjectFiles* 468*LabelPosition**VcDateLine* 660*LabelSizeDependence**VcLayer* 1080*LateEndDateDataFieldIndex**VcScheduler* 1370*LateStartDateDataFieldIndex**VcScheduler* 1371*LayerCollection**VcGantt* 764*LayerFormat**VcLayer* 1080*LayerName**VcCurve* 587*LayerShape**VcLayer* 1080*Left**VcLegendView* 1133*VcRect* 1287*VcWorldView* 1443*LeftActualValue**VcLegendView* 1134*VcWorldView* 1444*LeftMargin**VcLayerFormatField* 1121, 1122*VcTableFormatField* 1405*Legend**VcMapEntry* 1207*LegendElementsArrangement**VcBoundingBox* 474*LegendElementsBottomMargin**VcBoundingBox* 474*LegendElementsMaximumColumnCount**VcBoundingBox* 474*LegendElementsMaximumRowCount**VcBoundingBox* 475*LegendElementsTopMargin*

- VcBoundingBox* 475
- LegendFont*
 - VcBoundingBox* 475
- LegendText*
 - VcLayer* 1083
- LegendTitle*
 - VcBoundingBox* 476
- LegendTitleFont*
 - VcBoundingBox* 476
- LegendTitleVisible*
 - VcBoundingBox* 477
- LegendView*
 - VcGantt* 764
- Level*
 - VcGroupLevelLayout* 986
- LevelMaximumForPagebreaks*
 - VcHierarchyLevelLayout* 1016
- LineColor*
 - VcBox* 483
 - VcCalendarGrid* 544
 - VcCurve* 587
 - VcDateLine* 660
 - VcDateLineGrid* 679
 - VcInterval* 1050
 - VcLayer* 1083
 - VcLinkAppearance* 1170
 - VcNumericScale* 1245
 - VcSection* 1377, 1378
- LineColorDataFieldIndex*
 - VcCalendarGrid* 544
 - VcDateLineGrid* 680
 - VcLayer* 1084
- LineColorMapName*
 - VcCalendarGrid* 544
 - VcDateLineGrid* 680
 - VcLayer* 1084
- LineFormatCollection*
 - VcGantt* 765
- LineThickness*
 - VcBox* 483
 - VcCalendarGrid* 545
 - VcCurve* 588
 - VcDateLine* 661
 - VcDateLineGrid* 680
 - VcInterval* 1050
 - VcLayer* 1084
 - VcLinkAppearance* 1170
- LineType*
 - VcBox* 484
 - VcCalendarGrid* 546
 - VcCurve* 589
 - VcDateLine* 662
 - VcDateLineGrid* 681
 - VcInterval* 1051
 - VcLayer* 1085
 - VcLinkAppearance* 1171
- LinkAppearanceCollection*
 - VcGantt* 765
- LinkCollection*
 - VcGantt* 765
- LinkDataTableName*
 - VcResourceScheduler2* 1304
- LinkDurationDataFieldIndex*
 - VcScheduler* 1371
- LinkDurationFieldIndex*
 - VcResourceScheduler2* 1306
- LinkPredecessorDataFieldIndex*
 - VcGantt* 766
- LinkPredecessorOperationIDFieldIndex*
 - VcResourceScheduler2* 1306
- LinkPredecessorTaskIDFieldIndex*
 - VcResourceScheduler2* 1307
- LinksDataTableName*

- VcGantt 767
- LinkSuccessorDataFieldIndex
 - VcGantt 767
- LinkSuccessorOperationIDFieldIndex
 - VcResourceScheduler2 1307
- LinkSuccessorTaskIDFieldIndex
 - VcResourceScheduler2 1308
- LinkTypeDataFieldIndex
 - VcGantt 768
- MajorTicks
 - VcNumericScale 1245
 - VcRibbon 1357
- MajorTicksEx
 - VcNumericScale 1245
- MapCollection
 - VcGantt 769
- MarginsShownInInches
 - VcPrinter 1274
- MarkBox
 - VcBox 485
- MarkCurve
 - VcCurve 590
- MarkedNodesFilter
 - VcFilterCollection 709
- MarkGroup
 - VcGroup 967
- MarkingColor
 - VcWorldView 1444
- MarkNode
 - VcNode 1216
- MaxHorizontalPagesCount
 - VcPrinter 1274
- MaximumEndDataFieldIndex
 - VcLayer 1086
- MaximumTextLineCount
 - VcBoxFormatField 516
 - VcTableFormatField 1406
- MaxVerticalPagesCount
 - VcPrinter 1275
- Millimeter
 - VcMapEntry 1208
- MinimumRowHeight
 - VcGantt 769
- MinimumStartDataFieldIndex
 - VcLayer 1087
- MinimumTextLineCount
 - VcBoxFormatField 517
 - VcTableFormatField 1406
- MinimumWidth
 - VcBoxFormatField 517
 - VcLayerFormatField 1122
- MinorTicks
 - VcNumericScale 1246
 - VcRibbon 1357
- MinorTicksEx
 - VcNumericScale 1247
- Mode
 - VcWorldView 1445
- ModificationsAllowed
 - VcGroupLevelLayout 986
- MouseProcessingEnabled
 - VcGantt 770
- Moveable
 - VcBox 486
 - VcDateLine 663
 - VcLayer 1087
- MoveAllSelectedNodes
 - VcGantt 770
- MoveLayersAsNodeWithShiftKey
 - VcGantt 771
- MoveMode
 - VcNode 1216
- MoveNodeAlways
 - VcGantt 771

- MoveNodeWhenMarked*
 - VcGantt* 772
- MultiplePrimaryKeysAllowed*
 - VcDataTable* 638
- MultiState*
 - VcTableFormatField* 1407
- Name*
 - VcBox* 486
 - VcBoxFormat* 505
 - VcCalendar* 526
 - VcCalendarGrid* 547
 - VcCalendarProfile* 566
 - VcCurve* 591
 - VcDataTable* 638
 - VcDataTableField* 648
 - VcDateLine* 663
 - VcDateLineGrid* 682
 - VcDefinitionField* 698
 - VcFilter* 703
 - VcGroup* 968
 - VcGroupLevelLayout* 987
 - VcHistogram* 1024
 - VcInterval* 1052
 - VcLayer* 1087
 - VcLineFormat* 1142
 - VcLinkAppearance* 1172
 - VcMap* 1191
 - VcNumericScale* 1247
 - VcTable* 1382
 - VcTableFormat* 1394
 - VcTimeScale* 1419
 - VcUpdateBehavior* 1429
- NewNodesViaDoubleClick*
 - VcGantt* 772
- NodeCalendarNameDataFieldIndex*
 - VcGantt* 773
- NodeCollection*
 - VcGantt* 773
 - VcGroup* 968
- NodeDurationDataFieldIndex*
 - VcGantt* 773
- NodeEndDateDataFieldIndex*
 - VcGantt* 774
- NodeID*
 - VcBox* 487
- NodeLevelLayout*
 - VcGantt* 774
- NodeRowNumberDataFieldIndex*
 - VcGantt* 774
- NodesDataTableName*
 - VcGantt* 775
- NodeSeparationLinesVisible*
 - VcHierarchyLevelLayout* 1016
- NodesInHeader*
 - VcGroup* 969
- NodesInHeaders*
 - VcGroupLevelLayout* 987
 - VcHierarchyLevelLayout* 1016
- NodesInHeadersDataFieldIndex*
 - VcGroupLevelLayout* 987
 - VcHierarchyLevelLayout* 1017
- NodesInHeadersMapName*
 - VcGroupLevelLayout* 988
 - VcHierarchyLevelLayout* 1017
- NodesOverlaid*
 - VcGroup* 969
 - VcGroupLevelLayout* 988
 - VcHierarchyLevelLayout* 1018
- NodeStartDateDataFieldIndex*
 - VcGantt* 775
- NodeTooltipTextField*
 - VcGantt* 776
- NominalScaleMaximum*
 - VcHistogram* 1025

- NominalScaleMinimum*
 - VcHistogram* 1025
- NonWorkInterval*
 - VcLayer* 1088
- NonWorkIntervalBackColorAsARGB*
 - VcLayer* 1088
- NonWorkIntervalBackColorDataFieldIndex*
 - VcLayer* 1089
- NonWorkIntervalBackColorMapName*
 - VcLayer* 1090
- NonWorkIntervalLineColor*
 - VcLayer* 1091
- NonWorkIntervalLineColorDataFieldIndex*
 - VcLayer* 1091
- NonWorkIntervalLineColorMapName*
 - VcLayer* 1091
- NonWorkIntervalLineThickness*
 - VcLayer* 1092
- NonWorkIntervalLineType*
 - VcLayer* 1093
- NonWorkIntervalPattern*
 - VcLayer* 1094
- NonWorkIntervalPatternColorAsARGB*
 - VcLayer* 1097
- NonWorkIntervalPatternColorDataFieldIndex*
 - VcLayer* 1098
- NonWorkIntervalPatternDataFieldIndex*
 - VcLayer* 1098
- NonWorkIntervalPatternMapName*
 - VcLayer* 1099
- NonWorkIntervalShape*
 - VcLayer* 1100
- NoOfColumns*
- VcTable* 1383
- NoOfInitialRows*
 - VcGantt* 776
- Number*
 - VcMapEntry* 1208
- NumericScaleCollection*
 - VcHistogram* 1026
- ObjectDrawEventsEnabled*
 - VcLayer* 1100
- ObservedDST*
 - VcDateLineGrid* 683
 - VcRibbon* 1358
- OLEDragHorizontalMovementAllowed*
 - VcGantt* 777
- OLEDragMode*
 - VcGantt* 777
- OLEDragViaDiagram*
 - VcGantt* 778
- OLEDragViaTable*
 - VcGantt* 778
- OLEDragWithOwnMouseCursor*
 - VcGantt* 779
- OLEDragWithPhantom*
 - VcGantt* 779
- OLEDropMode*
 - VcGantt* 780
- OperationDataTableName*
 - VcResourceScheduler2* 1309
- OperationLoadPerItemFieldIndex*
 - VcResourceScheduler2* 1310
- OperationMaximumInterruptionTimeFieldIndex*
 - VcResourceScheduler2* 1310
- OperationMinimumSupplementTimeFieldIndex*
 - VcResourceScheduler2* 1311
- OperationOverlapQuantityFieldIndex*

- VcResourceScheduler2* 1312
- OperationPostLoadFieldIndex*
 - VcResourceScheduler2* 1314
- OperationPostOffsetFieldIndex*
 - VcResourceScheduler2* 1314
- OperationPreparationLoadFieldIndex*
 - VcResourceScheduler2* 1315
- OperationPreparationOffsetFieldIndex*
 - VcResourceScheduler2* 1316
- OperationResultEndDateFieldIndex*
 - VcResourceScheduler2* 1316
- OperationResultPostEndDateFieldIndex*
 - VcResourceScheduler2* 1317
- OperationResultPreparationStartDateFieldIndex*
 - VcResourceScheduler2* 1317
- OperationResultProcessingTimeFieldIndex*
 - VcResourceScheduler2* 1318
- OperationResultSelectedTimingResourceIDFieldIndex*
 - VcResourceScheduler2* 1318
- OperationResultStartDateFieldIndex*
 - VcResourceScheduler2* 1319
- OperationResultStatusFieldIndex*
 - VcResourceScheduler2* 1320
- OperationRouteFieldIndex*
 - VcResourceScheduler2* 1320
- OperationSequenceNumberFieldIndex*
 - VcResourceScheduler2* 1321
- OperationStartLockDateFieldIndex*
 - VcResourceScheduler2* 1322
- OperationTaskIDFieldIndex*
 - VcResourceScheduler2* 1322
- OperationWorkInProgressFieldIndex*
 - VcResourceScheduler2* 1323
- Operator*
 - VcFilterSubCondition* 717
- OptimizedNodesSortDataFieldIndex*
 - VcGroupLevelLayout* 988
- OptimizedNodesSortOrder*
 - VcGroupLevelLayout* 989
- Orientation*
 - VcPrinter* 1275
- Origin*
 - VcBox* 487
- OutgoingLinks*
 - VcNode* 1217
- OutputFormatForCenterDate*
 - VcInfoWindow* 1037
- OutputFormatForDuration*
 - VcInfoWindow* 1039
- OutputFormatForEndDate*
 - VcInfoWindow* 1040
- OutputFormatForStartDate*
 - VcInfoWindow* 1041
- OverlaidNodesSortDataFieldIndex*
 - VcGroupLevelLayout* 989
- OverlaidNodesSortOrder*
 - VcGroupLevelLayout* 990
- OverlapLayerEnabled*
 - VcGantt* 781
- OverlapLayerName*
 - VcGantt* 781
- OverloadResultsCalendarName*
 - VcCurve* 591
- PagebreakMode*
 - VcGroupLevelLayout* 990
 - VcHierarchyLevelLayout* 1018
- PageDescription*
 - VcPrinter* 1276
- PageDescriptionString*
 - VcPrinter* 1276

- PageFrame*
 - VcPrinter* 1277
- PageNumberMode*
 - VcPrinter* 1277
- PageNumbers*
 - VcPrinter* 1278
- PagePaddingEnabled*
 - VcPrinter* 1278
- PaperSize*
 - VcPrinter* 1279
- ParentHWnd*
 - VcLegendView* 1134
 - VcWorldView* 1446
- PartialLoadThreshold*
 - VcGantt* 782
- Pattern*
 - VcCalendarGrid* 547
 - VcInterval* 1053
 - VcMapEntry* 1209
- Pattern2Color*
 - VcCurve* 592
- PatternBackgroundColorAsARGB*
 - VcBoxFormatField* 518
 - VcLineFormatField* 1154
 - VcNumericScale* 1248
 - VcRibbon* 1358
 - VcTableFormatField* 1407
- PatternBackgroundColorDataFieldIndex*
 - VcLineFormatField* 1154
 - VcTableFormatField* 1408
- PatternBackgroundColorMapName*
 - VcLineFormatField* 1154
 - VcTableFormatField* 1408
- PatternColor*
 - VcCurve* 592
- PatternColorAsARGB*
- VcBoxFormatField* 519
- VcCalendarGrid* 550
- VcInterval* 1056
- VcLayer* 1101
- VcLineFormatField* 1155
- VcNumericScale* 1248
- VcRibbon* 1359
- VcTableFormatField* 1408
- PatternColorDataFieldIndex*
 - VcCalendarGrid* 551
 - VcLayer* 1101
 - VcLineFormatField* 1155
 - VcTableFormatField* 1409
- PatternColorMapName*
 - VcCalendarGrid* 551
 - VcLayer* 1102
 - VcLineFormatField* 1156
 - VcTableFormatField* 1409
- PatternDataFieldIndex*
 - VcCalendarGrid* 552
- PatternEx*
 - VcBoxFormatField* 519
 - VcLineFormatField* 1156
 - VcNumericScale* 1248
 - VcRibbon* 1359
 - VcTableFormatField* 1410
- PatternExDataFieldIndex*
 - VcLineFormatField* 1160
 - VcTableFormatField* 1413
- PatternExMapName*
 - VcLineFormatField* 1160
 - VcTableFormatField* 1413
- PatternMapName*
 - VcCalendarGrid* 552
- Period*
 - VcDateLineGrid* 683
- PhantomLayerHeight*

- VcGantt 783
- PlanningEndDate
 - VcResourceScheduler2 1324
- PlanningStartDate
 - VcResourceScheduler2 1324
- PlanningStrategy
 - VcResourceScheduler2 1325
- PointsEquidistant
 - VcCurve 593
- Position
 - VcRibbon 1363
 - VcTable 1383
- PredecessorLayerName
 - VcLinkAppearance 1173
- PredecessorNode
 - VcLink 1166
- PrePortSymbol
 - VcLinkAppearance 1173
- PrimaryKey
 - VcDataTableField 649
- PrintDate
 - VcPrinter 1279
- Printer
 - VcGantt 784
- PrinterName
 - VcPrinter 1280
- Priority
 - VcBox 488
 - VcCalendarGrid 552
 - VcDateLine 664
 - VcDateLineGrid 684
 - VcLayerFormatField 1122
- ReferenceDate
 - VcDateLineGrid 684
 - VcInfoWindow 1043
 - VcRibbon 1363
- ReferencePoint
 - VcBox 488
- RelationshipFieldIndex
 - VcDataTableField 649
- ReOptimizeNodesInGroupsEnabled
 - VcPrinter 1280
- RepeatTableTimeScale
 - VcPrinter 1281
- Resizing
 - VcBox 489
- ResourceCalendarNameFieldIndex
 - VcResourceScheduler2 1326
- ResourceCapacityType
 - VcResourceScheduler2 1327
- ResourceCapacityTypeFieldIndex
 - VcResourceScheduler2 1327
- ResourceConstraintTypeFieldIndex
 - VcResourceScheduler2 1329
- ResourceDataTableName
 - VcResourceScheduler2 1329
- ResourceEfficiencyFieldIndex
 - VcResourceScheduler2 1331
- ResourceGroupDataTableName
 - VcResourceScheduler2 1332
- ResourceGroupIDFieldIndex
 - VcResourceScheduler2 1333
- ResourceNameFieldIndex
 - VcResourceScheduler2 1334
- ResourceResultLoadCurveNamePrefix
 - VcResourceScheduler2 1335
- ResourceResultStockCurveNamePrefix
 - VcResourceScheduler2 1336
- ResourceScheduler2
 - VcGantt 784
- ResourceSelectionStrategy
 - VcResourceScheduler2 1336

- ResourceType*
 - VcResourceScheduler2* 1338
- RestoreAutoCollapsedGroups*
 - VcGroupLevelLayout* 990
 - VcHierarchyLevelLayout* 1018
- RestoreAutoExpandedGroups*
 - VcGroupLevelLayout* 991
 - VcHierarchyLevelLayout* 1019
- ResultProcessingStepCount*
 - VcResourceScheduler2* 1339
- Ribbon*
 - VcSection* 1378
 - VcTimeScale* 1420
- Right*
 - VcRect* 1288
- RightMargin*
 - VcLayerFormatField* 1123
 - VcTableFormatField* 1414
- RightTable*
 - VcGantt* 784
- RightTableDiagramWidthRatio*
 - VcGantt* 785
- RightTableDiagramWidthRatioEx*
 - VcGantt* 785
- RoundedLinkSlantsEnabled*
 - VcGantt* 786
- RoutingType*
 - VcLinkAppearance* 1174
- RowBackColorAsARGB*
 - VcGroupLevelLayout* 991
 - VcHistogram* 1026
 - VcNodeLevelLayout* 1230
- RowBackColorDataFieldIndex*
 - VcGroupLevelLayout* 992
 - VcNodeLevelLayout* 1231
- RowBackColorMapName*
 - VcGroupLevelLayout* 992
- VcNodeLevelLayout* 1231
- RowHeightReductionEnabled*
 - VcGantt* 786
- RowMargins*
 - VcGantt* 787
- RowPattern*
 - VcGroupLevelLayout* 992
 - VcHistogram* 1026
 - VcNodeLevelLayout* 1231
- RowPatternColorAsARGB*
 - VcGroupLevelLayout* 996
 - VcHistogram* 1027
 - VcNodeLevelLayout* 1235
- RowPatternColorDataFieldIndex*
 - VcGroupLevelLayout* 996
 - VcNodeLevelLayout* 1235
- RowPatternColorMapName*
 - VcGroupLevelLayout* 997
 - VcNodeLevelLayout* 1236
- RowPatternDataFieldIndex*
 - VcGroupLevelLayout* 997
 - VcNodeLevelLayout* 1236
- RowPatternMapName*
 - VcGroupLevelLayout* 997
 - VcNodeLevelLayout* 1236
- RowsBelowCollapsed*
 - VcGroup* 969
- Sash3DStyleEnabled*
 - VcGantt* 787
- SashThickness*
 - VcGantt* 787
- ScalingMode*
 - VcPrinter* 1281
- ScheduledProjectEndDate*
 - VcScheduler* 1371
- ScheduledProjectStartDate*
 - VcScheduler* 1372

- Scheduler*
 - VcGantt* 788
- ScheduleSuccessorsOnlyEnabled*
 - VcScheduler* 1372
- ScrollBarMode*
 - VcLegendView* 1135
 - VcWorldView* 1446
- ScrollEventsEnabled*
 - VcGantt* 788
- SecondsPerWorkday*
 - VcCalendar* 527
- Section*
 - VcTimeScale* 1420
- SelectedRowBackColorAsARGB*
 - VcGantt* 788
- SeparationLineColor*
 - VcGroupLevelLayout* 998
 - VcHierarchyLevelLayout* 1019
 - VcNodeLevelLayout* 1237
 - VcTableFormat* 1395
- SeparationLineColorDataFieldIndex*
 - VcGroupLevelLayout* 998
- SeparationLineColorMapName*
 - VcGroupLevelLayout* 998
- SeparationLineInterval*
 - VcNodeLevelLayout* 1237
- SeparationLineThickness*
 - VcGroupLevelLayout* 999
 - VcHierarchyLevelLayout* 1020
 - VcNodeLevelLayout* 1237
- SeparationLineType*
 - VcGroupLevelLayout* 1000
 - VcHierarchyLevelLayout* 1020
 - VcNodeLevelLayout* 1238
- ShowCalendarGrids*
 - VcGroupLevelLayout* 1001
 - VcHistogram* 1027
- VcNodeLevelLayout* 1239
- VcTimeScale* 1421
- ShowDateGrids*
 - VcTimeScale* 1421
- ShowDateLineGrids*
 - VcGroupLevelLayout* 1001
- ShowDateLines*
 - VcGroupLevelLayout* 1002
 - VcNodeLevelLayout* 1240
- ShowGroupNodes*
 - VcGroupLevelLayout* 1002
- ShowNonWorkInterval*
 - VcGantt* 789
- ShowSeparationLines*
 - VcGroupLevelLayout* 1002
 - VcHierarchyLevelLayout* 1022
 - VcNodeLevelLayout* 1240
- ShowSeparationLinesAtTop*
 - VcGroupLevelLayout* 1003
 - VcNodeLevelLayout* 1240
- ShowSnapLines*
 - VcGantt* 789
- ShowSnapMarkings*
 - VcGantt* 790
- ShowTimeScaleDialog*
 - VcGantt* 790
- ShowToolTip*
 - VcGantt* 791
- Sizeable*
 - VcLayer* 1102
- SnapTarget*
 - VcCalendarGrid* 553
 - VcDateLine* 664
 - VcDateLineGrid* 685
- SnapTargetMode*
 - VcNode* 1217
- SnapTargetNodesSelectionMode*

- VcGantt 791
- SortDataFieldIndex
 - VcGroupLevelLayout 1003
 - VcNodeLevelLayout 1241
- SortField
 - VcGantt 792
- SortOrder
 - VcGantt 792
 - VcGroupLevelLayout 1003
 - VcNodeLevelLayout 1241
- Specification
 - VcBox 489
 - VcBoxFormat 506
 - VcCalendar 527
 - VcCalendarGrid 553
 - VcCalendarProfile 566
 - VcCurve 593
 - VcDateLine 665
 - VcFilter 703
 - VcGroupLevelLayout 1004
 - VcInterval 1056
 - VcLayer 1103
 - VcLineFormat 1142
 - VcLinkAppearance 1175
 - VcMap 1191
 - VcUpdateBehavior 1429
- StackReferenceName
 - VcCurve 594
- StartDataFieldIndex
 - VcLayer 1103
- StartDate
 - VcSection 1379
- StartDateForAutomaticScheduling
 - VcScheduler 1373
- StartDateNotEarlierThanDataFieldIndex
 - VcScheduler 1373
- StartDateTime
 - VcInterval 1057
- StartMonth
 - VcInterval 1057
- StartSnapTarget
 - VcCalendarGrid 554
 - VcLayer 1103
- StartTime
 - VcInterval 1058
- StartupSinglePage
 - VcPrinter 1282
- StartWeekday
 - VcInterval 1058
- StringsCaseSensitive
 - VcFilter 704
- SubCondition
 - VcFilter 704
- SubConditionCount
 - VcFilter 705
- SubGroups
 - VcGroup 970
- SubRowMargins
 - VcGantt 793
- SuccessorLayerName
 - VcLinkAppearance 1176
- SuccessorNode
 - VcLink 1166
- SuccPortSymbol
 - VcLinkAppearance 1176
- SummaryBarsVisible
 - VcGantt 793
 - VcGroupLevelLayout 1004
 - VcHierarchyLevelLayout 1022
- SuperGroup
 - VcGroup 970
 - VcNode 1218
- SuppressTruncatedText

- VcLayerFormatField* 1124
- Table*
 - VcGantt* 794
- TableCollection*
 - VcGantt* 794
- TableColumnRanges*
 - VcPrinter* 1282
- TableDiagramWidthRatio*
 - VcGantt* 794
- TableDiagramWidthRatioEx*
 - VcGantt* 795
- TableFormatCollection*
 - VcTable* 1383
- TableWidthAdoptionFromViewOnScreen*
 - VcPrinter* 1283
- TaskDataTableName*
 - VcResourceScheduler2* 1339
- TaskDueDateFieldIndex*
 - VcResourceScheduler2* 1341
- TaskPlanningStrategyFieldIndex*
 - VcResourceScheduler2* 1341
- TaskPriorityFieldIndex*
 - VcResourceScheduler2* 1342
- TaskQuantityFieldIndex*
 - VcResourceScheduler2* 1343
- TaskReleaseDateFieldIndex*
 - VcResourceScheduler2* 1344
- TaskResultEndDateFieldIndex*
 - VcResourceScheduler2* 1345
- TaskResultPostEndDateFieldIndex*
 - VcResourceScheduler2* 1345
- TaskResultPreparationStartDateFieldIndex*
 - VcResourceScheduler2* 1346
- TaskResultProcessingStepFieldIndex*
 - VcResourceScheduler2* 1346
- TaskResultProcessingTimeFieldIndex*
 - VcResourceScheduler2* 1347
- TaskResultRouteFieldIndex*
 - VcResourceScheduler2* 1347
- TaskResultStartDateFieldIndex*
 - VcResourceScheduler2* 1348
- Text*
 - VcBoundingBox* 477
 - VcDateLine* 665
 - VcInterval* 1058
- TextAlignment*
 - VcRibbon* 1363
- TextDataFieldIndex*
 - VcLayerFormatField* 1124
 - VcLineFormatField* 1160
 - VcTableFormatField* 1414
- TextFont*
 - VcBoundingBox* 478
 - VcBoxFormatField* 523
 - VcLayerFormatField* 1125
 - VcLineFormatField* 1161
 - VcTableFormatField* 1415
- TextFontColor*
 - VcBoxFormatField* 523
 - VcLayerFormatField* 1125
 - VcLineFormatField* 1161
 - VcTableFormatField* 1415
- TextFontColorDataFieldIndex*
 - VcLayerFormatField* 1125
 - VcLineFormatField* 1161
 - VcTableFormatField* 1415
- TextFontColorMapName*
 - VcLayerFormatField* 1126
 - VcLineFormatField* 1162
 - VcTableFormatField* 1416
- TextFontDataFieldIndex*
 - VcLayerFormatField* 1126

- VcLineFormatField* 1162
- VcTableFormatField* 1416
- TextFontMapName*
 - VcLayerFormatField* 1126
 - VcLineFormatField* 1163
 - VcTableFormatField* 1416
- TextLineCount*
 - VcLayerFormatField* 1127
 - VcLineFormatField* 1163
- TextLineCountDataFieldIndex*
 - VcLayerFormatField* 1127
- TextLineCountMapName*
 - VcLayerFormatField* 1128
- ThreeDEffect*
 - VcLayer* 1104
 - VcNumericScale* 1252
 - VcTableFormat* 1395
 - VcTimeScale* 1422
- TickColor*
 - VcNumericScale* 1252
 - VcRibbon* 1364
- TickPosition*
 - VcRibbon* 1364
- TimeColumnEndDate*
 - VcPrinter* 1283
- TimeColumnStartDate*
 - VcPrinter* 1284
- TimeScaleAdjustment*
 - VcPrinter* 1284
- TimeScaleCollection*
 - VcGantt* 795
- TimeScaleEnd*
 - VcGantt* 796
- TimeScaleStart*
 - VcGantt* 796
- TimeUnit*
 - VcCurve* 594
- VcGantt* 797
- VcInterval* 1059
- TimeUnitsPerStep*
 - VcGantt* 797
- Title*
 - VcNumericScale* 1252
- ToleranceTimeOnASAPDueDates*
 - VcResourceScheduler2* 1348
- ToleranceTimeOnJITReleaseDates*
 - VcResourceScheduler2* 1349
- ToleranceTimeOnStartLockDates*
 - VcResourceScheduler2* 1349
- ToolTipChangeDuration*
 - VcGantt* 798
- ToolTipDuration*
 - VcGantt* 798
- ToolTipPointerDuration*
 - VcGantt* 799
- ToolTipShowAfterClick*
 - VcGantt* 799
- Top*
 - VcLegendView* 1135
 - VcRect* 1288
 - VcWorldView* 1447
- TopActualValue*
 - VcLegendView* 1136
 - VcWorldView* 1447
- TopMargin*
 - VcLayerFormatField* 1128, 1129
 - VcTableFormatField* 1417
- TotalFloatDataFieldIndex*
 - VcScheduler* 1373
- TrackingSpaceBackColorAsARGB*
 - VcGantt* 800
- TrackingSpacePattern*
 - VcGantt* 800
- TrackingSpacePatternColorAsARGB*

- VcGantt 803
- TurningAnnotationEnabled
 - VcDateLine 666
 - VcDateLineGrid 685
- Type
 - VcBoundingBox 479
 - VcBoxFormatField 523
 - VcCalendar 527
 - VcCalendarProfile 566
 - VcDataTableField 650
 - VcDefinitionField 699
 - VcInterval 1059
 - VcMap 1191
 - VcRibbon 1364
 - VcTableFormatField 1417
 - VcUpdateBehaviorContext 1439
- Unit
 - VcDateLineGrid 686
 - VcNumericScale 1253
 - VcSection 1379
- UnitEx
 - VcNumericScale 1254
- UnitLabel
 - VcNumericScale 1254
- UnitSeparation
 - VcRibbon 1365
- UnitsPerStep
 - VcCurve 595
- UnitWidth
 - VcNumericScale 1255
 - VcSection 1380
- UnitWidthEx
 - VcSection 1380
- UpdateBehaviorCollection
 - VcGantt 804
- UpdateBehaviorName
 - VcBox 490
 - VcCurve 595
 - VcDateLine 666
 - VcNode 1218
 - VcNumericScale 1255
 - VcTable 1384
 - VcTimeScale 1422
 - VcWorldView 1448
- UpdateMode
 - VcUpdateBehaviorContext 1440
- UsedAsOverlapLayer
 - VcLayer 1104
- UseGraphicalAttributes
 - VcInterval 1059
- UseGraphicalAttributesOfIntervals
 - VcCalendarGrid 554
- UseHigherDiagramHistogramHeightRatioPrecision
 - VcGantt 804
- UseHigherTableDiagramWidthRatioPrecision
 - VcGantt 805
- UseReferenceDate
 - VcDateLineGrid 686
 - VcInfoWindow 1043
 - VcRibbon 1366
- UseSnapTargetsInInteractions
 - VcGantt 805
- UseTwinLineSashPhantom
 - VcGantt 806
- ValencyDataFieldIndex
 - VcCurve 596
- VerticalOffset
 - VcLayer 1105
- VerticalOffsetDataFieldIndex
 - VcLayer 1105
- VerticalOffsetMapName
 - VcLayer 1105

- ViewComponentsBackColor*
 - VcGantt* 806
- ViewComponentsBorderColor*
 - VcGantt* 807
- Visible*
 - VcBox* 490
 - VcCalendarGrid* 554
 - VcCurve* 596
 - VcDateLine* 666
 - VcDateLineGrid* 687
 - VcGroup* 971
 - VcGroupLevelLayout* 1005
 - VcHistogram* 1027
 - VcInfoWindow* 1043
 - VcLayer* 1106
 - VcLegendView* 1136
 - VcLinkAppearance* 1177
 - VcTable* 1384
 - VcWorldView* 1448
- VisibleDataFieldIndex*
 - VcCalendarGrid* 555
 - VcDateLine* 667
 - VcDateLineGrid* 687
- VisibleInLegend*
 - VcLayer* 1106
- VisibleMapName*
 - VcCalendarGrid* 555
 - VcDateLine* 667
 - VcDateLineGrid* 687
- WaitCursorEnabled*
 - VcGantt* 807
- Width*
 - VcLegendView* 1137
 - VcRect* 1288
 - VcWorldView* 1448
- WidthActualValue*
 - VcLegendView* 1137
- VcWorldView* 1449
- WindowMode*
 - VcLegendView* 1138
- WorkInProcessType*
 - VcResourceScheduler2* 1350
- WorldView*
 - VcGantt* 807
- WritingDebugFilesEnabled*
 - VcResourceScheduler2* 1350
- ZoomFactor*
 - VcGantt* 808
- ZoomFactorAsDouble*
 - VcPrinter* 1285
- ZoomingPerMouseWheelAllowed*
 - VcGantt* 808
- Eigenschaftenseite**
 - Allgemeines 217
 - Außenbereich 230
 - Knoten 232
 - Layout 245
 - Objekte 250
 - Verbindungen 252
 - Zeitrechnung 254
 - Zusätzliche Ansichten 241
- Einrastwerkzeuge**
 - Linien 790
 - Manuelle Selektion 791
 - Markierung 790
- Einrastziel**
 - Markiermodus für Knoten 1217
- Einrastziele**
 - Verwendung erlaubt/nicht erlaubt 805
- Enabled**
 - Eigenschaft von *VcGantt* 749
- EnableSupplyTextEntryEvent**
 - Eigenschaft von

- VcGantt 749
- EndDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1073
- EndDateForAutomaticScheduling**
 - Eigenschaft von
 - VcScheduler 1370
- EndDateNotLaterThanDataFieldIndex**
 - Eigenschaft von
 - VcScheduler 1370
- EndDateTime**
 - Eigenschaft von
 - VcInterval 1049
- EndLoading**
 - Methode von
 - VcGantt 815
- EndMonth**
 - Eigenschaft von
 - VcInterval 1049
- EndSnapTarget**
 - Eigenschaft von
 - VcCalendarGrid 542
 - VcLayer 1074
- Endtermin**
 - berechnen 35
- EndTime**
 - Eigenschaft von
 - VcInterval 1049
- EndWeekday**
 - Eigenschaft von
 - VcInterval 1050
- Ereignis**
 - Änderungen am XGantt 912
 - Rückgabewert 750
 - Sicherheitsprüfung 751
 - Tool Tip Text 751
- Ereignisse 100**
- Error**
 - VcGantt 852
- ErrorAsVariant**
 - VcGantt 853
- KeyDown**
 - VcGantt 853
- KeyPress**
 - VcGantt 854
- KeyUp**
 - VcGantt 854
- OLECompleteDrag**
 - VcGantt 855
- OLEDragDrop**
 - VcGantt 856
- OLEDragOver**
 - VcGantt 857
- OLEGiveFeedback**
 - VcGantt 858
- OLESetData**
 - VcGantt 858
- OLEStartDrag**
 - VcGantt 859
- OnBoxCreate**
 - VcGantt 860
- OnBoxCreateComplete**
 - VcGantt 861
- OnBoxLClick**
 - VcGantt 861
- OnBoxLDbIClick**
 - VcGantt 861
- OnBoxModify**
 - VcGantt 862
- OnBoxModifyCompleteEx**
 - VcGantt 863
- OnBoxRClick**
 - VcGantt 863
- OnCalendarGridRClick**

- VcGantt 864
- OnCurveLClick
 - VcGantt 865
- OnCurveLDbClick
 - VcGantt 866
- OnCurveModifyComplete
 - VcGantt 866
- OnCurveModifyEx
 - VcGantt 866
- OnCurveModifyEx2
 - VcGantt 867
- OnCurveModifyExAsString
 - VcGantt 868
- OnCurveRClick
 - VcGantt 869
- OnDataRecordCreate
 - VcGantt 870
- OnDataRecordCreateComplete
 - VcGantt 871
- OnDataRecordDelete
 - VcGantt 872
- OnDataRecordDeleteComplete
 - VcGantt 873
- OnDataRecordModify
 - VcGantt 873
- OnDataRecordModifyComplete
 - VcGantt 874
- OnDataRecordNotFound
 - VcGantt 875
- OnDateLineModify
 - VcGantt 875
- OnDateLineRClick
 - VcGantt 875
- OnDeleteCurvePoint
 - VcGantt 876
- OnDeleteCurvePointEx
 - VcGantt 877
- OnDiagramLClick
 - VcGantt 878
- OnDiagramLDbClick
 - VcGantt 878
- OnDiagramRClick
 - VcGantt 879
- OnGroupDelete
 - VcGantt 880
- OnGroupLClick
 - VcGantt 880
- OnGroupLDbClick
 - VcGantt 881
- OnGroupModify
 - VcGantt 881
- OnGroupModifyComplete
 - VcGantt 882
- OnGroupModifyEx
 - VcGantt 883
- OnGroupRClick
 - VcGantt 883
- OnGroupsMark
 - VcGantt 884
- OnGroupsMarkComplete
 - VcGantt 885
- OnHelpRequested
 - VcGantt 885
- OnHistogramLClick
 - VcGantt 886
- OnHistogramLDbClick
 - VcGantt 886
- OnHistogramRClick
 - VcGantt 887
- OnHistogramsHeight
 - VcGantt 888
- OnHistogramsHeightChanged
 - VcGantt 889
- OnHistogramsHeightModifyEx

<i>VcGantt</i> 889	<i>VcGantt</i> 902
<i>OnInsertCurvePoint</i>	<i>OnMouseMove</i>
<i>VcGantt</i> 890	<i>VcGantt</i> 903
<i>OnInsertCurvePointEx</i>	<i>OnMouseUp</i>
<i>VcGantt</i> 890	<i>VcGantt</i> 904
<i>OnInteractionEndComplete</i>	<i>OnNodeCreate</i>
<i>VcGantt</i> 891	<i>VcGantt</i> 905
<i>OnInteractionModeChange</i>	<i>OnNodeCreateCompleteEx</i>
<i>VcGantt</i> 892	<i>VcGantt</i> 905
<i>OnInteractionModeChangeComplete</i>	<i>OnNodeDelete</i>
<i>VcGantt</i> 893	<i>VcGantt</i> 906
<i>OnInteractionObjectChangingComplete</i>	<i>OnNodeDeleteCompleteEx</i>
<i>VcGantt</i> 893	<i>VcGantt</i> 907
<i>OnInteractionStartComplete</i>	<i>OnNodeLClick</i>
<i>VcGantt</i> 895	<i>VcGantt</i> 907
<i>OnLegendViewClosed</i>	<i>OnNodeLDbIClick</i>
<i>VcGantt</i> 896	<i>VcGantt</i> 908
<i>OnLinkCreate</i>	<i>OnNodeModifyComplete</i>
<i>VcGantt</i> 896	<i>VcGantt</i> 909
<i>OnLinkCreateComplete</i>	<i>OnNodeModifyCompleteEx</i>
<i>VcGantt</i> 897	<i>VcGantt</i> 909
<i>OnLinkDelete</i>	<i>OnNodeModifyEx</i>
<i>VcGantt</i> 898	<i>VcGantt</i> 910
<i>OnLinkDeleteComplete</i>	<i>OnNodeRClick</i>
<i>VcGantt</i> 898	<i>VcGantt</i> 911
<i>OnLinkLClickCltn</i>	<i>OnNodeResizeStart</i>
<i>VcGantt</i> 899	<i>VcGantt</i> 912
<i>OnLinkLDbIClickCltn</i>	<i>OnNodesMarkComplete</i>
<i>VcGantt</i> 900	<i>VcGantt</i> 913
<i>OnLinkRClickCltn</i>	<i>OnNodesMarkEx</i>
<i>VcGantt</i> 900	<i>VcGantt</i> 913
<i>OnModifyComplete</i>	<i>OnNumericScaleLClick</i>
<i>VcGantt</i> 901	<i>VcGantt</i> 914
<i>OnMouseDbIClick</i>	<i>OnNumericScaleLDbIClick</i>
<i>VcGantt</i> 902	<i>VcGantt</i> 915
<i>OnMouseDown</i>	<i>OnNumericScaleRClick</i>
	<i>VcGantt</i> 915

- OnNumericScaleRescale*
 - VcGantt* 916
- OnObjectDrawCompleteEx*
 - VcGantt* 917
- OnObjectDrawEx*
 - VcGantt* 918
- OnOptimizeTableColumnWidth*
 - VcGantt* 920
- OnPreScrollComponent*
 - VcGantt* 921
- OnPreScrollDiagramHor*
 - VcGantt* 923
- OnResourceSchedulingProgress*
 - VcGantt* 924
- OnResourceSchedulingWarning*
 - VcGantt* 925
- OnScrollComponent*
 - VcGantt* 928
- OnScrollDiagramHor*
 - VcGantt* 930
- OnSelectField*
 - VcGantt* 931
- OnShowCurveNameInMenu*
 - VcGantt* 932
- OnShowDate*
 - VcGantt* 933
- OnShowInPlaceEditor*
 - VcGantt* 933
- OnStatusLineText*
 - VcGantt* 935
- OnSupplyTextEntry*
 - VcGantt* 935
- OnSupplyTextEntryAsVariant*
 - VcGantt* 950
- OnTableCaptionLClick*
 - VcGantt* 950
- OnTableCaptionLDbClick*
 - VcGantt* 951
- OnTableCaptionRClick*
 - VcGantt* 951
- OnTableColumnWidth*
 - VcGantt* 952
- OnTableColumnWidthModifyComplete*
 - VcGantt* 953
- OnTableWidth*
 - VcGantt* 953
- OnTableWidthModifyEx*
 - VcGantt* 954
- OnTimeScaleChangeComplete*
 - VcGantt* 955
- OnTimeScaleEndModifyComplete*
 - VcGantt* 955
- OnTimeScaleLClick*
 - VcGantt* 955
- OnTimeScaleLDbClick*
 - VcGantt* 956
- OnTimeScaleRClick*
 - VcGantt* 956
- OnTimeScaleSectionRescaleCompleteEx*
 - VcGantt* 957
- OnTimeScaleSectionRescaleEx*
 - VcGantt* 958
- OnTimeScaleSectionStartModify*
 - VcGantt* 958
- OnTimeScaleStartModifyComplete*
 - VcGantt* 959
- OnToolTipText*
 - VcGantt* 959
- OnToolTipTextAsVariant*
 - VcGantt* 961
- OnViewComponentsSizeModifyComplete*
 - VcGantt* 961

OnWorldViewClosed
VcGantt 962
OnZoomFactorModifyComplete
VcGantt 962
Error
Ereignis von
VcGantt 852
ErrorAsVariant
Ereignis von
VcGantt 853
Esker ActiveX Plug-In 20
Evaluate
Methode von
VcFilter 706
Event
Sicherheitsprüfung 751
EventReturnStatus
Eigenschaft von
VcGantt 750
Event-Sicherheitsprüfung 226
EventsSecurityCheck
Eigenschaft von
VcGantt 751
EventText
Eigenschaft von
VcGantt 751
Export 429
ExportGraphicsToFile
Methode von
VcGantt 815
ExtendedDataTables
Eigenschaft von
VcGantt 751
ExtendedEditingBehavior
Eigenschaft von
VcGantt 752

F

Faltmarkierungen 418
Fehlerbehebung 852
Fehlermeldungen 453
Feldinhalte
ändern 406
Field
siehe auch
VcField 700
FieldByIndex
Methode von
VcDataDefinitionTable 616, 621
FieldByName
Methode von
VcDataDefinitionTable 617, 621
FieldsSeparatedByLines
Eigenschaft von
VcBoxFormat 504
VcTableFormat 1392
FieldText
Eigenschaft von
VcBox 482
FilePath
Eigenschaft von
VcGantt 752
Files
Eigenschaft von
DataObject 461
Fill2Color
Eigenschaft von
VcCurve 577
Fill2Pattern
Eigenschaft von
VcCurve 578
Fill2ReferenceName
Eigenschaft von

- VcCurve 581
- FillColor**
 - Eigenschaft von
 - VcCurve 582
- FillPattern**
 - Eigenschaft von
 - VcCurve 582
- FillReferenceName**
 - Eigenschaft von
 - VcCurve 585
- Filter 101**
 - Anzahl 709
 - bearbeiten 279
 - Knoten selektieren 728
 - markierte Knoten 709
 - Name 703
 - siehe auch
 - VcFilter 701
 - über Index 711
 - Vergleichswert 281
 - verwalten 277
 - verwenden 47
 - Zugriff über Filternamen 712
- FilterByIndex**
 - Methode von
 - VcFilterCollection 711
- FilterByName**
 - Methode von
 - VcFilterCollection 712
- FilterCollection**
 - Eigenschaft von
 - VcGantt 753
 - siehe auch
 - VcFilterCollection 708
- FilterName**
 - Eigenschaft von
 - VcCurve 586
- VcFilterSubCondition 716
 - VcLayer 1074
 - VcLinkAppearance 1169
 - VcTableFormat 1392
- FilterSubCondition**
 - siehe auch
 - VcFilterSubCondition 714
- FirstBox**
 - Methode von
 - VcBoxCollection 500
- FirstCalendar**
 - Methode von
 - VcCalendarCollection 537
- FirstCalendarGrid**
 - Methode von
 - VcCalendarGridCollection 562
- FirstCalendarProfile**
 - Methode von
 - VcCalendarProfileCollection 572
- FirstCurve**
 - Methode von
 - VcCurveCollection 610
- FirstDataRecord**
 - Methode von
 - VcDataRecordCollection 633
- FirstDataTable**
 - Methode von
 - VcDataTableCollection 643
- FirstDataTableField**
 - Methode von
 - VcDataTableFieldCollection 656
- FirstDateLine**
 - Methode von
 - VcDateLineCollection 673
- FirstDateLineGrid**
 - Methode von
 - VcDateLineGridCollection 693

FirstField

Methode von
VcDataDefinitionTable 617, 622

FirstFilter

Methode von
VcFilterCollection 712

FirstFormat

Methode von
VcBoxFormatCollection 511
VcLineFormatCollection 1147
VcTableFormatCollection 1397

FirstGroup

Methode von
VcGroupCollection 975

FirstGroupLevelLayout

Methode von
VcGroupLevelLayoutCollection
1009

FirstHistogram

Methode von
VcHistogramCollection 1035

FirstInterval

Methode von
VcIntervalCollection 1065

FirstLayer

Methode von
VcLayerCollection 1112

FirstLink

Methode von
VcLinkCollection 1186

FirstLinkAppearance

Methode von
VcLinkAppearanceCollection 1182

FirstMap

Methode von
VcMapCollection 1198

FirstMapEntry

Methode von
VcMap 1193

FirstNode

Methode von
VcNodeCollection 1226

FirstNumericScale

Methode von
VcNumericScaleCollection 1258

FirstTable

Methode von
VcTableCollection 1388

FirstTimeScale

Methode von
VcTimeScaleCollection 1426

FirstUpdateBehavior

Methode von
VcUpdateBehaviorCollection 1434

FitChartIntoView

Methode von
VcGantt 818

FitHistogramsIntoView

Methode von
VcGantt 818

FitRangeIntoView

Methode von
VcGantt 819
VcHistogram 1028

FoldingMarksType

Eigenschaft von
VcPrinter 1271

Font

Eigenschaft von
VcDateLine 659
VcNumericScale 1243
VcRibbon 1356
VcTimeScale 1419

FontAntiAliasingEnabled

Eigenschaft von

VcGantt 754

FontBody

Eigenschaft von

VcMapEntry 1205

FontColor

Eigenschaft von

VcDateLine 659

VcNumericScale 1244

VcRibbon 1357

VcTimeScale 1419

FontName

Eigenschaft von

VcMapEntry 1205

Fonts 455

FontSize

Eigenschaft von

VcMapEntry 1206

FormatByIndex

Methode von

VcBoxFormatCollection 511

VcLineFormatCollection 1147

VcTableFormatCollection 1398

FormatByName

Methode von

VcBoxFormatCollection 512

VcLineFormatCollection 1148

VcTableFormatCollection 1398

Formatfeld

Anzahl der Felder 505

FormatCollection-Objekt 1397

FormatField

Eigenschaft von

VcBoxFormat 504

VcLayerFormat 1116

VcLineFormat 1141

VcTableFormat 1393

FormatFieldCount

Eigenschaft von

VcBoxFormat 505

VcLayerFormat 1116

VcLineFormat 1141

VcTableFormat 1393

FormatName

Eigenschaft von

VcBox 482

VcBoxFormatField 515

VcDateLineGrid 678

VcLayerFormatField 1120

VcLineFormatField 1153

VcTableFormatField 1402

Formular

anpassen 29

FreeFloatDataFieldIndex

Eigenschaft von

VcScheduler 1370

FullUsageOfPlanningUnitsEnabled

Eigenschaft von

VcResourceScheduler2 1303

G

Gesamtdiagramm 428, 431

GetActualExtent

Methode von

VcBox 491

GetActualScaleValues

Methode von

VcHistogram 1029

GetActualScaleValuesAsVariant

Methode von

VcHistogram 1029

GetAValueFromARGB

Methode von

VcGantt 819

GetBValueFromARGB

Methode von

VcGantt 820

GetCurrentComponentStart

Methode von

VcGantt 821

GetCurrentViewDates

Methode von

VcGantt 821

GetCurrentViewDatesAsString

Methode von

VcGantt 822

GetCurrentViewDatesAsVariant

Methode von

VcGantt 823

GetCurrentYValues

Methode von

VcHistogram 1029

GetCurrentYValuesAsVariant

Methode von

VcHistogram 1030

GetData

Methode von

DataObject 462

GetDate

Methode von

VcGantt 823

GetDateAsString

Methode von

VcGantt 823

GetEndOfPreviousWorktime

Methode von

VcCalendar 529

GetFirstOverload

Methode von

VcCurve 598

GetFirstOverloadAsVariant

Methode von

VcCurve 599

GetFirstOverloadEx

Methode von

VcCurve 599

GetFormat

Methode von

DataObject 464

GetGValueFromARGB

Methode von

VcGantt 824

GetLinkByID

Methode von

VcGantt 824

GetLinkByIDs

Methode von

VcGantt 825

GetMapEntry

Methode von

VcMap 1194

GetNewUniqueID

Methode von

VcDataRecordCollection 633

GetNextIntervalBorder

Methode von

VcCalendar 530

GetNextOverload

Methode von

VcCurve 600

GetNextOverloadAsVariant

Methode von

VcCurve 601

GetNextOverloadEx

Methode von

VcCurve 601

GetNodeByID

Methode von

- VcGantt 825
- GetPositionInView**
 - Methode von
 - VcNode 1220
- GetPositionInViewAsVariant**
 - Methode von
 - VcNode 1220
- GetPreviousIntervalBorder**
 - Methode von
 - VcCalendar 530
- GetRValueFromARGB**
 - Methode von
 - VcGantt 826
- GetStartOfInterval**
 - Methode von
 - VcCalendar 531
- GetStartOfNextWorktime**
 - Methode von
 - VcCalendar 531
- GetTopLeftPixel**
 - Methode von
 - VcBox 492
- GetValues**
 - Methode von
 - VcCurve 602
- GetValuesAsVariant**
 - Methode von
 - VcCurve 603
- GetValuesEx**
 - Methode von
 - VcCurve 603
- GetViewComponentSize**
 - Methode von
 - VcGantt 827
- GetViewComponentSizeAsVariant**
 - Methode von
 - VcGantt 828
- GetXOffset**
 - Methode von
 - VcBox 493
- GetYOffsetAsVariant**
 - Methode von
 - VcBox 493
- Gitternetzlinien 436**
- Grafik**
 - exportieren 66
- Grafiken**
 - festlegen 376
- Grafikformat 103**
- Grafisches Grundelement**
 - aktueller Scrollwert 821, 843
- GraphicsFileName**
 - Eigenschaft von
 - VcBorderBox 473
 - VcLayer 1075
 - VcMapEntry 1206
 - VcTableFormatField 1403
- GraphicsFileNameDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1076
 - VcTableFormatField 1404
- GraphicsFileNameMapName**
 - Eigenschaft von
 - VcLayer 1077
 - VcTableFormatField 1404
- GraphicsHeight**
 - Eigenschaft von
 - VcBoxFormatField 515
 - VcTableFormatField 1405
- Group**
 - siehe auch
 - VcGroup 964
- GroupByName**
 - Methode von

- VcGroupCollection* 976
- GroupCollection**
 - Eigenschaft von*
 - VcGantt* 754
 - siehe auch
 - VcGroupCollection* 974
- GroupDataFieldIndex**
 - Eigenschaft von*
 - VcGroupLevelLayout* 985
- GroupingField**
 - Eigenschaft von*
 - VcGantt* 755
- GroupingLevel**
 - Eigenschaft von*
 - VcGroup* 966
- GroupingModificationsAllowed**
 - Eigenschaft von*
 - VcGantt* 755
- GroupingOrderField**
 - Eigenschaft von*
 - VcGantt* 756
- GroupingSortOrder**
 - Eigenschaft von*
 - VcGantt* 756
- GroupInvisible**
 - Eigenschaft von*
 - VcGroup* 967
- GroupLevelLayout**
 - siehe auch
 - VcGroupLevelLayout* 978
- GroupLevelLayoutByIndex**
 - Methode von*
 - VcGroupLevelLayoutCollection* 1009
- GroupLevelLayoutByName**
 - Methode von*
- VcGroupLevelLayoutCollection* 1010
- GroupLevelLayoutCollection**
 - Eigenschaft von*
 - VcGantt* 757
 - siehe auch
 - VcGroupLevelLayoutCollection* 1006
- GroupNodes**
 - Methode von*
 - VcGantt* 828
- GroupOptimizationOnInteractionsEnabled**
 - Eigenschaft von*
 - VcGantt* 757
- GroupsInvisible**
 - Eigenschaft von*
 - VcGroupLevelLayout* 985
- GroupsInvisibleCollapsedMapName**
 - Eigenschaft von*
 - VcGroupLevelLayout* 985
- GroupsInvisibleDataFieldIndex**
 - Eigenschaft von*
 - VcGroupLevelLayout* 985
- Gruppe 1218**
 - aktualisieren 973
 - Anzahl 975
 - bearbeiten 814
 - Datenfeld 965
 - Datensatz 971
 - ID 967
 - kollabiert 965, 981
 - löschen 972
 - Name 968, 987
 - sichtbar 971, 1005
 - zugeordneter Datensatz 972
- Gruppe anzeigen 967**
- Gruppen**

- automatisch expandieren 981, 1014
- automatisch kollabieren 980, 1014
- automatisch wiederherstellen 990, 991, 1018, 1019
- Daten bearbeiten 403
- markieren 884
- Optimierung bei Interaktionen 228
- Seitenumbruch 990
- verändern 883
- vertikal verschieben 980
- vertikal verschieben im Diagramm 980
- Gruppenknoten**
 - anzeigen 298
- Gruppentitelzeile**
 - Füllmuster 992, 1231
 - Hintergrundfarbe 991, 1230
- Gruppierenebenenlayout**
 - Gruppierenebene 986
 - Name des Kalendergitters 982, 983, 984, 985
- Gruppierung 108, 294, 298, 299**
 - alle Vorgänge aller Gruppen in einer Zeile/in getrennten Zeilen/kollabieren/expandieren 111
 - an- bzw. abschalten 828
 - Datenfeld für Gruppierkriterium 755
 - Datenfeld zur Sortierung der Gruppen 756
 - Gruppe löschen 880
 - Gruppe verändern 881
 - Gruppe verändern abgeschlossen 882
 - Gruppensortierung 294
 - Gruppierenebene 966
 - interaktives Umgruppieren von Knoten 110, 111
 - Kalender 295
 - Kalendergitter 1001, 1027, 1230
 - Knoten in Gruppen separieren 736
 - Kollabieren/Expandieren erlaubt 297, 755, 986
 - mehrstufig 969
 - Obergruppe 970
 - Sortierung optimierter Knoten 295
 - Sortierung sich überlappender Knoten 294
 - Sortierungsrichtung 294, 756, 1003, 1241
 - Sortierungsrichtung optimierter Knoten 295
 - Sortierungsrichtung überlappender Knoten 295
 - Stichtaglinie 984
 - Stichtaglinien anzeigen 1002
 - Stichtaglinien für Knoten 1240
 - Stichtaglinien für Untergruppen 984
 - Trennlinien 297
 - Trennlinienfarbe 998
 - Überschneidungen von Vorgängen in einer Gruppe sichtbar machen 446
 - Untergruppen 970
 - vertikale Rasterlinien 1001
 - zu Anfang kollabiert 297
- Gruppierungsebene**
 - Liniendicke 999, 1238
 - Trennlinienfarbe 998, 1237
- Gruppierungsebenen**
 - Gruppen anzeigen 985
 - Gruppenknoten anzeigen 1002
 - Kalendergitter für Vorgänge anzeigen 1239
 - Name des Kalendergitters von Vorgängen 1230
 - Trennlinien anzeigen 1002, 1003, 1240
- Gruppierungsebenenlayout**

Name der Zuordnungstabelle für kollabierte Gruppenkörper 982

Name der Zuordnungstabelle für nicht sichtbare Gruppen 985

H

Height

Eigenschaft von

VcLayer 1078

VcLegendView 1132

VcRect 1286

VcWorldView 1442

HeightActualValue

Eigenschaft von

VcLegendView 1133

VcWorldView 1443

HeightDataFieldIndex

Eigenschaft von

VcLayer 1078

HeightMapName

Eigenschaft von

VcLayer 1079

Hidden

Eigenschaft von

VcDataTableField 647

VcDefinitionField 698

Hierarchie 114, 291, 292, 293

alle Knoten einer Gruppe auf dieser Ebene in einer Zeile 1016

ändern 909

Code-neuberechnung 841

Hierarchieeigenschaften ausfragen 759

Seitenumbruch nach Gruppen 1018

Summenbalken interaktiv verschieben 116

Vorgänge interaktiv verschieben 115

Hierarchieebene

Zuordnungstabelle 1015

Hierarchieebenen

Knoten-Layout optimieren 1018

Knotentrennlinien anzeigen 1016

Liniendicke 1020

Linientyp 1021

Seitenumbruch auf bestimmte Ebenen festlegen 1016

Seitenumbruch nach jeder Gruppe 1014

Sortierindex 1015

Summenbalken anzeigen 1022

Trennlinien anzeigen 1022

Trennlinienfarbe 1019

HierarchyDataFieldIndex

Eigenschaft von

VcGantt 758

VcHierarchyLevelLayout 1015

HierarchyLevelLayout

Eigenschaft von

VcGantt 759

siehe auch

VcHierarchyLevelLayout 1013

Hilfe-Ereignis 885

Hintergrundfarbe

markierte row 248

Histogram

Datenquelle der Kurve einstellen 368

Eigenschaft von

VcCurve 587

VcNumericScale 1244

siehe auch

VcHistogram 1023

HistogramByIndex

Methode von

VcHistogramCollection 1035

HistogramByName

Methode von
 VcHistogramCollection 1035

HistogramCollection
 Eigenschaft von
 VcGantt 759
 siehe auch
 VcHistogramCollection 1032

Histogramm 117, 246
 alle Histogramme in ein Fenster 818
 bearbeiten 363
 CurveCollection 607
 Extremwerte der Kurven 1029, 1030
 Füllmuster 1026
 Hintergrundfarbe 1026
 Höhenverhältnis von Diagramm und
 Histogramm 746
 Höhenverhältnis zwischen Diagramm
 und Histogramm ändern 888, 889
 Kalender zuweisen 1024
 Kurven 365
 Kurven in Kontextmenü anzeigen
 932
 Kurven markieren 590
 Kurven-Collection 1024
 Kurvenpunkt einfügen 890, 891
 Kurvenpunkt löschen 876, 877
 Maximalwert der numerischen Skala
 1025
 Maximalwert der numerischen Skala
 des (ersten) Histogramms 829
 Minimalwert der numerischen Skala
 1025
 Name 1024
 numerische Skala interaktiv skalieren
 916
 Reihenfolge 1030
 sichtbar 1027
 Skala in das Diagramm einpassen
 1028

Skalen-Collection 1026
Tatsächliche Extremwerte der
numerischen Skala 1029
Trennlinienfarbe 759
Typ des Skalenstreifens einstellen
369
Verhältnis der Höhe des Histogramms
zur Gesamthöhe des Diagramms
746

verwalten 361

Histogrammauflistung

aktives Histogramm 1033
Anzahl Histogramme 1033
Enumerator 1032
Erstes Histogramm 1035
Histogramm anlegen 1034
Histogramm löschen 1034
Histogramm über Index 1035
Histogramm über Name 1035
nächstes Histogramm 1036

Histogramme

erstellen 51

Histogrammkurve

Anzahl der Einheiten pro Schritt 595
Filter 586
Füllfarbe 582
Füllfarbe der 2. Referenzkurve 577
Füllmuster 582
Füllmuster der 2. Referenzkurve 578
interaktiv verändern 867, 868, 869
Kurvenpunkt löschen 597, 598
Kurvenpunkte äquidistant 593
Layer 587
Liniendicke 588
Linienfarbe 587
Linientyp 589
Mustervordergrundfarbe 592

Mustervordergrundfarbe der 2.
 Referenzkurve 592
 Name 591
 Overload 598, 599, 600, 601, 602
 Quelle 576
 sichtbar 596
 Stapelbezug 586, 594
 Typ 576
 Werte setzen 605
 Wertigkeitsdatenfeld 596
 y-Wert zu einem Datum 602, 603,
 604
 y-Werte aller Punkte auf Null setzen
 597
 y-Werte erhöhen 575
 Zeiteinheit 594
 zweite Referenzkurve 581
HistogramSeparationLineColor
Eigenschaft von
 VcGantt 759
HistogramSetMaxYValue
Methode von
 VcGantt 829
Höhenverhältnis
 Diagrammbereich/Histogramm 247
HorAlignment
Eigenschaft von
 VcDateLineGrid 679
HorizontalOffset
Eigenschaft von
 VcLayer 1079
HTML 12
HTML-Seite 20
hWnd 760
Eigenschaft von
 VcGantt 760

ID

Eigenschaft von
 VcDataRecord 626
 VcDefinitionField 698
 VcGroup 967
 VcLink 1165
 VcNode 1215

Identifiable

Eigenschaft von
 VcCalendarGrid 543
 VcDateLine 660

IdentifyField

Methode von
 VcGantt 829

IdentifyFormatField

Methode von
 VcBox 493
 VcTable 1385

IdentifyInterval

Methode von
 VcCalendarGrid 556

IdentifyIntervalAsVariant

Methode von
 VcCalendarGrid 557

IdentifyLayerAt

Methode von
 VcGantt 830

IdentifyLayerAtAsVariant

Methode von
 VcGantt 831

IdentifyObject

Methode von
 VcDataRecord 627
 VcGantt 831

IdentifyObjectAt

- Methode von
 - VcGantt 833
- IdentifyObjectAtAsVariant**
 - Methode von
 - VcGantt 834
- IncomingLinks**
 - Eigenschaft von
 - VcNode 1215
- IndentColumn**
 - Eigenschaft von
 - VcTableFormat 1394
- IndentWidth**
 - Eigenschaft von
 - VcTableFormat 1394
- Index**
 - Eigenschaft von
 - VcBoxFormatField 516
 - VcDataTableField 648
 - VcFilterSubCondition 717
 - VcLayerFormatField 1121
 - VcLineFormatField 1153
 - VcTableFormatField 1405
- Individual behavior for each node layout 187**
- Informationsfenster 760**
- InfoWindow 760**
 - Eigenschaft von
 - VcGantt 760
 - siehe auch
 - VcInfoWindow 1037
- ini-Datei 440**
- InInteractionEventsEnabled**
 - Eigenschaft von
 - VcGantt 760
- In-Place-Editieren**
 - Gruppen im Diagramm 222
 - Gruppen in Tabelle 221
 - Knoten im Diagramm 221
 - Knoten in Tabelle 221
- InPlaceEditingOnGroupsInDiagramEnabled**
 - Eigenschaft von
 - VcGantt 761
- InPlaceEditingOnGroupsInTableEnabled**
 - Eigenschaft von
 - VcGantt 761
- InPlaceEditingOnNodesInDiagramEnabled**
 - Eigenschaft von
 - VcGantt 762
- InPlaceEditingOnNodesInTableEnabled**
 - Eigenschaft von
 - VcGantt 763
- InsertLinkRecord**
 - Methode von
 - VcGantt 834
- InsertNodeRecord**
 - Methode von
 - VcGantt 835
- Installation 14**
- InteractionMode**
 - Eigenschaft von
 - VcGantt 764
- Interaktion**
 - Markierung mehrerer Boxen 728
- Interaktionen**
 - Einrastlinien anzeigen 239
 - Einrastmarkierungen anzeigen 239
 - Einrastziele definieren 239
 - Tabellen- und Diagrammbereich 40
 - Vorgänge 42
- Interaktionsmodi 764**
- Interaktives Verschieben**

früheste Startzeit 1087
 späteste Endzeit 1086
Internet 12, 66, 383, 429
Interval
 siehe auch
 VcInterval 1045
IntervalByIndex
 Methode von
 VcIntervalCollection 1065
IntervalByName
 Methode von
 VcIntervalCollection 1065
IntervalCollection
 Eigenschaft von
 VcCalendar 526
 VcCalendarProfile 565
 siehe auch
 VcIntervalCollection 1062
Intervall
 Anzahl 1063
 Dauer 1048
 Enddatum und -zeit 1049
 Endmonat 1049
 Endzeit 1049
 erster Wochentag 1058
 erstes Intervall 1065
 Hintergrundfarbe 1047
 hinzufügen 1063
 identifizieren 556
 Kalenderprofil 1047
 kopieren 1064
 letzter Wochentag 1050
 Linientyp 1051
 löschen 1066
 Muster 1053
 Musterfarbe 1056
 nächstes Intervall 1066

Name 1052
 Reihenfolge 1060
 Startdatum und -zeit 1057
 Startmonat 1057
 Startzeit 1058
 Tag des ersten Monats 1048
 Tag des letzten Monats 1048
 Typ 1059
 über Index 1065
 Verwendung der grafischen Attribute
 554, 1059
 Zeistreifenbeschriftung 1058
 Zeiteinheit 1059
 Zugriff über Intervallnamen 1066
Intervall-Collection
 aktualisieren 1067
Intervalle
 bearbeiten 341, 345, 347, 348, 350
 Liniendicke 1051
 Linienfarbe 1050
IsEditable
 Eigenschaft von
 VcUpdateBehavior 1428
 VcUpdateBehaviorContext 1439
IsValid
 Methode von
 VcFilter 707
 VcFilterSubCondition 718
IsWorktime
 Methode von
 VcCalendar 531
Item
 Eigenschaft von
 DataObjectFiles 468

K

Kalender 85

aktiver 534
aktualisieren 532
Anzahl 535
Anzahl der Sekunden eines
Arbeitsstages 527
Dauer 528
festlegen 339
Name 526, 566
Typ 527
über Index 536
zu Knoten zuweisen 736

Kalendergitter 359

Datenfeldindex in Zuordnungstabelle
für Kalender 542
Datenfeldindex in Zuordnungstabelle
für Sichtbarkeit 555, 687
Einrastziel am Datum 553
Hintergrundfarbe 541
Liniendicke 545
Linienfarbe 544
Linienfarbenzuordnungstabelle 544
Linientyp 546
Mustertyp 547
Name 547
Name der Zuordnungstabelle für
Kalender 542
Name der Zuordnungstabelle für
Sichtbarkeit 555, 688
Spezifikation 553
Startdatum als Einrastziel 542, 554
verwalten 301

Kalenderliniengitter 1375

Kalenderprofil

Anzahl 569
Reihenfolge 567
Typ 566
über Index 570
Zugriff über Kalenderprofilnamen 571

Kalenderprofil verwalten

Dialogfeld: 343

KeyDown

Ereignis von
VcGantt 853

KeyPress

Ereignis von
VcGantt 854

KeyUp

Ereignis von
VcGantt 854

Kleinstes Zeitintervall 218

Knoten 136

aktualisieren 1224
Aktualisierung der optimierten
Anordnung 973
alle Daten 1214
alle Knoten einer Gruppe auf dieser
Ebene in einer Zeile 987, 988,
1017
alle Knoten einer Gruppe in einer
Zeile 969
alle Layer verschieben mit
Umschalttaste 771
bearbeiten 398, 814
Daten aktualisieren 850
Daten bearbeiten 398
Datenfeld 1214
Datensatz 1219
durch Filter selektieren 728
erzeugen 136, 427, 428, 905, 906
Gruppe 1218
herausführende Verbindungen 1217
hineinführende Verbindungen 1215
ID 1215
Informationsfenster 760
interaktiv anlegen erlaubt 1370
interaktiv erzeugen 448

- interaktiv umgruppieren 110, 111
 - interaktiv verschieben 1216
 - Kalender zuweisen 237
 - Knotenmarkierung im Diagramm 234
 - Knotenmarkierung in der Tabelle 234
 - laden 835
 - Layout optimieren 969, 988
 - löschen 136, 431, 811, 906, 907, 1219
 - löschen, ausschneiden, kopieren, einfügen 397, 401
 - markieren 389, 909, 913, 1216
 - markierte Knoten gemeinsam verschieben 238, 770
 - markierten Knoten als Ganzes verschieben 238, 772
 - Markierung durch Gummirechteck 730, 731
 - neue anlegen erlaubt 729
 - neue Knoten bearbeiten 237, 749
 - neue Knoten per Doppelklick 237, 772
 - neue zulassen 237
 - optimierte Anordnung 758
 - Tabellenfeld identifizieren 829
 - verändern 910
 - verschieben 391, 393, 732
 - Zeile im sichtbaren Diagrammbereich 1221
 - Zeilennummern aktualisieren 851
 - zugeordneter Datensatz 1222
 - Knoten verschieben**
 - Verbindungstyp berücksichtigen 228
 - Knotenaussehen**
 - Farbzuordnungstabelle 999
 - Knotenreihenfolge**
 - ändern in Diagramm 240
 - ändern in Tabelle 240
 - Knotenzeilen**
 - Anzahl beim Start 245
 - minimale Höhe 245
 - Komplettansicht 139, 429, 807**
 - schließen-Ereignis 896, 962
 - Konfiguration 67, 220, 440, 739**
 - speichern 813
 - Kontext des Aktualisierungsverhaltens**
 - Aktualisierungsmodus 1440
 - editierbar 1439
 - Typ 1439
 - Verzögerungszeit 1438
 - Kontextmenü**
 - abschalten 449
 - für Boxen 438
 - für das Diagramm 426
 - für das Histogramm 424
 - für die Zeitskala 436
 - für Gruppen 434
 - für Knoten 431
 - für Verbindungen 433
 - Kontextmenü für Boxen 224**
 - Kundendienst 26**
 - Kurve**
 - Ereignis Modifizierung 866
 - löschen 611
 - über Index 609
 - Kurven**
 - markieren 590
 - Kurvendatenquelle 368**
- L
- LabelPosition**
 - Eigenschaft von VcDateLine 660
 - LabelSizeDependence**
 - Eigenschaft von

VcLayer 1080

Ladevorgang

Ende 815

LateEndDateDataFieldIndex

Eigenschaft von

VcScheduler 1370

LateStartDateDataFieldIndex

Eigenschaft von

VcScheduler 1371

Layer 141

3D-Effekt 267

Anzahl 1110

Collection aktualisieren 1114

Dauer 271

End-Termin-Feld 270

Höhe 267

identifizieren 557, 830, 831

in Legende 265

kopieren 1111

Layer bearbeiten 267

Layerform 268

Layerformat bearbeiten 273

Linienfarbenzuordnungstabelle 1084,
1092

löschen 1114

markieren 389

Phantomhöhe 783

Reihenfolge 1107

siehe auch

VcLayer 1068

Startdatum als Einrastziel 1074, 1103

Start-Termin-Feld 270

über Index 1112

verschieben 395

verwenden 44

Layer verschieben

Mit Umschalttaste 239

LayerByIndex

Methode von

VcLayerCollection 1112

LayerByName

Methode von

VcLayerCollection 1113

LayerCollection

Eigenschaft von

VcGantt 764

siehe auch

VcLayerCollection 1109

Layerformat

Anzahl der Felder 1116

LayerFormat

Eigenschaft von

VcLayer 1080

siehe auch

VcLayerFormat 1115

LayerFormatField

siehe auch

VcLayerFormatField 1118

LayerName

Eigenschaft von

VcCurve 587

LayerShape

Eigenschaft von

VcLayer 1080

Leerseiten unterdrücken 418

Left

Eigenschaft von

VcLegendView 1133

VcRect 1287

VcWorldView 1443

LeftActualValue

Eigenschaft von

VcLegendView 1134

VcWorldView 1444

LeftMargin*Eigenschaft von**VcLayerFormatField* 1121, 1122*VcTableFormatField* 1405**Legend***Eigenschaft von**VcMapEntry* 1207**Legende**

Anordnung 380

Anordnung 381

Attribute 380

festlegen 376

Titel 380

LegendElementsArrangement*Eigenschaft von**VcBoundingBox* 474**LegendElementsBottomMargin***Eigenschaft von**VcBoundingBox* 474**LegendElementsMaximumColumnCount***Eigenschaft von**VcBoundingBox* 474**LegendElementsMaximumRowCount***Eigenschaft von**VcBoundingBox* 475**LegendElementsTopMargin***Eigenschaft von**VcBoundingBox* 475**Legendenansicht** 145, 429, 764**LegendFont***Eigenschaft von**VcBoundingBox* 475**LegendText***Eigenschaft von**VcLayer* 1083**LegendTitle***Eigenschaft von**VcBoundingBox* 476**LegendTitleFont***Eigenschaft von**VcBoundingBox* 476**LegendTitleVisible***Eigenschaft von**VcBoundingBox* 477**Legendview** 145, 764**LegendView***Eigenschaft von**VcGantt* 764

siehe auch

VcLegendView 1131**Level***Eigenschaft von**VcGroupLevelLayout* 986**LevelMaximumForPagebreaks***Eigenschaft von**VcHierarchyLevelLayout* 1016**LineColor***Eigenschaft von**VcBox* 483*VcCalendarGrid* 544*VcCurve* 587*VcDateLine* 660*VcDateLineGrid* 679*VcInterval* 1050*VcLayer* 1083*VcLinkAppearance* 1170*VcNumericScale* 1245*VcSection* 1377, 1378**LineColorDataFieldIndex***Eigenschaft von**VcCalendarGrid* 544*VcDateLineGrid* 680*VcLayer* 1084

LineColorMapName

Eigenschaft von

VcCalendarGrid 544

VcDateLineGrid 680

VcLayer 1084

LineFormat

siehe auch

VcLineFormat 1140

LineFormatCollection

Eigenschaft von

VcGantt 765

siehe auch

VcLineFormatCollection 1144

LineFormatField

siehe auch

VcLineFormatField 1150

LineThickness

Eigenschaft von

VcBox 483

VcCalendarGrid 545

VcCurve 588

VcDateLine 661

VcDateLineGrid 680

VcInterval 1050

VcLayer 1084

VcLinkAppearance 1170

LineType

Eigenschaft von

VcBox 484

VcCalendarGrid 546

VcCurve 589

VcDateLine 662

VcDateLineGrid 681

VcInterval 1051

VcLayer 1085

VcLinkAppearance 1171

Linie bearbeiten 337

Linienformate

verwalten 283, 285

Linienformatfeld

Hintergrundfarbe des Musters 1154

Liniengitter 358

Einrastziel am Datum 685

verwalten 304

Linineformatfeld

Datumsausgabeformat 1153

Link

siehe auch

VcLink 1164

LinkAppearance

siehe auch

VcLinkAppearance 1169

LinkAppearanceByIndex

Methode von

VcLinkAppearanceCollection 1182

LinkAppearanceByName

Methode von

VcLinkAppearanceCollection 1183

LinkAppearanceCollection

Eigenschaft von

VcGantt 765

siehe auch

VcLinkAppearanceCollection 1179

LinkAppearance-Objekt

Anzahl in Collection 1180

Enumerator-Objekt 1180

Iteration, Erstwert 1182

Iteration, Folgewert 1183

über Index 1182

über Name 1183

LinkCollection

Eigenschaft von

VcGantt 765

siehe auch

VcLinkCollection 1185

LinkDataTableName

Eigenschaft von

VcResourceScheduler2 1304

LinkDurationDataFieldIndex

Eigenschaft von

VcScheduler 1371

LinkDurationFieldIndex

Eigenschaft von

VcResourceScheduler2 1306

LinkPredecessorDataFieldIndex

Eigenschaft von

VcGantt 766

LinkPredecessorOperationIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1306

LinkPredecessorTaskIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1307

LinksDataTableName

Eigenschaft von

VcGantt 767

LinkSuccessorDataFieldIndex

Eigenschaft von

VcGantt 767

LinkSuccessorOperationIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1307

LinkSuccessorTaskIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1308

LinkTypeDataFieldIndex

Eigenschaft von

VcGantt 768

Live Update 147

Lizenzierung 15, 229, 382

Probleme 439

Lizenzinformationen

anfordern 384

M

MajorTicks

Eigenschaft von

VcNumericScale 1245

VcRibbon 1357

MajorTicksEx

Eigenschaft von

VcNumericScale 1245

MakeARGB

Methode von

VcGantt 836

Map 769

Anzahl der Einträge 1190

Anzahl der Maps 1196

durch Zuordnungstabelle bestimmte
Vorgänge aktualisieren 1201

Eintrag erzeugen 1192

Eintrag löschen 1193

Name 1191

siehe auch

VcMap 1189

MapByIndex

Methode von

VcMapCollection 1199

MapByName

Methode von

VcMapCollection 1199

MapCollection

Eigenschaft von

VcGantt 769

siehe auch

VcMapCollection 1195

Map-Eintrag

Datenfeld-Inhalt 1204

Farbwert 1204

Grafikdatei 1207

Legendentext 1207

Millimeter 1208

Muster 1209

Schriftart 1205

Schriftgrad 1205

Schriftgröße 1206

MapEntry

siehe auch

VcMapEntry 1203

MarginsShownInInches

Eigenschaft von

VcPrinter 1274

MarkBox

Eigenschaft von

VcBox 485

MarkCurve

Eigenschaft von

VcCurve 590

MarkedNodesFilter

Eigenschaft von

VcFilterCollection 709

MarkGroup

Eigenschaft von

VcGroup 967

Markieren

Gummirechteck 229

Knoten bzw. Layer 389

Knoten im Diagramm 234

Knoten in der Tabelle 234

Markieren/Demarkieren

Ende der Operation 885, 913

MarkingColor

Eigenschaft von

VcWorldView 1444

MarkNode

Eigenschaft von

VcNode 1216

MaxHorizontalPagesCount

Eigenschaft von

VcPrinter 1274

MaximumEndDataFieldIndex

Eigenschaft von

VcLayer 1086

MaximumTextLineCount

Eigenschaft von

VcBoxFormatField 516

VcTableFormatField 1406

MaxVerticalPagesCount

Eigenschaft von

VcPrinter 1275

Methoden

AboutBox

VcGantt 809

Add

DataObjectFiles 469

VcBoxCollection 497

VcBoxFormatCollection 509

VcCalendarCollection 535

VcCalendarGridCollection 559

VcCalendarProfileCollection 569

VcCurveCollection 608

VcDataRecordCollection 631

VcDataTableCollection 640

VcDataTableFieldCollection 653

VcDateLineCollection 670

VcDateLineGridCollection 690

VcFilterCollection 710

VcGroupLevelLayoutCollection
1007

VcIntervalCollection 1063

VcLayerCollection 1110

- VcLineFormatCollection* 1145
- VcLinkAppearanceCollection* 1180
- VcMapCollection* 1197
- VcUpdateBehaviorCollection* 1433
- AddBySpecification*
 - VcBoxCollection* 498
 - VcBoxFormatCollection* 510
 - VcCalendarCollection* 535
 - VcCalendarGridCollection* 560
 - VcCalendarProfileCollection* 570
 - VcCurveCollection* 608
 - VcDateLineCollection* 671
 - VcDateLineGridCollection* 691
 - VcFilterCollection* 710
 - VcGroupLevelLayoutCollection* 1008
 - VcIntervalCollection* 1064
 - VcLayerCollection* 1111
 - VcLineFormatCollection* 1146
 - VcLinkAppearanceCollection* 1181
 - VcMapCollection* 1197
 - VcUpdateBehaviorCollection* 1433
- AddDuration*
 - VcCalendar* 528
- AddSubCondition*
 - VcFilter* 705
- AnchorToNode*
 - VcBox* 491
- BorderBox*
 - VcBorderArea* 471
- BoxByIndex*
 - VcBoxCollection* 498
- BoxByName*
 - VcBoxCollection* 499
- CalcDuration*
 - VcCalendar* 528
- CalculateCurrentWidth*
 - VcLayer* 1107
- CalculateLineCount*
 - VcLayerFormatField* 1129
- CalendarByIndex*
 - VcCalendarCollection* 536
- CalendarByName*
 - VcCalendarCollection* 536
- CalendarGridByIndex*
 - VcCalendarGridCollection* 560
- CalendarGridByName*
 - VcCalendarGridCollection* 561
- CalendarProfileByIndex*
 - VcCalendarProfileCollection* 570
- CalendarProfileByName*
 - VcCalendarProfileCollection* 571
- Clear*
 - DataObject* 462
 - DataObjectFiles* 469
 - VcCalendar* 529
 - VcCurve* 597
 - VcGantt* 809
- ClearAll*
 - VcGantt* 810
- Context*
 - VcUpdateBehavior* 1430
- ConvertDistance*
 - VcGantt* 810
- Copy*
 - VcBoxCollection* 499
 - VcBoxFormatCollection* 510
 - VcCalendarCollection* 537
 - VcCalendarGridCollection* 561
 - VcCalendarProfileCollection* 571
 - VcCurveCollection* 609
 - VcDataTableCollection* 641
 - VcDataTableFieldCollection* 654
 - VcDateLineCollection* 671

- VcDateLineGridCollection* 691
- VcFilterCollection* 711
- VcGroupLevelLayoutCollection* 1008
- VcIntervalCollection* 1064
- VcLayerCollection* 1111
- VcLineFormatCollection* 1146
- VcLinkAppearanceCollection* 1181
- VcMapCollection* 1198
- VcUpdateBehaviorCollection* 1434
- CopyFormatField*
 - VcBoxFormat* 506
 - VcLayerFormat* 1117
 - VcLineFormat* 1143
- CopySubCondition*
 - VcFilter* 706
- CreateDataField*
 - VcDataDefinitionTable* 615, 620
- CreateEntry*
 - VcMap* 1192
- CreateHistogram*
 - VcHistogramCollection* 1034
- CurveByIndex*
 - VcCurveCollection* 609
- CurveByName*
 - VcCurveCollection* 610
- DataRecord*
 - VcGroup* 971
 - VcLink* 1167
 - VcNode* 1219
- DataRecordByID*
 - VcDataRecordCollection* 632
- DataTableByIndex*
 - VcDataTableCollection* 642
- DataTableByName*
 - VcDataTableCollection* 642
- DataTableFieldByIndex*
 - VcDataTableFieldCollection* 655
- DataTableFieldByName*
 - VcDataTableFieldCollection* 655
- DateLineByIndex*
 - VcDateLineCollection* 672
- DateLineByName*
 - VcDateLineCollection* 673
- DateLineGridByIndex*
 - VcDateLineGridCollection* 692
- DateLineGridByName*
 - VcDateLineGridCollection* 692
- DeleteDataRecord*
 - VcDataRecord* 627
- DeleteEntry*
 - VcMap* 1193
- DeleteGroup*
 - VcGroup* 972
- DeleteHistogram*
 - VcHistogramCollection* 1034
- DeleteLink*
 - VcLink* 1167
- DeleteLinkRecord*
 - VcGantt* 811
- DeleteNode*
 - VcNode* 1219
- DeleteNodeRecord*
 - VcGantt* 811
- DeletePoint*
 - VcCurve* 597
- DeletePointAsVariant*
 - VcCurve* 598
- DetectDataTableFieldName*
 - VcGantt* 811
- DetectDataTableName*
 - VcGantt* 812
- DetectFieldIndex*
 - VcGantt* 812

- DetermineIDOfFirstOperationByTaskID*
 - VcResourceScheduler2* 1351
- DetermineIDOfLastOperationByTaskID*
 - VcResourceScheduler2* 1351
- DumpConfiguration*
 - VcGantt* 813
- EditGroup*
 - VcGantt* 814
- EditLink*
 - VcGantt* 814
- EditNode*
 - VcGantt* 814
- EndLoading*
 - VcGantt* 815
- Evaluate*
 - VcFilter* 706
- ExportGraphicsToFile*
 - VcGantt* 815
- FieldByIndex*
 - VcDataDefinitionTable* 616, 621
- FieldByName*
 - VcDataDefinitionTable* 617, 621
- FilterByIndex*
 - VcFilterCollection* 711
- FilterByName*
 - VcFilterCollection* 712
- FirstBox*
 - VcBoxCollection* 500
- FirstCalendar*
 - VcCalendarCollection* 537
- FirstCalendarGrid*
 - VcCalendarGridCollection* 562
- FirstCalendarProfile*
 - VcCalendarProfileCollection* 572
- FirstCurve*
 - VcCurveCollection* 610
- FirstDataRecord*
 - VcDataRecordCollection* 633
- FirstDataTable*
 - VcDataTableCollection* 643
- FirstDataTableField*
 - VcDataTableFieldCollection* 656
- FirstDateLine*
 - VcDateLineCollection* 673
- FirstDateLineGrid*
 - VcDateLineGridCollection* 693
- FirstField*
 - VcDataDefinitionTable* 617, 622
- FirstFilter*
 - VcFilterCollection* 712
- FirstFormat*
 - VcBoxFormatCollection* 511
 - VcLineFormatCollection* 1147
 - VcTableFormatCollection* 1397
- FirstGroup*
 - VcGroupCollection* 975
- FirstGroupLevelLayout*
 - VcGroupLevelLayoutCollection* 1009
- FirstHistogram*
 - VcHistogramCollection* 1035
- FirstInterval*
 - VcIntervalCollection* 1065
- FirstLayer*
 - VcLayerCollection* 1112
- FirstLink*
 - VcLinkCollection* 1186
- FirstLinkAppearance*
 - VcLinkAppearanceCollection* 1182
- FirstMap*
 - VcMapCollection* 1198
- FirstMapEntry*

- VcMap 1193
- FirstNode
 - VcNodeCollection 1226
- FirstNumericScale
 - VcNumericScaleCollection 1258
- FirstTable
 - VcTableCollection 1388
- FirstTimeScale
 - VcTimeScaleCollection 1426
- FirstUpdateBehavior
 - VcUpdateBehaviorCollection 1434
- FitChartIntoView
 - VcGantt 818
- FitHistogramsIntoView
 - VcGantt 818
- FitRangeIntoView
 - VcGantt 819
 - VcHistogram 1028
- FormatByIndex
 - VcBoxFormatCollection 511
 - VcLineFormatCollection 1147
 - VcTableFormatCollection 1398
- FormatByName
 - VcBoxFormatCollection 512
 - VcLineFormatCollection 1148
 - VcTableFormatCollection 1398
- GetActualExtent
 - VcBox 491
- GetActualScaleValues
 - VcHistogram 1029
- GetActualScaleValuesAsVariant
 - VcHistogram 1029
- GetAValueFromARGB
 - VcGantt 819
- GetBValueFromARGB
 - VcGantt 820
- GetCurrentComponentStart
 - VcGantt 821
- GetCurrentViewDates
 - VcGantt 821
- GetCurrentViewDatesAsString
 - VcGantt 822
- GetCurrentViewDatesAsVariant
 - VcGantt 823
- GetCurrentYValues
 - VcHistogram 1029
- GetCurrentYValuesAsVariant
 - VcHistogram 1030
- GetData
 - DataObject 462
- GetDate
 - VcGantt 823
- GetDateAsString
 - VcGantt 823
- GetEndOfPreviousWorktime
 - VcCalendar 529
- GetFirstOverload
 - VcCurve 598
- GetFirstOverloadAsVariant
 - VcCurve 599
- GetFirstOverloadEx
 - VcCurve 599
- GetFormat
 - DataObject 464
- GetGValueFromARGB
 - VcGantt 824
- GetLinkByID
 - VcGantt 824
- GetLinkByIDs
 - VcGantt 825
- GetMapEntry
 - VcMap 1194
- GetNewUniqueID
 - VcDataRecordCollection 633

- GetNextIntervalBorder*
 - VcCalendar* 530
- GetNextOverload*
 - VcCurve* 600
- GetNextOverloadAsVariant*
 - VcCurve* 601
- GetNextOverloadEx*
 - VcCurve* 601
- GetNodeByID*
 - VcGantt* 825
- GetPositionInView*
 - VcNode* 1220
- GetPositionInViewAsVariant*
 - VcNode* 1220
- GetPreviousIntervalBorder*
 - VcCalendar* 530
- GetRValueFromARGB*
 - VcGantt* 826
- GetStartOfInterval*
 - VcCalendar* 531
- GetStartOfNextWorktime*
 - VcCalendar* 531
- GetTopLeftPixel*
 - VcBox* 492
- GetValues*
 - VcCurve* 602
- GetValuesAsVariant*
 - VcCurve* 603
- GetValuesEx*
 - VcCurve* 603
- GetViewComponentSize*
 - VcGantt* 827
- GetViewComponentSizeAsVariant*
 - VcGantt* 828
- GetXYOffset*
 - VcBox* 493
- GetXYOffsetAsVariant*
 - VcBox* 493
- GroupByName*
 - VcGroupCollection* 976
- GroupLevelLayoutByIndex*
 - VcGroupLevelLayoutCollection* 1009
- GroupLevelLayoutByName*
 - VcGroupLevelLayoutCollection* 1010
- GroupNodes*
 - VcGantt* 828
- HistogramByIndex*
 - VcHistogramCollection* 1035
- HistogramByName*
 - VcHistogramCollection* 1035
- HistogramSetMaxYValue*
 - VcGantt* 829
- IdentifyField*
 - VcGantt* 829
- IdentifyFormatField*
 - VcBox* 493
 - VcTable* 1385
- IdentifyInterval*
 - VcCalendarGrid* 556
- IdentifyIntervalAsVariant*
 - VcCalendarGrid* 557
- IdentifyLayerAt*
 - VcGantt* 830
- IdentifyLayerAtAsVariant*
 - VcGantt* 831
- IdentifyObject*
 - VcDataRecord* 627
 - VcGantt* 831
- IdentifyObjectAt*
 - VcGantt* 833
- IdentifyObjectAtAsVariant*
 - VcGantt* 834

- InsertLinkRecord*
 - VcGantt* 834
- InsertNodeRecord*
 - VcGantt* 835
- IntervalByIndex*
 - VcIntervalCollection* 1065
- IntervalByName*
 - VcIntervalCollection* 1065
- IsValid*
 - VcFilter* 707
 - VcFilterSubCondition* 718
- IsWorktime*
 - VcCalendar* 531
- LayerByIndex*
 - VcLayerCollection* 1112
- LayerByName*
 - VcLayerCollection* 1113
- LinkAppearanceByIndex*
 - VcLinkAppearanceCollection* 1182
- LinkAppearanceByName*
 - VcLinkAppearanceCollection* 1183
- MakeARGB*
 - VcGantt* 836
- MapByIndex*
 - VcMapCollection* 1199
- MapByName*
 - VcMapCollection* 1199
- NextBox*
 - VcBoxCollection* 500
- NextCalendar*
 - VcCalendarCollection* 538
- NextCalendarGrid*
 - VcCalendarGridCollection* 562
- NextCalendarProfile*
 - VcCalendarProfileCollection* 572
- NextCurve*
 - VcCurveCollection* 611
- NextDataRecord*
 - VcDataRecordCollection* 634
- NextDataTable*
 - VcDataTableCollection* 643
- NextDataTableField*
 - VcDataTableFieldCollection* 656
- NextDateLine*
 - VcDateLineCollection* 674
- NextDateLineGrid*
 - VcDateLineGridCollection* 693
- NextField*
 - VcDataDefinitionTable* 618, 622
- NextFilter*
 - VcFilterCollection* 713
- NextFormat*
 - VcBoxFormatCollection* 512
 - VcLineFormatCollection* 1148
 - VcTableFormatCollection* 1399
- NextGroup*
 - VcGroupCollection* 976
- NextGroupLevelLayout*
 - VcGroupLevelLayoutCollection* 1010
- NextHistogram*
 - VcHistogramCollection* 1036
- NextInterval*
 - VcIntervalCollection* 1066
- NextLayer*
 - VcLayerCollection* 1113
- NextLink*
 - VcLinkCollection* 1187
- NextLinkAppearance*
 - VcLinkAppearanceCollection* 1183
- NextMap*
 - VcMapCollection* 1200
- NextMapEntry*
 - VcMap* 1194

- NextNode*
 - VcNodeCollection* 1227
- NextNumericScale*
 - VcNumericScaleCollection* 1258
- NextTable*
 - VcTableCollection* 1388
- NextTimeScale*
 - VcTimeScaleCollection* 1426
- NextUpdateBehavior*
 - VcUpdateBehaviorCollection* 1435
- NodeRowInView*
 - VcNode* 1221
- NumericScaleByIndex*
 - VcNumericScaleCollection* 1259
- NumericScaleByName*
 - VcNumericScaleCollection* 1259
- Open*
 - VcGantt* 837
- OptimizeColumnWidth*
 - VcTable* 1385
- OptimizeTimeScaleStartEnd*
 - VcGantt* 838
- OutlineIndent*
 - VcNode* 1221
- OutlineOutdent*
 - VcNode* 1222
- PageLayout*
 - VcGantt* 838
- PrintDirectEx*
 - VcGantt* 839
- PrinterSetup*
 - VcGantt* 839
- PrintIt*
 - VcGantt* 840
- PrintPreview*
 - VcGantt* 840
- PrintToFile*
 - VcGantt* 840
- Process*
 - VcResourceScheduler2* 1352
- PutInOrderAfter*
 - VcCalendarProfile* 567
 - VcDateLine* 668
 - VcHistogram* 1030
 - VcInterval* 1060
 - VcLayer* 1107
 - VcLinkAppearance* 1178
 - VcUpdateBehavior* 1430
- RecalculateAllStructureCodes*
 - VcGantt* 841
- RelatedDataRecord*
 - VcDataRecord* 628
 - VcGroup* 972
 - VcLink* 1167
 - VcNode* 1222
- Remove*
 - DataObjectFiles* 469
 - VcBoxCollection* 501
 - VcBoxFormatCollection* 513
 - VcCalendarCollection* 538
 - VcCalendarGridCollection* 563
 - VcCalendarProfileCollection* 572
 - VcCurveCollection* 611
 - VcDataRecordCollection* 634
 - VcDateLineCollection* 674
 - VcDateLineGridCollection* 694
 - VcFilterCollection* 713
 - VcGroupLevelLayoutCollection* 1011
 - VcIntervalCollection* 1066
 - VcLayerCollection* 1114
 - VcLineFormatCollection* 1149
 - VcLinkAppearanceCollection* 1184
 - VcMapCollection* 1200

- VcUpdateBehaviorCollection* 1435
- RemoveFormatField*
 - VcBoxFormat* 507
 - VcLayerFormat* 1117
 - VcLineFormat* 1143
- RemoveSubCondition*
 - VcFilter* 707
- ReOptimizeNodes*
 - VcGroup* 973
- Reset*
 - VcGantt* 841
- SaveAsEx*
 - VcGantt* 842
- Schedule*
 - VcGantt* 843
- ScheduleProject*
 - VcScheduler* 1374
- ScrollComponentStartTo*
 - VcGantt* 843
- ScrollToDate*
 - VcGantt* 844
- ScrollToGroupLine*
 - VcGantt* 845
- ScrollToNode*
 - VcGantt* 845
- ScrollToNodeLine*
 - VcGantt* 846
- ScrollToValue*
 - VcHistogram* 1030
- SelectCalendarProfiles*
 - VcCalendarProfileCollection* 573
- SelectGroups*
 - VcGroupCollection* 977
- SelectLinks*
 - VcLinkCollection* 1187
- SelectMaps*
 - VcMapCollection* 1201
- SelectNodes*
 - VcNodeCollection* 1227
- SetData*
 - DataObject* 465
- SetPositionInView*
 - VcNode* 1223
- SetValues*
 - VcCurve* 604
- SetXYOffset*
 - VcBox* 494
- SetXYOffsetByTopLeftPixel*
 - VcBox* 494
- ShowExportGraphicsDialog*
 - VcGantt* 847
- SortGroups*
 - VcGantt* 848
- SortNodes*
 - VcGantt* 848
- SuspendUpdate*
 - VcGantt* 849
- TableByIndex*
 - VcTableCollection* 1389
- TableByName*
 - VcTableCollection* 1389
- TimeScaleByIndex*
 - VcTimeScaleCollection* 1427
- TimeScaleByName*
 - VcTimeScaleCollection* 1427
- Update*
 - VcBoxCollection* 501
 - VcCalendar* 532
 - VcCalendarCollection* 538
 - VcCalendarGridCollection* 563
 - VcCalendarProfileCollection* 573
 - VcDataRecordCollection* 635
 - VcDataTableCollection* 644
 - VcDateLineCollection* 675

- VcDateLineGridCollection* 694
- VcGroupLevelLayoutCollection* 1011
- VcIntervalCollection* 1067
- VcLayerCollection* 1114
- VcLegendView* 1139
- VcLinkAppearanceCollection* 1184
- VcMapCollection* 1201
- UpdateBehaviorByIndex*
 - VcUpdateBehaviorCollection* 1436
- UpdateBehaviorByName*
 - VcUpdateBehaviorCollection* 1436
- UpdateDataRecord*
 - VcDataRecord* 629
- UpdateGroup*
 - VcGroup* 973
- UpdateLink*
 - VcLink* 1168
- UpdateLinkRecord*
 - VcGantt* 850
- UpdateNode*
 - VcNode* 1224
- UpdateNodeRecord*
 - VcGantt* 850
- UpdateRowNumberFields*
 - VcGantt* 851
- Zoom*
 - VcGantt* 851
- Millimeter**
 - Eigenschaft von
 - VcMapEntry* 1208
- MinimumRowHeight**
 - Eigenschaft von
 - VcGantt* 769
- MinimumStartDataFieldIndex**
 - Eigenschaft von
 - VcLayer* 1087
- MinimumTextLineCount**
 - Eigenschaft von
 - VcBoxFormatField* 517
 - VcTableFormatField* 1406
- MinimumWidth**
 - Eigenschaft von
 - VcBoxFormatField* 517
 - VcLayerFormatField* 1122
- MinorTicks**
 - Eigenschaft von
 - VcNumericScale* 1246
 - VcRibbon* 1357
- MinorTicksEx**
 - Eigenschaft von
 - VcNumericScale* 1247
- Mode**
 - Eigenschaft von
 - VcWorldView* 1445
- ModificationsAllowed**
 - Eigenschaft von
 - VcGroupLevelLayout* 986
- Modus**
 - Knoten erzeugen 427, 428
 - Selektier-Modus 426
 - Verbindung ziehen 427
 - Verschiebe-Modus 426
- MouseProcessingEnabled**
 - Eigenschaft von
 - VcGantt* 770
- Moveable**
 - Eigenschaft von
 - VcBox* 486
 - VcDateLine* 663
 - VcLayer* 1087
- MoveAllSelectedNodes**
 - Eigenschaft von
 - VcGantt* 770

MoveLayersAsNodeWithShiftKey

Eigenschaft von

VcGantt 771

MoveMode

Eigenschaft von

VcNode 1216

MoveNodeAlways

Eigenschaft von

VcGantt 771

MoveNodeWhenMarked

Eigenschaft von

VcGantt 772

MultiplePrimaryKeysAllowed

Eigenschaft von

VcDataTable 638

MultiState

Eigenschaft von

VcTableFormatField 1407

MultiState-Felder 153

Muster 338

N

Name

Eigenschaft von

VcBox 486

VcBoxFormat 505

VcCalendar 526

VcCalendarGrid 547

VcCalendarProfile 566

VcCurve 591

VcDataTable 638

VcDataTableField 648

VcDateLine 663

VcDateLineGrid 682

VcDefinitionField 698

VcFilter 703

VcGroup 968

VcGroupLevelLayout 987

VcHistogram 1024

VcInterval 1052

VcLayer 1087

VcLineFormat 1142

VcLinkAppearance 1172

VcMap 1191

VcNumericScale 1247

VcTable 1382

VcTableFormat 1394

VcTimeScale 1419

VcUpdateBehavior 1429

Navigation

Diagramm 387

Tabelle 387

Netscape 20

New properties and API calls 188

NewNodesViaDoubleClick

Eigenschaft von

VcGantt 772

NextBox

Methode von

VcBoxCollection 500

NextCalendar

Methode von

VcCalendarCollection 538

NextCalendarGrid

Methode von

VcCalendarGridCollection 562

NextCalendarProfile

Methode von

VcCalendarProfileCollection 572

NextCurve

Methode von

VcCurveCollection 611

NextDataRecord

Methode von

- VcDataRecordCollection* 634
- NextDataTable**
 - Methode von
 - VcDataTableCollection* 643
- NextDataTableField**
 - Methode von
 - VcDataTableFieldCollection* 656
- NextDateLine**
 - Methode von
 - VcDateLineCollection* 674
- NextDateLineGrid**
 - Methode von
 - VcDateLineGridCollection* 693
- NextField**
 - Methode von
 - VcDataDefinitionTable* 618, 622
- NextFilter**
 - Methode von
 - VcFilterCollection* 713
- NextFormat**
 - Methode von
 - VcBoxFormatCollection* 512
 - VcLineFormatCollection* 1148
 - VcTableFormatCollection* 1399
- NextGroup**
 - Methode von
 - VcGroupCollection* 976
- NextGroupLevelLayout**
 - Methode von
 - VcGroupLevelLayoutCollection* 1010
- NextHistogram**
 - Methode von
 - VcHistogramCollection* 1036
- NextInterval**
 - Methode von
 - VcIntervalCollection* 1066
- NextLayer**
 - Methode von
 - VcLayerCollection* 1113
- NextLink**
 - Methode von
 - VcLinkCollection* 1187
- NextLinkAppearance**
 - Methode von
 - VcLinkAppearanceCollection* 1183
- NextMap**
 - Methode von
 - VcMapCollection* 1200
- NextMapEntry**
 - Methode von
 - VcMap* 1194
- NextNode**
 - Methode von
 - VcNodeCollection* 1227
- NextNumericScale**
 - Methode von
 - VcNumericScaleCollection* 1258
- NextTable**
 - Methode von
 - VcTableCollection* 1388
- NextTimeScale**
 - Methode von
 - VcTimeScaleCollection* 1426
- NextUpdateBehavior**
 - Methode von
 - VcUpdateBehaviorCollection* 1435
- Node**
 - siehe auch
 - VcNode* 1213
- NodeCalendarNameDataFieldIndex**
 - Eigenschaft von
 - VcGantt* 773
- NodeCollection**

- Eigenschaft von
 - VcGantt 773
 - VcGroup 968
 - siehe auch
 - VcNodeCollection 1225
- NodeDurationDataFieldIndex**
 - Eigenschaft von
 - VcGantt 773
- NodeEndDateDataFieldIndex**
 - Eigenschaft von
 - VcGantt 774
- NodeID**
 - Eigenschaft von
 - VcBox 487
- NodeLevelLayout**
 - Eigenschaft von
 - VcGantt 774
 - siehe auch
 - VcNodeLevelLayout 1229
- NodeRowInView**
 - Methode von
 - VcNode 1221
- NodeRowNumberDataFieldIndex**
 - Eigenschaft von
 - VcGantt 774
- NodesDataTableName**
 - Eigenschaft von
 - VcGantt 775
- NodeSeparationLinesVisible**
 - Eigenschaft von
 - VcHierarchyLevelLayout 1016
- NodesInHeader**
 - Eigenschaft von
 - VcGroup 969
- NodesInHeaders**
 - Eigenschaft von
 - VcGroupLevelLayout 987
- VcHierarchyLevelLayout 1016
- NodesInHeadersDataFieldIndex**
 - Eigenschaft von
 - VcGroupLevelLayout 987
 - VcHierarchyLevelLayout 1017
- NodesInHeadersMapName**
 - Eigenschaft von
 - VcGroupLevelLayout 988
 - VcHierarchyLevelLayout 1017
- NodesOverlaid**
 - Eigenschaft von
 - VcGroup 969
 - VcGroupLevelLayout 988
 - VcHierarchyLevelLayout 1018
- NodeStartDateDataFieldIndex**
 - Eigenschaft von
 - VcGantt 775
- NodeTooltipTextField**
 - Eigenschaft von
 - VcGantt 776
- NominalScaleMaximum**
 - Eigenschaft von
 - VcHistogram 1025
- NominalScaleMinimum**
 - Eigenschaft von
 - VcHistogram 1025
- NonWorkInterval**
 - Eigenschaft von
 - VcLayer 1088
- NonWorkIntervalBackColorAsARGB**
 - Eigenschaft von
 - VcLayer 1088
- NonWorkIntervalBackColorDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1089
- NonWorkIntervalBackColorMapName**

- Eigenschaft von
 - VcLayer 1090
- NonWorkIntervalLineColor**
 - Eigenschaft von
 - VcLayer 1091
- NonWorkIntervalLineColorDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1091
- NonWorkIntervalLineColorMapName**
 - Eigenschaft von
 - VcLayer 1091
- NonWorkIntervalLineThickness**
 - Eigenschaft von
 - VcLayer 1092
- NonWorkIntervalLineType**
 - Eigenschaft von
 - VcLayer 1093
- NonWorkIntervalPattern**
 - Eigenschaft von
 - VcLayer 1094
- NonWorkIntervalPatternColorAsARGB**
 - Eigenschaft von
 - VcLayer 1097
- NonWorkIntervalPatternColorDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1098
- NonWorkIntervalPatternDataFieldIndex**
 - Eigenschaft von
 - VcLayer 1098
- NonWorkIntervalPatternMapName**
 - Eigenschaft von
 - VcLayer 1099
- NonWorkIntervalShape**
 - Eigenschaft von
 - VcLayer 1100
- NoOfColumns**
 - Eigenschaft von
 - VcTable 1383
- NoOfInitialRows**
 - Eigenschaft von
 - VcGantt 776
- Number**
 - Eigenschaft von
 - VcMapEntry 1208
- NumericScale**
 - siehe auch
 - VcNumericScale 1242
- NumericScaleByIndex**
 - Methode von
 - VcNumericScaleCollection 1259
- NumericScaleByName**
 - Methode von
 - VcNumericScaleCollection 1259
- NumericScaleCollection**
 - Eigenschaft von
 - VcHistogram 1026
 - siehe auch
 - VcNumericScaleCollection 1256
- Numerische Skala**
 - 3D-Effekt 1252
 - aktiv 1257
 - Anzahl 1257
 - Einheiten 1253
 - Einheiten, Beschriftung 1254
 - Einheiten, Breite 1255
 - Hauptmarkierung 1245, 1246
 - Hintergrundfarbe des Musters 1248
 - Histogramm, zugehöriges 1244
 - Maximalwert 1025
 - Minimalwert 1025
 - Muster 1248

Musterfarbe 1248
 Name 1247
 Nebenmarkierung 1246, 1247
 Schriftfarbe 1244
 Schriftgrad 1243
 skalierbar 225
 Titel 1253
 über Index 1259

O

Object Draw Ereignisse

aktiviert 1100

ObjectDrawEventsEnabled

Eigenschaft von

VcLayer 1100

Objects as snap targets 187

Objekt

identifizieren 831, 833, 834

Objekte

DataObject 460
 DataObjectFiles 467
 VcBorderArea 471
 VcBoundingBox 472
 VcBox 480
 VcBoxCollection 496
 VcBoxFormat 503
 VcBoxFormatCollection 508
 VcBoxFormatField 514
 VcCalendar 525
 VcCalendarCollection 533
 VcCalendarGrid 539
 VcCalendarGridCollection 558
 VcCalendarProfile 565
 VcCalendarProfileCollection 568
 VcCurve 574
 VcCurveCollection 606
 VcDataDefinition 613

VcDataDefinitionTable 614, 619
 VcDataRecord 624
 VcDataRecordCollection 630
 VcDataTable 636
 VcDataTableCollection 639
 VcDataTableField 645
 VcDataTableFieldCollection 652
 VcDateLine 657
 VcDateLineCollection 669
 VcDateLineGrid 676
 VcDateLineGridCollection 689
 VcDefinitionField 696
 VcField 700
 VcFilter 701
 VcFilterCollection 708
 VcFilterSubCondition 714
 VcGantt 719
 VcGroup 964
 VcGroupCollection 974
 VcGroupLevelLayout 978
 VcGroupLevelLayoutCollection 1006
 VcHierarchyLevelLayout 1013
 VcHistogram 1023
 VcHistogramCollection 1032
 VcInfoWindow 1037
 VcInterval 1045
 VcIntervalCollection 1062
 VcLayer 1068
 VcLayerCollection 1109
 VcLayerFormat 1115
 VcLayerFormatField 1118
 VcLegendView 1131
 VcLineFormat 1140
 VcLineFormatCollection 1144
 VcLineFormatField 1150
 VcLink 1164
 VcLinkAppearance 1169

- VcLinkAppearanceCollection 1179
- VcLinkCollection 1185
- VcMap 1189
- VcMapCollection 1195
- VcMapEntry 1203
- VcNode 1213
- VcNodeCollection 1225
- VcNodeLevelLayout 1229
- VcNumericScale 1242
- VcNumericScaleCollection 1256
- VcPrinter 1261
- VcRect 1286
- VcResourceScheduler2 1289
- VcRibbon 1353
- VcScheduler 1367
- VcSection 1375
- VcTable 1381
- VcTableCollection 1386
- VcTableFormat 1390
- VcTableFormatCollection 1396
- VcTableFormatField 1400
- VcTimeScale 1418
- VcTimeScaleCollection 1424
- VcUpdateBehavior 1428
- VcUpdateBehaviorCollection 1431
- VcUpdateBehaviorContext 1438
- VcWorldView 1441
- ObserveDST**
 - Eigenschaft von
 - VcDateLineGrid 683
 - VcRibbon 1358
- OLE Drag & Drop 155**
 - beendet 855
 - Cursor während eines OLE-Drag-Vorgangs in Zielkomponente gesetzt 779
 - Daten auf Ziel abgelegt 856
 - Daten über Ziel gezogen 857
 - Drag-Vorgang ausgeführt 859
 - Ereignis des Drop-Ziels 858
 - Knoten aus anderer VARCHART-ActiveX-Komponente in die aktuelle ziehen erlaubt 780
 - OLEGiveFeedback 858
 - Phantom während eines OLE-Drag-Vorgangs 779
 - Ziehen über Steuerelement-Grenze hinaus erlaubt 778
- OLE Drag&Drop 777**
- OLECompleteDrag**
 - Ereignis von
 - VcGantt 855
- OLEDragDrop**
 - Ereignis von
 - VcGantt 856
- OLE-DragDrop**
 - Im Diagramm erlaubt 778
 - In der Tabelle erlaubt 778
- OLEDragHorizontalMovementAllowed**
 - Eigenschaft von
 - VcGantt 777
- OLEDragMode**
 - Eigenschaft von
 - VcGantt 777
- OLEDragOver**
 - Ereignis von
 - VcGantt 857
- OLEDragViaDiagram**
 - Eigenschaft von
 - VcGantt 778
- OLEDragViaTable**
 - Eigenschaft von
 - VcGantt 778
- OLEDragWithOwnMouseCursor**
 - Eigenschaft von

- VcGantt 779
- OLEDragWithPhantom**
 - Eigenschaft von VcGantt 779
- OLEDropMode**
 - Eigenschaft von VcGantt 780
- OLEGiveFeedback**
 - Ereignis von VcGantt 858
- OLESetData**
 - Ereignis von VcGantt 858
- OLEStartDrag**
 - Ereignis von VcGantt 859
- OnBoxCreate**
 - Ereignis von VcGantt 860
- OnBoxCreateComplete**
 - Ereignis von VcGantt 861
- OnBoxLClick**
 - Ereignis von VcGantt 861
- OnBoxLDbIClick**
 - Ereignis von VcGantt 861
- OnBoxModify**
 - Ereignis von VcGantt 862
- OnBoxModifyCompleteEx**
 - Ereignis von VcGantt 863
- OnBoxRClick**
 - Ereignis von VcGantt 863
- OnCalendarGridRClick**
 - Ereignis von VcGantt 864
- OnCurveLClick**
 - Ereignis von VcGantt 865
- OnCurveLDbIClick**
 - Ereignis von VcGantt 866
- OnCurveModifyComplete**
 - Ereignis von VcGantt 866
- OnCurveModifyEx**
 - Ereignis von VcGantt 866
- OnCurveModifyEx2**
 - Ereignis von VcGantt 867
- OnCurveModifyExAsString**
 - Ereignis von VcGantt 868
- OnCurveRClick**
 - Ereignis von VcGantt 869
- OnDataRecordCreate**
 - Ereignis von VcGantt 870
- OnDataRecordCreateComplete**
 - Ereignis von VcGantt 871
- OnDataRecordDelete**
 - Ereignis von VcGantt 872
- OnDataRecordDeleteComplete**
 - Ereignis von VcGantt 873
- OnDataRecordModify**

- Ereignis von
 - VcGantt 873
- OnDataRecordModifyComplete**
 - Ereignis von
 - VcGantt 874
- OnDataRecordNotFound**
 - Ereignis von
 - VcGantt 875
- OnDateLineModify**
 - Ereignis von
 - VcGantt 875
- OnDateLineRClick**
 - Ereignis von
 - VcGantt 875
- OnDeleteCurvePoint**
 - Ereignis von
 - VcGantt 876
- OnDeleteCurvePointEx**
 - Ereignis von
 - VcGantt 877
- OnDiagramLClick**
 - Ereignis von
 - VcGantt 878
- OnDiagramLDbIClick**
 - Ereignis von
 - VcGantt 878
- OnDiagramRClick**
 - Ereignis von
 - VcGantt 879
- OnGroupDelete**
 - Ereignis von
 - VcGantt 880
- OnGroupLClick**
 - Ereignis von
 - VcGantt 880
- OnGroupLDbIClick**
 - Ereignis von
 - VcGantt 881
- OnGroupModify**
 - Ereignis von
 - VcGantt 881
- OnGroupModifyComplete**
 - Ereignis von
 - VcGantt 882
- OnGroupModifyEx**
 - Ereignis von
 - VcGantt 883
- OnGroupRClick**
 - Ereignis von
 - VcGantt 883
- OnGroupsMark**
 - Ereignis von
 - VcGantt 884
- OnGroupsMarkComplete**
 - Ereignis von
 - VcGantt 885
- OnHelpRequested**
 - Ereignis von
 - VcGantt 885
- OnHistogramLClick**
 - Ereignis von
 - VcGantt 886
- OnHistogramLDbIClick**
 - Ereignis von
 - VcGantt 886
- OnHistogramRClick**
 - Ereignis von
 - VcGantt 887
- OnHistogramsHeight**
 - Ereignis von
 - VcGantt 888
- OnHistogramsHeightChanged**
 - Ereignis von
 - VcGantt 889

OnHistogramsHeightModifyEx

Ereignis von

VcGantt 889

OnInsertCurvePoint

Ereignis von

VcGantt 890

OnInsertCurvePointEx

Ereignis von

VcGantt 890

OnInteractionEndComplete

Ereignis von

VcGantt 891

OnInteractionModeChange

Ereignis von

VcGantt 892

OnInteractionModeChangeComplete

Ereignis von

VcGantt 893

OnInteractionObjectChangingComplete

Ereignis von

VcGantt 893

OnInteractionStartComplete

Ereignis von

VcGantt 895

OnLegendViewClosed

Ereignis von

VcGantt 896

OnLinkCreate

Ereignis von

VcGantt 896

OnLinkCreateComplete

Ereignis von

VcGantt 897

OnLinkDelete

Ereignis von

VcGantt 898

OnLinkDeleteComplete

Ereignis von

VcGantt 898

OnLinkLClickCltn

Ereignis von

VcGantt 899

OnLinkLDbIClickCltn

Ereignis von

VcGantt 900

OnLinkRClickCltn

Ereignis von

VcGantt 900

OnModifyComplete

Ereignis von

VcGantt 901

OnMouseDbIClk

Ereignis von

VcGantt 902

OnMouseDown

Ereignis von

VcGantt 902

OnMouseMove

Ereignis von

VcGantt 903

OnMouseUp

Ereignis von

VcGantt 904

OnNodeCreate

Ereignis von

VcGantt 905

OnNodeCreateCompleteEx

Ereignis von

VcGantt 905

OnNodeDelete

Ereignis von

VcGantt 906

OnNodeDeleteCompleteEx

- Ereignis von
 - VcGantt 907
- OnNodeLClick**
 - Ereignis von
 - VcGantt 907
- OnNodeLDbIClick**
 - Ereignis von
 - VcGantt 908
- OnNodeModifyComplete**
 - Ereignis von
 - VcGantt 909
- OnNodeModifyCompleteEx**
 - Ereignis von
 - VcGantt 909
- OnNodeModifyEx**
 - Ereignis von
 - VcGantt 910
- OnNodeRClick**
 - Ereignis von
 - VcGantt 911
- OnNodeResizeStart**
 - Ereignis von
 - VcGantt 912
- OnNodesMarkComplete**
 - Ereignis von
 - VcGantt 913
- OnNodesMarkEx**
 - Ereignis von
 - VcGantt 913
- OnNumericScaleLClick**
 - Ereignis von
 - VcGantt 914
- OnNumericScaleLDbIClick**
 - Ereignis von
 - VcGantt 915
- OnNumericScaleRClick**
 - Ereignis von
 - VcGantt 915
- OnNumericScaleRescale**
 - Ereignis von
 - VcGantt 916
- OnObjectDrawCompleteEx**
 - Ereignis von
 - VcGantt 917
- OnObjectDrawEx**
 - Ereignis von
 - VcGantt 918
- OnOptimizeTableColumnWidth**
 - Ereignis von
 - VcGantt 920
- OnPreScrollComponent**
 - Ereignis von
 - VcGantt 921
- OnPreScrollDiagramHor**
 - Ereignis von
 - VcGantt 923
- OnResourceSchedulingProgress**
 - Ereignis von
 - VcGantt 924
- OnResourceSchedulingWarning**
 - Ereignis von
 - VcGantt 925
- OnScrollComponent**
 - Ereignis von
 - VcGantt 928
- OnScrollDiagramHor**
 - Ereignis von
 - VcGantt 930
- OnSelectField**
 - Ereignis von
 - VcGantt 931
- OnShowCurveNameInMenu**
 - Ereignis von
 - VcGantt 932

OnShowDate

Ereignis von

VcGantt 933

OnShowInPlaceEditor

Ereignis von

VcGantt 933

OnStatusLineText

Ereignis von

VcGantt 935

OnSupplyTextEntry

Ereignis von

VcGantt 935

OnSupplyTextEntryAsVariant

Ereignis von

VcGantt 950

**OnSupplyTextEntry-Ereignis
aktivieren 750**

OnSupplyTextEntry-Ereignisse 225

OnTableCaptionLClick

Ereignis von

VcGantt 950

OnTableCaptionLDbIClick

Ereignis von

VcGantt 951

OnTableCaptionRClick

Ereignis von

VcGantt 951

OnTableColumnWidth

Ereignis von

VcGantt 952

OnTableColumnWidthModifyComplete

Ereignis von

VcGantt 953

OnTableWidth

Ereignis von

VcGantt 953

OnTableWidthModifyEx

Ereignis von

VcGantt 954

OnTimeScaleChangeComplete

Ereignis von

VcGantt 955

OnTimeScaleEndModifyComplete

Ereignis von

VcGantt 955

OnTimeScaleLClick

Ereignis von

VcGantt 955

OnTimeScaleLDbIClick

Ereignis von

VcGantt 956

OnTimeScaleRClick

Ereignis von

VcGantt 956

OnTimeScaleSectionRescaleCompleteEx

Ereignis von

VcGantt 957

OnTimeScaleSectionRescaleEx

Ereignis von

VcGantt 958

OnTimeScaleSectionStartModify

Ereignis von

VcGantt 958

OnTimeScaleStartModifyComplete

Ereignis von

VcGantt 959

OnToolTipText

Ereignis von

VcGantt 959

OnToolTipTextAsVariant

Ereignis von

VcGantt 961

OnToolTipText-Ereignisse 225**OnViewComponentsSizeModifyComplete**

Ereignis von

VcGantt 961

OnWorldViewClosed

Ereignis von

VcGantt 962

OnZoomFactorModifyComplete

Ereignis von

VcGantt 962

Open

Methode von

VcGantt 837

OperationDataTableName

Eigenschaft von

VcResourceScheduler2 1309

OperationLoadPerItemFieldIndex

Eigenschaft von

VcResourceScheduler2 1310

OperationMaximumInterruptionTimeFieldIndex

Eigenschaft von

VcResourceScheduler2 1310

OperationMinimumSupplementTimeFieldIndex

Eigenschaft von

VcResourceScheduler2 1311

OperationOverlapQuantityFieldIndex

Eigenschaft von

VcResourceScheduler2 1312

OperationPostLoadFieldIndex

Eigenschaft von

VcResourceScheduler2 1314

OperationPostOffsetFieldIndex

Eigenschaft von

VcResourceScheduler2 1314

OperationPreparationLoadFieldIndex

Eigenschaft von

VcResourceScheduler2 1315

OperationPreparationOffsetFieldIndex

Eigenschaft von

VcResourceScheduler2 1316

OperationResultEndDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1316

OperationResultPostEndDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1317

OperationResultPreparationStartDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1317

OperationResultProcessingTimeFieldIndex

Eigenschaft von

VcResourceScheduler2 1318

OperationResultSelectedTimingResourceIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1318

OperationResultStartDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1319

OperationResultStatusFieldIndex

Eigenschaft von

VcResourceScheduler2 1320

OperationRouteFieldIndex

Eigenschaft von

VcResourceScheduler2 1320

OperationSequenceNumberFieldIndex

Eigenschaft von

VcResourceScheduler2 1321

OperationStartLockDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1322

OperationTaskIDFieldIndex

Eigenschaft von

VcResourceScheduler2 1322

OperationWorkInProcessFieldIndex

Eigenschaft von

VcResourceScheduler2 1323

Operator

Eigenschaft von

VcFilterSubCondition 717

OptimizeColumnWidth

Methode von

VcTable 1385

OptimizedNodesSortDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 988

OptimizedNodesSortOrder

Eigenschaft von

VcGroupLevelLayout 989

OptimizeTimeScaleStartEnd

Methode von

VcGantt 838

Orientation

Eigenschaft von

VcPrinter 1275

Origin

Eigenschaft von

VcBox 487

OutgoingLinks

Eigenschaft von

VcNode 1217

OutlineIndent

Methode von

VcNode 1221

OutlineOutdent

Methode von

VcNode 1222

OutputFormatForCenterDate

Eigenschaft von

VcInfoWindow 1037

OutputFormatForDuration

Eigenschaft von

VcInfoWindow 1039

OutputFormatForEndDate

Eigenschaft von

VcInfoWindow 1040

OutputFormatForStartDate

Eigenschaft von

VcInfoWindow 1041

OverlaidNodesSortDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 989

OverlaidNodesSortOrder

Eigenschaft von

VcGroupLevelLayout 990

OverlapLayerEnabled

Eigenschaft von

VcGantt 781

OverlapLayerName

Eigenschaft von

VcGantt 781

OverloadResultsCalendarName

Eigenschaft von

VcCurve 591

Overload-Termine

Kalender für Intervalle 591

P

PagebreakMode

Eigenschaft von

VcGroupLevelLayout 990

VcHierarchyLevelLayout 1018

PageDescription

Eigenschaft von

- VcPrinter 1276
- PageDescriptionString**
 - Eigenschaft von
 - VcPrinter 1276
- PageFrame**
 - Eigenschaft von
 - VcPrinter 1277
- PageLayout**
 - Methode von
 - VcGantt 838
- PageNumberMode**
 - Eigenschaft von
 - VcPrinter 1277
- PageNumbers**
 - Eigenschaft von
 - VcPrinter 1278
- PagePaddingEnabled**
 - Eigenschaft von
 - VcPrinter 1278
- PaperSize**
 - Eigenschaft von
 - VcPrinter 1279
- Papiergröße 1279**
- ParentHWnd**
 - Eigenschaft von
 - VcLegendView 1134
 - VcWorldView 1446
- PartialLoadThreshold**
 - Eigenschaft von
 - VcGantt 782
- Pattern**
 - Eigenschaft von
 - VcCalendarGrid 547
 - VcInterval 1053
 - VcMapEntry 1209
- Pattern2Color**
 - Eigenschaft von
- VcCurve 592
- PatternBackgroundColorAsARGB**
 - Eigenschaft von
 - VcBoxFormatField 518
 - VcLineFormatField 1154
 - VcNumericScale 1248
 - VcRibbon 1358
 - VcTableFormatField 1407
- PatternBackgroundColorDataFieldIndex**
 - Eigenschaft von
 - VcLineFormatField 1154
 - VcTableFormatField 1408
- PatternBackgroundColorMapName**
 - Eigenschaft von
 - VcLineFormatField 1154
 - VcTableFormatField 1408
- PatternColor**
 - Eigenschaft von
 - VcCurve 592
- PatternColorAsARGB**
 - Eigenschaft von
 - VcBoxFormatField 519
 - VcCalendarGrid 550
 - VcInterval 1056
 - VcLayer 1101
 - VcLineFormatField 1155
 - VcNumericScale 1248
 - VcRibbon 1359
 - VcTableFormatField 1408
- PatternColorDataFieldIndex**
 - Eigenschaft von
 - VcCalendarGrid 551
 - VcLayer 1101
 - VcLineFormatField 1155
 - VcTableFormatField 1409
- PatternColorMapName**

- Eigenschaft von*
 - VcCalendarGrid* 551
 - VcLayer* 1102
 - VcLineFormatField* 1156
 - VcTableFormatField* 1409
- PatternDataFieldIndex**
 - Eigenschaft von*
 - VcCalendarGrid* 552
- PatternEx**
 - Eigenschaft von*
 - VcBoxFormatField* 519
 - VcLineFormatField* 1156
 - VcNumericScale* 1248
 - VcRibbon* 1359
 - VcTableFormatField* 1410
- PatternExDataFieldIndex**
 - Eigenschaft von*
 - VcLineFormatField* 1160
 - VcTableFormatField* 1413
- PatternExMapName**
 - Eigenschaft von*
 - VcLineFormatField* 1160
 - VcTableFormatField* 1413
- PatternMapName**
 - Eigenschaft von*
 - VcCalendarGrid* 552
- PDF-Dateien**
 - Export 163
- Performance 451**
- Period**
 - Eigenschaft von*
 - VcDateLineGrid* 683
- Pfad 753**
- Phantom**
 - typgerechte Anzeige von Verbindungen 740
- PhantomLayerHeight**
- Eigenschaft von*
 - VcGantt* 783
- PlanningEndDate**
 - Eigenschaft von*
 - VcResourceScheduler2* 1324
- PlanningStartDate**
 - Eigenschaft von*
 - VcResourceScheduler2* 1324
- PlanningStrategy**
 - Eigenschaft von*
 - VcResourceScheduler2* 1325
- PointsEquidistant**
 - Eigenschaft von*
 - VcCurve* 593
- Position**
 - Eigenschaft von*
 - VcRibbon* 1363
 - VcTable* 1383
- PredecessorLayerName**
 - Eigenschaft von*
 - VcLinkAppearance* 1173
- PredecessorNode**
 - Eigenschaft von*
 - VcLink* 1166
- PrePortSymbol**
 - Eigenschaft von*
 - VcLinkAppearance* 1173
- Primärschlüssel**
 - zusammengesetzt 638
- PrimaryKey**
 - Eigenschaft von*
 - VcDataTableField* 649
- PrintDate**
 - Eigenschaft von*
 - VcPrinter* 1279
- PrintDirectEx**
 - Methode von

VcGantt 839

Printer

Eigenschaft von

VcGantt 784

siehe auch

VcPrinter 1261

PrinterName

Eigenschaft von

VcPrinter 1280

PrinterSetup

Methode von

VcGantt 839

PrintIt

Methode von

VcGantt 840

PrintPreview

Methode von

VcGantt 840

PrintToFile

Methode von

VcGantt 840

Priorität

Boxen 314

Priority

Eigenschaft von

VcBox 488

VcCalendarGrid 552

VcDateLine 664

VcDateLineGrid 684

VcLayerFormatField 1122

Process

Methode von

VcResourceScheduler2 1352

Projektanfang 217

Projektdaten berechnen 1374

Projektende 217

Publizieren im Internet 66, 383, 429

PutInOrderAfter

Methode von

VcCalendarProfile 567

VcDateLine 668

VcHistogram 1030

VcInterval 1060

VcLayer 1107

VcLinkAppearance 1178

VcUpdateBehavior 1430

R

RecalculateAllStructureCodes

Methode von

VcGantt 841

Rect

siehe auch

VcRect 1286

ReferenceDate

Eigenschaft von

VcDateLineGrid 684

VcInfoWindow 1043

VcRibbon 1363

ReferencePoint

Eigenschaft von

VcBox 488

Referenzkurve 121

RelatedDataRecord

Methode von

VcDataRecord 628

VcGroup 972

VcLink 1167

VcNode 1222

RelationshipFieldIndex

Eigenschaft von

VcDataTableField 649

Remove

Methode von

- DataObjectFiles* 469
- VcBoxCollection* 501
- VcBoxFormatCollection* 513
- VcCalendarCollection* 538
- VcCalendarGridCollection* 563
- VcCalendarProfileCollection* 572
- VcCurveCollection* 611
- VcDataRecordCollection* 634
- VcDateLineCollection* 674
- VcDateLineGridCollection* 694
- VcFilterCollection* 713
- VcGroupLevelLayoutCollection* 1011
- VcIntervalCollection* 1066
- VcLayerCollection* 1114
- VcLineFormatCollection* 1149
- VcLinkAppearanceCollection* 1184
- VcMapCollection* 1200
- VcUpdateBehaviorCollection* 1435
- RemoveFormatField**
 - Methode von
 - VcBoxFormat* 507
 - VcLayerFormat* 1117
 - VcLineFormat* 1143
- RemoveSubCondition**
 - Methode von
 - VcFilter* 707
- ReOptimizeNodes**
 - Methode von
 - VcGroup* 973
- ReOptimizeNodesInGroupsEnabled**
 - Eigenschaft von
 - VcPrinter* 1280
- RepeatTableTimeScale**
 - Eigenschaft von
 - VcPrinter* 1281
- Reset**
 - Methode von
 - VcGantt* 841
- Resizing**
 - Eigenschaft von
 - VcBox* 489
- ResourceCalendarNameFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1326
- ResourceCapacityType**
 - Eigenschaft von
 - VcResourceScheduler2* 1327
- ResourceCapacityTypeFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1327
- ResourceConstraintTypeFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1329
- ResourceDataTableName**
 - Eigenschaft von
 - VcResourceScheduler2* 1329
- ResourceEfficiencyFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1331
- ResourceGroupDataTableName**
 - Eigenschaft von
 - VcResourceScheduler2* 1332
- ResourceGroupIDFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1333
- ResourceNameFieldIndex**
 - Eigenschaft von
 - VcResourceScheduler2* 1334
- ResourceResultLoadCurveNamePrefix**
 - Eigenschaft von
 - VcResourceScheduler2* 1335

ResourceResultStockCurveNamePrefix*Eigenschaft von**VcResourceScheduler2* 1336**ResourceScheduler2***Eigenschaft von**VcGantt* 784

siehe auch

VcResourceScheduler2 1289**ResourceSelectionStrategy***Eigenschaft von**VcResourceScheduler2* 1336**ResourceType***Eigenschaft von**VcResourceScheduler2* 1338**Ressourcenplanung 158****Ressourcen-Planung**

Arbeitsauftrag, Auftragsmenge 1344
 Arbeitsauftrag, Enddatum 1345
 Arbeitsauftrag, Fälligkeitsdatum 1341
 Arbeitsauftrag, Freigabedatum 1344
 Arbeitsauftrag, Planungsstrategie 1342
 Arbeitsauftrag, Reihenfolge 1346
 Arbeitsauftrag, Route 1347
 Arbeitsauftrag, Startdatum 1348
 Arbeitsauftrag, Tabellename 1340
 Arbeitsauftrag, Verarbeitungszeit 1347
 Arbeitsauftrag, Priorität 1343
 Aufgabe, Enddatum der Nachlaufphase 1345
 Aufgabe, Startdatum der Vorbereitungsphase 1346
 Auslastungskurve 1335
 Debug-Dateien 1350
 Durchführung 1352
 Effizienz 1332

Enddatum des Planungszeitraums 1324

Ereignisse 1302

Fertigstellung 1350

Kalender, Standardname 1303

Kalendername 1326

Kapazität 1327, 1328

Kapazitätsauslastung, vollständigen 1303

Lagerhaltungskurve 1336

linkData-Tabelle 1305

Operation, Datumsbindung 1322

Operation, Dauer 1318

Operation, Enddatum 1316

Operation, Enddatum der Nachlaufphase 1317

Operation, Faktor für Auftragsmenge 1310

Operation, Fertigstellung 1323

Operation, ID der zeitdauerbestimmenden Ressource 1319

Operation, maximale Unterbrechungszeit 1311

Operation, Mindest-Ergänzungszeit 1312

Operation, Nachlaufzeit 1314

Operation, Nachlaufzeit von Ressource unabhängig 1315

Operation, Reihenfolge 1321

Operation, Routen 1321

Operation, Standardwert für maximale Unterbrechungszeit 1302

Operation, Startdatum 1319

Operation, Startdatum der Vorbereitungsphase 1317

Operation, Status 1320

Operation, Überlappungsmenge 1313

Operation, Vorbereitungszeit 1315

Operation, Vorlaufzeit von Ressource unabhängig 1316

Operation, zugehöriges Task 1323

Operationen, Tabellenname 1309

Planungsfortschritt 925

Planungsstrategie 1326

Ressource, Auswahl 1336

Ressource, Gruppe 1334

Ressource, Höchstwert Belegungslimit 1296

Ressource, Name 1334

Ressource, Name der Gruppentabelle 1333

Ressource, Tabellenname 1331

Ressource, Tiefswert Belegungslimit 1296

Ressourcen-Auswahlstrategie 1299

Ressourcenbedingung 1329

Ressourcenkapazität absolut oder relativ 1297

Ressourcen-Planung

- Kalen~der bei
- Min~dest~er~gän~zungs~zeit 1300

Ressourcen-Typ 1338

Startdatum des Planungszeitraums 1325

Tasks, Anzahl der 1339

Toleranz des Fälligkeitsdatums 1348

Toleranz des festen Startdatums 1349

Toleranz des Freigabedatums 1349

Verbindung 1306

Verbindung, Nachfolgeoperation 1308

Verbindung, Nachfolgerauftrag 1308

Verbindung, Vorgängerauftrag 1307

Verbindung, Vorgängeroperation 1306

Warnungen 925

Zeiteinheit, grundlegende 1301

zeitliche Schrittweite 1301

Zuweisung, Datensatzerzeugung 1294

Zuweisung, Operation 1297

Zuweisung, Ressource 1298

Zuweisung, Tabellenname 1293

Zuweisungstabelle, Zuweisung sichtbar 1295

RestoreAutoCollapsedGroups

Eigenschaft von

- VcGroupLevelLayout 990
- VcHierarchyLevelLayout 1018

RestoreAutoExpandedGroups

Eigenschaft von

- VcGroupLevelLayout 991
- VcHierarchyLevelLayout 1019

ResultProcessingStepCount

Eigenschaft von

- VcResourceScheduler2 1339

Return Status 100

Ribbon

Eigenschaft von

- VcSection 1378
- VcTimeScale 1420

siehe auch

- VcRibbon 1353

Right

Eigenschaft von

- VcRect 1288

RightMargin

Eigenschaft von

- VcLayerFormatField 1123
- VcTableFormatField 1414

RightTable

Eigenschaft von

- VcGantt 784

RightTableDiagramWidthRatio

Eigenschaft von

VcGantt 785

RightTableDiagramWidthRatioEx

Eigenschaft von

VcGantt 785

RoundedLinkSlantsEnabled

Eigenschaft von

VcGantt 786

RoutingType

Eigenschaft von

VcLinkAppearance 1174

RowBackColorAsARGB

Eigenschaft von

VcGroupLevelLayout 991

VcHistogram 1026

VcNodeLevelLayout 1230

RowBackColorDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 992

VcNodeLevelLayout 1231

RowBackColorMapName

Eigenschaft von

VcGroupLevelLayout 992

VcNodeLevelLayout 1231

RowHeightReductionEnabled

Eigenschaft von

VcGantt 786

RowMargins

Eigenschaft von

VcGantt 787

RowPattern

Eigenschaft von

VcGroupLevelLayout 992

VcHistogram 1026

VcNodeLevelLayout 1231

RowPatternColorAsARGB

Eigenschaft von

VcGroupLevelLayout 996

VcHistogram 1027

VcNodeLevelLayout 1235

RowPatternColorDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 996

VcNodeLevelLayout 1235

RowPatternColorMapName

Eigenschaft von

VcGroupLevelLayout 997

VcNodeLevelLayout 1236

RowPatternDataFieldIndex

Eigenschaft von

VcGroupLevelLayout 997

VcNodeLevelLayout 1236

RowPatternMapName

Eigenschaft von

VcGroupLevelLayout 997

VcNodeLevelLayout 1236

RowsBelowCollapsed

Eigenschaft von

VcGroup 969

Rückgabewerte 100**S****Sash**

3D-Ansicht an/aus 787

Dicke 787

Doppelte Phantomlinie beim
interaktiven Verschieben an/aus
806**Sash3DStyleEnabled**

Eigenschaft von

VcGantt 787

SashThickness

Eigenschaft von

- VcGantt 787
- SaveAsEx**
 - Methode von
 - VcGantt 842
- ScalingMode**
 - Eigenschaft von
 - VcPrinter 1281
- Schedule**
 - Methode von
 - VcGantt 843
- ScheduledProjectEndDate**
 - Eigenschaft von
 - VcScheduler 1371
- ScheduledProjectStartDate**
 - Eigenschaft von
 - VcScheduler 1372
- ScheduleProject**
 - Methode von
 - VcScheduler 1374
- Scheduler**
 - Eigenschaft von
 - VcGantt 788
 - siehe auch
 - VcScheduler 1367
- ScheduleSuccessorsOnlyEnabled**
 - Eigenschaft von
 - VcScheduler 1372
- Schichtkalender**
 - Zeitstreifenbeschriftung 1058
- Schnittmarkierungen 418**
- Schriftattribute**
 - Dialogfelder 747
- Schriften**
 - Anti-Aliasing 754
- Scroll events**
 - aktivieren/deaktivieren 225
- ScrollBarMode**
 - Eigenschaft von
 - VcLegendView 1135
 - VcWorldView 1446
- ScrollComponentStartTo**
 - Methode von
 - VcGantt 843
- Scrollen 921, 923, 928, 930**
 - zu der Zeile eines Gruppenknotens 845
 - zu der Zeile eines Knotens 846
 - zu einem bestimmten Datum 844
 - zu einem Knoten 845
 - zu Wert im Histogramm 1030
- ScrollEventsEnabled**
 - Eigenschaft von
 - VcGantt 788
- ScrollToDate**
 - Methode von
 - VcGantt 844
- ScrollToGroupLine**
 - Methode von
 - VcGantt 845
- ScrollToNode**
 - Methode von
 - VcGantt 845
- ScrollToNodeLine**
 - Methode von
 - VcGantt 846
- ScrollToValue**
 - Methode von
 - VcHistogram 1030
- SecondsPerWorkday**
 - Eigenschaft von
 - VcCalendar 527
- Section**
 - Eigenschaft von
 - VcTimeScale 1420

- siehe auch
 - VcSection 1375
- Seite einrichten 428, 838**
- Seitenansicht 1282**
- Seitennummerierung 419**
- Seitenränder 420**
- SelectCalendarProfiles**
 - Methode von
 - VcCalendarProfileCollection 573
- SelectedRowBackColorAsARGB**
 - Eigenschaft von
 - VcGantt 788
- SelectGroups**
 - Methode von
 - VcGroupCollection 977
- SelectLinks**
 - Methode von
 - VcLinkCollection 1187
- SelectMaps**
 - Methode von
 - VcMapCollection 1201
- SelectNodes**
 - Methode von
 - VcNodeCollection 1227
- Selektier-Modus 426**
- SeparationLineColor**
 - Eigenschaft von
 - VcGroupLevelLayout 998
 - VcHierarchyLevelLayout 1019
 - VcNodeLevelLayout 1237
 - VcTableFormat 1395
- SeparationLineColorDataFieldIndex**
 - Eigenschaft von
 - VcGroupLevelLayout 998
- SeparationLineColorMapName**
 - Eigenschaft von
 - VcGroupLevelLayout 998
- SeparationLineInterval**
 - Eigenschaft von
 - VcNodeLevelLayout 1237
- SeparationLineThickness**
 - Eigenschaft von
 - VcGroupLevelLayout 999
 - VcHierarchyLevelLayout 1020
 - VcNodeLevelLayout 1237
- SeparationLineType**
 - Eigenschaft von
 - VcGroupLevelLayout 1000
 - VcHierarchyLevelLayout 1020
 - VcNodeLevelLayout 1238
- SetData**
 - Methode von
 - DataObject 465
- SetPositionInView**
 - Methode von
 - VcNode 1223
- SetValues**
 - Methode von
 - VcCurve 604
- SetXYOffset**
 - Methode von
 - VcBox 494
- SetXYOffsetByTopLeftPixel**
 - Methode von
 - VcBox 494
- ShowCalendarGrids**
 - Eigenschaft von
 - VcGroupLevelLayout 1001
 - VcHistogram 1027
 - VcNodeLevelLayout 1239
 - VcTimeScale 1421
- ShowDateGrids**
 - Eigenschaft von
 - VcTimeScale 1421

ShowDateLineGrids

Eigenschaft von
VcGroupLevelLayout 1001

ShowDateLines

Eigenschaft von
VcGroupLevelLayout 1002
VcNodeLevelLayout 1240

ShowExportGraphicsDialog

Methode von
VcGantt 847

ShowGroupNodes

Eigenschaft von
VcGroupLevelLayout 1002

ShowNonWorkInterval

Eigenschaft von
VcGantt 789

ShowSeparationLines

Eigenschaft von
VcGroupLevelLayout 1002
VcHierarchyLevelLayout 1022
VcNodeLevelLayout 1240

ShowSeparationLinesAtTop

Eigenschaft von
VcGroupLevelLayout 1003
VcNodeLevelLayout 1240

ShowSnapLines

Eigenschaft von
VcGantt 789

ShowSnapMarkings

Eigenschaft von
VcGantt 790

ShowTimeScaleDialog

Eigenschaft von
VcGantt 790

ShowToolTip

Eigenschaft von
VcGantt 791

Sizeable

Eigenschaft von
VcLayer 1102

Skalenstreifentyp 369

Snap target DATE LINE 189

Snap target LAYER 188

SnapTarget

Eigenschaft von
VcCalendarGrid 553
VcDateLine 664
VcDateLineGrid 685

SnapTargetMode

Eigenschaft von
VcNode 1217

SnapTargetNodesSelectionMode

Eigenschaft von
VcGantt 791

SortDataFieldIndex

Eigenschaft von
VcGroupLevelLayout 1003
VcNodeLevelLayout 1241

SortField

Eigenschaft von
VcGantt 792

SortGroups

Methode von
VcGantt 848

Sortieren

Sortiereigenschaften ausfragen 774

Sortierung 165, 299, 848

Sortierfelder 792
Sortierrichtung 792

SortNodes

Methode von
VcGantt 848

SortOrder

Eigenschaft von

VcGantt 792

VcGroupLevelLayout 1003

VcNodeLevelLayout 1241

Specification

Eigenschaft von

VcBox 489

VcBoxFormat 506

VcCalendar 527

VcCalendarGrid 553

VcCalendarProfile 566

VcCurve 593

VcDateLine 665

VcFilter 703

VcGroupLevelLayout 1004

VcInterval 1056

VcLayer 1103

VcLineFormat 1142

VcLinkAppearance 1175

VcMap 1191

VcUpdateBehavior 1429

Sprachanpassung 172

StackReferenceName

Eigenschaft von

VcCurve 594

StartDateFieldIndex

Eigenschaft von

VcLayer 1103

StartDate

Eigenschaft von

VcSection 1379

StartDateForAutomaticScheduling

Eigenschaft von

VcScheduler 1373

StartDateNotEarlierThanDataFieldIndex

Eigenschaft von

VcScheduler 1373

StartDateTime

Eigenschaft von

VcInterval 1057

StartMonth

Eigenschaft von

VcInterval 1057

StartSnapTarget

Eigenschaft von

VcCalendarGrid 554

VcLayer 1103

StartTime

Eigenschaft von

VcInterval 1058

StartUpSinglePage

Eigenschaft von

VcPrinter 1282

StartWeekday

Eigenschaft von

VcInterval 1058

Statuszeilentext 935

Stichtagdatum 414

Stichtaglinie

bearbeiten 374

Beschriftungsposition 660

DateLineCollection 670

Drehung der Beschriftung um 90 Grad 666

Einrastziel am Datum 664

immer aktuelles Datum 658

interaktiv verschiebbar 663

Liniendicke 661

Linienfarbe 660

Linientyp 662

Name 663

Position 658

Priorität 664

Schriftattribute 659

Schriftfarbe 659
 sichtbar 666
 Text 665

Stichtaglinie, individuelle
 Datenfeld 659, 667
 Zuordnungstabelle 667

Stichtaglinien 173, 374, 414, 436, 1377
 festlegen 371
 interaktiv verschieben 875

Strg+C, Strg+X und Strg+V 741

Strg-C, -X und -V verarbeiten 223

StringsCaseSensitive
 Eigenschaft von
 VcFilter 704

SubCondition
 Eigenschaft von
 VcFilter 704

SubConditionCount
 Eigenschaft von
 VcFilter 705

SubGroups
 Eigenschaft von
 VcGroup 970

SubRowMargins
 Eigenschaft von
 VcGantt 793

SuccessorLayerName
 Eigenschaft von
 VcLinkAppearance 1176

SuccessorNode
 Eigenschaft von
 VcLink 1166

SuccPortSymbol
 Eigenschaft von
 VcLinkAppearance 1176

SummaryBarsVisible

Eigenschaft von
 VcGantt 793
 VcGroupLevelLayout 1004
 VcHierarchyLevelLayout 1022

Summenbalken 112

anzeigen 297
 sichtbar 793

SuperGroup

Eigenschaft von
 VcGroup 970
 VcNode 1218

SuppressTruncatedText

Eigenschaft von
 VcLayerFormatField 1124

SuspendUpdate

Methode von
 VcGantt 849

T

Tabelle 178

aktiv 1387
 Anzahl 1387
 Anzahl der Spalten 1383
 bearbeiten 329
 Breitenverhältnis Tabelle/Diagramm 407
 festlegen 327
 Iteration 1388
 Iteration-Initial 1388
 Name 1382
 Position 1383
 rechte Tabelle 784
 Saltenbreite optimieren 1385
 sichtbar 1384
 Spalten 329
 Spaltenbreite 1382
 Spaltenbreite ändern 408

- Spaltenbreite automatisch anpassen 731
- Spaltenbreite optimieren 223, 920
- Spaltentitel 1381
- Tabellenbreite 953
- Tabellenformat bearbeiten 332
- Tabellenformate 329, 1383
- Tabellenspaltenbreite 952, 953
- über Index 1389
- Verhältnis der Breite der linken Tabelle zur Gesamtbreite des Diagramms 795
- Verhältnis der Breite der rechten Tabelle zur Gesamtbreite des Diagramms 785
- Tabellenbearbeitung**
 - erweitert 222, 752
- Tabellendatenfeld**
 - Anzahl in Collection 653
 - Iteration, Enumerator Objekt 653
 - Iteration, Erstwert 656
 - Iteration, Folgewerte 656
 - kopieren 654
 - über Index 655
 - über Name 655
 - zu Collection hinzufügen 653
- Tabellenformat**
 - 3D-Effekt 1395
 - Anzahl der Tabellenspalten 1393
 - Anzeige von +/- 1391
 - Einzug, betroffene Spalte 1394
 - Einzugsgröße für Text 1394
 - Enumerator-Objekt 1391
 - Feld über Index 1393
 - Filter 1392
 - Name 1394
 - Trennlinien sichtbar 1392
 - Trennlinienfarbe 1395
 - über Index 1398
- Tabellenformatfeld**
 - Füllmuster 1156, 1410
 - Hintergrundfarbe des Musters 1407
 - maximale Zeilenzahl 1406
 - minimale Zeilenzahl 1406
 - Musterfarbe 1409
- Tabellenzeile**
 - einfügen 409
- Table**
 - Eigenschaft von*
 - VcGantt 794
 - siehe auch
 - VcTable 1381
- TableByIndex**
 - Methode von*
 - VcTableCollection 1389
- TableByName**
 - Methode von*
 - VcTableCollection 1389
- TableCollection**
 - Eigenschaft von*
 - VcGantt 794
 - siehe auch
 - VcTableCollection 1386
- TableColumnRanges**
 - Eigenschaft von*
 - VcPrinter 1282
- TableDiagramWidthRatio**
 - Eigenschaft von*
 - VcGantt 794
- TableDiagramWidthRatioEx**
 - Eigenschaft von*
 - VcGantt 795
- TableFormat**
 - siehe auch
 - VcTableFormat 1390

TableFormatCollection

Eigenschaft von

VcTable 1383

siehe auch

VcTableFormatCollection 1396

TableFormatField

siehe auch

VcTableFormatField 1400

TableWidthAdoptionFromViewOnScreen

Eigenschaft von

VcPrinter 1283

TaskDataTableName

Eigenschaft von

VcResourceScheduler2 1339

TaskDueDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1341

TaskPlanningStrategyFieldIndex

Eigenschaft von

VcResourceScheduler2 1341

TaskPriorityFieldIndex

Eigenschaft von

VcResourceScheduler2 1342

TaskQuantityFieldIndex

Eigenschaft von

VcResourceScheduler2 1343

TaskReleaseDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1344

TaskResultEndDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1345

TaskResultPostEndDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1345

TaskResultPreparationStartDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1346

TaskResultProcessingStepFieldIndex

Eigenschaft von

VcResourceScheduler2 1346

TaskResultProcessingTimeFieldIndex

Eigenschaft von

VcResourceScheduler2 1347

TaskResultRouteFieldIndex

Eigenschaft von

VcResourceScheduler2 1347

TaskResultStartDateFieldIndex

Eigenschaft von

VcResourceScheduler2 1348

Taste

Ereignis beim Drücken 853

Ereignis beim Drücken und Loslassen
854

Ereignis beim Loslassen 854

Technische Voraussetzungen 13

Teildiagramm 428, 431

Temporäre Datendatei 221

Terminlinie

über Index 672

Terminliniengitter

Beschriftung mittig 678

Beschriftung oben 678

Beschriftung unten 677

horizontale Ausrichtung der
Beschriftung 679

Liniendicke 681

Linienfarbe 679, 680, 1084, 1091

Linienfarbenzuordnungstabelle 680

Linientyp 681

Name 682

Periode 683

- Priorität 684
- Referenzdatum 677, 686, 1043
- Rerferenzdatum 685, 1043
- sichtbar 687
- Sommerzeit berücksichtigen 683, 1358
- Zeiteinheit 686
- Terminstichlinie**
 - Reihenfolge 668
- Text**
 - Eigenschaft von
 - VcBoundingBox 477
 - VcDateLine 665
 - VcInterval 1058
- TextAlignment**
 - Eigenschaft von
 - VcRibbon 1363
- Textausgabe 936, 950**
- TextDataFieldIndex**
 - Eigenschaft von
 - VcLayerFormatField 1124
 - VcLineFormatField 1160
 - VcTableFormatField 1414
- Texte**
 - festlegen 376
- Texte, Grafiken und Legende festlegen 376**
- TextFont**
 - Eigenschaft von
 - VcBoundingBox 478
 - VcBoxFormatField 523
 - VcLayerFormatField 1125
 - VcLineFormatField 1161
 - VcTableFormatField 1415
- TextFontColor**
 - Eigenschaft von
 - VcBoxFormatField 523
- TextFontColorDataFieldIndex**
 - Eigenschaft von
 - VcLayerFormatField 1125
 - VcLineFormatField 1161
 - VcTableFormatField 1415
- TextFontColorMapName**
 - Eigenschaft von
 - VcLayerFormatField 1126
 - VcLineFormatField 1162
 - VcTableFormatField 1416
- TextFontDataFieldIndex**
 - Eigenschaft von
 - VcLayerFormatField 1126
 - VcLineFormatField 1162
 - VcTableFormatField 1416
- TextFontMapName**
 - Eigenschaft von
 - VcLayerFormatField 1126
 - VcLineFormatField 1163
 - VcTableFormatField 1416
- TextLineCount**
 - Eigenschaft von
 - VcLayerFormatField 1127
 - VcLineFormatField 1163
- TextLineCountDataFieldIndex**
 - Eigenschaft von
 - VcLayerFormatField 1127
- TextLineCountMapName**
 - Eigenschaft von
 - VcLayerFormatField 1128
- ThreeDEffect**
 - Eigenschaft von
 - VcLayer 1104
 - VcNumericScale 1252

VcTableFormat 1395

VcTimeScale 1422

TickColor

Eigenschaft von

VcNumericScale 1252

VcRibbon 1364

TickPosition

Eigenschaft von

VcRibbon 1364

TimeColumnEndDate

Eigenschaft von

VcPrinter 1283

TimeColumnStartDate

Eigenschaft von

VcPrinter 1284

TimeScale

siehe auch

VcTimeScale 1418

TimeScaleAdjustment

Eigenschaft von

VcPrinter 1284

TimeScaleByIndex

Methode von

VcTimeScaleCollection 1427

TimeScaleByName

Methode von

VcTimeScaleCollection 1427

TimeScaleCollection

Eigenschaft von

VcGantt 795

siehe auch

VcTimeScaleCollection 1424

TimeScaleEnd

Eigenschaft von

VcGantt 796

TimeScaleStart

Eigenschaft von

VcGantt 796

TimeUnit

Eigenschaft von

VcCurve 594

VcGantt 797

VcInterval 1059

TimeUnitsPerStep

Eigenschaft von

VcGantt 797

Title

Eigenschaft von

VcNumericScale 1252

ToleranceTimeOnASAPDueDates

Eigenschaft von

VcResourceScheduler2 1348

ToleranceTimeOnJITReleaseDates

Eigenschaft von

VcResourceScheduler2 1349

ToleranceTimeOnStartLockDates

Eigenschaft von

VcResourceScheduler2 1349

Tooltip 791, 960, 961

Datenfeld für Text 233, 776

ToolTip

Dauer bis zur Anzeige 799

Erscheinungsdauer 798

Verschwinden auf Klick 799

Wechseldauer 798

ToolTipChangeDuration

Eigenschaft von

VcGantt 798

ToolTipDuration

Eigenschaft von

VcGantt 798

ToolTipPointerDuration

Eigenschaft von

VcGantt 799

Tooltips 225

zur Laufzeit 180

ToolTipShowAfterClick

Eigenschaft von

VcGantt 799

Top

Eigenschaft von

VcLegendView 1135

VcRect 1288

VcWorldView 1447

TopActualValue

Eigenschaft von

VcLegendView 1136

VcWorldView 1447

TopMargin

Eigenschaft von

VcLayerFormatField 1128, 1129

VcTableFormatField 1417

TotalFloatDataFieldIndex

Eigenschaft von

VcScheduler 1373

Trackingbereich

Füllmuster 800

Hintergrundfarbe 800

Musterfarbe 803

TrackingSpaceBackColorAsARGB

Eigenschaft von

VcGantt 800

TrackingSpacePattern

Eigenschaft von

VcGantt 800

TrackingSpacePatternColorAsARGB

Eigenschaft von

VcGantt 803

Treeview-Stil 328**Trennlinien**

Intervall 1237

TurningAnnotationEnabled

Eigenschaft von

VcDateLine 666

VcDateLineGrid 685

Type

Eigenschaft von

VcBoundingBox 479

VcBoxFormatField 523

VcCalendar 527

VcCalendarProfile 566

VcDataTableField 650

VcDefinitionField 699

VcInterval 1059

VcMap 1191

VcRibbon 1364

VcTableFormatField 1417

VcUpdateBehaviorContext 1439

U**Unicode 181****Unit**

Eigenschaft von

VcDateLineGrid 686

VcNumericScale 1253

VcSection 1379

UnitEx

Eigenschaft von

VcNumericScale 1254

UnitLabel

Eigenschaft von

VcNumericScale 1254

UnitSeparation

Eigenschaft von

VcRibbon 1365

UnitsPerStep

Eigenschaft von

VcCurve 595

UnitWidth

Eigenschaft von

VcNumericScale 1255

VcSection 1380

UnitWidthEx

Eigenschaft von

VcSection 1380

Update

Methode von

VcBoxCollection 501

VcCalendar 532

VcCalendarCollection 538

VcCalendarGridCollection 563

VcCalendarProfileCollection 573

VcDataRecordCollection 635

VcDataTableCollection 644

VcDateLineCollection 675

VcDateLineGridCollection 694

VcGroupLevelLayoutCollection
1011

VcIntervalCollection 1067

VcLayerCollection 1114

VcLegendView 1139

VcLinkAppearanceCollection 1184

VcMapCollection 1201

UpdateBehavior

siehe auch

VcUpdateBehavior 1428

UpdateBehaviorByIndex

Methode von

VcUpdateBehaviorCollection 1436

UpdateBehaviorByName

Methode von

VcUpdateBehaviorCollection 1436

UpdateBehaviorCollection

Eigenschaft von

VcGantt 804

siehe auch

VcUpdateBehaviorCollection 1431

UpdateBehaviorContext

siehe auch

VcUpdateBehaviorContext 1438

UpdateBehaviorName

Eigenschaft von

VcBox 490

VcCurve 595

VcDateLine 666

VcNode 1218

VcNumericScale 1255

VcTable 1384

VcTimeScale 1422

VcWorldView 1448

UpdateDataRecord

Methode von

VcDataRecord 629

UpdateGroup

Methode von

VcGroup 973

UpdateLink

Methode von

VcLink 1168

UpdateLinkRecord

Methode von

VcGantt 850

UpdateMode

Eigenschaft von

VcUpdateBehaviorContext 1440

UpdateNode

Methode von

VcNode 1224

UpdateNodeRecord

Methode von

VcGantt 850

UpdateRowNumberFields

- Methode von
 - VcGantt 851
 - URL 739**
 - UsedAsOverlapLayer**
 - Eigenschaft von
 - VcLayer 1104
 - UseGraphicalAttributes**
 - Eigenschaft von
 - VcInterval 1059
 - UseGraphicalAttributesOfIntervals**
 - Eigenschaft von
 - VcCalendarGrid 554
 - UseHigherDiagramHistogramHeightRatioPrecision**
 - Eigenschaft von
 - VcGantt 804
 - UseHigherTableDiagramWidthRatioPrecision**
 - Eigenschaft von
 - VcGantt 805
 - UseReferenceDate**
 - Eigenschaft von
 - VcDateLineGrid 686
 - VcInfoWindow 1043
 - VcRibbon 1366
 - UseSnapTargetsInInteractions**
 - Eigenschaft von
 - VcGantt 805
 - UseTwinLineSashPhantom**
 - Eigenschaft von
 - VcGantt 806
- # V
- ValencyDataFieldIndex**
 - Eigenschaft von
 - VcCurve 596
 - VARCHART XGantt**
 - automatisch skalieren 29
 - im Formular platzieren 29
 - VcBorderArea 471**
 - BorderBox 471
 - VcBorderBox 472**
 - Alignment 472
 - GraphicsFileName 473
 - LegendElementsArrangement 474
 - LegendElementsBottomMargin 474
 - LegendElementsMaximumColumnCount 474
 - LegendElementsMaximumRowCount 475
 - LegendElementsTopMargin 475
 - LegendFont 475
 - LegendTitle 476
 - LegendTitleFont 476
 - LegendTitleVisible 477
 - Text 477
 - TextFont 478
 - Type 479
 - VcBox 480**
 - AnchoringInteractionsAllowed 481
 - AnchoringLineVisible 481
 - AnchorToNode 491
 - FieldText 482
 - FormatName 482
 - GetActualExtent 491
 - GetTopLeftPixel 492
 - GetXYOffset 493
 - GetXYOffsetAsVariant 493
 - IdentifyFormatField 493
 - LineColor 483
 - LineThickness 483
 - LineType 484
 - MarkBox 485
 - Moveable 486

Name 486
NodeID 487
Origin 487
Priority 488
ReferencePoint 488
Resizing 489
SetXYOffset 494
SetXYOffsetByTopLeftPixel 494
Specification 489
UpdateBehaviorName 490
Visible 490

VcBoxCollection 496

_NewEnum 496
Add 497
AddBySpecification 498
BoxByIndex 498
BoxByName 499
Copy 499
Count 497
FirstBox 500
NextBox 500
Remove 501
Update 501

VcBoxFormat 503

_NewEnum 503
CopyFormatField 506
FieldsSeparatedByLines 504
FormatField 504
FormatFieldCount 505
Name 505
RemoveFormatField 507
Specification 506

VcBoxFormatCollection 508

_NewEnum 508
Add 509
AddBySpecification 510
Copy 510

Count 509
FirstFormat 511
FormatByIndex 511
FormatByName 512
NextFormat 512
Remove 513

VcBoxFormatField 514

Alignment 514
FormatName 515
GraphicsHeight 515
Index 516
MaximumTextLineCount 516
MinimumTextLineCount 517
MinimumWidth 517
PatternBackgroundColorAsARGB 518
PatternColorAsARGB 519
PatternEx 519
TextFont 523
TextFontColor 523
Type 523

VcCalendar 525

AddDuration 528
CalcDuration 528
CalendarProfileCollection 526
Clear 529
GetEndOfPreviousWorktime 529
GetNextIntervalBorder 530
GetPreviousIntervalBorder 530
GetStartOfInterval 531
GetStartOfNextWorktime 531
IntervalCollection 526
IsWorktime 531
Name 526
SecondsPerWorkday 527
Specification 527
Type 527

- Update* 532
- VcCalendarCollection 533**
 - _NewEnum* 533
 - Active* 534
 - Add* 535
 - AddBySpecification* 535
 - CalendarByIndex* 536
 - CalendarByName* 536
 - Copy* 537
 - Count* 535
 - FirstCalendar* 537
 - NextCalendar* 538
 - Remove* 538
 - Update* 538
- VcCalendarGrid 539**
 - BackColorAsARGB* 540
 - BackColorDataFieldIndex* 541
 - BackColorMapName* 541
 - CalendarName* 541
 - CalendarNameDataFieldIndex* 542
 - CalendarNameMapName* 542
 - EndSnapTarget* 542
 - Identifiable* 543
 - IdentifyInterval* 556
 - IdentifyIntervalAsVariant* 557
 - LineColor* 544
 - LineColorDataFieldIndex* 544
 - LineColorMapName* 544
 - LineThickness* 545
 - LineType* 546
 - Name* 547
 - Pattern* 547
 - PatternColorAsARGB* 550
 - PatternColorDataFieldIndex* 551
 - PatternColorMapName* 551
 - PatternDataFieldIndex* 552
 - PatternMapName* 552
- Priority* 552
- SnapTarget* 553
- Specification* 553
- StartSnapTarget* 554
- UseGraphicalAttributesOfIntervals* 554
- Visible* 554
- VisibleDataFieldIndex* 555
- VisibleMapName* 555
- VcCalendarGridCollection 558**
 - _NewEnum* 558
 - Add* 559
 - AddBySpecification* 560
 - CalendarGridByIndex* 560
 - CalendarGridByName* 561
 - Copy* 561
 - Count* 559
 - FirstCalendarGrid* 562
 - NextCalendarGrid* 562
 - Remove* 563
 - Update* 563
- VcCalendarProfile 565**
 - IntervalCollection* 565
 - Name* 566
 - PutInOrderAfter* 567
 - Specification* 566
 - Type* 566
- VcCalendarProfileCollection 568**
 - _NewEnum* 569
 - Add* 569
 - AddBySpecification* 570
 - CalendarProfileByIndex* 570
 - CalendarProfileByName* 571
 - Copy* 571
 - Count* 569
 - FirstCalendarProfile* 572
 - NextCalendarProfile* 572

- Remove* 572
- SelectCalendarProfiles* 573
- Update* 573
- VcCurve 574**
 - Addend* 575
 - Clear* 597
 - CurveSource* 576
 - CurveType* 576
 - DeletePoint* 597
 - DeletePointAsVariant* 598
 - Fill2Color* 577
 - Fill2Pattern* 578
 - Fill2ReferenceName* 581
 - FillColor* 582
 - FillPattern* 582
 - FillReferenceName* 585
 - FilterName* 586
 - GetFirstOverload* 598
 - GetFirstOverloadAsVariant* 599
 - GetFirstOverloadEx* 599
 - GetNextOverload* 600
 - GetNextOverloadAsVariant* 601
 - GetNextOverloadEx* 601
 - GetValues* 602
 - GetValuesAsVariant* 603
 - GetValuesEx* 603
 - Histogram* 587
 - LayerName* 587
 - LineColor* 587
 - LineThickness* 588
 - LineType* 589
 - MarkCurve* 590
 - Name* 591
 - OverloadResultsCalendarName* 591
 - Pattern2Color* 592
 - PatternColor* 592
 - PointsEquidistant* 593
 - SetValues* 604
 - Specification* 593
 - StackReferenceName* 594
 - TimeUnit* 594
 - UnitsPerStep* 595
 - UpdateBehaviorName* 595
 - ValencyDataFieldIndex* 596
 - Visible* 596
- VcCurveCollection 606**
 - _NewEnum* 607
 - Add* 608
 - AddBySpecification* 608
 - Copy* 609
 - Count* 607
 - CurveByIndex* 609
 - CurveByName* 610
 - FirstCurve* 610
 - NextCurve* 611
 - Remove* 611
- VcDataDefinition 613**
 - DefinitionTable* 613
- VcDataDefinitionTable 614, 619**
 - _NewEnum* 614, 619
 - Count* 615, 620
 - CreateDataField* 615, 620
 - FieldByIndex* 616, 621
 - FieldByName* 617, 621
 - FirstField* 617, 622
 - NextField* 618, 622
- VcDataRecord 624**
 - AllData* 624
 - DataField* 625
 - DataTableName* 626
 - DeleteDataRecord* 627
 - ID* 626
 - IdentifyObject* 627
 - RelatedDataRecord* 628

- UpdateDataRecord* 629
- VcDataRecordCollection 630**
 - _NewEnum* 630
 - Add* 631
 - Count* 631
 - DataRecordByID* 632
 - FirstDataRecord* 633
 - GetNewUniqueID* 633
 - NextDataRecord* 634
 - Remove* 634
 - Update* 635
- VcDataTable 636**
 - DataRecordCollection* 636
 - DataTableFieldCollection* 637
 - Description* 637
 - MultiplePrimaryKeysAllowed* 638
 - Name* 638
- VcDataTableCollection 639**
 - _NewEnum* 639
 - Add* 640
 - Copy* 641
 - Count* 640
 - DataTableByIndex* 642
 - DataTableByName* 642
 - FirstDataTable* 643
 - NextDataTable* 643
 - Update* 644
- VcDataTableField 645**
 - DataTableName* 646
 - DateFormat* 646
 - Editable* 647
 - Hidden* 647
 - Index* 648
 - Name* 648
 - PrimaryKey* 649
 - RelationshipFieldIndex* 649
 - Type* 650
- VcDataTableFieldCollection 652**
 - _NewEnum* 652
 - Add* 653
 - Copy* 654
 - Count* 653
 - DataTableFieldByIndex* 655
 - DataTableFieldByName* 655
 - FirstDataTableField* 656
 - NextDataTableField* 656
- VcDateLine 657**
 - AlwaysCurrentDate* 658
 - Date* 658
 - DateDataFieldIndex* 659
 - Font* 659
 - FontColor* 659
 - Identifiable* 660
 - LabelPosition* 660
 - LineColor* 660
 - LineThickness* 661
 - LineType* 662
 - Moveable* 663
 - Name* 663
 - Priority* 664
 - PutInOrderAfter* 668
 - SnapTarget* 664
 - Specification* 665
 - Text* 665
 - TurningAnnotationEnabled* 666
 - UpdateBehaviorName* 666
 - Visible* 666
 - VisibleDataFieldIndex* 667
 - VisibleMapName* 667
- VcDateLineCollection 669**
 - _NewEnum* 669
 - Add* 670
 - AddBySpecification* 671
 - Copy* 671

- Count* 670
- DateLineByIndex* 672
- DateLineByName* 673
- FirstDateLine* 673
- NextDateLine* 674
- Remove* 674
- Update* 675
- VcDateLineGrid 676**
 - AdjustToReferenceDate* 677
 - AnnotationAtBottom* 677
 - AnnotationAtCenter* 678
 - AnnotationAtTop* 678
 - FormatName* 678
 - HorAlignment* 679
 - LineColor* 679
 - LineColorDataFieldIndex* 680
 - LineColorMapName* 680
 - LineThickness* 680
 - LineType* 681
 - Name* 682
 - ObserveDST* 683
 - Period* 683
 - Priority* 684
 - ReferenceDate* 684
 - SnapTarget* 685
 - TurningAnnotationEnabled* 685
 - Unit* 686
 - UseReferenceDate* 686
 - Visible* 687
 - VisibleDataFieldIndex* 687
 - VisibleMapName* 687
- VcDateLineGridCollection 689**
 - _NewEnum* 689
 - Add* 690
 - AddBySpecification* 691
 - Copy* 691
 - Count* 690
 - DateLineGridByIndex* 692
 - DateLineGridByName* 692
 - FirstDateLineGrid* 693
 - NextDateLineGrid* 693
 - Remove* 694
 - Update* 694
- VcDefinitionField 696**
 - DateFormat* 696
 - Editable* 697
 - Hidden* 698
 - ID* 698
 - Name* 698
 - Type* 699
- VcField 700**
 - DataFieldID* 700
- VcFilter 701**
 - _NewEnum* 702
 - AddSubCondition* 705
 - CopySubCondition* 706
 - DataDefinitionTable* 702
 - DatesWithHourAndMinute* 703
 - Evaluate* 706
 - IsValid* 707
 - Name* 703
 - RemoveSubCondition* 707
 - Specification* 703
 - StringsCaseSensitive* 704
 - SubCondition* 704
 - SubConditionCount* 705
- VcFilterCollection 708**
 - _NewEnum* 708
 - Add* 710
 - AddBySpecification* 710
 - Copy* 711
 - Count* 709
 - FilterByIndex* 711
 - FilterByName* 712

- FirstFilter* 712
- MarkedNodesFilter* 709
- NextFilter* 713
- Remove* 713
- VcFilterSubCondition** 714
 - ComparisonValueAsString* 714
 - ConnectionOperator* 715
 - DataFieldIndex* 716
 - FilterName* 716
 - Index* 717
 - IsValid* 718
 - Operator* 717
- VcGantt** 719
 - AboutBox* 809
 - ActiveNodeFilter* 728
 - AllowMultipleBoxMarking* 728
 - AllowNewBoxes* 729
 - AllowNewNodes* 729
 - AllowNumericScaleRescale* 730
 - AllowPanningMode* 730
 - AllowSelectionViaRubberRect* 731
 - AllowTableColumnWidthOptimization* 731
 - AllowTimescaleRescale* 732
 - AllowVerticalNodeMovement* 732
 - AllowVerticalNodeMovementViaTable* 732
 - Arrangement* 733
 - ArrowKeyMode* 733
 - ArrowKeyStepSizeMultiplier* 735
 - AssignCalendarToNodes* 735
 - BarSeparationGroupBy* 736
 - BorderArea* 737
 - BoxCollection* 737
 - BoxFormatCollection* 737
 - CalendarCollection* 738
 - CalendarGridCollection* 738
 - CalendarProfileCollection* 738
 - Clear* 809
 - ClearAll* 810
 - ConfigurationName* 739
 - ConsiderLinkRelationTypesOnNodeDragging* 740
 - ContextMenuForBoxesEnabled* 740
 - ConvertDistance* 810
 - CtrlCXVProcessing* 741
 - CurrentVersion* 741
 - DataDefinition* 742
 - DataTableCollection* 742
 - DateLineCollection* 742
 - DateLineGridCollection* 743
 - DateOutputFormat* 743
 - DeleteLinkRecord* 811
 - DeleteNodeRecord* 811
 - DetectDataTableFieldName* 811
 - DetectDataTableName* 812
 - DetectFieldIndex* 812
 - DiagramAlternatingRowBackColor* 745
 - DiagramBackColor* 745
 - DiagramHistogramHeightRatio* 746
 - DiagramHistogramHeightRatioEx* 746
 - DiagramVisible* 747
 - DialogFont* 747
 - DirectDataWritingModeEnabled* 747
 - DoubleOutputFormat* 748
 - DumpConfiguration* 813
 - EditGroup* 814
 - EditLink* 814
 - EditNewNode* 749
 - EditNode* 814
 - Enabled* 749
 - EnableSupplyTextEntryEvent* 749
 - EndLoading* 815

- Error* 852
- ErrorAsVariant* 853
- EventReturnStatus* 750
- EventsSecurityCheck* 751
- EventText* 751
- ExportGraphicsToFile* 815
- ExtendedDataTables* 751
- ExtendedEditingBehavior* 752
- FilePath* 752
- FilterCollection* 753
- FitChartIntoView* 818
- FitHistogramsIntoView* 818
- FitRangeIntoView* 819
- FontAntiAliasingEnabled* 754
- GetAValueFromARGB* 819
- GetBValueFromARGB* 820
- GetCurrentComponentStart* 821
- GetCurrentViewDates* 821
- GetCurrentViewDatesAsString* 822
- GetCurrentViewDatesAsVariant* 823
- GetDate* 823
- GetDateAsString* 823
- GetGValueFromARGB* 824
- GetLinkByID* 824
- GetLinkByIDs* 825
- GetNodeByID* 825
- GetRValueFromARGB* 826
- GetViewComponentSize* 827
- GetViewComponentSizeAsVariant* 828
- GroupCollection* 754
- GroupingField* 755
- GroupingModificationsAllowed* 755
- GroupingOrderField* 756
- GroupingSortOrder* 756
- GroupLevelLayoutCollection* 757
- GroupNodes* 828
- GroupOptimizationOnInteractionsEnabled* 757
- HierarchyDataFieldIndex* 758
- HierarchyLevelLayout* 759
- HistogramCollection* 759
- HistogramSeparationLineColor* 759
- HistogramSetMaxYValue* 829
- hWnd* 760
- IdentifyField* 829
- IdentifyLayerAt* 830
- IdentifyLayerAtAsVariant* 831
- IdentifyObject* 831
- IdentifyObjectAt* 833
- IdentifyObjectAtAsVariant* 834
- InfoWindow* 760
- InInteractionEventsEnabled* 760
- InPlaceEditingOnGroupsInDiagramEnabled* 761
- InPlaceEditingOnGroupsInTableEnabled* 761
- InPlaceEditingOnNodesInDiagramEnabled* 762
- InPlaceEditingOnNodesInTableEnabled* 763
- InsertLinkRecord* 834
- InsertNodeRecord* 835
- InteractionMode* 764
- KeyDown* 853
- KeyPress* 854
- KeyUp* 854
- LayerCollection* 764
- LegendView* 764
- LineFormatCollection* 765
- LinkAppearanceCollection* 765
- LinkCollection* 765
- LinkPredecessorDataFieldIndex* 766
- LinksDataTableName* 767
- LinkSuccessorDataFieldIndex* 767

- LinkTypeDataFieldIndex* 768
- MakeARGB* 836
- MapCollection* 769
- MinimumRowHeight* 769
- MouseProcessingEnabled* 770
- MoveAllSelectedNodes* 770
- MoveLayersAsNodeWithShiftKey* 771
- MoveNodeAlways* 771
- MoveNodeWhenMarked* 772
- NewNodesViaDoubleClick* 772
- NodeCalendarNameDataFieldIndex* 773
- NodeCollection* 773
- NodeDurationDataFieldIndex* 773
- NodeEndDateDataFieldIndex* 774
- NodeLevelLayout* 774
- NodeRowNumberDataFieldIndex* 774
- NodesDataTableName* 775
- NodeStartDateDataFieldIndex* 775
- NodeTooltipTextField* 776
- NoOfInitialRows* 776
- OLECompleteDrag* 855
- OLEDragDrop* 856
- OLEDragHorizontalMovementAllowed* 777
- OLEDragMode* 777
- OLEDragOver* 857
- OLEDragViaDiagram* 778
- OLEDragViaTable* 778
- OLEDragWithOwnMouseCursor* 779
- OLEDragWithPhantom* 779
- OLEDropMode* 780
- OLEGiveFeedback* 858
- OLESetData* 858
- OLEStartDrag* 859
- OnBoxCreate* 860
- OnBoxCreateComplete* 861
- OnBoxLClick* 861
- OnBoxLDbIClick* 861
- OnBoxModify* 862
- OnBoxModifyCompleteEx* 863
- OnBoxRClick* 863
- OnCalendarGridRClick* 864
- OnCurveLClick* 865
- OnCurveLDbIClick* 866
- OnCurveModifyComplete* 866
- OnCurveModifyEx* 866
- OnCurveModifyEx2* 867
- OnCurveModifyExAsString* 868
- OnCurveRClick* 869
- OnDataRecordCreate* 870
- OnDataRecordCreateComplete* 871
- OnDataRecordDelete* 872
- OnDataRecordDeleteComplete* 873
- OnDataRecordModify* 873
- OnDataRecordModifyComplete* 874
- OnDataRecordNotFound* 875
- OnDateLineModify* 875
- OnDateLineRClick* 875
- OnDeleteCurvePoint* 876
- OnDeleteCurvePointEx* 877
- OnDiagramLClick* 878
- OnDiagramLDbIClick* 878
- OnDiagramRClick* 879
- OnGroupDelete* 880
- OnGroupLClick* 880
- OnGroupLDbIClick* 881
- OnGroupModify* 881
- OnGroupModifyComplete* 882
- OnGroupModifyEx* 883
- OnGroupRClick* 883
- OnGroupsMark* 884
- OnGroupsMarkComplete* 885
- OnHelpRequested* 885

- OnHistogramLClick* 886
- OnHistogramLDbIClick* 886
- OnHistogramRClick* 887
- OnHistogramsHeight* 888
- OnHistogramsHeightChanged* 889
- OnHistogramsHeightModifyEx* 889
- OnInsertCurvePoint* 890
- OnInsertCurvePointEx* 890
- OnInteractionEndComplete* 891
- OnInteractionModeChange* 892
- OnInteractionModeChangeComplete* 893
- OnInteractionObjectChangingComplete* 893
- OnInteractionStartComplete* 895
- OnLegendViewClosed* 896
- OnLinkCreate* 896
- OnLinkCreateComplete* 897
- OnLinkDelete* 898
- OnLinkDeleteComplete* 898
- OnLinkLClickCltn* 899
- OnLinkLDbIClickCltn* 900
- OnLinkRClickCltn* 900
- OnModifyComplete* 901
- OnMouseDownClk* 902
- OnMouseDown* 902
- OnMouseMove* 903
- OnMouseUp* 904
- OnNodeCreate* 905
- OnNodeCreateCompleteEx* 905
- OnNodeDelete* 906
- OnNodeDeleteCompleteEx* 907
- OnNodeLClick* 907
- OnNodeLDbIClick* 908
- OnNodeModifyComplete* 909
- OnNodeModifyCompleteEx* 909
- OnNodeModifyEx* 910
- OnNodeRClick* 911
- OnNodeResizeStart* 912
- OnNodesMarkComplete* 913
- OnNodesMarkEx* 913
- OnNumericScaleLClick* 914
- OnNumericScaleLDbIClick* 915
- OnNumericScaleRClick* 915
- OnNumericScaleRescale* 916
- OnObjectDrawCompleteEx* 917
- OnObjectDrawEx* 918
- OnOptimizeTableColumnWidth* 920
- OnPreScrollComponent* 921
- OnPreScrollDiagramHor* 923
- OnResourceSchedulingProgress* 924
- OnResourceSchedulingWarning* 925
- OnScrollComponent* 928
- OnScrollDiagramHor* 930
- OnSelectField* 931
- OnShowCurveNameInMenu* 932
- OnShowDate* 933
- OnShowInPlaceEditor* 933
- OnStatusLineText* 935
- OnSupplyTextEntry* 935
- OnSupplyTextEntryAsVariant* 950
- OnTableCaptionLClick* 950
- OnTableCaptionLDbIClick* 951
- OnTableCaptionRClick* 951
- OnTableColumnWidth* 952
- OnTableColumnWidthModifyComplete* 953
- OnTableWidth* 953
- OnTableWidthModifyEx* 954
- OnTimeScaleChangeComplete* 955
- OnTimeScaleEndModifyComplete* 955
- OnTimeScaleLClick* 955
- OnTimeScaleLDbIClick* 956

- OnTimeScaleRClick* 956
- OnTimeScaleSectionRescaleCompleteEx* 957
- OnTimeScaleSectionRescaleEx* 958
- OnTimeScaleSectionStartModify* 958
- OnTimeScaleStartModifyComplete* 959
- OnToolTipText* 959
- OnToolTipTextAsVariant* 961
- OnViewComponentsSizeModifyComplete* 961
- OnWorldViewClosed* 962
- OnZoomFactorModifyComplete* 962
- Open* 837
- OptimizeTimeScaleStartEnd* 838
- OverlapLayerEnabled* 781
- OverlapLayerName* 781
- PageLayout* 838
- PartialLoadThreshold* 782
- PhantomLayerHeight* 783
- PrintDirectEx* 839
- Printer* 784
- PrinterSetup* 839
- PrintIt* 840
- PrintPreview* 840
- PrintToFile* 840
- RecalculateAllStructureCodes* 841
- Reset* 841
- ResourceScheduler2* 784
- RightTable* 784
- RightTableDiagramWidthRatio* 785
- RightTableDiagramWidthRatioEx* 785
- RoundedLinkSlantsEnabled* 786
- RowHeightReductionEnabled* 786
- RowMargins* 787
- Sash3DStyleEnabled* 787
- SashThickness* 787
- SaveAsEx* 842
- Schedule* 843
- Scheduler* 788
- ScrollComponentStartTo* 843
- ScrollEventsEnabled* 788
- ScrollToDate* 844
- ScrollToGroupLine* 845
- ScrollToNode* 845
- ScrollToNodeLine* 846
- SelectedRowBackColorAsARGB* 788
- ShowExportGraphicsDialog* 847
- ShowNonWorkInterval* 789
- ShowSnapLines* 789
- ShowSnapMarkings* 790
- ShowTimeScaleDialog* 790
- ShowToolTip* 791
- SnapTargetNodesSelectionMode* 791
- SortField* 792
- SortGroups* 848
- SortNodes* 848
- SortOrder* 792
- SubRowMargins* 793
- SummaryBarsVisible* 793
- SuspendUpdate* 849
- Table* 794
- TableCollection* 794
- TableDiagramWidthRatio* 794
- TableDiagramWidthRatioEx* 795
- TimeScaleCollection* 795
- TimeScaleEnd* 796
- TimeScaleStart* 796
- TimeUnit* 797
- TimeUnitsPerStep* 797
- ToolTipChangeDuration* 798
- ToolTipDuration* 798
- ToolTipPointerDuration* 799
- ToolTipShowAfterClick* 799

- TrackingSpaceBackColorAsARGB* 800
- TrackingSpacePattern* 800
- TrackingSpacePatternColorAsARGB* 803
- UpdateBehaviorCollection* 804
- UpdateLinkRecord* 850
- UpdateNodeRecord* 850
- UpdateRowNumberFields* 851
- UseHigherDiagramHistogramHeightRatioPrecision* 804
- UseHigherTableDiagramWidthRatioPrecision* 805
- UseSnapTargetsInInteractions* 805
- UseTwinLineSashPhantom* 806
- ViewComponentsBackColor* 806
- ViewComponentsBorderColor* 807
- WaitCursorEnabled* 807
- WorldView* 807
- Zoom* 851
- ZoomFactor* 808
- ZoomingPerMouseWheelAllowed* 808
- VcGroup 964**
 - BodyCollapsed* 965
 - DataField* 965
 - DataRecord* 971
 - DeleteGroup* 972
 - GroupingLevel* 966
 - GroupInvisible* 967
 - ID* 967
 - MarkGroup* 967
 - Name* 968
 - NodeCollection* 968
 - NodesInHeader* 969
 - NodesOverlaid* 969
 - RelatedDataRecord* 972
 - ReOptimizeNodes* 973
 - RowsBelowCollapsed* 969
 - SubGroups* 970
 - SuperGroup* 970
 - UpdateGroup* 973
 - Visible* 971
- VcGroupCollection 974**
 - _NewEnum* 974
 - Count* 975
 - FirstGroup* 975
 - GroupByName* 976
 - NextGroup* 976
 - SelectGroups* 977
- VcGroupLevelLayout 978**
 - AllowVerticalGroupMovementViaDiagram* 980
 - AllowVerticalGroupMovementViaTable* 980
 - AutoCollapseGroups* 980
 - AutoExpandTargetGroup* 981
 - BodiesCollapsed* 981
 - BodiesCollapsedDataFieldIndex* 982
 - BodiesCollapsedMapName* 982
 - CalendarGridName* 982
 - CalendarGridsWithChildGroups* 983
 - CalendarNameDataFieldIndex* 983
 - DateLineGridName* 983
 - DateLineGridsWithChildGroups* 984
 - DateLineName* 984
 - DateLinesWithChildGroups* 984
 - GroupDataFieldIndex* 985
 - GroupsInvisible* 985
 - GroupsInvisibleCollapsedMapName* 985
 - GroupsInvisibleDataFieldIndex* 985
 - Level* 986
 - ModificationsAllowed* 986
 - Name* 987
 - NodesInHeaders* 987

NodesInHeadersDataFieldIndex 987
NodesInHeadersMapName 988
NodesOverlaid 988
OptimizedNodesSortDataFieldIndex 988
OptimizedNodesSortOrder 989
OverlaidNodesSortDataFieldIndex 989
OverlaidNodesSortOrder 990
PagebreakMode 990
RestoreAutoCollapsedGroups 990
RestoreAutoExpandedGroups 991
RowBackColorAsARGB 991
RowBackColorDataFieldIndex 992
RowBackColorMapName 992
RowPattern 992
RowPatternColorAsARGB 996
RowPatternColorDataFieldIndex 996
RowPatternColorMapName 997
RowPatternDataFieldIndex 997
RowPatternMapName 997
SeparationLineColor 998
SeparationLineColorDataFieldIndex 998
SeparationLineColorMapName 998
SeparationLineThickness 999
SeparationLineType 1000
ShowCalendarGrids 1001
ShowDateLineGrids 1001
ShowDateLines 1002
ShowGroupNodes 1002
ShowSeparationLines 1002
ShowSeparationLinesAtTop 1003
SortDataFieldIndex 1003
SortOrder 1003
Specification 1004
SummaryBarsVisible 1004
Visible 1005

VcGroupLevelLayoutCollection 1006

_NewEnum 1006
Add 1007
AddBySpecification 1008
Copy 1008
Count 1007
FirstGroupLevelLayout 1009
GroupLevelLayoutByIndex 1009
GroupLevelLayoutByName 1010
NextGroupLevelLayout 1010
Remove 1011
Update 1011

VcHierarchyLevelLayout 1013

AutoCollapseGroups 1014
AutoExpandTargetGroup 1014
BodiesCollapsed 1014
BodiesCollapsedDataFieldIndex 1015
BodiesCollapsedMapName 1015
HierarchyDataFieldIndex 1015
LevelMaximumForPagebreaks 1016
NodeSeparationLinesVisible 1016
NodesInHeaders 1016
NodesInHeadersDataFieldIndex 1017
NodesInHeadersMapName 1017
NodesOverlaid 1018
PagebreakMode 1018
RestoreAutoCollapsedGroups 1018
RestoreAutoExpandedGroups 1019
SeparationLineColor 1019
SeparationLineThickness 1020
SeparationLineType 1020
ShowSeparationLines 1022
SummaryBarsVisible 1022

VcHistogram 1023

CalendarName 1024

- CurveCollection* 1024
- FitRangeIntoView* 1028
- GetActualScaleValues* 1029
- GetActualScaleValuesAsVariant* 1029
- GetCurrentYValues* 1029
- GetCurrentYValuesAsVariant* 1030
- Name* 1024
- NominalScaleMaximum* 1025
- NominalScaleMinimum* 1025
- NumericScaleCollection* 1026
- PutInOrderAfter* 1030
- RowBackColorAsARGB* 1026
- RowPattern* 1026
- RowPatternColorAsARGB* 1027
- ScrollToValue* 1030
- ShowCalendarGrids* 1027
- Visible* 1027
- VcHistogramCollection 1032**
 - _NewEnum* 1032
 - Active* 1033
 - Count* 1033
 - CreateHistogram* 1034
 - DeleteHistogram* 1034
 - FirstHistogram* 1035
 - HistogramByIndex* 1035
 - HistogramByName* 1035
 - NextHistogram* 1036
- VcInfoWindow 1037**
 - OutputFormatForCenterDate* 1037
 - OutputFormatForDuration* 1039
 - OutputFormatForEndDate* 1040
 - OutputFormatForStartDate* 1041
 - ReferenceDate* 1043
 - UseReferenceDate* 1043
 - Visible* 1043
- VcInterval 1045**
 - BackColorAsARGB* 1047
 - CalendarProfileName* 1047
 - DayInEndMonth* 1048
 - DayInStartMonth* 1048
 - Duration* 1048
 - EndTime* 1049
 - EndMonth* 1049
 - EndTime* 1049
 - EndWeekday* 1050
 - LineColor* 1050
 - LineThickness* 1050
 - LineType* 1051
 - Name* 1052
 - Pattern* 1053
 - PatternColorAsARGB* 1056
 - PutInOrderAfter* 1060
 - Specification* 1056
 - StartDateTime* 1057
 - StartMonth* 1057
 - StartTime* 1058
 - StartWeekday* 1058
 - Text* 1058
 - TimeUnit* 1059
 - Type* 1059
 - UseGraphicalAttributes* 1059
- VcIntervalCollection 1062**
 - _NewEnum* 1063
 - Add* 1063
 - AddBySpecification* 1064
 - Copy* 1064
 - Count* 1063
 - FirstInterval* 1065
 - IntervalByIndex* 1065
 - IntervalByName* 1065
 - NextInterval* 1066
 - Remove* 1066
 - Update* 1067

VcLayer 1068

BackColorAsARGB 1070
BackColorDataFieldIndex 1070
BackColorMapName 1071
CalculateCurrentWidth 1107
CompletionDataFieldIndex 1072
DurationDataFieldIndex 1073
EndDataFieldIndex 1073
EndSnapTarget 1074
FilterName 1074
GraphicsFileName 1075
GraphicsFileNameDataFieldIndex 1076
GraphicsFileNameMapName 1077
Height 1078
HeightDataFieldIndex 1078
HeightMapName 1079
HorizontalOffset 1079
LabelSizeDependence 1080
LayerFormat 1080
LayerShape 1080
LegendText 1083
LineColor 1083
LineColorDataFieldIndex 1084
LineColorMapName 1084
LineThickness 1084
LineType 1085
MaximumEndDataFieldIndex 1086
MinimumStartDataFieldIndex 1087
Moveable 1087
Name 1087
NonWorkInterval 1088
NonWorkIntervalBackColorAsARGB 1088
NonWorkIntervalBackColorDataFieldIndex 1089
NonWorkIntervalBackColorMapName 1090

NonWorkIntervalLineColor 1091
NonWorkIntervalLineColorDataFieldIndex 1091
NonWorkIntervalLineColorMapName 1091
NonWorkIntervalLineThickness 1092
NonWorkIntervalLineType 1093
NonWorkIntervalPattern 1094
NonWorkIntervalPatternColorAsARGB 1097
NonWorkIntervalPatternColorDataFieldIndex 1098
NonWorkIntervalPatternDataFieldIndex 1098
NonWorkIntervalPatternMapName 1099
NonWorkIntervalShape 1100
ObjectDrawEventsEnabled 1100
PatternColorAsARGB 1101
PatternColorDataFieldIndex 1101
PatternColorMapName 1102
PutInOrderAfter 1107
Sizeable 1102
Specification 1103
StartDataFieldIndex 1103
StartSnapTarget 1103
ThreeDEffect 1104
UsedAsOverlapLayer 1104
VerticalOffset 1105
VerticalOffsetDataFieldIndex 1105
VerticalOffsetMapName 1105
Visible 1106
VisibleInLegend 1106

VcLayerCollection 1109

_NewEnum 1109
Add 1110
AddBySpecification 1111
Copy 1111
Count 1110

- FirstLayer* 1112
- LayerByIndex* 1112
- LayerByName* 1113
- NextLayer* 1113
- Remove* 1114
- Update* 1114
- VcLayerFormat 1115**
 - _NewEnum* 1115
 - CopyFormatField* 1117
 - FormatField* 1116
 - FormatFieldCount* 1116
 - RemoveFormatField* 1117
- VcLayerFormatField 1118**
 - Alignment* 1119
 - BottomMargin* 1119, 1120
 - CalculateLineCount* 1129
 - ConstantText* 1120
 - FormatName* 1120
 - Index* 1121
 - LeftMargin* 1121, 1122
 - MinimumWidth* 1122
 - Priority* 1122
 - RightMargin* 1123
 - SuppressTruncatedText* 1124
 - TextDataFieldIndex* 1124
 - TextFont* 1125
 - TextFontColor* 1125
 - TextFontColorDataFieldIndex* 1125
 - TextFontColorMapName* 1126
 - TextFontDataFieldIndex* 1126
 - TextFontMapName* 1126
 - TextLineCount* 1127
 - TextLineCountDataFieldIndex* 1127
 - TextLineCountMapName* 1128
 - TopMargin* 1128, 1129
- VcLegendView 1131**
 - Border* 1131
 - BorderColor* 1132
 - Height* 1132
 - HeightActualValue* 1133
 - Left* 1133
 - LeftActualValue* 1134
 - ParentHWnd* 1134
 - ScrollBarMode* 1135
 - Top* 1135
 - TopActualValue* 1136
 - Update* 1139
 - Visible* 1136
 - Width* 1137
 - WidthActualValue* 1137
 - WindowMode* 1138
- VcLineFormat 1140**
 - _NewEnum* 1140
 - CopyFormatField* 1143
 - FormatField* 1141
 - FormatFieldCount* 1141
 - Name* 1142
 - RemoveFormatField* 1143
 - Specification* 1142
- VcLineFormatCollection 1144**
 - _NewEnum* 1144
 - Add* 1145
 - AddBySpecification* 1146
 - Copy* 1146
 - Count* 1145
 - FirstFormat* 1147
 - FormatByIndex* 1147
 - FormatByName* 1148
 - NextFormat* 1148
 - Remove* 1149
- VcLineFormatField 1150**
 - Alignment* 1151
 - ConstantText* 1151
 - DateOutputFormat* 1151

- FormatName* 1153
- Index* 1153
- PatternBackgroundColorAsARGB* 1154
- PatternBackgroundColorDataFieldIndex* 1154
- PatternBackgroundColorMapName* 1154
- PatternColorAsARGB* 1155
- PatternColorDataFieldIndex* 1155
- PatternColorMapName* 1156
- PatternEx* 1156
- PatternExDataFieldIndex* 1160
- PatternExMapName* 1160
- TextDataFieldIndex* 1160
- TextFont* 1161
- TextFontColor* 1161
- TextFontColorDataFieldIndex* 1161
- TextFontColorMapName* 1162
- TextFontDataFieldIndex* 1162
- TextFontMapName* 1163
- TextLineCount* 1163
- VcLink 1164**
 - AllData* 1164
 - DataField* 1165
 - DataRecord* 1167
 - DeleteLink* 1167
 - ID* 1165
 - PredecessorNode* 1166
 - RelatedDataRecord* 1167
 - SuccessorNode* 1166
 - UpdateLink* 1168
- VcLinkAppearance 1169**
 - FilterName* 1169
 - LineColor* 1170
 - LineThickness* 1170
 - LineType* 1171
 - Name* 1172
 - PredecessorLayerName* 1173
 - PrePortSymbol* 1173
 - PutInOrderAfter* 1178
 - RoutingType* 1174
 - Specification* 1175
 - SuccessorLayerName* 1176
 - SuccPortSymbol* 1176
 - Visible* 1177
- VcLinkAppearanceCollection 1179**
 - _NewEnum* 1179
 - Add* 1180
 - AddBySpecification* 1181
 - Copy* 1181
 - Count* 1180
 - FirstLinkAppearance* 1182
 - LinkAppearanceByIndex* 1182
 - LinkAppearanceByName* 1183
 - NextLinkAppearance* 1183
 - Remove* 1184
 - Update* 1184
- VcLinkCollection 1185**
 - _NewEnum* 1185
 - Count* 1186
 - FirstLink* 1186
 - NextLink* 1187
 - SelectLinks* 1187
- VcMap 1189**
 - _NewEnum* 1189
 - ConsiderFilterEntries* 1190
 - Count* 1190
 - CreateEntry* 1192
 - DeleteEntry* 1193
 - FirstMapEntry* 1193
 - GetMapEntry* 1194
 - Name* 1191
 - NextMapEntry* 1194
 - Specification* 1191

- Type* 1191
- VcMapCollection 1195**
 - _NewEnum* 1196
 - Add* 1197
 - AddBySpecification* 1197
 - Copy* 1198
 - Count* 1196
 - FirstMap* 1198
 - MapByIndex* 1199
 - MapByName* 1199
 - NextMap* 1200
 - Remove* 1200
 - SelectMaps* 1201
 - Update* 1201
- VcMapEntry 1203**
 - ColorAsARGB* 1203
 - DataFieldValue* 1204
 - FontBody* 1205
 - FontName* 1205
 - FontSize* 1206
 - GraphicsFileName* 1206
 - Legend* 1207
 - Millimeter* 1208
 - Number* 1208
 - Pattern* 1209
- VcNode 1213**
 - AllData* 1214
 - DataField* 1214
 - DataRecord* 1219
 - DeleteNode* 1219
 - GetPositionInView* 1220
 - GetPositionInViewAsVariant* 1220
 - ID* 1215
 - IncomingLinks* 1215
 - MarkNode* 1216
 - MoveMode* 1216
 - NodeRowInView* 1221
 - OutgoingLinks* 1217
 - OutlineIndent* 1221
 - OutlineOutdent* 1222
 - RelatedDataRecord* 1222
 - SetPositionInView* 1223
 - SnapTargetMode* 1217
 - SuperGroup* 1218
 - UpdateBehaviorName* 1218
 - UpdateNode* 1224
- VcNodeCollection 1225**
 - _NewEnum* 1225
 - Count* 1226
 - FirstNode* 1226
 - NextNode* 1227
 - SelectNodes* 1227
- VcNodeLevelLayout 1229**
 - CalendarGridName* 1230
 - DateLineName* 1230
 - RowBackColorAsARGB* 1230
 - RowBackColorDataFieldIndex* 1231
 - RowBackColorMapName* 1231
 - RowPattern* 1231
 - RowPatternColorAsARGB* 1235
 - RowPatternColorDataFieldIndex* 1235
 - RowPatternColorMapName* 1236
 - RowPatternDataFieldIndex* 1236
 - RowPatternMapName* 1236
 - SeparationLineColor* 1237
 - SeparationLineInterval* 1237
 - SeparationLineThickness* 1237
 - SeparationLineType* 1238
 - ShowCalendarGrids* 1239
 - ShowDateLines* 1240
 - ShowSeparationLines* 1240
 - ShowSeparationLinesAtTop* 1240
 - SortDataFieldIndex* 1241

- SortOrder* 1241
- VcNumericScale 1242**
 - DoubleOutputFormat* 1243
 - Font* 1243
 - FontColor* 1244
 - Histogram* 1244
 - LineColor* 1245
 - MajorTicks* 1245
 - MajorTicksEx* 1245
 - MinorTicks* 1246
 - MinorTicksEx* 1247
 - Name* 1247
 - PatternBackgroundColorAsARGB* 1248
 - PatternColorAsARGB* 1248
 - PatternEx* 1248
 - ThreeDEffect* 1252
 - TickColor* 1252
 - Title* 1252
 - Unit* 1253
 - UnitEx* 1254
 - UnitLabel* 1254
 - UnitWidth* 1255
 - UpdateBehaviorName* 1255
- VcNumericScaleCollection 1256**
 - _NewEnum* 1256
 - Active* 1257
 - Count* 1257
 - FirstNumericScale* 1258
 - NextNumericScale* 1258
 - NumericScaleByIndex* 1259
 - NumericScaleByName* 1259
- VcPrinter 1261**
 - AbsoluteBottomMarginInCM* 1262
 - AbsoluteBottomMarginInInches* 1263
 - AbsoluteLeftMarginInCM* 1263
 - AbsoluteLeftMarginInInches* 1263
 - AbsoluteRightMarginInCM* 1264
 - AbsoluteRightMarginInInches* 1264
 - AbsoluteTopMarginInCM* 1265
 - AbsoluteTopMarginInInches* 1265
 - Alignment* 1265
 - AllBorderBoxesShownOnCombinedControls* 1266
 - CombiningControlsEnabled* 1267
 - CurrentHorizontalPagesCount* 1267
 - CurrentVerticalPagesCount* 1268
 - CurrentZoomFactor* 1268
 - CuttingMarks* 1268
 - DateFormat* 1269
 - DefaultPrinterName* 1270
 - DiagramEnabled* 1270
 - DocumentName* 1271
 - FoldingMarksType* 1271
 - MarginsShownInInches* 1274
 - MaxHorizontalPagesCount* 1274
 - MaxVerticalPagesCount* 1275
 - Orientation* 1275
 - PageDescription* 1276
 - PageDescriptionString* 1276
 - PageFrame* 1277
 - PageNumberMode* 1277
 - PageNumbers* 1278
 - PagePaddingEnabled* 1278
 - PaperSize* 1279
 - PrintDate* 1279
 - PrinterName* 1280
 - ReOptimizeNodesInGroupsEnabled* 1280
 - RepeatTableTimeScale* 1281
 - ScalingMode* 1281
 - StartUpSinglePage* 1282
 - TableColumnRanges* 1282
 - TableWidthAdoptionFromViewOnScreen* 1283

- TimeColumnEndDate* 1283
- TimeColumnStartDate* 1284
- TimeScaleAdjustment* 1284
- ZoomFactorAsDouble* 1285
- VcRect** 1286
 - Bottom* 1286
 - Height* 1286
 - Left* 1287
 - Right* 1288
 - Top* 1288
 - Width* 1288
- VcResourceScheduler2** 1289
 - AssignmentDataTableName* 1292
 - AssignmentIsResultFieldIndex* 1294
 - AssignmentIsVisibleFieldIndex* 1294
 - AssignmentLoadOrConsumptionPerItemFieldIndex* 1295
 - AssignmentMaximumLoadFieldIndex* 1295
 - AssignmentMinimumLoadFieldIndex* 1296
 - AssignmentMinimumMaximumLoadType* 1297
 - AssignmentOperationIDFieldIndex* 1297
 - AssignmentResourceIDFieldIndex* 1298
 - AssignmentResourceSelectionStrategyFieldIndex* 1299
 - BaseCalendarUsageForSupplementTimes* 1300
 - BaseTimeUnit* 1301
 - BaseTimeUnitsPerStep* 1301
 - DataRecordEventsEnabled* 1302
 - DefaultOperationMaximumInterruptionTime* 1302
 - DefaultResourceCalendarName* 1303
 - DetermineIDOfFirstOperationByTaskID* 1351
 - DetermineIDOfLastOperationByTaskID* 1351
 - FullUsageOfPlanningUnitsEnabled* 1303
 - LinkDataTableName* 1304
 - LinkDurationFieldIndex* 1306
 - LinkPredecessorOperationIDFieldIndex* 1306
 - LinkPredecessorTaskIDFieldIndex* 1307
 - LinkSuccessorOperationIDFieldIndex* 1307
 - LinkSuccessorTaskIDFieldIndex* 1308
 - OperationDataTableName* 1309
 - OperationLoadPerItemFieldIndex* 1310
 - OperationMaximumInterruptionTimeFieldIndex* 1310
 - OperationMinimumSupplementTimeFieldIndex* 1311
 - OperationOverlapQuantityFieldIndex* 1312
 - OperationPostLoadFieldIndex* 1314
 - OperationPostOffsetFieldIndex* 1314
 - OperationPreparationLoadFieldIndex* 1315
 - OperationPreparationOffsetFieldIndex* 1316
 - OperationResultEndDateFieldIndex* 1316
 - OperationResultPostEndDateFieldIndex* 1317
 - OperationResultPreparationStartDateFieldIndex* 1317
 - OperationResultProcessingTimeFieldIndex* 1318
 - OperationResultSelectedTimingResourceIDFieldIndex* 1318
 - OperationResultStartDateFieldIndex* 1319
 - OperationResultStatusFieldIndex* 1320

- OperationRouteFieldIndex* 1320
- OperationSequenceNumberFieldIndex* 1321
- OperationStartLockDateFieldIndex* 1322
- OperationTaskIDFieldIndex* 1322
- OperationWorkInProgressFieldIndex* 1323
- PlanningEndDate* 1324
- PlanningStartDate* 1324
- PlanningStrategy* 1325
- Process* 1352
- ResourceCalendarNameFieldIndex* 1326
- ResourceCapacityType* 1327
- ResourceCapacityTypeFieldIndex* 1327
- ResourceConstraintTypeFieldIndex* 1329
- ResourceDataTableName* 1329
- ResourceEfficiencyFieldIndex* 1331
- ResourceGroupDataTableName* 1332
- ResourceGroupIDFieldIndex* 1333
- ResourceNameFieldIndex* 1334
- ResourceResultLoadCurveNamePrefix* 1335
- ResourceResultStockCurveNamePrefix* 1336
- ResourceSelectionStrategy* 1336
- ResourceType* 1338
- ResultProcessingStepCount* 1339
- TaskDataTableName* 1339
- TaskDueDateFieldIndex* 1341
- TaskPlanningStrategyFieldIndex* 1341
- TaskPriorityFieldIndex* 1342
- TaskQuantityFieldIndex* 1343
- TaskReleaseDateFieldIndex* 1344
- TaskResultEndDateFieldIndex* 1345
- TaskResultPostEndDateFieldIndex* 1345
- TaskResultPreparationStartDateFieldIndex* 1346
- TaskResultProcessingStepFieldIndex* 1346
- TaskResultProcessingTimeFieldIndex* 1347
- TaskResultRouteFieldIndex* 1347
- TaskResultStartDateFieldIndex* 1348
- ToleranceTimeOnASAPDueDates* 1348
- ToleranceTimeOnJITReleaseDates* 1349
- ToleranceTimeOnStartLockDates* 1349
- WorkInProgressType* 1350
- WritingDebugFilesEnabled* 1350
- VcRibbon 1353**
 - CalendarName* 1354
 - DateOutputFormat* 1354
 - Font* 1356
 - FontColor* 1357
 - MajorTicks* 1357
 - MinorTicks* 1357
 - ObserveDST* 1358
 - PatternBackgroundColorAsARGB* 1358
 - PatternColorAsARGB* 1359
 - PatternEx* 1359
 - Position* 1363
 - ReferenceDate* 1363
 - TextAlignment* 1363
 - TickColor* 1364
 - TickPosition* 1364
 - Type* 1364
 - UnitSeparation* 1365
 - UseReferenceDate* 1366
- VcScheduler 1367**

- ActualEndDateDataFieldIndex* 1368
- ActualStartDateDataFieldIndex* 1368
- AutomaticSchedulingEnabled* 1368
- DurationDataFieldIndex* 1369
- EarlyEndDateDataFieldIndex* 1369
- EarlyStartDateDataFieldIndex* 1369
- EndDateForAutomaticScheduling* 1370
- EndDateNotLaterThanDataFieldIndex* 1370
- FreeFloatDataFieldIndex* 1370
- LateEndDateDataFieldIndex* 1370
- LateStartDateDataFieldIndex* 1371
- LinkDurationDataFieldIndex* 1371
- ScheduledProjectEndDate* 1371
- ScheduledProjectStartDate* 1372
- ScheduleProject* 1374
- ScheduleSuccessorsOnlyEnabled* 1372
- StartDateForAutomaticScheduling* 1373
- StartDateNotEarlierThanDataFieldIndex* 1373
- TotalFloatDataFieldIndex* 1373
- VcSection 1375**
 - CalendarGridEx* 1375
 - Collapse* 1376
 - DateLineGrid* 1377
 - LineColor* 1377, 1378
 - Ribbon* 1378
 - StartDate* 1379
 - Unit* 1379
 - UnitWidth* 1380
 - UnitWidthEx* 1380
- VcTable 1381**
 - ColumnTitle* 1381
 - ColumnWidth* 1382
 - IdentifyFormatField* 1385
 - Name* 1382
 - NoOfColumns* 1383
 - OptimizeColumnWidth* 1385
 - Position* 1383
 - TableFormatCollection* 1383
 - UpdateBehaviorName* 1384
 - Visible* 1384
- VcTableCollection 1386**
 - _NewEnum* 1386
 - Active* 1387
 - Count* 1387
 - FirstTable* 1388
 - NextTable* 1388
 - TableByIndex* 1389
 - TableByName* 1389
- VcTableFormat 1390**
 - _NewEnum* 1391
 - CollapseColumn* 1391
 - FieldsSeparatedByLines* 1392
 - FilterName* 1392
 - FormatField* 1393
 - FormatFieldCount* 1393
 - IndentColumn* 1394
 - IndentWidth* 1394
 - Name* 1394
 - SeparationLineColor* 1395
 - ThreeDEffect* 1395
- VcTableFormatCollection 1396**
 - _NewEnum* 1396
 - Count* 1397
 - FirstFormat* 1397
 - FormatByIndex* 1398
 - FormatByName* 1398
 - NextFormat* 1399
- VcTableFormatField 1400**
 - Alignment* 1401
 - BottomMargin* 1401

- CombiField* 1402
- ConstantText* 1402
- FormatName* 1402
- GraphicsFileName* 1403
- GraphicsFileNameDataFieldIndex* 1404
- GraphicsFileNameMapName* 1404
- GraphicsHeight* 1405
- Index* 1405
- LeftMargin* 1405
- MaximumTextLineCount* 1406
- MinimumTextLineCount* 1406
- MultiState* 1407
- PatternBackgroundColorAsARGB* 1407
- PatternBackgroundColorDataFieldIndex* 1408
- PatternBackgroundColorMapName* 1408
- PatternColorAsARGB* 1408
- PatternColorDataFieldIndex* 1409
- PatternColorMapName* 1409
- PatternEx* 1410
- PatternExDataFieldIndex* 1413
- PatternExMapName* 1413
- RightMargin* 1414
- TextDataFieldIndex* 1414
- TextFont* 1415
- TextFontColor* 1415
- TextFontColorDataFieldIndex* 1415
- TextFontColorMapName* 1416
- TextFontDataFieldIndex* 1416
- TextFontMapName* 1416
- TopMargin* 1417
- Type* 1417
- VcTimeScale 1418**
 - BackgroundColor* 1418
 - Font* 1419
 - FontColor* 1419
 - Name* 1419
 - Ribbon* 1420
 - Section* 1420
 - ShowCalendarGrids* 1421
 - ShowDateGrids* 1421
 - ThreeDEffect* 1422
 - UpdateBehaviorName* 1422
- VcTimeScaleCollection 1424**
 - _NewEnum* 1424
 - Active* 1425
 - Count* 1425
 - FirstTimeScale* 1426
 - NextTimeScale* 1426
 - TimeScaleByIndex* 1427
 - TimeScaleByName* 1427
- VcUpdateBehavior 1428**
 - Context* 1430
 - IsEditable* 1428
 - Name* 1429
 - PutInOrderAfter* 1430
 - Specification* 1429
- VcUpdateBehaviorCollection 1431**
 - _NewEnum* 1431
 - Active* 1432
 - Add* 1433
 - AddBySpecification* 1433
 - Copy* 1434
 - Count* 1432
 - FirstUpdateBehavior* 1434
 - NextUpdateBehavior* 1435
 - Remove* 1435
 - UpdateBehaviorByIndex* 1436
 - UpdateBehaviorByName* 1436
- VcUpdateBehaviorContext 1438**
 - DelayTime* 1438
 - IsEditable* 1439

- Type* 1439
- UpdateMode* 1440
- VcWorldView 1441**
 - Border* 1441
 - BorderColor* 1442
 - Height* 1442
 - HeightActualValue* 1443
 - Left* 1443
 - LeftActualValue* 1444
 - MarkingColor* 1444
 - Mode* 1445
 - ParentHWnd* 1446
 - ScrollBarMode* 1446
 - Top* 1447
 - TopActualValue* 1447
 - UpdateBehaviorName* 1448
 - Visible* 1448
 - Width* 1448
 - WidthActualValue* 1449
- Verbindung**
 - bearbeiten 400
 - Datensatz 1167
 - ID 1165
 - Nachfolgerknoten 767
 - Vorgängerknoten 766
 - zugeordneter Datensatz 1167
- Verbindungen 182, 252**
 - abgerundete Schrägen 227, 228
 - aktualisieren 1168
 - Anzahl 1186
 - anzeigen 252, 323
 - bearbeiten 814
 - Daten 1164
 - Daten aktualisieren 850
 - Datenfeld 1165
 - erzeugen 896, 897
 - laden 835
 - löschen 811, 898, 1167
 - Nachfolgerknoten 253, 1166
 - Typ 253
 - Verbindungsausssehen verwalten 323
 - Vorgängerknoten 252, 1166
 - ziehen 427
- Verbindungsausssehen 186**
 - Filter 1169
 - Layer Nachfolger 1176
 - Layer Vorgänger 1173
 - Linienfarbe 1170
 - Linienstärke 1171
 - Linientyp 1171
 - Name 1172
 - Portsymbol für Nachfolgerknoten 1176
 - Portsymbol für Vorgängerknoten 1173
 - Reihenfolge 1178
 - sichtbar 1177
 - Typ der Verbindungsroute 1174
- Verbindungsausssehen-Auflistung**
 - hinzufügen 1180
 - hinzufügen über Spezifikation 1181
 - Kopieren 1181
 - löschen 1184
- Verhältnis Tabelle/Diagramm**
 - genauere Methoden 804, 805
- Verschiebehilfen 213**
- Verschiebe-Modus 426**
- Versionsnummer**
 - anzeigen 741
- VerticalOffset**
 - Eigenschaft von*
 - VcLayer* 1105
- VerticalOffsetDataFieldIndex**
 - Eigenschaft von*

VcLayer 1105

VerticalOffsetMapName

Eigenschaft von

VcLayer 1105

Verzeichnispfad 753

ViewComponentsBackColor

Eigenschaft von

VcGantt 806

ViewComponentsBorderColor

Eigenschaft von

VcGantt 807

Visible

Eigenschaft von

VcBox 490

VcCalendarGrid 554

VcCurve 596

VcDateLine 666

VcDateLineGrid 687

VcGroup 971

VcGroupLevelLayout 1005

VcHistogram 1027

VcInfoWindow 1043

VcLayer 1106

VcLegendView 1136

VcLinkAppearance 1177

VcTable 1384

VcWorldView 1448

VisibleDataFieldIndex

Eigenschaft von

VcCalendarGrid 555

VcDateLine 667

VcDateLineGrid 687

VisibleInLegend

Eigenschaft von

VcLayer 1106

VisibleMapName

Eigenschaft von

VcCalendarGrid 555

VcDateLine 667

VcDateLineGrid 687

Visual Studio 6.0 mit Visual C++/MFC 17

Vorgang

anlegen 42

Daten bearbeiten 43

Dauer ändern 42

löschen 43

verschieben 43

Vorgänge 136

bearbeiten 398

einer Gruppe in einer Zeile darstellen
291, 297

Hierarchische Anordnung 733

Reihenfolge sichern und wieder laden
447

W

WaitCursorEnabled

Eigenschaft von

VcGantt 807

Width

Eigenschaft von

VcLegendView 1137

VcRect 1288

VcWorldView 1448

WidthActualValue

Eigenschaft von

VcLegendView 1137

VcWorldView 1449

WindowMode

Eigenschaft von

VcLegendView 1138

WorkInProcessType

Eigenschaft von

VcResourceScheduler2 1350

Worldview 139, 807

WorldView

Eigenschaft von

VcGantt 807

Name UpdateBehavior 1448

siehe auch

VcWorldView 1441

WritingDebugFilesEnabled

Eigenschaft von

VcResourceScheduler2 1350

X

XP Visual Style

aktivieren 443

Z

Zeile

Hintergrundfarbe 789

minimale Höhe 769

Zeilenhintergrundfarbe

alternierend 248

Zeilenhöhe

Reduzierung 226

Zeitberechnung

automatisch 1368

Zeiteinheit 218, 797

bei Verschiebung 797

Zeitintervall

kleinstes 218

Zeitkritische Operationen

Wartecursor 228

zeitlicher Abstand 1306

Zeitrechnung 199, 254, 788, 843

Aktuelles Anfangsdatum 1368

Aktuelles Enddatum 1368

Dauer 1369

freier Puffer 1370

frühester Projektanfang 1372

frühestes Projektende 1371

frühestmögliches Anfangsdatum
1369

frühestmögliches Enddatum 1369

geplantes Anfangsdatum 1373

geplantes Enddatum 1370

Gesamtpuffer 1373

nur Knoten mit Vorgängern
berechnen 1372

Projektanfang 1373

spätestmögliches Anfangsdatum
1371

spätestmögliches Enddatum 1370

Verbindungsdauer 1371

Zeitrechnungseingabe 255

Zeitrechnungsergebnis 255

Zeitskala 202, 410

3D-Effekt 1422

aktiv 1425

Anfang 410, 796

Anfang und Ende 203

Anfang und Ende optimieren 838

anpassen 417, 1284

Anzahl 1425

bearbeiten 410, 436

Dialog 410

Ende 411, 796

festlegen 351

Hintergrundfarbe 1418

interaktiv skalieren erlaubt 732

interaktiv skalieren 958

Kalendergitter 1421

Name 1419

Schriftattribute 1419

Schriftfarbe 1419

- skalieren 224
- Spreizung begrenzen 444
- Teilbereich in das Diagramm einpassen 819
- über Index 1427
- vertikale Rasterlinien 1421
- Zeitskalenabschnitt 1420
- Zeitskalenabschnittsanfang interaktiv verändern 958
- Zeitskalen-Dialog anzeigen 224, 790
- Zeitstreifen 355, 1420
- Zeitskalenabschnitt 1420**
 - Anfang 1379
 - bearbeiten 354
 - Breite pro Zeiteinheit 1380
 - kollabieren* 1376
 - Zeiteinheit 1379
- Zeitskalenabschnitte 203, 352, 353**
 - Breite pro Zeiteinheit 354
 - Grenze zwischen Zeitskalenabschnitten 412
 - Kollabieren arbeitsfreier Zeiten 354
 - Skalierung 412
- Zeitstreifen 355, 1378, 1420**
 - Ausgabeformat von Terminen 1356
 - Ausrichtung der Beschriftung der Hauptmarkierungen 1363
 - Aussehen der Hauptmarkierungen 1365
 - Hauptmarkierung 1357
 - Hintergrundfarbe des Musters 1358
 - Kalendername 1354
 - Musterfarbe 1359
 - Mustertyp 1359
 - Nebenmarkierung 1358
 - Position 1363
 - Position der Markierungen 1364
 - Referenzdatum 1363, 1366
 - Schriftattribute 1356
 - Schriftfarbe 1357
 - Trennstrichfarbe 1364
 - Typ 1364
- Zeitumstellung 83**
- Zoom**
 - Methode von*
 - VcGantt* 851
- Zoomen 388, 851**
 - Diagramm an Fenstergröße anpassen und Seitenverhältnis beibehalten 818
 - per Mausrad 225, 808
 - Zoomfaktor 808
- Zoom-Ereignis 962**
- ZoomFactor**
 - Eigenschaft von*
 - VcGantt* 808
- ZoomFactorAsDouble**
 - Eigenschaft von*
 - VcPrinter* 1285
- ZoomingPerMouseWheelAllowed**
 - Eigenschaft von*
 - VcGantt* 808
- Zuordnungstabelle**
 - Typ 1191
 - über Index 1199
 - Zuordnungstabelle bearbeiten 309
- Zuordnungstabellen 208, 769**
 - Angabe von Wertebereichen durch Filter 1190
 - Zuordnungstabellen verwalten 307
- Zusatztext 419**