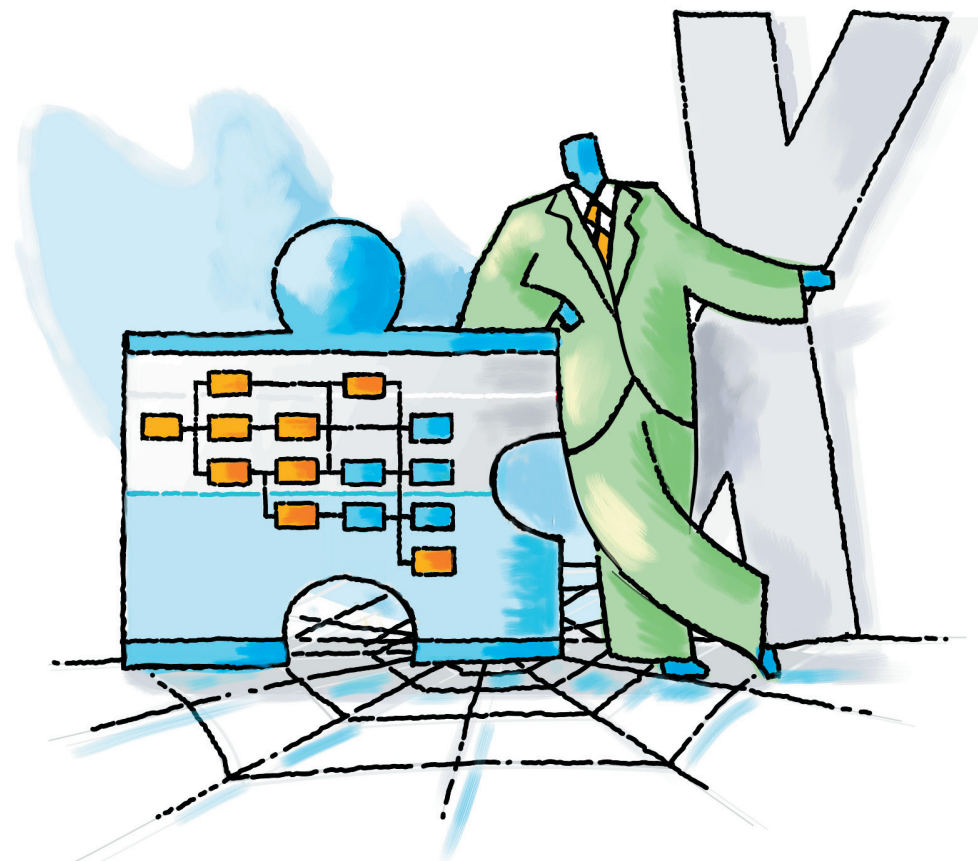


VARCHART XNet

ActiveX Edition 5.2
Benutzer- und
Referenzhandbuch



VARCHART XNet 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	9
1.1	Allgemeines zu VARCHART XNet	9
1.2	Technische Voraussetzungen	12
1.3	Installation	13
1.4	Auslieferung	14
1.5	Datenaustausch mit VARCHART XNet	15
1.6	VARCHART ActiveX in Visual Studio 6.0 oder 7.0 mit Visual C++/MFC	18
1.7	VARCHART ActiveX in HTML-Seiten	21
1.8	Unterstützung und Beratung	27

2	Tutorium	29
2.1	Überblick	29
2.2	VARCHART XNet zur Werkzeugsammlung hinzufügen	30
2.3	VARCHART XNet in einem Formular plazieren	31
2.4	VARCHART XNet automatisch skalieren	33
2.5	Schnittstelle einrichten	34
2.6	Der erste Lauf	38
2.7	Daten aus einer Datei einlesen	42
2.8	Flussrichtung des Netzdiagramms festlegen	45
2.9	Datei mit Projektdaten für die Design-Phase auswählen	47
2.10	Knoten und Verbindungen erzeugen und bearbeiten	48
2.11	Knoten und Verbindungen markieren	51
2.12	Filter für Knoten festlegen	52
2.13	Knotenaussehen festlegen	54
2.14	Knotenformate festlegen	58
2.15	Das Aussehen von Verbindungen festlegen	61
2.16	Positionen von Knoten und Verbindungsbeschriftungen speichern	66
2.17	Hilfsknoten übersichtlich anordnen	69

4 Inhaltsverzeichnis

2.18	Knoten gruppieren	74
2.19	Zeitrechnung des VARCHART XNet festlegen	76
2.20	Diagramm drucken	82
2.21	Diagramm exportieren	83
2.22	Konfigurationseinstellungen speichern	84

3 Wichtige Konzepte 85

3.1	Boxen	85
3.2	Daten	89
3.3	Datentabellen	90
3.4	Datumsangaben und Zeitemstellung	98
3.5	Ereignisse	100
3.6	Filter	101
3.7	Grafikformate	103
3.8	Gruppierung	108
3.9	Identifikation	111
3.10	In-Flow-Gruppierung	113
3.11	Knoten	116
3.12	Knotenaussehen	122
3.13	Knotenformat	124
3.14	Komplettansicht (World View)	127
3.15	Legendenansicht (Legend View)	129
3.16	OLE Drag & Drop	131
3.17	Schreiben von PDF-Dateien	135
3.18	Sprachanpassung von Textausgaben	137
3.19	Statuszeilentext	138
3.20	Tooltips zur Laufzeit	139
3.21	Unicode-Zeichen	140
3.22	Verbindungen	141
3.23	Verbindungsaussehen	147
3.24	Zeitrechnung	149
3.25	Zuordnungstabellen	151

4	Eigenschaftenseiten und Dialogfelder	157
4.1	Allgemeines	157
4.2	Eigenschaftenseite "Allgemeines"	159
4.3	Eigenschaftenseite "Außenbereich"	160
4.4	Eigenschaftenseite "Gruppierung"	162
4.5	Eigenschaftenseite "Knoten"	165
4.6	Eigenschaftenseite "Zusätzliche Ansichten"	170
4.7	Eigenschaftenseite "Objekte"	174
4.8	Eigenschaftenseite "Verbindungen"	176
4.9	Eigenschaftenseite "Zeitrechnung"	178
4.10	Dialogfeld "Datentabellen verwalten"	180
4.11	Dialogfeld "Filter verwalten"	183
4.12	Dialogfeld "Filter bearbeiten"	185
4.13	Dialogfeld "Zuordnungstabellen verwalten"	189
4.14	Dialogfeld "Zuordnungstabelle bearbeiten"	191
4.15	Dialogfeld "Zuordnung einstellen"	194
4.16	Dialogfeld "Knotenaussehen verwalten"	196
4.17	Dialogfeld "Knotenaussehen bearbeiten"	200
4.18	Dialogfeld "Boxen verwalten"	204
4.19	Dialogfeld "Box bearbeiten"	207
4.20	Dialogfeld "Boxformate/Knotenformate verwalten"	209
4.21	Dialogfeld "Boxformat bearbeiten"	211
4.22	Dialogfeld "Knotenformat bearbeiten"	214
4.23	Dialogfeld "Verbindungsformate verwalten"	219
4.24	Dialogfeld "Verbindungsformat bearbeiten"	221
4.25	Dialogfeld "Verbindungsaussehen verwalten"	224
4.26	Dialogfeld "In-Flow-Gruppierung bearbeiten"	228
4.27	Dialogfeld "Linienattribute bearbeiten"	231
4.28	Dialogfeld "Musterattribute bearbeiten"	232
4.29	Dialogfeld "Kalender festlegen"	233
4.30	Dialogfeld "Intervalle verwalten" (Kalender)	235
4.31	Dialogfeld "Kalenderprofile verwalten"	237
4.32	Dialogfeld "Intervalle verwalten" für Tagesprofil	239

6 Inhaltsverzeichnis

4.33	Dialogfeld "Intervalle verwalten" für Wochenprofil	241
4.34	Dialogfeld "Intervalle verwalten" für Variables Profil	242
4.35	Dialogfeld "Intervalle verwalten" für Jahresprofil	244
4.36	Dialogfeld "Texte, Grafiken und Legende festlegen"	245
4.37	Dialogfeld "Legendenattribute"	249
4.38	Dialogfeld "Lizenzierung"	251
4.39	Dialogfeld "Lizenzinformationen anfordern"	253

5 Benutzerschnittstelle 255

5.1	Übersicht	255
5.2	Navigation im Diagramm	256
5.3	Zoomen	257
5.4	Knotendaten bearbeiten	259
5.5	Verbindungen bearbeiten	261
5.6	Knoten und Verbindungen erzeugen	262
5.7	Knoten und Verbindungen markieren, löschen oder verschieben	264
5.8	Seite einrichten	265
5.9	Druckvorschau	269
5.10	Kontextmenü für das Diagramm	272
5.11	Kontextmenü für Knoten	276
5.12	Kontextmenü für Verbindungen	277
5.13	Kontextmenü für die Legende	278

6 Häufig gestellte Fragen 279

6.1	Wie kann das Steuerelement neu lizenziert werden? Was ist bei Problemen mit der Lizenzierung zu tun?	279
6.2	Wie kann erreicht werden, dass eine veränderte ini-Datei des VARCHART ActiveX-Steuerelementes verwendet wird?	280
6.3	Was müssen Benutzer von Borland Delphi beim Upgrade auf eine neue Version von VARCHART XNet tun?	281
6.4	Wieso können Knoten u. U. nicht interaktiv erzeugt werden?	282
6.5	Wieso können Verbindungen u. U. nicht interaktiv erzeugt werden?	283
6.6	Wie verhindert man das interaktive Erzeugen von Knoten?	284
6.7	Wie lassen sich die Standard-Kontextmenüs abschalten?	285

6.8	Was ist bei Problemen mit dem Drucken zu tun?	286
6.9	Wie lässt sich die Performance verbessern?	287
6.10	Fehlermeldungen	288

7 API Referenz 291

7.1	Objekttypen	291
7.2	DataObject	293
7.3	DataObjectFiles	299
7.4	VcBorderArea	302
7.5	VcBorderBox	303
7.6	VcBox	310
7.7	VcBoxCollection	322
7.8	VcBoxFormat	328
7.9	VcBoxFormatCollection	333
7.10	VcBoxFormatField	339
7.11	VcCalendar	349
7.12	VcCalendarCollection	357
7.13	VcCalendarProfile	363
7.14	VcCalendarProfileCollection	366
7.15	VcDataDefinition	372
7.16	VcDataDefinitionTable	373
7.17	VcDataRecord	378
7.18	VcDataRecordCollection	383
7.19	VcDataTable	390
7.20	VcDataTableCollection	393
7.21	VcDataTableField	399
7.22	VcDataTableFieldCollection	405
7.23	VcDefinitionField	410
7.24	VcFilter	414
7.25	VcFilterCollection	421
7.26	VcFilterSubCondition	427
7.27	VcGroup	431
7.28	VcGroupCollection	438

8 Inhaltsverzeichnis

7.29	VcInterval	441
7.30	VcIntervalCollection	449
7.31	VcLegendView	454
7.32	VcLink	462
7.33	VcLinkAppearance	467
7.34	VcLinkAppearanceCollection	476
7.35	VcLinkCollection	482
7.36	VcLinkFormat	486
7.37	VcLinkFormatCollection	490
7.38	VcLinkFormatField	496
7.39	VcMap	500
7.40	VcMapCollection	506
7.41	VcMapEntry	513
7.42	VcNet	521
7.43	VcNode	655
7.44	VcNodeAppearance	661
7.45	VcNodeAppearanceCollection	683
7.46	VcNodeCollection	689
7.47	VcNodeFormat	693
7.48	VcNodeFormatCollection	698
7.49	VcNodeFormatField	704
7.50	VcPrinter	717
7.51	VcRect	734
7.52	VcScheduler	737
7.53	VcWorldView	745

8 Index

755

1 Einleitung

1.1 Allgemeines zu VARCHART XNet

Die interaktive Komponente VARCHART XNet ist ein Element der Produktgruppe VARCHART-X. Diese Produktgruppe beinhaltet ActiveX-Steuer-elemente, die mit der VARCHART-Funktionsbibliothek der NETRONIC Software GmbH entwickelt wurden (VARCHART XGantt, VARCHART XNet, VARCHART XTree).

Mit VARCHART XNet erhalten Sie in wenigen Minuten die erste grafische Darstellung Ihrer Daten. Sie können VARCHART XNet durch seine vielfältigen Gestaltungsmöglichkeiten individuell an Kundenwünsche anpassen.

Mit VARCHART XNet können Sie logische Daten in Form von Netz-Diagrammen visualisieren, editieren und drucken. VARCHART XNet bietet sich für die Darstellung von Computernetzwerken, Workflows oder Klassendiagrammen an. Für die Darstellung von Projektdaten ist eine integrierte Zeitrechnung vorhanden.

Daten können über Dateien oder über die Programmierschnittstelle (API) eingelesen und geschrieben werden. Interaktiv können Sie dann neue Daten einfügen, ändern oder löschen.

Bei den Daten wird zwischen Knoten und Verbindungen unterschieden. Dies wird standardmäßig über zwei Datendefinitionstabellen organisiert. Seit der Version 4.0 ist es jetzt auch möglich, bis zu 90 Datentabellen anzulegen.

Das Datenformat der einzelnen VARCHART-ActiveX-Komponenten kann untereinander kompatibel gemacht werden (abhängig von den Einstellungen in der jeweiligen Datentabelle), so dass ein Datenaustausch oder eine vernetzte Benutzung innerhalb einer Anwendung möglich ist.

Der genaue Aufbau des Datenformats muss zur Entwurfszeit im VARCHART ActiveX-Steuerelement festgelegt werden.

Das Einstellen der Eigenschaften des VARCHART-ActiveX-Steuerelements kann zur Entwurfszeit bequem über Eigenschaftenseiten oder zur Laufzeit dynamisch über die Programmierschnittstelle erfolgen.

Durch die Vielzahl der zur Verfügung gestellten Ereignisse kann in das Standardverhalten der Interaktionen eingegriffen werden.

10 Einleitung

> **Leistungsmerkmale**

- Knoten können durch die Zuweisung verschiedener Knotenaussehen unterschiedlicher Priorität frei gestaltet werden.
- Grafische Attribute können datengesteuert über Filter zugeordnet werden. (Z. B. können alle Knoten einer bestimmten Gruppe in Gelb dargestellt werden.)
- Über sog. Knotenformate kann die grafische Gestaltung der Knoten mit den darin enthaltenen Daten sehr flexibel festgelegt werden.
- Die Positionen von Knoten und Verbindungen können mit dem Inhalt von Datenfeldern synchron gehalten werden, so dass sie speicherbar werden.
- Mögliche Flussrichtungen sind von links nach rechts sowie von oben nach unten. Der eingebaute und per Befehl an der Programmierschnittstelle zu aktivierende Layout-Algorithmus sorgt für möglichst kreuzungsfreie und aufgeräumte Darstellungen.
- Eine Gruppierung nach beliebigen Kriterien und die Anordnung der Gruppen sind möglich. Knoten lassen sich über ein individuell wählbares Datenfeld gruppieren. Die Gruppen sind leicht über ihren Titel zu identifizieren. Somit können auch größere Datenbestände übersichtlich dargestellt werden.
- Benutzerinteraktionen wie das Anlegen, Löschen und Verschieben von Knoten und Verbindungen sind verfügbar.
- Sie können Ihre Diagramme stufenlos zoomen. Beliebige Ausschnitte Ihres Diagramms können Sie interaktiv bildschirmfüllend darstellen lassen. Mit Hilfe der Bildlaufleisten können Sie dann das Fenster wie eine Lupe über der Darstellung verschieben und so auch die anderen Bereiche der Darstellung in derselben Vergrößerung betrachten.
- Wenn Sie einen Knoten über den Rand des Steuerelements verschieben, wird die Darstellung automatisch nachgescrollt (Autoscrolling).
- Hilfsknoten können übersichtlich dargestellt werden. (Knoten können auf demselben Rang positioniert werden wie ihre Vorgänger.)
- Das Aussehen von Verbindungen ist frei gestaltbar (z.B. Darstellung des kritischen Weges in Rot).
- Ihre Darstellungen können mit Titel und Legende gedruckt oder exportiert werden (Formate: VMF, WMF, JPG, BMP, EPS, GIF, PCX, PNG, TIF). (Zum Format VMF siehe Kapitel "Wichtige Begriffe: Viewer Metafile (*.vmf)".)

- Eine Druckausgabe mit Seitenaufteilung und Seitenvorschau ist integriert und ermöglicht das sofortige Ausdrucken aller Darstellungen. Auf Wunsch kann jede Grafik für die Ausgabe auf mehrere Seiten zerlegt werden und einzeln in der Vorschau betrachtet werden. Ist die Ausgabe einer Grafik auf mehrere Seiten festgelegt worden, können Sie die aufgeteilte Grafik jederzeit wieder in eine Gesamtgrafik zurückführen. Sie können dabei jeden Ausschnitt vergrößern.
- Das VARCHART-ActiveX-Steuerelement kann in eine HTML-Seite eingefügt werden, so dass es in einem Browser angezeigt werden kann. (Nähere Informationen finden Sie im Kapitel "Verwendung des VARCHART ActiveX-Steuerelementes in HTML-Seiten" der Einleitung.)

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\xnet\vcnet.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\xnet\vcnet.ocx"`

Sie können VARCHART XNet 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
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 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-XNet-Dateien:

- *vcnet.ocx* (Version 5.0)
- *vcnetd.dll*
- *vcpane32u.dll* (Version 5.5)
- *vcprct32u.dll* (Version 5.5)
- *vcwin32u.dll* (Version 5.5)
- *vxcsv32u.dll* (Version 1.320)

Microsoft Bibliotheken:

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

Die Datei *vcnet.ocx* muss mit dem Befehl `regsvr32 vcnet.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 *vcredist_vs2010_x86.exe* erfolgen.




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

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

1.5 Datenaustausch mit VARCHAR XNet

Der Datenaustausch erfolgt derzeit über Variants. Es kann eine Datei angesprochen werden oder über die API kommuniziert werden. Über die API wiederum kann der ganze Datensatz als ein String eingegeben bzw. ausgelesen werden, oder die Daten werden feldweise angesprochen. Der feldweise Zugriff ist als Eigenschaft für die VcNode- und die VcLink-Objekte möglich.

1.5.1 Schnittstellendefinition

Standardmäßig stehen 21 Datenfelder für die Knoten (Maindata-Tabelle) und sieben Datenfelder für die Verbindungen (Relations-Tabelle) zur Verfügung. Im Dialog **Datentabelle** "" **bearbeiten** können neue Felder angelegt , bestehende Felder gelöscht  oder Felder kopiert  werden. Die Felder können beliebig benannt und der gewünschte Datentyp festgelegt werden (Alphanumerisch, Integer, Datum/Zeit). Bei Datumsfeldern müssen Sie auch noch das verwendete Datumsformat angeben (z.B. DD.MM.YY). Die Datentypen sollten Sie später möglichst nicht mehr verändern, da Formate und Knoten danach ggf. auf falschen Datentypen basieren. Dadurch könnte ein fehlerhaftes Verhalten entstehen.

1.5.2 Aufbau von CSV-Dateien

Geben Sie für die Knoten (Maindata-Tabelle) pro Zeile einen Datensatz ein und trennen Sie die Datenfeldinhalte jeweils durch ein Semikolon. Wenn auch Verbindungen (Relations-Tabelle) eingelesen werden sollen, schreiben Sie in die Zeile nach den Maindata-Zeilen: "****". In die folgenden Zeilen können Sie nun die Datensätze für die Verbindungen eingeben.

Hinweis: Das Dateiformat CSV (Trennzeichen-getrennt) speichert nur Text und Werte. CSV-Dateien werden zur Zeit immer in ANSI geschrieben.

Das folgende Beispiel zeigt Ihnen den Aufbau von CSV-Dateien:

Code-Beispiel

```
1;1.;;;SWDevelopment;A;;GroupA;6;0;100;03.11.00;10.11.00;;0;;
2;1.2.;;;Design&Concept;C;;GroupC;10;0;50;02.11.00;18.11.00;;0;;
3;1.2.1.;;;Requirements;A;;GroupA;5;0;50;02.11.00;07.11.00;;0;;
```


1.5.3 CSV-Dateien verwenden

Die gewünschte Datei können Sie mit der Methode **Open** öffnen und mit der Methode **SaveAsEx** speichern. Wird bei **SaveAs** kein Name angegeben, so wird der beim vorherigen Anwenden der Methode **Open** angegebene Name verwendet .

Hinweis: CSV-Dateien können sowohl in ANSI als auch in Unicode gelesen und geschrieben werden (automatische Erkennung beim Einlesen).

Code-Beispiel

```
VcNet1.Open "c:\daten\beispiel1.net"  
...  
VcNet1.SaveAs ""  
' oder  
VcNet1.SaveAs "c:\daten\beispiel2.net"
```

1.5.4 Knoten- und Verbindungsdaten per API an VARCHART XNet übergeben

Über die Schnittstelle wird jeder Knoten mit der Methode **InsertNodeRecord** übergeben. Zum Einfügen der Verbindungen wird die Methode **InsertLinkRecord** verwendet. Mit der Methode **EndLoading** wird das Ende des Ladevorgangs angezeigt und die Aktualisierung der Grafik ausgelöst.

Code-Beispiel

```
Dim data As String  
data = "1;1.;;SWDevelopment;A;;GroupA;6;6;100;03.11.00;10.11.00;;0;;"  
VcNet1.InsertNodeRecord data  
VcNet1.EndLoading
```

1.5.5 Knotendaten aus VARCHART XNet ausfragen

Über die Eigenschaft **NodeCollection** erhält man ein **VcNodeCollection**-Objekt, mit dem man alle Knoten (**vcAll**), alle sichtbaren Knoten (**vcAllVisible**) oder nur die markierten Knoten (**vcMarked**) erfassen kann. Mit den Methoden **FirstNode** und **NextNode** erhält man die einzelnen Node-Objekte. Mit der Methode **SelectNodes** kann man die ausgewählte Knotenmenge einschränken. Die Node-Objekte können wiederum nach allen Datenfeldern des Knotens (**AllData**) oder nach speziellen Datenfeldern des Knotens (**DataField**) ausgefragt werden.

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection  
Dim node As VcNode  
Dim value As String
```

```

Set nodeCltn = VcNet1.NodeCollection
nodeCltn.SelectNodes (vcAll)
Set node = nodeCltn.FirstNode
Do Until node Is Nothing
    '
    ' Zugriff auf Feld 0 eines jeden Knotens
    '
    value = node.DataField(0)
    '
    ' Zugriff auf alle Daten
    '
    value = node.AllData
    Set node = nodeCltn.NextNode
Loop

```

1.5.6 Verbindungsdaten aus VARCHAR XNet ausfragen

Über die Eigenschaft **LinkCollection** erhält man ein **VcLinkCollection**-Objekt. Mit den Methoden **FirstLink** und **NextLink** erhält man die einzelnen Link-Objekte, die man wiederum nach allen Datenfeldern der Verbindung (**AllData**) oder nach speziellen Datenfeldern der Verbindung (**DataField**) ausfragen kann.

Code-Beispiel

```

Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim value As String

Set linkCltn = VcNet1.LinkCollection
Set link = linkCltn.FirstLink
Do Until link Is Nothing
    '
    ' Zugriff auf Feld 0 eines jeder Verbindung
    '
    value = link.DataField(0)
    '
    ' Zugriff auf alle Daten
    '
    value = link.AllData
    Set link = linkCltn.NextLink
Loop

```

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).

- 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 *xnet.h* und *xnet.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.

Danach sollten Sie die zuvor nicht deselektierte Klasse aus dem Projekt entfernen und stattdessen die von NETRONIC Software GmbH im Unterverzeichnis MFC des Installationsverzeichnisses des VARCHART-ActiveX-Steuerelements vorhandene Datei *xnet.cpp* einfügen. Eine entsprechende Headerdatei namens *xnet.h* steht dort ebenfalls bereit.

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

Code-Beispiel

```
#include "xnet.h"
using namespace XNet;
```

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

```
XNet::CVcTitle title = VcNet1.GetTitle();
```

In Ereignisprozeduren bekommen Sie statt Objekten nur LPDISPATCH-Zeiger übergeben, die Sie aber über die jeweilige **Attach**-Methode eines

20 Einleitung

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="VcNet1" WIDTH=700 HEIGHT=350  
  CLASSID="CLSID:3C415F1E-CFBA-11D2-B467-02608C4302A9"  
  CODEBASE="vcnet.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

```
VcNet1.ConfigurationName =
"http://www.netronic_test.com/xnet_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

<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.mspx?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

26 Einleitung

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 XNet die speziellen Anforderungen Ihres Netzdiagrammes erfüllt?

Sie wollen sich eine Vorstellung davon machen, wie viel Aufwand es ist, die eine oder andere Anforderung ihres Netz-Diagramms zu realisieren?

Sie machen gerade den Anfangstest mit VARCHART XNet 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 XNet 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 XNet vertraut, die notwendig sind, um die Netzplan-Komponente in eigene Anwendungen integrieren zu können.

Wir werden Schritt für Schritt die bedeutsamen Aspekte von VARCHART XNet 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 XNet 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 XNet bietet.

Als Entwicklungsumgebung für die Beispiele verwenden wir Visual Basic 6.0.

2.2 VARCHART XNet zur Werkzeugsammlung hinzufügen

Gehen Sie wie folgt vor, um das VARCHART-XNet-Steuerelement zur Werkzeugsammlung hinzuzufügen:

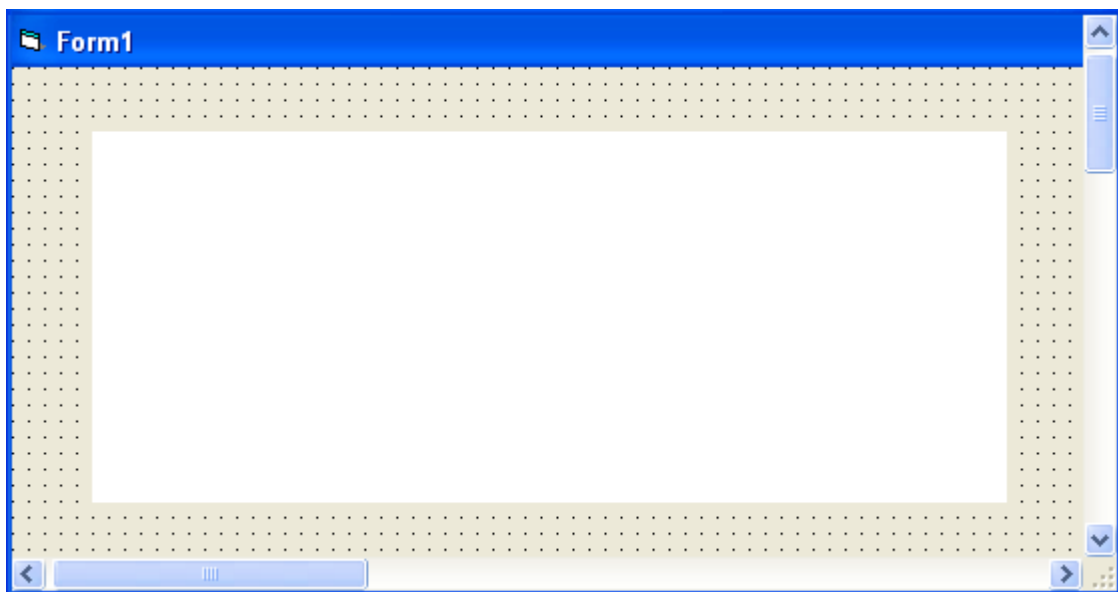
1. Wählen Sie im **Projekt--Menü** von Visual Basic die Option **Komponenten**.
2. Wählen Sie auf der Karteikarte **Steuerelemente** aus der Liste **NETRONIC VARCHART XNet** aus und klicken Sie auf **OK**.

Nach erfolgreicher Aufnahme des VARCHART-XNet-Steuerelements in die Werkzeugsammlung erscheint seine Ikone in der Werkzeugsammlung:



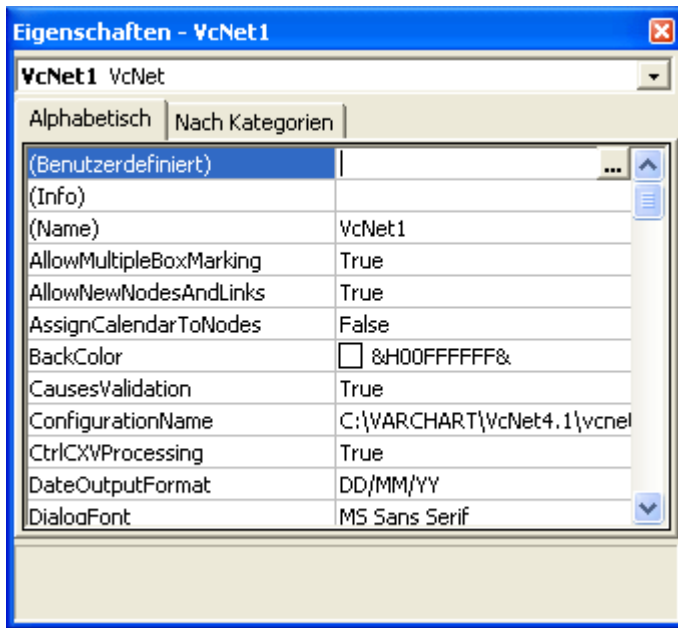
2.3 VARCHART XNet in einem Formular plazieren

Um VARCHART XNet in ein Visual-Basic-Formular einzufügen, müssen Sie es nach dem Hinzufügen nur in der Werkzeugsammlung anklicken und anschließend an der Stelle des Formulars, an der es eingebaut werden soll, mit der Maus einen Rahmen für VARCHART XNet ziehen. Dann erscheint das VARCHART-XNet-Steuerelement (zunächst als weiße Fläche) in der von Ihnen bestimmten Größe. Diese können Sie selbstverständlich mit Hilfe der Maus noch modifizieren.

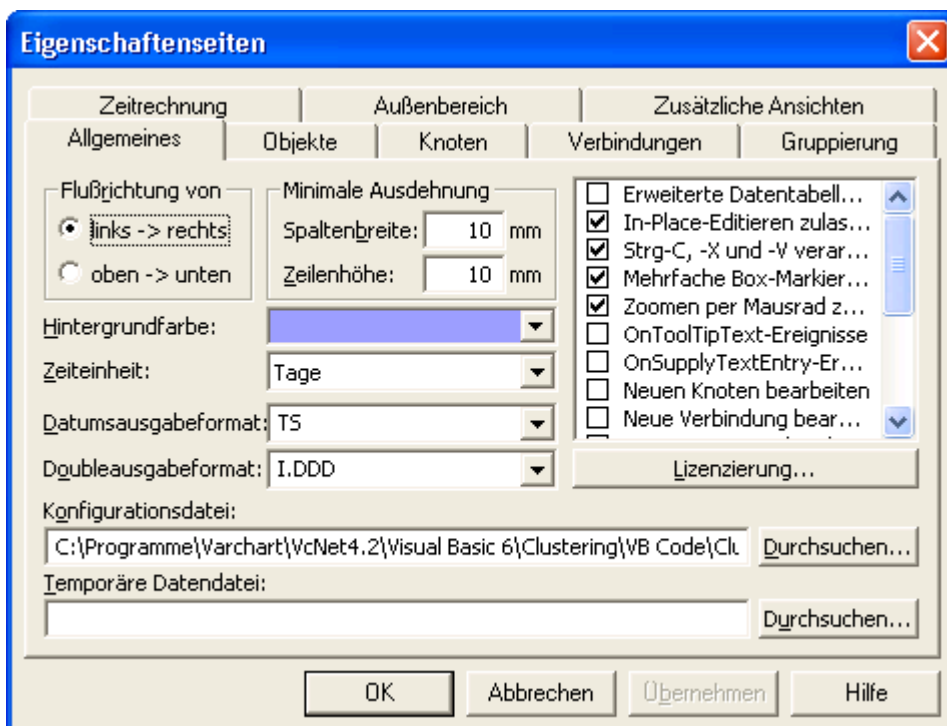


Im Eigenschaftfenster für VARCHART XNet können Sie mit Hilfe des Eintrags **Benutzerdefiniert** die VARCHART-XNet-Eigenschaftenseiten aktivieren, oder bei neueren Entwicklungsumgebungen, eine entsprechende Ikone aktivieren.

32 VARCHART XNet in einem Formular plazieren



Oder Sie können das VARCHART-XNet-Steuerelement im Formular markieren, die rechte Maustaste drücken und im Kontextmenü den Befehl **Eigenschaften** wählen.



Hinweis: Das eingefügte Steuerelement heißt hier und in allen Code-Beispielen **VcNet1**.

2.4 VARCHART XNet automatisch skalieren

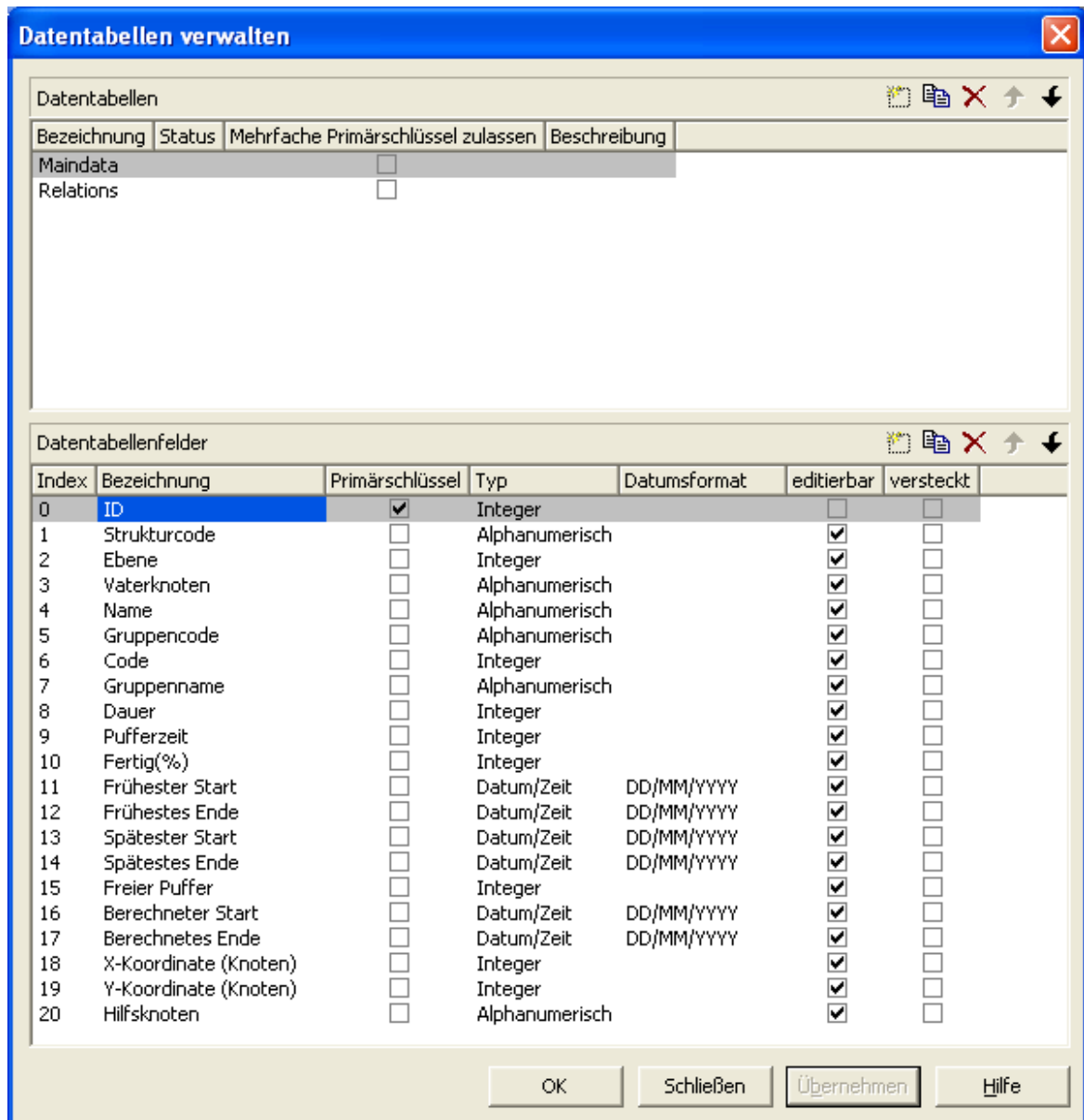
Wenn der rechte und der untere Rand des VARCHART-XNet-Steuerlements zur Laufzeit immer auf die Größe des gesamten Fensters angepasst werden sollen, können Sie das z. B. mit Hilfe des folgenden Codes erreichen:




Code-Beispiel

```
Private Sub Form_Resize()  
    If ScaleWidth - VcNet1.Left > 0 And _  
        ScaleHeight - VcNet1.Top > 0 Then  
        VcNet1.Width = ScaleWidth - VcNet1.Left  
        VcNet1.Height = ScaleHeight - VcNet1.Top  
    End If  
End Sub
```

2.5 Schnittstelle einrichten

Richten Sie nun die Schnittstelle ein, indem Sie die Datenfelder der Tabellen anpassen. Dazu klicken Sie bitte auf der Eigenschaftenseite **Objekte** auf die Schaltfläche **Datentabellen...** und öffnen damit den gleichnamigen Dialog.



Wählen Sie in der oberen Liste die Tabelle **Maindata** (Knotendaten). In der unteren Liste können nun neue Felder dieser Datentabelle angelegt , bestehende Felder gelöscht  oder Felder kopiert  werden. Ein Feldname lässt sich editieren, indem Sie ihn zweifach anklicken. Der Datentyp kann aus einer Kombobox ausgewählt werden.

Das Feld mit dem Index "0" hat standardmäßig den Namen "ID" und ist als "Alphanumerisch" vereinbart. Um der Beispiel-Schnittstelle gerecht zu

werden, benennen Sie das Feld bitte in "Nummer" um und wählen den Datentyp "Integer". Die ID soll nicht editierbar sein, damit sie im Standarddialog **Vorgänge bearbeiten** nicht überschrieben werden kann.

Aktivieren Sie für die Felder "Berechneter Start" und "Berechnetes Ende" das Kontrollkästchen **versteckt**, damit der Anwender deren Werte im Standarddialog **Vorgänge bearbeiten** nicht sieht.

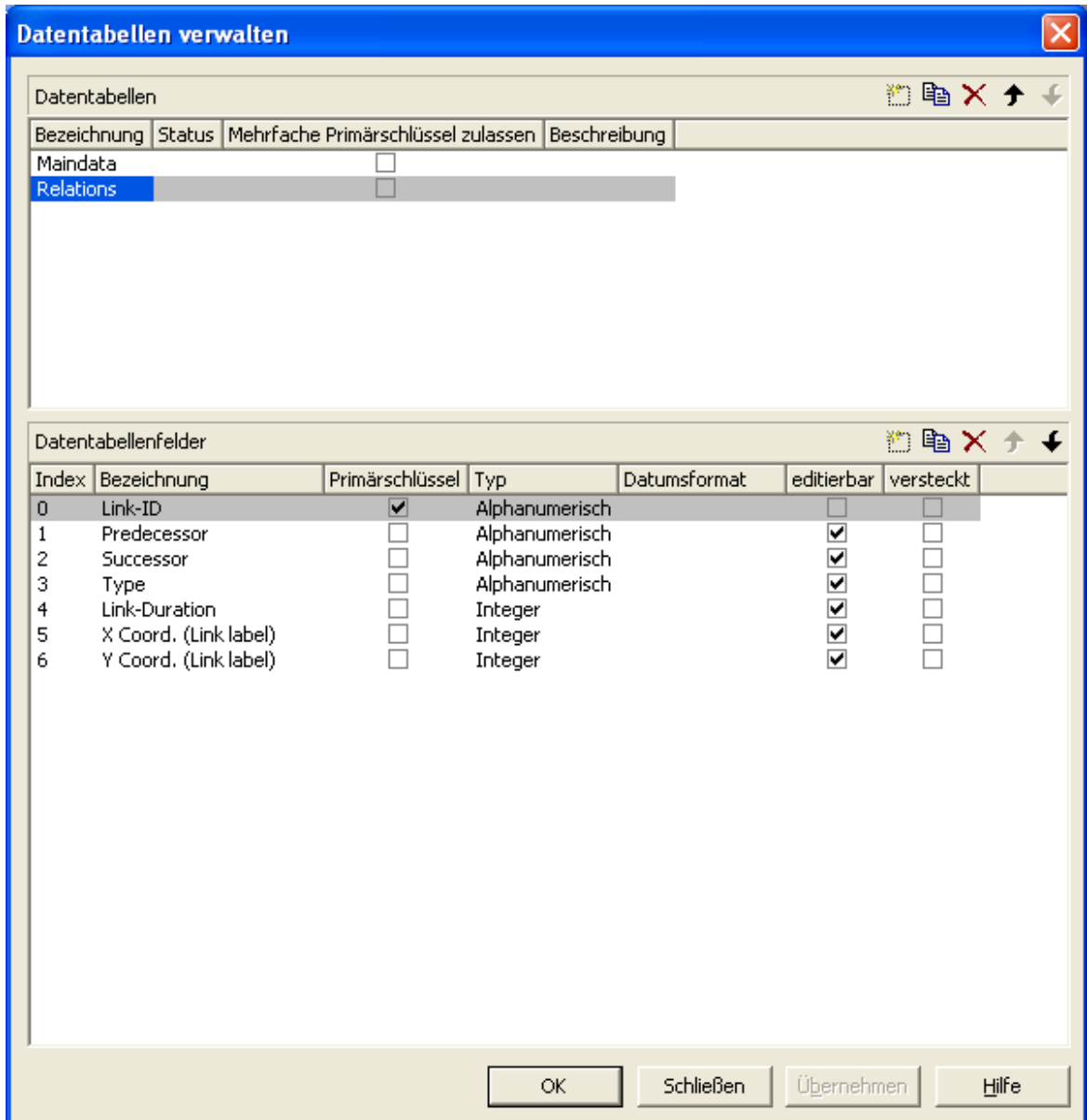
Bei den **Datum/Zeit**-Feldern können Sie angeben, in welchem Format das Datum übergeben wird. Wählen Sie "DD.MM.YYYY".

Um ein Feld zu bestimmen, durch das der Knoten eindeutig identifiziert werden kann, wählen Sie für das Feld **ID** bitte die Option **Primärschlüssel**.

Felder der Maindata-Tabelle:

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat
0	Nummer	True	Integer	
1	Strukturcode	False	Alphanumerisch	
2	Ebene	False	Integer	
3	Vaterknoten	False	Alphanumerisch	
4	Name	False	Alphanumerisch	
5	Gruppencode	False	Alphanumerisch	
6	Code	False	Integer	
7	Gruppenname	False	Alphanumerisch	
8	Dauer	False	Integer	
9	Pufferzeit	False	Integer	
10	fertig (%)	False	Integer	
11	Frühester Start	False	Datum/Zeit	DD.MM.YYYY
12	Frühestes Ende	False	Datum/Zeit	DD.MM.YYYY
13	Spätester Start	False	Datum/Zeit	DD.MM.YYYY
14	Spätestes Ende	False	Datum/Zeit	DD.MM.YYYY
15	Freier Puffer	False	Integer	
16	Berechneter Start	False	Datum/Zeit	DD.MM.YYYY
17	Berechnetes Ende	False	Datum/Zeit	DD.MM.YYYY
18	X-Koordinate (Knoten)	False	Integer	
19	Y-Koordinate (Knoten)	False	Integer	
20	Hilfsknoten	False	Alphanumerisch	

Wechseln Sie nun zur Tabelle "Relations", die für die logischen Verbindungen verwendet wird.



Wählen Sie für das Feld "Link-ID" die Option "Primärschlüssel" und deaktivieren das Kontrollkästchen **editierbar**.

Hinweis: Wenn Sie einen Namen eingeben, der bereits für ein anderes Feld vergeben ist, wird dieser nicht angenommen. Der vorherige Name erscheint wieder.

Durch Anklicken der Schaltfläche **Übernehmen** werden die Änderungen für diese Konfiguration festgeschrieben. Alternativ werden die Einstellungen auch bei **OK** und beim Wechsel der Eigenschaftenseite übernommen und stehen damit direkt in den anderen Eigenschaftenseiten zur Verfügung.

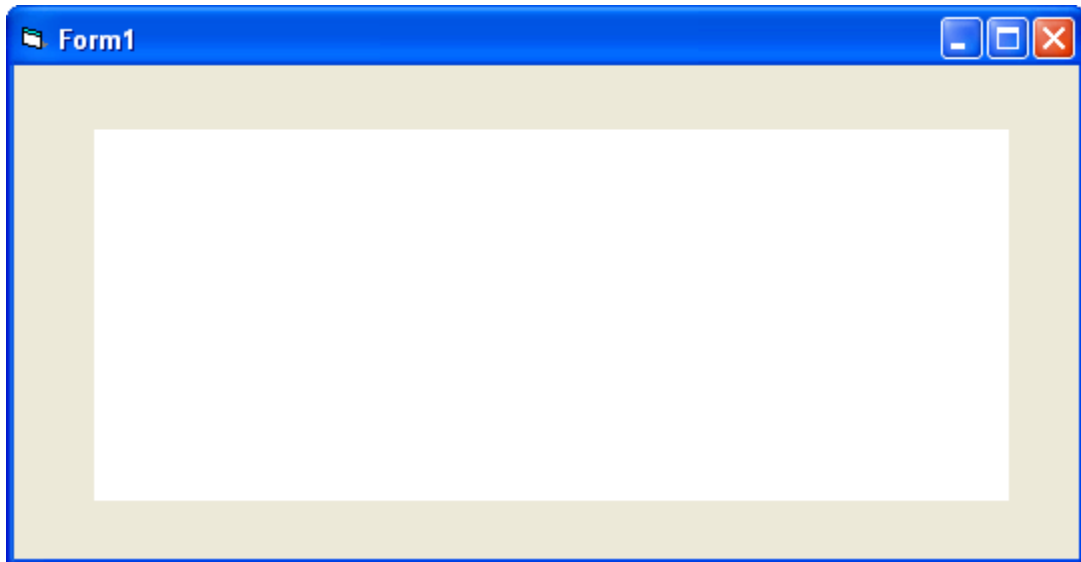
Felder der Relations-Tabelle:

Index	Bezeichnung	Primärschlüssel	Typ
0	Verbindungs-ID	True	Alphanumerisch

Index	Bezeichnung	Primärschlüssel	Typ
1	Vorgänger	False	Alphanumerisch
2	Nachfolger	False	Alphanumerisch
3	Verbindungstyp	False	Alphanumerisch
4	Zeitabstand	False	Integer
5	X-Koordinate (Verbindung)	False	Integer
6	Y-Koordinate (Verbindung)	False	Integer

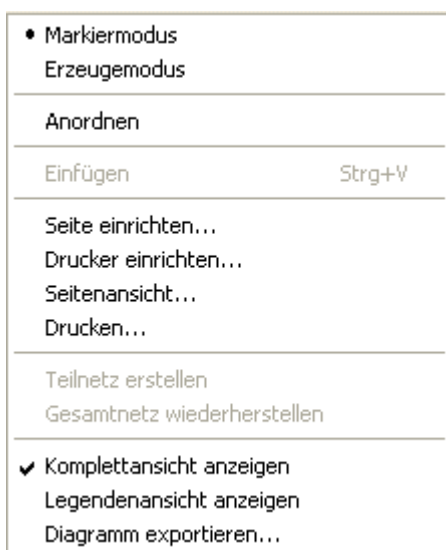
2.6 Der erste Lauf

Über **Ausführen – Starten**, die Funktionstaste **F5** oder die entsprechende Visual-Basic-Ikone (▶) starten Sie nun das Programm. Das angelegte Formular erscheint mit dem leeren Diagramm.



> Knoten und Verbindungen anlegen

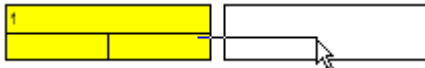
Varchart XNet besitzt zur Laufzeit zwei grundlegende Modi: den Markiermodus und den Erzeugemodus. Knoten und Verbindungen können Sie nur im Erzeugemodus anlegen. Um in den Erzeugemodus zu wechseln, klicken Sie mit der rechten Maustaste in den freien Diagrammbereich und wählen Sie im Kontextmenü den Befehl **Erzeugemodus**.



Im Erzeugemodus wird der Cursor im leeren Diagrammbereich zu einem Knotenphantom in der Form eines Rechtecks. Nun können Sie Knoten

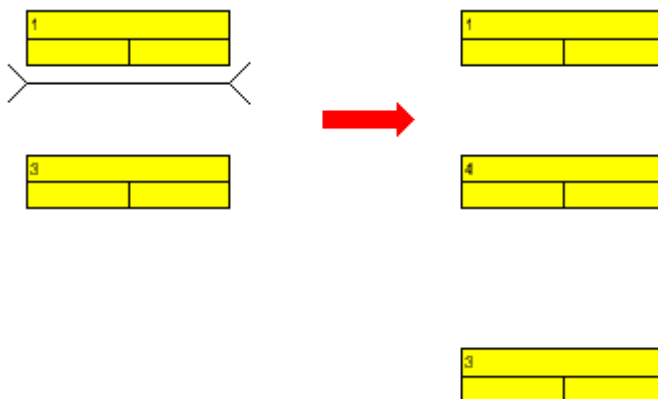
erzeugen, indem Sie mit der linken Maustaste in den Diagrammbereich klicken.

Verbindungen erzeugen Sie, indem Sie die Maus mit gedrückter linker Maustaste von dem einem Knoten zu einem anderen Knoten ziehen. Dabei wird der Cursor zu einem Pfeil-Symbol mit einem Knoten-Phantom.



Die Verbindung wird erzeugt, sobald Sie die Maustaste loslassen. Lassen Sie die Maustaste im leeren Diagrammbereich los, wird dort ein neuer Knoten zusammen mit einer Verbindung vom Ausgangsknoten erzeugt. Auf diese Weise können Sie Knoten und Verbindungen gemeinsam erzeugen.

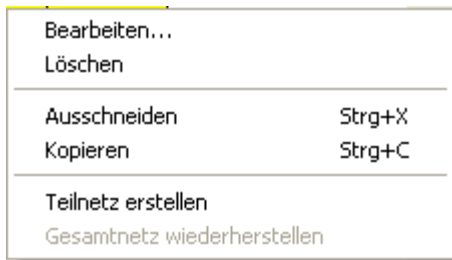
Wenn Sie die Maus zwischen zwei dicht übereinander oder nebeneinander stehende Vorgänge führen, um dazwischen einen neuen Vorgang anzulegen, verändert das Phantom seine Form und wird zu einer waagerechten Linie mit zwei nach innen gerichteten Pfeilspitzen ("Knochen"). Klicken Sie nun mit der linken Maustaste, so wird der neue Vorgang zwischen den beiden vorhandenen Vorgängen eingefügt.



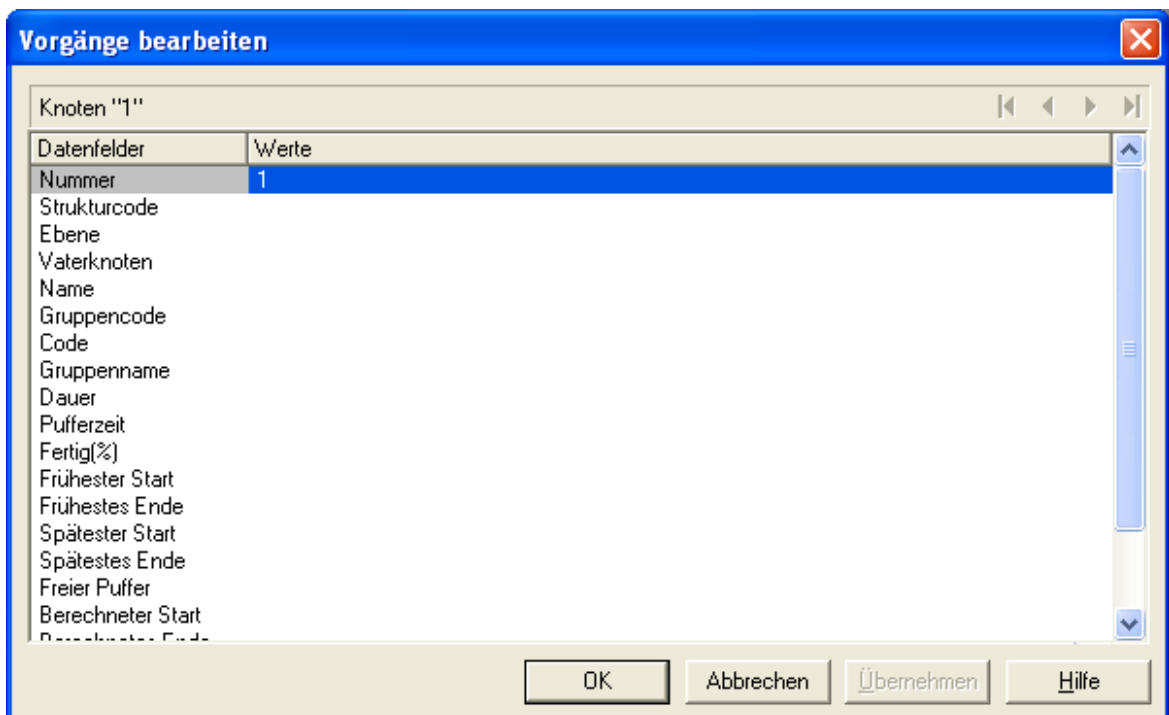
> Knoten bearbeiten

Um einen Knoten zu bearbeiten, wechseln Sie zum Markiermodus und doppelklicken Sie auf den Knoten. Dann erscheint das Dialogfeld **Vorgänge bearbeiten**.

Oder klicken Sie mit der rechten Maustaste auf den Knoten. Dann erscheint ein Kontextmenü, über das Sie den Knoten bearbeiten, kopieren, ausschneiden oder löschen können.

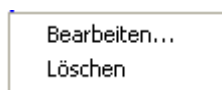


Wählen Sie im Kontextmenü die Option **Bearbeiten**, um das Dialogfeld **Vorgänge bearbeiten** für einen Knoten aufzurufen. Sie finden hier die Datenfelder wieder, die Sie auf der Eigenschaftenseite **Datendefinition** vereinbart haben. (Die Datenfelder, die dort als **versteckt** definiert worden sind, erscheinen nicht in diesem Dialogfeld. Die als **nur lesbar** definierten Datenfelder können in diesem Dialogfeld nicht verändert werden.)

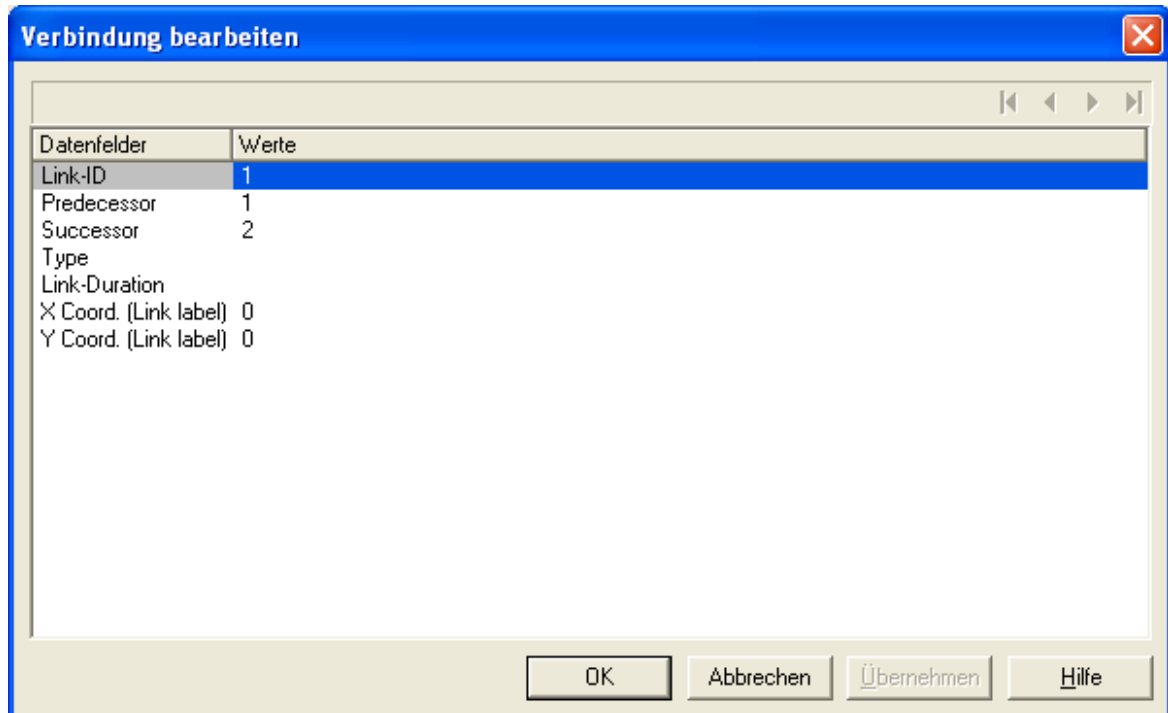


> Verbindung bearbeiten

Eine Verbindung können Sie im Erzeugemodus durch einfaches Ziehen der Maus von einem Knoten zum anderen erstellen. Um eine Verbindung zu bearbeiten, öffnen Sie entweder über den Befehl **Bearbeiten** des entsprechenden Kontextmenüs oder mit einem Doppelklick auf die Verbindung das Dialogfeld **Verbindung bearbeiten**.



Kontextmenü für Verbindungen



In diesem Dialogfeld können Sie alle Daten der Verbindung bearbeiten.

> **Knoten und Verbindungen interaktiv verschieben**

Knoten und Verbindungen können Sie mit der Maus beliebig verschieben. Positionieren Sie dazu im Markiermodus den Cursor auf einem Knoten oder einer Verbindung und drücken Sie die linke Maustaste. Der Cursor wird dann zu einem kleinen Quadrat mit vier Pfeilen. Sie können den ausgewählten Knoten bzw. die ausgewählte Verbindung beliebig verschieben, solange Sie die linke Maustaste gedrückt halten. Beim Verschieben eines Knotens werden die zugehörigen Verbindungen automatisch angepasst.

> **Zurück zum Design-Modus**

Beenden Sie nun den ersten Lauf, indem Sie das Formular schließen.

2.7 Daten aus einer Datei einlesen

Um das VARCHART XNet für die nächsten Schritte mit Daten zu füllen, können Sie die mitgelieferte Datei *tutorial.net* beim Start automatisch laden. (*tutorial.net* ist eine CSV-Datei entsprechend Ihrer eingestellten Schnittstelle. Zu Veränderungen der Schnittstelle siehe "Tutorium: Schnittstelle einrichten".)

Dazu reagieren Sie auf das Ereignis **Form_Load**:

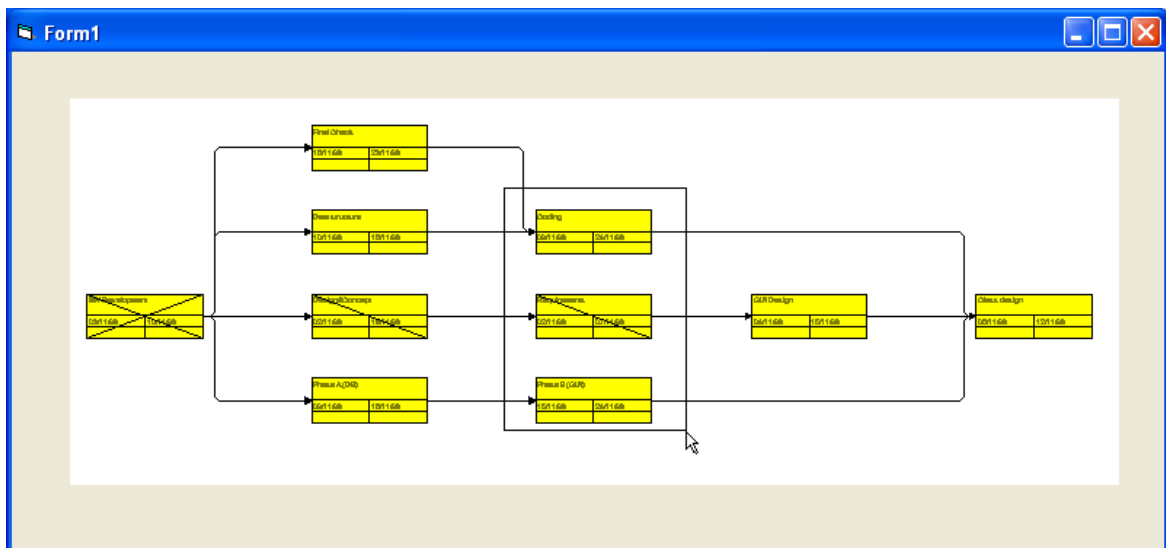
Code-Beispiel

```
Private Sub Form_Load()
    VcNet1.Open "C:\Programme\Varchart\xnet\tutorial.net"
End Sub
```

Die Pfadangabe ist abhängig von Ihrer Installation. Speichern Sie nun das Projekt. Wenn Sie nun das Programm starten, werden die Knoten und Verbindungen des Projektes angezeigt.

VARCHART XNet stellt ein Netz-Diagramm komplett dar.

Sie können einen Ausschnitt Ihres Diagramms bildschirmfüllend darstellen lassen, indem Sie mit gedrückter linker Maustaste ein Gummirechteck um den zu vergrößernden Ausschnitt aufziehen und dann (bei noch gedrückter linker Maustaste) auf die rechte Maustaste klicken.



Nun wird der gewählte Ausschnitt bildschirmfüllend dargestellt. Mit Hilfe der Bildlaufleisten können Sie das Fenster wie eine Lupe über der Darstellung verschieben und so auch die anderen Bereiche der Darstellung in derselben Vergrößerung betrachten.

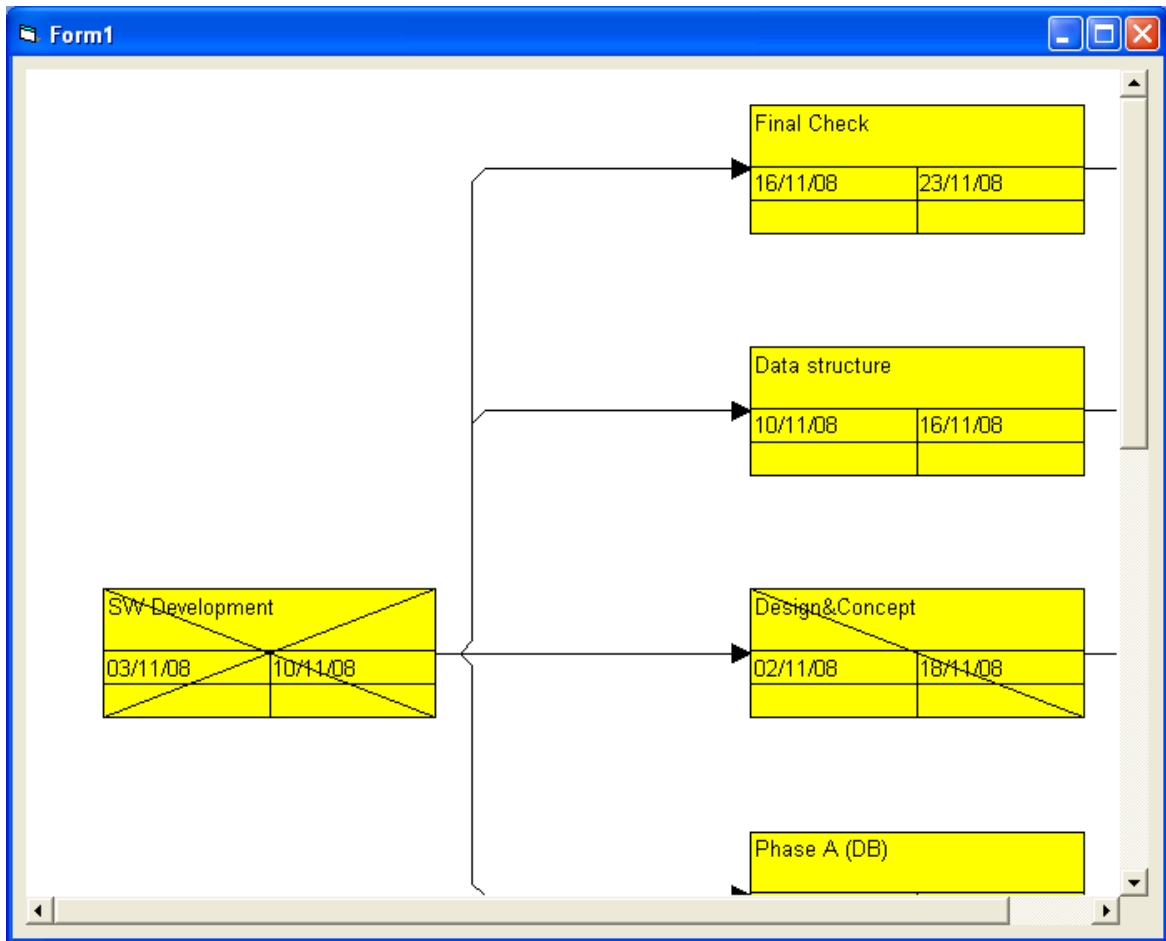
The screenshot shows a Windows form titled 'Form1' with a white background and a blue title bar. It contains three yellow rectangular boxes representing tasks in a Gantt chart. Each box is divided into three sections: a top section for the task name, a bottom-left section for the start date, and a bottom-right section for the end date. Arrows point from the left edge of the form to the start date of each task.

Task Name	Start Date	End Date
Coding	09/11/08	24/11/08
Requirements	02/11/08	07/11/08
Phase B (GUI)	15/11/08	24/11/08

Kehren Sie nun wieder zum Design-Modus zurück. Ergänzen Sie ggf. den folgenden Code, damit Scrollbars in X- und Y-Richtung ausgegeben werden. (Die Ausgabe von Scrollbars hängt von dem gewählten Zoomfaktor ab.)

Code-Beispiel

```
Private Sub Form_Load()
    VcNet1.Open "C:\Programme\Varchart\xnet\tutorial.net"
    VcNet1.Zoomfactor = 150
End Sub
```



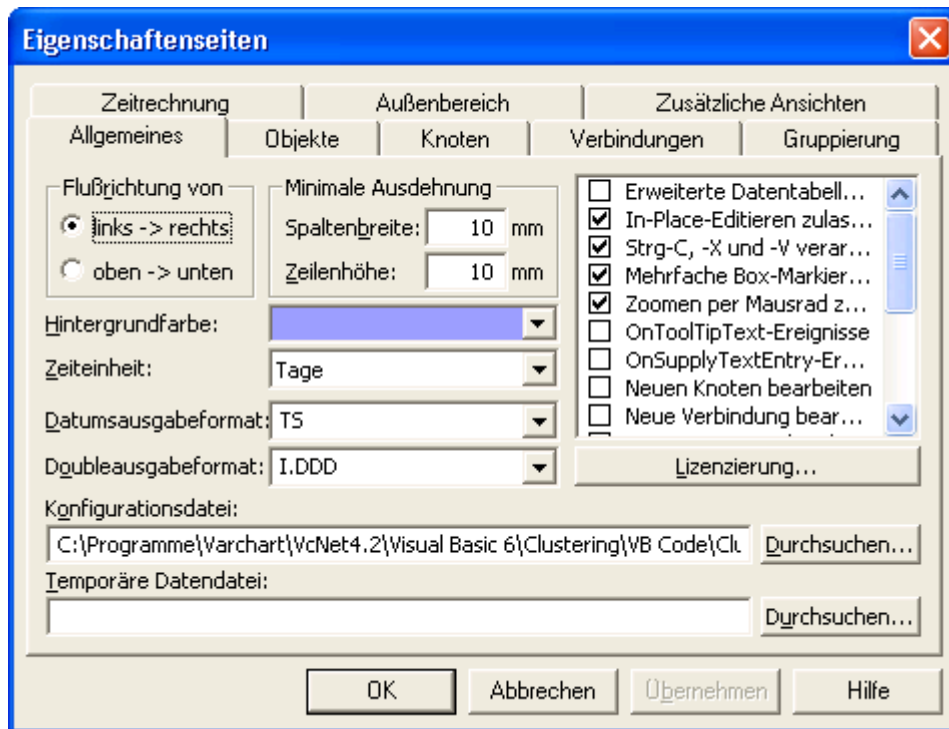
Kehren Sie wieder zum Design-Modus zurück.

Soll VARCHART XNet das gesamte Formular ausfüllen, dann muss folgendes eingestellt werden:

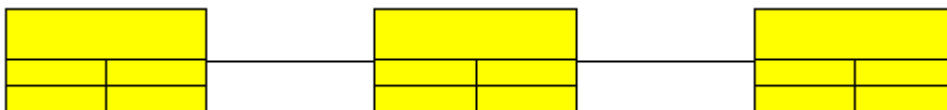
- Stellen Sie sicher, dass die Eigenschaften **Top** und **Left** den Wert 0 haben. Damit wird VARCHART XNet oben links im Formular positioniert.
- Tragen Sie für die VARCHART-XNet-Eigenschaften **Width** und **Height** die Formularwerte von **ScaleWidth** und **ScaleHeight** ein. (Dieser Schritt ist überflüssig, wenn Sie das VARCHART XNet wie oben beschrieben automatisch reskalieren lassen.)

2.8 Flussrichtung des Netzdiagramms festlegen

Auf der Eigenschaftenseite **Allgemeines** können Sie die wichtigsten allgemeinen Einstellungen für das VARCHART XNet festlegen.

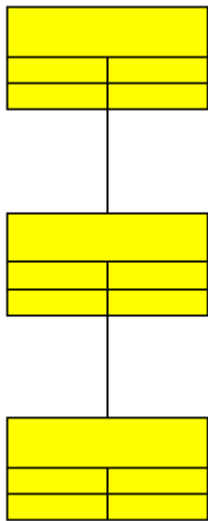


Legen Sie zunächst unter **Flussrichtung** fest, ob die Knoten von links nach rechts oder von oben nach unten angeordnet werden sollen. Probieren Sie beide Einstellungen mit den Daten der Beispieldatei *tutorial.net* aus. Sie können das schon in der Design-Phase tun. Wählen Sie dazu unter **Temporäre Datendatei** die Datei *tutorial.net* aus.



Flussrichtung von links nach rechts

46 Flussrichtung des Netzdiagramms festlegen



Flussrichtung von oben nach unten

Damit Sie Knoten anlegen können, muss die Option **Neue Knoten und Verbindungen zulassen** auf der Eigenschaftenseite **Allgemeines** aktiviert sein.

Um das Programm nun mit einem leeren Diagramm zu starten, setzen Sie die entsprechende Code-Zeile auf Kommentar:

Code-Beispiel

```
Private Sub Form_Load()  
    ' VcNet1.Open "C:\Programme\Varchart\xnet\tutorial.net"  
    VcNet1.Zoomfactor = 100  
End Sub
```

Starten Sie das Programm erneut über **Ausführen - Starten**, mit F5 oder über die entsprechende Visual-Basic-Ikone (▶). Das Formular erscheint mit dem leeren Netzdiagramm. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie den Erzeugemodus. Legen Sie per Mausklick einige Knoten neben- und untereinander an sowie einige Verbindungen. Klicken Sie nun mit der rechten Maustaste in den leeren Diagrammbereich und wählen Sie im Kontextmenü den Befehl **Anordnen**. VARCHART XNet ordnet die Knoten nun automatisch gemäß der gewählten Einstellung für die gewählte Flussrichtung an.

2.9 Datei mit Projektdaten für die Design-Phase auswählen

Im Feld **Temporäre Datendatei** auf der Eigenschaftenseite **Allgemeines** können Sie für die Design-Phase eine Datendatei mit Knoten- und Verbindungsdaten angeben oder auswählen, um die Veränderungen der Einstellungen direkt grafisch im Steuerelement kontrollieren zu können.

Klicken Sie auf die **Durchsuchen**-Schaltfläche, um das Windows-Dialogfeld **Laden bzw. Speichern** mit der Voreinstellung "Datendateien (*.net)" aufzurufen. Wählen Sie die mitgelieferte Beispieldatei *tutorial.net* und bestätigen Sie mit **Übernehmen**. Nun werden die Knoten, die in dieser Datei beschrieben sind, im Formular dargestellt.

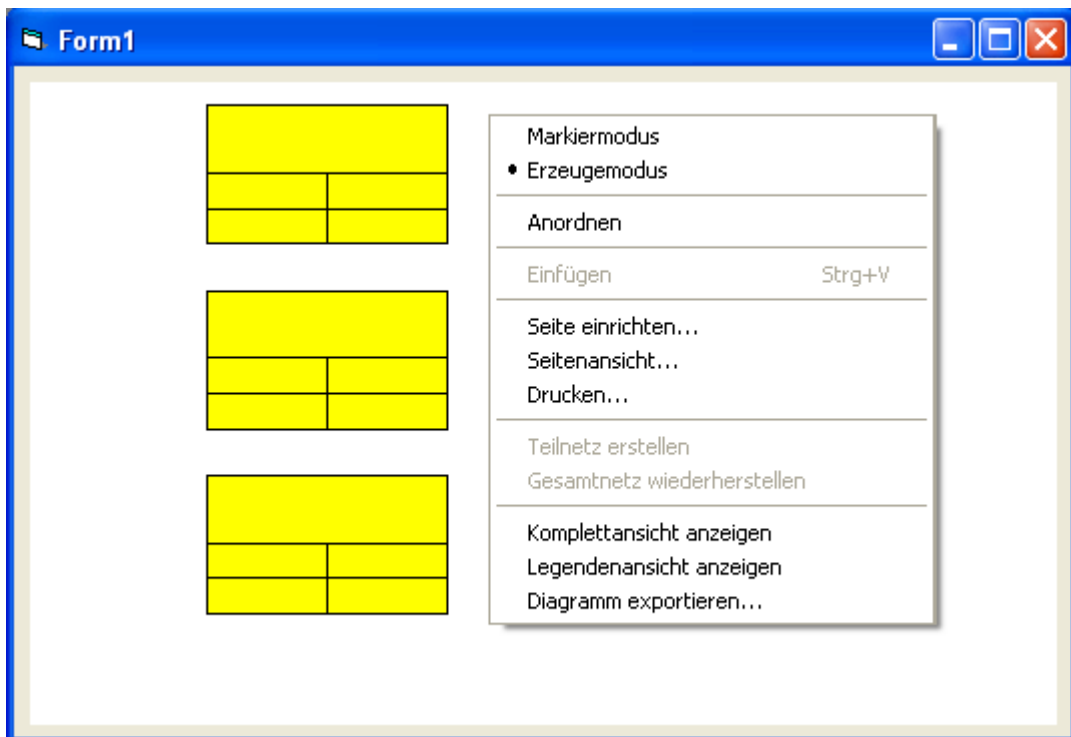
Probieren Sie nun aus, wie Änderungen in den Einstellungen zur Design-Phase dargestellt werden. Öffnen Sie dazu die Eigenschaftenseite **Knoten** und klicken Sie auf die Schaltfläche **Aussehen**. Das Dialogfeld **Knotenaussehen** öffnet sich. Das Knotenaussehen "Standard" ist markiert. Klicken Sie nun auf die **Bearbeiten**-Schaltfläche. Sie gelangen in das Dialogfeld **Knotenaussehen bearbeiten**. Ändern Sie hier verschiedene Einstellungen, beispielsweise die Hintergrundfarbe, die Linienfarbe, den Linientyp, den Schatten etc. Die Änderungen werden Ihnen zunächst im Vorschaufenster dieses Dialogfeldes angezeigt. Sobald Sie Ihre Änderungen mit **OK** oder **Übernehmen** bestätigen, werden die Knoten mit den geänderten Einstellungen im Formular dargestellt.

Beachten Sie bitte, dass die hier ausgewählte Datendatei nur für die Design-Phase gilt. Zur Laufzeit müssen Sie eine Datei mit der Methode **Open** öffnen oder das Netzdiagramm mit den Methoden **InsertNodeRecord** und **InsertLinkRecord** füllen.

2.10 Knoten und Verbindungen erzeugen und bearbeiten

Auf der Eigenschaftenseite **Allgemeines** können Sie festlegen, ob zur Laufzeit neue Knoten und Verbindungen erzeugt werden können (**Neue Knoten und Verbindungen zulassen**), und ob das Dialogfeld **Vorgänge bearbeiten** erscheint, sobald Sie einen neuen Knoten bzw. eine neue Verbindung erzeugt haben (**Neuen Knoten bearbeiten** bzw. **Neue Verbindung bearbeiten**).

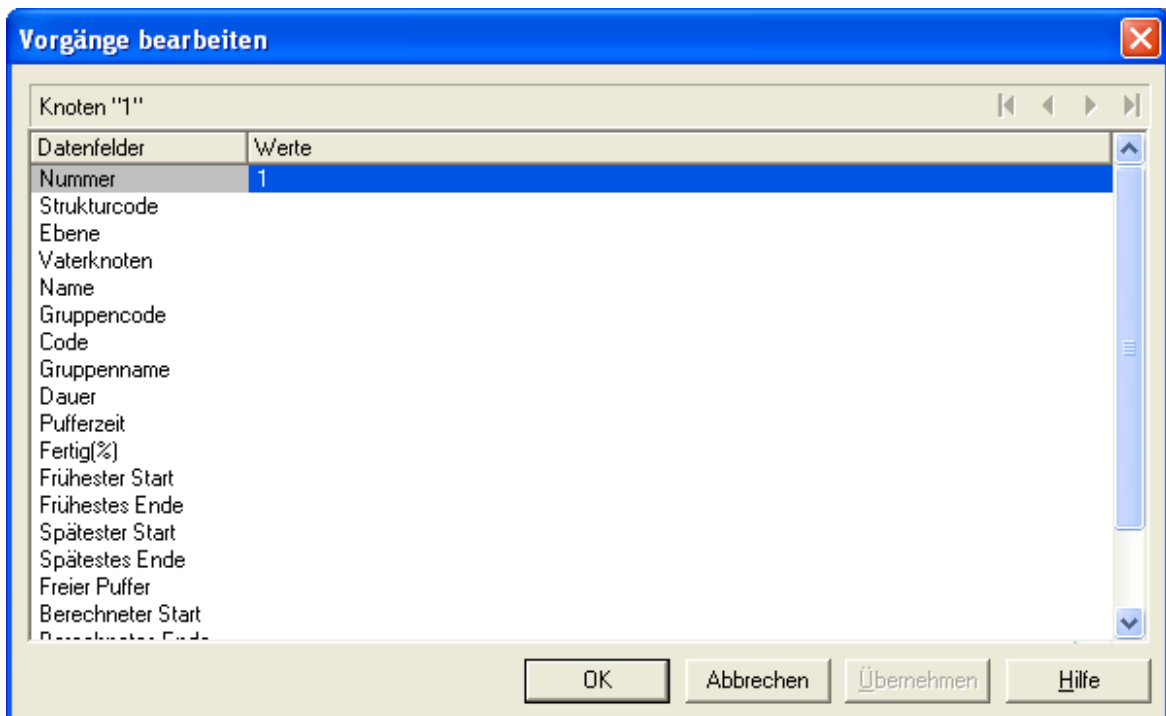
Aktivieren Sie zunächst nur die Option **Neue Knoten und Verbindungen zulassen** und deaktivieren Sie die Optionen **Neuen Knoten bearbeiten** und **Neue Verbindung bearbeiten**. Starten Sie die Applikation durch F5, öffnen Sie durch einen Klick auf die rechte Maustaste das Kontextmenü und wählen Sie hier den **Erzeugemodus**. Klicken Sie nun mit der linken Maustaste in das leere Diagramm. Mit jedem Mausklick wird ein neuer Knoten angelegt.



Wenn die Option **Neue Knoten und Verbindungen zulassen** deaktiviert ist, kann der Anwender zur Laufzeit auch im Erzeugemodus keine neuen Knoten und Verbindungen anlegen. Knoten und Verbindungen können dann nur über die API geladen werden.

Aktivieren Sie nun die Optionen **Neuen Knoten bearbeiten** und **Neue Verbindung bearbeiten**.

Starten Sie die Applikation, wechseln Sie zum Erzeugemodus und klicken Sie mit der rechten Maustaste auf einen Knoten. Das Kontextmenü für Knoten erscheint. Wählen Sie hier die Option **Bearbeiten**. Das Dialogfeld **Vorgänge bearbeiten** öffnet sich. Hier werden Ihnen alle Datenfelder des neuen Vorgangs, die Sie auf der Eigenschaftenseite **Datendefinition** vereinbart haben, mit ihren aktuellen Werten angezeigt. Sobald Sie die Daten mit **OK** bestätigen, wird der neue Vorgang an der Position des Cursors angelegt.



Erzeugen Sie nun einige Knoten und Verbindungen (vgl. "Tutorium: Der erste Lauf"). Sie können nun mehrere Knoten unmittelbar hintereinander bearbeiten. Markieren Sie diese, indem Sie sie bei gedrückter Strg-Taste mit der linken Maustaste anklicken. Klicken Sie dann auf die rechte Maustaste, während der Cursor auf einem der markierten Objekte steht. Dann öffnet sich das Kontextmenü für Knoten. Wählen Sie den Befehl **Bearbeiten**. Das Dialogfeld **Vorgänge bearbeiten** erscheint. Sie können hier die Daten der markierten Vorgänge sukzessive editieren.

In der Kopfzeile dieses Dialogfeldes wird die Identifikationsnummer des Vorgangs angezeigt, den Sie in der **Datenfelder/Werte**-Tabelle bearbeiten können. In Klammern hinter der Identifikationsnummer wird angegeben, um den wievielten der markierten Vorgänge es sich handelt.

Beim Öffnen dieses Dialogfeldes werden die Datenfelder und Werte des ersten markierten Knotens angezeigt. Sie können die Werte aller Datenfelder dieses Vorgangs bearbeiten. Mit den Pfeil-Schaltflächen können Sie zwischen den Knoten navigieren.

50 Knoten und Verbindungen erzeugen und bearbeiten

Schließen Sie nun das Formular und kehren Sie zum Design-Modus zurück.

2.11 Knoten und Verbindungen markieren

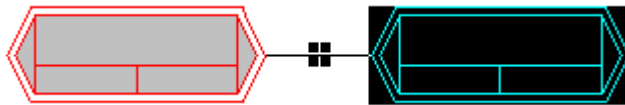
Auf den Eigenschaftenseiten **Knoten** bzw. **Verbindungen** können Sie aus der Kombobox **Markierungstyp** auswählen, in welcher Weise Knoten und Verbindungen markiert werden sollen.

Starten Sie das Programm und legen Sie im Erzeugemodus einige Knoten und Verbindungen an. Klicken Sie nun im Markiermodus einmal auf einen Knoten oder eine Verbindung, um sie zu markieren.

Mehrere Objekte (Knoten oder Verbindungen) können Sie markieren und toggeln, indem Sie sie bei gedrückter Strg-Taste mit der linken Maustaste anklicken. Mehrere Knoten oder Verbindungen können Sie markieren und sammeln, indem Sie diese bei gedrückter Shift-Taste anklicken. Dabei werden die zwischen den markierten Knoten liegenden Knoten und Verbindungen ebenfalls markiert.

Probieren Sie verschiedene Alternativen für den Knotenmarkierungstyp und den Verbindungsmarkierungstyp aus.

In der Abbildung sehen Sie einen durch Invertierung markierten Knoten sowie eine durch Pickmarks markierte Verbindung:

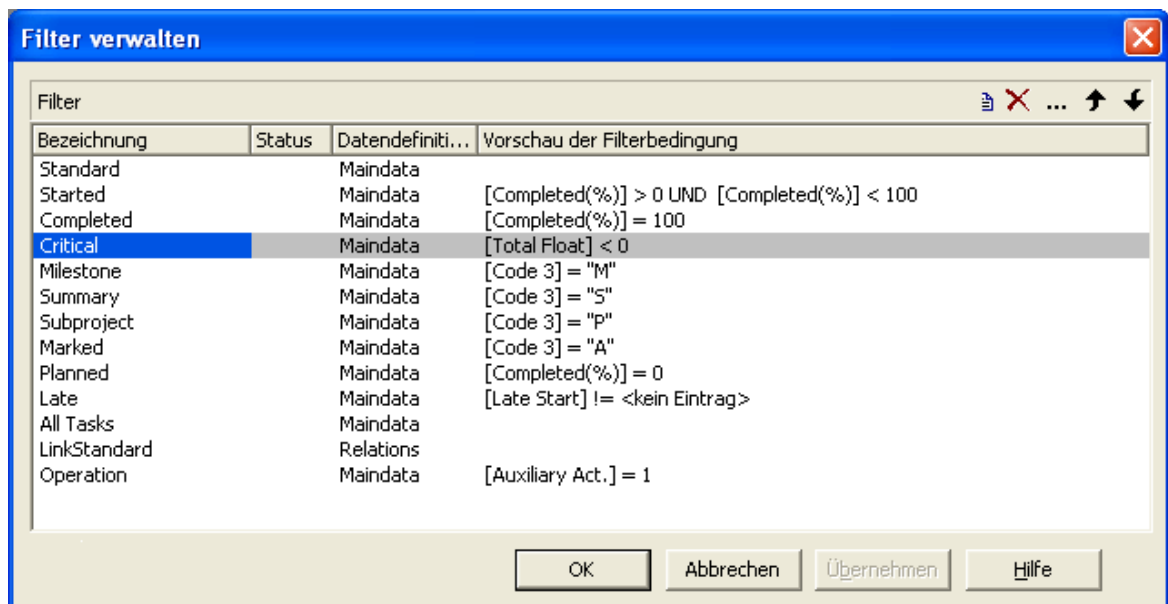


2.12 Filter für Knoten festlegen





Ein Filter enthält Kriterien zur Auswahl von bestimmten Daten, beispielsweise von Knoten und Verbindungen.

Wenn Sie Filter für das Knotenaussehen verwenden, erhalten genau die Knoten, die die Filterkriterien eines bestimmten Knotenaussehens erfüllen, dieses Aussehen.

Um einen Filter für ein Knotenaussehen zu bearbeiten, klicken Sie auf der Eigenschaftenseite **Objekte** auf die Schaltfläche **Filter**. Sie gelangen dann in das Dialogfeld **Filter verwalten**, in dem Sie Filter umbenennen, bearbeiten, neu definieren, kopieren und löschen können.



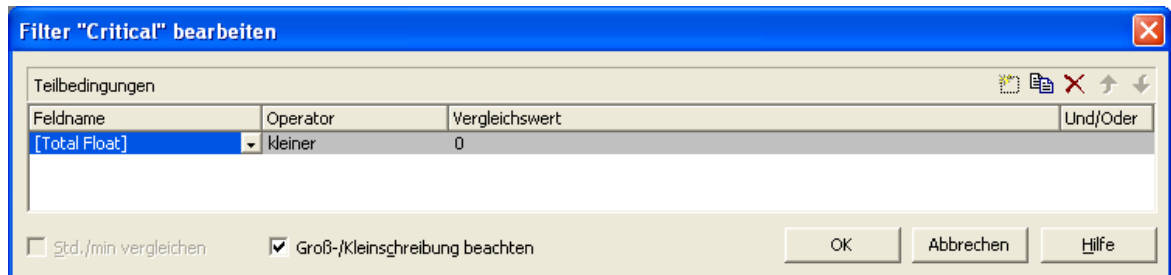
> Schaltflächen im Dialogfeld "Filter verwalten"

-  Filter hinzufügen
-  Filter kopieren
-  Filter löschen
-  Filter bearbeiten

> Filter erstellen und bearbeiten

Lernen Sie nun, wie Sie neue Filter erstellen und bearbeiten können. Klicken Sie dazu auf die Schaltfläche **Filter hinzufügen**. Der neue Filter wird am Ende der Liste angefügt. Ändern Sie seinen Namen in "Critical" um.

Bearbeiten Sie nun den neuen Filter. Klicken Sie dazu auf die Schaltfläche **Filter bearbeiten**. Sie gelangen dann in das gleichnamige Dialogfeld. Machen Sie die folgenden Festlegungen:



In der Kopfzeile wird der aktuelle Filter angezeigt, den Sie hier bearbeiten können.

Unter **Feldname** wird das Datenfeld angezeigt, dessen Inhalt mit dem Vergleichswert verglichen werden soll. Wählen Sie hier das Datenfeld "Total Float".

Unter **Operator** wird der aktuelle Vergleichsoperator angezeigt. Welche Operatoren möglich sind, hängt vom Typ des gewählten Datenfeldes ab. Wählen Sie hier "<".

Im Feld **Vergleichswert** können Sie angeben, mit welchem Wert der Eintrag des Datenfeldes verglichen werden soll. Der Typ des Vergleichswertes muss mit dem Typ des Datenfeldes übereinstimmen, das unter **Feldname** angegeben wurde. Wählen Sie hier "0".

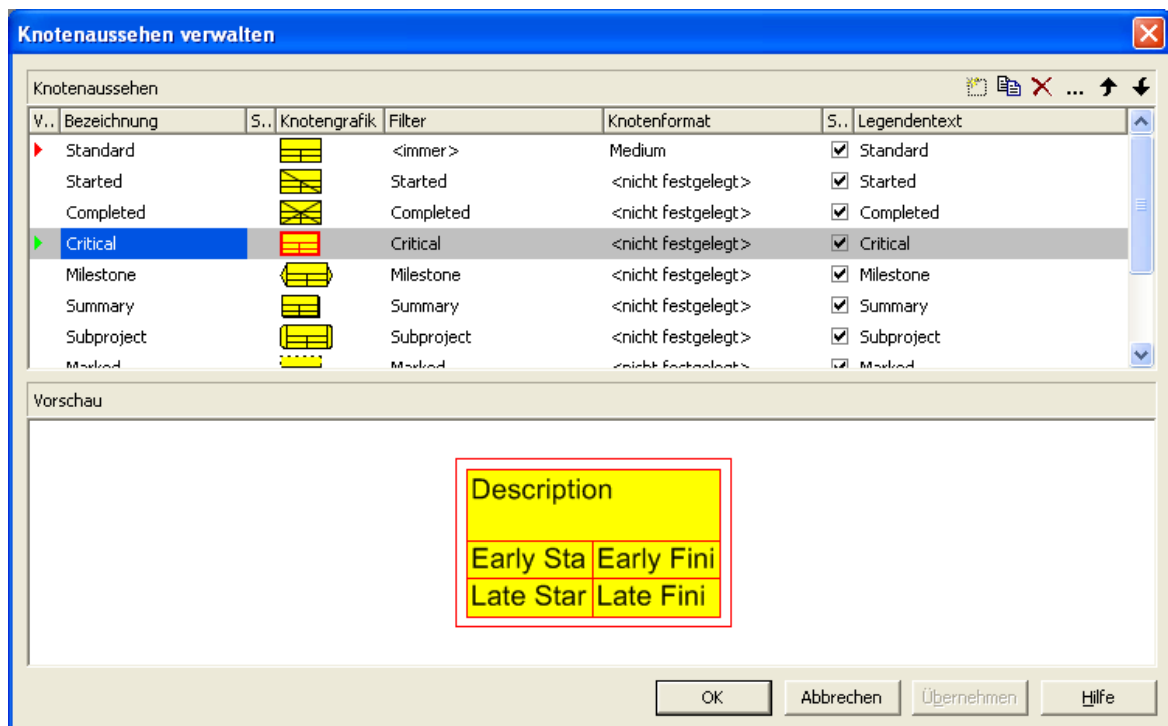
Unter **Und/Oder** können Sie eine Verknüpfung auswählen, wenn Sie mehrere Kriterien definieren möchten.

Definieren Sie den Filter wie in der Abbildung. Verlassen Sie dann das Dialogfeld **Filter bearbeiten** mit **OK**. Sie kehren in das Dialogfeld **Filter verwalten** zurück.

2.13 Knotenaussehen festlegen

VARCHART XNet bietet eine Vielzahl von Gestaltungsmöglichkeiten für das Aussehen von Knoten. Sie können das Aussehen von Knoten in Abhängigkeit von deren Daten festlegen. Beispielsweise können alle kritischen Knoten durch einen roten Hintergrund und eine doppelte schwarze Rahmenlinie gekennzeichnet werden, alle beendeten Knoten durch gekreuzte Linien etc. Diese grafischen Attribute werden als Knotenaussehen bezeichnet.

Öffnen Sie nun die Eigenschaftenseite **Objekte** und klicken Sie auf die **Knotenaussehen**-Schaltfläche. Sie gelangen in das Dialogfeld **Knotenaussehen verwalten**.




In der **Knotenaussehen**-Tabelle werden alle aktuell vorhandenen Knotenaussehen angezeigt. Markieren Sie ein Knotenaussehen nach dem anderen, um sie im Vorschauenfenster angezeigt zu bekommen.

Jedes Knotenaussehen ist mit einem Filter und einem Knotenformat verbunden. (Ausnahme: Das Knotenaussehen "Standard" ist nicht mit einem Filter verbunden.)

Der Filter gibt die Bedingungen an, unter denen ein Knoten dieses Knotenaussehen erhält. Beispielsweise ist das Knotenaussehen "Markiert" mit dem Filter "Markiert" verbunden. Dieser Filter selektiert alle markierten Knoten.

Erfüllt ein Vorgang die Filterkriterien mehrerer Knotenaussehen, werden diese Knotenaussehen für den Knoten grafisch überlagert. Begonnen wird dabei jeweils mit dem Knotenaussehen, das in der Liste ganz oben steht. Das Knotenaussehen, das ganz unten steht, wird als letztes zugewiesen und überlagert daher alle anderen.

Die niedrigste Position hat i. d. R. das Knotenaussehen "Standard", das normalerweise ganz oben in der Liste steht. Das Knotenaussehen "Standard" hat keinen Filter und wird auf alle Knoten angewendet.

  Sie können die Abarbeitungsreihenfolge der Knotenaussehen mit Hilfe der Pfeil-Schaltflächen verändern.

> **Knotenaussehen hinzufügen, kopieren, löschen und bearbeiten**

Sie können im Dialogfeld **Knotenaussehen verwalten** mit Hilfe der folgenden Schaltflächen Knotenaussehen hinzufügen, kopieren, löschen und bearbeiten:

 **Knotenaussehen hinzufügen**

 **Knotenaussehen kopieren**

 **Knotenaussehen löschen**

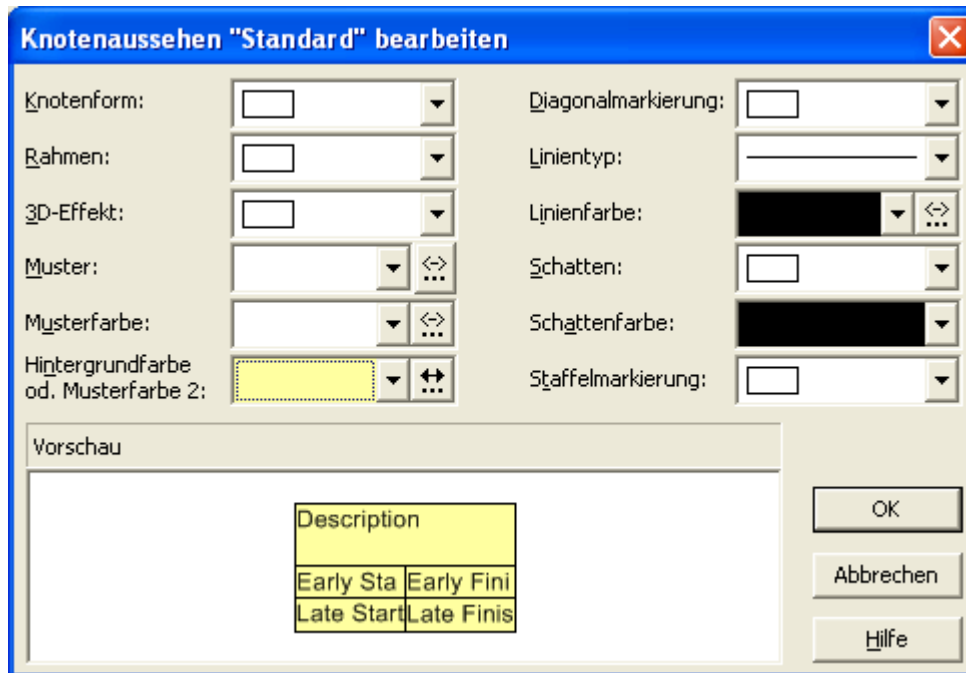
 **Knotenaussehen bearbeiten**

Hinweis: Alle Knotenaussehen außer den Standard-Knotenaussehen können gelöscht werden. Bevor das markierte Knotenaussehen tatsächlich gelöscht wird, erfolgt eine Rückfrage des Programms.

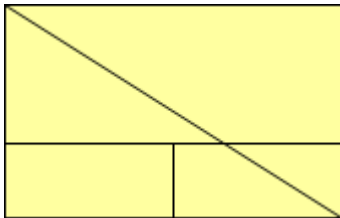
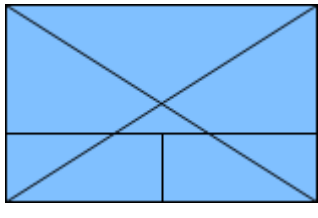
> **Knotenaussehen und Filter verwenden**

Lernen Sie nun den Umgang mit Knotenaussehen und Filtern an einem Beispiel kennen. Weisen Sie dazu dem Knotenaussehen "Abgeschlossen" die höchste und dem Knotenaussehen "Begonnen" die zweithöchste Priorität zu.

Bearbeiten Sie nun diese beiden Knotenaussehen. Markieren Sie das zu bearbeitende Knotenaussehen im Dialogfeld **Knotenaussehen verwalten** und klicken Sie auf die **Knotenaussehen bearbeiten**-Schaltfläche. Sie gelangen in das Dialogfeld **Knotenaussehen bearbeiten**. In der Kopfzeile wird der Name des aktuellen Knotenaussehens angezeigt. In diesem Dialogfeld können Sie die grafischen Attribute der Knoten festlegen. Außerdem können Sie hier festlegen, mit welchem Knotenformat und mit welchem Filter das Knotenaussehen kombiniert werden soll.



Legen Sie nun folgendes fest:

Knotenaussehen	Begonnen	Abgeschlossen
Filter	Begonnen	Abgeschlossen
Filterbedingung	fertig (%) größer als 0 und kleiner als 100	fertig (%) = 100
Hintergrundfarbe	rot	blau
Diagonalmarkierung	abwärts	gekreuzt
Aussehen		

Bestätigen Sie die Einstellungen mit **OK** und starten Sie das Programm. Erzeugen Sie einen Knoten, klicken Sie ihn doppelt an und bearbeiten Sie seine Daten im Dialogfeld **Vorgänge bearbeiten** in den folgenden drei Schritten:

- Geben Sie für "fertig (%)" den Wert "0" an: Der Knoten erscheint mit dem Standard-Knotenaussehen.
- Geben Sie als nächstes für "fertig (%)" eine Zahl größer als Null, aber kleiner als 100 an: Der Knoten erscheint mit dem "Begonnen"-Knotenaussehen, also in hellgelb und abwärts durchgestrichen.

- Geben Sie schließlich für "fertig (%)" den Wert "100" an: Der Knoten erscheint mit dem Knotenaussehen "Abgeschlossen", also in blau und mit gekreuzten Linien.

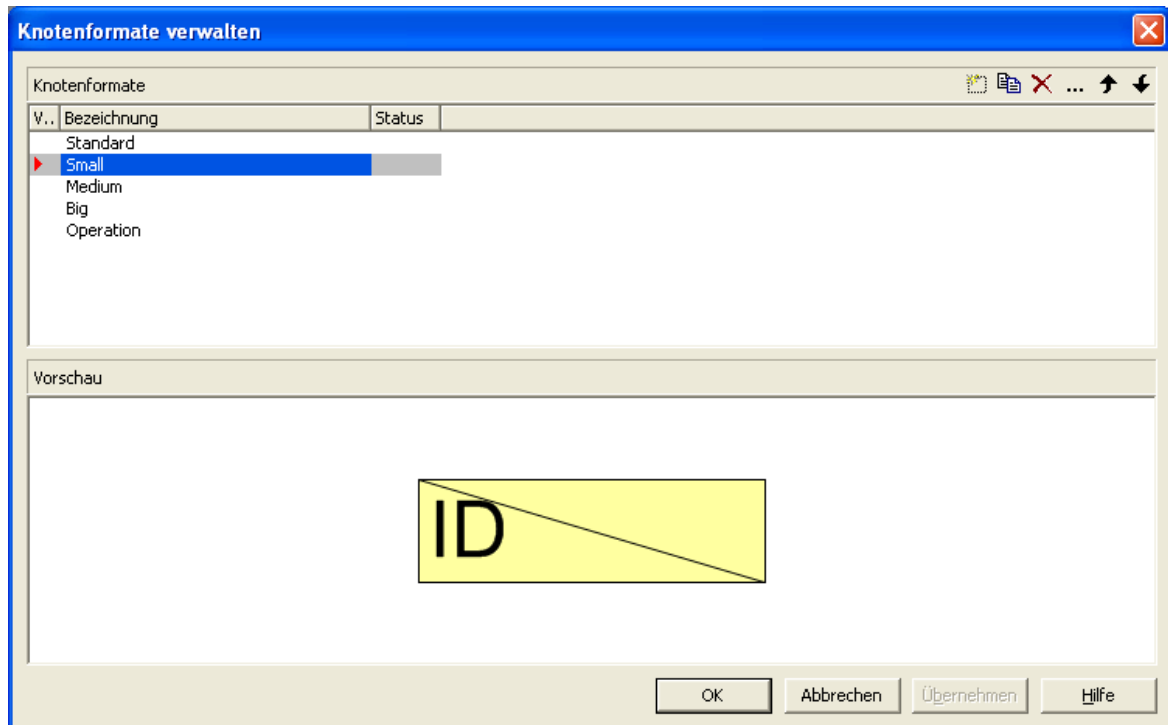
> **Knotenaussehen datenabhängig festlegen**

Die Hintergrundfarbe und die Linienfarbe eines Knotenaussehens können Sie mit Hilfe von Zuordnungstabellen in Abhängigkeit von den Daten der Knoten festlegen. Siehe hierzu das Kapitel "Wichtige Begriffe: Zuordnungstabellen".

2.14 Knotenformate festlegen





Jedes Knotenaussehen ist mit einem Knotenformat verbunden. Sie können die Knotenformate selbst festlegen.

Klicken Sie auf der Eigenschaftenseite **Objekte** auf die Schaltfläche **Knotenformate**. Sie gelangen dann in das Dialogfeld **Knotenformate verwalten**.



In der **Knotenformate**-Tabelle werden alle vorhandenen Knotenformate aufgeführt. Gehen Sie die Tabelle durch, um im Vorschaufenster das Aussehen des jeweils markierten Knotenformats zu betrachten.

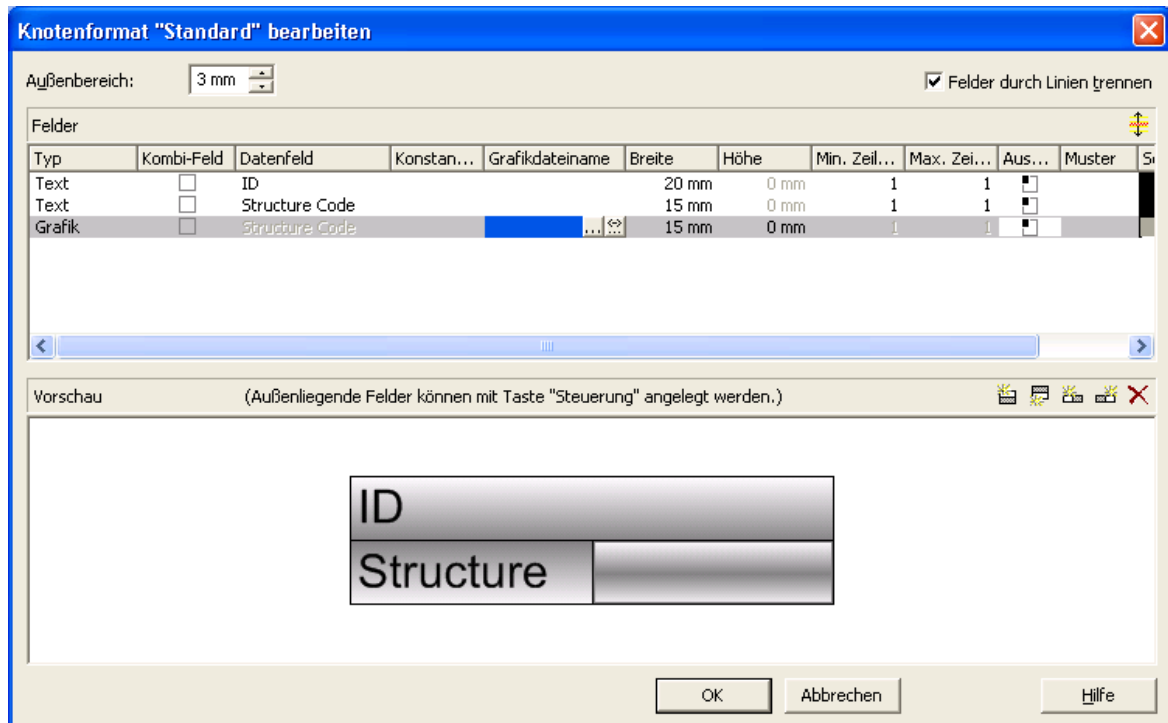
Sie können im Dialogfeld **Knotenformate verwalten** mit Hilfe der folgenden Schaltflächen Knotenformate hinzufügen, kopieren, löschen und bearbeiten:

-  **Knotenformat hinzufügen**
-  **Knotenformat kopieren**
-  **Knotenformat löschen**
-  **Knotenformat bearbeiten**

Hinweis: Das Knotenformat "Standard" sowie jedes in einem Knotenaussehen verwendete Knotenformat können nicht gelöscht werden.

> Knotenformat bearbeiten


Um ein Knotenformat zu bearbeiten, markieren Sie es und klicken Sie auf die Schaltfläche **Knotenformat bearbeiten**. Das folgende Dialogfeld erscheint:





In diesem Dialog können Sie für das gewählte Knotenformat Folgendes festlegen:

- ob die Knotenfelder durch Linien getrennt werden sollen
- den Außenbereich (Abstand in Millimetern, den Knoten mit diesem Knotenformat zu benachbarten Knoten und zum Rand der Darstellung halten)
- den Typ des aktuellen Knotenfeldes (Text oder Grafik)
- für den Typ Text: das Datenfeld, dessen Inhalt in dem aktuellen Knotenfeldes ausgegeben werden soll, oder einen konstanten Text
- für den Typ Grafik: Name und Pfad der Grafikdatei, die in dem gewählten Knotenfeld dargestellt wird
- die Breite und Höhe des markierten Knotenfeldes
- die maximale Anzahl von Textzeilen im aktuellen Knotenfeld
- die Ausrichtung des Textes bzw. der Grafik des markierten Knotenfeldes
- die Hintergrundfarbe des Knotenfeldes
- das Füllmuster des Knotenfeldes
- die Schriftart und -farbe des Knotenfeldes

> **Darstellung von Grafiken in Knotenfeldern**

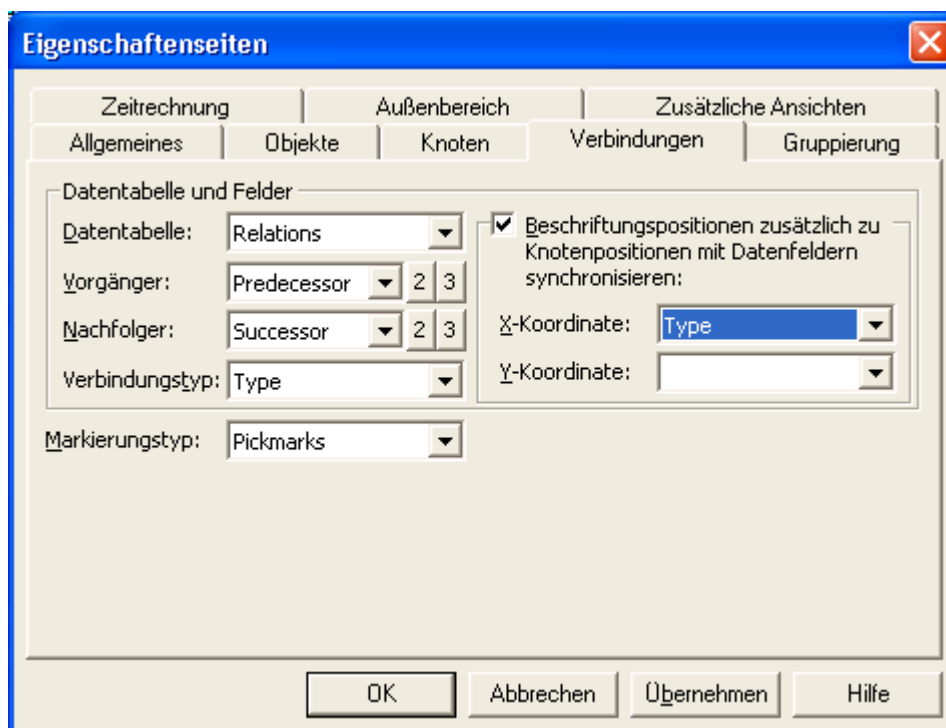
Sie können für ein Knotenfeld des Typs Grafik eine Grafikdatei wählen, indem Sie auf die Schaltfläche **Grafikdatei auswählen** () klicken und dann im gleichnamigen Windows-Dialog eine Datei wählen.

Sie können aber auch eine Zuordnung zwischen den Einträgen eines Datenfeldes und Grafikdateien herstellen. Klicken Sie dazu auf die Schaltfläche **Zuordnungen einstellen** () , um den gleichnamigen Dialog zu öffnen. Wenn eine Zuordnung vorgenommen worden ist, wird das durch ein Symbol neben dem Grafikdateinamen dargestellt () . Einzelheiten hierzu finden Sie in den Kapiteln "Eigenschaftenseiten und Dialogfelder" und "Wichtige Begriffe: Zuordnungstabellen".

2.15 Das Aussehen von Verbindungen festlegen

Sie haben in den letzten Abschnitten schon die Verwendung von Filtern für Knoten kennengelernt. Filter werden nicht nur für die Festlegung des Knotenaussehens, sondern auch des Verbindungsaussehens verwendet. Sie können beispielsweise ein jeweils unterschiedliches Aussehen der Verbindungen für unterschiedliche Verbindungstypen (z.B. Start-Start-, Ende-Ende-Verbindungen etc.) festlegen.

Um das Aussehen von Verbindungen festzulegen, öffnen Sie die Eigenschaftenseite **Verbindungen**.



In der Tabelle **Aussehen** enthält jede Zeile den Namen, den Filter und die Linienart eines Verbindungsaussehens.

Die letzte Zeile enthält den Eintrag "Neu...". Klicken Sie zweimal auf diesen Eintrag und definieren Sie das neue Verbindungsaussehen "FF-Verbindung". Sobald Sie in ein anderes Feld klicken, wird dieses Verbindungsaussehen der Liste hinzugefügt und ein neuer "Neu"-Eintrag erzeugt. Für das neu definierte Verbindungsaussehen werden zunächst derselbe Filter und dieselben Linienattribute wie für das letzte Verbindungsaussehen verwendet.

Klicken Sie nun auf das **Filter**-Feld des zu bearbeitenden Verbindungsaussehens. Eine Pfeil- und eine **Bearbeiten**-Schaltfläche erscheinen. Klicken Sie auf die Pfeil-Schaltfläche neben dem Filternamen, um alle verfügbaren Filter

für Verbindungsausssehen in der Kombobox angezeigt zu bekommen. Bis jetzt existiert nur ein Filter für Verbindungsausssehen ("LinkStandard").

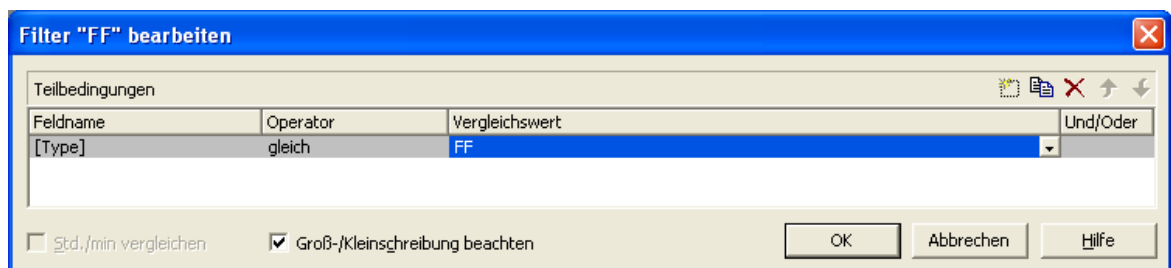
Definieren Sie nun einen neuen Filter für Verbindungsausssehen. Klicken Sie dazu auf die **Bearbeiten**-Schaltfläche des **Filter**-Feldes. Das Dialogfeld **Filter verwalten** öffnet sich, in dem Sie Filter hinzufügen, kopieren, bearbeiten und löschen können. Klicken Sie hier auf die **Filter hinzufügen**-Schaltfläche, um einen neuen Filter anzulegen. Benennen Sie diesen Filter um in "FF". Dieser Filter soll alle Ende-Ende-Verbindungen (FF) auswählen.

Bearbeiten Sie nun den Filter "FF". Klicken Sie nun auf die Schaltfläche **Filter bearbeiten**, um das gleichnamige Dialogfeld zu öffnen.

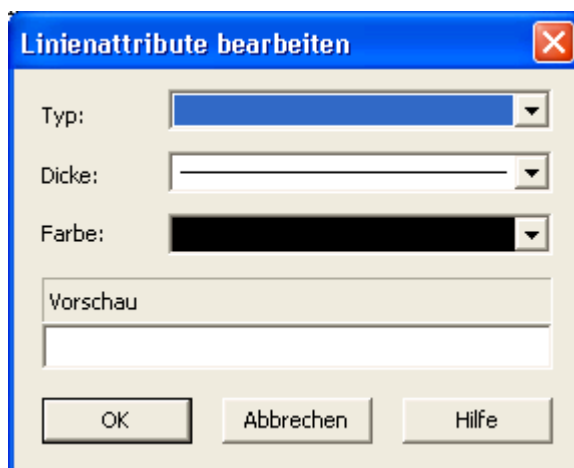
Definieren Sie die Filterbedingung "Verbindungstyp gleich FF". Damit werden alle Verbindungen vom Typ Ende-Ende (FF) mit dem als "FF-Verbindung" definierten Verbindungsausssehen dargestellt.

Beachten Sie bitte, dass alle Änderungen, die Sie an einem Filter durchführen, für diesen Filter allgemein und nicht nur für das aktuelle Verbindungsausssehen gelten.

Klicken Sie auf **OK** und anschließend im Dialog **Filter verwalten** nochmal auf **OK**.

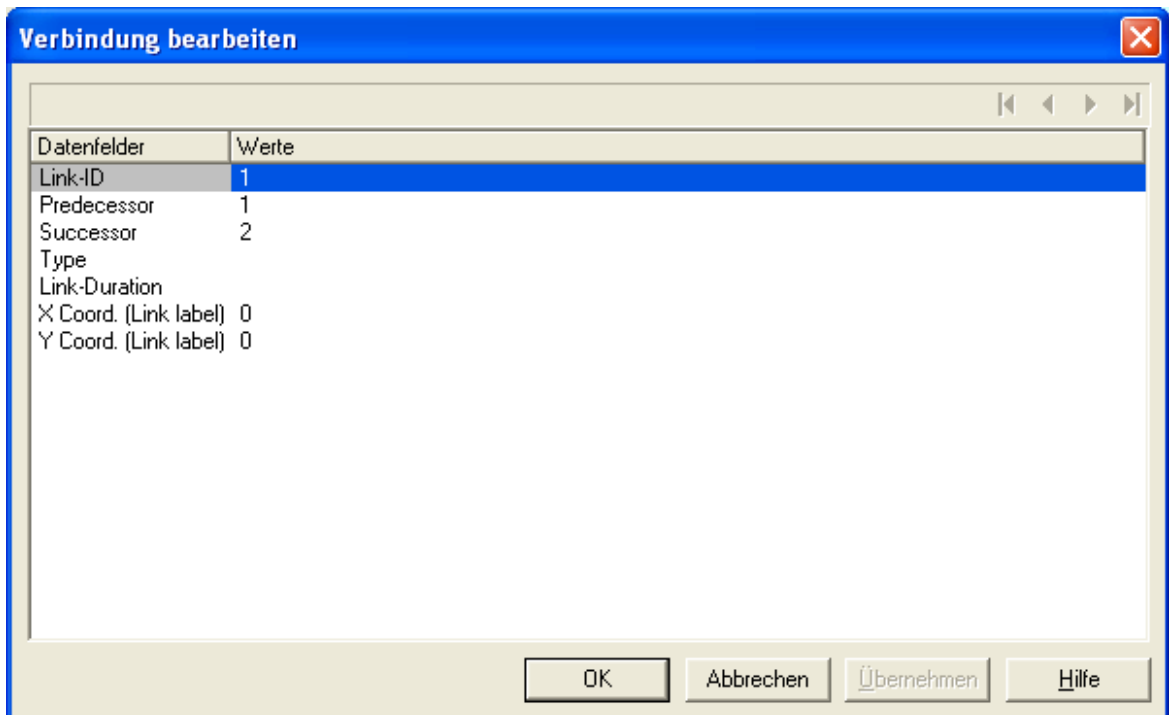


Legen Sie schließlich noch die Linienattribute für das Verbindungsausssehen "FF-Verbindung" fest. Klicken Sie dazu auf den **Linienart**-Eintrag dieses Feldes. Das Dialogfeld **Linienattribute bearbeiten** öffnet sich, in dem Sie das Aussehen der Verbindungslinien festlegen können.

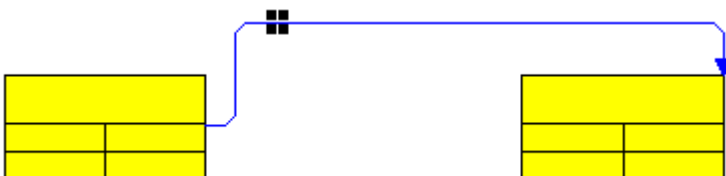


Wählen Sie eine blaue Linie für die Verbindungen vom Typ "FF-Verbindung", also für die Ende-Ende-Verbindungen.

Starten Sie nun das Programm mit F5 und erzeugen Sie zwei Knoten mit einer Verbindung. Klicken Sie mit der rechten Maustaste auf die Verbindung und wählen Sie den Befehl **Bearbeiten** aus dem Kontextmenü, um das Dialogfeld **Verbindung bearbeiten** aufzurufen.

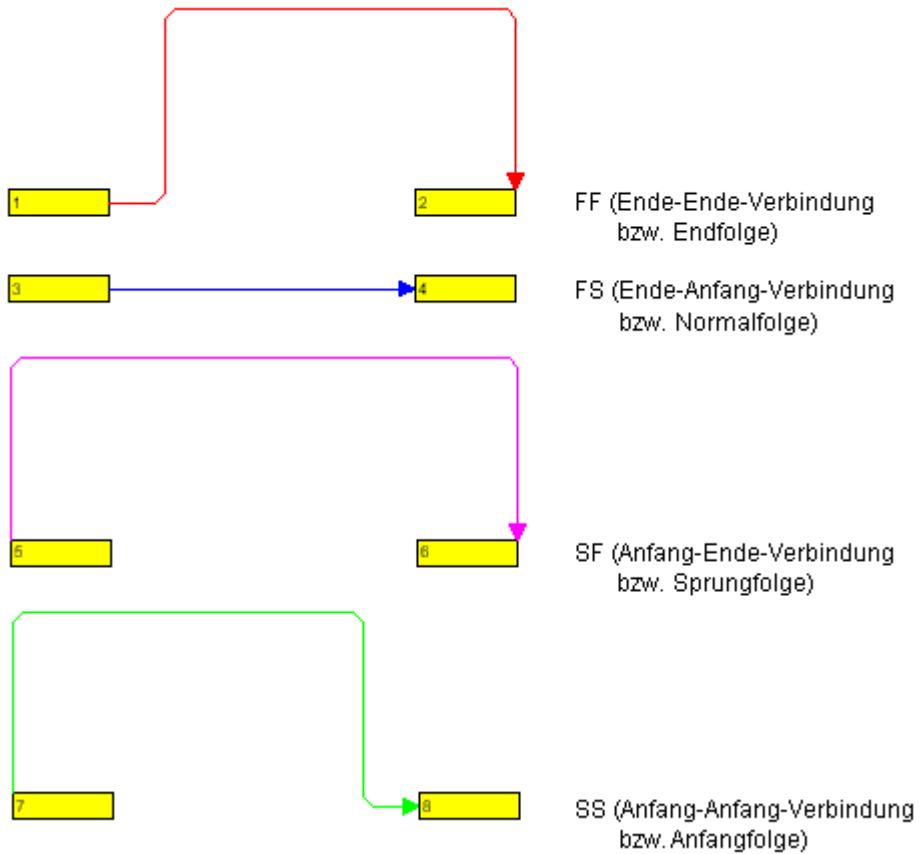


Geben Sie unter **Typ** nun "FF" ein und bestätigen Sie mit **OK**. Dann wird die markierte Verbindung zu einer Ende-Ende-Verbindung, die mit einer blauen Linie dargestellt wird.

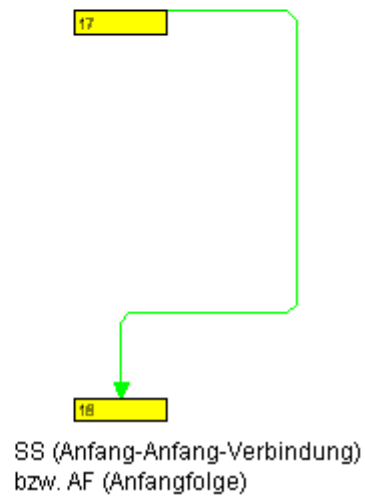
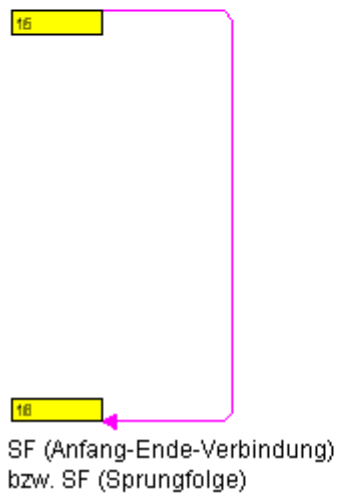
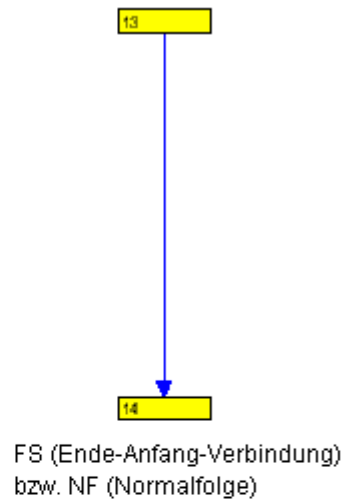
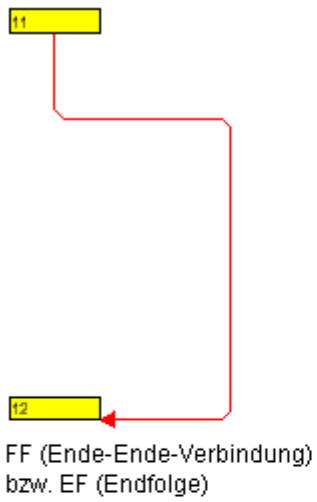


Eine Übersicht über das Aussehen der verschiedenen Verbindungstypen bei den beiden Flussrichtungen geben die folgenden Abbildungen:

64 Das Aussehen von Verbindungen festlegen



Flussrichtung von links nach rechts



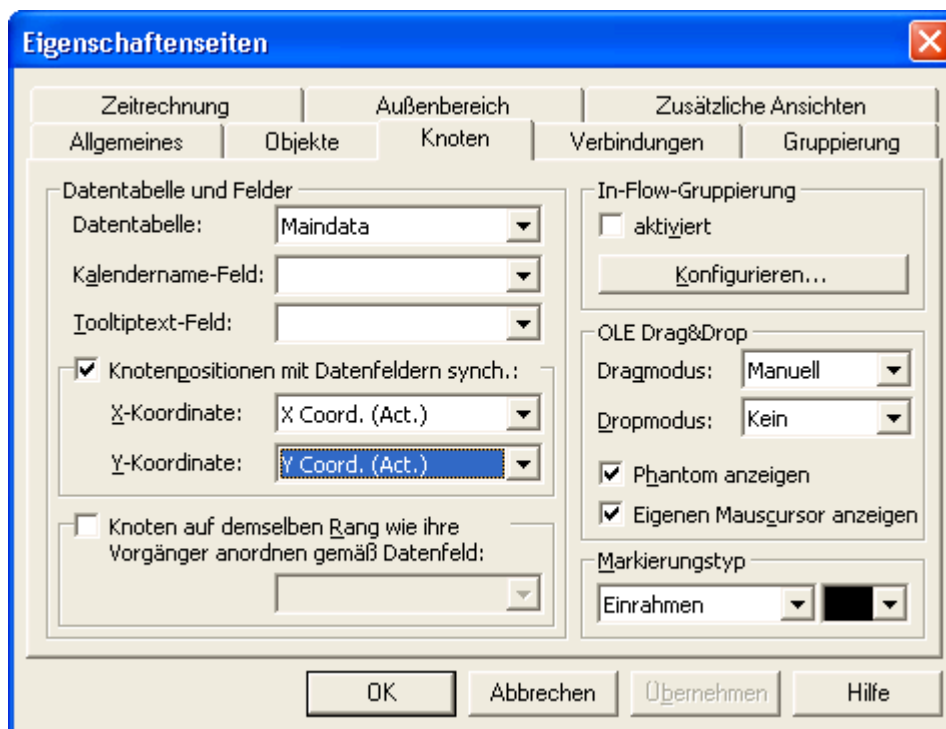
Flussrichtung von oben nach unten

2.16 Positionen von Knoten und Verbindungsbeschriftungen speichern

Damit die Positionen von Knoten und Verbindungsbeschriftungen nach dem Schließen Ihres Projekts wiederhergestellt werden können, müssen diese in bestimmten Datenfeldern gespeichert werden.

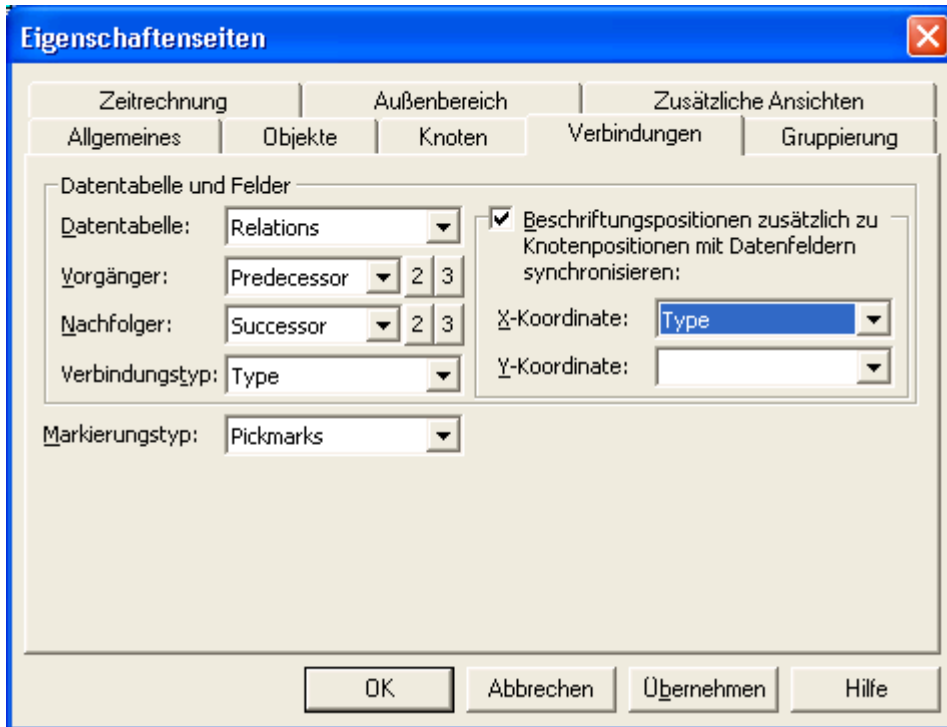
Um die Positionen der Knoten mit den entsprechenden Datenfeldern zu synchronisieren, aktivieren Sie auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Knotenpositionen mit Datenfeldern synchronisieren** und geben Sie folgende Datenfelder an:

- für die X-Koordinate: "X-Koordinate (Knoten)"
- für die Y-Koordinate: "Y-Koordinate (Knoten)"



Und um die Positionen der Verbindungsbeschriftungen mit den entsprechenden Datenfeldern zu synchronisieren, aktivieren Sie auf der Eigenschaftenseite **Verbindungen** das Kontrollkästchen **Positionen der Beschriftungen mit Datenfeldern synchronisieren** und geben Sie folgende Datenfelder an:

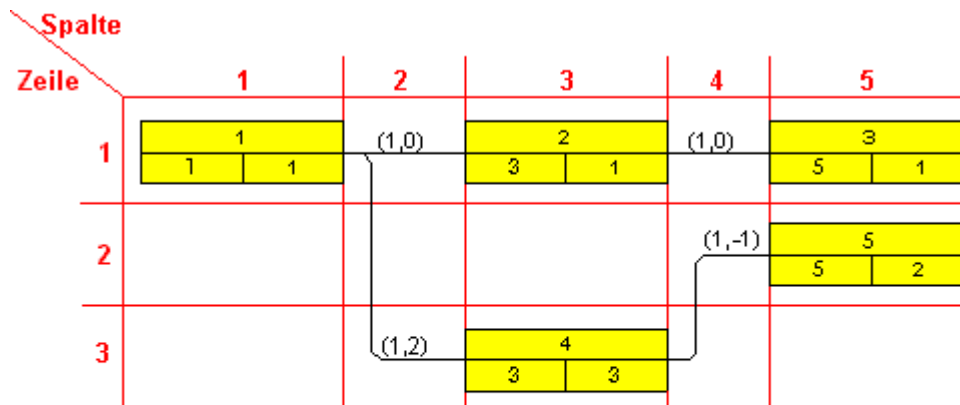
- für die X-Koordinate: "X-Koordinate (Verbindung)"
- für die Y-Koordinate: "Y-Koordinate (Verbindung)"



Die Positionen von Knoten und Verbindungsbeschriftungen im Diagramm werden als deren Koordinaten in einem Raster festgelegt.

- Die X- und Y-Koordinate eines Knotens geben die absolute Position dieses Knotens im Raster an.
- Die X- und Y-Koordinate einer Verbindungsbeschriftung geben die Position dieser Verbindungsbeschriftung relativ zu ihrem Vorgängerknoten an.

Das erste Feld des Rasters (ganz links oben) ist definiert als $(X,Y) = (1,1)$. Es ist für Knoten reserviert. Von links nach rechts und von oben nach unten wird jeweils in Schritten von 1 weitergezählt. In allen weiteren Feldern können Knoten oder Verbindungsbeschriftungen stehen. Knoten haben daher immer positive X- und Y-Koordinaten. Bei Verbindungsbeschriftungen sind alle Wertepaare außer $(0,0)$ möglich.

**Legende:**

Nummer		Knotenfeld
X-Position	Y-Position	

(x,y) Verbindungsbeschriftungsposition

Damit Sie die angelegten Knoten und Verbindungen sichern und wieder laden können, ergänzen Sie für das Ereignis FormClosing:

Code-Beispiel

```
Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
    VcNet1.SaveAsEx("C:\test.csv", VcEncoding.vcUnicodeEncoding)
End Sub
```

Code-Beispiel

```
private void Clustering_FormClosing(object sender, FormClosingEventArgs
e)
{
    vcNet1.SaveAsEx(@"...", VcEncoding.vcUnicodeEncoding);
}
```

2.17 Hilfsknoten übersichtlich anordnen

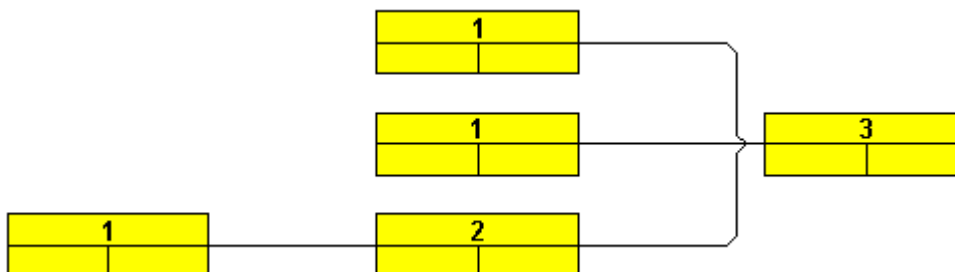
In vielen Fällen ist es sinnvoll, Hilfsknoten nicht in Flussrichtung anzuordnen, sondern entgegen der Flussrichtung unter- bzw. oberhalb ihrer Vorgänger (bei Flussrichtung von links nach rechts) bzw. neben ihren Vorgängern (bei Flussrichtung von oben nach unten). Dies ist in VARCHART XNet möglich, da es erlaubt, den Rang von Knoten zu reduzieren.

Unter dem Rang eines Knotens versteht man eine Zahl, die folgendermaßen definiert ist: Der Rang eines Knotens ohne Vorgänger ist 1. Der Rang eines Knotens mit Vorgängern ist gleich 1 plus Rang desjenigen seiner Vorgänger, der den höchsten Rang besitzt.

Durch diese Definition wird verhindert, dass in Netzdiagrammen Ringbeziehungen vorkommen.

Einige Beispiele:

- Der Rang eines Knotens mit einem Vorgänger, der keinen Vorgänger besitzt, ist $1+1=2$.
- Der Rang eines Knotens, der drei Vorgänger mit den Rängen 1, 1 bzw. 2 hat, ist $1+2=3$ (siehe Abbildung).



Ränge der Knoten bei Flussrichtung von links nach rechts

Den Rang von Knoten kann man sich folgendermaßen veranschaulichen:

- Bei Flussrichtung von links nach rechts entspricht der höchste Rang von allen Knoten in einer Diagrammspalte der Nummer dieser Diagrammspalte (gezählt werden hierbei nur die Diagrammspalten für Knoten, nicht die für Verbindungsbeschriftungen).
- Bei Flussrichtung von oben nach unten entspricht der höchste Rang von allen Knoten in einer Diagrammzeile der Nummer dieser Diagrammzeile (gezählt werden hierbei nur die Diagrammzeilen für Knoten, nicht die für Verbindungsbeschriftungen).

Ränge werden nur beim Aufruf von **Anordnen** (Kontextmenü für das Diagramm) berechnet. Sie dienen dem Layout-Algorithmus als Vorgabe für

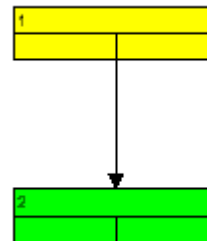
die Positionierung der Knoten in Flussrichtung. Bei Ringbeziehungen (Schleifen) zwischen Knoten wählt VARCHART XNet über einen eigenen Algorithmus Verbindungen aus, die bei der Rangberechnung vorübergehend ignoriert werden. Die ignorierten Verbindungen sind nach dem Layout als rückführende Verbindungen sichtbar. Dabei wird darauf geachtet, dass das Layout möglichst wenige Veränderungen gegenüber einem Layout ohne diese rückführenden Verbindungen aufweist.

In manchen Anwendungsfällen ist es sinnvoll, einen Knoten auf demselben Rang wie seinen Vorgänger zu plazieren, z. B. wenn dieser Knoten einen Hilfsknoten zu seinem Vorgänger darstellt. Der Rang eines solchen Knotens kann dazu um 1 reduziert werden.

Bei Flussrichtung von links nach rechts steht der Hilfsknoten, dessen Rang um 1 reduziert wurde, unter oder über seinem Vorgänger statt daneben.

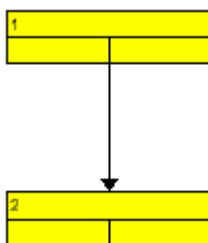


Zwei Knoten mit Rang 1 bzw. 2

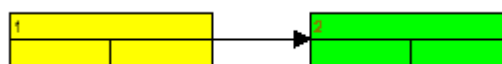


*Der Rang des 2. Knotens wurde um 1 reduziert. Anschließend wurde der Befehl **Anordnen** aufgerufen.*

Bei Flussrichtung von oben nach unten steht der Hilfsknoten, dessen Rang um 1 reduziert wurde, links oder rechts neben seinem Vorgänger statt darunter.



Zwei Knoten mit Rang 1 bzw. 2



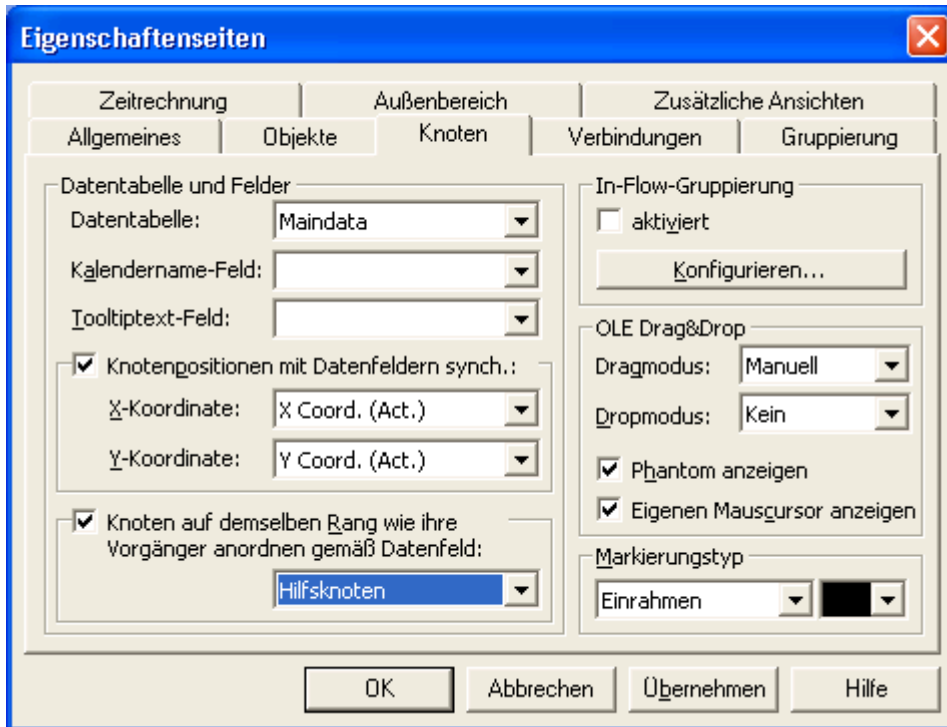
*Der Rang des 2. Knotens wurde um 1 reduziert. Anschließend wurde der Befehl **Anordnen** aufgerufen.*

Um den Rang eines Knotens reduzieren zu können, ist das Datenfeld "Hilfsknoten" vorgesehen. Der Eintrag in diesem Datenfeld legt fest, wie der Hilfsknoten positioniert wird. Mögliche Werte sind 0, 1, 2, 3.

Wert im Feld "Hilfsknoten"	Flussrichtung von oben nach unten	Flussrichtung von links nach rechts
0	Der Rang des Hilfsknotens wird nicht reduziert.	Der Rang des Hilfsknotens wird nicht reduziert.
1	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links oder rechts neben seinem Vorgänger statt darunter.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über oder unter seinem Vorgänger statt daneben.
2	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links von seinem Vorgänger.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über seinem Vorgänger.
3	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann rechts von seinem Vorgänger.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann unter seinem Vorgänger.

Lernen Sie nun an einem praktischen Beispiel kennen, wie Sie Hilfsknoten auf demselben Rang wie ihre Vorgänger anordnen können.

Wählen Sie auf der Eigenschaftenseite **Allgemeines** die **Flussrichtung** von links nach rechts. Kreuzen Sie dann auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Knoten auf demselben Rang wie ihre Vorgänger anordnen gemäß Datenfeld** an. Wählen Sie aus der Kombobox das Datenfeld "Hilfsknoten" aus. Von dem Wert eines Knotens im Datenfeld "Hilfsknoten" hängt dann ab, ob dieser Knoten auf demselben Rang positioniert wird wie sein Vorgänger.



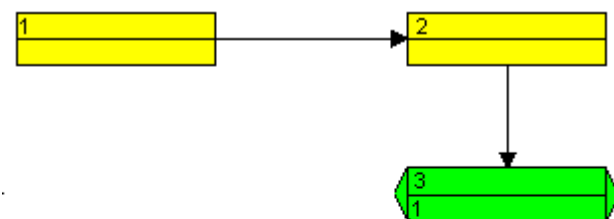
Starten Sie mit diesen Einstellungen das Programm, erzeugen Sie einige Knoten und verbinden Sie diese miteinander wie in der folgenden Abbildung:



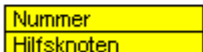
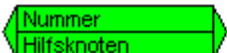
Doppelklicken Sie nun auf den 3. Knoten und geben Sie in dem **Vorgänge bearbeiten**-Dialogfeld unter "Hilfsknoten" den Wert 1 an. Sie erhalten dann die folgende Darstellung:



Öffnen Sie das Kontextmenü für das Diagramm und wählen Sie den Befehl **Anordnen**. Das Resultat sieht folgendermaßen aus:

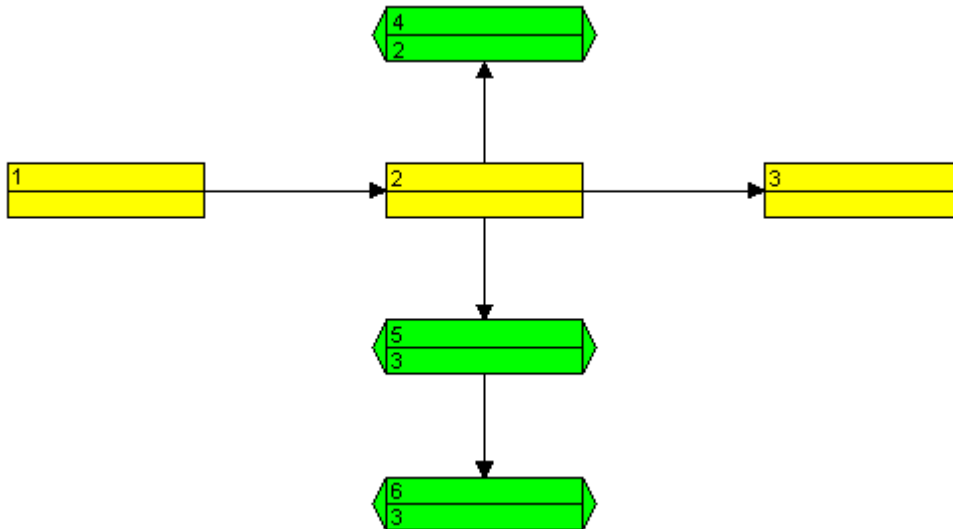


Legende:

	Standard
	Hilfsknoten

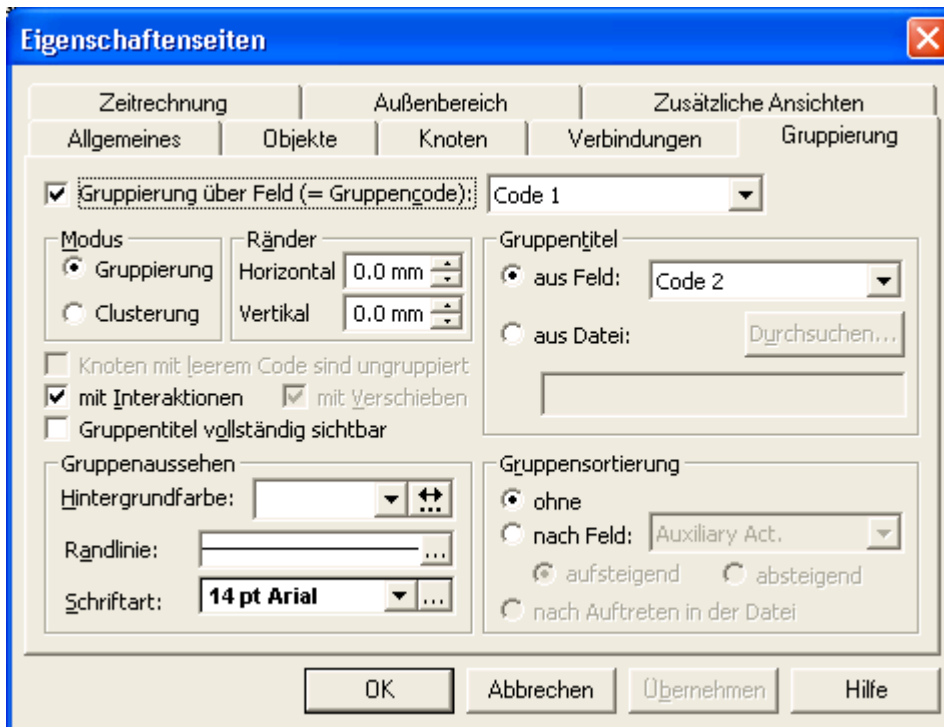
Der Knoten, dessen Rang reduziert worden ist, wird nun unterhalb statt rechts von seinem Vorgänger angeordnet.

Die folgende Abbildung zeigt eine Anordnung mit unterschiedlichen Hilfsknoten (Flussrichtung: von links nach rechts):



2.18 Knoten gruppieren

In vielen Fällen lassen sich Netzpläne übersichtlicher gestalten, indem Knoten gruppiert angeordnet werden. Die Gruppierung können Sie auf der Eigenschaftenseite **Gruppierung** festlegen.



Legen Sie zunächst unter **Gruppierung über Feld (= Gruppencode)** fest, nach welchem Feld die Gruppierung vorgenommen werden soll. Das Feld, das Sie hier auswählen, wird als **Gruppencode** bezeichnet. Wählen Sie für das Tutorium das Feld "Gruppencode". Alle Knoten, die denselben Eintrag für "Gruppencode" haben, werden so in einer Gruppe zusammengefasst.

Für die Beschriftung der Gruppen können Sie zwischen zwei Alternativen wählen: Entweder werden die Gruppentitel aus einem Feld Ihrer Wahl (**aus Feld**) oder aus einer Datei (**aus Datei**) genommen.

Aktivieren Sie für das Tutorium das Kontrollkästchen **aus Feld** und wählen Sie das Feld "Gruppenname". Das Feld, das Sie für den Gruppentitel wählen, muss nicht notwendigerweise mit dem Gruppencode (= Feld, nach dem gruppiert wird) identisch sein. Doch damit die Gruppen sinnvoll beschriftet werden, sollten die Einträge im **Gruppencode**- und im **Gruppentitel**-Feld miteinander korrespondieren.

Verwenden Sie für das Tutorium die folgende Tabelle, wenn Sie zur Laufzeit Knoten erzeugen und bearbeiten:

Gruppencode	Gruppenname
A	Anlagenplanung
B	Berechnung Gesamtanlage
C	Detailkonstruktion

Legen Sie nun fest, ob und ggf. nach welchem Kriterium die Gruppen sortiert werden. Gruppen können unsortiert bleiben oder sortiert werden, entweder nach einem Datenfeld Ihrer Wahl oder nach ihrem Auftreten in der Datei.

Aktivieren Sie unter **Gruppensortierung** die Option **nach Feld**, wählen Sie das Datenfeld "Gruppencode" und die Option **absteigend**. Dadurch werden die Gruppen in absteigender Reihenfolge nach ihrem Gruppencode sortiert.

Im Bereich **Gruppenaussehen** können Sie die Farbe, die Dicke und den Typ der Linien, mit denen die einzelnen Gruppen umrandet werden, sowie die Hintergrundfarbe der Gruppen und die Schriftattribute der Gruppentitel festlegen. Gestalten Sie das Gruppenaussehen nach Ihren Vorstellungen und starten Sie anschließend das Programm.

Legen Sie nun einige Knoten an und tragen Sie dafür jeweils Werte in die Datenfelder "Gruppencode" und "Gruppenname" ein. Beachten Sie dabei die Korrespondenz zwischen diesen Datenfeldern.

Sie erhalten eine Darstellung wie die folgende:

Anlagenplanung	
4	5
A	A
Berechnung Gesamtanlage	
1	2
B	B
Detail konstruktion	
3	
C	

Verschieben Sie nun einen Vorgang von einer Gruppe in eine andere. Der Wert des als Gruppencode vereinbarten Datenfeldes (hier "Gruppencode") wird dabei automatisch angepasst. (Sie können das überprüfen, indem Sie das **Vorgänge bearbeiten**-Dialogfeld für den verschobenen Vorgang aufrufen.)

2.19 Zeitrechnung des VARCHART XNet festlegen

Mit dem Scheduler von VARCHART XNet 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 XNet an Ihre Schnittstelle anpassen, indem Sie festlegen, welche Datenfelder für die Eingabe (**Zeitrechnungseingabe**) und die Ausgabe (**Zeitrechnungsergebnis**) des Schedulers verwendet werden sollen. Außerdem können Sie die Zeiteinheit festlegen, die für die Berechnung der Dauer in den entsprechenden Datenfeldern in Knoten und Verbindungen verwendet werden soll.



Wählen Sie unter **Zeitrechnungseingabe** für jede Eingabe aus, aus welchem Feld sie entnommen werden soll. Verwenden Sie für dieses Tutorium dieselben Festlegungen wie in der Abbildung.

Als Eingaben für die Zeitrechnung verwendet der Scheduler Datenfelder der Maindata- und der Relations-Tabelle. Die Grundlage der Berechnung sind die Dauer der einzelnen Vorgänge, deren logische Abhängigkeiten und der Projektanfang. Die Felder **Vorgänger**, **Nachfolger** und **Verbindungstyp** können in der **Zeitrechnungseingabe**-Tabelle nicht bearbeitet werden. Sie geben nur die auf den Eigenschaftenseiten **Verbindungen** vorgenommenen Einstellungen wieder.

Wählen Sie unter **Zeitrechnungsergebnis** für jede Ausgabe aus, in welches Feld sie geschrieben werden soll. Die Ausgabe des Schedulers erfolgt nur in Datenfelder der Maindata-Tabelle. Die Ausgaben werden wiederum in Datenfelder der Schnittstelle geschrieben. Als Ausgaben stehen zur Verfügung: **früh. Start, früh. Ende, spät. Start, spät. Ende, Freier Puffer** und **Gesamtpuffer**. Weisen Sie jeder dieser Ausgaben ein Datenfeld aus der in der Datendefinition vereinbarten Liste von Feldern zu (wie in der Abbildung).

Es gibt folgende Möglichkeiten, die Zeitrechnung zu beeinflussen:

1. Sie können einen Projektstart vorgeben. Das erreichen Sie über die API mit der VcNet-Methode **ScheduleProject**:

VcNet1.ScheduleProject "04.05.2000", 0

Mit der Methode **ScheduleProject** können Sie eine Vorwärts- und Rückwärtsberechnung des aktuellen Projekts durchführen. Bei Übergabe des Starttermins wird zunächst eine Vorwärtsberechnung, dann eine Rückwärtsberechnung durchgeführt. Bei Übergabe des Endtermins wird zunächst eine Rückwärtsberechnung, dann eine Vorwärtsberechnung durchgeführt. Es können auch Anfangs- und Enddatum übergeben werden, die Vorgänge erhalten dann entsprechende Pufferzeiten.

Mögliche Wahl der Parameter für die Methode ScheduleProject:

Anfang	Ende
Termin 1	0
0	Termin 2
Termin 1	Termin 2

2. Sie können aktuelle Start- bzw. Endtermine angeben. Die Knoten sind dann unverrückbar.
3. 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. Dann liegt der früheste Start eines Knoten nicht vor dem angegebenen Termin bzw. das späteste Ende nicht nach dem angegebenen Termin.

Probieren Sie nun die Zeitrechnung aus. Definieren Sie dazu drei Schaltflächen ("Command1", "Command2" und "Command3"), um die Zeitrechnung zur Laufzeit ausführen zu lassen. Benennen Sie die

Schaltflächen mit "Projektstart", "Projektende" und "Projektanfang und -ende" und schreiben Sie dafür den folgenden Code:

Code-Beispiel

```
Private Sub Command1_Click()
VcNet1.ScheduleProject "01.01.2000", 0
End Sub

Private Sub Command2_Click()
VcNet1.ScheduleProject 0, "01.02.2000"
End Sub

Private Sub Command3_Click()
VcNet1.ScheduleProject "01.01.2000", "01.02.2000"
End Sub
```

Geben Sie nun die folgenden Codezeilen ein, damit beim Starten des Programms einige Knoten und Verbindungen geladen werden.

Code-Beispiel

```
Private Sub Form_Load()

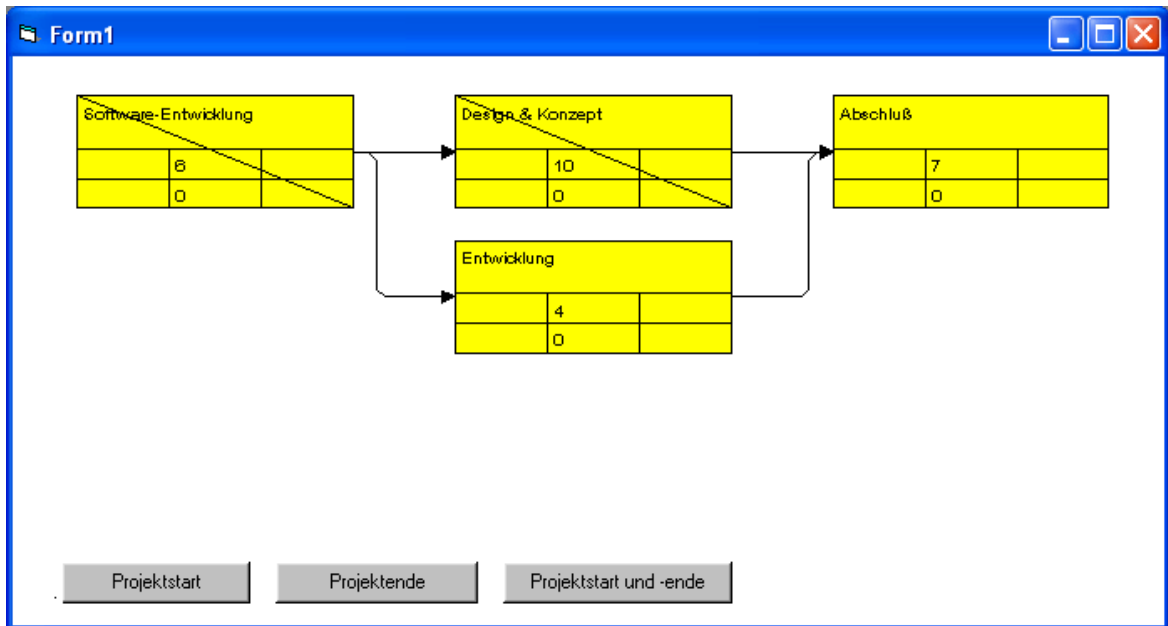
    VcNet1.InsertNodeRecord ("1;1.;;;Software-Entwicklung;A;; _
                             Gruppe A;6;0;10;;;0;")
    VcNet1.InsertNodeRecord ("2;1.2;;;Design & Konzept;C;; _
                             Gruppe A;10;0;50;;;0;")
    VcNet1.InsertNodeRecord ("3;1.2.2;;;B;; _
                             Gruppe A;7;0;0;;;0;")
    VcNet1.InsertNodeRecord ("4;1.2.4;;;Entwicklung;B;; _
                             Gruppe A;4;0;0;;;0;")

    VcNet1.InsertLinkRecord ("1;1;2;;;")
    VcNet1.InsertLinkRecord ("2;1;4;;;")
    VcNet1.InsertLinkRecord ("3;2;3;;;")
    VcNet1.InsertLinkRecord ("4;4;3;;;")

    VcNet1.EndLoading

End Sub
```

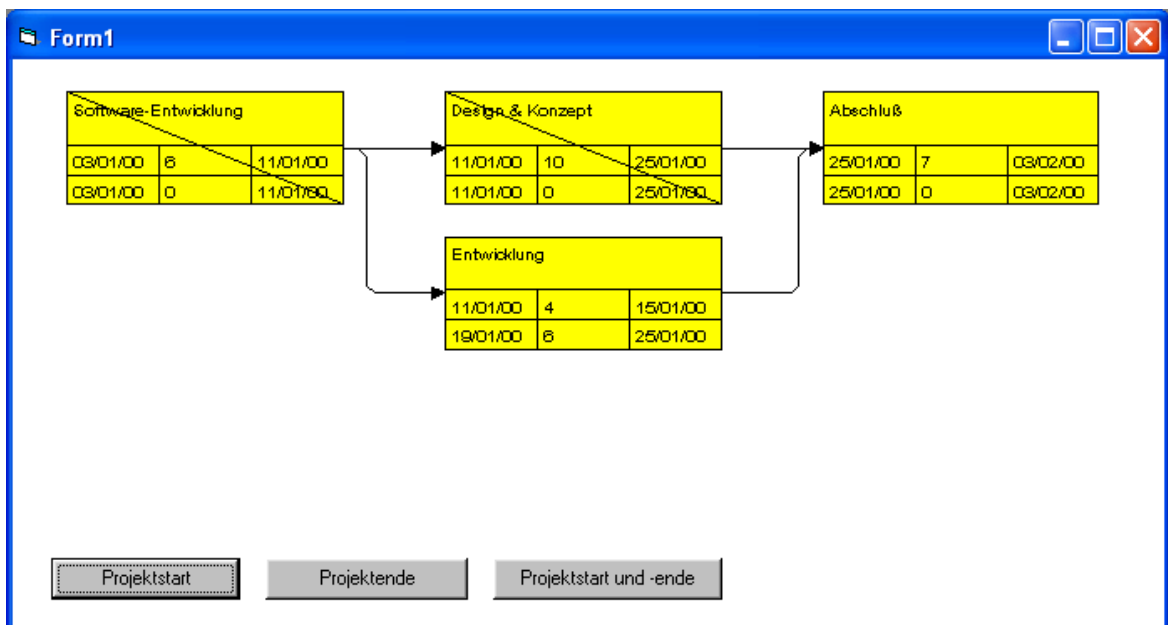
Starten Sie nun das Programm. Die Knoten und Verbindungen, die Sie über die API geladen haben, werden angezeigt:



verwendetes Knotenformat: "Groß"

Name		
Frühester Start	Dauer	Frühestes Ende
Spätester Start	Pufferzeit	Spätestes Ende

Klicken Sie dann auf die Schaltfläche "Projektstart". Nun werden die Termine so berechnet, dass der Starttermin berücksichtigt wird.

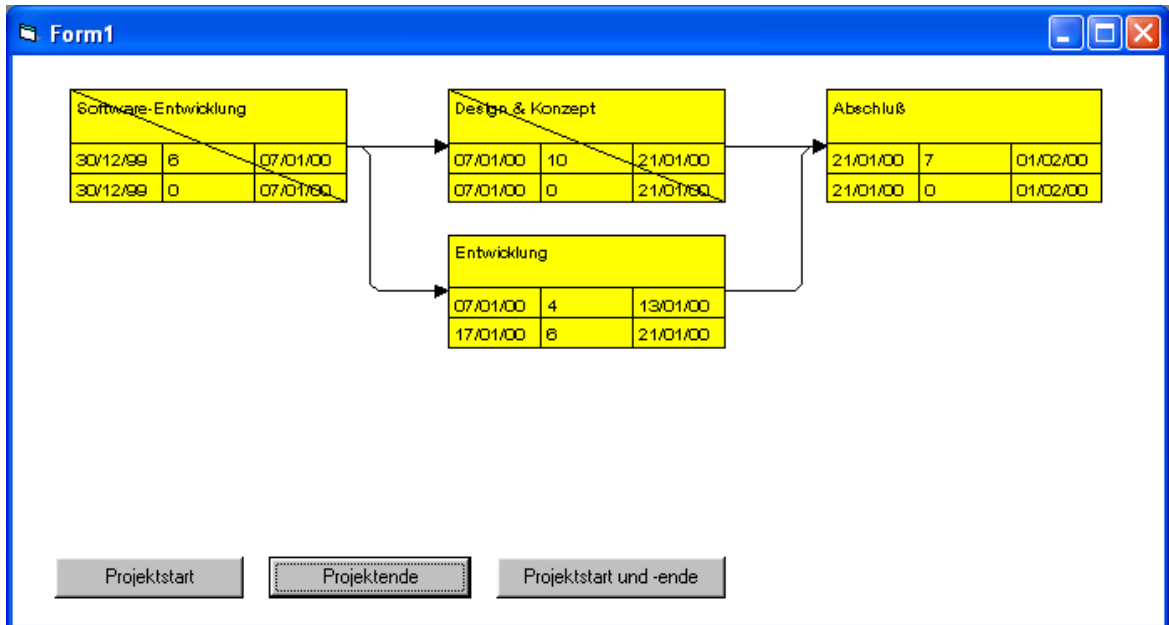


Beachten Sie bitte, dass bei der Berechnung der Termine der interne Kalender berücksichtigt wurde, in dem die Wochenenden als arbeitsfreie Zeiten festgelegt sind. (Der interne Kalender wird verwendet, wenn auf der

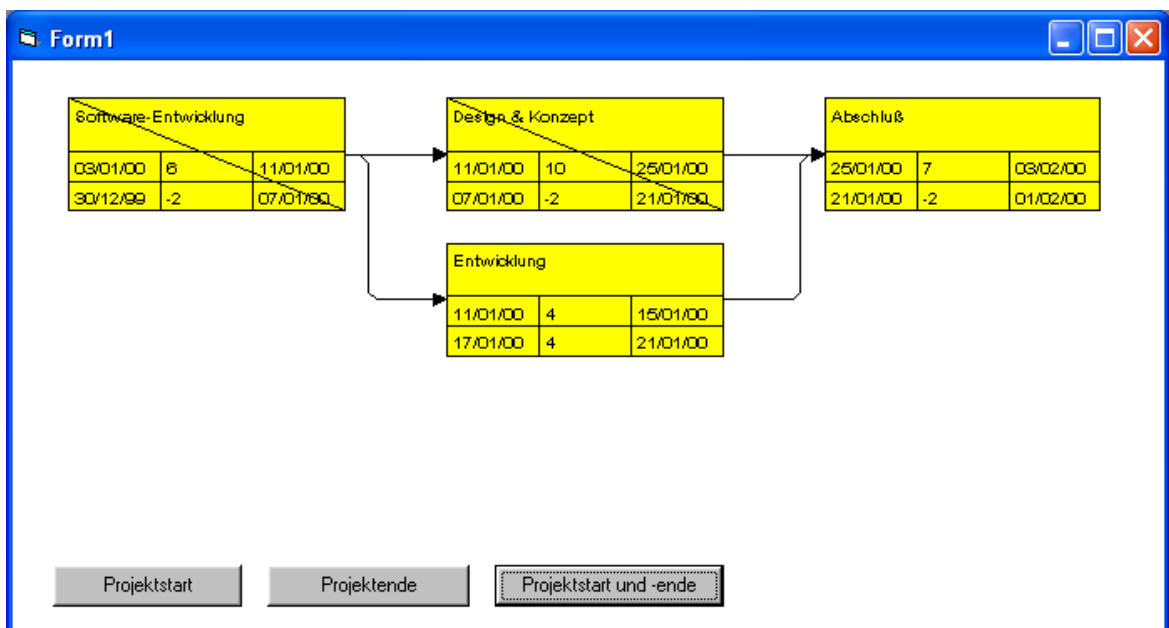
80 Zeitrechnung des VARCHART XNet festlegen

Eigenschaftenseite **Allgemeines** das Kontrollkästchen **Scheduler benutzt internen Kalender** aktiviert ist.)

Klicken Sie nun auf die Schaltfläche "Projektende". Nun werden die Termine so berechnet, dass der Endtermin berücksichtigt wird.



Klicken Sie schließlich auf die Schaltfläche "Projektstart und -ende". Nun werden die Termine so berechnet, dass der Start- und der Endtermin berücksichtigt werden.



Sie sehen, dass bei der Berücksichtigung von Start- und Endtermin negative Pufferzeiten auftreten.

2.20 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 VcNet 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 VcNet 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 VcNet 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 5.12, "Seite einrichten".

2.21 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**.

2.22 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 Dateierendungen. 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:**

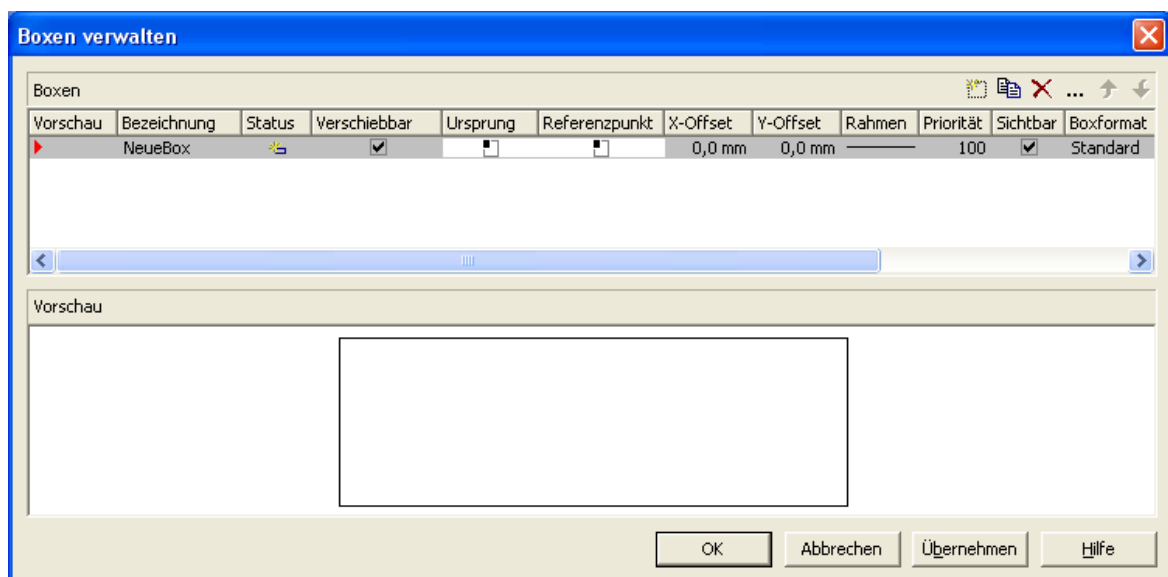
Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, aus der die Einstellungen geladen werden sollen. Wenn Sie einen vorhandenen Dateinamen angeben und auf die **Übernehmen**-Schaltfläche klicken, wird die gewählte Konfiguration geladen und von nun an mit der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

Hinweis: Die Einstellungen der Konfigurationsdatei werden nur einmal geladen, d. h. VARCHART XNet 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 VARCHART XNet 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
- 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)

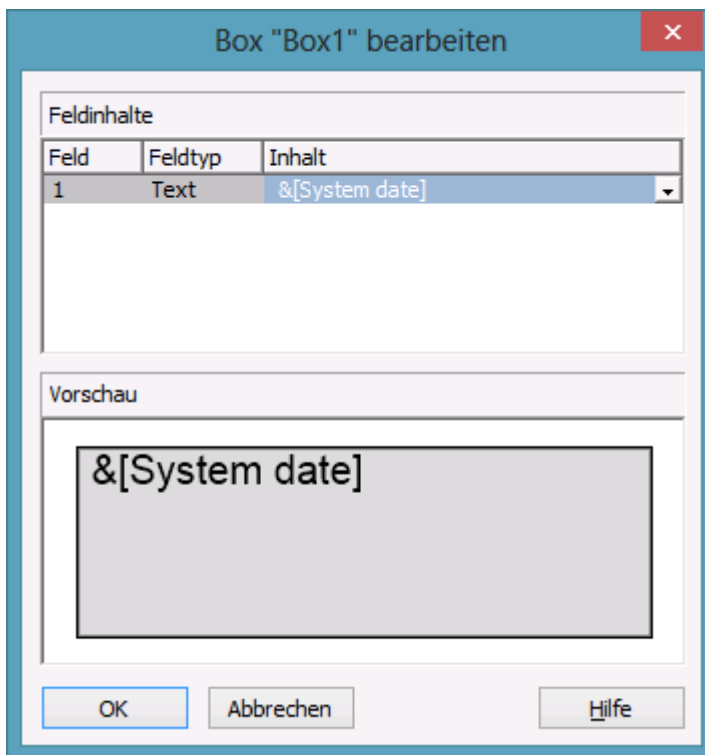
86 Wichtige Konzepte: Boxen

- 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).

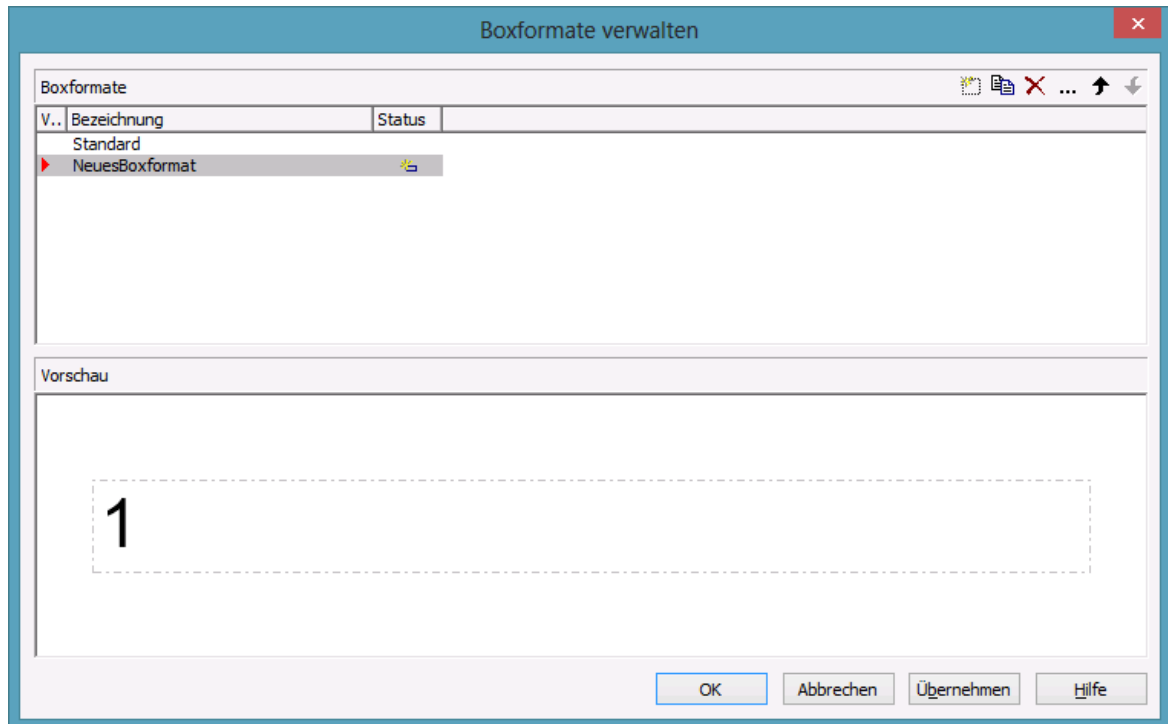
In der Spalte **Typ** wird der Feldtyp jedes Feldes angezeigt (Text oder Grafik).

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

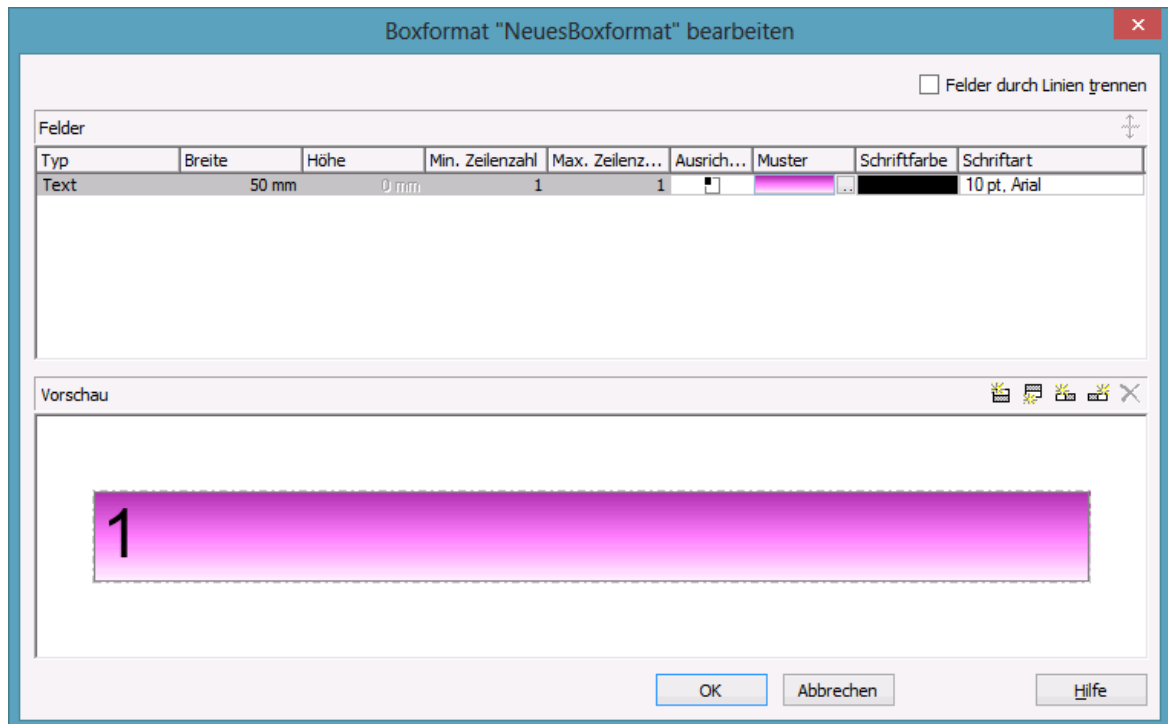
Für jede Box können Sie ein Boxformat wählen. Die Boxformate können Sie selbst festlegen.

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.

88 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 Daten

Knoten und Verbindungen werden grundsätzlich über die API geladen. Das kann entweder mit der Methode **Open** geschehen (Einlesen einer Datei) oder mit den Methoden **InsertNodeRecord** und **InsertLinkRecord**. Die Daten werden als String oder in einem Variant-Array übergeben, wobei die Werte für die einzelnen Datenfelder geordnet und mit Semikolon getrennt übergeben werden. Wenn in den Daten selbst ein Semikolon vorkommt, muss dieses Datenfeld in Anführungsstriche eingeschlossen werden. In Visual Basic wird statt der Anführungszeichen "+Chr\$(34)+" verwendet.

Code-Beispiel

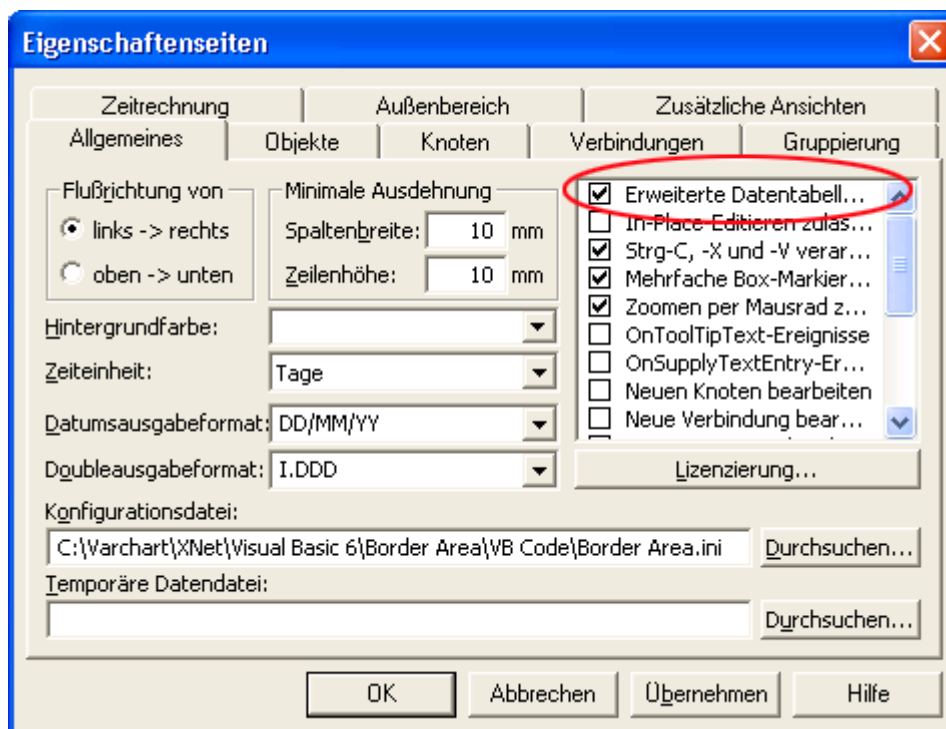
```
VcNet1.InsertNodeRecord "1;1.;;" + Chr$(34) + " Firma A; Abteilung B "  
+ Chr$(34) + ""
```

In eine Datei gespeichert werden die Daten mit der Methode **SaveAs**. Wollen Sie in Ihrer Applikation eigene Speichermechanismen verwenden, können Sie über die **NodeCollection** die Daten aller Knoten und über die **LinkCollection** die Daten aller Verbindungen zwischen den Knoten ausfragen.

3.3 Datentabellen

Zur Darstellung von Netz-Diagrammen verwendet VARCHART XNet zwei Standard-Datentabellen für Knoten und Verbindungen, deren Felder individuell festgelegt werden können. In VARCHART XNet 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.


Aus Kompatibilitätsgründen mit bestehenden Anwendungen arbeitet VARCHART XNet 4.0 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**:

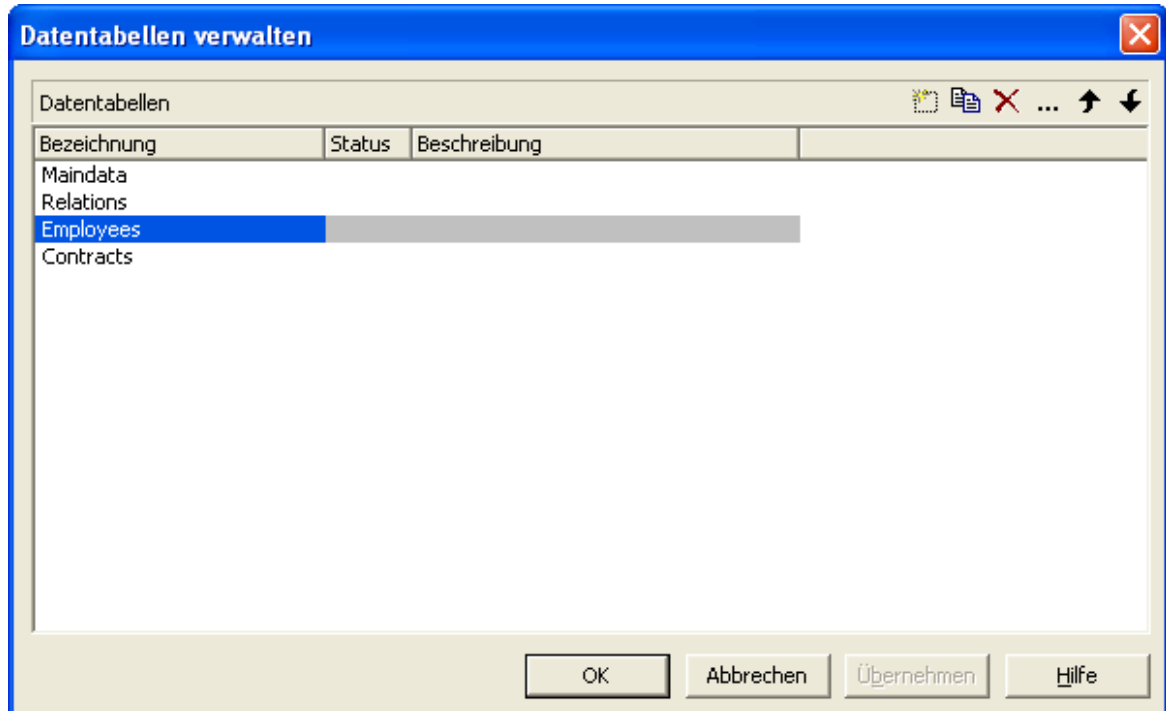






Alternativ können die erweiterten Datentabellen statt zur Designzeit auch zur Laufzeit über die API eingeschaltet werden, indem die Eigenschaft **ExtendedDataTables** des Objekts **VcNet** auf **True** gesetzt wird.

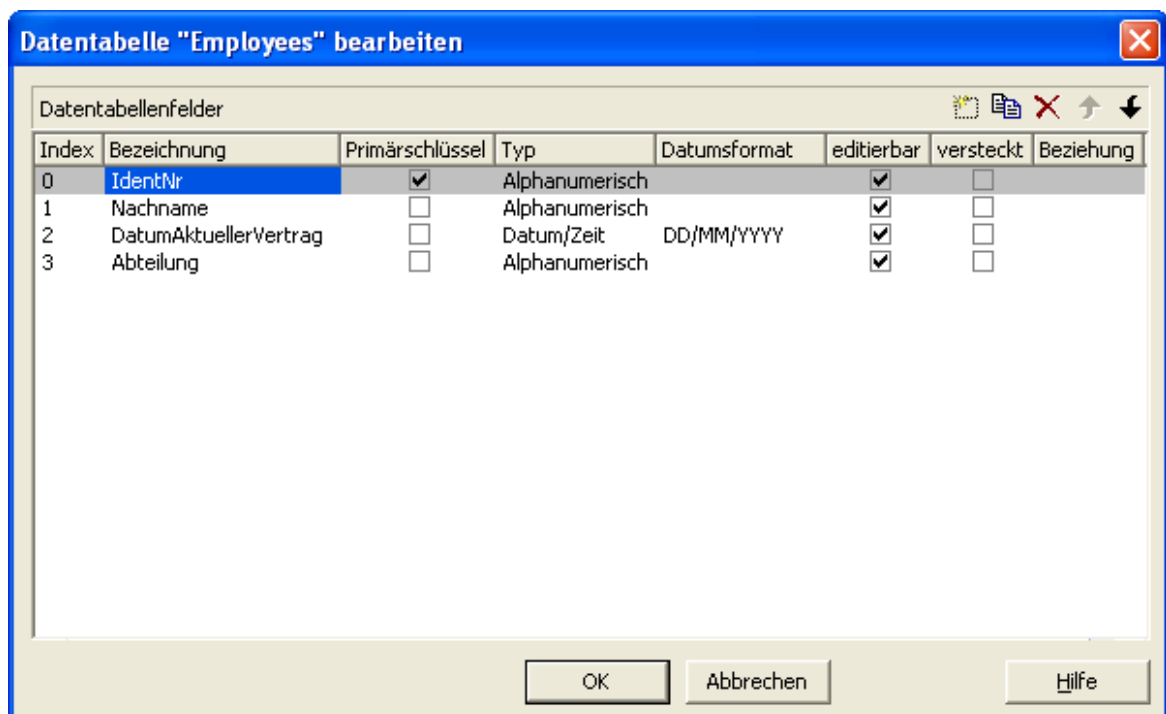
> Datentabellen verwalten

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 **Employees** und **Contracts** durch Klick auf  neu angelegt.



Über die Schaltfläche  gelangen Sie in den Dialog **Datentabelle bearbeiten**. Sie können in diesem Dialog neue Felder anlegen , bestehende Felder löschen  oder Felder kopieren .



92 Wichtige Konzepte: Datentabellen

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 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 **Employees** und **Contracts** durch die Angabe von **Employees:IdentNr** in der Spalte **Beziehung** vereinbart.

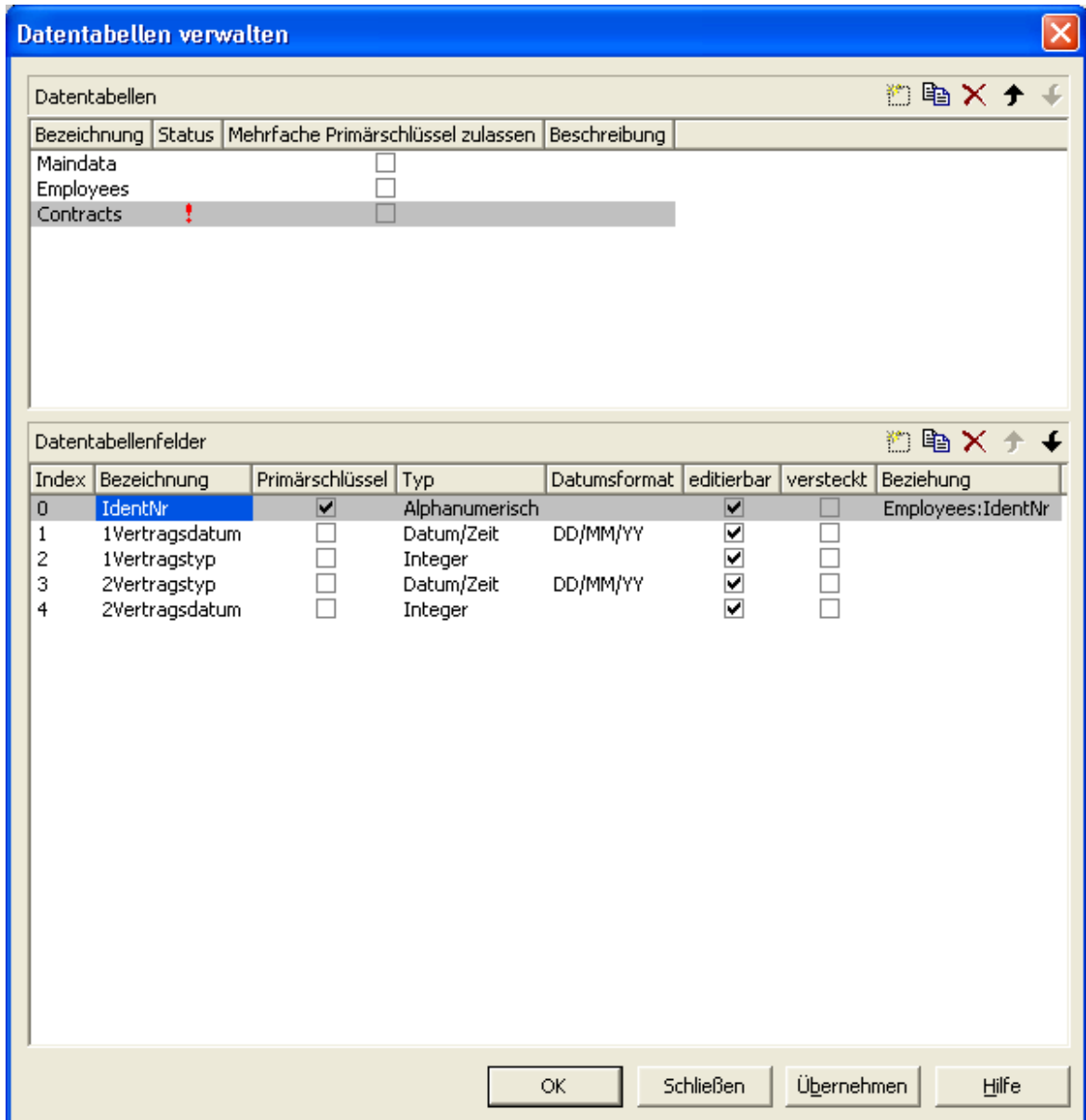


Tabelle Employees:

IdentNr	Nachname	Abteilung
1	Müller	Forschung und Entwicklung
2	Schmidt	Verwaltung

Tabelle Contracts:

ID	Vertragsdatum	Gehaltsstufe	MitarbeiterIdentNo
1	1996/08/01	3	1
2	2006/06/01	4	1

94 Wichtige Konzepte: Datentabellen

ID	Vertragsdatum	Gehaltsstufe	MitarbeiterIdentNo
3	1994/03/01	1	2
4	1996/05/01	2	2
5	2001/10/01	3	2

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

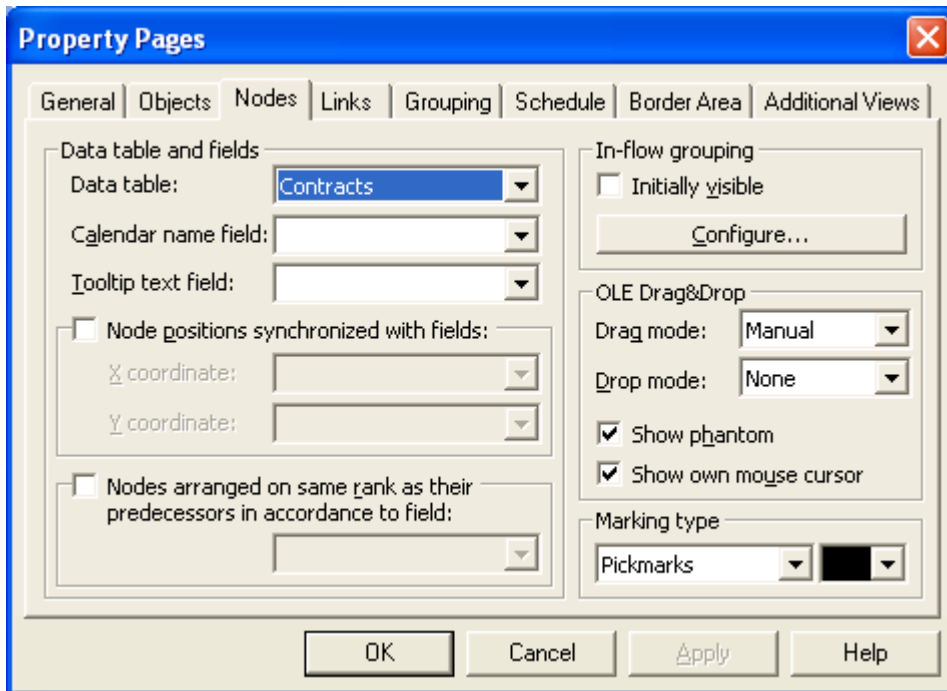
Set dataTableCltn = VcNet1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByName("Employees")

dataTable.DataRecordCollection.Add ("1;Müller;Forschung und
Entwicklung")
dataTable.DataRecordCollection.Add ("2;Schmidt;Verwaltung")

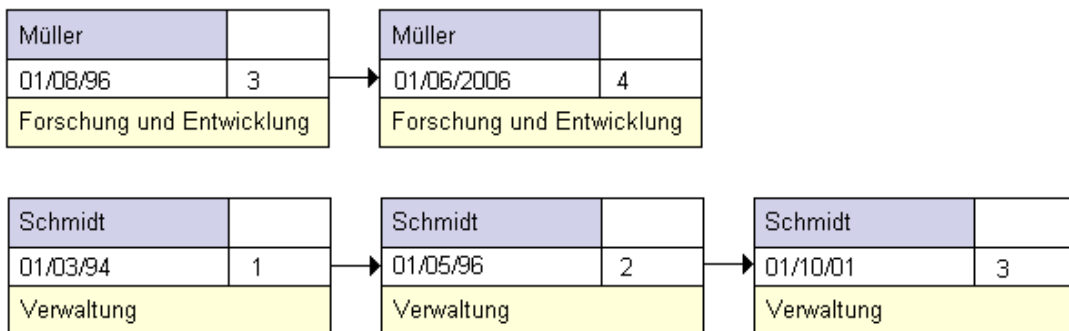
Set dataTable = dataTableCltn.DataTableByName("Contracts")
dataTable.DataRecordCollection.Add ("1;1996/08/01;3;1")
dataTable.DataRecordCollection.Add ("1;2006/06/01;4;1")
dataTable.DataRecordCollection.Add ("1;1994/03/01;1;2")
dataTable.DataRecordCollection.Add ("1;1996/05/01;2;2")
dataTable.DataRecordCollection.Add ("1;2001/10/01;3;2")

VcNet1.EndLoading
```

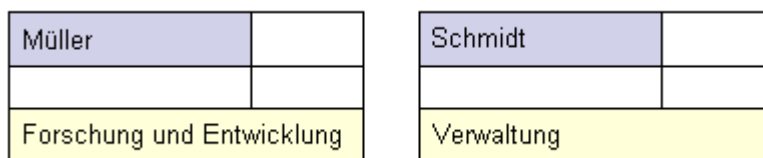
In Abhängigkeit davon, welche Tabelle unter **Datentabelle** bei der Eigenschaftenseite **Knoten** angegeben wird, ergibt sich eine andere Basis, aus der die grafische Darstellung der Knoten hergeleitet wird. Beim interaktiven Anlegen von Knoten ist es diejenige Tabelle, der automatisch neue Datensätze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



Dies ist das Ergebnis im Netz-Diagramm, wenn die Tabelle **Contracts** als Basis genommen wird. Die Einträge für **Nachname** stammen aus der Haupttabelle **Employees**.



Die sichtbaren Daten in XNet bestehen nur aus zwei Einträgen, wenn statt der Tabelle **Contracts** die Tabelle **Employees** herangezogen wird.



In der VARCHART XNet 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

```

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

VcNet1.ExtendedDataTablesEnabled = True
Set dataTable = VcNet1.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")

VcNet1.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

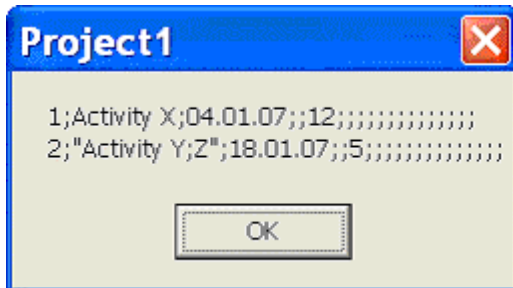
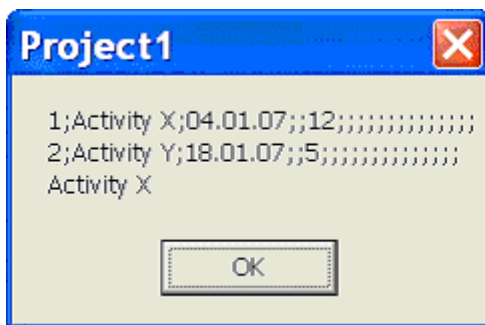
content = dataRec1.AllData & vbCr & dataRec2.AllData & vbCr &
dataRec1.DataField(Main_Name)
MsgBox (content)

'...

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

```

Erzeugte Ausgabe:



3.4 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.

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 Knoten oder Verbindungen. Mit Hilfe eines Filters können Sie alle Knoten bzw. Verbindungen, die die Filterbedingungen erfüllen, auswählen, um sie beispielsweise grafisch hervorzuheben.

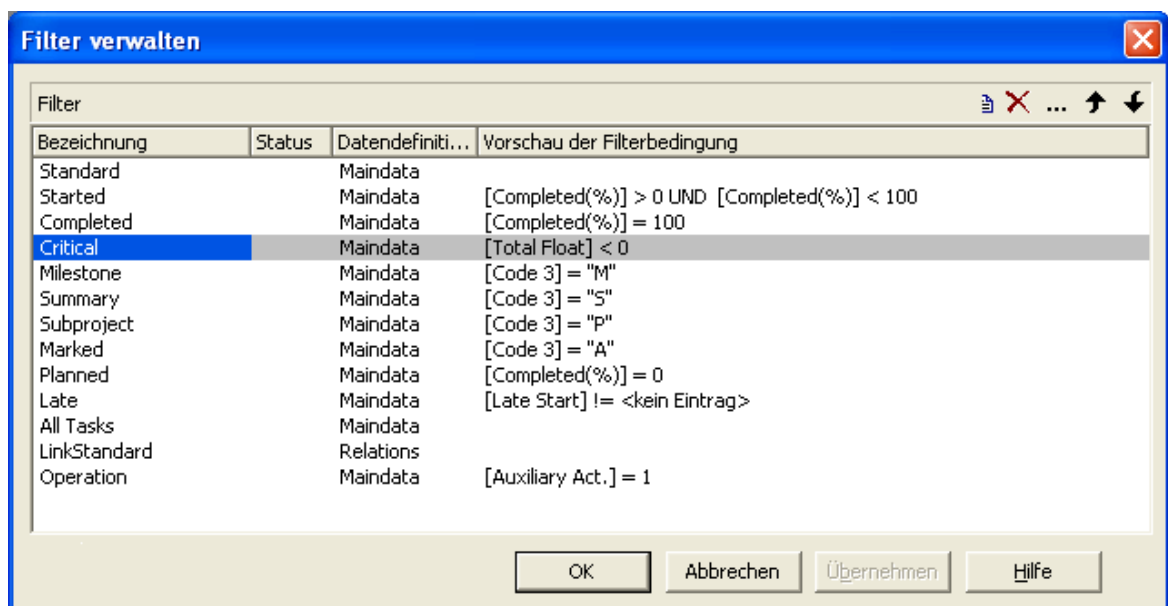
Wenn Sie einen Filter anwenden, werden die Informationen des Datensatzes eines Knotens bzw. einer Verbindung mit den Kriterien aus dem Filter verglichen. Es werden die Knoten bzw. Verbindungen ausgewählt, die die Filterkriterien erfüllen.

Beispielsweise können Sie den Filter "Alle Knoten mit Beginn ab Januar 2009" definieren, um alle Knoten auszuwählen, die ab Januar 2009 beginnen.

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

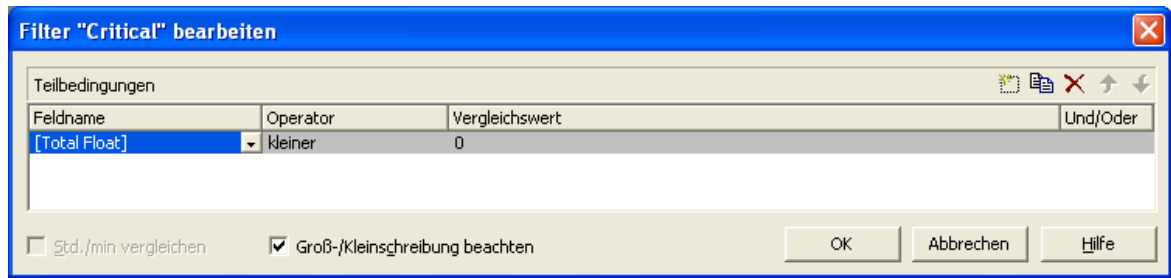
Sie erreichen das Dialogfeld **Filter verwalten** über die Eigenschaftenseite **Objekte**. Das Dialogfeld **Filter verwalten** für Verbindungen erreichen Sie außerdem über die Eigenschaftenseite **Verbindungen**.

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



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

102 Wichtige Konzepte: Filter



3.7 Grafikformate

VARCHART unterstützt die folgenden Grafikformate, was für den Export von Grafiken für Bedeutung ist, vor allem für die Aufrufe **VcNet1.ShowGraphicsExportDialog** und **VcNet1.ExportGraphics**.

Das XNet-Steurelement 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.

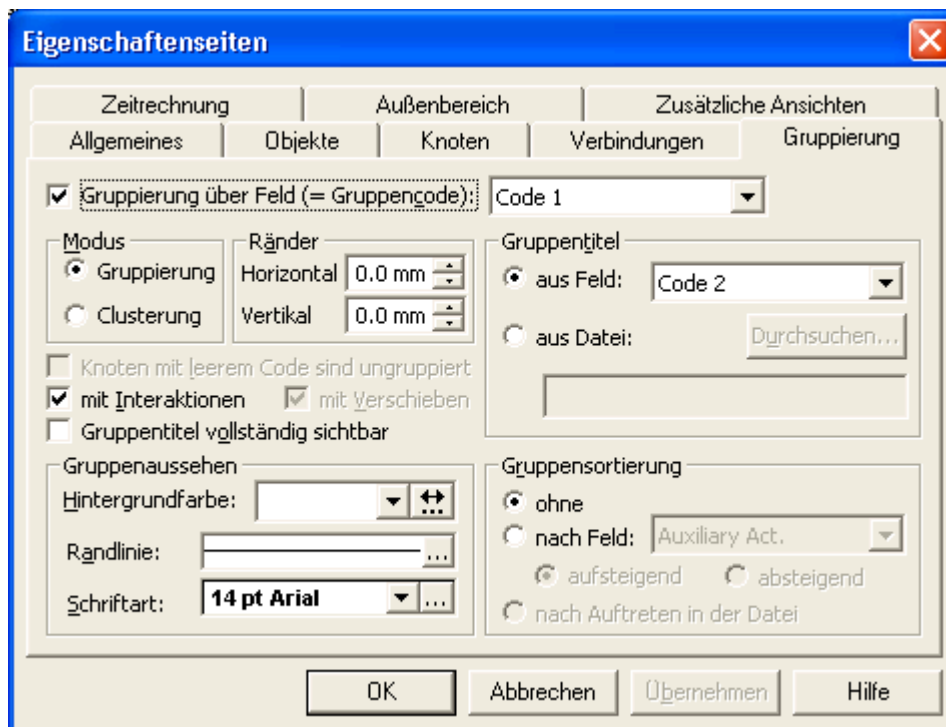
Formatbeschreibung siehe:

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

3.8 Gruppierung

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

Die Kriterien für die Gruppierung können Sie auf der Eigenschaftenseite **Gruppierung** einstellen.



Die Knoten werden nach dem Feld gruppiert, das Sie aus der Kombobox **Gruppierung über Feld (= Gruppencode)** auswählen. Legen Sie hier fest, nach welchem Feld gruppiert werden soll. Das Feld, das Sie hier auswählen, wird **Gruppencode** genannt. Alle Knoten, die in dem hier gewählten Feld denselben Eintrag haben, werden in einer Gruppe zusammengefasst.

Wenn der Anwender einen Knoten von einer Gruppe in eine andere schiebt, wird der Wert in dem Datenfeld, das als Gruppencode vereinbart wurde, automatisch angepasst.

Sie können wählen, ob die Gruppentitel aus einem Datenfeld oder aus einer Datei genommen werden sollen.

- Aktivieren Sie das Kontrollkästchen **aus Feld**, damit die Gruppentitel aus dem Datenfeld genommen werden, das Sie hier auswählen. Dieses Feld muss nicht notwendigerweise mit dem Gruppencode identisch sein. Doch damit die Gruppen sinnvoll beschriftet werden, sollten die Einträge im

Gruppencode-Feld und im **Gruppentitel-Feld** miteinander korrespondieren.

- Wenn Sie das Kontrollkästchen **aus Datei** aktivieren, werden die Gruppentitel aus der Datei übernommen, die Sie hier auswählen. Der Aufbau einer Gruppentitel-Datei erfolgt nach folgendem Schema:

A Gruppe A
 B Gruppe B
 C Gruppe C

> **Beispiel:**

"A" = *Kurztext*

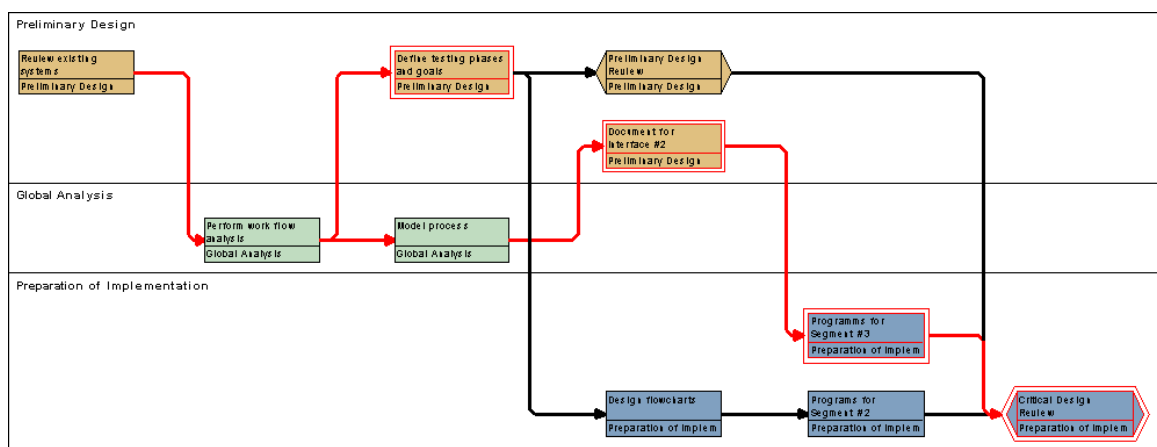
Leerzeichen = Trennzeichen

"Gruppe A" = *Langtext*

> **Darstellungsmodi: Gruppierung und Clusterung**

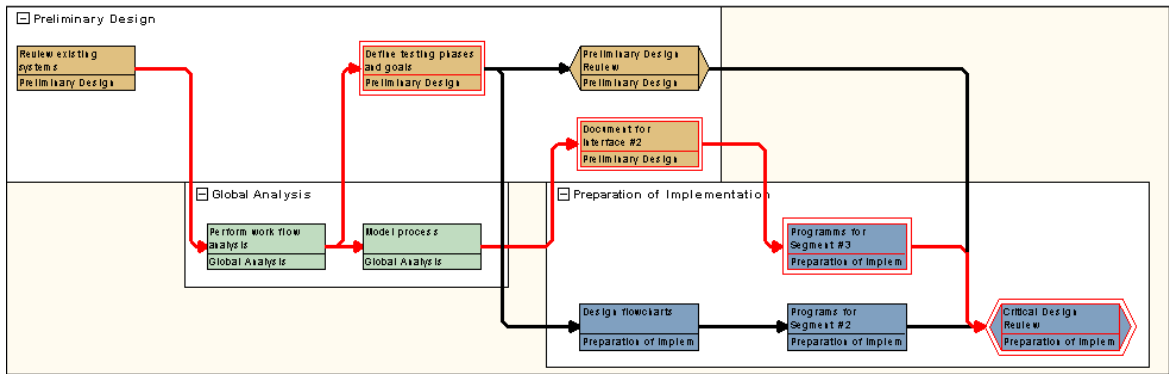
Sie können zwischen zwei Darstellungsmodi wählen:

- **Gruppierung:** normale Gruppendarstellung (die Breite und Höhe jeder Gruppe wird durch die Positionen der Knoten bestimmt; jede Gruppe nimmt jeweils die gesamte Breite bzw. Höhe des Netzdiagramms ein)
- **Clusterung:** Die Gruppen umschließen die darin enthaltenen Knoten möglichst platzsparend, wobei die Gruppen frei im Netzdiagramm verteilt sind. Im Cluster-Modus können die Gruppen durch Klick auf das Plus- bzw. Minus-Symbol kollabiert und expandiert werden (wenn dies über die Eigenschaft `VcNet.GroupInteractionsAllowed` eingeschaltet ist). Kollabierete Gruppen können mit der Maus verschoben werden wie Knoten.



Beispiel für Gruppierungsmodus

110 Wichtige Konzepte: Gruppierung



Beispiel für Cluster-Modus

In beiden Modi können Sie interaktiv Knoten verschieben, löschen oder neu erzeugen.

Den Gruppierungsmodus können Sie auf der Eigenschaftenseite **Gruppierung** unter **Modus** auswählen oder über die API mittels **VcNet.GroupModus** setzen.

> Sortierung der Gruppen

Sie können die Gruppen sortieren lassen: entweder **aufsteigend** oder **absteigend** nach einem Kriterium Ihrer Wahl (**nach Feld**) oder nach dem Auftreten der Gruppen in der Datei.

> Aussehen der Gruppen

Auch das Aussehen der Gruppen können Sie frei gestalten, indem Sie die Trennlinien zwischen den Gruppen (**Randlinie**), den **Hintergrund** der Gruppen und die **Schriftart** der Gruppentitel festlegen.

> Ereignisse

Auf folgende Ereignisse können Sie reagieren:

- **OnGroupCreate**
- **OnGroupDelete**
- **OnGroupLClick**
- **OnGroupLDbClick**
- **OnGroupModify**
- **OnGroupModifyComplete**
- **OnGroupRClick**

3.9 Identifikation

Auf der Eigenschaftenseite **Objekte** wählen Sie zunächst die Eigenschaftenseite **Datentabellen** aus. Nachdem Sie in der oberen Liste des Sialogs eine Datentabelle ausgewählt haben (z. B. die Maindata-Tabelle für Knoten), können Sie in der unteren das Datenfeld auswählen, das die Identifikation eines Knotens enthalten soll. Für die meisten Anwendungen ist es sinnvoll, diesem Feld den Status als Primärschlüssel zuzuweisen.

Die Identifikation der Knoten wird beispielsweise benötigt, um in den Datenfeldern von Verbindungen den Vorgänger- und den Nachfolgerknoten festzulegen. Zudem dient sie zur eindeutigen Identifizierung von Knoten und Verbindungen, um beispielsweise den Zusammenhang mit einer im Hintergrund arbeitenden Datenbank herstellen zu können.

Die Identifikation wird in den Methoden **GetNodeByID** und **GetLinkByID** verwendet, um auf einen bestimmten Knoten bzw. auf eine bestimmte Verbindung über seine bzw. ihre Identifikation zuzugreifen.

112 Wichtige Konzepte: Identifikation

Datentabellen verwalten

Datentabellen

Bezeichnung	Status	Mehrfache Primärschlüssel zulassen	Beschreibung
Maindata	<input type="checkbox"/>		
Relations	<input type="checkbox"/>		

Datentabellenfelder

Index	Bezeichnung	Primärschlüssel	Typ	Datumsformat	editierbar	versteckt
0	ID	<input checked="" type="checkbox"/>	Integer		<input type="checkbox"/>	<input type="checkbox"/>
1	Strukturcode	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Ebene	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Vaterknoten	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Name	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Gruppencode	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Code	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Gruppenname	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Dauer	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Pufferzeit	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Fertig(%)	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Frühester Start	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Frühestes Ende	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Spätester Start	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Spätestes Ende	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Freier Puffer	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Berechneter Start	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Berechnetes Ende	<input type="checkbox"/>	Datum/Zeit	DD/MM/YYYY	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	X-Koordinate (Knoten)	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Y-Koordinate (Knoten)	<input type="checkbox"/>	Integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Hilfsknoten	<input type="checkbox"/>	Alphanumerisch		<input checked="" type="checkbox"/>	<input type="checkbox"/>

OK Schließen Übernehmen Hilfe

3.10 In-Flow-Gruppierung

Bei einer In-Flow-Gruppierung werden die Knoten eines Netz-Diagramms nach einem Kriterium "in flow", also entsprechend der Flussrichtung angeordnet. Wenn die Orientierung also horizontal verläuft, werden die Knoten von links nach rechts sortiert; bei einer vertikalen Orientierung von oben nach unten.

Das Kriterium der Gruppierung kann zeitorientiert nach einem Terminfeld oder Code-orientiert nach einem Datenfeld erfolgen. Die Länge von Zeitintervallen können Sie festlegen:

13.01.2013	27.01.2013	10.02.2013
13.01.2013	27.01.2013	10.02.2013

Beispiel für eine zeitliche Gruppierung bei Links-Rechts-Orientierung

13.01.2013		13.01.2013
27.01.2013		27.01.2013

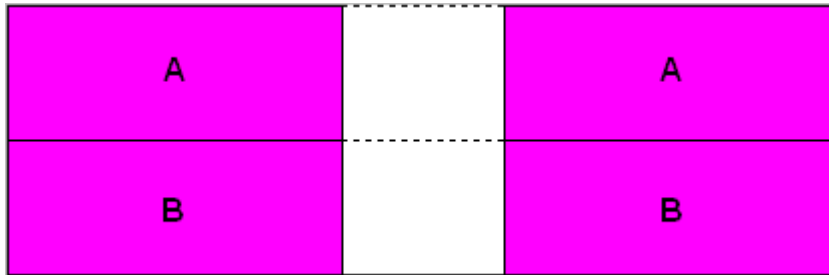
Beispiel für eine zeitliche Gruppierung bei Oben-Unten-Orientierung

Analog ist eine In-Flow-Gruppierung nach bestimmten Datenfeldern möglich.

A	B	C
A	B	C

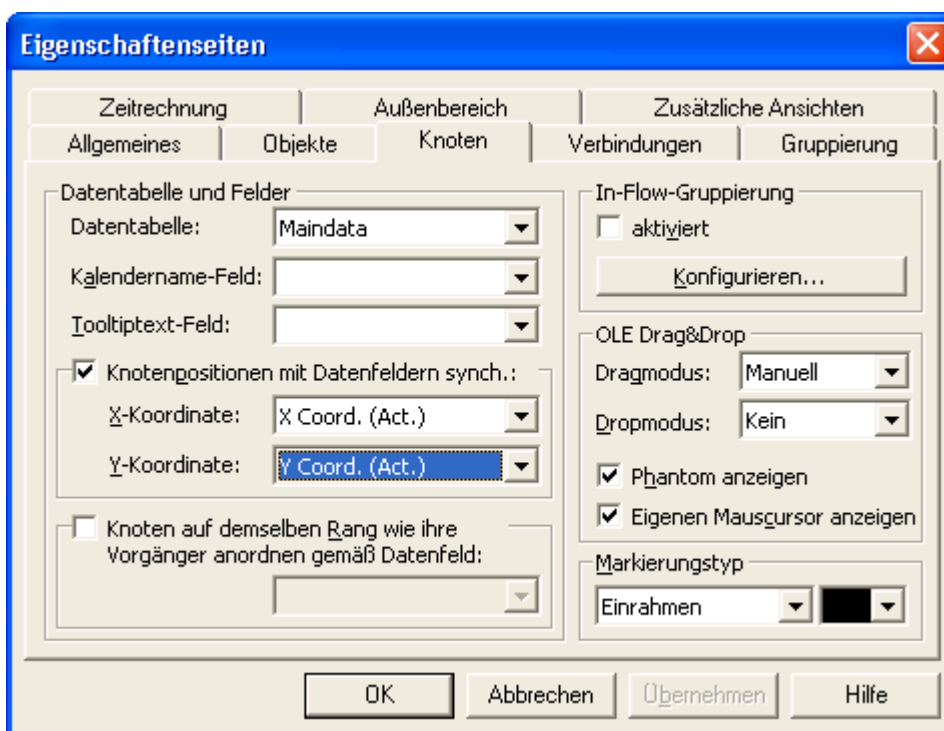
Beispiel für eine Gruppierung nach einem Datenfeld bei Links-Rechts-Orientierung

114 Wichtige Konzepte: In-Flow-Gruppierung



Beispiel für eine Gruppierung nach einem bestimmten Datenfeld bei Oben-Unten-Orientierung

Die In-Flow-Gruppierung können Sie auf der Eigenschaftenseite **Knoten** im Bereich **In-Flow-Gruppierung** aktivieren.



Die Kriterien und das Layout der In-Flow-Gruppierung können Sie bearbeiten, indem Sie die Schaltfläche **Konfigurieren** drücken.

Es erscheint der Dialog **In-Flow-Gruppierung bearbeiten**:

In-Flow-Gruppierung bearbeiten

Code aus Feld: **Act. Finish** Trennlinien:

Zeitintervall: **2 Wochen**

Titelleisten
 oben unten

Datumsformat: **DD.MM.YYYY**

Schrift: **18 pt Arial**

Hintergrundfarbe: **<nicht f...>**

Breite: **50 mm**

Texte
 aus Feld: **Completed(%)**
 aus Datei:

Vorschau

16.07.2017	30.07.2017	13.0
16.07.2017	30.07.2017	13.0

OK Abbrechen Hilfe

Bei einer Orientierung des Diagramms von links nach rechts können Sie oberhalb und/oder unterhalb des Diagrammbereichs Titelleisten ausgeben. Bei einer Orientierung von oben nach unten können Sie links und/oder rechts des Diagrammbereichs Titelleisten ausgeben. Außerdem können Sie festlegen, ob vertikale bzw. horizontale Trennlinien die Gruppengrenzen darstellen sollen. Die Farbe und die Beschriftung der Titelleisten sowie das Aussehen der Trennlinien lassen sich individuell gestalten.

3.11 Knoten

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

> Knoten erzeugen

Wenn auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten und Verbindungen zulassen** gewählt ist, kann der Anwender im Erzeugemodus neue Knoten per Mausklick erzeugen.

Ist dort außerdem das Kontrollkästchen **Neuen Knoten bearbeiten** aktiviert, öffnet sich das Dialogfeld **Vorgänge bearbeiten**, sobald ein Knoten durch Mausklick erzeugt wurde. Hier werden die Daten des interaktiv erzeugten neuen Knotens angezeigt, und Sie können sie nun bearbeiten.

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

Jedes interaktive Neuanlegen eines Knotens wird der Applikation mit dem Ereignis **OnNodeCreate** mitgeteilt.

> Markieren von Knoten

Auf der Eigenschaftenseite **Knoten** können Sie unter **Markierungstyp** festlegen, wie markierte Knoten dargestellt werden sollen.

Mögliche Alternativen sind:

- Ohne
- Einrahmen
- Einrahmen innen
- Invertieren
- Pickmarks
- Pickmarks innen

Hinweis: Wenn Sie den Markierungstyp "Ohne" ausgewählt haben, können Knoten nicht markiert werden.

Jedes Markieren/Demarkieren von Knoten wird der Applikation mit dem Ereignis **OnNodesMarkEx** mitgeteilt. Das Ende einer Markier-/ Demarkier-operation wird durch das Ereignis **OnNodesMarkComplete** mitgeteilt.

> Knoten löschen

Einen oder mehrere Knoten können Sie löschen, indem Sie sie bei gedrückter Umschalt- oder Strg-Taste mit der linken Maustaste markieren und dann die

rechte Maustaste drücken. Wählen Sie dann im Kontextmenü für Knoten den Befehl **Löschen** bzw. **Ausschneiden**.

Markierte Knoten können auch mit der **Entf**-Taste gelöscht werden.

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

Außerdem können Sie Knoten über die API mit der VARCHART-ActiveX-Methode **DeleteNodeRecord** oder mit der VcNode-Methode **DeleteNode** löschen.

> Ereignisse

Auf folgende Ereignisse können Sie reagieren:

- **OnNodeCreate**
- **OnNodeCreateCompleteEx**
- **OnNodeDelete**
- **OnNodeLClick**
- **OnNodeLDbClick**
- **OnNodeModify**
- **OnNodeModifyComplete**
- **OnNodeModifyEx**
- **OnNodeRClick**
- **OnNodesMarkComplete**
- **OnNodesMarkEx**

> Position von Knoten

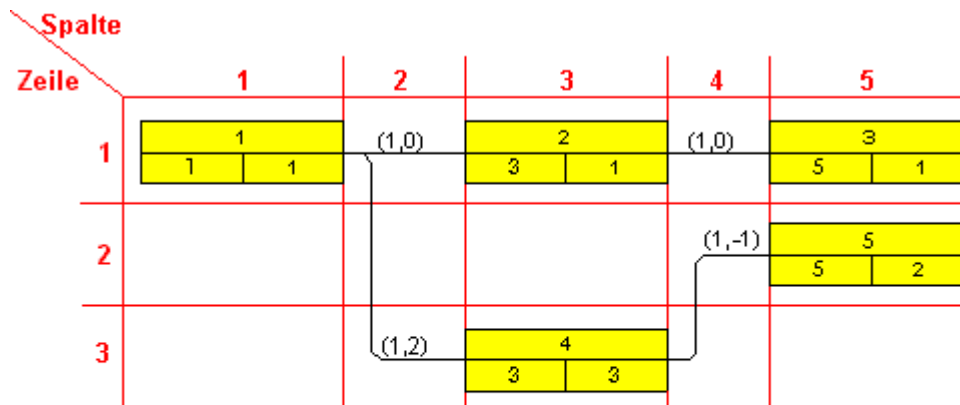
Die Positionen von Knoten und Verbindungsbeschriftungen im Diagramm werden als deren Koordinaten in einem Raster festgelegt.

Die X- und Y-Koordinate eines Knotens geben die absolute Position dieses Knotens im Raster an.

Die X- und Y-Koordinate einer Verbindungsbeschriftung geben die Position dieser Verbindungsbeschriftung relativ zu ihrem Vorgängerknoten an.

Das erste Feld des Rasters (ganz links oben) ist definiert als $(X,Y) = (1,1)$. Es ist für Knoten reserviert. Von links nach rechts und von oben nach unten wird jeweils in Schritten von 1 weitergezählt. In allen weiteren Feldern können Knoten oder Verbindungsbeschriftungen stehen. Knoten haben daher immer positive X- und Y-Koordinaten. Bei Verbindungsbeschriftungen sind alle Wertepaare außer (0,0) möglich.

118 Wichtige Konzepte: Knoten



Legende:

Nummer	
X-Position	Y-Position

 Knotenfeld

(x,y) Verbindungsbeschriftungsposition

> Speichern und Laden von Knotenpositionen

Wenn Sie die Knotenpositionen eines Diagramms wieder laden möchten, müssen Sie diese in bestimmten Datenfeldern speichern.

Um die Positionen der Knoten mit den entsprechenden Datenfeldern zu synchronisieren, aktivieren Sie auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Knotenpositionen mit Datenfeldern synchronisieren** und geben Sie folgende Datenfelder an:

- für die X-Koordinate: "X-Koordinate (Knoten)"
- für die Y-Koordinate: "Y-Koordinate (Knoten)"

(Voraussetzung ist, dass Sie diese Datenfelder beim Einrichten der Schnittstelle entsprechend definiert haben. Siehe hierzu "Tutorium: Schnittstelle einrichten".)

Das Dialogfeld zeigt ein Kontrollkästchen, das mit 'Knotenpositionen mit Datenfeldern synchronisieren:' beschriftet ist. Darunter befinden sich zwei Zeilen für die Koordinationen: 'X-Koordinate:' mit einem Dropdown-Menü, das 'X-Koordinate (Knoten)' zeigt, und 'Y-Koordinate:' mit einem Dropdown-Menü, das 'Y-Koordinate (Knoten)' zeigt.

Die Werte dieser Datenfelder können Sie im Dialogfeld **Vorgänge bearbeiten** ausfragen und ggf. bearbeiten.

> Rang eines Knotens

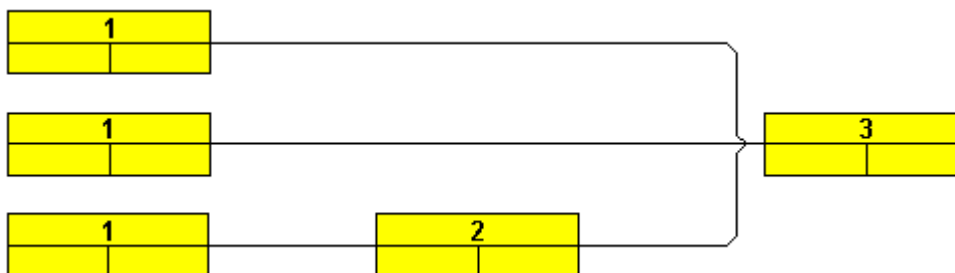
Unter dem Rang eines Knotens versteht man die größte Anzahl der Pfeile auf einem Weg zwischen Startknoten und dem betrachteten Knoten.

Der Rang eines Knotens ohne Vorgänger ist 1. Der Rang eines Knotens mit Vorgängern ist gleich 1 plus Rang desjenigen seiner Vorgänger, der den höchsten Rang besitzt.

Durch diese Definition wird verhindert, dass in Netzdiagrammen Ringbeziehungen (Schleifen) vorkommen.

Einige Beispiele:

- Der Rang eines Knotens mit einem Vorgänger, der keinen Vorgänger besitzt, ist $1+1=2$.
- Der Rang eines Knotens, der drei Vorgänger mit den Rängen 1, 1 bzw. 2 hat, ist $1+2=3$ (siehe Abbildung).



Ränge der Knoten bei Flussrichtung von links nach rechts

Den Rang von Knoten kann man sich folgendermaßen veranschaulichen:

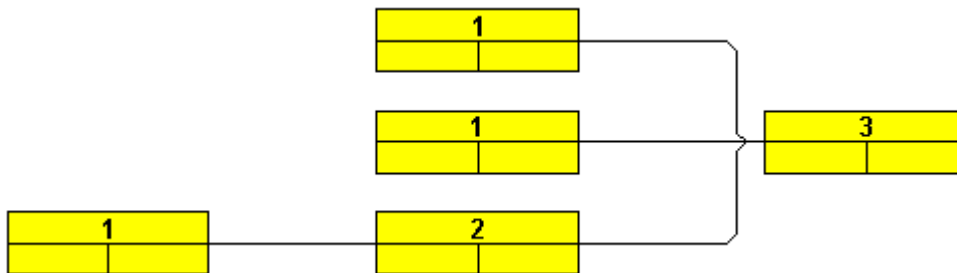
- Bei Flussrichtung von links nach rechts entspricht der höchste Rang von allen Knoten in einer Diagrammspalte der Nummer dieser Diagrammspalte (gezählt werden hierbei nur die Diagrammspalten für Knoten, nicht die für Verbindungsbeschriftungen).
- Bei Flussrichtung von oben nach unten entspricht der höchste Rang von allen Knoten in einer Diagrammzeile der Nummer dieser Diagrammzeile (gezählt werden hierbei nur die Diagrammzeilen für Knoten, nicht die für Verbindungsbeschriftungen).

Ränge werden nur beim Aufruf von **Anordnen** (Kontextmenü für das Diagramm) berechnet. Sie dienen dem Layout-Algorithmus als Vorgabe für die Positionierung der Knoten in Flussrichtung. Bei Ringbeziehungen zwischen Knoten wählt VARChart XNet über einen eigenen Algorithmus Verbindungen aus, die bei der Rangberechnung vorübergehend ignoriert werden, so dass das Layout möglichst natürlich aussieht. Die ignorierten Verbindungen sind nach dem Layout als rückführende Verbindungen sichtbar.

Über die Eigenschaft **ShortenedLinks** oder indem Sie das Kontrollkästchen **Verbindungen kürzen beim Anordnen** auf der Eigenschaftenseite **Allgemeines** aktivieren, erreichen Sie, dass die Positionen der Knoten beim

120 Wichtige Konzepte: Knoten

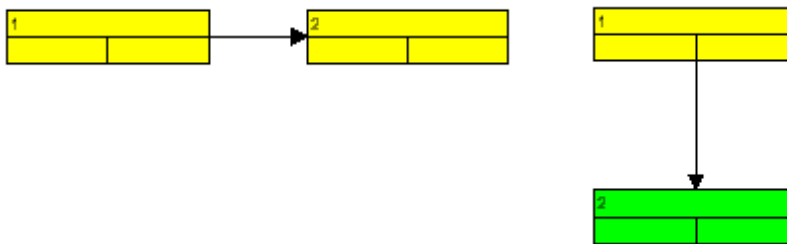
Layout soweit nach rechts bzw. unten verschoben werden, dass die Verbindungen zwischen den Knoten möglichst kurz werden.



> Hilfsknoten

In vielen Fällen ist es sinnvoll, Knoten nicht in Flussrichtung anzuordnen, sondern entgegen der Flussrichtung unter- bzw. oberhalb ihrer Vorgänger (bei Flussrichtung von links nach rechts) bzw. neben ihren Vorgängern (bei Flussrichtung von oben nach unten). Dies ist in VARCHART XNet möglich, da es erlaubt, den Rang von Knoten zu reduzieren.

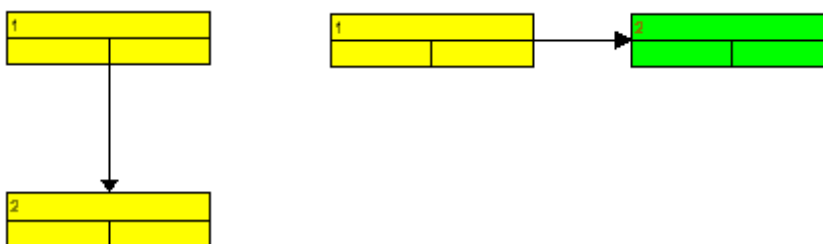
Bei Flussrichtung von links nach rechts steht der Hilfsknoten, dessen Rang um 1 reduziert wurde, unter oder über seinem Vorgänger statt daneben.



Zwei Knoten mit Rang 1 bzw. 2

*Der Rang des 2. Knotens wurde um 1 reduziert. Anschließend wurde der Befehl **Anordnen** aufgerufen.*

Bei Flussrichtung von oben nach unten steht der Hilfsknoten, dessen Rang um 1 reduziert wurde, links oder rechts neben seinem Vorgänger statt darunter.



*Zwei Knoten mit Rang 1 bzw. 2 Der Rang des 2. Knotens wurde um 1 reduziert.
Anschließend wurde der Befehl **Anordnen** aufgerufen.*

Um den Rang eines Knotens reduzieren zu können, ist das Datenfeld "Hilfsknoten" vorgesehen. Der Eintrag in diesem Datenfeld legt fest, wie der Hilfsknoten positioniert wird. Mögliche Werte sind 0, 1, 2, 3.

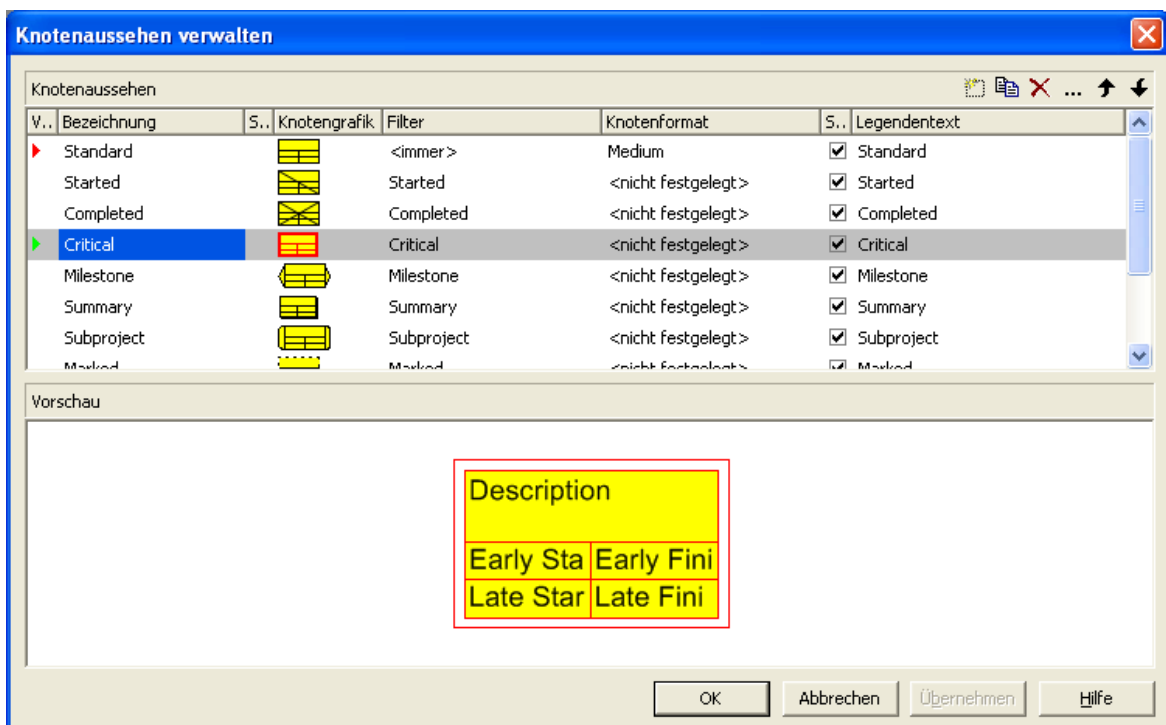
Wert im Feld "Hilfsknoten"	Flussrichtung von oben nach unten	Flussrichtung von links nach rechts
0	Der Rang des Hilfsknotens wird nicht reduziert.	Der Rang des Hilfsknotens wird nicht reduziert.
1	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links oder rechts neben seinem Vorgänger statt darunter.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über oder unter seinem Vorgänger statt daneben.
2	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links von seinem Vorgänger.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über seinem Vorgänger.
3	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann rechts von seinem Vorgänger.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann unter seinem Vorgänger.

Um Hilfsknoten auf demselben Rang zu positionieren wie ihre Vorgänger, aktivieren Sie auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Knoten auf demselben Rang wie seine Vorgänger anordnen gemäß Datenfeld**. Wählen Sie das Datenfeld "Hilfsknoten" aus, dessen Eintrag bestimmen soll, ob ein Knoten auf demselben Rang positioniert wird wie sein Vorgänger. (Dieses Datenfeld müssen Sie ggf. auf der Eigenschaftenseite **Datendefinition** erst definieren.) Es kann die Werte 0, 1, 2 oder 3 annehmen. Von dem Wert eines Knotens im Datenfeld "Hilfsknoten" hängt dann ab, ob ein Hilfsknoten auf demselben Rang positioniert wird wie sein Vorgänger und wie er positioniert wird.

3.12 Knotenaussehen

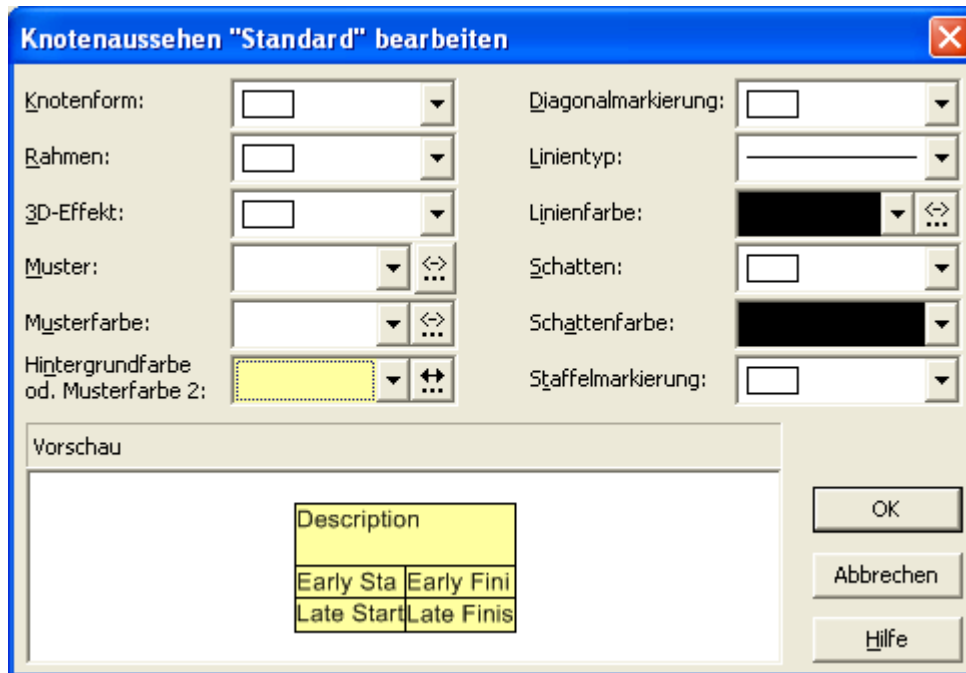
Das Aussehen von Knoten lässt sich in Abhängigkeit von deren Daten festlegen. Beispielsweise können alle Knoten der Abteilung A durch einen roten Hintergrund und eine doppelte schwarze Rahmenlinie gekennzeichnet werden, Knoten der Abteilung B durch einen blauen Hintergrund und eine einfache gelbe Rahmenlinie etc. Diese grafischen Attribute werden als Knotenaussehen bezeichnet.

Sie können Knotenaussehen festlegen, indem Sie auf der Eigenschaftenseite **Objekte** auf die **Knotenaussehen**-Schaltfläche klicken, um das Dialogfeld **Knotenaussehen verwalten** zu öffnen. Hier können Sie Knotenaussehen kopieren, bearbeiten, löschen und neu definieren oder deren Abarbeitungsreihenfolge verändern.



Jedes Knotenaussehen ist mit einem Filter und einem Knotenformat verbunden. Der Filter gibt die Bedingungen an, unter denen dieses Knotenaussehen auf einen Knoten angewandt wird. Beispielsweise ist das Knotenaussehen "Markiert" mit dem Filter "Markiert" verbunden, der alle markierten Knoten selektiert.

Um ein Knotenaussehen zu bearbeiten, klicken Sie auf die **Knotenaussehen bearbeiten**-Schaltfläche oder doppelklicken Sie auf das **Knotengrafik**-Feld. Sie gelangen dann in das folgende Dialogfeld:



Erfüllt ein Knoten die Filterkriterien mehrerer Knotenaussehen, werden diese Knotenaussehen für den Knoten grafisch überlagert. Begonnen wird dabei jeweils mit dem Knotenaussehen, das in der Tabelle ganz oben steht. Das Knotenaussehen, das ganz unten steht, wird als letztes zugewiesen und überlagert daher alle anderen. Die niedrigste Position hat i. d. R. das Knotenaussehen "Standard", das normalerweise ganz oben in der Tabelle steht. Es wird auf alle Knoten angewendet und kann nicht gelöscht werden.

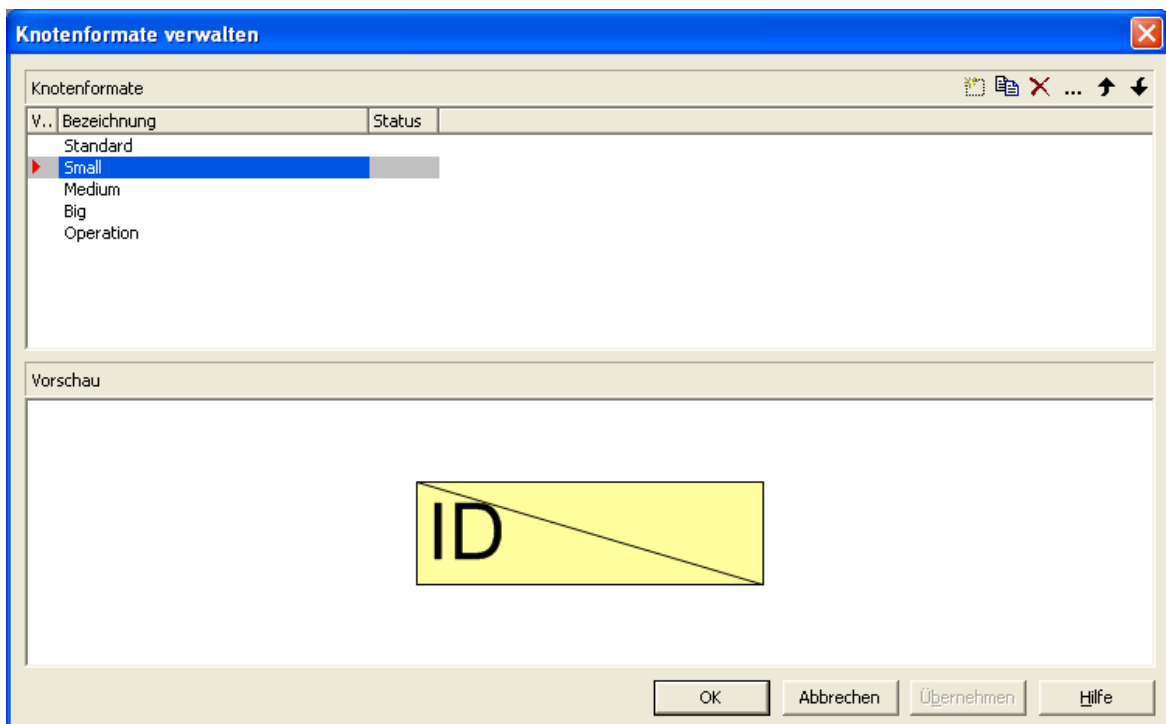
↑ ↓ Sie können die Abarbeitungsreihenfolge der Knotenaussehen mit Hilfe der Pfeil-Schaltflächen verändern.

Die Hintergrundfarbe und die Linienfarbe eines Knotenaussehens können Sie mit Hilfe von Zuordnungstabellen in Abhängigkeit von den Knotendaten festlegen. Siehe hierzu das Kapitel "Wichtige Begriffe: Zuordnungstabellen".

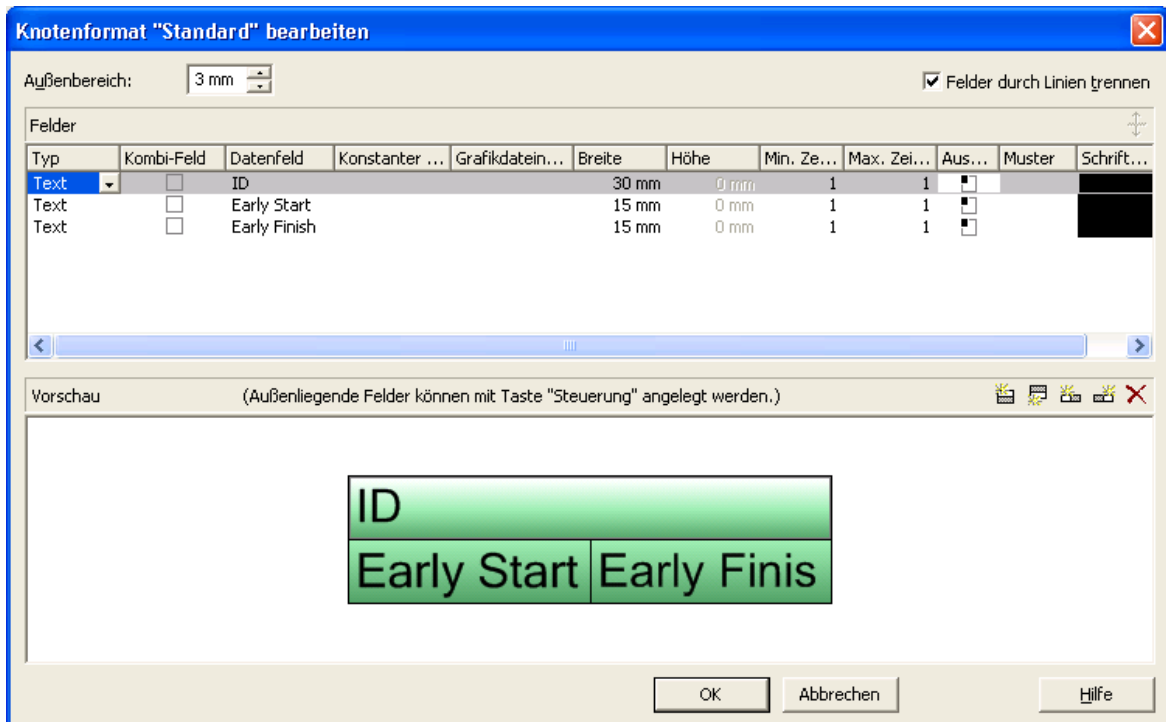
3.13 Knotenformat

Jedes Knotenaussehen ist mit einem Knotenformat verbunden. Sie können das Knotenformat für ein Knotenaussehen festlegen, indem Sie im Dialogfeld **Knotenaussehen verwalten** aus der Kombobox unter **Knotenformat** das gewünschte Knotenformat auswählen.

Knotenformate werden im Dialogfeld **Knotenformate verwalten** verwaltet. Dieses Dialogfeld erreichen Sie, indem Sie auf der Eigenschaftenseite **Objekte** auf die Schaltfläche **Knotenformate** klicken.



Um das aktuelle Knotenformat zu bearbeiten, klicken Sie auf die **Knotenformat bearbeiten**-Schaltfläche. Sie gelangen dann in das Dialogfeld **Knotenformat bearbeiten**, in dem Sie den Aufbau und das Aussehen des jeweiligen Knotens bearbeiten können.



In diesem Dialog können Sie für das gewählte Knotenformat Folgendes festlegen:

- ob die Knotenfelder durch Linien voneinander getrennt werden sollen
- den Außenbereich (Abstand in Millimetern, den Knoten mit diesem Knotenformat zu benachbarten Knoten und zum Rand der Darstellung halten)
- den Feldtyp des aktuellen Knotenfeldes (Text oder Grafik)
- für den Typ Text: das Datenfeld, dessen Inhalt in dem aktuellen Knotenfeldes ausgegeben werden soll, oder einen konstanten Text
- für den Typ Grafik: Name und Pfad der Grafikdatei, die in dem gewählten Knotenfeld dargestellt wird
- die Breite und Höhe des markierten Knotenfeldes
- die maximale Anzahl von Textzeilen im aktuellen Knotenfeld
- die Ausrichtung des Textes bzw. der Grafik des markierten Knotenfeldes
- das Füllmuster und die Musterfarben des Knotenfeldes
- die Schriftart und -farbe des Knotenfeldes

> **Datumsformat der Terminfelder von Knoten**

Das Datumsformat der Terminfelder von Knoten wird auf der Eigenschaftenseite **Allgemeines** festgelegt.

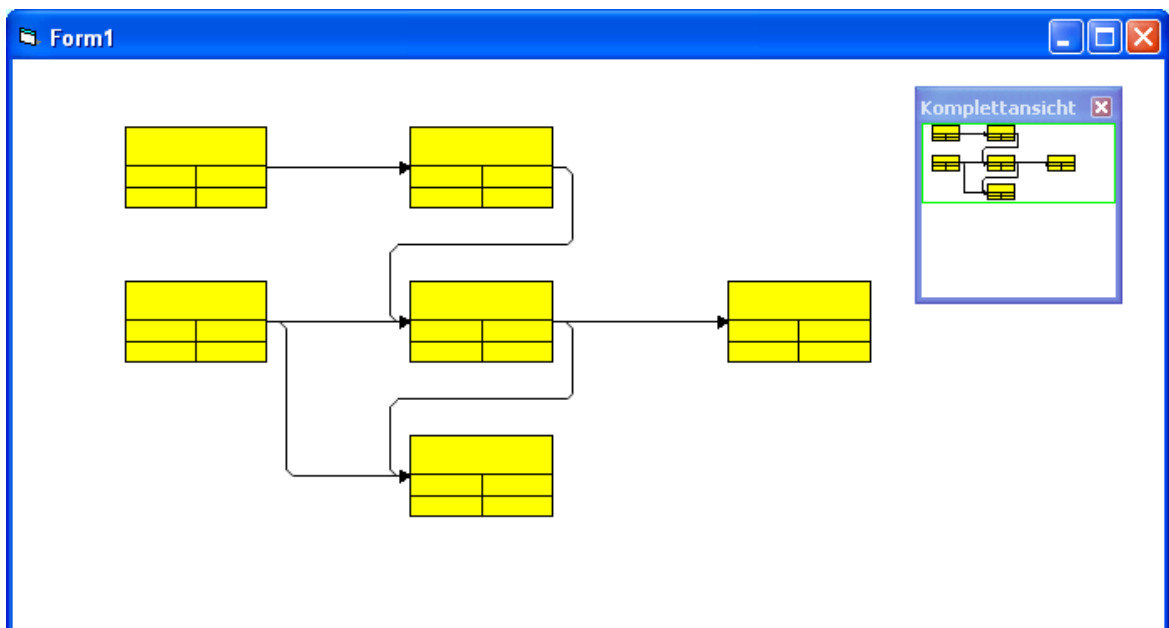
> **Darstellung von Grafiken in Knotenfeldern**

Sie können für ein Knotenfeld des Typs Grafik eine Grafikdatei wählen, indem Sie das Feld **Grafikdateiname** markieren, auf die dann erscheinende Schaltfläche **Grafikdatei auswählen** (...) klicken und anschließend im gleichnamigen Windows-Dialog eine Datei wählen.

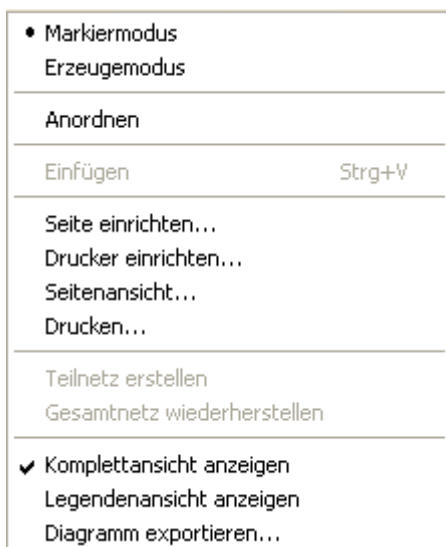
Sie können aber auch eine Zuordnung zwischen den Einträgen eines Datenfeldes und Grafikdateien herstellen. Klicken Sie dazu auf die Schaltfläche **Zuordnungen einstellen** (<>), um den gleichnamigen Dialog zu öffnen. Wenn eine Zuordnung vorgenommen worden ist, wird das durch ein Symbol neben dem Grafikdateinamen dargestellt (↔). Einzelheiten hierzu finden Sie in den Kapiteln "Eigenschaftenseiten und Dialogfelder" und "Wichtige Begriffe: Zuordnungstabellen".

3.14 Komplettansicht (World View)

Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm angezeigt wird. Ein Rahmen zeigt an, welchen Diagrammausschnitt 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 Rechtecks, 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.



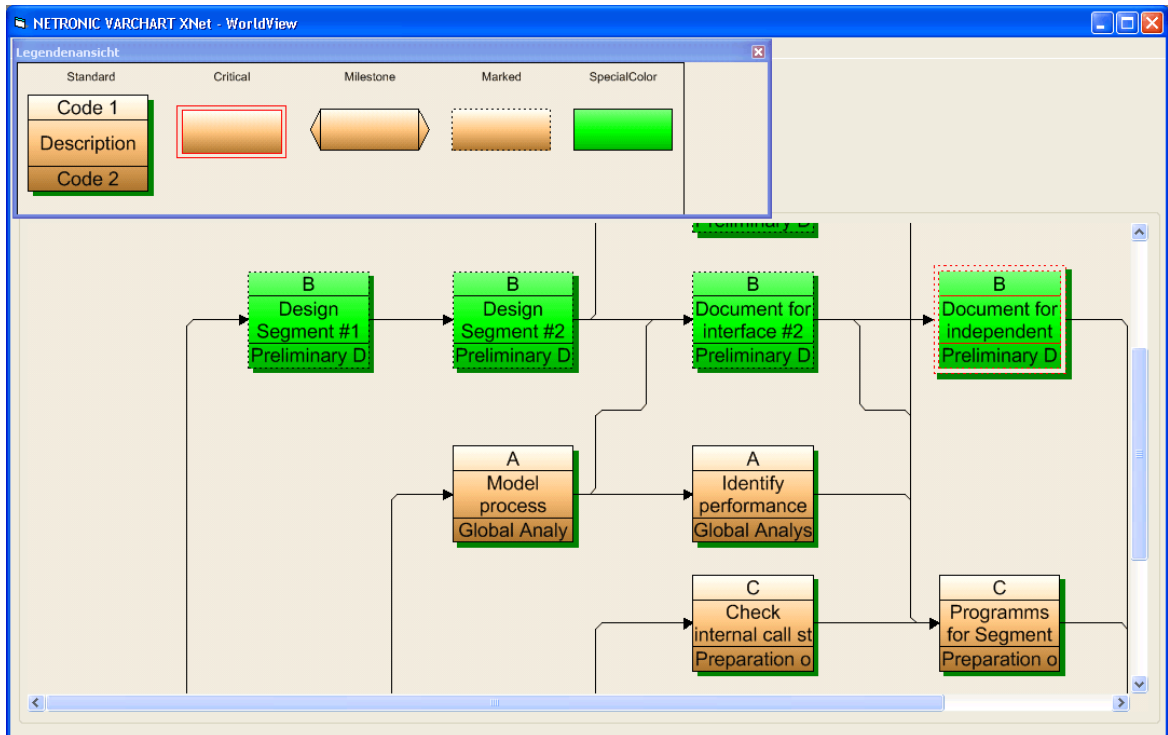
128 Wichtige Konzepte: Komplettansicht (World View)

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

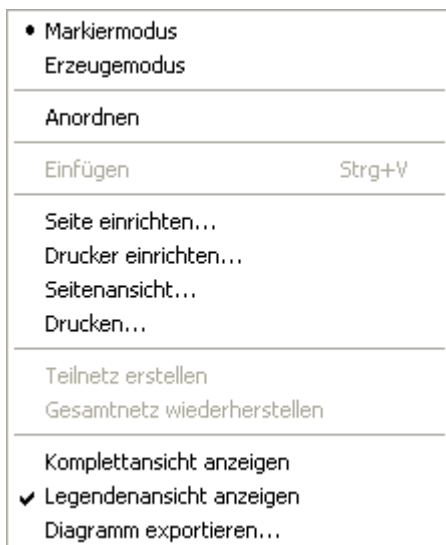
Alternativ können Sie die Optionen der Komplettansicht auch über die API-Eigenschaft **VcNet.VcWorldView** festlegen.

3.15 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 **VcBoundingBox**.

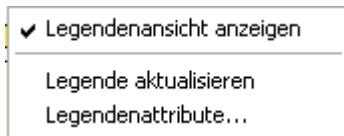


Zur Laufzeit können Sie über den Menüpunkt **Legendenansicht anzeigen** des Standard-Kontextmenüs die Legendenansicht ein- und ausschalten.



130 Wichtige Konzepte: Legendenansicht (Legend View)

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 **VcNet.VcLegendView** festlegen.

3.16 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 Knoten und deren Verbindungen und ganze Teilnetze verschoben werden. Dieser Modus wird entweder manuell über die VcNet-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.

> **Phantom während eines OLE-Drag-Vorgangs anzeigen**

Mit Hilfe des Kontrollkästchens **Phantom anzeigen** 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, neu attribuieren.

> **Eigenen Mauscursor anzeigen**

Mit Hilfe des Kontrollkästchens **Eigenen Mauscursor anzeigen** 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 das Kontrollkästchen **Eigenen Mauscursor anzeigen** 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 ausgeschaltetem Mauscursor möglich ist.

> **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 Ziellanwendung 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.

134 Wichtige Konzepte: OLE Drag & Drop

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.

3.17 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.

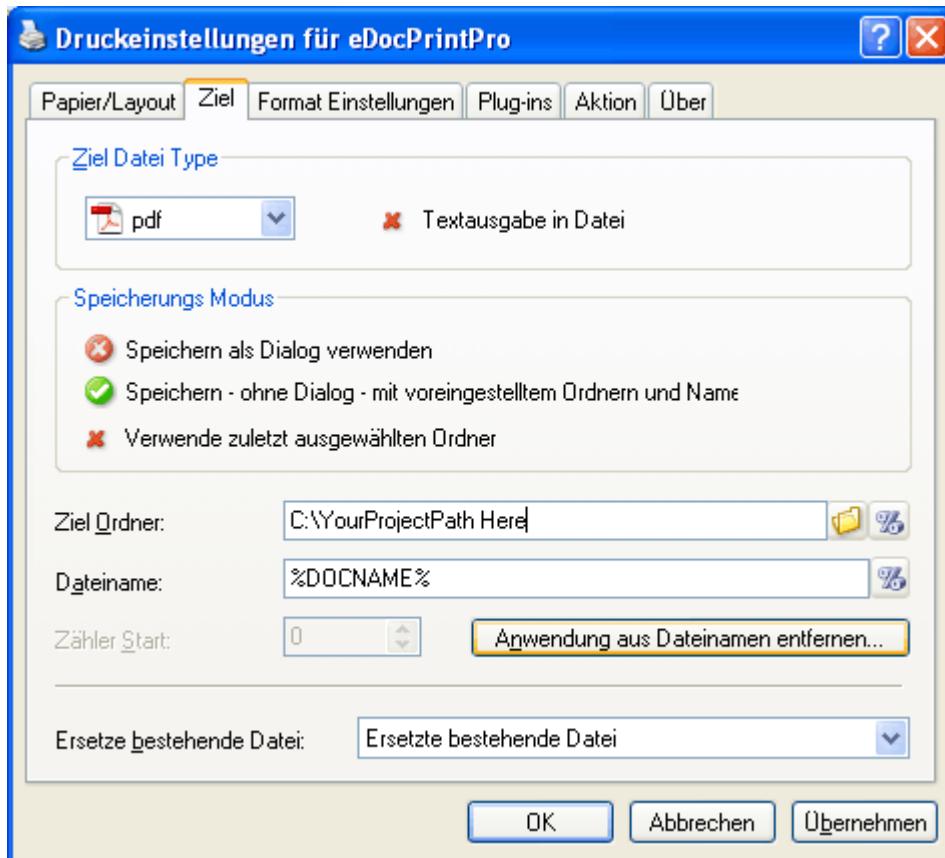
Die vorgenannten Anforderungen erfüllen z.B. der in der **Adobe Acrobat Suite** ab Version 6 enthaltene Druckertreiber [www.adobe.com] sowie der kostenlose Treiber **eDocPrintPro** [www.pdfprinter.at].

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

136 Wichtige Konzepte: Schreiben von PDF-Dateien

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

```
VcNet1.Printer.PrinterName = "eDocPrintPro"  
VcNet1.Printer.DocumentName = "abc.pdf"  
VcNet1.PrintEx
```

Ganz wenige Druckertreiber erfordern eine andere Code-Sequenz:

Code-Beispiel

```
VcNet1.Printer.PrinterName = "Win2PDF"  
VcNet1.PrintToFile "abc.pdf"
```

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

3.18 Sprachanpassung von Textausgaben

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

Setzen Sie dazu die VcNet-Eigenschaft **EnableSupplyTextEntryEvent** auf den Wert **True**, um das Ereignis zu aktivieren.

Code-Beispiel

```
VcNet1.EnableSupplyTextEntryEvent = True
```

Alternativ können Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **OnSupplyTextEntry-Ereignisse** aktivieren.

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

Code-Beispiel

```
Private Sub VcNet1_OnSupplyTextEntry(ByVal controlIndex As _
                                   VcNetLib.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
```

3.19 Statuszeilentext

Sie können das Ereignis **OnStatusLineText** verwenden, um Informationen über den mit der Maus berührten Knoten in einer Statusleiste bereitzustellen.

3.20 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 (Group, Node, None, LinkCollection) 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. Setzen Sie dazu die VcNet-Eigenschaft **ShowToolTip** auf den Wert True, um das Ereignis zu aktivieren.

Code-Beispiel

```
VcNet1.ShowToolTip = True
```

Alternativ können Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **OnToolTipText-Ereignisse** aktivieren.

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.

3.21 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.msp](http://www.microsoft.com/globaldev/getwr/steps/wrg_unicode.msp)). Unter **Start / Programme / Zubehör / Systemprogramme / Zeichentabelle** kann man sich in Windows 2000 und XP darüber informieren, welcher installierte Schriftarttyp welche Zeichen enthält.

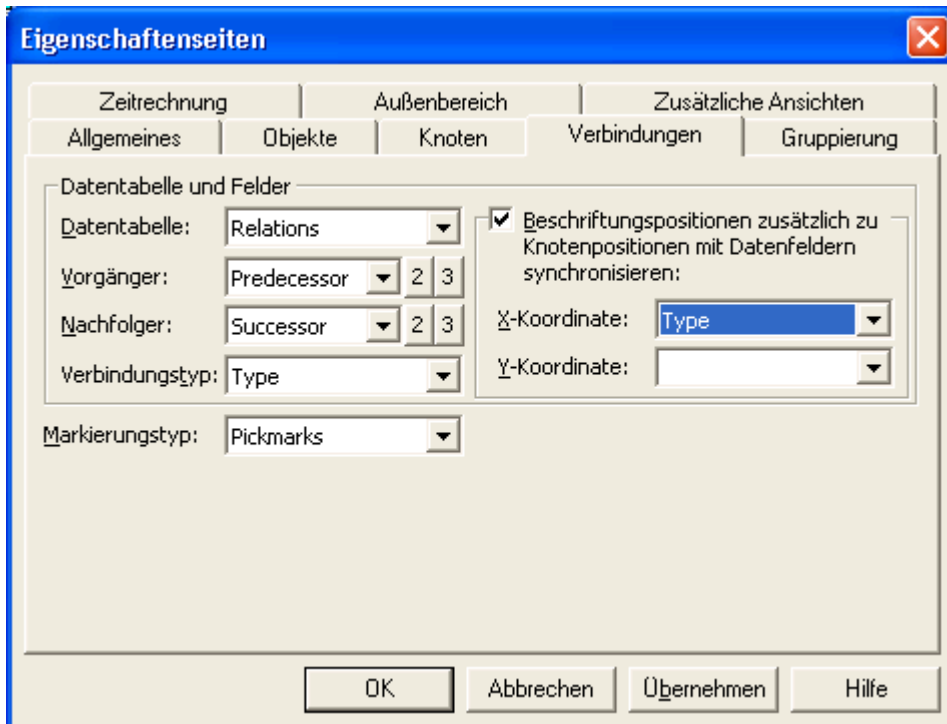
Beim Import von CSV-Dateien erkennt die Methode **VcGantt.Load** automatisch, ob die Datei im Unicode- oder ANSI-Format vorliegt.

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.

3.22 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.

> Datenfelder für die Verbindungen



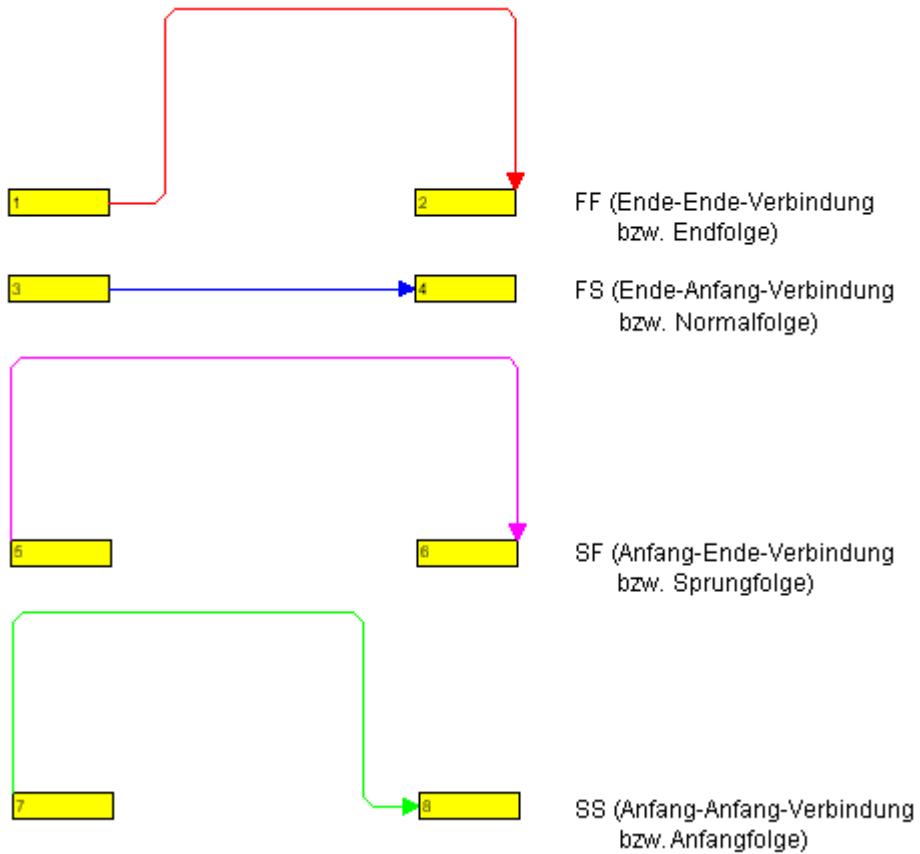
Auf der Eigenschaftenseite **Verbindungen** können Sie die Datenfelder festlegen, in denen die Identifizierung des Vorgänger- und des Nachfolgerknotens sowie des Verbindungstyps festgelegt ist.

> Verbindungstypen

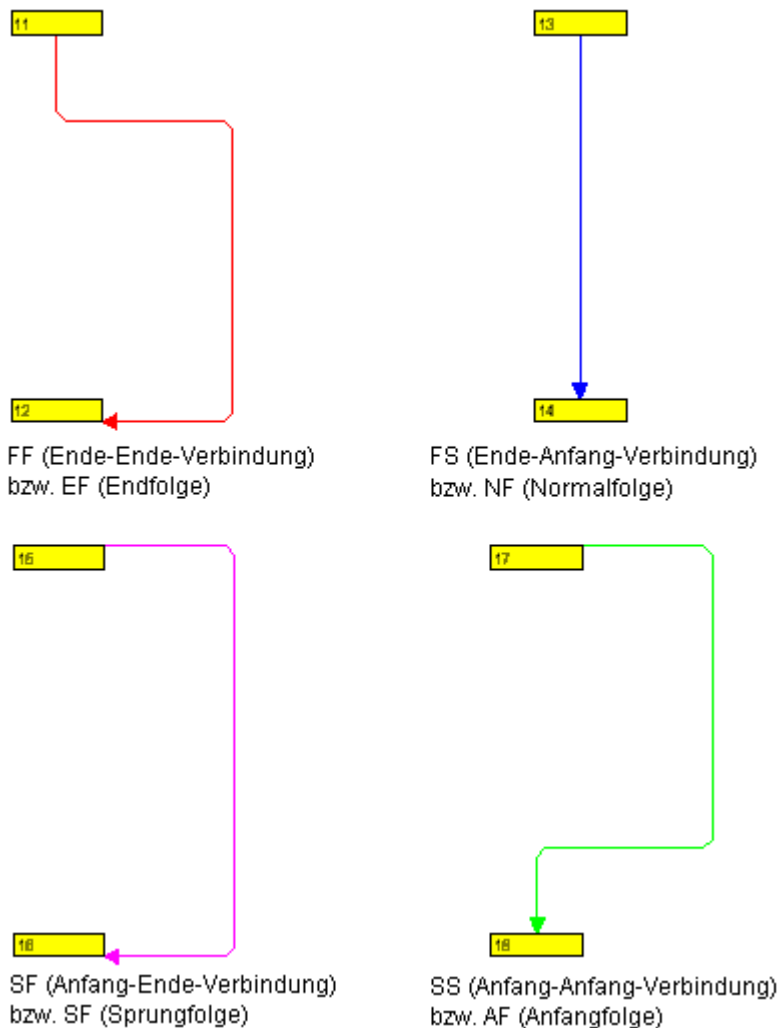
Unter **Verbindungstyp** legen Sie fest, aus welchem Datenfeld der Verbindungstyp gelesen werden soll.

Eine Übersicht über das Aussehen der verschiedenen Verbindungstypen bei den beiden Flussrichtungen geben die folgenden Abbildungen:

142 Wichtige Konzepte: Verbindungen



Flussrichtung von links nach rechts



Flussrichtung von oben nach unten

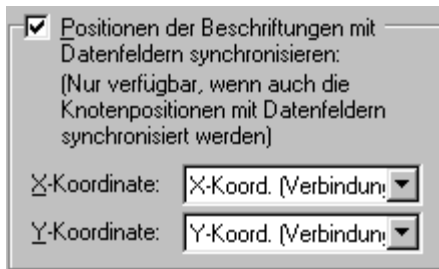
> Position von Verbindungsbeschriftungen

Damit Sie die Positionen der Verbindungsbeschriftungen eines Diagramms wieder laden können, müssen Sie diese mit den entsprechenden Datenfeldern synchronisieren. Aktivieren Sie dazu auf der Eigenschaftenseite **Verbindungen** das Kontrollkästchen **Positionen der Beschriftungen mit Datenfeldern synchronisieren** und geben Sie folgende Datenfelder an:

- für die X-Koordinate: "X-Koordinate (Verbindung)"
- für die Y-Koordinate: "Y-Koordinate (Verbindung)"

(Voraussetzung ist, dass Sie diese Datenfelder beim Einrichten der Schnittstelle entsprechend definiert haben. Siehe hierzu "Tutorium, Schnittstelle einrichten".)

144 Wichtige Konzepte: Verbindungen

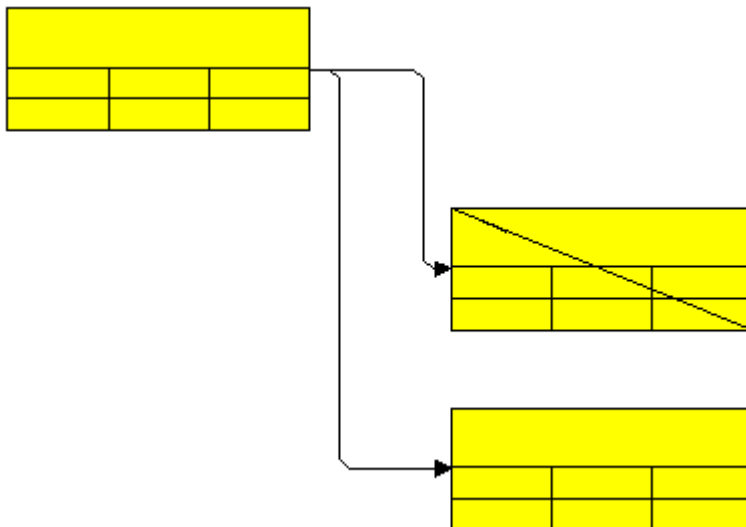


Die Werte dieser Datenfelder können Sie im Dialogfeld **Verbindung bearbeiten** ausfragen und ggf. bearbeiten.

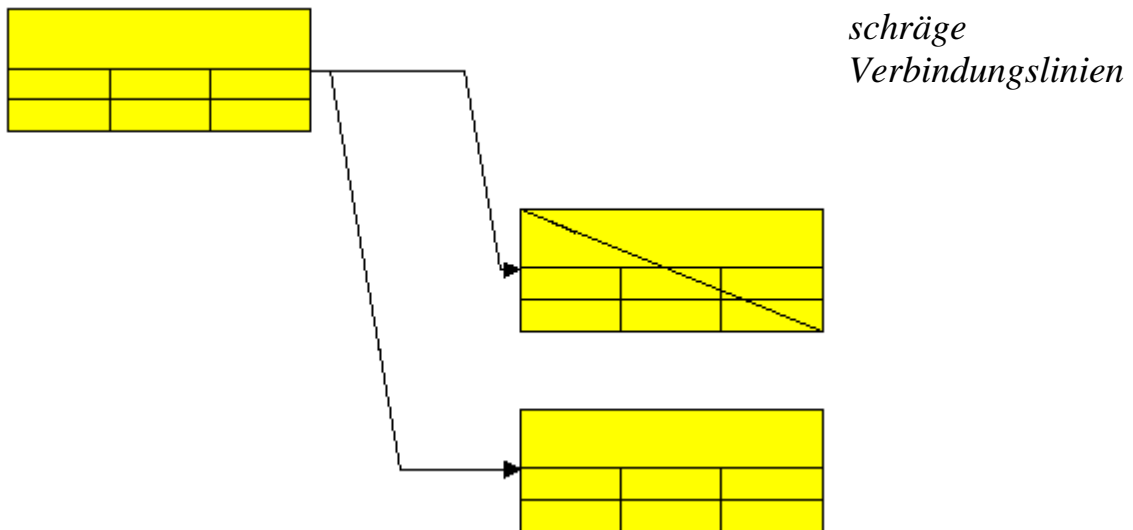
Ein Beispiel zu Positionen von Knoten- und Verbindungsbeschriftungen finden Sie im Abschnitt "Knoten" dieses Kapitels.

> **Orthogonale/Schräge Verbindungslinien**

Auf der Eigenschaftenseite **Allgemeines** können Sie unter **Schräge Tracks bei Verbindungen** festlegen, ob die Verbindungslinien orthogonal dargestellt werden sollen oder direkt an den horizontalen Linienstücken ansetzen und schräg verlaufen sollen. Alternativ können Sie dies über die VcNet-Eigenschaft **ObliqueTracksOnLinks** festlegen.



*orthogonale
Verbindungslinien*



> Verbindungen erzeugen

Der Anwender kann zur Laufzeit im Erzeugemodus Verbindungen zwischen Knoten ziehen, sofern auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten und Verbindungen zulassen** aktiviert ist. Wenn auf der Eigenschaftenseite **Allgemeines** außerdem die Option **Neue Verbindung bearbeiten** ist, erscheint beim Anlegen einer neuen Verbindung der Dialog **Verbindung bearbeiten**, in dem Sie die Daten der Verbindung direkt bearbeiten können.



Sie können Verbindungen auch über die API mit **InsertLinkRecord** anlegen. Jedes Neuanlegen einer Verbindung, gleich auf welche Weise, wird der Applikation mit dem Ereignis **OnLinkCreate** mitgeteilt.

> Verbindungen markieren

Zur Laufzeit können Sie im Markiermodus eine einzelne Verbindung markieren, indem Sie sie mit der linken Maustaste anklicken. Mehrere Verbindungen lassen sich sammeln und toggeln, indem Sie sie bei gedrückter Strg-Taste mit der linken Maustaste anklicken.

> Verbindungen bearbeiten

Sie können eine Verbindung bearbeiten, indem Sie sie mit der rechten Maustaste anklicken und dann im Kontextmenü den Befehl **Bearbeiten** wählen. Dann erscheint der Dialog **Verbindung bearbeiten**, in dem Sie alle Daten der Verbindung bearbeiten können.

> **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**

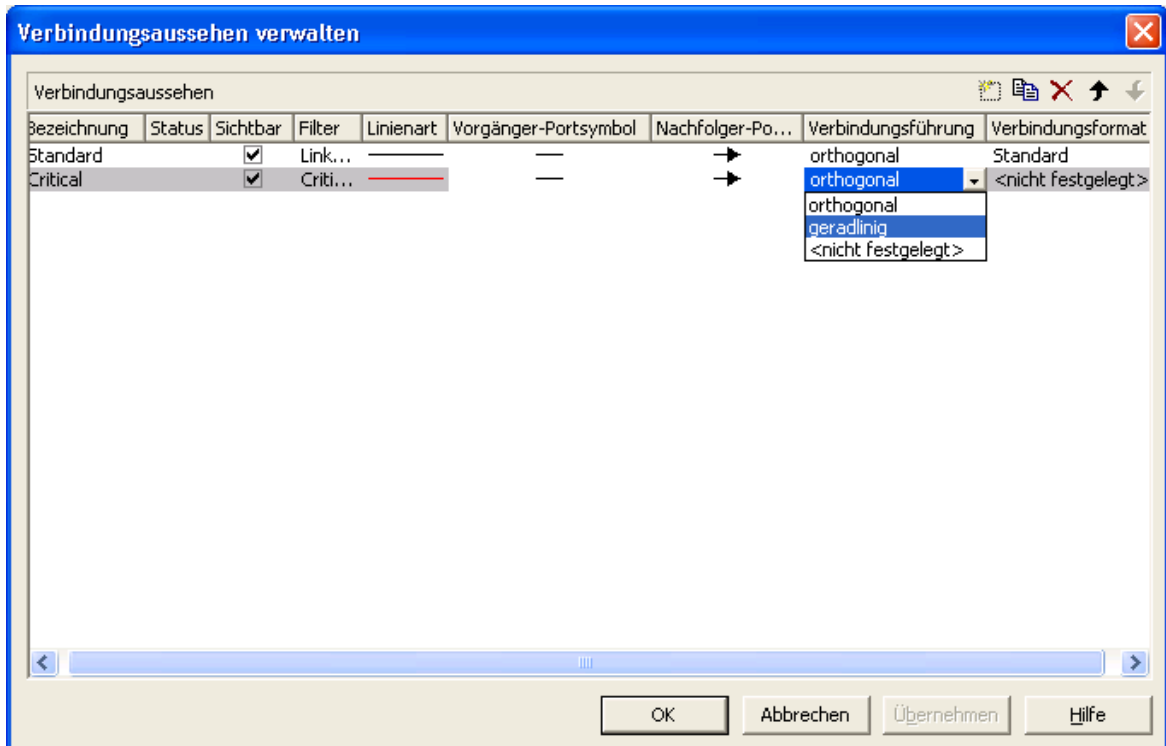
Auf folgende Ereignisse können Sie reagieren:

- **OnLinkCreate**
- **OnLinkCreateComplete**
- **OnLinkDelete**
- **OnLinkDeleteComplete**
- **OnLinkLClickCltn**
- **OnLinkLDbClickCltn**
- **OnLinkModify**
- **OnLinkModifyComplete**
- **OnLinkModifyEx**
- **OnLinkRClickCltn**

3.23 Verbindungsaussehen

Für die Verbindungen lassen sich im Dialog **Linienaussehen verwalten** unterschiedliche Verbindungsaussehen definieren, die den Verbindungen dynamisch über Filter zugewiesen werden.

> Neues Verbindungsaussehen definieren



> Verbindungsaussehen löschen

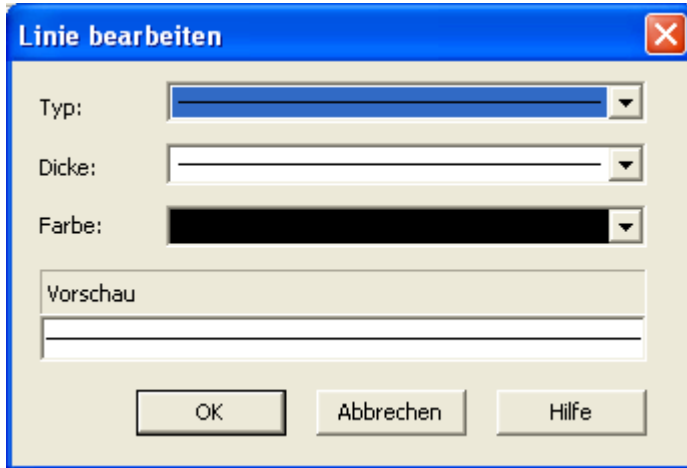
Mit Hilfe der **Entf**-Taste können Sie ein Verbindungsaussehen aus der **Aussehen**-Tabelle löschen.

> Filter festlegen

Um den Filter, der für ein bestimmtes Verbindungsaussehen verwendet werden soll, auszuwählen oder neu zu definieren, klicken Sie auf dessen **Filter**-Feld. Klicken Sie auf die Pfeil-Schaltfläche neben dem Filternamen, um die Kombobox mit allen verfügbaren Filtern zu öffnen, und wählen Sie hier einen Filter aus. Oder klicken Sie auf die **Bearbeiten**-Schaltfläche, um das Dialogfeld **Filter verwalten** zu öffnen. Hier können Sie Filter neu definieren, kopieren, bearbeiten und löschen. Änderungen, die Sie an einem Filter durchführen, gelten für jedes Verbindungsaussehen, das mit diesem Filter verbunden ist.

> **Verbindungslinien festlegen**

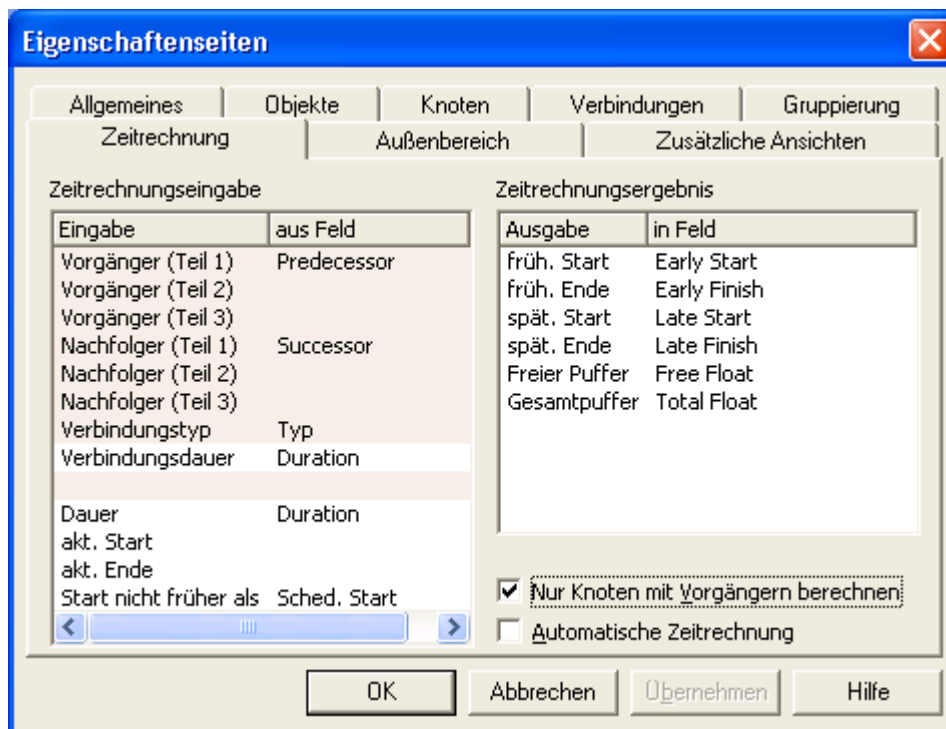
Wenn Sie auf den Eintrag des Feldes **Linienart** klicken, erscheint eine **Bearbeiten**-Schaltfläche, über die Sie das Dialogfeld **Linie bearbeiten** öffnen können. Hier können Sie das Aussehen der Verbindungslinien festlegen.



3.24 Zeitrechnung

Mit dem Scheduler von VARCHART XNet 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 XNet an Ihre Schnittstelle anpassen, indem Sie festlegen, welche Datenfelder für die Eingabe (**Zeitrechnungseingabe**) und die Ausgabe (**Zeitrechnungsergebnis**) des Schedulers verwendet werden sollen. Außerdem können Sie die Zeiteinheit festlegen, die für die Berechnung der Dauer in den entsprechenden Datenfeldern in Knoten und Verbindungen verwendet werden soll.



Unter **Zeitrechnungseingabe** können Sie für jede Eingabe auswählen, aus welchem Feld sie entnommen werden soll. Als Eingaben für die Zeitrechnung verwendet der Scheduler Datenfelder der Maindata- und der Relations-Tabelle. Die Ausgabe des Schedulers erfolgt nur in Datenfelder der Maindata-Tabelle.

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**, **Nachfolger** und **Verbindungstyp** können in der **Zeitrechnungseingabe**-Tabelle nicht bearbeitet werden. Sie

geben nur die auf der Eigenschaftenseite **Verbindungen** vorgenommenen Festlegungen wieder.

Die Ausgaben werden wiederum in Datenfelder der Schnittstelle geschrieben. Als Ausgaben stehen zur Verfügung: **früh. Start, früh. Ende, spät. Start, spät. Ende, Freier Puffer** und **Gesamtpuffer**. Jeder dieser Ausgaben können Sie ein Feld aus der in der Datendefinition vereinbarten Liste von Feldern zuweisen.

Es gibt folgende Möglichkeiten, die Zeitrechnung zu beeinflussen:

1. Sie können einen Projektstart vorgeben. Das erreichen Sie über die API mit der VcNet-Methode **ScheduleProject**:

VcNet1.ScheduleProject "04.05.2000", 0

Mit der Methode **ScheduleProject** können Sie eine Vorwärts- und Rückwärtsberechnung des aktuellen Projekts durchführen. Bei Übergabe des Starttermins wird zunächst eine Vorwärtsberechnung, dann eine Rückwärtsberechnung durchgeführt. Bei Übergabe des Endtermins wird zunächst eine Rückwärtsberechnung, dann eine Vorwärtsberechnung durchgeführt. Es können auch Anfangs- und Enddatum übergeben werden, die Vorgänge erhalten dann entsprechende Pufferzeiten.

Mögliche Wahl der Parameter für die Methode ScheduleProject:

Anfang	Ende
Termin 1	0
0	Termin 2
Termin 1	Termin 2

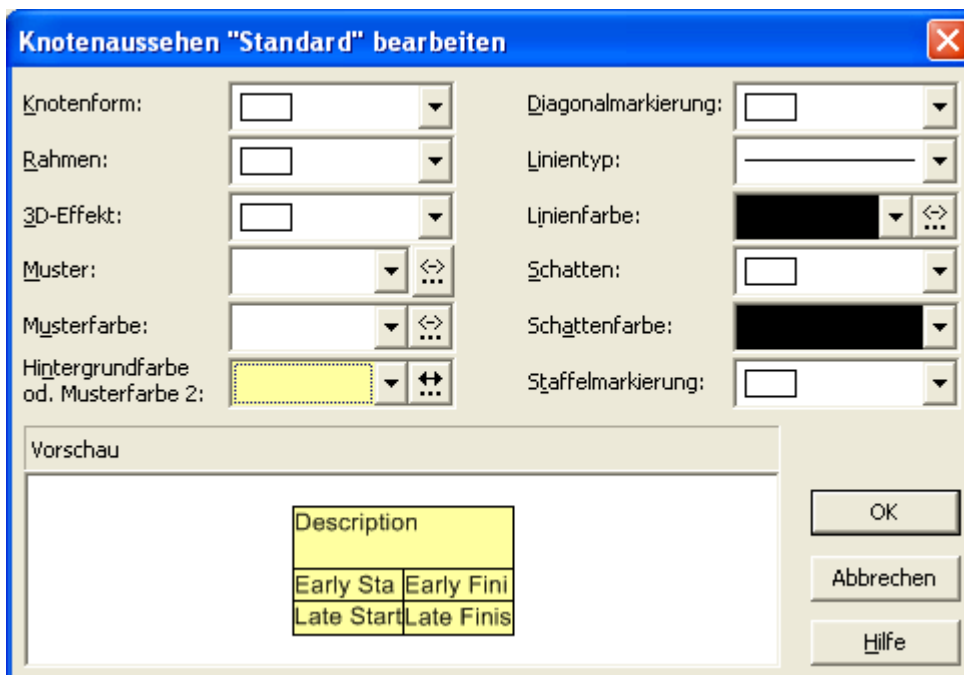
2. Sie können aktuelle Start- bzw. Endtermine angeben. Die Knoten sind dann unverrückbar.
3. Sie können für die Bedingungen "Start nicht früher als" und "Ende nicht später als" Referenztermine angeben. Dazu wird auf der Eigenschaftenseite **Zeitrechnung** in der linken Tabelle für die entsprechenden Werte auch jeweils ein Feld aus der Datendefinition festgelegt. Dann liegt der früheste Start eines Knoten nicht vor dem angegebenen Termin bzw. das späteste Ende nicht nach dem angegebenen Termin.

3.25 Zuordnungstabellen

Zuordnungstabellen dienen dazu, bestimmte Eigenschaften datenfeld-abhängig festzulegen, ohne unter Verwendung vieler Filter eine Vielzahl ähnlicher Layer definieren zu müssen. Auch das Knotenaussehen und das Knotenformat können über Zuordnungstabellen datenabhängig gestaltet werden.

> Knotenaussehen datenabhängig festlegen

Für jedes Knotenaussehen können die Hintergrundfarbe und die Linienfarbe über Zuordnungstabellen festgelegt werden. Klicken Sie dazu im Dialogfeld **Knotenaussehen bearbeiten** auf die zweite Schaltfläche für die **Hintergrundfarbe** bzw. **Linienfarbe** (☰).

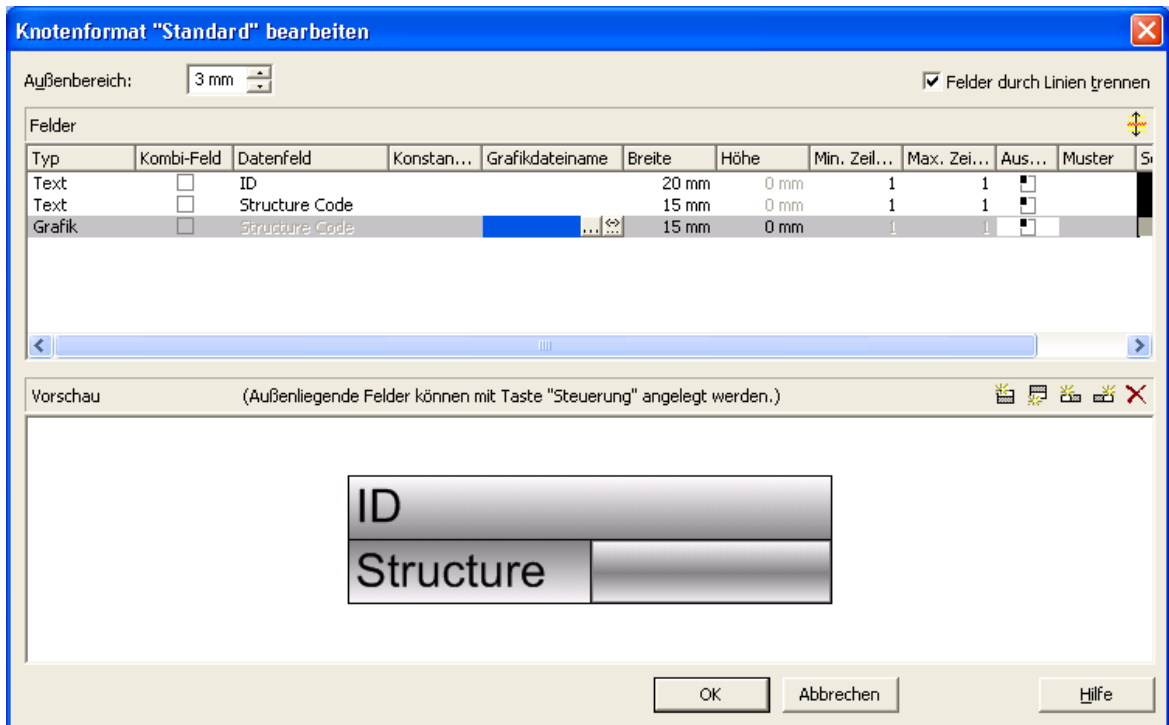



Sie gelangen dann in den Dialog **Zuordnung einstellen**.


> Grafik eines Knotenformats datenabhängig festlegen

In den Feldern der Knotenformate können Grafiken über eine Zuordnungstabelle datenabhängig ausgegeben werden.

152 Wichtige Konzepte: Zuordnungstabellen

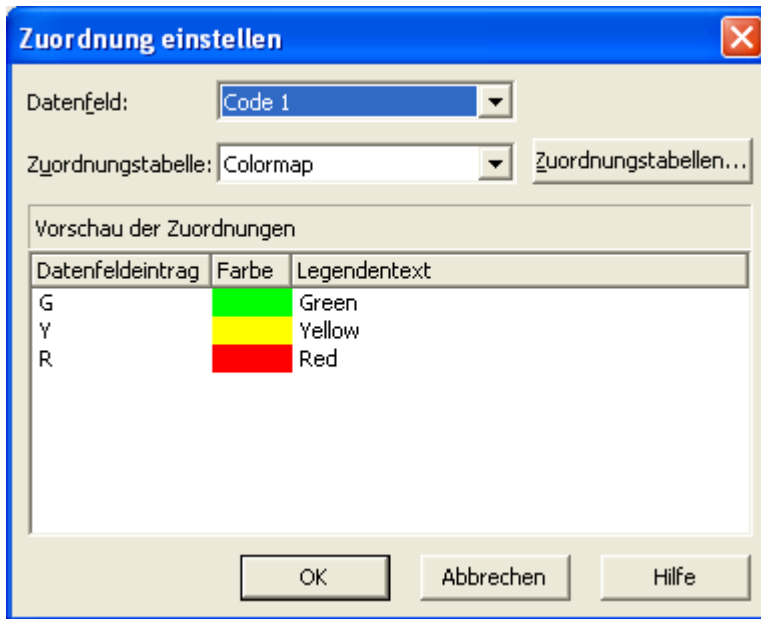


 Um eine Zuordnung zwischen den Einträgen eines Datenfeldes vom Typ Grafik und Grafikdateien herzustellen, klicken Sie im Feld **Grafikdateiname** auf die 2. Schaltfläche (**Zuordnungen einstellen**). Der gleichnamige Dialog erscheint dann.

Wenn Sie dort eine Zuordnung vorgenommen haben, erscheint ein Symbol () im Feld **Grafikdateiname**, sobald Sie die entsprechende Zeile verlassen.

> Zuordnung einstellen

Im Dialogfeld **Zuordnung einstellen** können Sie festlegen, dass in einem bestimmten Knotenformatfeld vom Typ Grafik datenabhängig Grafikdateien dargestellt werden sollen, bzw. dass die Hintergrundfarbe eines bestimmten Knotenaussehens datenabhängig sein soll.



Wählen Sie dazu in der ersten Kombobox das **Datenfeld**, von dessen Einträgen die Grafikdatei des zu bearbeitenden Knotenformatfeldes bzw. die Hintergrundfarbe des zu bearbeitenden Knotenaussehens abhängen soll. Wählen Sie dann in der zweiten Kombobox die **Zuordnungstabelle**, die den einzelnen Datenfeldeinträgen eine Grafikdatei bzw. eine Farbe und einen Legendentext zuordnet.

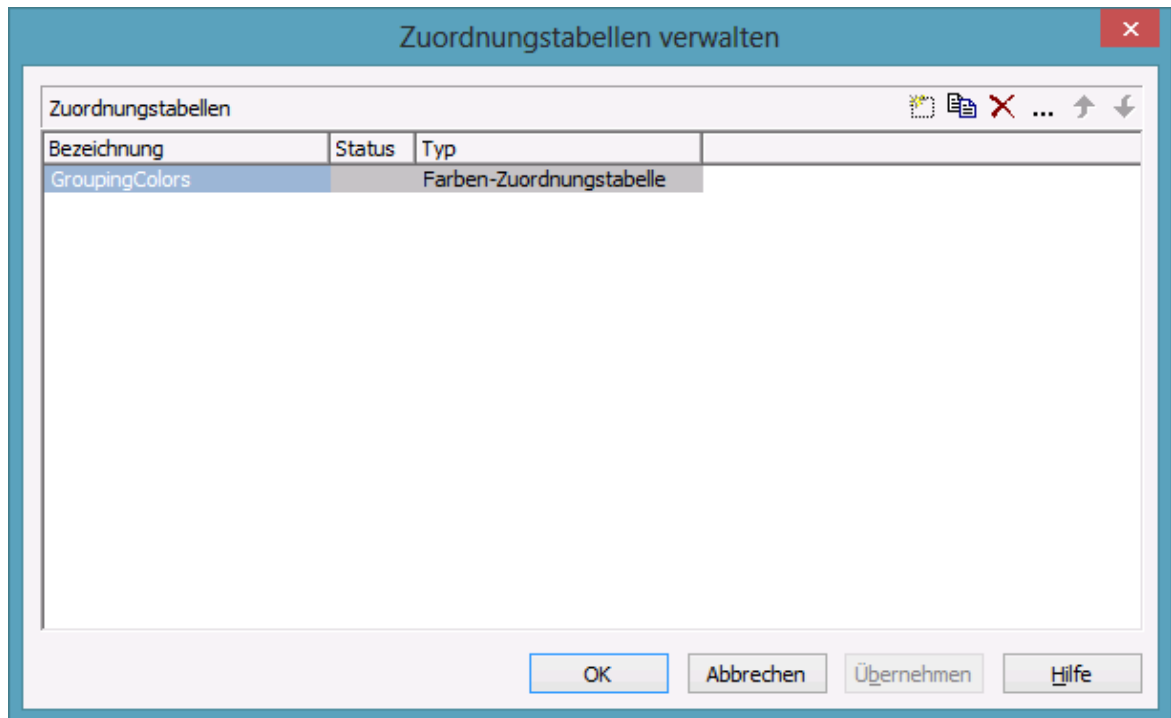
In der **Vorschau der Zuordnungen** wird dargestellt, wie die gewählte Zuordnungstabelle den einzelnen Datenfeldeinträgen eine Grafikdatei bzw. eine Hintergrundfarbe und einen Legendentext zuordnet.

> **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.

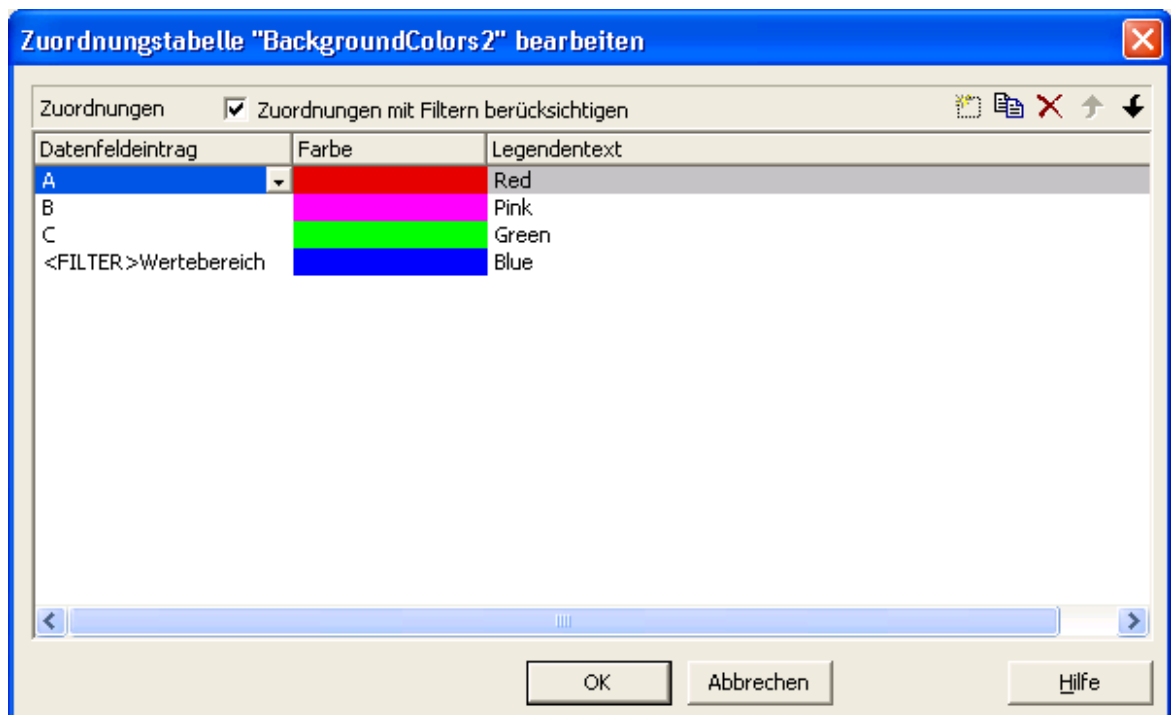
Sie können aus verschiedenen Typen von Zuordnungstabellen auswählen, je nachdem, ob den Datenfeldinhalten Farben, Muster, Grafiken, Schrifttypen, Längen oder Nummern zugeordnet werden sollen.

154 Wichtige Konzepte: Zuordnungstabellen



> 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.

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.


4 Eigenschaftenseiten und Dialogfelder

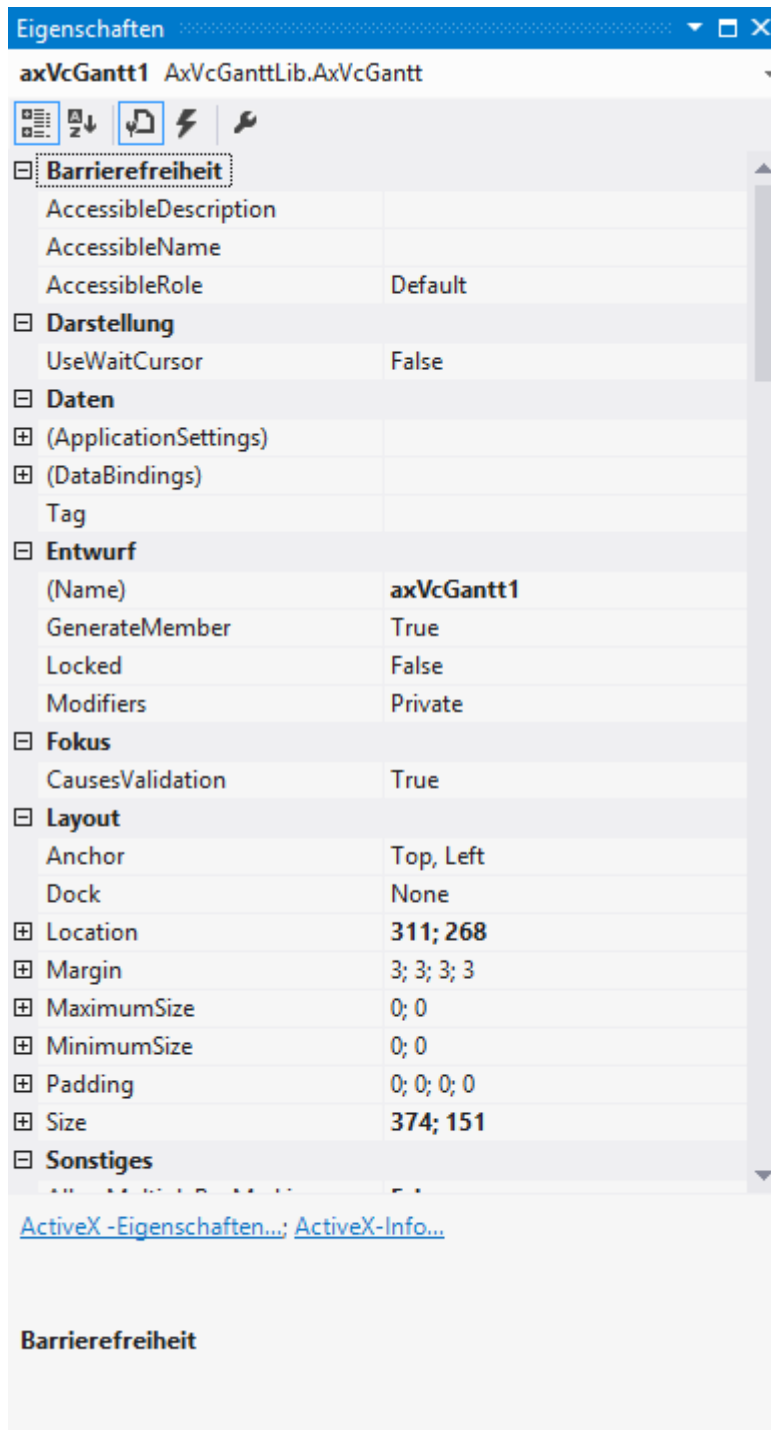
4.1 Allgemeines

Durch die Eigenschaftenseiten kann VARCHART XNet 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.

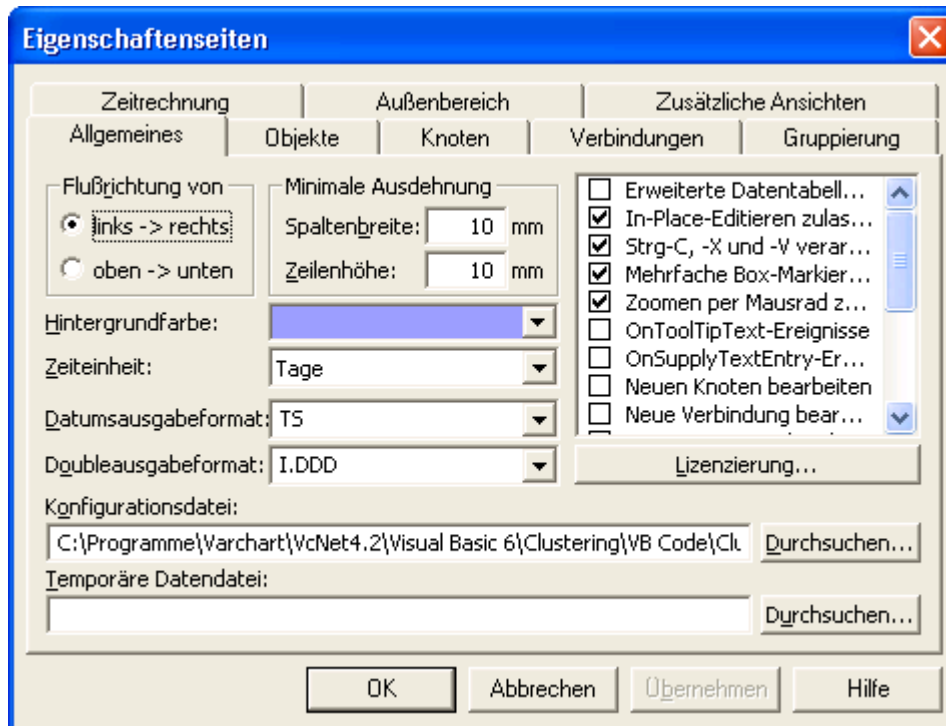
oder

- Im Eigenschaftfenster (kann mit F4 geöffnet werden) klicken Sie in der Symbolleiste auf das ganz rechts stehende Symbol .



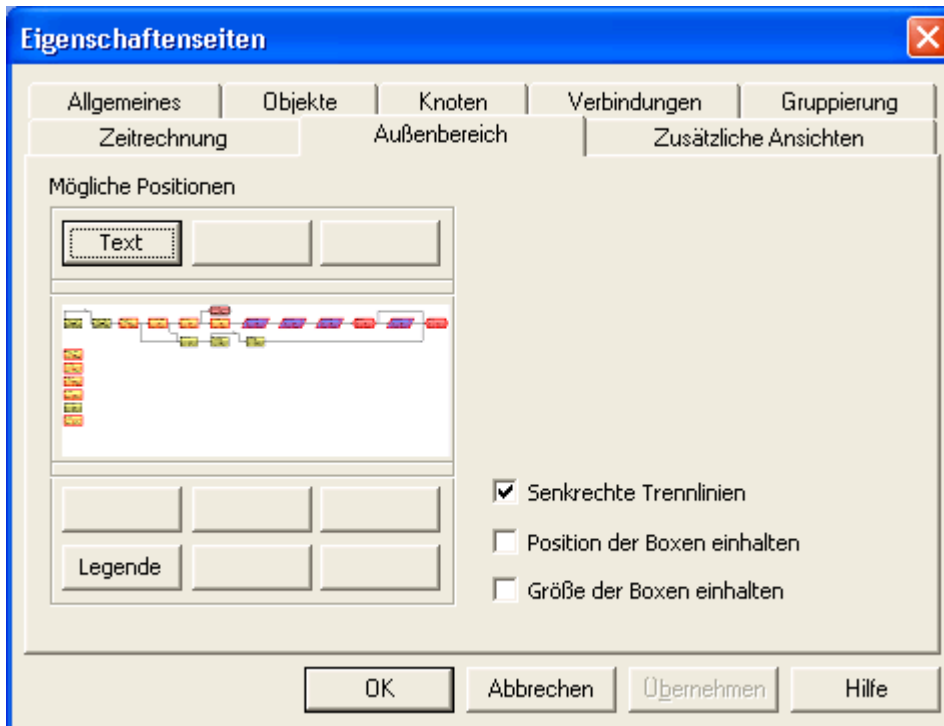
Nähere Informationen zu jeder Eigenschaftenseite bzw. jedem Dialogfeld erhalten Sie, indem Sie auf die **Hilfe**-Schaltfläche klicken oder die F1-Taste drücken. Sie erhalten dann direkt die Online-Hilfe zu der Eigenschaftenseite bzw. dem Dialogfeld.

4.2 Eigenschaftenseite "Allgemeines"



Auf dieser Eigenschaftenseite können Sie allgemeine Einstellungen für VARCHART XNet vornehmen.

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.

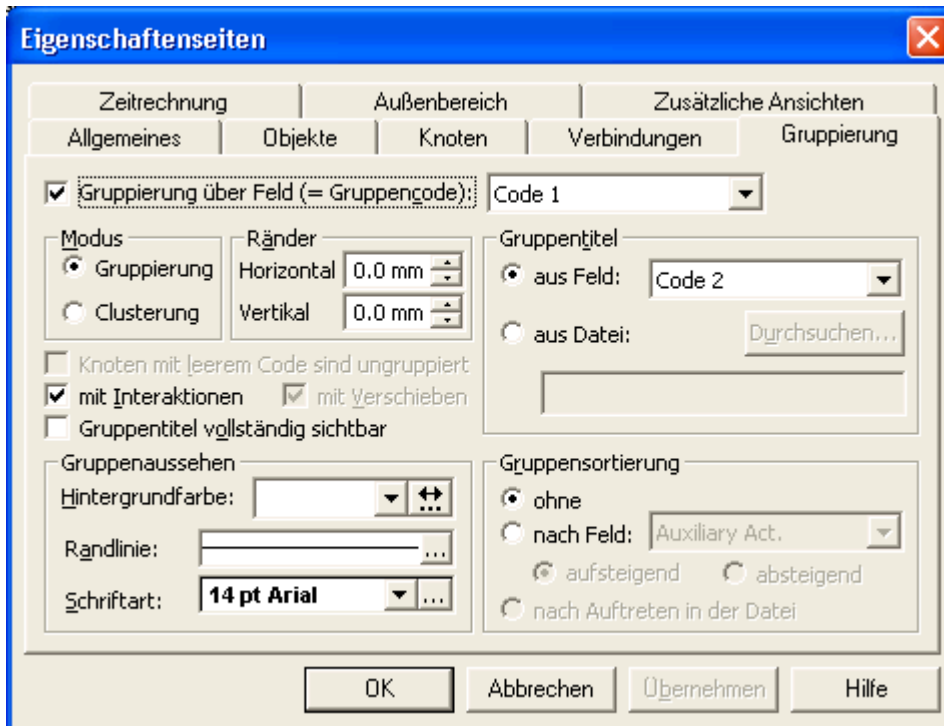
Position der Boxen einhalten

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 "Gruppierung"



Gruppierung über Feld (= Gruppencode)

Aktivieren Sie dieses Kontrollkästchen, wenn die Knoten gruppiert dargestellt werden sollen. Nur wenn dieses Kontrollkästchen aktiviert ist, sind die übrigen Optionen dieser Eigenschaftenseite aktiviert.

Legen Sie hier fest, nach welchem Feld gruppiert werden soll. Das Feld, das Sie hier auswählen, wird **Gruppencode** genannt. Alle Vorgänge, die in dem hier gewählten Feld denselben Eintrag haben, werden in einer Gruppe zusammengefasst.

Modus

Wählen Sie hier den Modus:

- **Gruppierung:** normale Gruppendarstellung (die Breite und Höhe jeder Gruppe wird durch die Positionen der Knoten bestimmt; jede Gruppe nimmt jeweils die gesamte Breite bzw. Höhe des Netzdiagramms ein)
- **Clustering:** Die Gruppen umschließen die darin enthaltenen Knoten möglichst platzsparend, wobei die Gruppen frei im Netzdiagramm verteilt sind.

Ränder

Stellen Sie hier die Breite der horizontalen bzw. vertikalen Ränder der Gruppen ein. Möglich sind 0 bis 9,9 mm.

Knoten mit leerem Code sind ungruppiert

(nur für Modus Clusterung) Wenn dieses Kontrollkästchen aktiviert ist, werden Knoten, deren Gruppencode der leere String "" ist, nicht innerhalb einer Gruppe dargestellt. Andernfalls werden sie in einer eigenen Gruppe für Vorgänge ohne Gruppencode-Angabe dargestellt.

mit Interaktionen

Wenn das Kontrollkästchen aktiviert ist, können die Gruppen interaktiv kollabiert bzw. expandiert werden (mit Hilfe des Plus- bzw. Minus-Zeichens neben dem Gruppentitel).

mit Verschieben

(nur für Modus Clusterung) Wenn das Kontrollkästchen aktiviert ist, können die Cluster interaktiv verschoben werden.

Gruppentitel vollständig sichtbar

Wenn diese Option ausgewählt ist, sind beim horizontalen Rollen die Gruppentitel stets sichtbar.

Hintergrundfarbe

Wählen Sie hier die Hintergrundfarbe für die Gruppen.

Randlinie

Hier wird das Aussehen der Randlinien der Gruppen angezeigt. Um es zu bearbeiten, klicken Sie auf die **Bearbeiten**-Schaltfläche. Sie gelangen dann in das Dialogfeld **Linie bearbeiten**, in dem Sie Farbe, Typ und Dicke der Linien, mit denen die einzelnen Gruppen getrennt werden, festlegen können.

Schriftart

In diesem Feld werden Schriftart und Schriftfarbe der Gruppentitel angezeigt. Um die Schriftfarbe zu bearbeiten, klicken Sie auf die Pfeil-Schaltfläche, um

die Farbauswahltable zu öffnen. Um die Schriftart zu bearbeiten, klicken Sie auf die **Bearbeiten**-Schaltfläche. Sie gelangen dann in das Windows-Dialogfeld **Schriftart**.

Gruppentitel aus Feld

Aktivieren Sie dieses Kontrollkästchen, damit die Gruppentitel aus dem Datenfeld genommen werden, das Sie hier auswählen. Dieses Feld muss nicht notwendigerweise mit dem Gruppencode identisch sein. Doch damit die Gruppen sinnvoll beschriftet werden, sollten die Einträge im **Gruppencode**-Feld und im **Gruppentitel**-Feld miteinander korrespondieren.

Gruppentitel aus Datei

Wenn Sie dieses Kontrollkästchen aktivieren, werden die Gruppentitel aus der Datei übernommen, die Sie hier auswählen. Über die **Durchsuchen**-Schaltfläche gelangen Sie in das Windows-Dialogfeld **Gruppentiteldatei auswählen**, in dem Sie die Datei öffnen können, aus der die Gruppentitel entnommen werden sollen.

Standardmäßig werden die Gruppentitel aus einer Datei vom Typ *.txt gelesen. Sie können aber auch einen anderen Dateityp dafür festlegen.

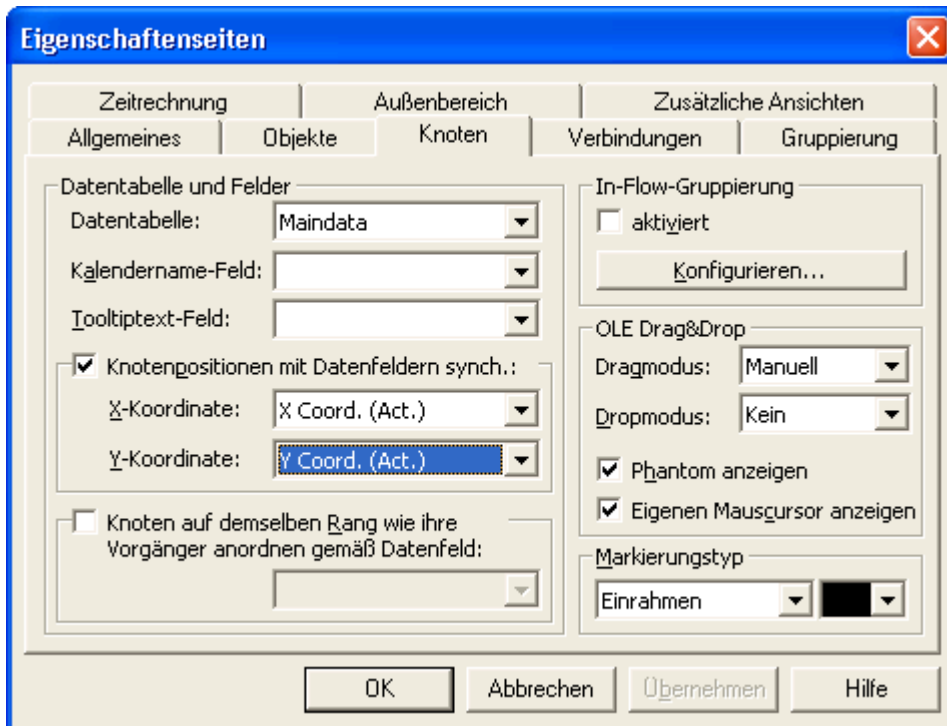
Wird ein relativer Dateiname angegeben, so wird die Datei zur Laufzeit zuerst in dem Verzeichnis gesucht, das in der VARCHART-ActiveX-Eigenschaft **FilePath** gesetzt ist. Wird sie dort nicht gefunden, wird sie zuerst im gerade aktiven Arbeitsverzeichnis der Applikation und dann im Installationsverzeichnis des VARCHART-ActiveX-Steuerelements gesucht.

Gruppensortierung

Hier können Sie festlegen, ob und ggf. nach welchem Kriterium die Gruppen sortiert werden. Sie können zwischen den folgenden Alternativen wählen:

- **ohne:** Die Gruppen werden nicht sortiert.
- **nach Feld:** Sie können ein Feld angeben, nach dem die Gruppen sortiert werden sollen, und festlegen, ob **aufsteigend** oder **absteigend** sortiert werden soll.
- **nach Auftreten in der Datei:** Die Gruppen werden in der Reihenfolge sortiert, in der sie in der Gruppentiteldatei auftreten.

4.5 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 **VcNet.NodesDataTableName** festgelegt werden.

Kalendernamen-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 auf der Eigenschaftenseite **Allgemeines** die Option **Scheduler benutzt internen Kalender** aktiviert sein. Außerdem müssen die Kalender vor dem Laden der Vorgänge erzeugt worden sein.

Diese Option kann auch über die Eigenschaft **VcNet.NodeCalendarName-DataFieldIndex** 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 **VcNet.NodeToolTipTextField** gesetzt werden.

Knotenpositionen mit Datenfeldern synchronisieren

Aktivieren Sie dieses Kontrollkästchen, um die Knotenpositionen mit Datenfeldern Ihrer Wahl zu synchronisieren. Wählen Sie dann je ein Datenfeld aus, aus dem die X- bzw. Y-Koordinate jeder Knotenposition gelesen und wohin sie zurückgeschrieben werden soll. Die Synchronisierung der Knotenpositionen mit den Datenfeldern ist dann erforderlich, wenn die Knotenpositionen nach dem Schließen Ihres Projekts wiederhergestellt werden müssen.

Knoten auf demselben Rang wie ihre Vorgänger anordnen gemäß Datenfeld

Wenn bestimmte Knoten auf demselben Rang positioniert werden sollen wie ihre Vorgänger, aktivieren Sie dieses Kontrollkästchen. Wählen Sie dann das Datenfeld aus, dessen Eintrag bestimmt, ob ein bestimmter Knoten auf demselben Rang positioniert wird wie sein Vorgänger (z. B. das Datenfeld "Hilfsknoten"). Dieses Datenfeld müssen Sie ggf. auf der Eigenschaftenseite **Datendefinition** erst definieren. Es kann die Werte 0, 1, 2 oder 3 annehmen.

Wert im gewählten Datenfeld	Flussrichtung von oben nach unten	Flussrichtung von links nach rechts
0	Der Rang des Hilfsknotens wird	Der Rang des Hilfsknotens wird

Wert im gewählten Datenfeld	Flussrichtung von oben nach unten	Flussrichtung von links nach rechts
	nicht reduziert.	nicht reduziert.
1	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links oder rechts neben seinem Vorgänger statt darunter.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über oder unter seinem Vorgänger statt daneben.
2	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann links von seinem Vorgänger	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann über seinem Vorgänger.
3	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann rechts von seinem Vorgänger.	Der Rang des Hilfsknotens wird um 1 reduziert. Der Hilfsknoten steht dann unter seinem Vorgänger.

Zur Erläuterung: Unter dem Rang eines Knotens versteht man eine Zahl, die folgendermaßen definiert ist: Der Rang eines Knotens ohne Vorgänger ist 1. Der Rang eines Knotens mit Vorgängern ist gleich 1 plus Rang desjenigen seiner Vorgänger, der den höchsten Rang besitzt.

(Weitere Informationen finden Sie im Kapitel "Wichtige Begriffe: Knoten".)

In-Flow-Gruppierung

Über die Schaltfläche **Konfigurieren** lässt sich das Dialogfeld **In-Flow-Gruppierung bearbeiten** öffnen. Wenn das **aktiviert**-Kontrollkästchen aktiviert ist, ist die In-Flow-Gruppierung bei Programmstart aktiv.

Dragmodus

Mit dem OLE-Dragmodus können Sie festlegen, ob das Ziehen eines Knotens über die Grenze des VARCHART-XNet-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-XNet-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 XNet 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 XNet 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.

Dropmodus

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

Mögliche Alternativen sind:

- **Kein:** Knoten aus einer anderen VARCHART-XNet-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.

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 **VcNet.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 **VcNet.OLEDragWithOwn-MouseCursor** festgelegt werden.

Markierungstyp

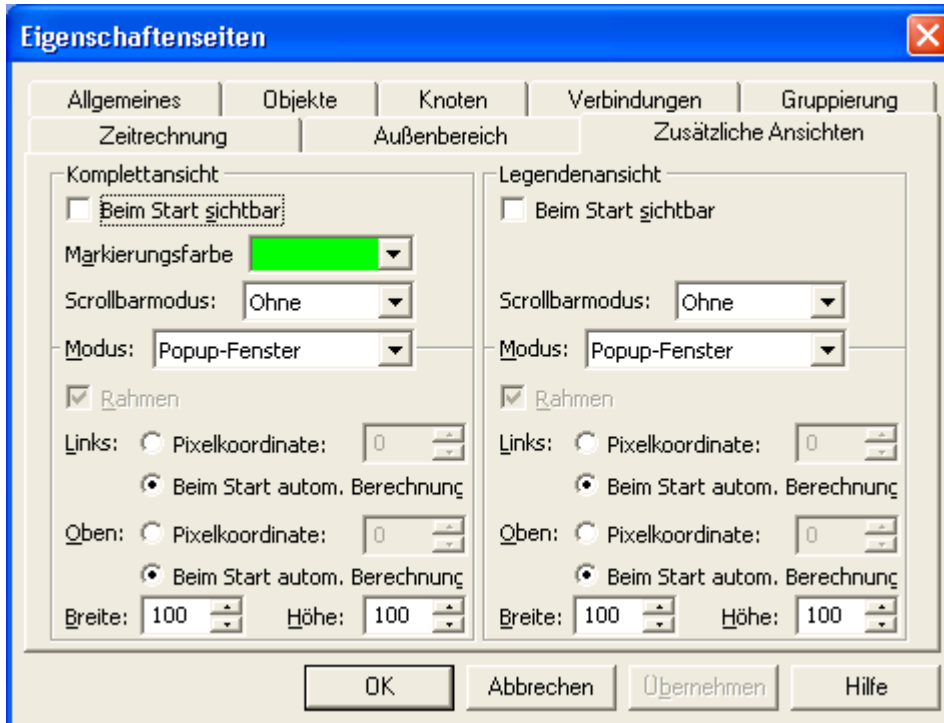
Wählen Sie hier aus, ob und welche Art der interaktiven Markierung von Knoten dem Anwender geboten werden soll. Folgende Alternativen stehen zur Verfügung:

- ohne
- Einrahmen
- Einrahmen innen
- Invertieren
- Pickmarks
- Pickmarks innen

Hinweis: Wenn Sie den Markierungstyp "Ohne" ausgewählt haben, werden Knoten nicht grafisch markiert.

Weiterhin können Sie hier eine Farbe für die Markierung auswählen.

4.6 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 angezeigt wird. Ein Rahmen darin zeigt an, welchen Ausschnitt des Diagramms das Hauptfenster gerade anzeigt.

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 **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.

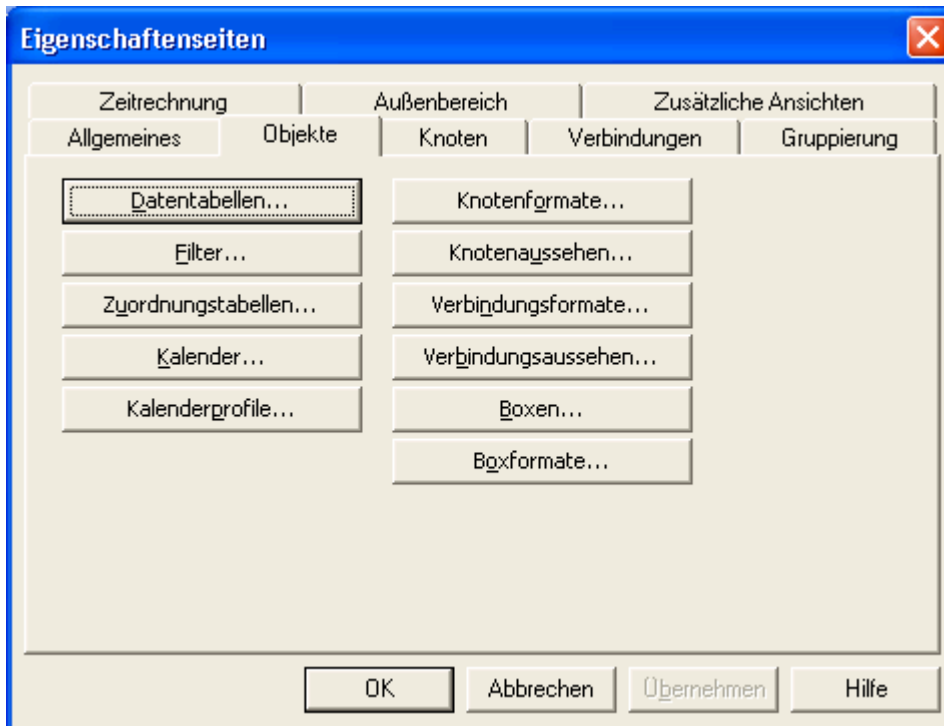
Breite

*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.

4.7 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**.

Knotenformate

Über diese Schaltfläche öffnen Sie das Dialogfeld **Knotenformate verwalten**.

Knotenaussehen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Knotenaussehen verwalten**.

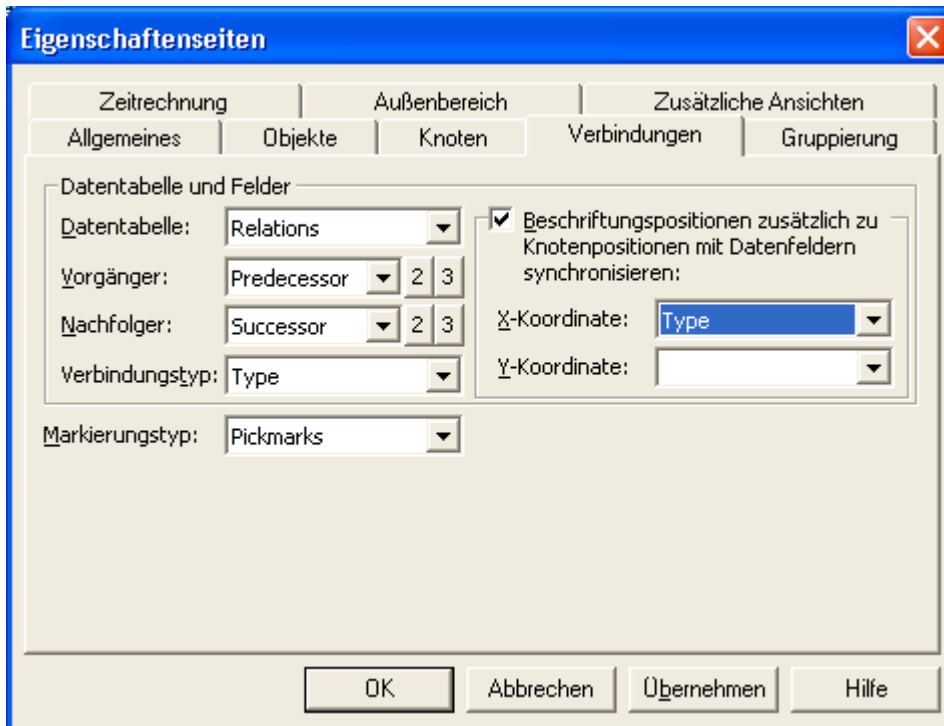
Boxen

Über diese Schaltfläche öffnen Sie das Dialogfeld **Boxen verwalten**.

Boxformate

Über diese Schaltfläche öffnen Sie das Dialogfeld **Boxformate verwalten**.

4.8 Eigenschaftenseite "Verbindungen"



Auf dieser Eigenschaftenseite können Sie wählen, ob die Verbindungen zwischen Knoten dargestellt werden, und ihr Aussehen festlegen.

Datentabelle

Wählen Sie hier eine Datentabelle aus, die die Felder der Verbindungen enthält. Diese Option kann auch über die Eigenschaft **VcNet.LinksDataTableName** festgelegt werden.

Vorgänger

Wählen Sie hier das Datenfeld bzw. die Datenfelder aus der zuvor eingestellten Datentabelle aus, in dem die Identifizierung des Vorgängerknotens der Verbindung enthalten ist/enthalten sind.

Nachfolger

Wählen Sie hier das Datenfeld aus der **Relations**-Tabelle aus, in dem die Identifizierung des Nachfolgerknotens der Verbindungen enthalten ist.

Verbindungstyp

Wählen Sie hier das Datenfeld aus, in dem die Information über den Typ von Verbindungen enthalten sein soll. Es gibt folgende Verbindungstypen:

- Anfang-Anfang (AA oder SS) (Anfangsfolge)
- Anfang-Ende (AE oder SF) (Sprungfolge)
- Ende-Anfang (EA oder FS) (Normalfolge)
- Ende-Ende (EE oder FF) (Endfolge).

In Klammern stehen jeweils die möglichen Feldinhalte, die den jeweiligen Typ repräsentieren.

Markierungstyp

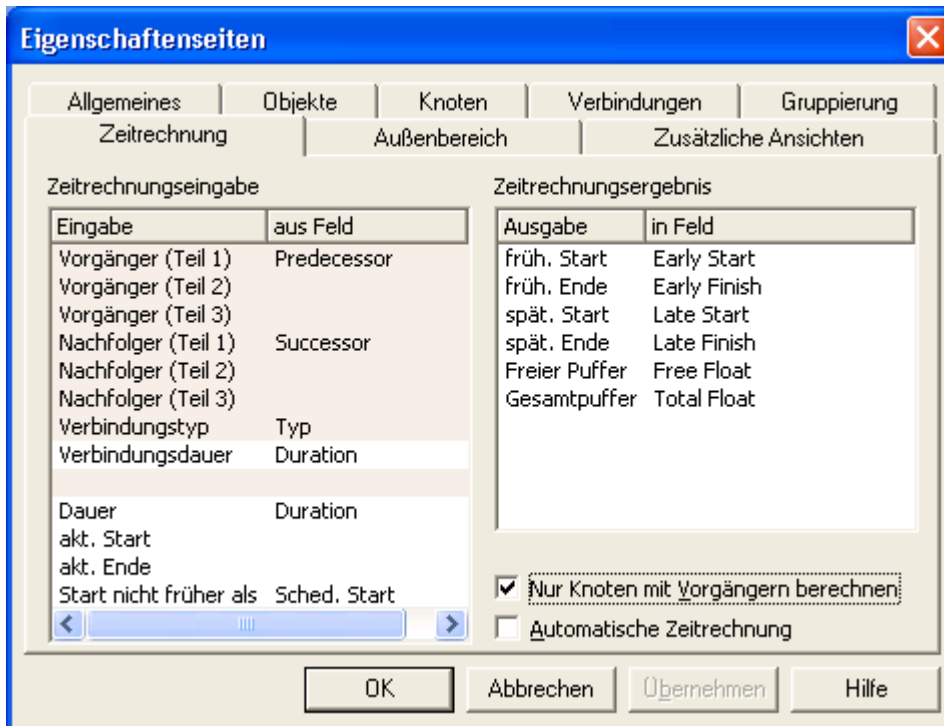
Wählen Sie hier aus, ob und welche Art der interaktiven Markierung von Verbindungen dem Anwender geboten werden soll. Folgende Alternativen stehen zur Verfügung:

- Einrahmen
- Invertieren
- ohne
- Pickmarks

Beschriftungspositionen zusätzlich zu Knotenpositionen mit Datenfeldern synchronisieren

Wählen Sie je ein Datenfeld aus, aus dem die X- bzw. Y-Koordinate der Position jeder Verbindungsbeschriftung gelesen und wohin sie zurückgeschrieben werden soll. Die Synchronisierung der Positionen der Verbindungsbeschriftungen mit den Datenfeldern ist erforderlich, damit die Positionen der Verbindungsbeschriftungen nach dem Schließen Ihres Projekts wiederhergestellt werden können.

4.9 Eigenschaftenseite "Zeitrechnung"



Mit Hilfe dieser Eigenschaftenseite können Sie die Zeitrechnung des VARCHART XNet an Ihre Schnittstelle anpassen, indem Sie festlegen, welche Datenfelder für die Eingabe (**Zeitrechnungseingabe**) und die Ausgabe (**Zeitrechnungsergebnis**) des Schedulers verwendet werden sollen.

Zeitrechnungseingabe

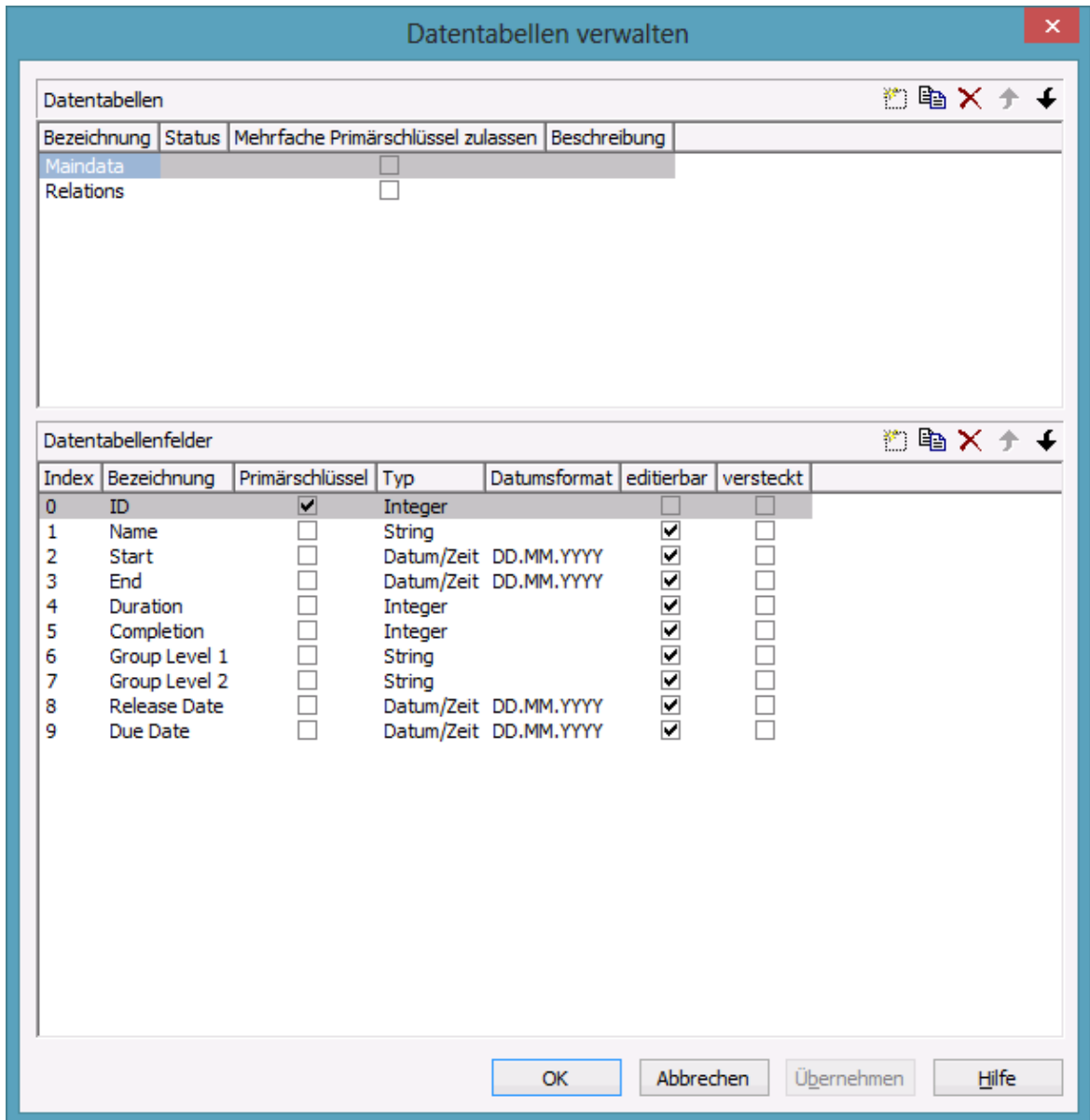
Wählen Sie hier für jede Eingabe aus, aus welchem Feld sie entnommen werden soll. Als Eingaben für die Zeitrechnung verwendet der Scheduler Datenfelder der jeweils eingestellten Knoten- und Verbindungstabellen.. Die Grundlage der Berechnung sind die Dauer der einzelnen Vorgänge, deren logische Abhängigkeiten 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 "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 Datentabelle 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.

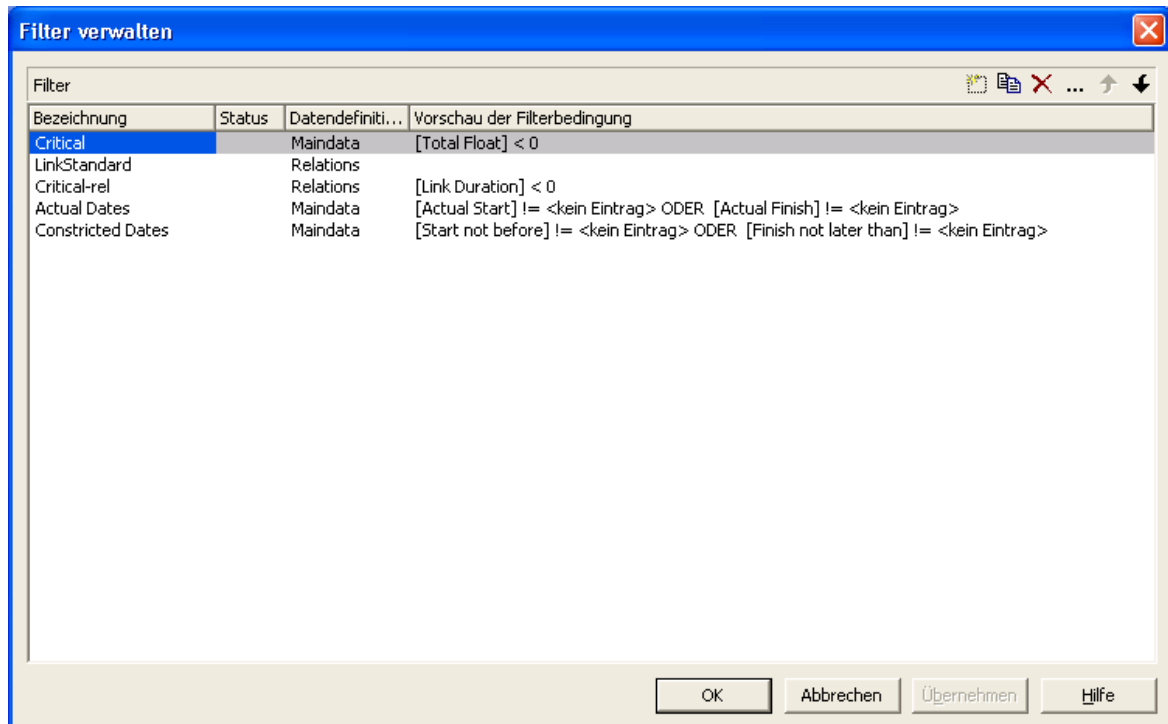
Dieses Datumsformat ist nur für die Dateneingabe, aber nicht für die Darstellung von Daten in Ihrer Grafik relevant.

- **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.

4.11 Dialogfeld "Filter verwalten"





Sie erreichen dieses Dialogfeld

- von der Eigenschaftenseite **Objekte**
- aus dem Dialog **Knotenaussehen verwalten**
- aus dem Dialog **Verbindungsaussehen verwalten**.

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.


Datendefinitionstabelle

Hier wird die Datendefinitionstabelle (**Maindata** bzw. **Relations**) angezeigt, die dem jeweiligen Filter zu Grundeliegt (vgl. Eigenschaftenseite **Datendefinition**).

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.

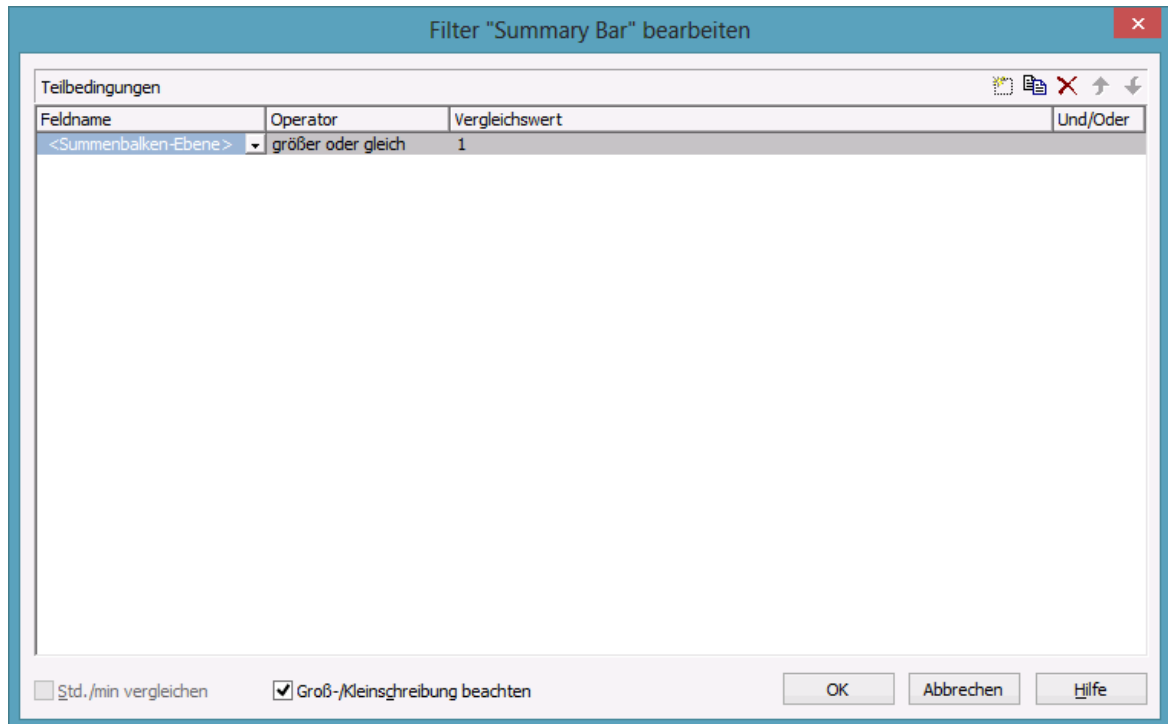
Filter bearbeiten

 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.

4.12 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.

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.)

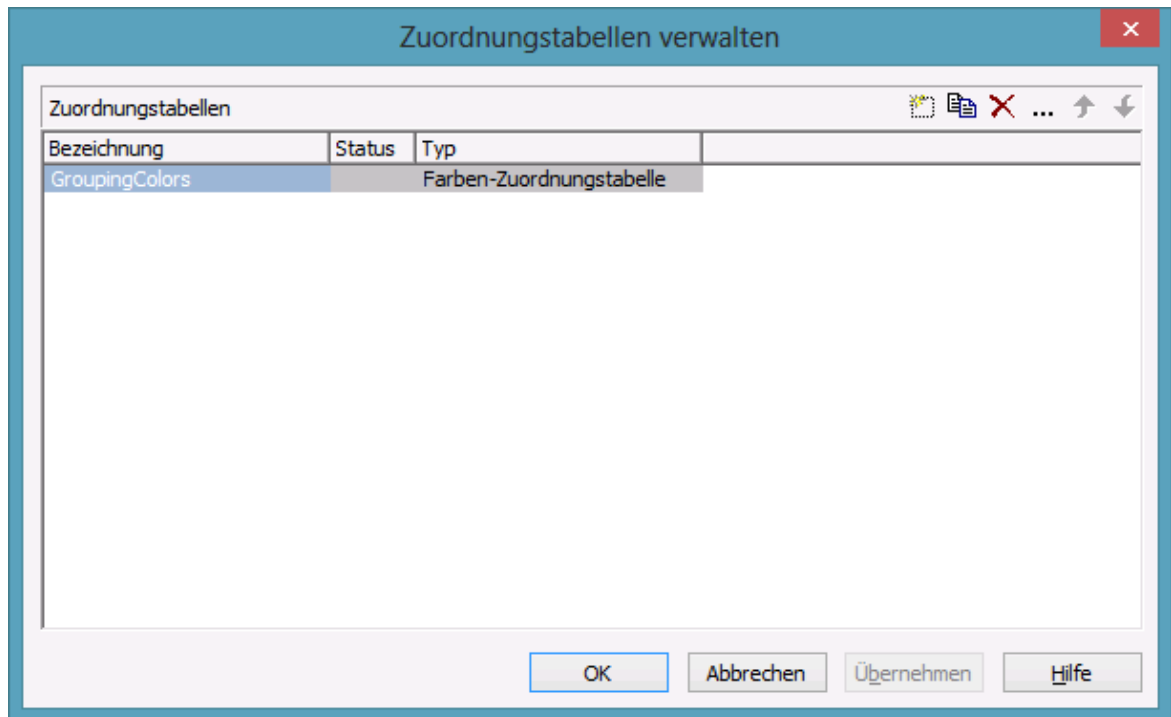
Std./min vergleichen

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.

4.13 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 (für spätere Anwendungen)

- Grafikdateien-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.

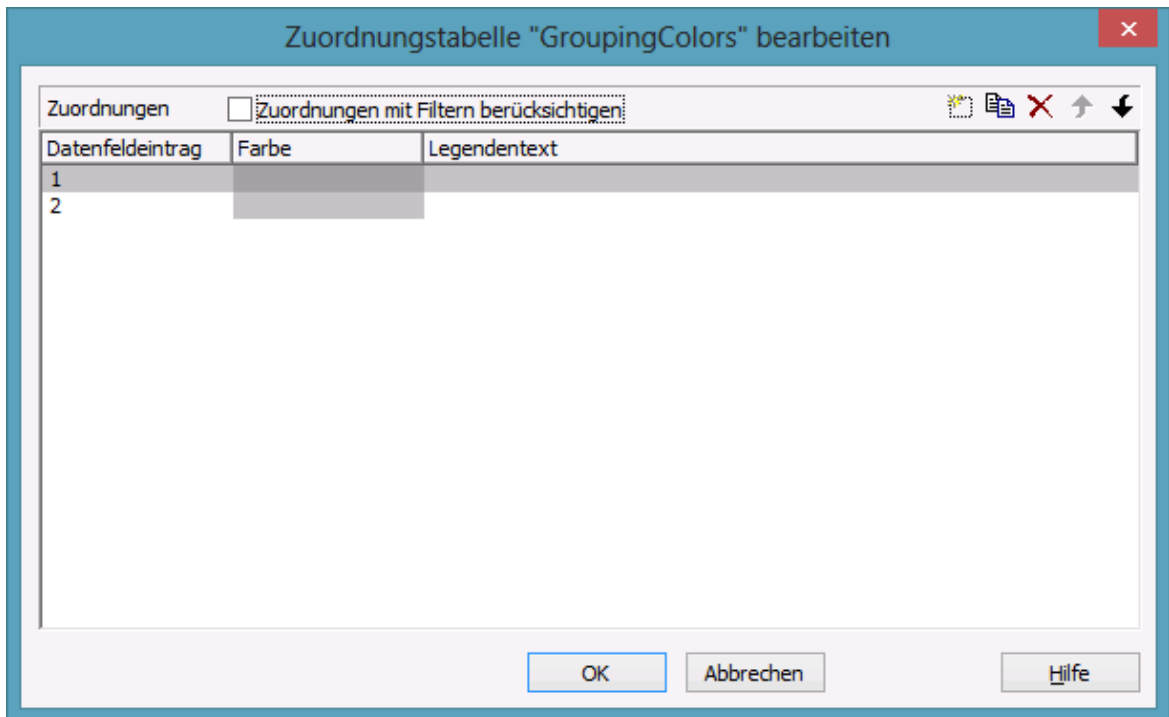
Zuordnungstabelle bearbeiten

 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.

4.14 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. eine Grafikdatei und den Legendentext vereinbaren können.

Grafikdateiname

Um die Farbe bzw. die Grafikdatei für einen Datenfeldeintrag festzulegen, klicken Sie auf das entsprechende Feld. Dann erscheint eine Schaltfläche, über die Sie in den Dialog zur Auswahl der Grafikdatei gelangen.

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-Steuerelements gesucht.

Farbe/Grafikdateiname


Um die Farbe bzw. die Grafikdatei für einen Datenfeldeintrag festzulegen, klicken Sie auf das entsprechende Feld. Dann erscheint eine Schaltfläche, über die Sie in den Dialog zur Auswahl der Farbe bzw. der Grafikdatei gelangen.

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-Steuerelements gesucht.

Legendentext

(nur für Farb-Zuordnungstabellen) 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.

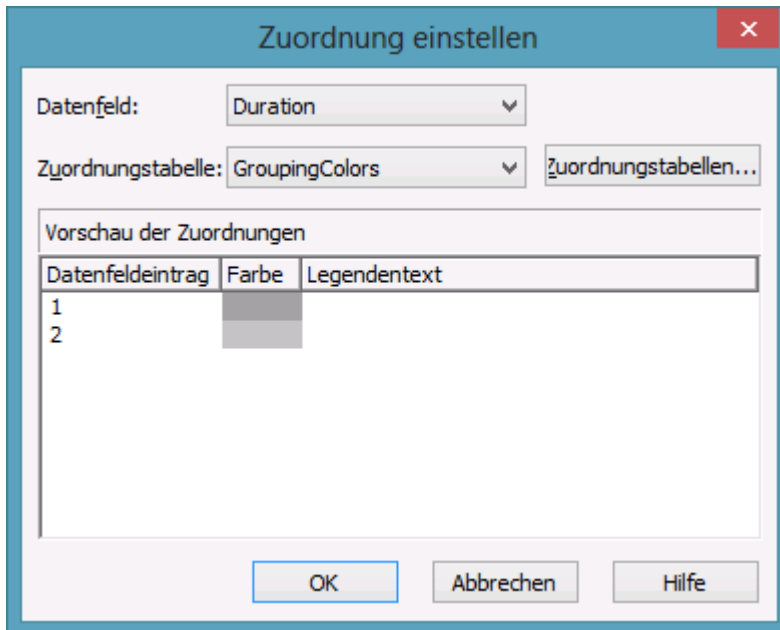
Zuordnung löschen


 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.

4.15 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.

Datenfeld

Wählen Sie hier das Datenfeld aus, von dessen Einträgen die Farbe oder das Muster 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.

Zuordnungstabelle

(nur aktiv, wenn ein Datenfeld gewählt wurde) Wählen Sie hier die Zuordnungstabelle aus, die den einzelnen Datenfeldeinträgen eine Farbe bzw. eine Grafikdatei und einen Legendentext zuordnet.

Zuordnungstabellen

Klicken Sie auf diese Schaltfläche, um das Dialogfeld **Zuordnungstabellen verwalten** aufzurufen. Hier können Sie Zuordnungstabellen erstellen, bearbeiten, kopieren oder löschen.

Zuordnungstabellen

Klicken Sie auf diese Schaltfläche, um das Dialogfeld **Zuordnungstabellen verwalten** aufzurufen. Hier können Sie Zuordnungstabellen erstellen, bearbeiten, kopieren oder löschen.

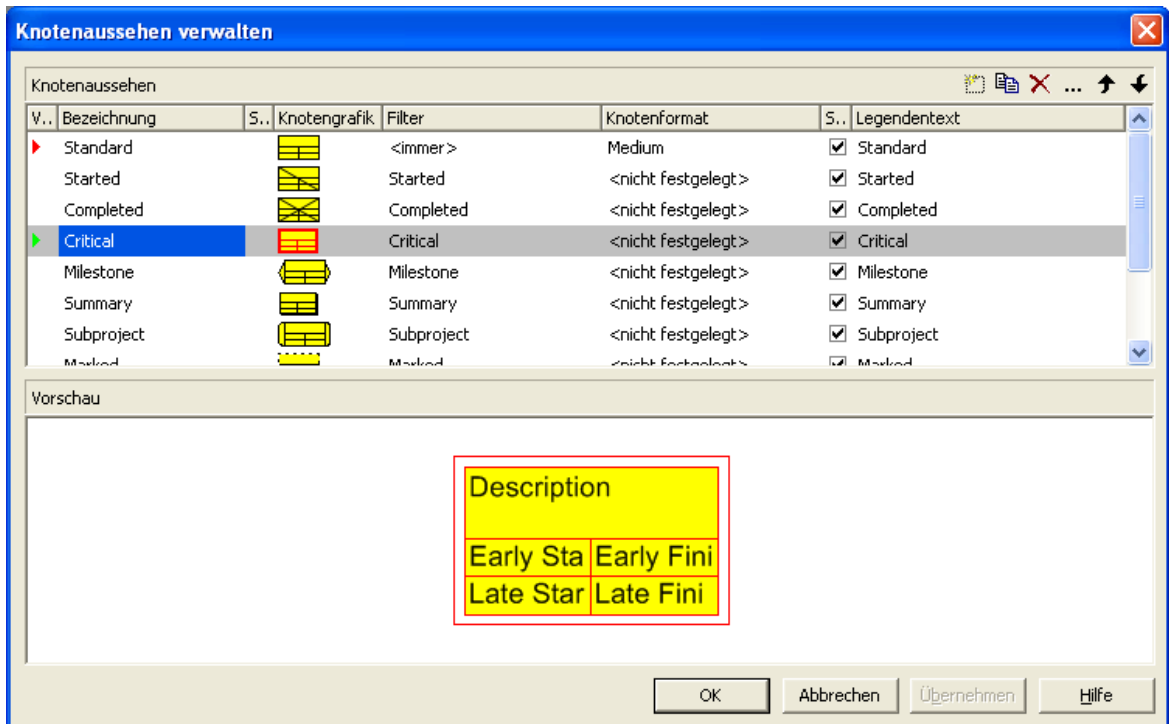
Vorschau der Zuordnungen

Hier wird die gewählte Zuordnungstabelle dargestellt: alle Datenfelder sowie die diesen zugeordneten Attribute.

Vorschau der Zuordnungen

Hier wird die gewählte Zuordnungstabelle dargestellt: alle Datenfelder, die diesen zugeordneten Farben und Legendentexte bzw. die Grafikdateinamen.

4.16 Dialogfeld "Knotenaussehen verwalten"



Sie gelangen über die Eigenschaftenseite **Objekte** in dieses Dialogfeld.

Die Darstellung von Knoten wird festgelegt, indem diesen ein oder mehrere Knotenaussehen dynamisch über Filter zugewiesen werden. Aus der Überlagerung der Attribute aller Knotenaussehen, die auf einen Knoten zutreffen, ergibt sich das gesamte Knotenaussehen.

Vorschau



Alle Knotenaussehen, die in der **Vorschau**-Spalte durch eine kleine rote Pfeilspitze markiert sind, werden im Vorschaufenster überlagert und mit den durch die Abarbeitungsreihenfolge bestimmten Überlagerungen angezeigt.

Das Knotenaussehen, auf dem der Cursor gerade steht, wird durch eine grüne Pfeilspitze gekennzeichnet.

Bezeichnung

Diese Spalte enthält die Namen aller vorhandenen Knotenaussehen. Die Namen sind editierbar.

Status

In dieser Spalte wird jedes Knotenaussehen, das seit dem Aufruf des Dialogs hinzugefügt () und/oder geändert () worden ist.

Knotengrafik

Hier wird jedes Knotenaussehen dargestellt. Um die Knotengrafik, d. h. die grafischen Attribute des Knotenaussehens, zu verändern, klicken Sie auf die **Knotenaussehen bearbeiten**-Schaltfläche oberhalb der Tabelle oder doppelklicken Sie auf den Eintrag **Knotengrafik**. Sie gelangen dann in den Dialog **Knotenaussehen bearbeiten**.

Filter

Der Filter, der mit einem Knotenaussehen verbunden ist, bestimmt, welche Knoten mit dem betreffenden Knotenaussehen versehen werden.

Sie können für die meisten Knotenaussehen den Filter auswählen. Nur für die Knotenaussehen "Standard" und "Anschlussknoten" ist der Filter festgelegt ("*<immer>*" bzw. "*<Anschlussknoten>*").

Um einem Knotenaussehen einen Filter zuzuweisen, markieren Sie das **Filter**-Feld. Dann erscheinen eine Schaltfläche für eine Kombobox mit allen verfügbaren Filtern sowie eine **Bearbeiten**-Schaltfläche (außer bei den Knotenaussehen mit fest zugewiesenem Filter). Wählen Sie aus der Kombobox einen Filter für das Knotenaussehen aus oder klicken Sie auf die **Bearbeiten**-Schaltfläche des **Filter**-Feldes, um den Dialog **Filter verwalten** aufzurufen. Dort können Sie Filter bearbeiten, kopieren, neu definieren oder löschen.

Knotenformat

Ein Knotenformat definiert die Anzahl, die Anordnung und das Aussehen der Felder, die zur Beschriftung eines Knotens verwendet werden. Wählen Sie hier das Knotenformat für das Knotenaussehen aus. Markieren Sie dazu das **Knotenformat**-Feld. Dann erscheinen eine Schaltfläche für eine Kombobox mit allen verfügbaren Knotenformaten sowie eine **Bearbeiten**-Schaltfläche. Wählen Sie aus der Kombobox ein Knotenformat für die Knotenaussehen aus. Oder klicken Sie auf die **Bearbeiten**-Schaltfläche des **Knotenformat**-Feldes, um den Dialog **Knotenformate verwalten** aufzurufen. Dort können Sie Knotenformate bearbeiten, kopieren, hinzufügen oder löschen.

Sichtbar in Legende

Aktivieren Sie dieses Kontrollkästchen für alle Knotenaussehen, die in der Legende dargestellt werden sollen.

Legendentext

In dieser Spalte können Sie für jedes Knotenaussehen einen Legendentext eintragen.


Knotenaussehen hinzufügen

 Ein neues Knotenaussehen wird am Ende der Liste hinzugefügt.


Knotenaussehen kopieren

 Das markierte Knotenaussehen wird kopiert.

Knotenaussehen löschen

 Knotenaussehen, die Sie nicht mehr verwenden, können Sie über diese Schaltfläche löschen. Es erfolgt noch eine Abfrage, ob Sie das markierte Knotenaussehen wirklich löschen wollen. Das Standard-Knotenaussehen kann nicht gelöscht werden.

Knotenaussehen bearbeiten


 Über diese Schaltfläche gelangen Sie in das Dialogfeld **Knotenaussehen bearbeiten**.


Knotenaussehen früher/später abarbeiten

Wenn einem Knoten mehrere Knotenaussehen zugewiesen sind, werden die einzelnen Knotenaussehen nacheinander abgearbeitet. In der Tabelle sind die Knotenaussehen nach ihrer Abarbeitungsreihenfolge sortiert. Das Standard-Knotenaussehen steht immer ganz oben in der Tabelle, da es immer zugewiesen ist und immer zuerst abgearbeitet wird. Das Knotenaussehen, das zuletzt abgearbeitet wird, steht ganz unten in der Tabelle.

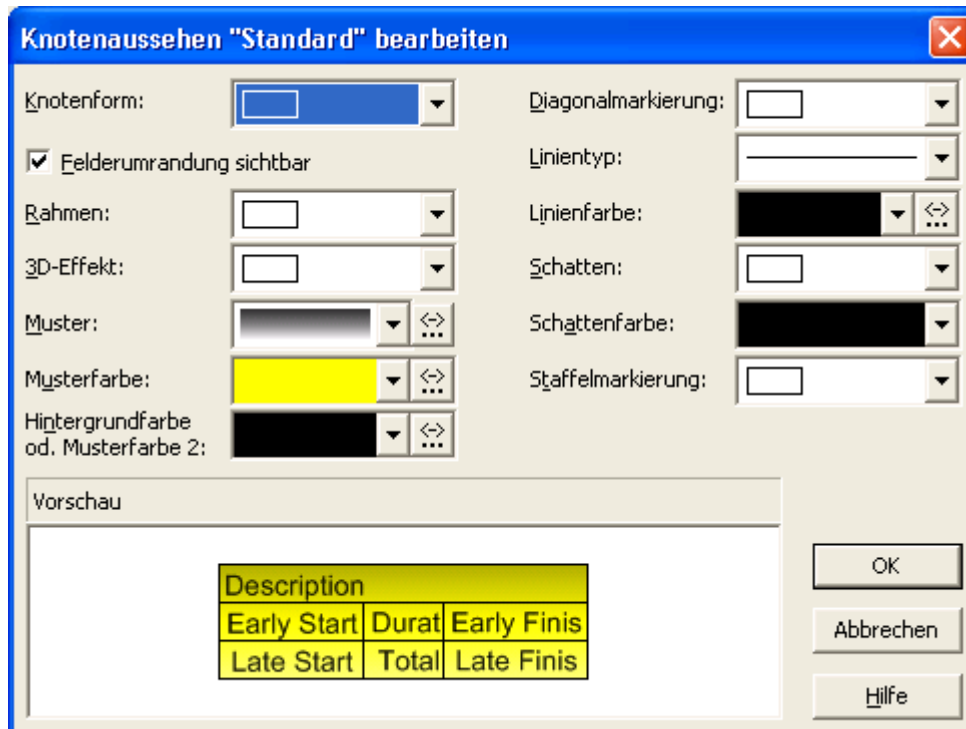
Wenn mehrere Knotenaussehen auf einen Knoten zutreffen, werden die Attribute jedes Knotenaussehens durch die Attribute aller Knotenaussehen, die später abgearbeitet werden, ersetzt. Nur die Attribute, deren Wert "nicht festgelegt" ist, ersetzen nicht die Attribute ihrer Vorgänger.

Mit Hilfe dieser Schaltflächen können Sie die Position eines Knotenaussehens in der Abarbeitungsreihenfolge verändern:

 Wenn Sie die Schaltfläche **Knotenaussehen früher abarbeiten** anklicken, steigt das markierte Knotenaussehen um eine Position in der Tabelle und wird entsprechend früher abgearbeitet.

 Wenn Sie die Schaltfläche **Knotenaussehen später abarbeiten** anklicken, fällt das markierte Knotenaussehen um eine Position in der Tabelle und wird entsprechend später abgearbeitet.

4.17 Dialogfeld "Knotenaussehen bearbeiten"



In der Titelzeile wird der Name des Knotenaussehens angezeigt, das Sie in diesem Dialogfeld bearbeiten können.

Wenn einem Knoten mehrere Knotenaussehen zugewiesen sind, werden die Attribute jedes Knotenaussehens durch die Attribute aller Knotenaussehen, die eine höhere Priorität haben, ersetzt. Nur die Attribute, deren Wert <nicht festgelegt> ist, ersetzen nicht die Attribute ihrer Vorgänger.

Knotenform

Wählen Sie hier die Knotenform für das aktuelle Knotenaussehen aus. Zur Auswahl stehen verschiedene Knotenformen sowie die Einträge <nicht festgelegt> und <ohne Rahmen>.

Felderumrandung sichtbar

Mit dieser Eigenschaft kann festgelegt werden, ob der Rahmen um die innenliegenden Felder sichtbar ist oder nicht. Die Außenrandlinie der Form ist davon nicht betroffen, daher wirkt sich diese Eigenschaft bei den möglichen Rahmenformen unterschiedlich aus und hat z.B. beim Typ **vcRectangle** keine Auswirkung.

Diese Option kann auch über die Eigenschaft **VcNodeAppearance.FrameAroundFieldsVisible** gesetzt werden.

Rahmen


Über dieses Feld legen Sie fest, ob die Knoten einen einfachen Rahmen oder einen Doppelrahmen erhalten.


3D-Effekt

Über dieses Feld bestimmen Sie, ob die Knoten einen 3D-Effekt erhalten oder nicht.

Muster

Hier wird das standardmäßige Muster des Knotenaussehens angezeigt.

 Sie können über die Pfeil-Schaltfläche die Muster-Liste öffnen, um daraus ein Muster auszuwählen.


 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung des Musters vereinbaren.


 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche gefüllt dargestellt.

Achtung: Wenn im dem Knotenaussehen zugewiesenen Knotenformat die Hintergrundfarbe für ein Feld nicht auf transparent gesetzt ist, dann scheint das hier definierte Muster mit seinen Farben auch nicht durch!

Musterfarbe

Hier können Sie die standardmäßige Farbe des Musters festlegen.

 Sie können über die Pfeil-Schaltfläche die Farbzusammenstellungstabelle öffnen, um daraus eine Farbe für das Muster auszuwählen. Es stehen auch transparente Farben zur Verfügung.


 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Farbe für das Muster vereinbaren.


 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche gefüllt dargestellt.

Achtung: Wenn im dem Knotenaussehen zugewiesenen Knotenformat die Hintergrundfarbe für ein Feld nicht auf transparent gesetzt ist, dann scheint das hier definierte Muster mit seinen Farben auch nicht durch!

Hintergrundfarbe oder Musterfarbe 2

Wählen Sie hier die Hintergrundfarbe für das aktuell ausgewählte Knotenaussehen aus.

 Sie können über die Pfeil-Schaltfläche die Farbzordnungstabelle öffnen, um daraus eine Farbe auszuwählen. Auch transparente Farben stehen zur Verfügung.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Hintergrundfarbe vereinbaren.

 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche gefüllt dargestellt.

Achtung: Wenn im dem Knotenaussehen zugewiesenen Knotenformat die Hintergrundfarbe für ein Feld nicht auf transparent gesetzt ist, dann scheint das hier definierte Muster mit seinen Farben auch nicht durch!

Diagonalmarkierung


Über dieses Feld bestimmen Sie, ob die Knoten eine Diagonalmarkierung erhalten und ggf. welche.


Linientyp

Wählen Sie hier die Linienart für die Umrandung der Knoten.

Linienfarbe

Wählen Sie hier die Linienfarbe für die Umrandung der Knoten aus.

 Sie können über die Pfeil-Schaltfläche die Farbzordnungstabelle öffnen, um daraus eine Farbe auszuwählen.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Linienfarbe vereinbaren.

 Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche gefüllt dargestellt.

Schatten

Über dieses Feld steuern Sie, ob die Knoten einen Schatten erhalten oder nicht.

Schattenfarbe

Legen Sie hier die Farbe fest, die ggf. der Schatten bzw. die Staffelmarkierung erhalten soll.

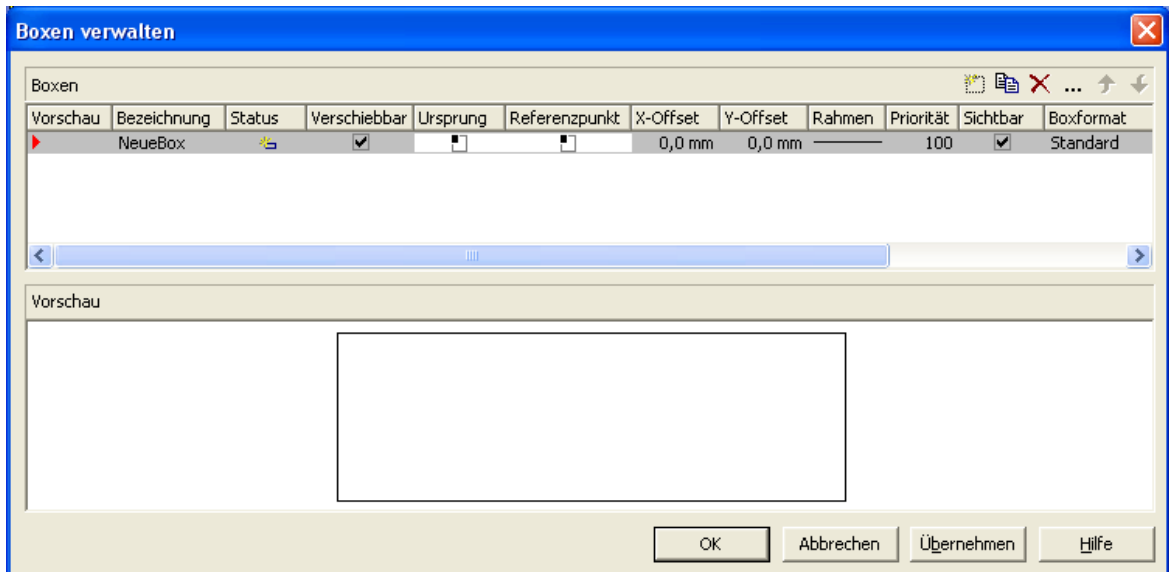
Staffelmarkierung

Über dieses Feld bestimmen Sie, ob die Knoten gestaffelt dargestellt werden sollen (max. 8fach) oder nicht.

Vorschau

In dem Fenster wird das Erscheinungsbild des aktuell festgelegten Knotenaussehens dargestellt.

4.18 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.

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



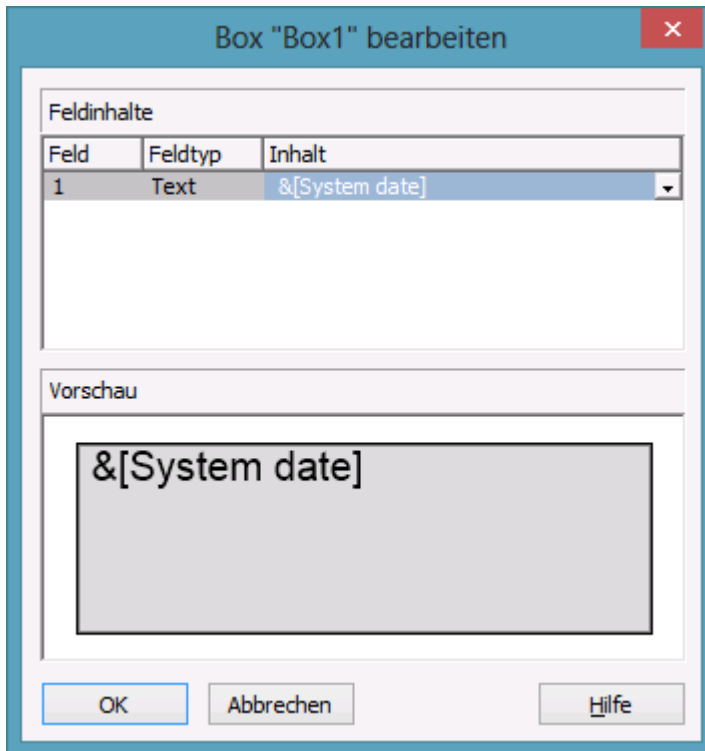
Sie gelangen in das Dialogfeld **Box bearbeiten**.

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.

4.19 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

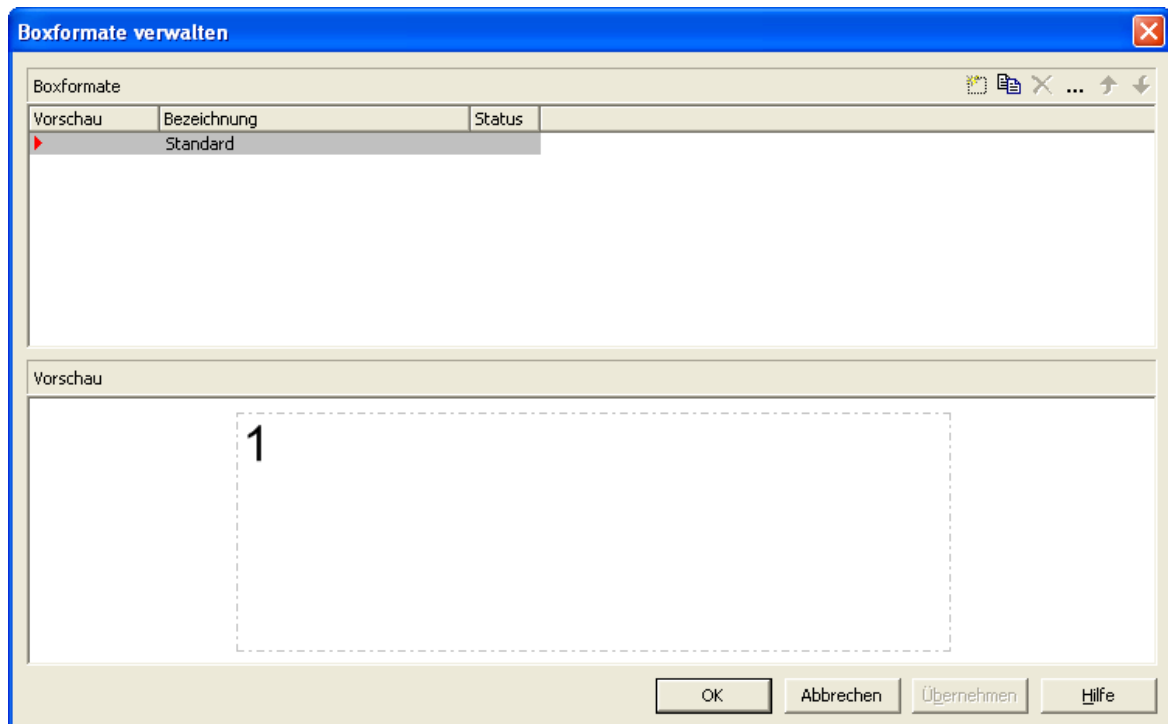
Hier können Sie den Inhalt des Feldes bzw. den Namen einer Grafikdatei eingeben.

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.

208 Dialogfeld "Box bearbeiten"

Mögliche Grafikformate: WMF, JPG, BMP, GIF, PCX, PNG, TIF.

4.20 Dialogfeld "Boxformate/Knotenformate verwalten"



Sie gelangen in dieses Dialogfeld über die Eigenschaftenseite **Objekte**.



Vorschau

Das in der Vorschau­spalte markierte Format wird im Vorschauen­fenster 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.


Box/Knotenformat hinzufügen

 Ein neues Format mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Box/Knotenformat kopieren

 Das markierte Format wird kopiert.



Box/Knotenformat 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.

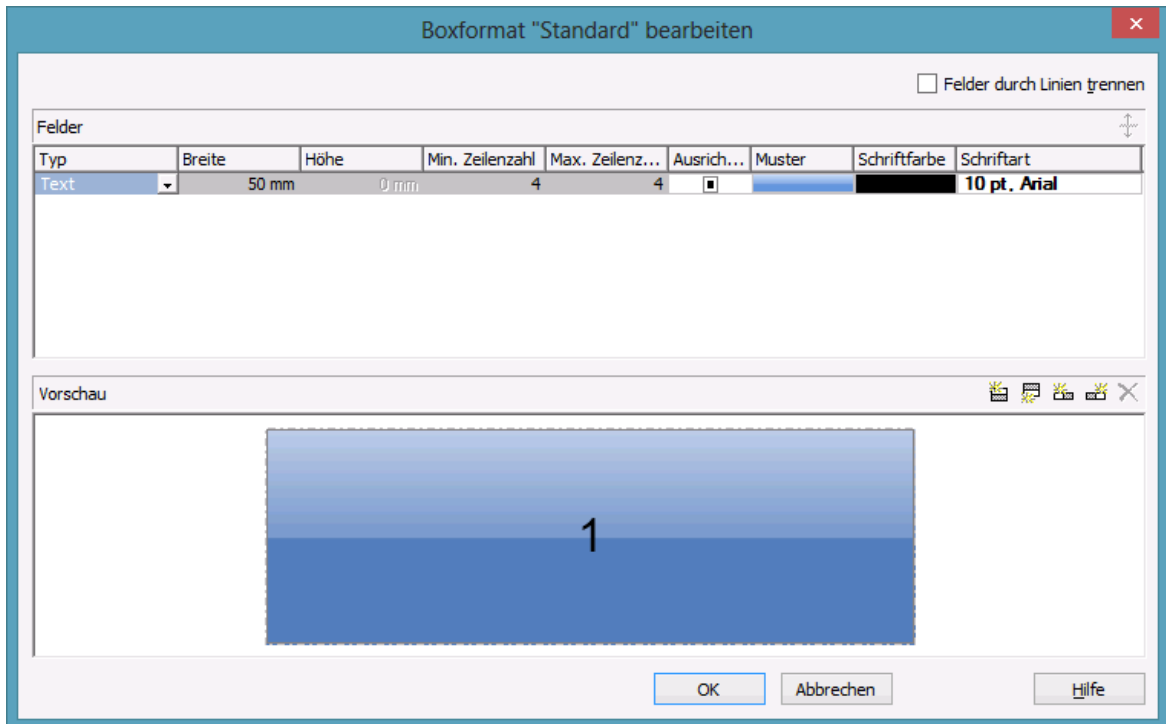
Box/Knotenformat bearbeiten

 Sie gelangen in das Dialogfeld **Boxformat bearbeiten** bzw. **Knotenformat bearbeiten**.

Box/Knotenformat eine Zeile nach oben/ unten

  Mit Hilfe dieser Schaltflächen können Sie das markierte Format in der Tabelle eine Zeile nach oben bzw. unten schieben.

4.21 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

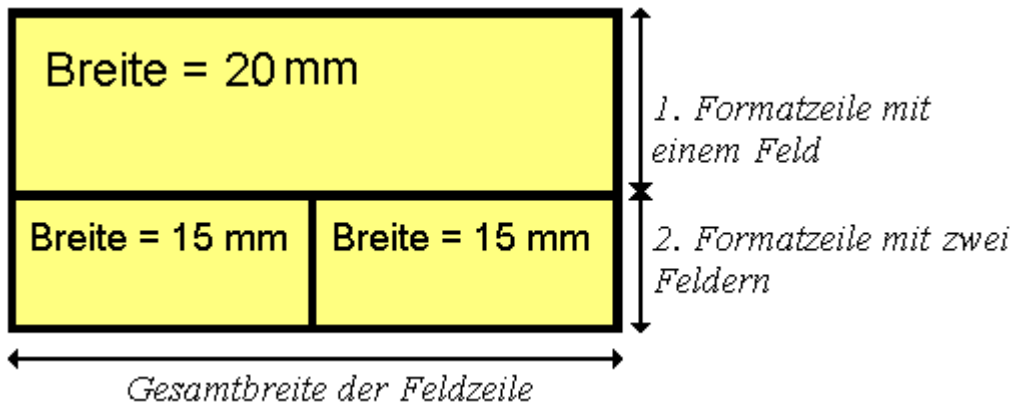
Aktivieren Sie dieses Kontrollkästchen, wenn die Felder der Box durch Linien getrennt werden sollen.

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

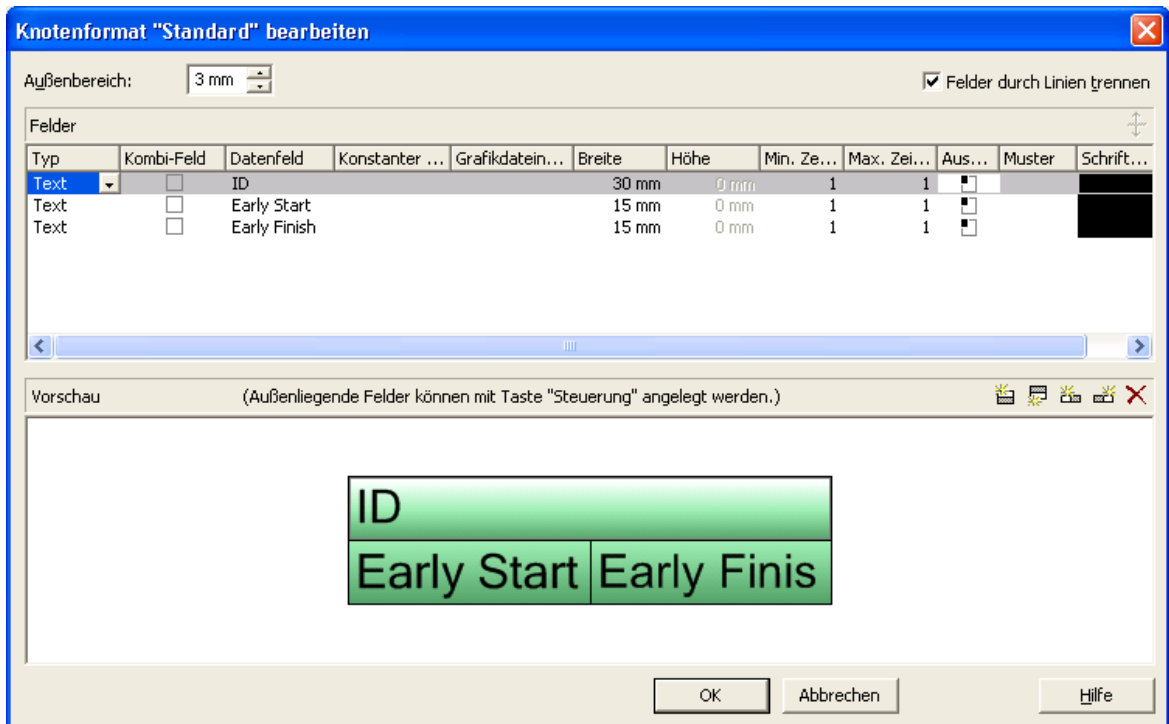
 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

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.

4.22 Dialogfeld "Knotenformat bearbeiten"



Sie gelangen in dieses Dialogfeld, indem Sie im Dialogfeld **Knotenformate verwalten** auf die **Knotenformat bearbeiten**-Schaltfläche klicken.

Außenbereich

Legen Sie hier den Abstand in Millimetern fest, den Knoten mit diesem Knotenformat zu benachbarten Knoten und zum Rand der Darstellung halten sollen. Standardmäßig beträgt der Außenbereich 3 mm. Bei kleineren Werten kann es gelegentlich zu Überlagerungen von grafischen Elementen kommen. Daher sollten Sie den Standardwert nur in begründeten Fällen unterschreiten.

Felder durch Linien trennen

Aktivieren Sie dieses Kontrollkästchen, wenn die einzelnen Felder durch Linien getrennt werden sollen.

Typ

Wählen Sie hier den Feldtyp (Text oder Grafik).

Kombi-Feld

Wenn dieses Kontrollkästchen aktiviert ist, können in dem Knotenfeld 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. 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 Knotenfeld ausgegeben werden soll.


Grafikdateiname

Hier werden Name und Pfad der Grafikdatei angezeigt, die in dem gewählten Knotenfeld 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 Knotenfeld 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.

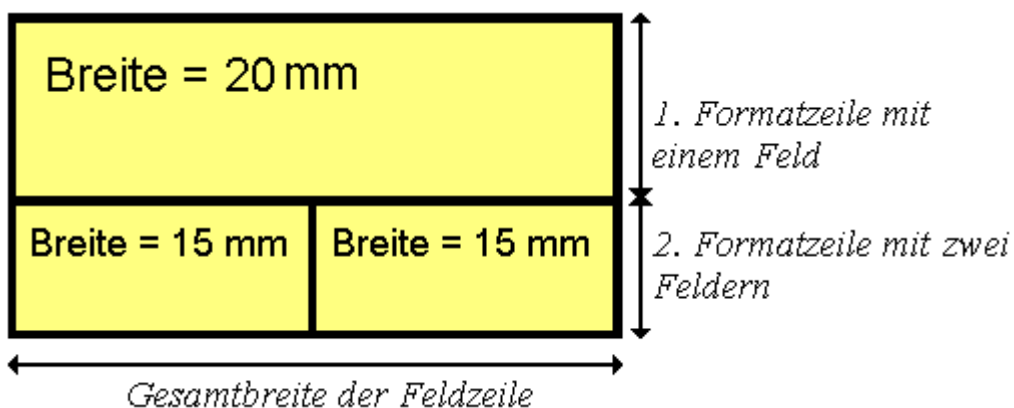
Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der 2. Schaltfläche fett dargestellt ().

 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.

Breite

Legen Sie hier die Breite in Millimetern für das markierte Feld fest. Die maximale Breite eines Feldes beträgt 99 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 beträgt 99 mm.


Min./Max. Zeilenzahl



(*nur für den Typ Text*) Bestimmen Sie die minimale und die maximale Anzahl der Textzeilen des aktuellen Feldes. Die maximale Zeilenzahl eines Feldes beträgt neun.

Ausrichtung

Hier bestimmen Sie die Ausrichtung des Textes bzw. der Grafik des markierten Feldes.

Muster


Legen Sie hier 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.



: Über diese Schaltfläche im Dialog **Musterattribute bearbeiten** gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie für das jeweilige Attribut eine datenabhängige Zuordnung vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt (.

Wird hier nichts festgelegt, wird automatisch das Attribut benutzt, das für das Knotenaussehen vereinbart wurde.

Schriftfarbe

(*nur für den Typ Text*) Die Schriftfarbe des Feldes wird hier angezeigt, und Sie können sie hier bearbeiten. 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, und Sie können sie hier bearbeiten. Wenn Sie auf dieses Feld klicken, erscheint eine Schaltfläche (...), über die Sie den Windows-Dialog **Schriftart** öffnen können.

selektierte Eigenschaft in alle Felder übernehmen

 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

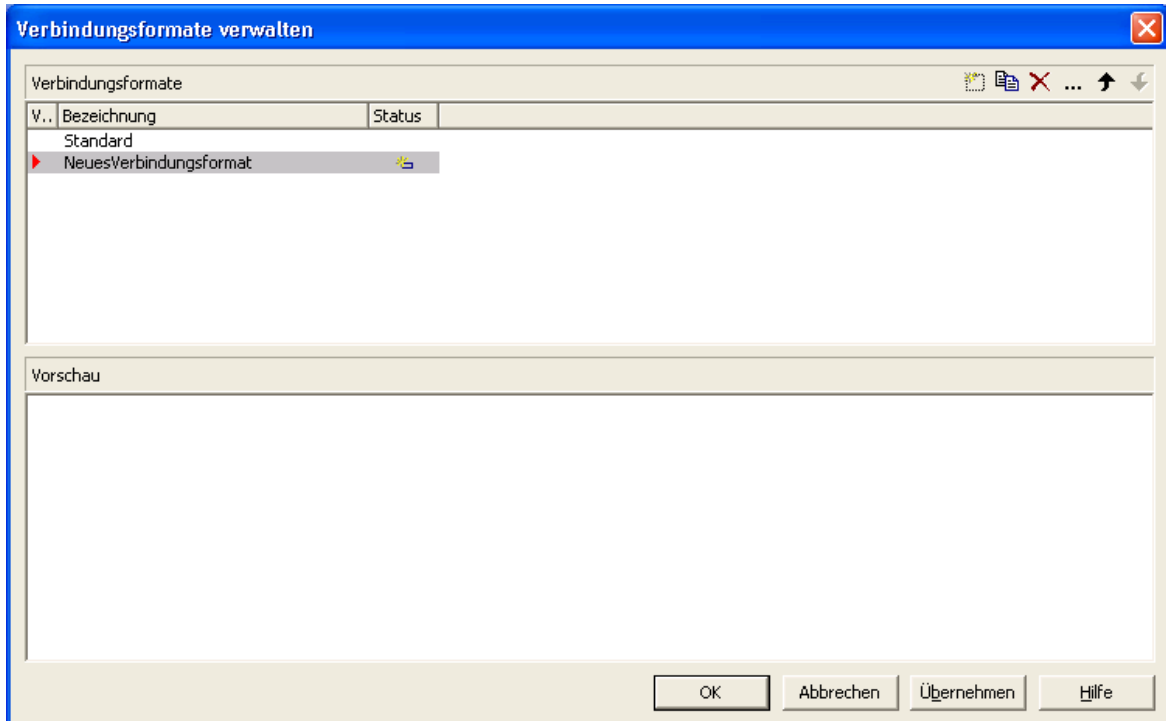
Vorschau

Hier wird das aktuelle Knotenformat dargestellt. Wenn Sie im Vorschau-fenster 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. Sie können auch die Entf-Tast benutzen, um Felder zu löschen.

Wenn Sie Felder außerhalb des Knotens anlegen möchten, so ist zusätzlich die Strg-Taste zu drücken.

4.23 Dialogfeld "Verbindungsformate verwalten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte** oder aus dem Dialog **Verbindungsausssehen verwalten** durch Klick auf **...** im Feld **Verbindungsformate**.

Vorschau

In dieser Spalte wird das jeweils im Vorschaufenster angezeigte Verbindungsformat gekennzeichnet.


Bezeichnung

In dieser Spalte werden die Bezeichnungen aller Verbindungsformate aufgeführt. Die Bezeichnungen sind editierbar.

Status

In dieser Spalte wird jedes Verbindungsformat gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt (✚) und/oder geändert (⚠) wurde.


Verbindungsformat hinzufügen

 Ein neues Linienformat mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.



Verbindungsformat kopieren

 Das markierte Linienformat wird kopiert.

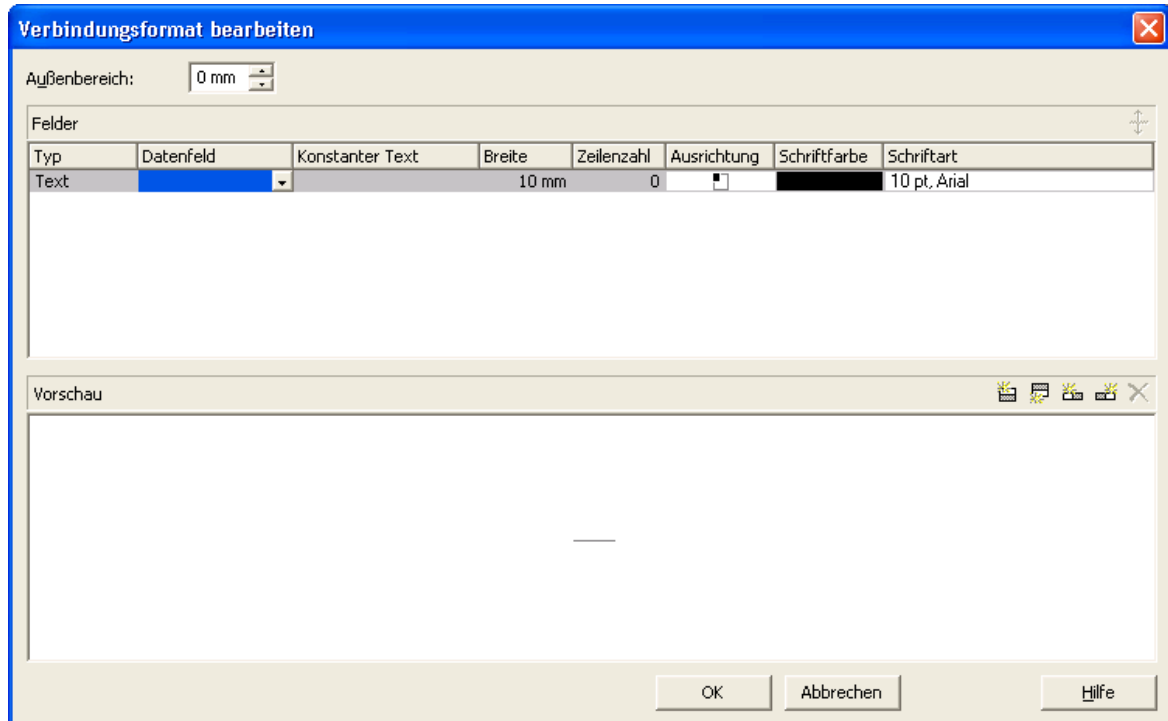
Verbindungsformat löschen

 Der Filter, den Sie in der Liste markiert haben, wird gelöscht. Es können nur Filter gelöscht werden, die zur Zeit nicht benutzt werden.

Verbindungsformat eine Zeile nach oben/unten

  Mit Hilfe dieser Schaltflächen können Sie das markierte Linienformat eine Zeile nach oben/unten schieben.

4.24 Dialogfeld "Verbindungsformat bearbeiten"



Sie gelangen in dieses Dialogfeld, wenn Sie im Dialog **Verbindungsformate verwalten** auf die Schaltfläche  klicken.

Außenbereich

Legen Sie hier den Abstand in Millimetern fest, den Verbindungen mit diesem Verbindungsformat zu benachbarten Knoten halten sollen. Standardmäßig beträgt der Außenbereich 3 mm. Bei kleineren Werten kann es gelegentlich zu Überlagerungen von grafischen Elementen kommen. Daher sollten Sie den Standardwert nur in begründeten Fällen unterschreiten.

Typ

Der Feldtyp ist Text.

Datenfeld

Wählen Sie hier das Datenfeld, dessen Inhalt in dem aktuellen Feld ausgegeben werden soll. 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 99 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.


Zeilenzahl

Bestimmen Sie die Anzahl der Textzeilen des aktuellen Feldes. Die maximale Zeilenzahl eines Feldes beträgt neun.


Ausrichtung

Hier bestimmen Sie die Ausrichtung des Textes des markierten Feldes.


Schriftfarbe

Die Schriftfarbe des Feldes wird hier angezeigt, und Sie können sie hier bearbeiten. Wenn Sie auf dieses Feld klicken, erscheint eine Schaltfläche () , über die Sie die Farbauswahltabelle öffnen können.

Schriftart

Die Schriftart des Feldes wird hier angezeigt, und Sie können sie hier bearbeiten. Wenn Sie auf dieses Feld klicken, erscheint eine Schaltfläche () , über die Sie den Windows-Dialog **Schriftart** öffnen können.

selektierte Eigenschaft in alle Felder übernehmen

 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

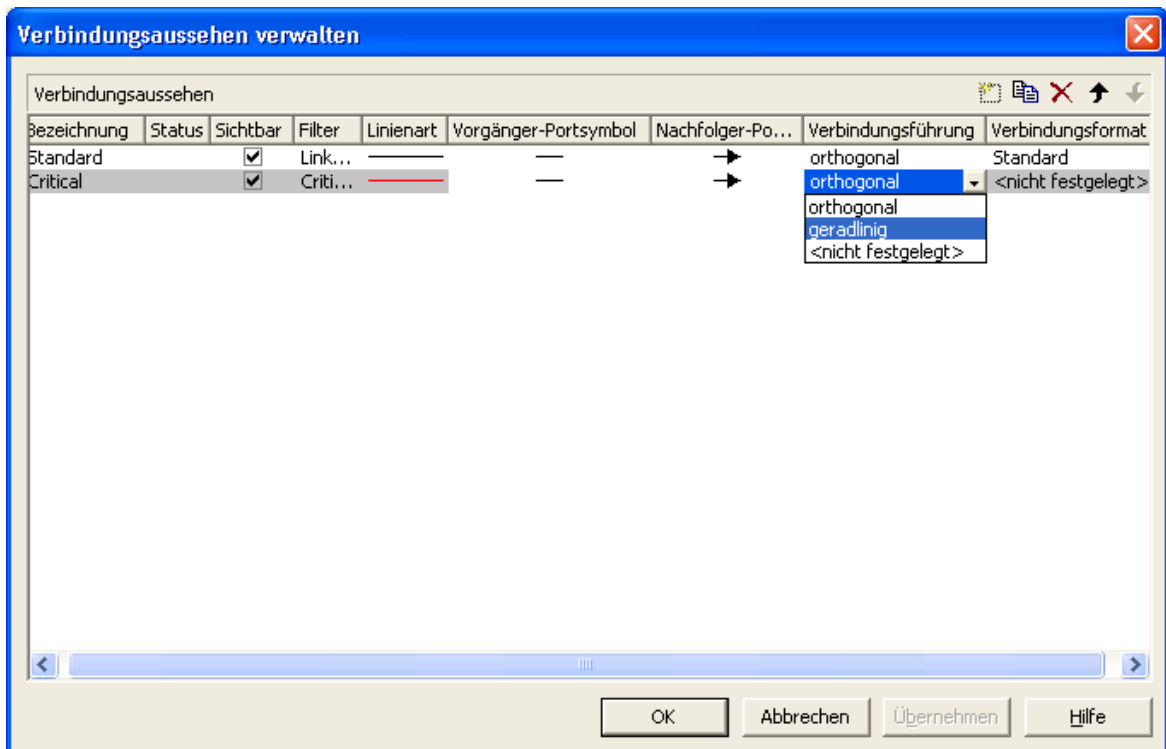
Vorschau

Hier wird das aktuelle Verbindungsformat dargestellt. Wenn Sie im Vorschauenfenster 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. Sie können auch die Entf-Tast benutzen, um Felder zu löschen.

4.25 Dialogfeld "Verbindungsausssehen verwalten"





Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**.

Bezeichnung

In dieser Spalte werden die Namen der verfügbaren Verbindungsausssehen angezeigt. Die Namen sind editierbar.

Diese Option kann auch über die Eigenschaft **VcLinkAppearance.Name** festgelegt werden.

Status

In der Spalte **Status** wird jedes Verbindungsausssehen 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.

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

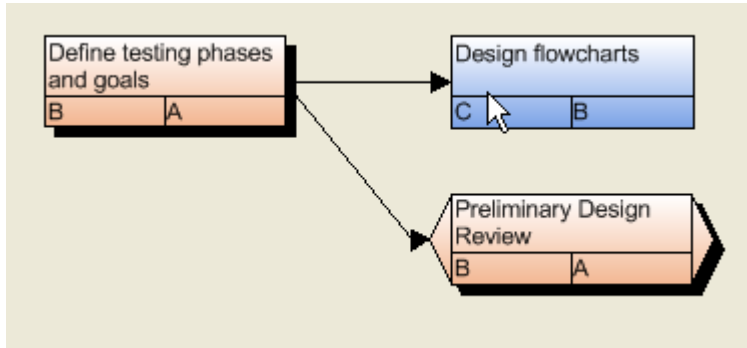
Wählen Sie hier für jedes Verbindungsaussehen ein Portsymbol, das die Einmündung zum Nachfolgerknoten kennzeichnet.

Diese Option kann auch über die Eigenschaft **VcLink Appearance.SuccessorPortSymbol** festgelegt werden.

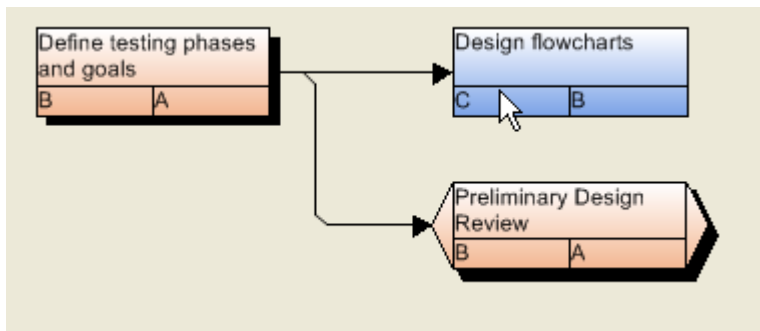
Verbindungsführung

Wählen Sie hier die Art der Verbindungsführung. Der Eintrag <nicht festgelegt> ist erst ab der zweiten Zeile der Tabelle mit den Verbindungsaussehen wählbar, da die erste Zeile immer das Standard-Verbindungsaussehen 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 **VcLink-Appearance-Eigenschaft RoutingType** setzen.




Gerader Verbindungstyp




Orthogonaler Verbindungstyp

Verbindungsformat


Wählen Sie hier durch Klick auf  ein vorhandenes Linienformat aus oder klicken auf **sch-bearbeiten.gif** um den Dialog **Verbindungsformate verwalten** zu öffnen, in dem Sie Verbindungsformate erstellen bzw. bearbeiten können.

Diese Option kann auch über die Eigenschaft **VcLinkAppearance.FormatName** festgelegt werden.


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.

4.26 Dialogfeld "In-Flow-Gruppierung bearbeiten"

Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Knoten**. Sie können hier die Kriterien und das Layout der In-Flow-Gruppierung bearbeiten. Bei einer Orientierung des Diagramms von links nach rechts können Sie oberhalb und/oder unterhalb des Diagrammbereichs eine Titelleiste ausgeben. Bei einer Orientierung von oben nach unten können Sie links und/oder rechts des Diagrammbereichs eine Titelleiste ausgeben.

Code aus Feld

Wählen Sie das Datenfeld aus, das die In-Flow-Gruppierung bestimmen soll.

Zeitintervall

*(Nur verfügbar, falls unter **Code aus Feld** ein Termindatenfeld gewählt ist)*
Legen Sie hier das Zeitintervall fest, in das die Titelleisten unterteilt werden sollen (z. B. 1 Sekunde, 1 Minute, 1 Stunde, 1 Tag, 2 Monate, 1 Jahr).

Trennlinien

Wenn Sie dieses Kontrollkästchen aktivieren, werden vertikale Trennlinien (bei Anordnung von links nach rechts) bzw. horizontale Trennlinien (bei Anordnung von oben nach unten) im Diagramm ausgegeben. Falls Sie unter **Code aus Feld** ein Termindatenfeld gewählt haben, wird der Abstand dieser Trennlinien durch den unter **Zeitintervall** festgelegten Abstand bestimmt. Andernfalls wird nach jedem Wert des Datenfeldes eine Trennlinie gezogen.

Das Aussehen der Trennlinien können Sie festlegen, indem Sie über die **Bearbeiten**-Schaltfläche den Dialog **Linienattribute öffnen** aufrufen. Hier können Sie Typ, Dicke und Farbe der Trennlinien festlegen.


Titelleisten oben/unten bzw. links/rechts

Hier können Sie festlegen, ob beschriftete Titelleisten ausgegeben werden sollen:

- Anordnung von links nach rechts: Titelleisten oberhalb und/oder unterhalb des Diagramms
- Anordnung von oben nach unten: Titelleisten links und/oder rechts vom Diagramm.

Schrift

Die aktuelle Schriftart und Schriftfarbe der Titelleisten werden hier angezeigt.

 Über diese Schaltfläche öffnen Sie die Farbauswahltabelle, in der Sie die Schriftfarbe auswählen können.

 Über die Schaltfläche gelangen Sie in den Windows-Dialog **Schriftart**.

Hintergrundfarbe

Wählen Sie hier die Hintergrundfarbe der Titelleisten aus.

Breite

(Nur für Orientierung von oben nach unten) Legen Sie hier die Breite der vertikalen Titelleisten in mm fest.

Datumsformat

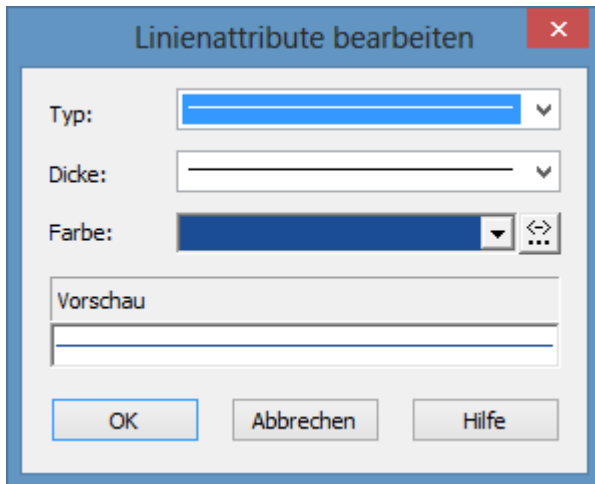
Wählen Sie diese Option, wenn Sie für **Code aus Feld** ein Datumsfeld gewählt haben, und legen Sie dann das Terminformat für die Beschriftung der Titelleisten fest.


Texte

Legen Sie hier die Beschriftung der Titelleisten fest:

- **aus Feld:** Wählen Sie diese Option, wenn die Beschriftung der Titelleisten durch ein Datenfeld bestimmt werden soll, und wählen Sie dann das entsprechende Datenfeld aus.
- **aus Datei:** Wählen Sie diese Option, wenn die Beschriftung der Titelleisten durch eine Datei bestimmt werden soll, und geben Sie ggf. den Namen dieser Datei an.

4.27 Dialogfeld "Linienattribute bearbeiten"



Der Dialog zum Bearbeiten der Linienattribute, der jeweils über die Schaltfläche  aufgerufen werden kann, steht zur Verfügung für das Verbindungsausssehen 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.

4.28 Dialogfeld "Musterattribute bearbeiten"



Der Muster-Dialog, der jeweils über die Schaltfläche  aufgerufen werden kann, steht zur Verfügung für Box- und Knotenformate.

Muster

Hier können Sie ein Hintergrundmuster auswählen.

Musterfarbe

Wählen Sie hier die Vordergrund-Farbe des Musters aus.

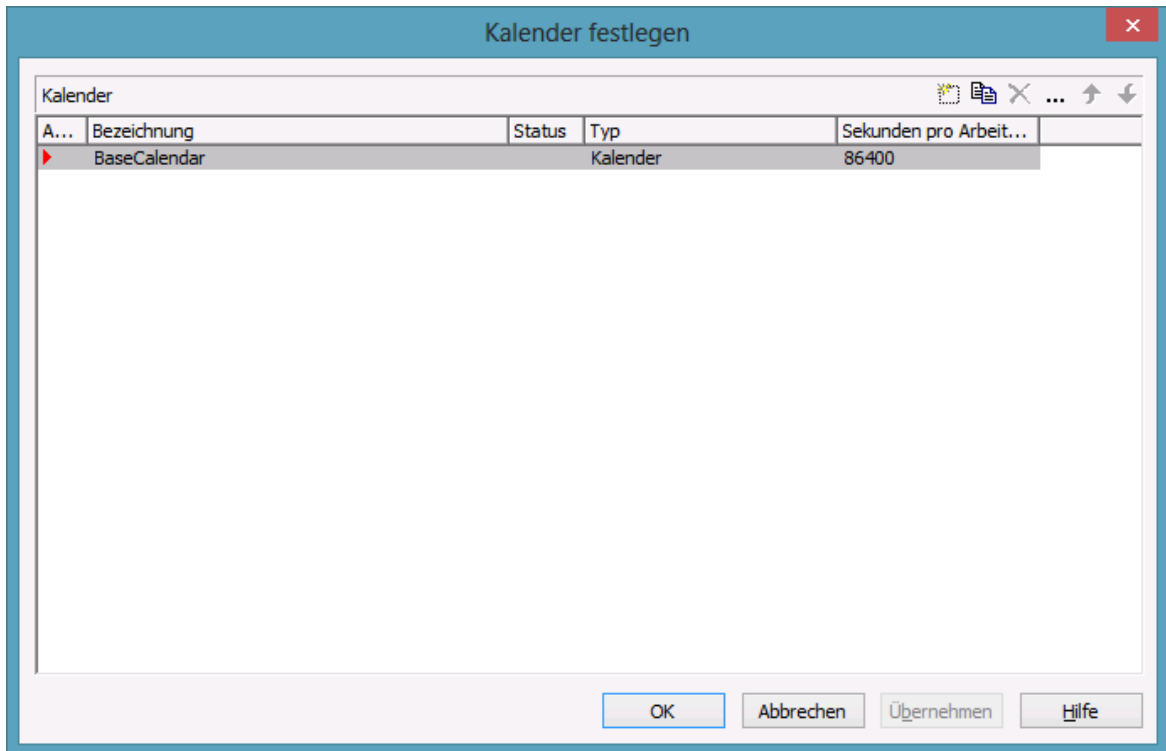
Hintergrundfarbe oder Musterfarbe 2

Wählen Sie hier die Hintergrund-Farbe oder eine zweite Musterfarbe aus.

Vorschau

Hier wird das Aussehen des Musters mit den gewählten Einstellungen angezeigt.

4.29 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.

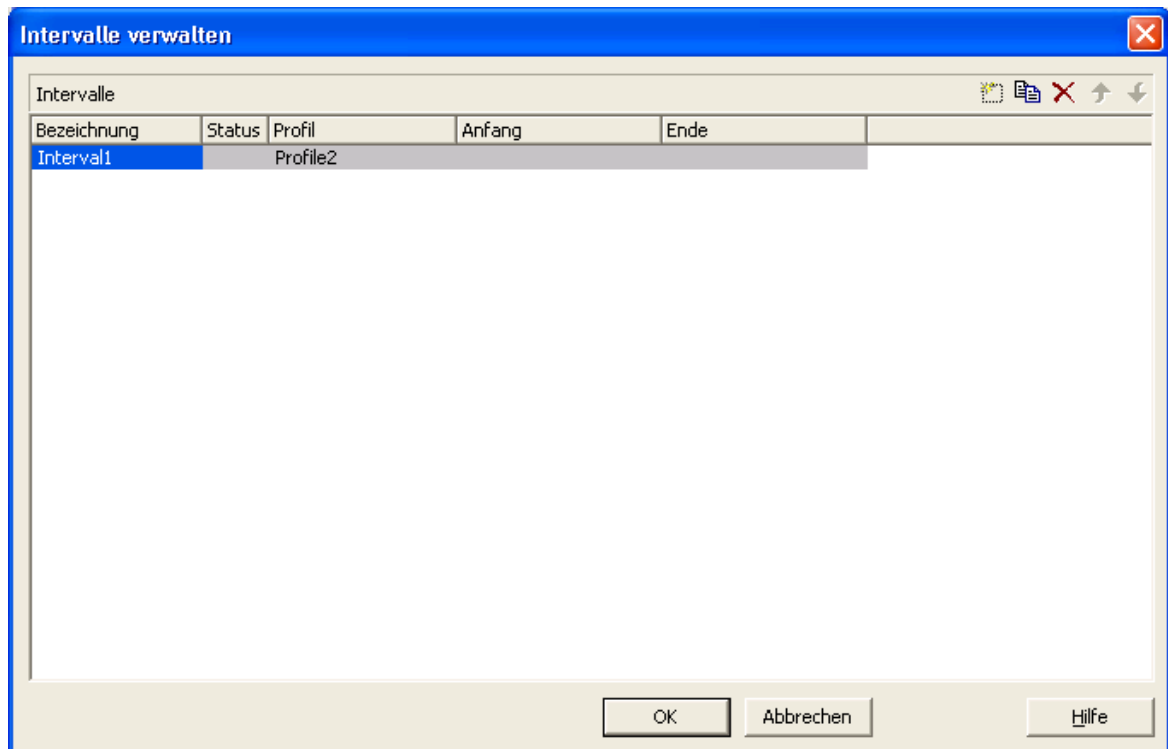
Kalender löschen

 Der markierte Kalender wird gelöscht.

Kalender bearbeiten

 Wenn Sie diese Schaltfläche anklicken, öffnet sich der Dialog **Kalender bearbeiten**.

4.30 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.

Intervall hinzufügen

 Ein neues Intervall mit einem Standardnamen wird angelegt. Doppelklicken Sie auf den markierten Namen um ihn zu verändern.

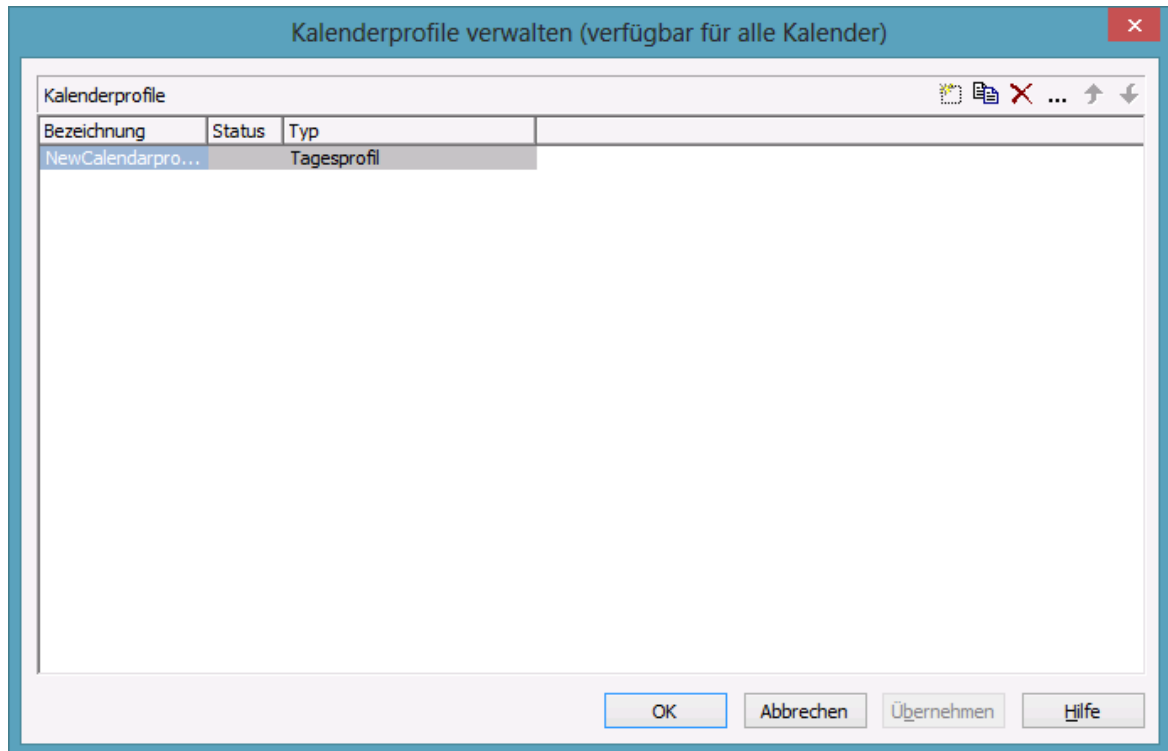
Intervall kopieren

 Es wird eine Kopie des markierten Intervalls angelegt.

Intervall löschen

 Das markierte Intervall wird gelöscht.

4.31 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.

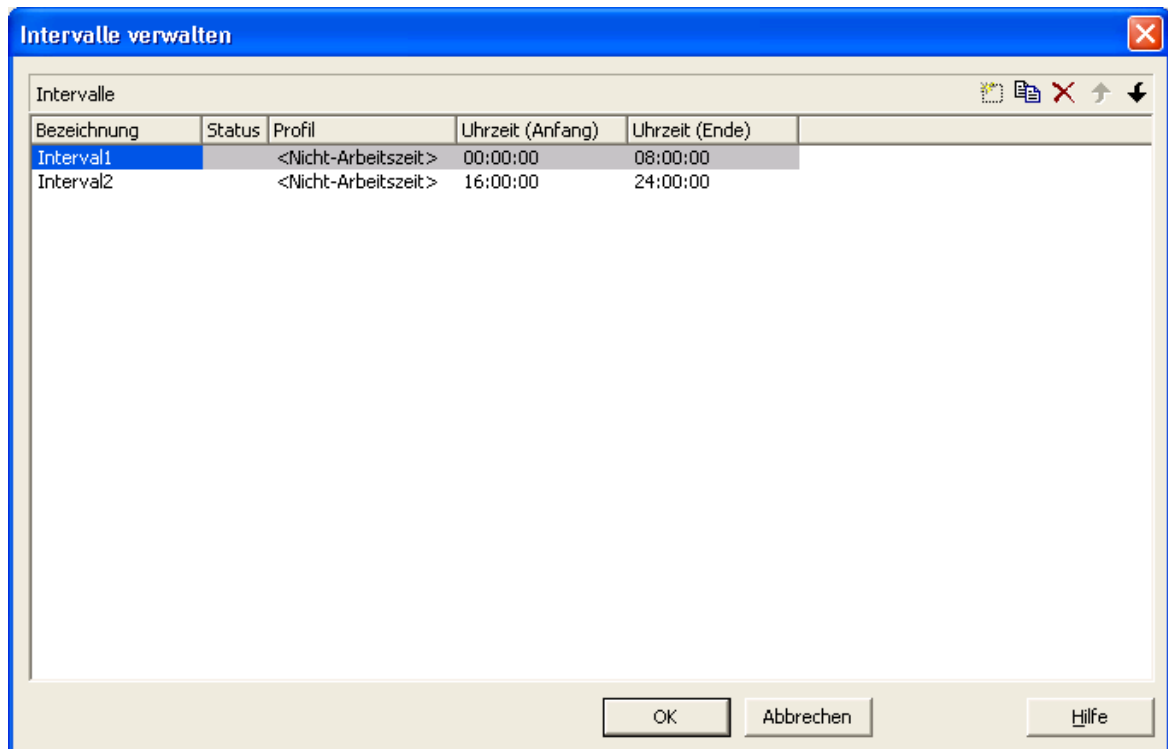
Kalenderprofil löschen

 Das markierte Kalenderprofil wird gelöscht.

Kalenderprofil bearbeiten

 Wenn Sie diese Schaltfläche anklicken, öffnet sich der Dialog **Intervalle verwalten** (Kalenderprofile).

4.32 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.

Intervall hinzufügen



Ein neues Intervall mit einem Standardnamen wird angelegt. Doppelklicken Sie auf den markierten Namen um ihn zu verändern.

Intervall kopieren



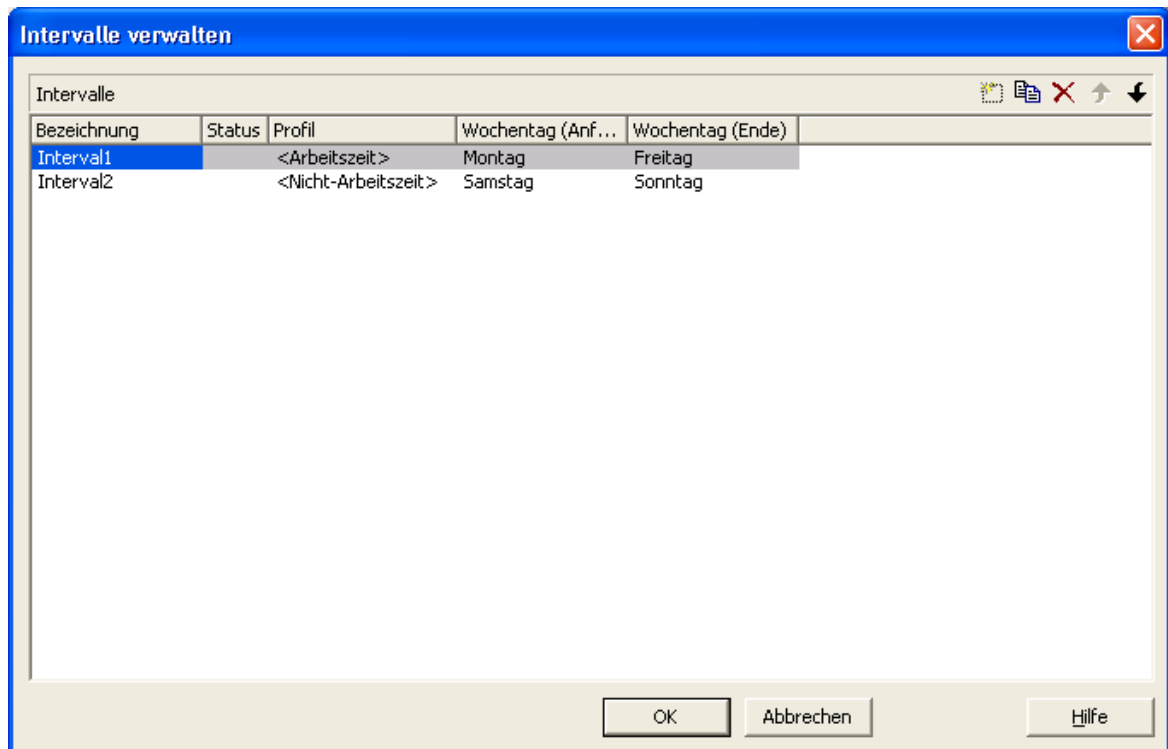
Es wird eine Kopie des markierten Intervalls angelegt.

Intervall löschen




Das markierte Intervall wird gelöscht.

4.33 Dialogfeld "Intervalle verwalten" für Wochenprofil

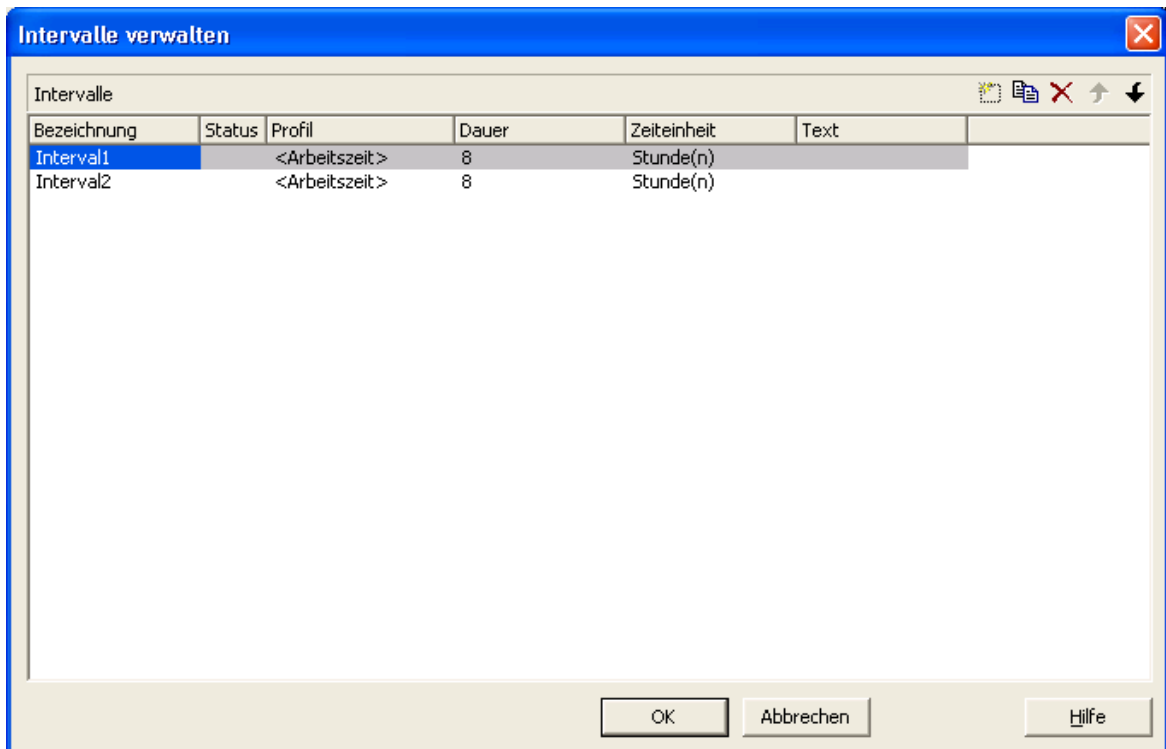


Zu diesem Dialogfeld gelangen Sie, wenn Sie auf der Eigenschaftenseite "Objekte" das Dialogfeld "Kalenderprofile verwalten" aktivieren und dann die "Bearbeiten"-Schaltfläche eines Kalenderprofils drücken. Die unterschiedlichen Profiltypen bieten entsprechende Optionen; dieses Dialogfeld dient dazu, für ein Wochenprofil Intervalle einzurichten und zu verändern.

Wochentag Anfang/Wochentag Ende

Durch Klick auf  können Sie den ersten/letzten Wochentag des Intervalls festlegen.

4.34 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.

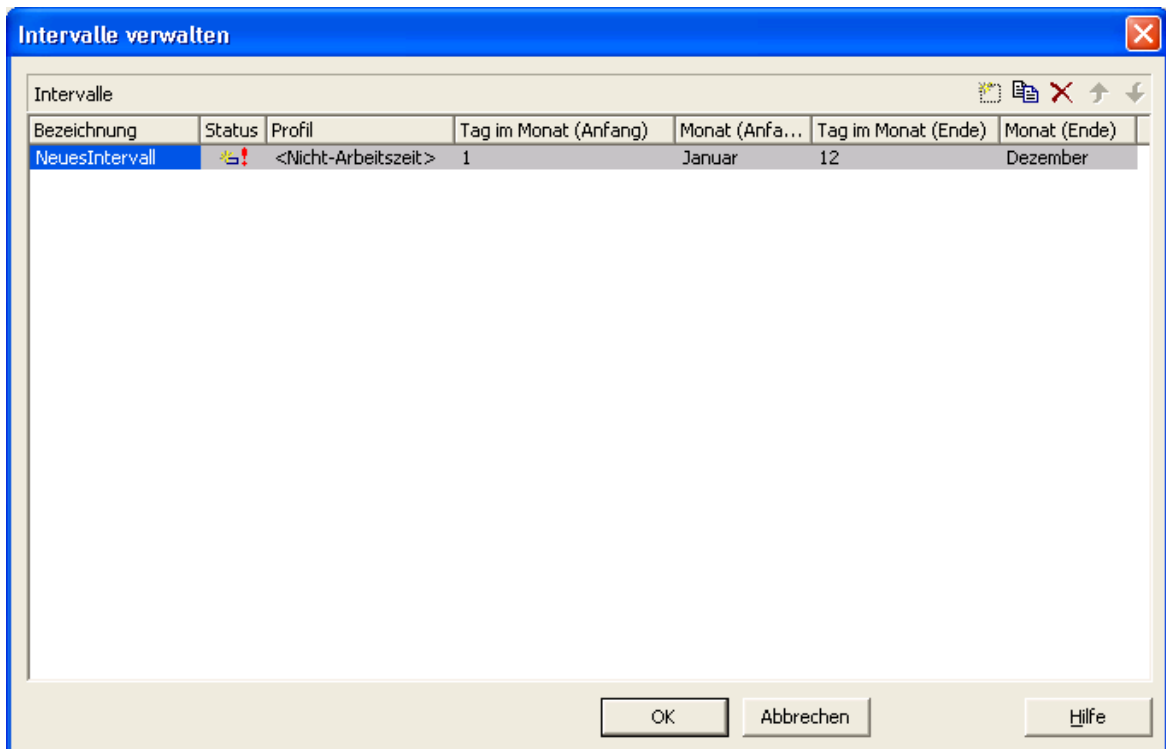
Zeiteinheit

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.

4.35 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.

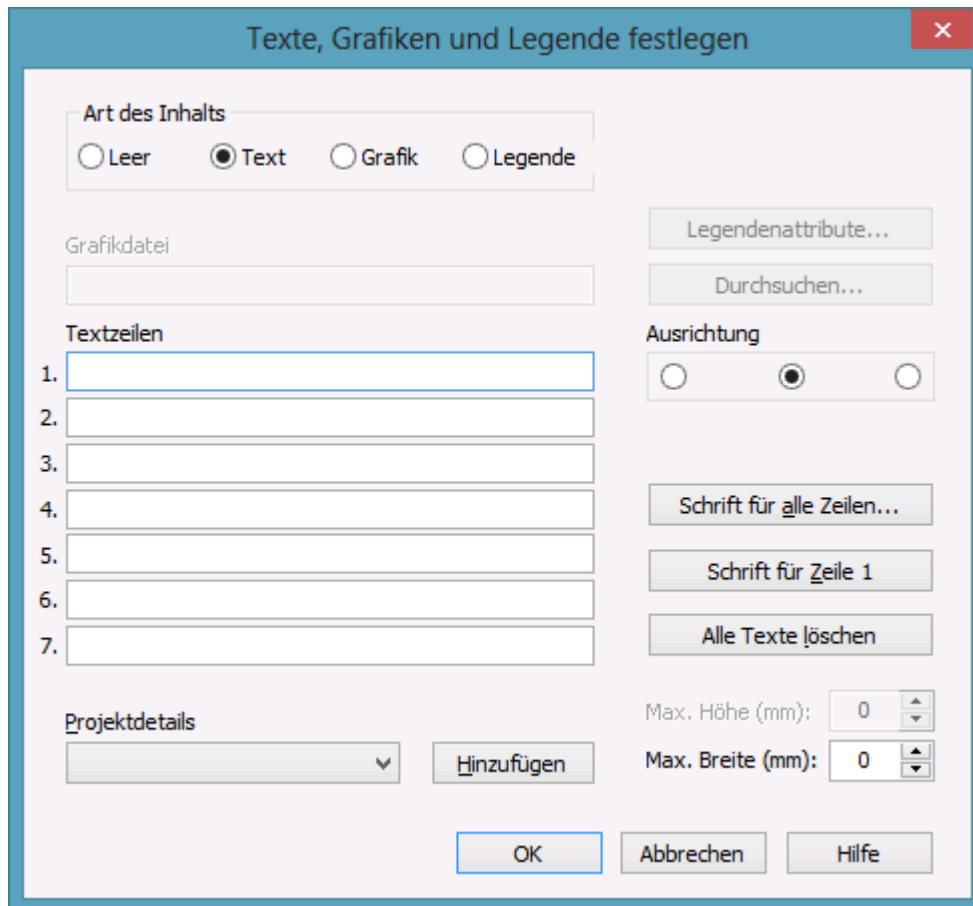
Tag im Monat (Anfang)/Tag im Monat (Ende)

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.

4.36 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.

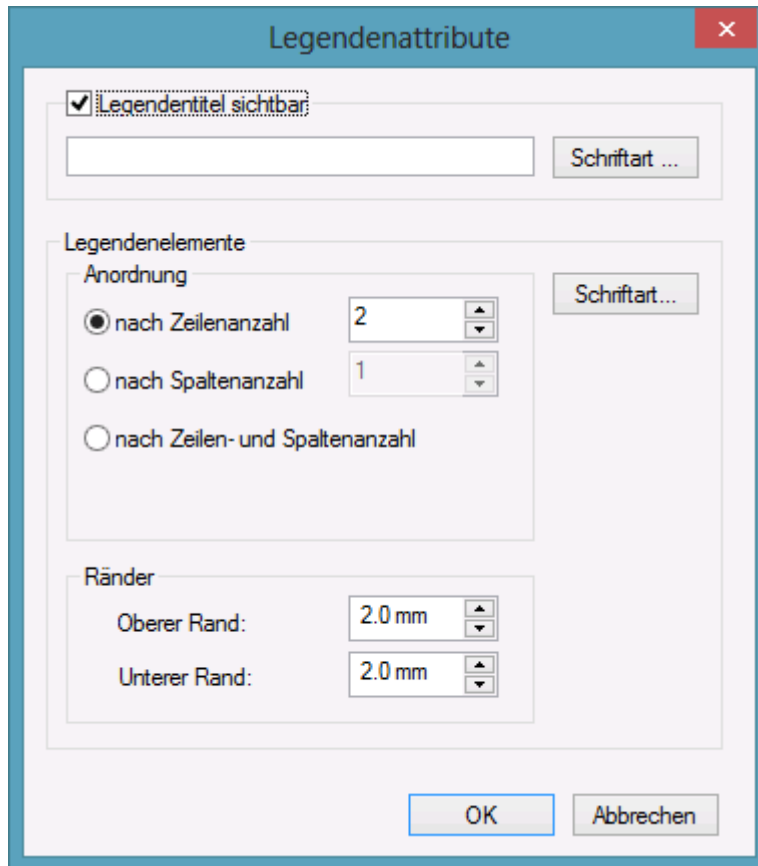
Max. Höhe (mm)

*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.*

4.37 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 Layer, so werden die überzähligen Layer nicht dargestellt.

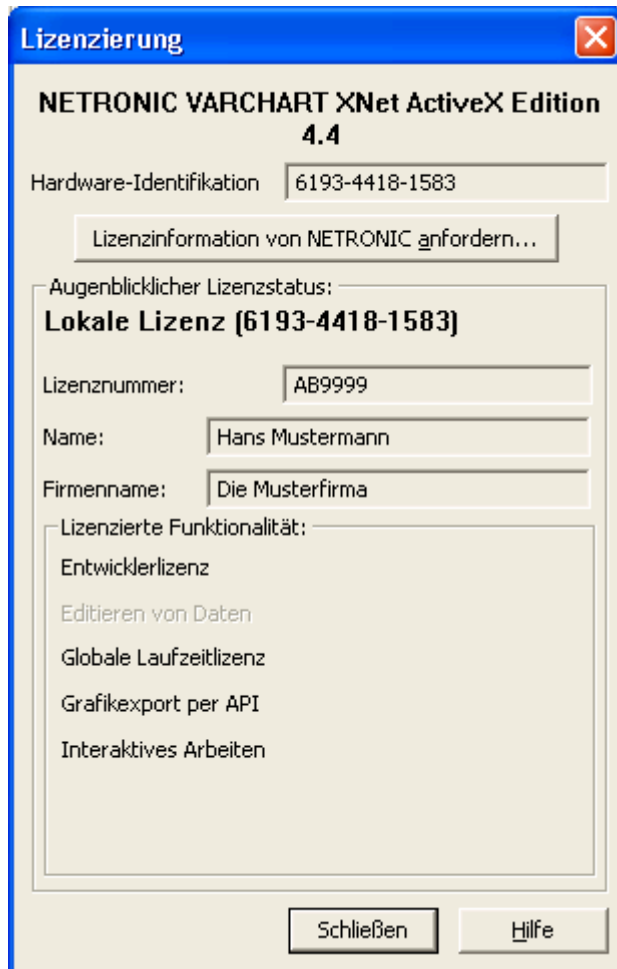
Ränder

- 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.

4.38 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**
- **Globale Laufzeitlizenz** (das VARCHART ActiveX-Steuerelement läuft im Runtime-Modus auf jedem anderen Rechner.)
- **Einzelplatz-Laufzeitlizenzen** (das VARCHART ActiveX-Steuerelement 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.

4.39 Dialogfeld "Lizenzinformationen anfordern"

Lizenzinformationen anfordern

NETRONIC VARCHART XNet ActiveX Edition 4.4

Hardware-Identifikation:

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 (**vcnet.lic**) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei **vcnet.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 im Diagramm
- Zoomen
- Knoten und Verbindungen erzeugen
- Knoten und Verbindungen markieren, löschen und verschieben
- Knoten oder Verbindungen bearbeiten
- Legende bearbeiten
- Seite einrichten
- Druckvorschau verwenden

Kontextmenüs (rechte Maustaste):

- für das Diagramm
- für Knoten
- für Verbindungen
- für die Legende

Bei allen Interaktionen wird ein Ereignis ausgelöst, sodass Sie im Programm darüber informiert werden und ggf. darauf reagieren können.

5.2 Navigation im Diagramm

Sie können mit Hilfe der Pfeil-Tasten mit der Markierung von einem Knoten zum anderen springen.

Sie können bei gedrückter Strg-Taste mit Hilfe der Pfeil-Tasten im Diagramm scrollen.

Weitere Tastenkombinationen für die Navigation im Diagramm:

- **Strg + Pos1:** an den linken oberen Diagrammrand scrollen
- **Strg + Ende:** an den rechten unteren Diagrammrand scrollen
- **Strg + Bild rauf/runter:** an den oberen/ unteren Diagrammrand scrollen
- **Strg + Num +** (Nummernblock): Zoom in
- **Strg + Num -** (Nummernblock): Zoom out
- **Strg + Num *** (Nummernblock): zum nächsten Knoten scrollen
- **Strg + Num /** (Nummernblock): Komplettansicht

Mit Hilfe von **Strg + C**, **Strg + X** bzw. **Strg + V** können Sie markierte Knoten kopieren, ausschneiden bzw. einfügen. Mit Hilfe der **Entf**-Taste können Sie markierte Knoten löschen.

5.3 Zoomen

Mithilfe der folgenden Tastenkombinationen lässt sich die Darstellung vergrößern bzw. verkleinern:

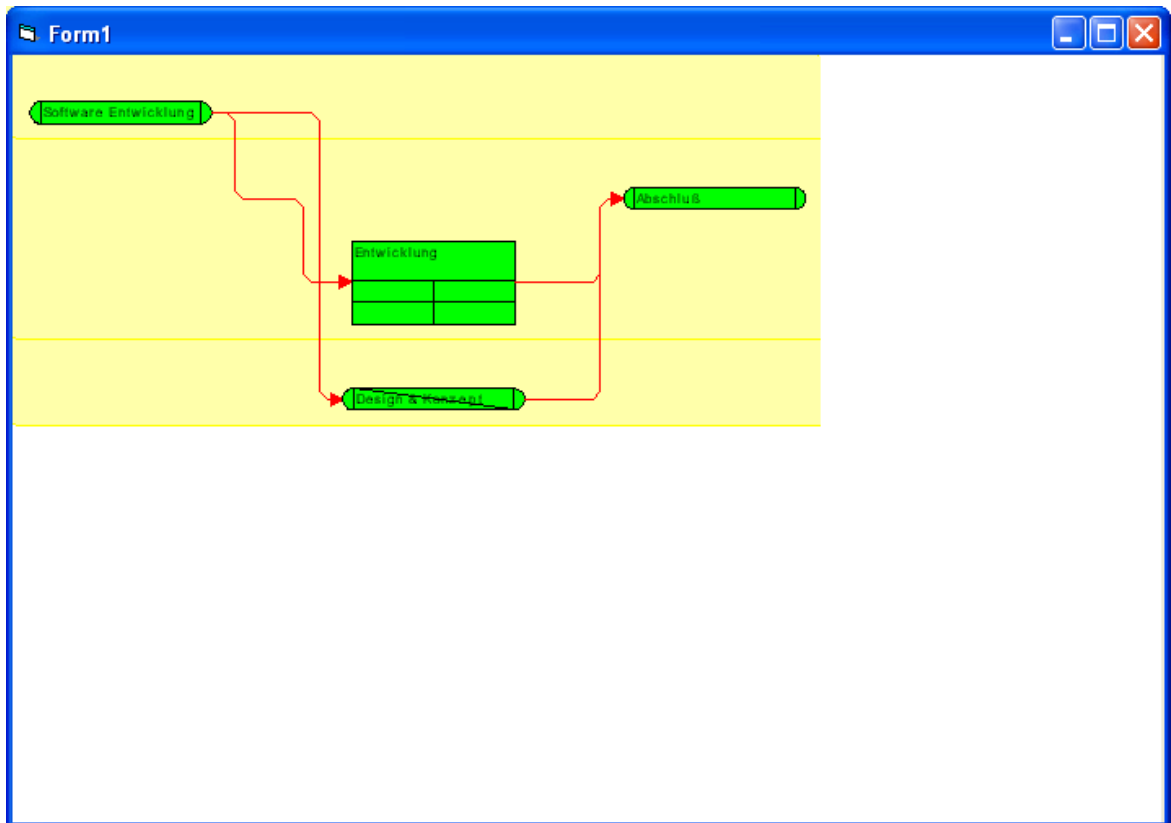
- **Strg + -** (Nummernblock): Verkleinern
- **Strg + +** (Nummernblock): Vergrößern

Auch die Maus kann zum Zoomen genutzt werden:

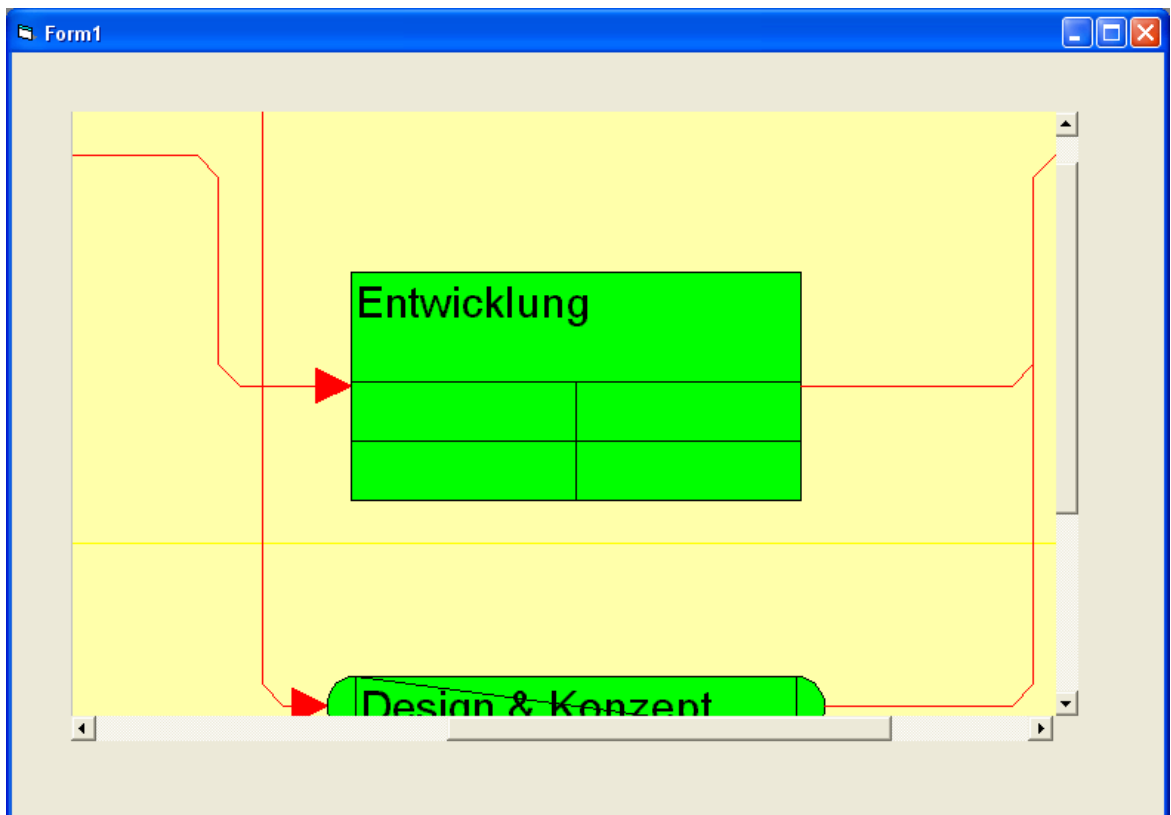
- Drehen Sie das Mousrad während Sie die Strg-Taste gedrückt halten. Dazu muss das Zoomen per Mousrad zugelassen sein. Dies geschieht entweder über die Option **Zoomen per Mousrad zulassen** auf der Eigenschaftenseite **Allgemeines** oder über die API-Eigenschaft **VcNet1.-ZoomingPerMouseWheelAllowed**. (Diese Eigenschaft ist standardmäßig deaktiviert.)
- Sie können einen Ausschnitt Ihres Diagramms bildschirmfüllend darstellen lassen, indem Sie mit gedrückter linker Maustaste ein Rechteck um den zu vergrößernden Ausschnitt aufziehen und dann (bei noch gedrückter linker Maustaste) die rechte Maustaste drücken. Mit Hilfe der Bildlaufleiste können Sie dann das Fenster wie eine Lupe über der Darstellung verschieben und so auch die anderen Bereiche der Darstellung in derselben Vergrößerung betrachten.

Mit der API-Methode **ShowAlwaysCompleteView** können Sie die Darstellung so einstellen, dass stets das komplette Diagramm angezeigt wird. Der Zoomfaktor passt sich bei jeder Änderung des Diagramms automatisch an. Der maximale Zoomfaktor von 100% wird nicht überschritten, die Knoten werden also höchstens in Originalgröße dargestellt.

Weitere Information zu den Zoommöglichkeiten für den Druck finden Sie in Kapitel 5.21 "Seite einrichten".



vor dem Zoomen

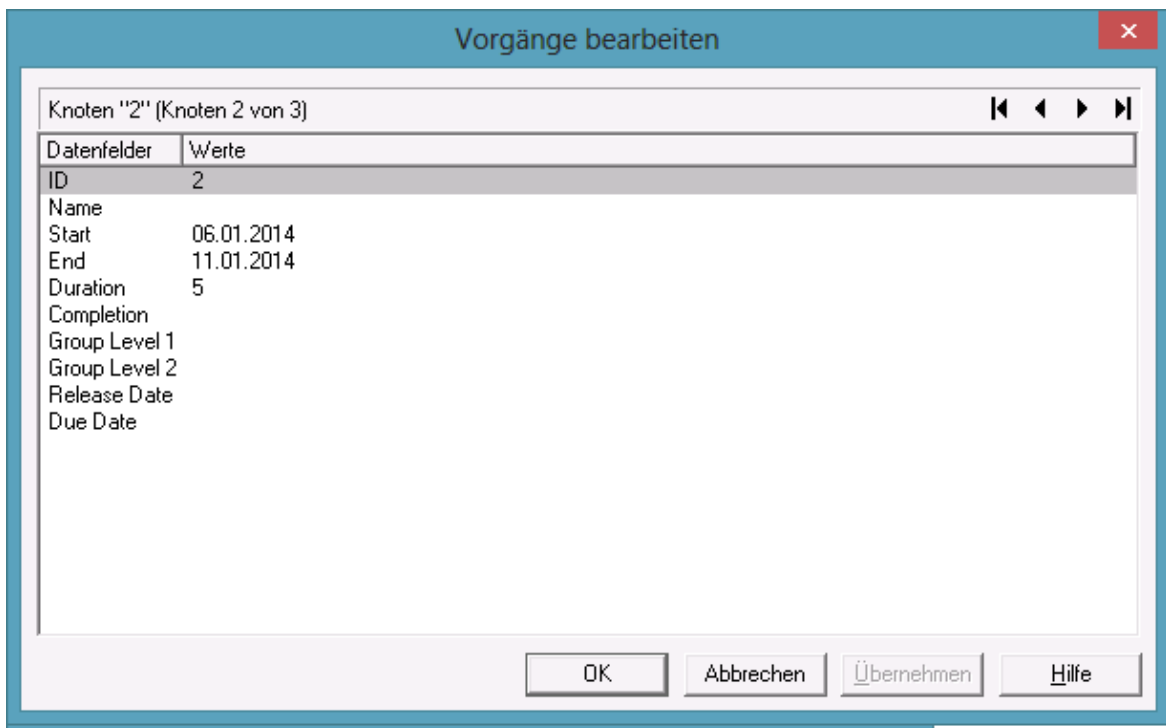


nach dem Zoomen

5.4 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



Durch einen Doppelklick auf einen Knoten wird das Ereignis **OnNodeLDbIClick** 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

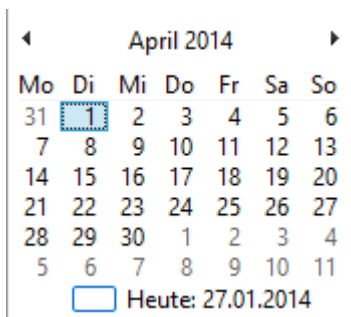
In dieser Spalte werden alle Datenfelder angezeigt, durch die der markierte Knoten beschrieben wird und die **nicht** im Dialog **Datentabellen verwalten**

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.

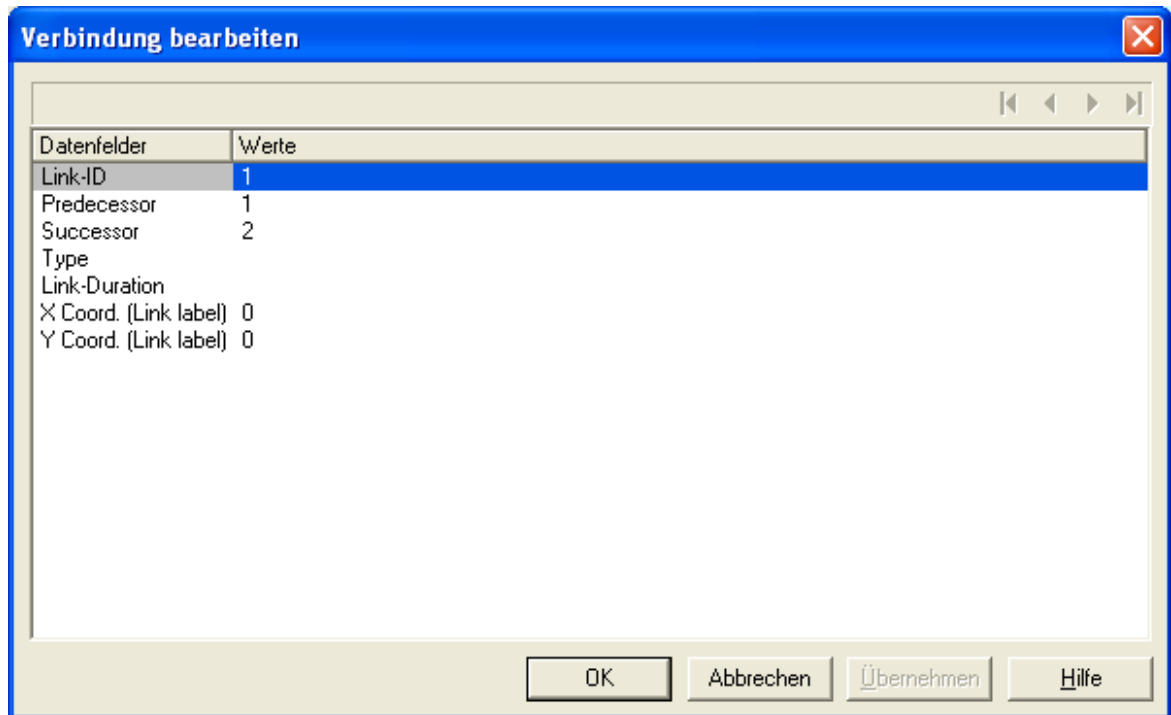
Wenn Sie hier ein Datenfeld vom Typ **Datum/Zeit** bearbeiten, erscheint ein Datumsdialog, in dem Sie das gewünschte Datum anklicken können. Fehler durch die Eingabe eines falschen Datumsformats werden so vermieden.



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.

5.5 Verbindungen bearbeiten



Dieses Dialogfeld erreichen Sie durch einen Doppelklick auf eine Verbindung (Ereignis **OnLinkLDbIcIck**). Hier können Sie die Daten einer markierten Verbindung ansehen und bearbeiten.

An erster Stelle wird die Identifikation (ID) der markierten Verbindung angezeigt.

Datenfelder

In dieser Spalte werden alle Datenfelder angezeigt, durch die die markierte Verbindung beschrieben wird und die nicht in der Datendefinition als versteckt definiert sind. Welche Datenfelder verfügbar sind, hängt von Ihrer Datendefinition ab.

Werte

Sie können in dieser Spalte alle Werte der markierten Verbindung direkt bearbeiten, sofern sie nicht auf der Eigenschaftenseite **Datendefinition** als **nur lesbar** definiert worden sind.

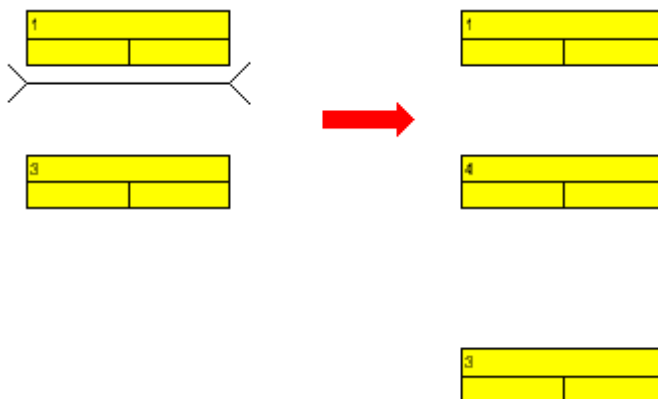
5.6 Knoten und Verbindungen erzeugen

VARCHART XNet besitzt zur Laufzeit zwei grundlegende Modi: den Markiermodus und den Erzeugemodus. Knoten und Verbindungen können Sie nur im Erzeugemodus anlegen. Um in den Erzeugemodus zu wechseln, klicken Sie mit der rechten Maustaste in den freien Diagrammbereich und wählen Sie im Kontextmenü den Befehl **Erzeugemodus**.

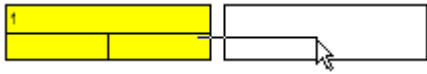


Im Erzeugemodus wird der Cursor im leeren Diagramm zu einem Knotenphantom in der Form eines Rechtecks. Nun können Sie Knoten erzeugen, indem Sie mit der linken Maustaste in den Diagrammbereich klicken.

Wenn Sie die Maus zwischen zwei dicht übereinander oder nebeneinander stehende Vorgänge führen, um dazwischen einen neuen Vorgang anzulegen, verändert das Phantom seine Form und wird zu einer waagerechten Linie mit zwei nach innen gerichteten Pfeilspitzen ("Knochen"). Klicken Sie nun mit der linken Maustaste, so wird der neue Vorgang zwischen den beiden vorhandenen Vorgängen eingefügt.



Verbindungen erzeugen Sie, indem Sie die Maus mit gedrückter linker Maustaste von einem Knoten zu einem anderen Knoten ziehen. Dabei wird der Cursor zu einem Pfeil-Symbol mit einem Knoten-Phantom.



Die Verbindung wird erzeugt, sobald Sie die Maustaste loslassen. Lassen Sie die Maustaste im leeren Diagrammbereich los, wird dort ein neuer Knoten zusammen mit einer Verbindung vom Ausgangsknoten erzeugt. Auf diese Weise können Sie Knoten und Verbindungen gemeinsam erzeugen.

5.7 Knoten und Verbindungen markieren, löschen oder verschieben

Einen Knoten oder eine Verbindung markieren Sie, indem Sie den Knoten bzw. die Verbindung mit der linken Maustaste anklicken. Mehrere Knoten werden markiert, indem Sie diese bei gedrückter Umschalt- oder Strg-Taste mit der linken Maustaste anklicken. (Bei gedrückter Umschalt-Taste werden dabei zusätzlich die Verbindungen markiert.) Sie können nun beispielsweise alle markierten Knoten auf einmal mit der **Entfernen**-Taste oder mit dem Befehl **Löschen** aus dem Kontextmenü für Knoten löschen.

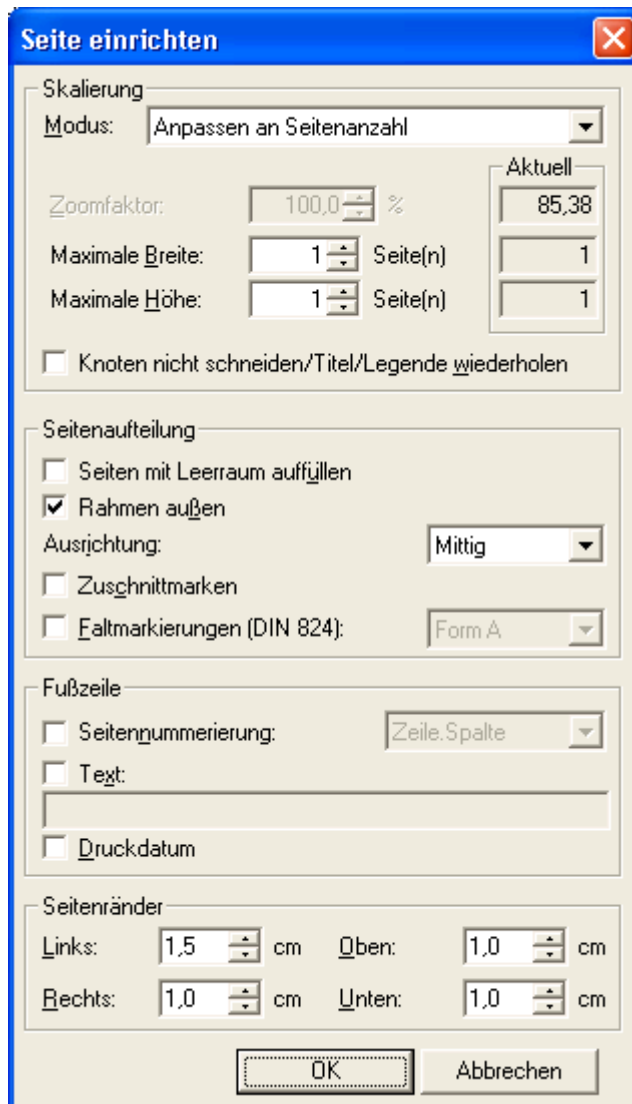
Sie können auch mehrere Knoten und die zugehörigen Verbindungen auf einmal markieren, indem Sie mit der Maus ein Rechteck darum aufziehen.

Wenn Sie im Markiermodus den Cursor auf einem Knoten positionieren und die linke Maustaste drücken, können Sie ihn beliebig verschieben, solange Sie die linke Maustaste gedrückt halten. Dabei werden die zugehörigen Verbindungen automatisch angepasst.

Wenn Sie im Markiermodus den Cursor auf einer Verbindung positionieren und die linke Maustaste drücken, wird der Cursor zu einem kleinen Quadrat mit vier Pfeilen. Sie können die ausgewählte Verbindung beliebig verschieben, solange Sie die linke Maustaste gedrückt halten.

5.8 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.



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.

Knoten nicht durchtrennen/Titel/Legende wiederholen

Aktivieren Sie dieses Kontrollkästchen, damit bei der Ausgabe des Diagramms auf mehreren Seiten Knoten nicht durchtrennt werden, und damit Titel und Legende - sofern vorhanden - auf jeder Seite ausgegeben werden.

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 **Knoten nicht durchtrennen/Titel 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

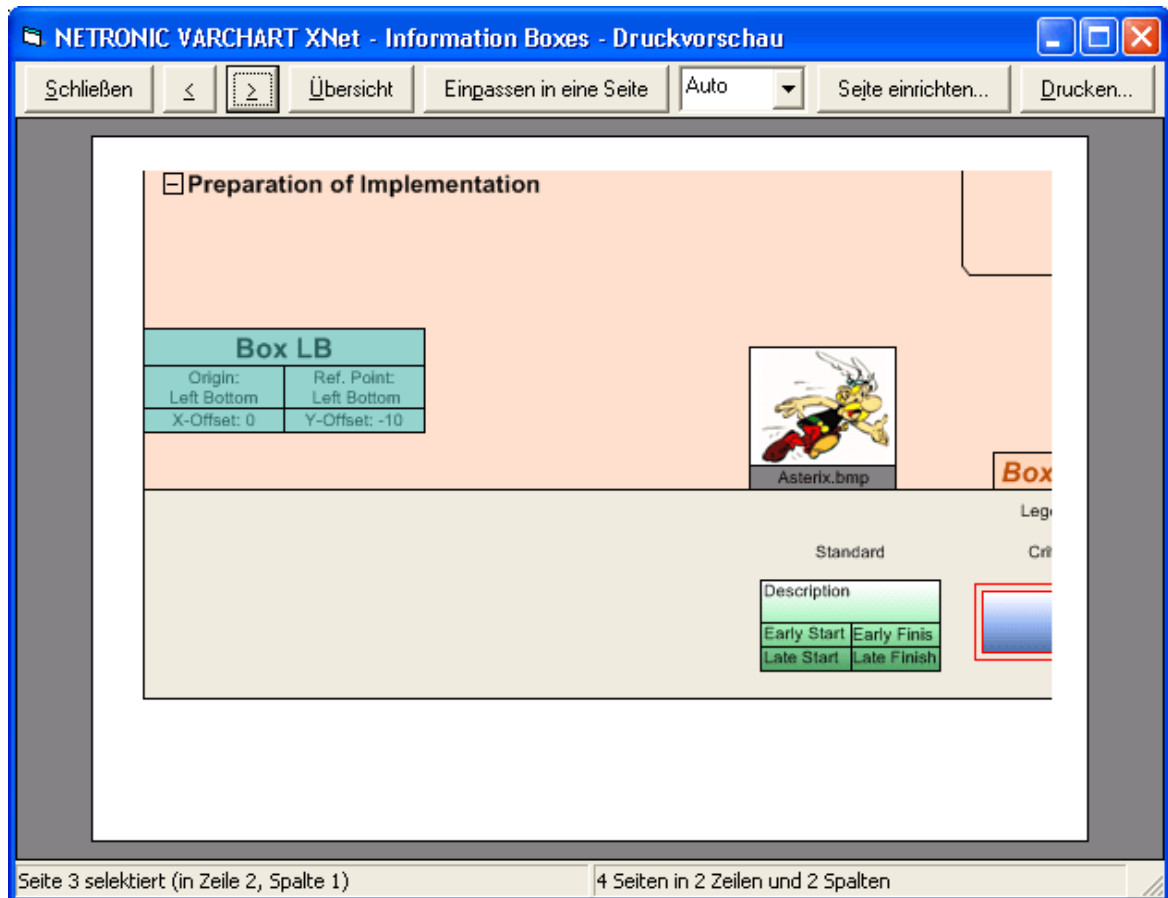
Druckdatum

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.

5.9 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


einzelnen 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 Mousrades 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 Mousrad 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

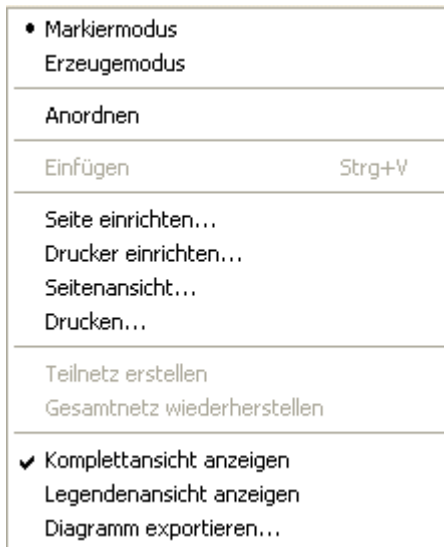
Über diese Schaltfläche gelangen Sie in den Windows-Dialog **Drucken** und können von dort aus den Druckvorgang einleiten.

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.

5.10 Kontextmenü für das Diagramm

Wenn Sie die rechte Maustaste drücken, wenn der Cursor im Diagrammbereich (nicht auf einem Objekt) steht, öffnet sich das folgende Kontextmenü:



Markiermodus

Der Markiermodus ist der Standardmodus.

Erzeugemodus

Dieser Modus kann nur eingeschaltet werden, wenn auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten und Verbindungen zulassen** gesetzt ist.

Der Cursor wird im leeren Diagrammbereich zu einem Knotenphantom in der Form eines Rechtecks. Jeder Klick mit der linken Maustaste erzeugt in diesem Modus einen neuen Knoten. Wenn die Option **Neuen Knoten bearbeiten** auf der Eigenschaftenseite **Allgemeines** 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 im Erzeugemodus die Maus von einem Knoten zu einem anderen ziehen, wird der Cursor zu einem kleinen Bleistift, und eine Verbindung zwischen diesen Knoten wird erzeugt. Wenn die Option **Neue Verbindung bearbeiten** auf der Eigenschaftenseite **Allgemeines** gewählt ist, öffnet sich das Dialogfeld **Vorgänge bearbeiten**, sobald Sie die Maustaste loslassen. Hier können Sie alle Daten der neu angelegten Verbindung bearbeiten.



Der Erzeugemodus lässt sich auf zwei Arten aktivieren:

1. über das standardmäßige Kontextmenü im Diagrammbereich
2. über das Setzen der VcNet-Eigenschaft **InteractionMode** auf den Wert **vcCreateNodesAndLinks**.

Anordnen

Über diesen Menüpunkt werden die vom Anwender verschobenen Knoten inklusive der zugehörigen Verbindungen optimal angeordnet.

Einfügen

Dieser Menüpunkt steht nur zur Verfügung, wenn Sie zuvor einen oder mehrere Knoten ausgeschnitten oder kopiert haben. Diese(n) können Sie dann mit diesem Befehl an der Position des Cursors einfügen.

Seite einrichten

Sie gelangen in das Dialogfeld **Seite einrichten**.

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**.

Druckvorschau

Sie gelangen in das Dialogfeld **Druckvorschau**.

Drucken

Wenn Sie diese Option wählen, gelangen Sie in das Windows-Dialogfeld **Drucken**.

Teilnetz erstellen

(nur aktiv, wenn Knoten markiert sind) Wählen Sie diese Option, um ein Teilnetz aus den markierten Knoten darzustellen.

Gesamtnetz wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teilnetz erstellen** gewählt wurde)* Wählen Sie diese Option, um das Gesamtnetz wiederherzustellen.

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 VcNet 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

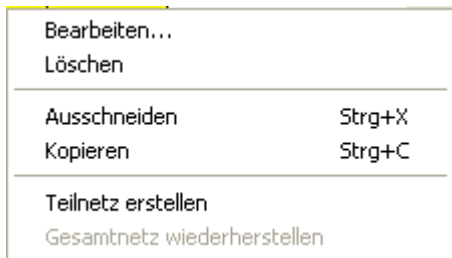
komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.

- 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".

5.11 Kontextmenü für Knoten

Wenn ein oder mehrere Knoten markiert sind und Sie die rechte Maustaste drücken, öffnet sich das folgende Kontextmenü:



Bearbeiten

Wenn Sie diese Option wählen, öffnet sich das Dialogfeld **Vorgänge bearbeiten**.

Löschen

Wenn Sie diese Option wählen, werden die markierten Knoten gelöscht.

Ausschneiden

Mit diesem Befehl können Sie die markierten Knoten ausschneiden.

Kopieren

Mit diesem Befehl können Sie die markierten Knoten kopieren.

Teilnetz erstellen

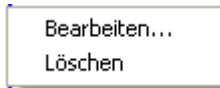
Wählen Sie diese Option, um ein Teilnetz aus den markierten Knoten darzustellen.

Gesamtnetz wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teilnetz erstellen** gewählt wurde)* Wählen Sie diese Option, um das Gesamtnetz wiederherzustellen.

5.12 Kontextmenü für Verbindungen

Wenn Sie eine Verbindung mit der rechten Maustaste anklicken, erscheint folgendes Menü:



Bearbeiten

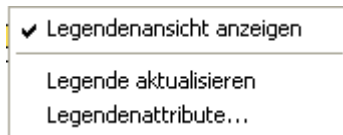
Mit diesem Befehl öffnen Sie das Dialogfeld **Verbindung bearbeiten**, in dem Sie alle Daten der markierten Verbindung bearbeiten können.

Löschen

Wenn Sie die markierte Verbindung löschen möchten, bestätigen Sie dies durch einen Klick mit der linken Maustaste.

5.13 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".

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 (vcnet.lic) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei *vcnet.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-Steuer-elementes verwendet wird?

Einige Einstellungen für das VARCHART ActiveX-Steuer-element 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-Steuer-elementes verwendet.

6.3 Was müssen Benutzer von Borland Delphi beim Upgrade auf eine neue Version von VARCHART XNet tun?

Nach dem Upgrade oder Update von VARCHART XNet 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 XNet* 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 *VcNetLib_TLB.pas* und *VcNetLib_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 *vcnet.ocx* aus und klicken Sie auf **Öffnen**. Jetzt sehen Sie *NETRONIC VARCHART XNet* 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 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 **Allgemeines** das Kontrollkästchen **Neue Knoten zulassen** aktiviert ist.

Kontrollieren Sie außerdem, ob in Ihrem Programmcode die VARCHART-VcNet-Eigenschaft **AllowNewNodesAndLinks** auf **False** gesetzt wurde; dann können ebenfalls keine neuen Knoten angelegt werden.

6.5 Wieso können Verbindungen u. U. nicht interaktiv erzeugt werden?

Wenn Sie zur Laufzeit keine Verbindungen zwischen Knoten interaktiv anlegen können, sind verschiedene Ursachen dafür möglich:

1. Kontrollieren Sie, ob auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **Neue Knoten und Verbindungen zulassen** aktiviert ist. Wenn es aktiviert ist, können Sie Knoten und Verbindungen interaktiv anlegen.
2. Wenn Sie nun immer noch keine Verbindungen sehen können, prüfen Sie Ihre Einstellungen für die Verbindungen. Möglicherweise können Sie Verbindungen zwar anlegen, diese aber nicht sehen. Öffnen Sie dazu die Eigenschaftenseite **Verbindungen** und kontrollieren Sie dort die Linienart jedes Verbindungsaussehens. Wenn Sie feststellen, dass Verbindungslinien dieselbe Farbe haben wie der Diagrammhintergrund, wählen Sie eine Farbe, die sich deutlich davon abhebt.
3. Prüfen Sie die Filterbedingungen. Falsch definierte Filterbedingungen können zur Folge haben, dass Verbindungen nicht dargestellt werden.
4. Wenn das Verbindungsausssehen sinnvoll definiert ist, Sie aber immer noch keine Verbindungen sehen, prüfen Sie, ob die Datenfelder für die Verbindungen (**Vorgänger**, **Nachfolger**, **Verbindungstyp**) richtig definiert sind.

6.6 Wie verhindert man das interaktive Erzeugen von Knoten?

Sie können auf verschiedene Weisen verhindern, dass Knoten und Verbindungen interaktiv angelegt werden können:

1. Deaktivieren Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **Neue Knoten und Verbindungen zulassen**.
2. Setzen Sie Rückgabestatus von **OnNodeCreate** auf **vcRetStatFalse**, damit interaktiv erzeugte Knoten wieder gelöscht werden.
3. Ergänzen Sie die folgenden Codezeilen:

Code-Beispiel

```
Sub Form_Load
    VcNet1.AllowNewNodesAndLinks = False
End Sub
```

6.7 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 VcNet1_OnDiagramRClick(ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub

' Kontextmenü für Verbindungen abschalten
Private Sub VcNet1_OnLinkRClickCltn(ByVal linkCltn As _
    VcNetLib.VcLinkCollection, _
    ByVal x As Long, _
    ByVal y As Long, _
    returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub

' Kontextmenü für Knoten abschalten
Private Sub VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
    ByVal location As _
    VcNetLib.LocationEnum, _
    ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub
```

6.8 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.

6.9 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

```
VcNet1.SuspendUpdate (True)

    If updateFlag Then
        For Each node In nodeCltn
            If node.DataField(2) < "07.09.98" Then
                node.DataField(13) = "X"
                node.UpdateNode
                counter = counter + 1
            End If
        Next node
    Else
        For Each node In nodeCltn
            If node.DataField(2) < "07.09.98" Then
                node.DataField(13) = ""
                node.UpdateNode
                counter = counter + 1
            End If
        Next node
    End If

VcNet1.SuspendUpdate (False)
```

Die Performance lässt sich analog auch beim Update von vielen Verbindungen verbessern.

> 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.

6.10 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-Steurelement 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!

7 API Referenz

7.1 Objekttypen

- DataObject
- DataObjectFiles
- VcBorderArea
- VcBorderBox
- VcBox
- VcBoxCollection
- VcBoxFormat
- VcBoxFormatCollection
- VcBoxFormatField
- VcCalendar
- VcCalendarCollection
- VcCalendarProfile
- VcCalendarProfileCollection
- VcDataDefinition
- VcDataDefinitionTable
- VcDataRecord
- VcDataRecordCollection
- VcDataTable
- VcDataTableCollection
- VcDataTableField
- VcDataTableFieldCollection
- VcDefinitionField
- VcFilter
- VcFilterCollection
- VcFilterSubCondition
- VcGroup
- VcGroupCollection
- VcInterval
- VcIntervalCollection
- VcLegendView
- VcLink
- VcLinkAppearance
- VcLinkAppearanceCollection

- VcLinkCollection
- VcLinkFormat
- VcLinkFormatCollection
- VcLinkFormatField
- VcMap
- VcMapCollection
- VcMapEntry
- VcNet
- VcNode
- VcNodeAppearance
- VcNodeAppearanceCollection
- VcNodeCollection
- VcNodeFormat
- VcNodeFormatCollection
- VcNodeFormatField
- VcPrinter
- VcRect
- VcScheduler
- 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

- Files

Methoden

- Clear
- GetData
- GetFormat
- SetData

Eigenschaften

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
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
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 **RegisterClip-**

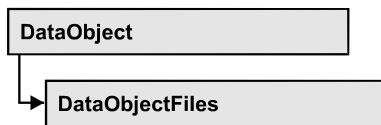
boardFormat() registriert sind. Hierbei sind jedoch einige Einschränkungen zu berücksichtigen:

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.
⇒ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vcCFTxt; 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
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

300 API-Referenz: DataObjectFiles

Code-Beispiel

```
Private Sub VcNet1_OLEDragOver(ByVal data As VcNetLib.DataObject, effect As Long, ByVal button As Integer, ByVal Shift As Integer, ByVal x As Long, ByVal y As Long, ByVal state As VcNetLib.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 Eigenschaftename weggelassen werden. Beispiel: DataObjectFiles(0) erfragt den ersten Dateinamen.

	Datentyp	Beschreibung
Parameter: ⇒ index	Long	Index des Dateinamens {0...Count-1}
Eigenschaftswert	String	Dateiname

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
Rückgabewert	Void	

Clear

Methode von DataObjectFiles

Mit dieser Methode können Sie alle Dateinamen der Liste löschen.

	Datentyp	Beschreibung
Rückgabewert	Void	

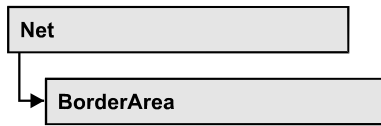
Remove

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.
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

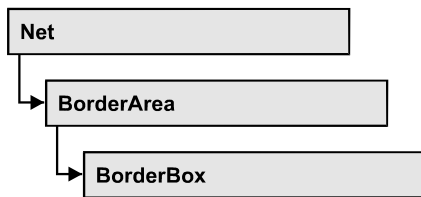
Code-Beispiel

```

Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
  
```

7.5 VcBoundingBox



Ein Objekt vom Typ **VcBoundingBox** 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 VcBoundingBox

Mit dieser Eigenschaft können Sie die Ausrichtung dieses BorderBox-Objektes festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoundingBoxAlignmentEnum Mögliche Werte: vcBBXACentered -1 vcBBXALeft -3	Ausrichtung der BorderBox 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

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxTR As VcBoundingBox
```

```

Set borderArea = VcNet1.BorderArea
Set bBoxTR = borderArea.BorderBox(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

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

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

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

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 VcBoundingBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTLegend
logThis (bBoxBBL.LegendFont.Name)
```

LegendTitle

Eigenschaft von VcBoundingBox

Mit dieser Eigenschaft können Sie den Legendentitel angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendentitel

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBoundingBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
```

LegendTitleFont**Eigenschaft von VcBoundingBox**

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 VcBoundingBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vbBXTLegend
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)

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBoundingBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(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}
Eigenschaftswert	String	Text in Textfeldern

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBoundingBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(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	Zeilenindex {0..6}
Eigenschaftswert	StdFont	Schriftattribute des Textes

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxTL As VcBorderBox

Set borderArea = VcNet1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)

bBoxTL.TextFont(i).Bold = False
bBoxTL.TextFont(i).Italic = False
bBoxTL.TextFont(i).Name = "Symbol"
```

Code Sample in C#

```
/ Text for Title
VcBorderBox borderBox =
VcNet1.BorderArea.BorderBox(VcBorderBoxPosition.vcBBXPTopCentered);
borderBox.Type = VcBorderBoxType.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.

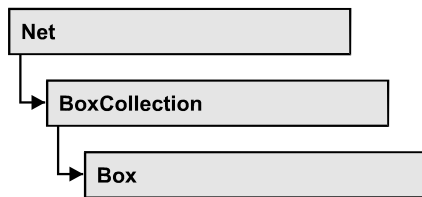
	Datentyp	Beschreibung
Eigenschaftswert	BorderBoxTypeEnum	Typ der Box
	Mögliche Werte: vcBBXTGraphics 3 vcBBXTLegend 4 vcBBXTNothing 0 vcBBXTText 1 vcBBXTTextWithGraphics 2	Grafik Legende Leer Text Text und Grafik

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcNet1.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

- FieldText
- FormatName
- LineColor
- LineThickness
- LineType
- MarkBox
- Moveable
- Name
- Origin
- Priority
- ReferencePoint
- Specification
- UpdateBehaviorName
- Visible

Methoden

- GetActualExtent
- GetTopLeftPixel
- GetXYOffset
- GetXYOffsetAsVariant
- IdentifyFormatField
- SetXYOffset
- SetXYOffsetByTopLeftPixel

Eigenschaften

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	Feldindex
Eigenschaftswert	String	Feldinhalt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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 = VcNet1.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 = VcNet1.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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineThickness = 2
```

LineStyle

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Typ der Randlinie der Box festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	LineStyleEnum	Linientyp Standardwert: vcSolid
	Mögliche Werte: vcDashed 4 vcDashedDotted 5 vcDotted 3 vcLineStyle0 100 vcLineStyle1 101 vcLineStyle10 110 vcLineStyle11 111 vcLineStyle12 112 vcLineStyle13 113 vcLineStyle14 114 vcLineStyle15 115 vcLineStyle16 116 vcLineStyle17 117 vcLineStyle18 118 vcLineStyle2 102 vcLineStyle3 103	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0 <hr/> Linientyp 1 <hr style="border-top: 1px dashed black;"/> Linientyp 10 <hr style="border-top: 1px dotted black;"/> Linientyp 11 <hr style="border-top: 1px dash-dot black;"/> Linientyp 12 <hr style="border-top: 1px long-dash black;"/> Linientyp 13 <hr style="border-top: 1px solid black;"/> Linientyp 14 <hr style="border-top: 1px dashed black;"/> Linientyp 15 <hr style="border-top: 1px solid black;"/> Linientyp 16 <hr style="border-top: 1px dashed black;"/> Linientyp 17 <hr style="border-top: 1px dashed black;"/> Linientyp 18 <hr style="border-top: 1px dotted black;"/> Linientyp 2 <hr style="border-top: 1px dotted black;"/> Linientyp 3 <hr style="border-top: 1px dashed black;"/>

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 = VcNet1.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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox
Dim boxName As String

Set boxCltn = VcNet1.boxCollection
Set box = boxCltn.FirstBox
boxName = box.Name
MsgBox boxName
```

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 Mögliche Werte:	Ursprungspunkt der Box

vcBOBottomLeft	27	unten links
vcBOBottomRight	29	unten rechts
vcBOCenterCenter	25	mittig mittig
vcBOCenterLeft	24	mittig links
vcBOCenterRight	26	mittig rechts
vcBOTopCenter	22	oben mittig
vcBOTopLeft	21	oben links
vcBOTopRight	23	oben rechts

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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: vcBRPBottomLeft 27 vcBRPBottomRight 29 vcBRPCenterCenter 25 vcBRPCenterLeft 24 vcBRPCenterRight 26	unten links unten rechts mittig mittig mittig links mittig rechts

vcBRPTopCenter 22	oben mittig
vcBRPTopLeft 21	oben links
vcBRPTopRight 23	oben rechts

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.ReferencePoint = vcBRPCenterRight
```

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	Boxspezifikation

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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

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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.BoxCollection
Set box = boxCltn.FirstBox
box.Visible = False
```

Methoden

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
↔ height	Integer	Höhe der Box
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
↔ y	Integer	y-Wert des Offsets
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
↔ yOffset	Integer	y-Wert des Offsets
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
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
⇒ yOffset	Integer	y-Wert des Offsets
Rückgabewert	Boolean	Offset wird gesetzt (True)/nicht gesetzt (False)

Code-Beispiel

```
Dim OffsetSet As Boolean
OffsetSet = VcNet1.boxCollection.FirstBox.SetXYOffset(100, 100)
```

SetXYOffsetByTopLeftPixel

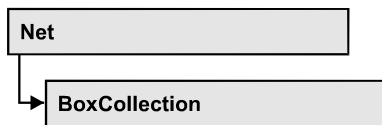
Methode von VcBox

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
⇒ y	Integer	y-Wert des Offsets
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 VcNet1.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 = VcNet1.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	Name der Box
Rückgabewert	VcBox	Neues Boxobjekt

Code-Beispiel

```
Set newBox = VcNet1.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	Boxspezifikation
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	Index der Box
Rückgabewert	VcBox	Ermitteltes Boxobjekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Set boxCltn = VcNet1.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
Rückgabewert	VcBox	Box

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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
⇒ newBoxName	String	Name der neuen Box
Rückgabewert	VcBox	Boxobjekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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 = VcNet1.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 = VcNet1.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
Rückgabewert	Boolean	Box gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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.

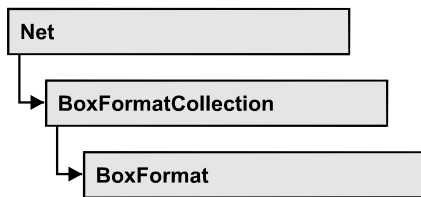
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True)/ nicht erfolgt (False)

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcNet1.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).

Code-Beispiel

```
Dim boxFormat As VcBoxFormat

Set boxFormat = VcNet1.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
Eigenschaftswert	VcBoxFormatField	Boxformatfeld

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcNet1.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

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcNet1.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

Code-Beispiel

```
Dim boxFormat As VcBoxFormat

For Each boxFormat In VcNet1.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

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
Rückgabewert	VcBoxFormatField	Boxformatfeld-Objekt

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcNet1.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 nummeriert sind.

	Datentyp	Beschreibung
Parameter:		
⇒ index	Integer	Index des zu löschenden Boxformatfeldes

332 API-Referenz: VcBoxFormat

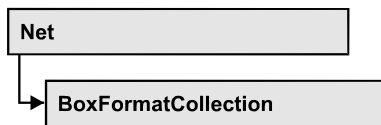
Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim i As Integer

boxFormat = VcNet1.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 VcNet1.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 = VcNet1.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

Rückgabewert	VcBoxFormat	Neues Boxformatobjekt
---------------------	-------------	-----------------------

Code-Beispiel

```
Set newBoxFormat = VcNet1.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	Boxformatspezifikation
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	Name des zu kopierenden Boxformats
⇒ newFormatName	String	Name des neuen Boxformats
Rückgabewert	VcBoxFormat	Boxformatobjekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcNet1.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 = VcNet1.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
Rückgabewert	VcBoxFormat	Ermitteltes Boxformat-Objekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set boxFormatCltn = VcNet1.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
Rückgabewert	VcBoxFormat	Boxformat

Code-Beispiel

```
Dim formatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set formatCltn = VcNet1.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 = VcNet1.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	Name des Boxformats

338 API-Referenz: VcBoxFormatCollection

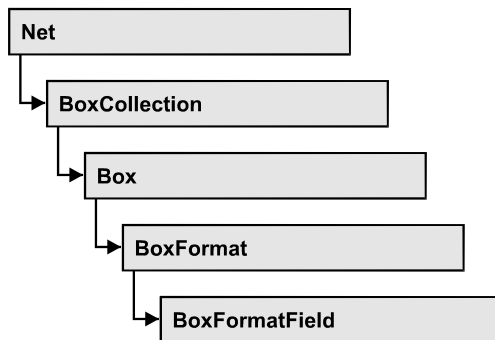
Rückgabewert	Boolean	Boxformat gelöscht (True)/nicht gelöscht (False)
---------------------	---------	--

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcNet1.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.

340 API-Referenz: VcBoxFormatField

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	Ausrichtung des Feldinhalts 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 = VcNet1.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

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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 0 ... 200

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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 = VcNet1.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 0 ... 20

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField
```

```

Set boxFormatCltn = VcNet1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTTText
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 **vcFFTTText** 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 0 ... 20

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTTText
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

Code-Beispiel

```

Dim boxFormatCltn As VcBoxFormatCollection

```

```
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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 Boxformatfeldes Standardwert: -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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.

344 API-Referenz: VcBoxFormatField

	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Boxformatfeldes

Code-Beispiel




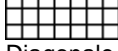

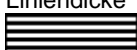
```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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	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	
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien	
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 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
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 boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Pattern = vcSingleColoredNoPattern
```

TextFont

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Boxformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFText** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Boxformatfeldes

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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 **vcFFText** gesetzt wurde.

348 API-Referenz: VcBoxFormatField

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Boxformatfelds Standardwert: -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.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 Boxformatfelds erfragen.

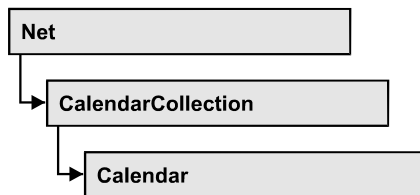
	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldTypeEnum Mögliche Werte: vcFFTGraphics 64 vcFFTText 36	Typ des Boxformatfeldes Grafik Text

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcNet1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTGraphics
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.

Eigenschaften

- CalendarProfileCollection
- IntervalCollection
- Name
- SecondsPerWorkday
- Specification

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 Intervallaufistung, 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

Code-Beispiel

```
Dim calendar As VcCalendar
Dim calendarName As String

Set calendar = VcNet1.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	Spezifikation des Kalenders

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

352 API-Referenz: VcCalendar

⇒ 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 = VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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.

	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 = VcNet1.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.

356 API-Referenz: VcCalendar

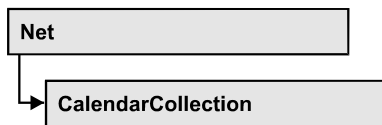
	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
Dim calendar As VcCalendar
```

```
Set calendar = VcNet1.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 VcNet1.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 = VcNet1.WorkdayCollection.CreateWorkday("Work day")
workday.AddNonWorkInterval "00:00:00", "00:00:00"
workday.AddWorkInterval "08:00:00", "16:30:00"

Set freeday = VcNet1.WorkdayCollection.CreateWorkday("Workfree day")
freeday.AddNonWorkInterval "00:00:00", "00:00:00"

Set calendarCltn = VcNet1.calendarcollection
Set calendar = calendarCltn.AddCalendar("New Calendar")

Set workweek = VcNet1.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
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	Kalenderspezifikation

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	Index des Kalenders
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	Name des Kalenders
Rückgabewert	VcCalendar	Kalender

Code-Beispiel

```
Dim calendarCltn As VcCalendarCollection

Set calendarCltn = VcNet1.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
⇒ newCalendarName	String	Name des neuen Kalenders
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 **NextCalendar** 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 = VcNet1.CalendarCollection
Set calendar = calendarCltn.FirstCalendar

While Not calendar Is Nothing
    List1.AddItem (calendar.Name)
```

362 API-Referenz: VcCalendarCollection

```
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)

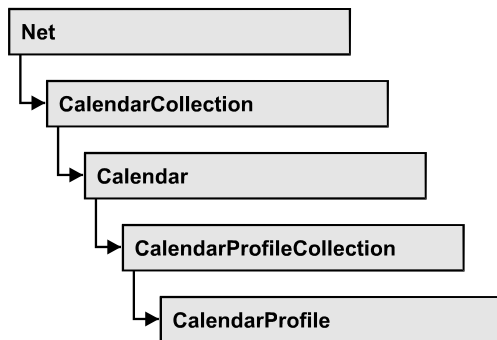
Update

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 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 Intervallaufistung, 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

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

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

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.

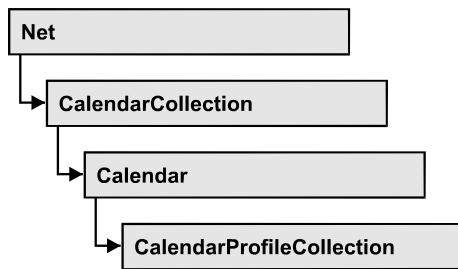
	Datentyp	Beschreibung
Parameter: refNameParam	String	Name des Kalenderprofils, hinter das das aktuelle Kalenderprofil gesetzt werden soll.

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()
```

7.14 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	Name des Kalenderprofils
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	Kalenderprofilspezifikation
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	Index des Kalenderprofils
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	Name des CalendarProfile-Objekts
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	Name des zu kopierenden Kalenderprofils
⇒ newprofileName	String	Name des neuen Kalenderprofils
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
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	Auszuwählender Kalenderprofiltyp

Rückgabewert	Long	Anzahl der ausgewählten Kalenderprofile
---------------------	------	---

Code-Beispiel

```
Dim calendarProfileCltn As VcCalendarProfileCollection
```

```
Set calendarProfileCltn = VcNet1.CalendarProfileCollection  
calendarProfileCltn.SelectCalendarProfile (vcSelected)
```

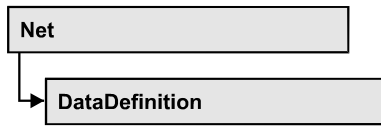
Update

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)

7.15 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 VcNet-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

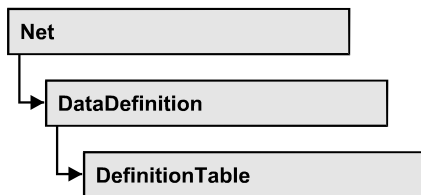
Code-Beispiel

```

Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable

Set dataDefinition = VcNet1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
  
```

7.16 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 VcNet1.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 = VcNet1.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	Name des neuen Feldes
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = _ VcNet1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.CreateDataField("Description")
dataDefinitionField.Type = vcDefFieldAlphanumericType
VcNet1.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	Index des Feldes
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField
Set dataDefinitionTable = VcNet1.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 über seinen Index

angesprochen werden. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	Beschreibung
Parameter:		
⇒ fieldName	String	Feldname
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinition = VcNet1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FieldByName("Code 1")
```

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 = VcNet1.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 = VcNet1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
While Not dataDefinitionField Is Nothing
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Wend

or

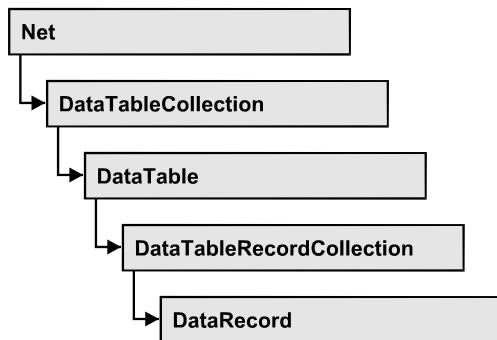
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcNet1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
    List1.AddItem dataDefinitionField.Name
    Set dataDefinitionField = dataDefinitionTable.NextField
Next

```

7.17 VcDataRecord



Ein Datensatz ist das logische Grundelement eines Objektes in einem Diagramm, z.B. eines Knotens o.ä. 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 = VcNet1.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
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord
Set dataTable = VcNet1.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 = VcNet1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecCltn.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 vcObjTypeGroup 7 vcObjTypeLinkCollection 3 vcObjTypeNode 2 vcObjTypeNodeInLegend 17 vcObjTypeNone 0	Objekttyp Box Objekttyp Gruppe Objekttyp LinkCollection Objekttyp Knoten Objekttyp Knoten im Legendbereich kein Objekt
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	Index des Datenfeldes
Rückgabewert	VcDataRecord	Zugeordneter Datensatz

Code-Beispiel

```
Private Sub VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, ByVal location As VcNetLib.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 = VcNet1.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

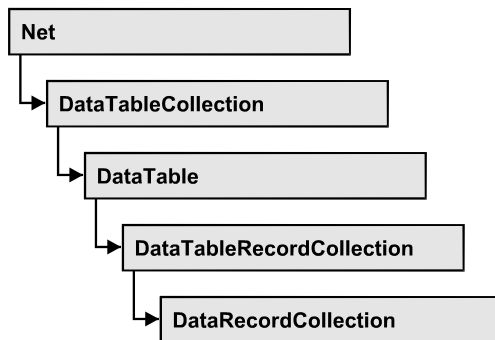
Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcNet1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

dataRecord.DataField(1) = "Node Two"
dataRecord.UpdateDataRecord
```

7.18 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 enthaltenen 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 = VcNet1.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 Data-Record-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datensätze im Collection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection

Set dataTable = VcNet1.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 DataRecord-Collection 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 **VcNet.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 = VcNet1.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)
VcNet1.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
Rückgabewert	VcDataRecord	Datensatzobjekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcNet1.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 = VcNet1.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 = VcNet1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcNet1.SuspendUpdate True

Set dataRecord = dataRecordCltn.FirstDataRecord
While Not dataRecord Is Nothing
    dataRecord.DataField(4) = "10"
    dataRecord.UpdateDataRecord
    Set dataRecord = dataRecordCltn.NextDataRecord
Wend

VcNet1.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 = VcNet1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcNet1.SuspendUpdate True

Set dataRecord = dataRecordCltn.FirstDataRecord
While Not dataRecord Is Nothing
    dataRecord.DataField(4) = "10"
    dataRecord.UpdateDataRecord
    Set dataRecord = dataRecordCltn.NextDataRecord
Wend

VcNet1.SuspendUpdate False
VcNet1.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 **VcNet.End-Loading** aufgerufen werden, damit die Änderung wirksam wird.

	Datentyp	Beschreibung
Parameter:		
⇒ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)

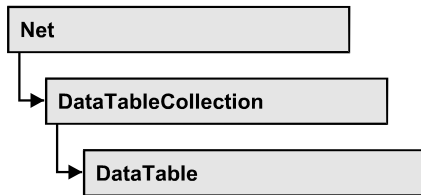
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)
---------------------	---------	---

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord

Set dataTable = VcNet1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
dataRecordCltn.Update("1;1.8.2017;;8")
VcNet1.EndLoading()
```

7.19 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 = VcNet1.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 = VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcNet1.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

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.

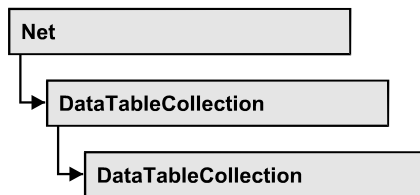
	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle Standardwert: Leere Zeichenkette

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcNet1.DataTableCollection.DataTableByIndex(0)
MsgBox dataTable.Name
```

7.20 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 XNet-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 = VcNet1.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 = VcNet1.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 VcNet 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
Rückgabewert	VcDataTable	Neu angelegte Datentabelle

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcNet1.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)
⇒ newDataTableName	String	Name der neu zu erstellenden Datentabelle (Zieltabelle)
Rückgabewert	VcDataTable	Neu erstelltes Datentabellen-Objekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcNet1.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	Index der Datentabelle
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcNet1.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	Name der Tabelle
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcNet1.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 = VcNet1.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 = VcNet1.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 vermieden.

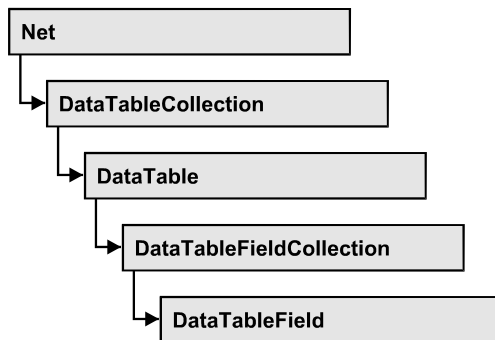
	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

dataTableCltn = VcNet1.DataTableCollection
dataTable = dataTableCltn.Add("Resources")
dataTable.DataTableFieldCollection.Add ("Id")
dataTableCltn.Update
```

7.21 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

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Editable = False
VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Hidden = True
VcNet1.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.

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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.DataTableByName("Operation")
Set dataTableField = dataTable.DataTableFieldCollection.Add("Start")
VcNet1.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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField
Dim isPrimaryKey As Boolean

Set dataTable = VcNet1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Id")
dataTableField.PrimaryKey = True
VcNet1.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
Dim dataTableOperation As VcDataTable
Dim dataOperationFieldId As VcDataTableField
Dim dataOperationFieldName As VcDataTableField
Dim dataOperationFieldTaskId As VcDataTableField

'Create table Task
dataTableTask = VcNet1.DataTableCollection.Add("Task")
dataTaskFieldId = dataTableTask.DataTableFieldCollection.Add("Id")
dataTaskFieldId.PrimaryKey = True
dataTaskFieldName = dataTableTask.DataTableFieldCollection.Add("Name")
dataTaskFieldName.Type = vcDataTableFieldAlphanumericType

'Create table Operation
dataTableOperation = VcNet1.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
```

404 API-Referenz: VcDataTableField

```
dataOperationFieldTaskId.RelationshipFieldIndex =  
VcNet1.DetectFieldIndex("Task", "Id")
```

Type

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Datentyp des Feldes setzen oder erfragen.

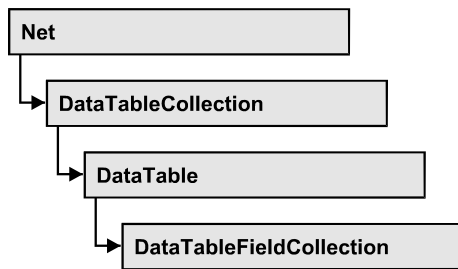
Hinweis: Durch Setzen der Eigenschaft **Type** kann sich die Eigenschaft **DateFormat** ändern. Durch Setzen des Eigenschaftswertes auf **vcDataTableAlphanumeric** oder **vcDataTableFieldInteger** wird ein eventuell eingestelltes Datumsformat auf "" gesetzt.

	Datentyp	Beschreibung
Eigenschaftswert	DataTableFieldTypeEnum	Datentyp des Feldes, kann maximal 512 Zeichen enthalten Standardwert: VcDataTableFieldIntegerType

Code-Beispiel

```
Dim dataTable As VcDataTable  
Dim dataTableField As VcDataTableField  
  
Set dataTable = VcNet1.DataTableCollection.DataTableByName("Operation")  
Set dataTableField =  
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")  
dataTableField.Type = vcDataTableFieldDateTimeType  
VcNet1.DataTableCollection.Update
```

7.22 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 = VcNet1.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 = VcNet1.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	Name des neuen Datentabellenfeldes
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Add("Priority")
VcNet1.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	Name des zu kopierenden Datentabellenfeldes (Quellfeld)
⇒ newDataTableFieldName	String	Name des neu zu erstellenden Datentabellenfeldes (Zielfeld)
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Copy("Name", "NewName")
VcNet1.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	Index des Datentabellenfeldes

Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld
---------------------	------------------	-------------------------------

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.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	Name des Datentabellenfeldes
Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcNet1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.DataTableFieldBy("Name")
dataTableField.Editable = False
VcNet1.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 = VcNet1.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**).

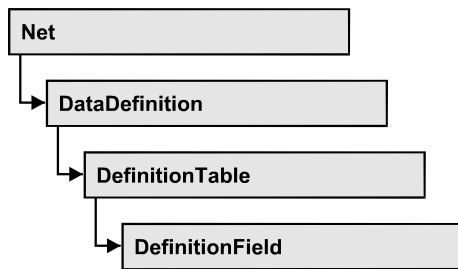
	Datentyp	Beschreibung
Rückgabewert	VcDataTableField	Nachfolgendes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableFieldCltn As VcDataTableFieldCollection
Dim dataTableField As VcDataTableField
Dim i As Integer

Set dataTable = VcNet1.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
```

7.23 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 `VcNet`-Objektes verwendet wird.

Das Format der Datumsausgabe in der Grafik wird über die Eigenschaft **DateOutputFormat** bei `VcNet` 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

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.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

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.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.

412 API-Referenz: VcDefinitionField

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Definitionsfeld versteckt/nicht versteckt Standardwert: False

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.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 Daten-
definitionstabelle abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Definitionsfeldes

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.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 Datendefi-
nitionstabelle setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Definitionsfeldes

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.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 = ""

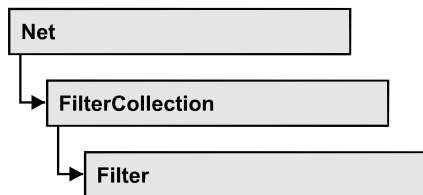
	Datentyp	Beschreibung
Eigenschaftswert	DefinitionFieldTypeEnum	Typ des Definitionsfeldes Standardwert: vcDefFieldIntegerType
	Mögliche Werte: vcDefFieldAlphanumericType 1 vcDefFieldDateTimeType 4 vcDefFieldIntegerType 2	Datentyp Alphanumerisch: "" Datentyp Datum: DD.MM.YYYY Datentyp Integer (32 Bit): ""

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcNet1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.CreateDataField("Start")
dataDefField.Type = vcDefFieldDateTimeType
```

7.24 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)

Name

Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie den Namen des Filters erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Filters

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcNet1.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

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)

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}
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

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)}
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

⇒ 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)}
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

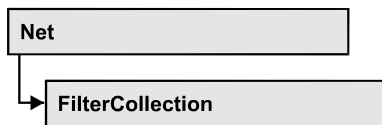
Methode von VcFilter

Mit dieser Methode können Sie eine Filterbedingung mit Hilfe der Indexangabe löschen.

420 API-Referenz: VcFilter

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index der zu löschenden Filterbedingung

7.25 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 VcNet1.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 = VcNet1.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 VcNet1.ActiveNodeFilter = VcNet1.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	Name des Filters
Rückgabewert	VcFilter	Neues Filterobjekt

Code-Beispiel

```
Set newFilter = VcNet1.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	Filterspezifikation
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	String	Name des zu kopierenden Filters
⇒ newName	String	Name des neuen Filters
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	Index des Filters
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	Name des Filters
Rückgabewert	VcFilter	Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcNet1.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 = VcNet1.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 = VcNet1.FilterCollection
Set filter = filterCltn.FirstFilter

While Not filter Is Nothing
    Listbox.AddItem filter.Name
    Set filter = filterCltn.NextFilter
Wend
```

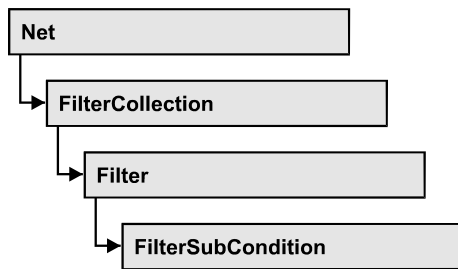
Remove

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
Rückgabewert	Boolean	Filter gelöscht (True)/nicht gelöscht (False)

7.26 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

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 Mögliche Werte: vcAnd 1 vcInvalidConnOp 0 vcOr 2	Operator für die Verknüpfung mit der folgenden Filterbedingung 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.

Sonderwert: -1: kein Datenfeld (ungültig)

	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

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

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.

430 API-Referenz: VcFilterSubCondition

	Datentyp	Beschreibung
Eigenschaftswert	OperatorEnum	Vergleichsoperator
	Mögliche Werte:	
	vcDateEarlier 27	Datum früher als
	vcDateEarlierOrEqual 28	Datum früher als oder gleich
	vcDateEqual 25	Datum gleich
	vcDateIn 31	Datum in
	vcDateLater 29	Datum später als
	vcDateLaterOrEqual 30	Datum später als oder gleich
	vcDateNotEqual 26	Datum ungleich
	vcDateNotIn 32	Datum nicht in
	vcIntEqual 9	Integer gleich
	vcIntGreater 13	Integer größer
	vcIntGreaterOrEqual 14	Integer größer oder gleich
	vcIntIn 15	Integer in
	vcIntLess 11	Integer kleiner als
	vcIntLessOrEqual 12	Integer kleiner als oder größer
	vcIntNotEqual 10	Integer ungleich
	vcIntNotIn 16	Integer nicht in
	vcInvalidOp 0	ungültiger Operator
	vcStringBeginsWith 3	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

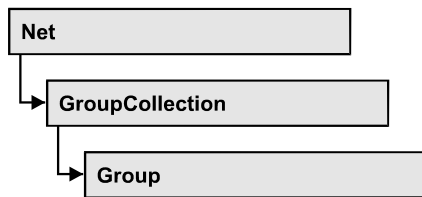
IsValid

Methode von VcFilterSubCondition

Diese Methode prüft, ob die Filterbedingung korrekt formuliert ist.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Filterbedingung korrekt (True)/ nicht korrekt (False)

7.27 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

- BackColor
- Collapsed
- LineColor
- LineThickness
- LineType
- Name
- NodeCollection
- Title
- TitleLineCount
- X
- Y

Methoden

- SetXY

Eigenschaften

BackColor

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie die Hintergrundfarbe einer Gruppe setzen oder erfragen. Die Standard-Farbe ist weiß.

432 API-Referenz: VcGroup

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcNet1.GroupCollection
Set group = groupCltn.FirstGroup

group.BackColor = RGB(128, 128, 128)
```

Collapsed

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob eine Gruppe kollabiert (True) oder expandiert (False) ist. Diese Eigenschaft kann nur beim Modus Clustering verwendet werden (GroupMode = vcGMClustering). Das Setzen kann auch interaktiv erfolgen, wenn die Eigenschaft VcNet.GroupInteractionsAllowed eingeschaltet ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Gruppe kollabiert/expandiert

Code-Beispiel

```
Private Sub VcNet1_OnGroupLClick(ByVal group As VcNetLib.VcGroup, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    If group.Collapsed = False Then
        group.Collapsed = True
    Else
        group.Collapsed = False
    End If

End Sub
```

LineColor

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie die Linienfarbe für die Randlinie einer Gruppe erfragen oder festlegen. Sie können diese Eigenschaft auch auf der Eigenschaftenseite <bI>Gruppierung festlegen.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	Color	RGB-Farbwerte {0...255},{0...255},{0...255}
Eigenschaftswert	System.Drawing.Color	RGB-Farbwerte {0...255},{0...255},{0...255}

LineThickness

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie die Stärke der Randlinie der Gruppe 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	Integer	Liniendicke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm Standardwert: Wie im Dialog definiert

LineStyle

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie den (Rand-)Linientyp einer Gruppe erfragen oder festlegen. Sie können diese Eigenschaft auch im auf der Eigenschaftenseite **Gruppierung** festlegen.

Eigenschaftswert	Datentyp	Beschreibung
	LineStyleEnum	Linientyp
	Mögliche Werte:	
	vcDashed 4	Linientyp gestrichelt
	vcDashedDotted 5	Linientyp gestrichelt-gepunktet
	vcDotted 3	Linientyp gepunktet
	vcLineStyle0 100	Linientyp 0 _____
	vcLineStyle1 101	Linientyp 1 _ _ _ _ _
	vcLineStyle10 110	Linientyp 10 _
	vcLineStyle11 111	Linientyp 11 _
	vcLineStyle12 112	Linientyp 12 _
	vcLineStyle13 113	Linientyp 13 _
	vcLineStyle14 114	Linientyp 14 _
	vcLineStyle15 115	Linientyp 15 _
	vcLineStyle16 116	Linientyp 16 _
	vcLineStyle17 117	Linientyp 17 _
	vcLineStyle18 118	Linientyp 18 _
	vcLineStyle2 102	Linientyp 2 _
	vcLineStyle3 103	Linientyp 3 _
	vcLineStyle4 104	Linientyp 4 _
	vcLineStyle5 105	Linientyp 5 _
	vcLineStyle6 106	Linientyp 6 _
	vcLineStyle7 107	Linientyp 7 _
	vcLineStyle8 108	Linientyp 8 _
	vcLineStyle9 109	Linientyp 9 _
	vcNone 1	Kein Linientyp
	vcNotSet -1	Kein Linientyp zugewiesen
	vcSolid 2	Linientyp durchgezogen

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

Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup
Dim groupName As String

Set groupCltn = VcNet1.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 groupCollection As VcGroupCollection
Dim group As VcGroup
Dim nodeCollection As VcNodeCollection
Dim numberOfNodes As Integer

Set groupCollection = VcNet1.GroupCollection
Set group = groupCollection.FirstGroup
Set nodeCollection = group.NodeCollection

nodeCollection.SelectNodes (vcAll)
numberOfNodes = nodeCollection.Count

Dim groupCollection As VcGroupCollection
Dim group As VcGroup
Dim nodeCollection As VcNodeCollection

Dim name As String
Dim number As Long

Set groupCollection = VcNet1.GroupCollection
Set group = groupCollection.FirstGroup
Set nodeCollection = group.NodeCollection

name = group.Name
```

436 API-Referenz: VcGroup

```
number = nodeCollection.Count
```

Title

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie den Gruppentitel erfragen oder festlegen. Der Titel wird in der obersten Zeile der Gruppe ausgegeben. Wenn Sie diese Eigenschaft nicht festlegen, keinen Dateinamen bei VcNet.GroupDescriptionName angeben und kein Feld bei VcNet.GroupTitleField angeben, so wird in der obersten Zeile der Gruppenname ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Gruppentitel

Code-Beispiel

```
Dim groupCollection As VcGroupCollection
Dim group As VcGroup
Dim nodeCollection As VcNodeCollection
Dim groupTitle As String

Set groupCollection = VcNet1.GroupCollection
Set group = groupCollection.FirstGroup

groupTitle = group.Title
```

TitleLineCount

Eigenschaft von VcGroup

Mit dieser Eigenschaft können Sie bei der aktuellen Gruppe die Anzahl der Zeilen für den Titeltext einstellen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer 1 ... 5	Anzahl der Zeilen für den Titeltext Standardwert: 1

Code-Beispiel

```
Dim groupCollection As VcGroupCollection
Dim group As VcGroup
Dim nodeCollection As VcNodeCollection

Set groupCollection = VcNet1.GroupCollection
Set group = groupCollection.FirstGroup
group.TitleLineCount = 5
```

X**Nur-Lese-Eigenschaft von VcGroup**

Diese Eigenschaft gibt die aktuelle x-Position der Gruppe in Bandnummern an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	X-Koordinate

Y**Nur-Lese-Eigenschaft von VcGroup**

Diese Eigenschaft gibt die aktuelle y-Position der Gruppe in Bandnummern an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Y-Koordinate

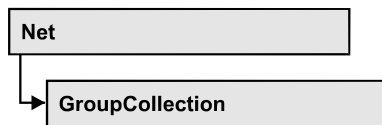
Methoden

SetXY**Methode von VcGroup**

Mit dieser Methode können Sie die Position der Gruppe setzen. Diese Methode kann nur im Visualisierungsmodus Clusterung (GroupMode = vcGMClustering) angewendet werden, und nur wenn die Gruppe kollabiert ist oder wenn diese Methode im **OnGroupCreate**-Ereignis aufgerufen wird.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate (in Bandnummern)
⇒ y	Long	Y-Koordinate (in Bandnummern)
Rückgabewert	Boolean	Koordinaten erfolgreich gesetzt (True)/nicht erfolgreich gesetzt (False)

7.28 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`

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 VcNet1.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 = VcNet1.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 = VcNet1.GroupCollection
Set group = groupCltn.FirstGroup
```


GroupByName

Methode von VcGroupCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Gruppe zugreifen. Zuvor muss mit der Methode **SelectGroups** die gewünschte Gruppe ausgewählt worden sein. Existiert keine Gruppe unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇒ Rückgabewert	VcGroup	Gruppe
⇒ groupName	String	Gruppenbezeichnung
Rückgabewert	VcGroup	Gruppe

Code-Beispiel

```
Dim groupCollection As VcGroupCollection
Dim group As VcGroup

Set groupCollection = VcNet1.GroupCollection
Set group = groupCollection.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

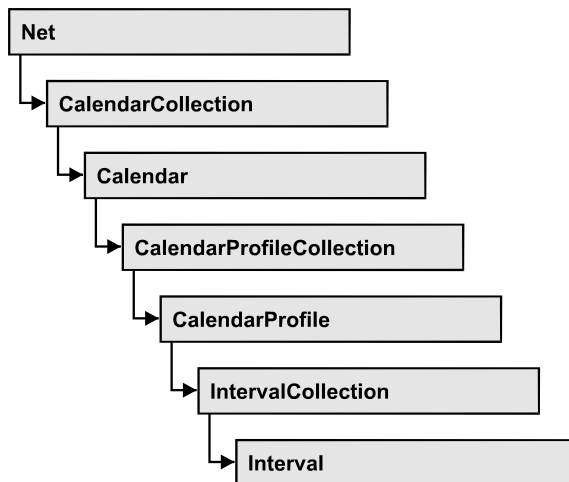
Code-Beispiel

```
Dim groupCltn As VcGroupCollection
Dim group As VcGroup

Set groupCltn = VcNet1.GroupCollection
Set group = groupCltn.FirstGroup

While Not group Is Nothing
    List1.AddItem group.Name
    Set group = groupCltn.NextGroup
Wend
```

7.29 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

- CalendarProfileName
- DayInEndMonth
- DayInStartMonth
- EndDateTime
- EndMonth
- EndTime
- EndWeekday
- Name
- Specification
- StartDateTime
- StartMonth
- StartTime
- StartWeekday
- Type

Methoden

- PutInOrderAfter

Eigenschaften

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

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

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

EndDateTime

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	Endmonat 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

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 Mögliche Werte: vcFriday 5 vcMonday 1 vcSaturday 6 vcSunday 7 vcThursday 4 vcTuesday 2 vcWednesday 3	Letzter Wochentag des Intervalls Wochentag Freitag Wochentag Montag Wochentag Samstag Wochentag Sonntag Wochentag Donnerstag Wochentag Dienstag Wochentag Mittwoch

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

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

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

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	Typ des Intervalls

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.

448 API-Referenz: VcInterval

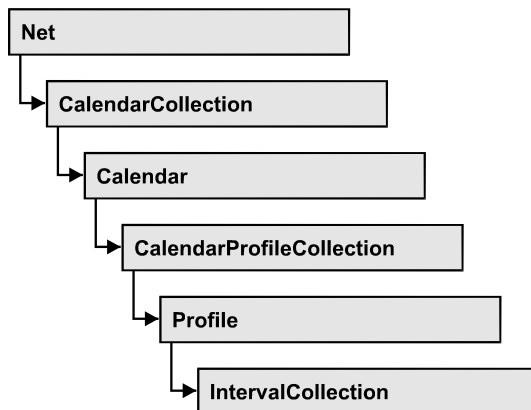
	Datentyp	Beschreibung
Parameter: refNameParam	String	Name des Intervalls, hinter das das aktuelle Intervall gesetzt werden soll.
Rückgabewert	Void	

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()
```

7.30 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	Name des Intervalls
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	Intervallspezifikation
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	Name des zu kopierenden Intervalls
⇒ newIntervalName	String	Name des neuen Intervalls
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	Index des Intervalls
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	Name des Intervallobjekts
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	Name des Intervalls
Rückgabewert	Boolean	Intervall gelöscht (True)/nicht gelöscht (False)

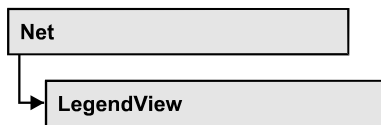
Update

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)

7.31 VcLegendView



Ein Objekt vom Typ **VcLegendView** bezeichnet das Legendenansicht-Fenster.

Eigenschaften

- Border
- 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

Code-Beispiel

```
VcNet1.LegendView.Mode = vcNotFixed
VcNet1.LegendView.Border = True
```

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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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 (VcNet1.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	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig.
	vcHorizontalScrollBar 1	Anzeige einer horizontalen Bildlaufleiste, wenn nötig.
	vcNoScrollBar 0	Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt.

458 API-Referenz: VcLegendView

vcVerticalScrollBar 2

Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

Code-Beispiel

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

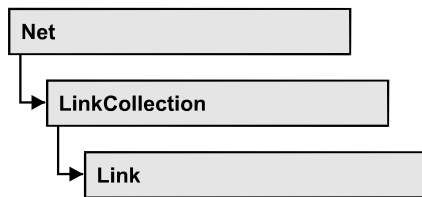
```
VcNet1.LegendView.WindowMode = vcNotFixed
```

Methoden**Update****Methode von VcLegendView**

Mit dieser Methode wird die Legende aktualisiert.

	Datentyp	Beschreibung

7.32 VcLink



Ein Verbindungsobjekt stellt die logische und grafische Verbindung zwischen zwei Knoten dar. Das Aussehen wird durch diejenigen LinkAppearance-Objekte bestimmt, deren Filter auf die Verbindungsdaten zutreffen. Sie können Verbindungen entweder interaktiv oder über die Methode **InsertLinkRecord** des **VcNet**-Objektes erzeugen.

Eigenschaften

- AllData
- DataField
- ID
- MarkLink
- 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 = VcNet1.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	Index des Datenfeldes
Eigenschaftswert	Variant	Inhalt des Datenfelds

Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink
Dim message As String

Set linkCltn = VcNet1.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

MarkLink

Eigenschaft von VcLink

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob diese Verbindung markiert ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verbindung markiert (True)/nicht markiert (False)

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 = VcNet1.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 = VcNet1.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 VcNet1_OnLinkRClick(ByVal link As VcNetLib.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	Index des Datenfeldes, das den Schlüssel enthält
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

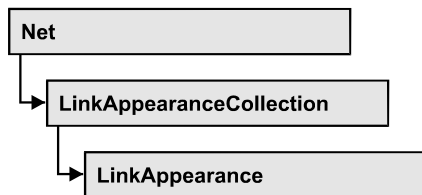
Code-Beispiel

```
Dim linkCltn As VcLinkCollection
Dim link As VcLink

Set linkCltn = VcNet1.LinkCollection
Set link = linkCltn.FirstLink

link.DataField(2) = "10"
link.UpdateLink
```

7.33 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 **Aussehen**-Tabelle voreingestellt werden.

Eigenschaften

- FilterName
- FormatName
- LineColor
- LineThickness
- LineType
- Name
- PrePortSymbol
- RoutingType
- Specification
- 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.

468 API-Referenz: VcLinkAppearance

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Filterobjekt

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim filterOfLinkApp As String

Set linkAppearanceCltn = VcNet1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.LinkAppearanceByName("Blue")
filterOfLinkApp = linkAppearance.Filter
```

FormatName

Eigenschaft von VcLinkAppearance

Mit dieser Eigenschaft können Sie das Format des LinkAppearance-Objekts setzen oder erfragen. Ist keine Eigenschaft gesetzt, wird die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren LinkAppearance-Objektes übernommen, dessen Filterbedingungen für die Verbindung ebenfalls zutreffen und dessen Eigenschaftswert nicht leer ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Verbindungsformat-Objekts oder leere Zeichenkette

Code-Beispiel

```
Dim linkAppearanceCollection As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim format1 As VcLinkFormat

Set linkAppearanceCollection = VcNet1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCollection.FirstLinkAppearance

Set format1 = linkAppearance.format
MsgBox (format1.name)
```

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 = VcNet1.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.

470 API-Referenz: VcLinkAppearance

	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 = VcNet1.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	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	Linientyp gestrichelt Linientyp gestrichelt-gepunktet Linientyp gepunktet Linientyp 0 <hr/> Linientyp 1 <hr style="border-top: 1px dashed black;"/> Linientyp 10 <hr style="border-top: 1px dotted black;"/> Linientyp 11 <hr style="border-top: 1px dash-dot black;"/> Linientyp 12 <hr style="border-top: 1px solid black;"/> Linientyp 13 <hr style="border-top: 1px solid black;"/> Linientyp 14 <hr style="border-top: 1px solid black;"/> Linientyp 15 <hr style="border-top: 1px solid black;"/> Linientyp 16 <hr style="border-top: 1px dashed black;"/> Linientyp 17 <hr style="border-top: 1px dashed black;"/>

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 = VcNet1.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

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcNet1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance









nameLinkApp = linkAppearance.name
```


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	Vorgänger-Portsymbol Keines
	vcLPSSemiCircle 96	Vorgänger-Portsymbol Halbkreis 

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcNet1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCltn.FirstLinkAppearance

linkAppearance.PrePortSymbol = vcLPSFilledDoubleSemiCircle
```

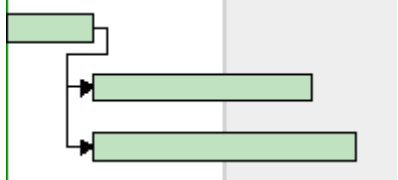
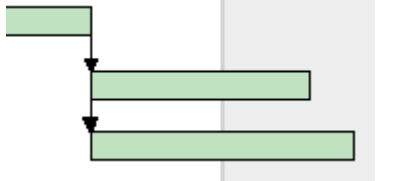
RoutingType

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 waagrecht und senkrecht und zeigen eine orthogonale Form. 
	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 Verbindungsaussehens mit der Methode **VcLinkAppearanceCollection.Add-BySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Verbindungsaussehens

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 = VcNet1.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

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance
Dim nameLinkApp As String

Set linkAppearanceCltn = VcNet1.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.

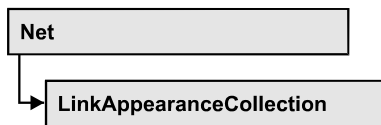
	Datentyp	Beschreibung
Parameter: refLinkAppearanceName	String	Name des Verbindungsausssehens, hinter das das aktuelle Verbindungsausssehen gesetzt werden soll.
Rückgabewert	Void	

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()
```

7.34 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 Verbindungsaussichten zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **LinkAppearanceByName** und **LinkAppearanceByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Verbindungsaussichten 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 VcNet1.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 = VcNet1.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.

478 API-Referenz: VcLinkAppearanceCollection

	Datentyp	Beschreibung
Parameter: ⇒ newName	String	Name des LinkAppearance-Objekts
Rückgabewert	VcLinkAppearance	Neues LinkAppearance-Objekt

Code-Beispiel

```
Set newLinkAppearance = VcNet1.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	Verbindungsaussehen-Spezifikation
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
⇒ newName	String	Name des neuen Verbindungsaussehens
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 = VcNet1.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

480 API-Referenz: VcLinkAppearanceCollection

Rückgabewert	VcLinkAppearance	LinkAppearance-Objekt
---------------------	------------------	-----------------------

Code-Beispiel

```
Dim linkAppearanceCltn As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCltn = VcNet1.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 = VcNet1.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	Name des Verbindungsaussehens
Rückgabewert	Boolean	Verbindungsaussehen gelöscht (True)/nicht gelöscht (False)

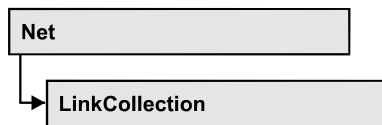
Update

Methode von VcLinkAppearanceCollection

Mit dieser Methode können Sie eine Auflistung von Verbindungsaussichten aktualisieren, nachdem Sie sie verändert haben.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Aktualisierung erfolgt (True) / nicht erfolgt (False)

7.35 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 VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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.

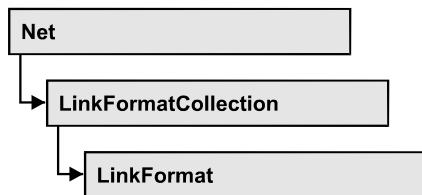
	Datentyp	Beschreibung
Parameter: ⇒ selectionType	SelectionTypeEnum Mögliche Werte: vcAll 0 vcAllLinksCausingCycles 7 vcAllLinksInCycles 6 vcAllVisible 1 vcMarked 2	Auszuwählende Verbindungen Alle Objekte im Diagramm werden ausgewählt. Wird diese Selektion gewählt, dann befinden sich in der LinkCollection 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

Code-Beispiel

```
Dim linkCollection As VcLinkCollection

Set linkCollection = VcNet1.LinkCollection
linkCollection.SelectGroups (vcAllMarked)
```

7.36 VcLinkFormat



Ein Objekt vom Typ VcLinkFormat legt Inhalt und Erscheinungsbild einer Verbindung fest. Verbindungsformate werden zur Designzeit im Dialogfeld **Verbindungsformate verwalten**, das Sie über die Eigenschaftenseite **Verbindungen** erreichen, verwaltet und bearbeitet.

Eigenschaften

- _NewEnum
- FormatField
- FormatFieldCount
- Name
- Specification

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcLinkFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Verbindungsformatfeld-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 VcLinkFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

FormatField**Nur-Lese-Eigenschaft von VcLinkFormat**

Mit dieser Eigenschaft können Sie ein VcLinkFormatField-Objekt per Index erfragen. 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	Index des Verbindungsformatfeldes 0FormatFieldCount-1
Eigenschaftswert	VcLinkFormatField	Verbidnugsformatfeld

FormatFieldCount**Nur-Lese-Eigenschaft von VcLinkFormat**

Mit dieser Eigenschaft können Sie die Anzahl der Felder eines Verbindungsformats ermitteln.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Felder im Verbindungsformat

Code-Beispiel

```
Dim formatCollection As VcLinkFormatCollection
Dim format As VcLinkFormat
Dim nameofFormat As String

Set formatCollection = VcNet1.LinkFormatCollection
Set format = formatCollection.FormatByName("Standard")

numberOfFormatField = format.FormatFieldCount
```


Name

Eigenschaft von VcLinkFormat

Mit dieser Eigenschaft können Sie den Namen des Verbindungsformats erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Verbindungsformats

Code-Beispiel

```
Dim format As VcLinkFormat
Dim formatName As String

Set format = VcNet1.LinkFormatCollection.FirstFormat
formatName = format.Name
```

Specification

Nur-Lese-Eigenschaft von VcLinkFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Verbindungsformats 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 Verbindungsformats mit der Methode **VcLinkFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Verbindungsformats

Methoden

CopyFormatField

Methode von VcLinkFormat

Mit dieser Methode können Sie ein Verbindungsformatfeld kopieren. Das neue VcLinkFormatField-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 Verbindungsformatfeldes oberhalb unterhalb links von außerhalb, oberhalb außerhalb, unterhalb außerhalb, links von außerhalb, rechts von rechts von
⇒ refIndex	Integer	Index des Referenz-Verbindungsformatfeldes
Rückgabewert	VcLinkFormatField	Verbindungsformatfeld-Objekt

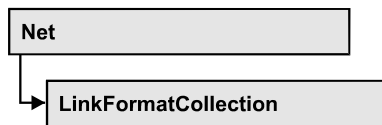
RemoveFormatField

Methode von VcLinkFormat

Mit dieser Methode können Sie ein Verbindungsformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Verbindungsformatfelder neu festgesetzt, so dass sie wieder fortlaufend numeriert sind.

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des zu löschenden Verbindungsformatfeldes

7.37 VcLinkFormatCollection



In einem Objekt vom Typ `VcLinkFormatCollection` sind automatisch alle verfügbaren Verbindungsformate zusammengefasst. Über **For Each linkFormat In LinkFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Verbindungsformate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaft **FormatByName**. Die Anzahl der im Auflistungsobjekt vorhandenen Verbindungsformate kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstFormat`
- `FormatByIndex`
- `FormatByName`
- `NextFormat`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcLinkFormatCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Verbindungsformat-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 VcLinkFormat
For Each format In VcNet1.LinkFormatCollection
    Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcLinkFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Verbindungsformatobjekte in der NodeFormat-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Verbindungsformate

Code-Beispiel

```
Dim formatCltn As VcLinkFormatCollection
Dim numberOfFormats As Long

Set formatCltn = VcNet1.LinkFormatCollection
numberOfFormats = formatCltn.Count
```

Methoden

Add

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie ein neues Verbindungsformat in der Verbindungsformat-Auflistung anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcLinkFormat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

Das Verbindungsformat besitzt standardmäßig folgende Eigenschaften:

- ein einziges Feld

- WidthOfExteriorSurrounding: 3 mm

Das Feld hat folgende Eigenschaften:

- Type: vcFFTText
- TextDataFieldIndex: in der Eigenschaftenseite **Allgemeines** festgelegte IDMinimumWidth: 3000
- Alignment: vcFFACenter
- BackColor: -1 (transparent)
- TextFontColor: RGB(0,0,0) (schwarz)
- TextFont: Arial, 10, normal
- LeftMargin, RightMargin, TopMargin, BottomMargin: 0,3 mm
- MinimumTextLineCount, MaximumTextLineCount: 1

	Datentyp	Beschreibung
Parameter: ⇒ newName	String	Name des Verbindungsformats
Rückgabewert	VcLinkFormat	Verbindungsformat-Objekt

Code-Beispiel

```
Set newLinkFormat = VcNet1.LinkFormatCollection.Add("linkformat1")
```

AddBySpecification

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie ein Verbindungsformat über eine Verbindungsformat-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Verbindungsformat-Objekten. Die Spezifikation eines Verbindungsformats kann erfragt (siehe VcLinkFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Verbindungsformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ formatSpecification	String	Verbindungsformat-Spezifikation
Rückgabewert	VcLinkFormat	Neues Verbindungsformat-Objekt

Copy

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie ein Verbindungsformat kopieren. Wenn das Verbindungsformat mit dem angegebenen Namen existiert und der Name des neuen Verbindungsformats noch nicht verwendet wird, wird das neue Verbindungsformat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ fromName	String	Name des zu kopierenden Verbindungsformats
⇒ newName	String	Name des neuen Verbindungsformats
Rückgabewert	VcLinkFormat	Verbindungsformat-Objekt

FirstFormat

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie auf das erste Verbindungsformat der LinkFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Verbindungsformate zu iterieren. Existiert kein Verbindungsformat in der LinkFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcLinkFormat	Erstes Verbindungsformat

Code-Beispiel

```
Dim format As VcLinkFormat
Set format = VcNet1.LinkFormatCollection.FirstFormat
```

FormatByIndex

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie auf ein einzelnes Verbindungsformat über ihren Index zugreifen. Existiert kein Verbindungsformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	Integer	Index des Verbindungsformats

Code-Beispiel

```
Dim linkFormatCltn As VcLinkFormatCollection

Set linkFormatCltn = VcNet1.LinkFormatCollection
Set linkFormat = linkFormatCltn.LinkFormatByIndex(2)
linkFormat.WidthOfExteriorSurrounding = 2
```

FormatByName

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Verbindungsformat zugreifen. Existiert kein Verbindungsformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ formatName	String	Name des Verbindungsformats
Rückgabewert	VcLinkFormat	Verbindungsformat

Code-Beispiel

```
Dim formatCollection As VcLinkFormatCollection
Dim format As VcLinkFormat

Set formatCollection = VcNet1.LinkFormatCollection
Set format = formatCollection.FormatByName("Standard")
```

NextFormat

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Verbindungsformate der Verbindungsformat-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	VLinkFormat	Nachfolgendes Verbindungsformat

Code-Beispiel

```
Dim formatCollection As VcLinkFormatCollection
Dim format As VcLinkFormat

Set formatCollection = VcNet1.LinkFormatCollection
Set format = formatCollection.FirstFormat

While Not format Is Nothing
    List1.AddItem format.Name
    Set format = formatCollection.NextFormat
Wend
```

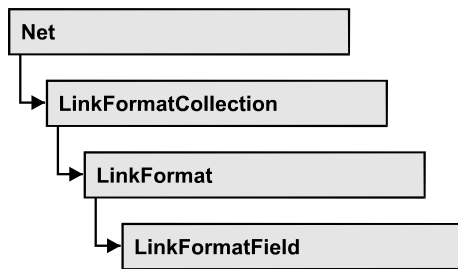
Remove

Methode von VcLinkFormatCollection

Mit dieser Methode können Sie ein Knotenformat löschen. Wenn das Verbindungsformat noch irgendwo verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ name	String	Name des Verbindungsformats
Rückgabewert	Boolean	Verbindungsformat gelöscht (True)/nicht gelöscht (False)

7.38 VcLinkFormatField



Ein Objekt vom Typ **VcLinkFormatField** stellt ein Verbindungsformatfeld, also ein Feld eines VcLinkFormat-Objekts dar. Ein Verbindungsformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Linkformat untergebracht ist.

Eigenschaften

- Alignment
- ConstantText
- FormatName
- Index
- MinimumWidth
- TextDataFieldIndex
- TextFont
- TextFontColor
- TextLineCount

Eigenschaften

Alignment

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Linkformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum	Ausrichtung des Feldinhalts
	Mögliche Werte: vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25	unten unten links unten rechts unten mittig

vcFFALeft 24	links
vcFFARight 26	rechts
vcFFATop 22	oben
vcFFATopLeft 21	oben links
vcFFATopRight 23	oben rechts

ConstantText

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Linkformatfeld ausgeben, falls der Typ des Linkformatfeldes auf **vcFFTText** und falls die Eigenschaft **TextDataFieldIndex** auf **-1** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text

FormatName

Nur-Lese-Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie den Namen des Linkformats erfragen, zu dem dieses Linkformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Linienformat-Objektes

Index

Nur-Lese-Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie den Index des Linkformatfelds im zugehörigen Linkformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Tabellenformatfeldes

MinimumWidth

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Linkformatfeldes 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 Layerformatfeldes in mm 0 ... 99

TextDataFieldIndex

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTText** den Index des Datenfelds, dessen Inhalt in dem Linkformatfeld dargestellt werden soll, erfragen oder setzen. Wenn Sie den Index auf -1 setzen, wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes

TextFont

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Linkformatfeldes 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 VcLinkFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Linkformatfeldes 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

TextLineCount

Eigenschaft von VcLinkFormatField

Mit dieser Eigenschaft können Sie die Zeilenanzahl erfragen oder setzen, wenn die Größe des Beschriftungsfeldes mehrere zulässt.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeilenzahl

7.39 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 Zuordnungstabelleobjekte 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 VcNet1.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 = VcNet1.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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapName As String

Set mapCltn = VcNet1.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

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	Typ der Zuordnungstabelle Beliebig (nur zum Selektieren verwendet) Farben Schriften

vcGraphicsFileMap	7	Grafikdatei
vcMillimeterMap	9	Millimeter
vcNumberMap	10	Zahlen
vcPatternMap	3	Schraffuren
vcTextMap	6	Text

Code-Beispiel

```
Dim mapCollection As VcMapCollection
Dim map As VcMap

Set mapCollection = VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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

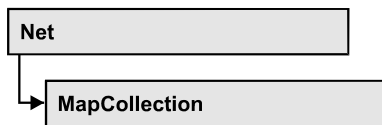
```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcNet1.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
```

7.40 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 InMapCollection** 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 VcNet1.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 = VcNet1.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	Name der Zuordnungstabelle
Rückgabewert	VcMap	Neues Zuordnungstabellenobjekt

Code-Beispiel

```
Set newMap = VcNet1.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	Zuordnungstabellenspezifikation
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
⇒ newMapName	String	Name der neuen Zuordnungstabelle
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 = VcNet1.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	Index der Zuordnungstabelle
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	Name der Map
Rückgabewert	VcMap	Map

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcNet1.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 = VcNet1.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
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	Auszuwählender Map-Typ Beliebig (nur zum Selektieren verwendet) Farben Schriften Grafikdatei Millimeter Zahlen

512 API-Referenz: VcMapCollection

	vcPatternMap 3 vcTextMap 6	Schraffuren Text
Rückgabewert	Long	Anzahl der Ausgewählten Maps

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcNet1.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

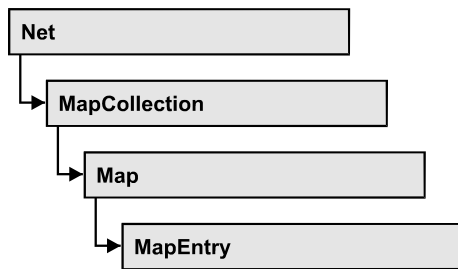
Set mapCltn = VcNet1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

While Not mapEntry.DataFieldValue = "A"
    Set mapEntry = map.NextMapEntry
Wend

mapEntry.Color = RGB(0, 0, 0)

mapCltn.Update
```

7.41 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
- 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 = VcNet1.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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim dataFieldValue As String

Set mapCltn = VcNet1.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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontBodyOfMapEntry As FontBodyEnum

Set mapCltn = VcNet1.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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontNameOfMapEntry As String

Set mapCltn = VcNet1.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
```

516 API-Referenz: VcMapEntry

```
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontSizeOfMapEntry As Long

Set mapCltn = VcNet1.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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry




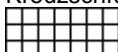

Set mapCltn = VcNet1.MapCollection
mapCltn.SelectMaps (vcGraphicsFileMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

mapEntry.GraphicsFileName = AppPath & "\picture1.bmp"
```

Pattern

Eigenschaft von VcMapEntry

für *Schraffuren-Zuordnungstabellen (vcPatternMap)*: Mit dieser Eigenschaft können Sie den Schraffurtyp des Zuordnungstabelleneintrags erfragen oder festlegen.

	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
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien
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
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

520 API-Referenz: VcMapEntry

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 mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim pattern As FillPatternEnum

Set mapCltn = VcNet1.mapCollection
mapCltn.SelectMaps (vcPatternMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

pattern = vcBDiagonalPattern
```

7.42 VcNet

Net

Ein Objekt vom Typ VcNet bezeichnet das VARCHART-XNet-Steuer-element selbst. Über Ereignisse können Sie dessen Interaktionen kontrollieren. Das VcNet-Objekt kann durch eine Vielzahl von Eigenschaften und Methoden den Anforderungen entsprechend konfiguriert werden.

Eigenschaften

- ActiveNodeFilter
- AllowMultipleBoxMarking
- AllowNewNodesAndLinks
- AssignCalendarToNodes
- BorderArea
- BoxCollection
- BoxFormatCollection
- CalendarCollection
- CalendarProfileCollection
- ConfigurationName
- CtrlCXVProcessing
- CurrentVersion
- DataDefinition
- DataTableCollection
- DateOutputFormat
- DiagramBackColor
- DialogFont
- DoubleOutputFormat
- EditNewLink
- EditNewNode
- Enabled
- EnableSupplyTextEntryEvent
- EventReturnStatus
- EventText
- ExtendedDataTables
- FilePath
- FilterCollection
- FontAntiAliasingEnabled
- GroupCollection
- GroupDescriptionName

- GroupField
- GroupHorizontalMargin
- Grouping
- GroupingTitlesFullyVisible
- GroupInteractionsAllowed
- GroupMode
- GroupMovingAllowed
- GroupSortField
- GroupSortMode
- GroupTitleField
- GroupVerticalMargin
- hWnd
- InFlowGroupDescriptionName
- InFlowGroupField
- InFlowGroupingEnabled
- InFlowGroupSeparationLineColor
- InFlowGroupSeparationLineType
- InFlowGroupTimeInterval
- InFlowGroupTitleField
- InFlowGroupTitlesBackColor
- InFlowGroupTitlesFont
- InFlowGroupTitlesVisibleAtBottomOrRight
- InFlowGroupTitlesVisibleAtTopOrLeft
- InFlowGroupTitleTimeFormat
- InFlowGroupVerticalCaptionWidth
- InPlaceEditingAllowed
- InteractionMode
- InterfaceNodesShown
- LegendView
- LinkAnnotationColumnNumberDataFieldIndex
- LinkAnnotationRowNumberDataFieldIndex
- LinkAppearanceCollection
- LinkCollection
- LinkFormatCollection
- LinkPredecessorDataFieldIndex
- LinksDataTableName
- LinkSuccessorDataFieldIndex
- LinkTypeDataFieldIndex
- MapCollection
- MinimumColumnWidth

- MinimumRowHeight
- MouseProcessingEnabled
- NodeAppearanceCollection
- NodeCalendarNameDataFieldIndex
- NodeChangeRankToPredecessorRankDataFieldIndex
- NodeCollection
- NodeColumnNumberDataFieldIndex
- NodeFormatCollection
- NodeRowNumberDataFieldIndex
- NodesDataTableName
- NodeTooltipTextField
- ObliqueTracksOnLinks
- OLEDragMode
- OLEDragWithOwnMouseCursor
- OLEDragWithPhantom
- OLEDropMode
- Orientation
- Printer
- RoundedLinkSlantsEnabled
- Scheduler
- ScrollOffsetX
- ScrollOffsetY
- ShortenedLinks
- ShowToolTip
- StraightLinkDrawing
- TimeUnit
- ToolTipChangeDuration
- ToolTipDuration
- ToolTipPointerDuration
- ToolTipShowAfterClick
- UngroupedNodesAllowed
- WaitCursorEnabled
- WorldView
- ZoomFactor
- ZoomingPerMouseWheelAllowed

Methoden

- AboutBox
- Arrange
- Clear

- CopyNodesIntoClipboard
- CutNodesIntoClipboard
- DeleteLinkRecord
- DeleteNodeRecord
- DetectDataTableFieldName
- DetectDataTableName
- DetectFieldIndex
- DumpConfiguration
- EditLink
- EditNode
- EndLoading
- ExportGraphicsToFile
- GetAValueFromARGB
- GetBValueFromARGB
- GetGValueFromARGB
- GetLinkByID
- GetLinkByIDs
- GetNodeByID
- GetRValueFromARGB
- IdentifyFormatField
- IdentifyFormatFieldAsVariant
- IdentifyObjectAt
- IdentifyObjectAtAsVariant
- InsertLinkRecord
- InsertNodeRecord
- MakeARGB
- Open
- PageLayout
- PasteNodesFromClipboard
- PixelsToRaster
- PixelsToRasterAsVariant
- PrintDirectEx
- PrinterSetup
- PrintIt
- PrintPreview
- PrintToFile
- RasterToPixels
- RasterToPixelsAsVariant
- Reset
- SaveAsEx

- ScheduleProject
- ScrollToNodePosition
- ShowAlwaysCompleteView
- ShowExportGraphicsDialog
- SuspendUpdate
- UpdateLinkRecord
- UpdateNodeRecord
- Zoom
- ZoomOnMarkedNodes

Ereignisse

- Error
- ErrorAsVariant
- KeyDown
- KeyPress
- KeyUp
- OLECompleteDrag
- OLEDragDrop
- OLEDragOver
- OLEGiveFeedback
- OLESetData
- OLEStartDrag
- OnBoxLClick
- OnBoxLDbClick
- OnBoxModifyComplete
- OnBoxModifyCompleteEx
- OnBoxRClick
- OnDataRecordCreate
- OnDataRecordCreateComplete
- OnDataRecordDelete
- OnDataRecordDeleteComplete
- OnDataRecordModify
- OnDataRecordModifyComplete
- OnDataRecordNotFound
- OnDiagramLClick
- OnDiagramLDbClick
- OnDiagramRClick
- OnGiveFeedbackForNodeCreating
- OnGroupCreate
- OnGroupDelete

- OnGroupLClick
- OnGroupLDbClick
- OnGroupModify
- OnGroupModifyComplete
- OnGroupRClick
- OnHelpRequested
- OnLegendViewClosed
- OnLinkCreate
- OnLinkCreateComplete
- OnLinkDelete
- OnLinkDeleteComplete
- OnLinkLClickCltn
- OnLinkLDbClickCltn
- OnLinkModifyComplete
- OnLinkModifyEx
- OnLinkRClickCltn
- OnLinksMark
- OnLinksMarkComplete
- OnModifyComplete
- OnMouseDown
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnNodeCreate
- OnNodeCreateCompleteEx
- OnNodeDelete
- OnNodeDeleteCompleteEx
- OnNodeLClick
- OnNodeLDbClick
- OnNodeModifyComplete
- OnNodeModifyCompleteEx
- OnNodeModifyEx
- OnNodeRClick
- OnNodesMarkComplete
- OnNodesMarkEx
- OnSelectField
- OnShowInPlaceEditor
- OnStatusLineText
- OnSupplyTextEntry
- OnSupplyTextEntryAsVariant

- OnToolTipText
- OnToolTipTextAsVariant
- OnWorldViewClosed
- OnZoomFactorModifyComplete

Eigenschaften

ActiveNodeFilter

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Filter festlegen oder erfragen, der die Menge der darzustellenden Knoten bestimmt.

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Filterobjekt Standardwert: Nothing

Code-Beispiel

```
Dim filter As VcFilter
Dim filterName As String

Set filter = VcNet1.ActiveNodeFilter

If Not filter Is Nothing Then
    filterName = filter.Name
End If

Set VcNet1.ActiveNodeFilter = VcNet1.FilterCollection. _
    FilterByName("Filter_1")
```

AllowMultipleBoxMarking

Eigenschaft von VcNet

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

Code-Beispiel

```
VcNet1.AllowMultipleBoxMarking = True
```

AllowNewNodesAndLinks**Eigenschaft von VcNet**

Mit dieser Eigenschaft wird dem Anwender das Anlegen neuer Knoten und Verbindungen erlaubt (True) oder gesperrt (False). Wird diese Eigenschaft auf False gesetzt, kann der Erzeugemodus nicht eingeschaltet werden.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/nicht in Kraft (False) Standardwert: True

Code-Beispiel

```
Dim boole As Boolean
```

```
boole = VcNet1.AllowNewNodesAndLinks
```

AssignCalendarToNodes**Eigenschaft von VcNet**

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

Code-Beispiel

```
VcNet1.AssignCalendarToNodes = False
```

BorderArea**Nur-Lese-Eigenschaft von VcNet**

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 = VcNet1.BorderArea
```

BoxCollection**Nur-Lese-Eigenschaft von VcNet**

Diese Eigenschaft ermöglicht den Zugriff auf die BoxAuflistung und damit auf die verwendeten Boxen.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxCollection	BoxCollection-Objekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Set boxCltn = VcNet1.BoxCollection
```

BoxFormatCollection**Nur-Lese-Eigenschaft von VcNet**

Mit dieser Eigenschaft haben Sie Zugriff auf die BoxFormat-Auflistung, in der alle zur Verfügung stehenden Box-Formate enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxFormatCollection	BoxFormatCollection-Objekt

CalendarCollection

Nur-Lese-Eigenschaft von VcNet

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
Set calendarCltn = VcNet1.CalendarCollection
```

CalendarProfileCollection

Nur-Lese-Eigenschaft von VcNet

Mit dieser Eigenschaft haben Sie Zugriff auf die Kalenderprofilauflistung, in der alle zur Verfügung stehenden Kalenderprofile enthalten sind.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim calendarProfileCltn As VcCalendarProfileCollection
Dim calendarProfile As VcCalendarProfile
Set calendarProfileCltn = VcNet1.CalendarProfileCollection
```

ConfigurationName

Eigenschaft von VcNet

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 *vcnet.ini* sollte sich im gleichen Verzeichnis befinden wie das *vcnet.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.

Hinweis: Beim Einlesen einer neuen Konfigurationsdatei gehen die bestehenden Daten verloren und müssen ggf. wieder eingelesen werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Dateiname, ggf. mit Pfad

Code-Beispiel

```
VcNet1.ConfigurationName = "c:\VARCHART\XNet\sample.ini"
' or:
VcNet1.ConfigurationName = "http://members.tripod.de/netronic_te/_
                             xnet_sample.ini"
```

CtrlCXVProcessing

Eigenschaft von VcNet

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

Code-Beispiel

```
VcNet1.CtrlCXVProcessing = False
```

CurrentVersion**Nur-Lese-Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie die aktuelle Versionsnummer eines VARCHART XNet-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 **vcnet.ocx** im Tab **Version** eingesehen sowie über die FILEVERSION-Ressource dieser Datei gelesen werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Versionsnummer

Code-Beispiel

```
MsgBox VcNet1.CurrentVersion
```

DataDefinition**Nur-Lese-Eigenschaft von VcNet**

Diese Eigenschaft ermöglicht den Zugriff auf die aktuelle Datendefinition, um darin Namen oder Feldtypen abzufragen.

Die DataDefinition des VcNet hat zwei Datendefinitionstabellen: **vcMaindata** und **vcRelations**. Jede Datendefinitionstabelle umfasst eine Menge von Definitionsfeldern, auf die Sie zugreifen können.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataDefinition	DataDefinition-Objekt

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField
```

```

Set dataDefinition = VcNet1.dataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
Set definitionField = dataDefinitionTable.FirstField

While Not definitionField Is Nothing
    Listbox.AddItem definitionField.Name
    Set definitionField = dataDefinitionTable.NextField
Wend

```

DataTableCollection

Nur-Lese-Eigenschaft von VcNet

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 = VcNet1.DataTableCollection
For Each dataTable In dataTableCltn
    List1.AddItem (dataTable.Name)
Next

```

DateOutputFormat

Eigenschaft von VcNet

Mit dieser Eigenschaft stellen Sie das Ausgabeformat von Terminen 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

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	Datumsformat {DMYhms:;}

Code-Beispiel

```
VcNet1.DateOutputFormat = "DD.MM.YY"
```

DiagramBackColor

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Hintergrundfarbe Ihres Netzdiagramms setzen oder erfragen. Die Standard-Farbe ist weiß.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte {0...255},{0...255},{0...255} Standardwert: (255,255,255)

Code-Beispiel

```
VcNet1.DiagramBackColor = RGB(255, 204, 204)
```

DialogFont

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Schriftgröße und den Schrifttyp der zur Laufzeit erscheinenden Dialogfelder im VARCHART XNet erfragen oder festlegen. Als Objekt wird ein Font-Objekt Ihrer Programmierumgebung erwartet, z. B. bei Visual Basic ein Objekt der Klasse **StdFont**.

	Datentyp	Beschreibung
Eigenschaftswert	String	Schriftname

Code-Beispiel

```
Dim font As New StdFont
font.Size = 18
font.name = "Courier"
VcNet1.DialogFont = font
```

DoubleOutputFormat

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Ausgabeformat von Zahlen als double-Wert im Netz-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.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenfolge, die das Double-Format beschreibt, z.B. "I,DDDD ppm"

Code-Beispiel

```
VcNet1.DoubleOutputFormat = "I,DDDD ppm"
```

EditNewLink**Eigenschaft von VcNet**

Mit dieser Eigenschaft wird festgelegt, ob bei der Erzeugung einer neuen Verbindung der Dialog **Daten bearbeiten** erscheinen soll. Die Eigenschaft **AllowNewNodesAndLinks** muss auf True gesetzt sein, damit interaktiv eine neue Verbindung erzeugt werden kann.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft

Code-Beispiel

```
VcNet1.EditNewLink = False
```

EditNewNode**Eigenschaft von VcNet**

Mit dieser Eigenschaft wird festgelegt, ob bei der Erzeugung eines neuen Knotens der Dialog **Vorgänge bearbeiten** erscheinen soll. Dabei muss die Eigenschaft **AllowNewNodesAndLinks** auf True gesetzt sein, damit überhaupt ein neuer Knoten erzeugt werden kann.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vorgänge bearbeiten -Dialogfeld erscheint /erscheint nicht

Code-Beispiel

```
VcNet1.EditNewNode = False
```

Enabled**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie das **VARCHART-XNet**-Steuerelement so abschalten, dass es auf Maus- und Tastenbefehle nicht reagiert.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	VARCHART-ActiveX-Steuerelement angeschaltet/abgeschaltet

Code-Beispiel

```
VcNet1.Enabled = False
```

EnableSupplyTextEntryEvent**Eigenschaft von VcNet**

Mit dieser Eigenschaft aktivieren Sie das **OnSupplyTextEntry**-Ereignis, über das Sie die Texte der **VARCHART-XNet**-Komponente verändern können, z. B. um sie in eine andere Sprache zu übersetzen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft

Code-Beispiel

```
VcNet1.EnableSupplyTextEntryEvent = True
```

EventReturnStatus**Eigenschaft von VcNet**

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	Rückgabewert des Ereignisses Standardwert: vcRetStatOK
	Mögliche Werte: vcRetStatDefault 2 vcRetStatFalse 0 vcRetStatNoPopup 4 vcRetStatOK 1	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 VcNet1_OnDiagramRClick(ByVal x As Long, ByVal y As Long,
returnStatus As Variant)

    VcNet1.EventReturnStatus = vcRetStatNoPopup

End Sub
```

EventText

Nur-Lese-Eigenschaft von VcNet

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	Tool Tip

Code-Beispiel

```
Private Sub VcNet1_OnSupplyTextEntry(ByVal controlIndex As
VcNetLib.TextEntryIndexEnum, TextEntry As String, returnStatus As Variant)

    VcNet1.EventText = "Order189"

End Sub
```

ExtendedDataTables

Eigenschaft von VcNet

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.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	False: nur zwei Datentabellen (Maindata und Relations) True: bis zu 99 Datentabellen Standardwert: False

Code-Beispiel

```
VcNet1.ExtendedDataTables = True
```

FilePath

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie einen Verzeichnispfad setzen, damit auch Grafik- und Gruppentitel-Dateien, 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: " "

Code-Beispiel

```
Dim graphicsPath As String
graphicsPath = App.Path & "\bitmaps"
VcNet1.FilePath = graphicsPath
```

FilterCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf die Filter-Auflistung, die alle vorhandenen Filter enthält.

	Datentyp	Beschreibung
Eigenschaftswert	VcFilterCollection	FilterCollection-Objekt

Code-Beispiel

```
Dim filterCollection As VcFilterCollection
Set filterCollection = VcNet1.FilterCollection
```

FontAntiAliasingEnabled

Eigenschaft von VcNet

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 Knotenfeld 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	Textzeichen werden optisch geglättet / nicht geglättet.

GroupCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht bei gruppierter Darstellung den Zugriff auf die Gruppen-Auflistung, die alle vorhandenen Gruppen enthält.

	Datentyp	Beschreibung
Eigenschaftswert	VcGroupCollection	GroupCollection-Objekt

Code-Beispiel

```
Dim groupCollection As VcGroupCollection
Set groupCollection = VcNet1.GroupCollection
```

GroupDescriptionName

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Namen einer Datei setzen oder erfragen, aus der die Gruppentitel den vorhandenen Gruppen von Knoten zugeordnet werden sollen. Sie ist eine einfache Textdatei, in der pro Zeile eine Zuordnung gemacht wird. Am Anfang der Zeile steht der Gruppenname, dahinter folgt, getrennt durch ein Leerzeichen, der Titel. Innerhalb eines Namens sind keine Leerzeichen erlaubt. Ein leerer Gruppenname wird durch "" gekennzeichnet. Die Standard-Zuordnung für einen Namen ist "".

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-Steuerelements gesucht.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name

Code-Beispiel

```
VcNet1.GroupDescriptionName = "C:\varchart\xnet\samples\net.des"
```

GroupField

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie festlegen oder erfragen, welches Feld aus der Datendefinitionstabelle als Gruppierkriterium verwendet werden soll. Die Sortierung der Gruppen erfolgt automatisch in alphabetischer, numerisch aufsteigender oder absteigender Reihenfolge (s. auch **GroupSortField**). Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datendefinitions-Feldindex

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinition = VcNet1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FieldByName("Code 1")

VcNet1.GroupField = dataDefinitionField.ID
```

GroupHorizontalMargin

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den linken und rechten Rand von Gruppen einstellen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Single 0 ... 9.9 mm	Breite des linken/rechten Randes von Gruppen in mm Standardwert: 0

Code-Beispiel

```
VcNet1.GroupHorizontalMargin = 1.1
```

Grouping

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Gruppieren von Knoten ein- oder ausschalten. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft

Code-Beispiel

```
VcNet1.Grouping = True
```

GroupingTitlesFullyVisible**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Gruppentitel sichtbar bleiben, auch wenn der dargestellte Ausschnitt des Diagramms per Rollbalken verschoben wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Sichtbar (True)/ nicht sichtbar (False) Standardwert: False

GroupInteractionsAllowed**Eigenschaft von VcNet**

Diese Eigenschaft legt fest, ob die Gruppen interaktiv kollabiert bzw. expandiert werden können (durch das Plus- bzw. Minus-Zeichen neben dem Gruppentitel).

Beim interaktiven Kollabieren/Expandieren wird das Ereignis **OnGroupModify** ausgelöst.

Diese Eigenschaft sollte nicht umgeschaltet werden, wenn bereits Gruppen im Chart sichtbar sind.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/nicht in Kraft (False) Standardwert: True

Code-Beispiel

```
VcNet1.GroupInteractionsAllowed = False
```


GroupMode

Eigenschaft von VcNet

Diese Eigenschaft legt den Visualisierungsmodus der Gruppen fest:

- **Gruppierung:** normale Gruppendarstellung (die Breite und Höhe jeder Gruppe wird durch die Positionen der Knoten bestimmt; jede Gruppe nimmt jeweils die gesamte Breite bzw. Höhe des Netzdiagramms ein)
- **Clusterung:** Die Gruppen umschließen die darin enthaltenen Knoten möglichst platzsparend, wobei die Gruppen frei im Netzdiagramm verteilt sind.

Die Eigenschaft sollte nicht umgeschaltet werden, wenn bereits Gruppen im Chart sichtbar sind. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Gruppierung** eingestellt werden.

Weitere Informationen finden Sie im Kapitel "Wichtige Begriffe: Gruppierung".

	Datentyp	Beschreibung
Eigenschaftswert	GroupModeEnum Mögliche Werte: vcGMClustering 1 vcGMGrouping 0	Visualisierungsmodus Standardwert: 0 Clusterung normale Gruppendarstellung

Code-Beispiel

```
VcNet1.GroupMode = vcGMClustering
```

GroupMovingAllowed

Eigenschaft von VcNet

Mit dieser Eigenschaft wird dem Anwender bei Clusterung das Verschieben kollabierter Cluster erlaubt (True) oder verboten (False). Diese Eigenschaft kann auch auf der Eigenschaftenseite **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verschieben kollabierter Cluster erlaubt (True)/nicht erlaubt (False) Standardwert: True

Code-Beispiel

```
Dim boole As Boolean

boole = VcNet1.GroupMovingAllowed
```

GroupSortField**Eigenschaft von VcNet**

Mit dieser Eigenschaft stellen Sie ein, welches Feld der Datendefinitionstabelle zum Sortieren der Gruppen verwendet werden soll. Standardmäßig werden die Gruppen in alphabetischer Reihenfolge nach dem **GroupField** sortiert. Durch Verwendung von **GroupSortField** können Sie jede beliebige Reihenfolge erzielen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datendefinitions-Feldindex

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinition = VcNet1.dataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FieldByName("Code 3")

VcNet1.GroupSortField = dataDefinitionField.ID
```

GroupSortMode**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie die Sortierreihenfolge der Gruppen festlegen oder erfragen. Als Standard ist die Reihenfolge "aufsteigend nach dem Gruppenfeld" (vcAscending) eingestellt. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	GroupSortModeEnum	Sortierreihenfolge der Gruppen Standardwert: vcAscending
	Mögliche Werte: vcAscending 65 vcDescending 68 vcGroupDescriptionOrder 88	aufsteigende Sortierung nach dem Gruppenfeld absteigende Sortierung nach dem Gruppenfeld Sortierung nach der Reihenfolge der Gruppen (GroupDescriptionOrder), die in der Gruppentitel-Datei festgelegt sind (s. GroupDescriptionName).

546 API-Referenz: VcNet

vcNodeOrder 73

Sortierung nach der Reihenfolge der Erzeugung der Knoten

Code-Beispiel

```
VcNet1.GroupSortMode = vcDescending
```

GroupTitleField

Eigenschaft von VcNet

Mit dieser Eigenschaft können sie den Datendefinitions-Feldindex aus dem Knotendatensatz erfragen oder festlegen, aus dem der Gruppentitel verwendet werden soll. Ein Gruppentitel dient der Beschriftung einer Gruppe und wird in der obersten Zeile ausgegeben. Es empfiehlt sich, bei allen Knoten mit gleichem Gruppennamen denselben Titel zu verwenden, da sonst nicht absehbar ist, welcher Titel angezeigt wird. Wenn allerdings ein Dateiname in der Eigenschaft **GroupDescriptionName** steht, wird der dort vermerkte Titel benutzt.

Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer 0 ... 9.9 mm	Datendefinitions-Feldindex

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinition = VcNet1.dataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FieldByName("Code 2")

VcNet1.GroupTitleField = dataDefinitionField.ID
```

GroupVerticalMargin

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den oberen und unteren Rand von Gruppen einstellen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Gruppierung** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Single 0 ... 9.9 mm	Höhe des oberen/unteren Randes von Gruppen in mm Standardwert: 0

Code-Beispiel

```
VcNet1.GroupVerticalMargin = 0.9
```

hWnd**Nur-Lese-Eigenschaft von VcNet**

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))
```

InFlowGroupDescriptionName**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie die Datei, die die Gruppentiteltexte für die In-Flow-Gruppierung enthält, erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datei, die die Titeltexte für die In-Flow-Gruppierung enthält

InFlowGroupField

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Datenfeld, das die In-Flow-Gruppierung bestimmt, erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Feld, nach dem gruppiert wird Standardwert: -1

Code-Beispiel

```
VcNet1.InFlowGroupField = 16
```

InFlowGroupingEnabled

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die In-Flow-Gruppierung aktivieren oder deaktivieren. Es wird dann automatisch eine neue Layoutberechnung für das Netzdiagramm durchgeführt (siehe VcNet-Methode **Arrange**). Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	In-Flow-Gruppierung aktiviert (True)/ deaktiviert (False) Standardwert: False

Code-Beispiel

```
VcNet1.InFlowGroupingEnabled = True
```

InFlowGroupSeparationLineColor

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Farbe der Trennlinien der In-Flow-Gruppen erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Farbe der Trennlinien der In-Flow-Gruppen

InFlowGroupSeparationLineType

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Typ der Trennlinien der In-Flow-Gruppen erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

Eigenschaftswert	Datentyp	Beschreibung
	LineTypeEnum	Typ der Trennlinien der In-Flow-Gruppen
	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

InFlowGroupTimeInterval

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Intervall der In-Flow-Gruppierung (z. B. 1 Sekunde, 1 Minute, 1 Stunde, 1 Tag, 2 Monate, 1 Jahr) festlegen oder erfragen. Diese Eigenschaft kann auch im Dialog **In-Flow-Gruppierung bearbeiten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	TimeOrientationIntervalEnum	Intervall der In-Flow-Gruppierung
	Mögliche Werte: vcDay 5 vcFifteenMinutes 1848 vcFifteenSeconds 1845 vcFourHours 1853 vcHalfYear 1242 vcHour 6 vcMinute 7 vcMonth 3 vcQuarter 2 vcSecond 8 vcSixHours 1854 vcThirtyMinutes 1849 vcThirtySeconds 1846 vcThreeHours 1852 vcTwelveHours 1855 vcTwoHours 1851 vcTwoWeeks 1238 vcTwoYears 1245 vcWeek 4 vcYear 1	Tag 15 Minuten 15 Sekunden 4 Stunden halbes Jahr Stunde Minute Monat Quartal (3 Monate) Sekunde 6 Stunden 30 Minuten 30 Sekunden 3 Stunden 12 Stunden 2 Stunden zwei Wochen zwei Jahre Woche Jahr

Code-Beispiel

```
VcNet1.InFlowGroupTimeInterval = vcDay
```

InFlowGroupTitleField

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Datenfeld, das die Titel der In-Flow-Gruppen enthält, erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeld, das die Titel der In-Flow-Gruppen enthält

Code-Beispiel

```
VcNet1.InFlowGroupTitleField = 1
```

InFlowGroupTitlesBackColor

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Hintergrundfarbe der Titel der In-Flow-Gruppen erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Hintergrundfarbe der Titel der In-Flow-Gruppen

InFlowGroupTitlesFont

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Schriftattribute der Titel der In-Flow-Gruppen erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart der Titel der In-Flow-Gruppen

InFlowGroupTitlesVisibleAtBottomOrRight

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Titel der In-Flow-Gruppen am unteren oder rechten Rand der Grafik sichtbar sind. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Sichtbar (True)/ nicht sichtbar (False)

InFlowGroupTitlesVisibleAtTopOrLeft

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Titel der In-Flow-Gruppen am oberem oder linken Rand der Grafik sichtbar sind. Sie

können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Sichtbar (True)/ nicht sichtbar (False)

InFlowGroupTitleTimeFormat

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Datums-/Zeit-Ausgabeformat für die In-Flow-Gruppierung nach Datumsfeld erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Datums-/Zeit-Ausgabeformat für die In-Flow-Gruppierung nach Datumsfeld Standardwert: DD.MM.YYYY

InFlowGroupVerticalCaptionWidth

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie die Breite der Titelleisten bei Orientierung von oben nach unten in mm erfragen oder setzen. Sie können diese Eigenschaft auch im Dialog **In-Flow-Gruppierung bearbeiten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite der Titelleisten bei Orientierung von oben nach unten in mm Standardwert: 50

Code-Beispiel

```
VcNet1.InFlowGroupVerticalCaptionWidth = 30
```

InPlaceEditingAllowed

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob in Knotenfeldern und Boxen das direkte Editieren 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.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren in Knotenfeldern möglich (True) / nicht möglich (False) Standardwert: True

Code-Beispiel

```
VcNet1.InPlaceEditingAllowed = True
```

InteractionMode

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie einen der verfügbaren Interaktionsmodi einstellen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	InteractionModeEnum	Interaktionsmodus Standardwert: vcPointer
	Mögliche Werte: vcCreateLink 4 vcCreateNodesAndLinks 1 vcPointer 0	Erzeugemodus für Verbindungen Erzeugemodus für Knoten und Verbindungen Selektiermodus

Code-Beispiel

```
VcNet1.InteractionMode = vcCreateNodesAndLinks
```

InterfaceNodesShown

Eigenschaft von VcNet

Mit Hilfe dieser Eigenschaft können Sie festlegen, ob bei der Erstellung eines Teildiagramms auch die Anschlussknoten dargestellt werden sollen (True) oder nicht (False). Das Aussehen dieser Anschlussknoten kann man im Dialog **Knotenaussehen bearbeiten** bestimmen, indem man als Spezialfilter <InterfaceNodes> angibt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft Standardwert: True

Code-Beispiel

```
VcNet1.InterfaceNodesShown = False
```

LegendView**Nur-Lese-Eigenschaft von VcNet**

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 = VcNet1.LegendView
legendview.Visible = True
```

LinkAnnotationColumnNumberDataFieldIndex**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, in das die Spaltennummer jeder Verbindungsbeschriftung geschrieben werden soll. Das Setzen dieser Eigenschaft ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Spaltennummer jeder Verbindungsbeschriftung enthält

LinkAnnotationRowNumberDataFieldIndex**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, in das die Zeilennummer jeder Verbindungsbeschriftung geschrieben werden soll. Das Setzen dieser Eigenschaft ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Zeilennummer jeder Verbindungsbeschriftung enthält

LinkAppearanceCollection

Nur-Lese-Eigenschaft von VcNet

Mit dieser Eigenschaft haben Sie Zugriff auf die LinkAppearance-Auflistung, in der alle vorhandenen LinkAppearance-Objekte enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkAppearanceCollection	LinkAppearanceCollection-Objekt

Code-Beispiel

```
Dim linkAppearanceCollection As VcLinkAppearanceCollection
Dim linkAppearance As VcLinkAppearance

Set linkAppearanceCollection = VcNet1.LinkAppearanceCollection
Set linkAppearance = linkAppearanceCollection.FirstLinkAppearance
```

LinkCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf die Link-Auflistung, die alle vorhandenen Verbindungen enthält.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkCollection	LinkCollection-Objekt

Code-Beispiel

```
Dim linkCollection As VcLinkCollection
Dim numberOfLinks As Integer

Set linkCollection = VcNet1.LinkCollection
numberOfLinks = linkCollection.Count
```

LinkFormatCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf die LinkFormat-Auflistung und damit auf die verwendeten Verbindungsformate.

	Datentyp	Beschreibung
Eigenschaftswert	VcLinkFormatCollection	LinkFormatCollection-Objekt

Code-Beispiel

```
Dim formatCollection As VcLinkFormatCollection
Set formatCollection = VcNet1.LinkFormatCollection
```

LinkPredecessorDataFieldIndex**Eigenschaft von VcNet**

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.

	Datentyp	Beschreibung
Parameter:		
Rückgabewert	Long	Feldindex aus der Datendefinitionstabelle
Eigenschaftswert	Integer	Index des Vorgängerknotens {0...2}

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcNet2.DataTableCollection.Add("LinkDataTable")
VcNet1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add("Predecessor")
dataTable.DataTableFieldCollection.Add("Successor")
VcNet1.DataTableCollection.Update

VcNet1.LinkPredecessorDataFieldIndex(0) =
VcNet1.DetectFieldIndex("LinkDataTable", "Id")
VcNet1.LinkSuccessorDataFieldIndex(0) = VcNet1.DetectFieldIndex("LinkDataTable",
"Id")

'Load Data
Set dataTable = VcNet1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcNet1.EndLoading
```

LinksDataTableName**Eigenschaft von VcNet**

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.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle aus der die Verbindungen kommen

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcNet2.DataTableCollection.Add("LinkDataTable")
VcNet1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")
dataTable.DataTableFieldCollection.Add ("Successor")
VcNet1.DataTableCollection.Update

VcNet1.LinkPredecessorDataFieldIndex(0) =
VcNet1.DetectFieldIndex("LinkDataTable", "Id")
VcNet1.LinkSuccessorDataFieldIndex(0) = VcNet1.DetectFieldIndex("LinkDataTable",
"Id")

'Load Data
Set dataTable = VcNet1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcNet1.EndLoading
```

LinkSuccessorDataFieldIndex

Eigenschaft von VcNet

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 wurde

	Datentyp	Beschreibung
Parameter: identifizierIndex	Integer	Index des Vorgängerknotens {0...2}
Eigenschaftswert	Long	Feldindex aus der Datendefinitionstabelle

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Link DataTable
Set dataTable = VcNet2.DataTableCollection.Add("LinkDataTable")
VcNet1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")
dataTable.DataTableFieldCollection.Add ("Successor")
VcNet1.DataTableCollection.Update

VcNet1.LinkPredecessorDataFieldIndex(0) =
VcNet1.DetectFieldIndex("LinkDataTable", "Id")
VcNet1.LinkSuccessorDataFieldIndex(0) = VcNet1.DetectFieldIndex("LinkDataTable",
"Id")
```

```
'Load Data
Set dataTable = VcNet1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcNet1.EndLoading
```

LinkTypeDataFieldIndex

Eigenschaft von VcNet

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.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Verbindungstypen enthält

Code-Beispiel

```
Dim dataTable As VcDataTable

'create Link DataTable
Set dataTable = VcNet1.DataTableCollection.Add("LinkDataTable")
VcNet1.LinksDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
dataTable.DataTableFieldCollection.Add ("Predecessor")
dataTable.DataTableFieldCollection.Add ("Successor")
dataTable.DataTableFieldCollection.Add("LinkType")
VcNet1.DataTableCollection.Update
```

MapCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf die Map-Auflistung, in der bestimmte Zuordnungstabellen enthalten sind. Die Zuordnungstabellen werden durch die Methode **VcMapCollection.SelectMaps** ausgewählt.

	Datentyp	Beschreibung
Eigenschaftswert	VcMapCollection	MapCollection-Objekt

Code-Beispiel

```
Dim mapCollection As VcMapCollection

Set mapCollection = VcNet1.MapCollection
mapCollection.SelectMaps vcAnyMap
```

MinimumColumnWidth

Eigenschaft von VcNet

Mit dieser Eigenschaft wird die minimale Breite einer Spalte (Einheit: mm) eingestellt. Die eingestellte Breite sollte etwa der mittleren Breite eines Knotens entsprechen. Damit Verbindungen bei der Orientierung von links nach rechts weniger Platz auf dem Bildschirm einnehmen, kann die Breite weiter herabgesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long {1...1 000}	Mindestspaltenbreite in mm Standardwert: 1

Code-Beispiel

```
VcNet1.MinimumColumnWidth = 100
```

MinimumRowHeight

Eigenschaft von VcNet

Mit dieser Eigenschaft wird die minimale Höhe einer Zeile (Einheit: 1/100 mm) eingestellt. Die eingestellte Höhe sollte etwa der mittleren Höhe eines Knotens entsprechen. Damit Verbindungen bei der Orientierung von oben nach unten weniger Platz auf dem Bildschirm einnehmen, kann die Höhe weiter herabgesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** 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 {1...1 000}	Mindestzeilenhöhe in mm Standardwert: 1

Code-Beispiel

```
VcNet1.MinimumRowHeight = 100
```


MouseProcessingEnabled

Eigenschaft von VcNet

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 XNet 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

NodeAppearanceCollection

Nur-Lese-Eigenschaft von VcNet

Mit dieser Eigenschaft haben Sie Zugriff auf die NodeAppearance-Auflistung, in der alle zur Verfügung stehenden NodeAppearance-Objekte enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeAppearanceCollection	NodeAppearanceCollection-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.BackColor = RGB(100, 100, 100)
```

NodeCalendarNameDataFieldIndex

Eigenschaft von VcNet

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.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das den Namen des für einen Knoten zu verwendenden Kalenders enthält.

NodeChangeRankToPredecessorRankDataFieldIndex

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, in das der Rang des Vorläuferknotens eingetragen wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes, das die Rangnummer des Vorläuferknotens enthält

NodeCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf eine Menge von Knoten, abhängig von der Einstellung von SelectNodes. Sie ermöglicht entweder den Zugriff auf alle Knoten, nur die markierten oder nur die sichtbaren.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	NodeCollection-Objekt

Code-Beispiel

```
Dim nodeCollection As VcNodeCollection
Dim node As VcNode
Dim numberNodes As Integer

Set nodeCollection = VcNet1.NodeCollection
numberNodes = nodeCollection.Count
```

NodeColumnNumberDataFieldIndex

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen oder erfragen, in das die Spaltennummer jedes Vorgangs geschrieben werden soll. Das Setzen dieser Eigenschaft ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Spaltennummer jedes Vorgangs enthält

NodeFormatCollection

Nur-Lese-Eigenschaft von VcNet

Diese Eigenschaft ermöglicht den Zugriff auf die NodeFormat-Auflistung und damit auf die verwendeten Knotenformate.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeFormatCollection	NodeFormatCollection-Objekt

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Set formatCollection = VcNet1.NodeFormatCollection
```

NodeRowNumberDataFieldIndex

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie den Index des Datenfeldes festlegen/erfragen, in das die Zeilennummer jedes Vorgangs geschrieben werden soll. Das Setzen dieser Eigenschaft ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, das die Zeilennummer jedes Vorgangs enthält

Code-Beispiel

```
Private Sub Form_Load()
    VcNet1.NodeRowNumberDataFieldIndex = VcNet1.DetectFieldIndex("NodeDataTable",
"SortNumber")
    'Load data
    Call loadData
    VcNet1.UpdateRowNumberFields
    VcNet1.SaveAsEx "C:\ProjectData.txt", vcUnicodeEncoding
End Sub
```

NodesDataTableName

Eigenschaft von VcNet

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.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle aus der die Knotendaten kommen

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Node DataTable
Set dataTable = VcNet1.DataTableCollection.Add("NodeDataTable")
VcNet1.NodesDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
'Load Data
Set dataTable = VcNet1.DataTableCollection.DataTableByName("NodeDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;Node One;")
Set dataRecord = dataTable.DataRecordCollection.Add("2;Node Two;")

Set dataTable = VcNet1.DataTableCollection.DataTableByName("LinkDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;1;2;")
VcNet1.EndLoading
```

NodeTooltipTextField

Eigenschaft von VcNet

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

Code-Beispiel

```
VcNet1.NodeTooltipTextField = 1
```

ObliqueTracksOnLinks

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Verbindungslinien direkt an den horizontalen Linienstücken ansetzen und schräg verlaufen oder ob die Verbindungslinien orthogonal dargestellt werden. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean {True, False}	schräge Verbindungslinien (True)/orthogonale Verbindungslinien (False) Standardwert: False

Code-Beispiel

```
VcNet1.ObliqueTracksOnLinks = True
```

OLEDragMode

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Ziehen eines Knotens über die Grenze der VARCHART-XNet-Komponente hinaus erlaubt sein soll. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Knoten** einstellen.

Wenn **OLEDragMode** auf **vcOLEDragManual** gesetzt wird, müssen Sie die Methode **OLEDrag** aufrufen, um das Ziehen eines Knotens zu starten. Bei **vcOLEDragAutomatic** hingegen wird das Ziehen eines Knotens über die Grenzen der VARCHART-XNet-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 XNet 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.

Das OLE Drag & Drop-Verhalten des VARCHART XNet 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-XNet-Komponente heraus Standardwert: vcOLEDragManual
	Mögliche Werte: vcOLEDragAutomatic 1 vcOLEDragManual 0	Methode OLEDrag wird automatisch aufgerufen Methode OLEDrag muß eigens aufgerufen werden.

Code-Beispiel

```
VcNet1.OLEDragMode = vcOLEDragAutomatic
```

OLEDragWithOwnMouseCursor

Nur-Lese-Eigenschaft von VcNet

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 Cursor 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.

Außerdem können bei eingeschaltetem Cursor und der auf **vcOLEDropManual** gesetzten Eigenschaft Objekte außerhalb der Anlagerungsstellen eines Knotens nicht fallengelassen werden, während dies bei ausgeschaltetem Cursor möglich ist.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Knoten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mauszeiger wird/wird nicht in der Zielkomponente gesetzt Standardwert: True

Code-Beispiel

```
VcNet1.OLEDragWithOwnMouseCursor = False
```

OLEDragWithPhantom

Eigenschaft von VcNet

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.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Knoten** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Phantom erscheint/erscheint nicht Standardwert: True

Code-Beispiel

```
VcNet1.OLEDragWithPhantom = False
```

OLEDropMode

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob Knoten und ihre Verbindungen aus einer anderen VARCHART-XNet-Komponente in die aktuelle Komponente herein gezogen werden dürfen.

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.

Das OLE Drag & Drop-Verhalten von VARCHART XNet 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.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Knoten** setzen.

	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

```
VcNet1.OLEDropMode = vcOLEDropAutomatic
```

Orientation**Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie die Flussrichtung des Diagramms einstellen oder erfragen. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	LayoutOrientationEnum	Von links nach rechts/von oben nach unten
	Mögliche Werte:	
	vcLeftToRight 0	Flussrichtung im Netzdiagramm von links nach rechts
	vcTopToBottom 1	Flussrichtung im Netzdiagramm von oben nach unten

Code-Beispiel

```
VcNet1.Orientation = vcLeftToRight
```

Printer**Eigenschaft von VcNet**

Mit dieser Methode 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 = VcNet1.Printer.ZoomFactor
printerCuttingMarks = VcNet1.Printer.CuttingMarks
```

RoundedLinkSlantsEnabled**Nur-Lese-Eigenschaft von VcNet**

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob die Schrägen bei Verbindungen vom Routing-Typ **vcLRTOrthogonal** 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	in

Code-Beispiel

```
VcNet1.RoundedLinkSlantsEnabled = True
```

Scheduler**Nur-Lese-Eigenschaft von VcNet**

Diese Eigenschaft gibt das VcScheduler-Objekt zurück.

	Datentyp	Beschreibung
Eigenschaftswert	VcScheduler	Gibt das VcScheduler-Objekt zurück

ScrollOffsetX**Eigenschaft von VcNet**

Mit dieser Eigenschaft kann der aktuelle Scroll-Offset des angezeigten Ausschnitts in x-Richtung gespeichert werden und bei einem neuen Start der

gleichen Anwendung wieder gesetzt werden. Letzteres setzt voraus, dass zuvor auch der Zoomfaktor auf dieselbe Weise gesetzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Scroll-Offset in x-Richtung

ScrollOffsetY

Eigenschaft von VcNet

Mit dieser Eigenschaft kann der aktuelle Scroll-Offset des angezeigten Ausschnitts in y-Richtung gespeichert werden und bei einem neuen Start der gleichen Anwendung wieder gesetzt werden. Letzteres setzt voraus, dass zuvor auch der Zoomfaktor auf dieselbe Weise gesetzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Scroll-Offset in y-Richtung

ShortenedLinks

Eigenschaft von VcNet

Diese Eigenschaft wirkt sich auf das Layout eines Netzdiagramms aus und wird von der Methode **Arrange** berücksichtigt. Wird die Eigenschaft auf **True** gesetzt, werden alle Knoten so nahe wie möglich an ihre Nachfolgerknoten herangeschoben, um die Verbindungen möglichst kurz zu halten. Wird sie auf **False** gesetzt, werden Knoten grundsätzlich so weit links bzw. oben wie möglich angeordnet. Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft

Code-Beispiel

```
VcNet1.ShortenedLinks = False
VcNet1.Arrange
```

ShowToolTip

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie das Ereignis **OnToolTipText** aktivieren oder deaktivieren. Mit Hilfe dieses Ereignisses können Sie die Texte der Tooltips bestimmen.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft Standardwert: False

Code-Beispiel

```
VcNet1.ShowToolTip = True
```

StraightLinkDrawing

Eigenschaft von VcNet

Ist diese Eigenschaft auf **True** gesetzt, können Verbindungen zwischen Knoten direkt, das heißt, nicht orthogonal mit Knicken, gezeichnet werden.

Diese Eigenschaft macht die Eigenschaft **ObliqueTracksOnLinks** unwirksam, wenn sie gesetzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Zeichnen von Verbindungen eingeschaltet (True)/ausgeschaltet (False) Standardwert: False

TimeUnit

Eigenschaft von VcNet

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	Zeiteinheit Standardwert: vcDay

Code-Beispiel

```
Dim timeUnit As TimeUnitEnum
timeUnit = VcNet1.TimeUnit
```

ToolTipChangeDuration

Eigenschaft von VcNet

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

Code-Beispiel

```
VcNet1.ToolTipText = "Object"
VcNet1.ToolTipChangeDuration = 1000
```

ToolTipDuration

Eigenschaft von VcNet

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

Code-Beispiel

```
VcNet1.ToolTipText = "Object"
VcNet1.ToolTipDuration = 1000
```

ToolTipPointerDuration**Eigenschaft von VcNet**

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

Code-Beispiel

```
VcNet1.ToolTipText = "Object"
VcNet1.ToolTipPointerDuration = 1000
```

ToolTipShowAfterClick**Eigenschaft von VcNet**

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

Code-Beispiel

```
VcNet1.ToolTipShowAfterClick = True
```

UngroupedNodesAllowed**Eigenschaft von VcNet**

Diese Eigenschaft legt fest, ob Knoten, deren Gruppencode der leere String "" ist, nicht innerhalb einer Gruppe oder in einer eigenen Gruppe für Vorgänge ohne Gruppencode-Angabe dargestellt werden.

Diese Eigenschaft ist nur beim Gruppierungsmodus Clusterung (GroupMode = vcGMClustering) aktiv.

Sie sollte nicht umgeschaltet werden, wenn bereits Gruppen im Chart sichtbar sind.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Gruppierung** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/nicht in Kraft (False) Standardwert: False

Code-Beispiel

```
VcNet1.UngroupedNodesAllowed = True
```

WaitCursorEnabled

Nur-Lese-Eigenschaft von VcNet

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 Eigenschaftenseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Wartecursor wird/wird nicht gesetzt

WorldView

Nur-Lese-Eigenschaft von VcNet

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 = VcNet1.WorldView
```

```
worldview.Visible = True
```

ZoomFactor

Eigenschaft von VcNet

Mit dieser Eigenschaft kann der absolute Zoomfaktor der Bildschirmdarstellung in % angegeben oder erfragt werden (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Integer {0...1000}	Zoomfaktor

Code-Beispiel

```
VcNet1.ZoomFactor = 150
```

ZoomingPerMouseWheelAllowed

Eigenschaft von VcNet

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das Zoomen per Mausrad zugelassen ist. Um zu zoomen, muss der Anwender dann die Strg-Taste festhalten und das Mausrad drehen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zoomen erlaubt (True)/nicht erlaubt (False)

Code-Beispiel

```
VcNet1.ZoomingPerMouseWheelAllowed = False
```

Methoden

AboutBox

Methode von VcNet

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

```
VcNet1.AboutBox
```

Arrange

Methode von VcNet

Mit dieser Methode wird eine neue Layoutberechnung für das Netzdiagramm durchgeführt. Dabei wird die Eigenschaft **ShortenedLinks** berücksichtigt.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.Arrange
```

Clear

Methode von VcNet

Mit dieser Methode werden alle grafisch darstellbaren Objekte aus dem Diagramm gelöscht. Das Ergebnis dieser Methode ist ein "leeres" Chart (wie beim ersten Start gemäß den Einstellungen in der INI-Datei), in das nun wieder Objekte geladen werden können.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich gelöscht {True}

Code-Beispiel

```
VcNet1.Clear
```

CopyNodesIntoClipboard

Methode von VcNet

Mit dieser Methode werden die markierten Knoten und Verbindungen vom Netzdiagramm in den Zwischenspeicher kopiert. Verbindungen werden

allerdings nur kopiert, wenn sowohl Vorgänger als auch Nachfolger markiert sind. Siehe auch die Methoden **CutNodesIntoClipboard** und **PasteNodesFromClipboard**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.CopyNodesIntoClipboard
```

CutNodesIntoClipboard

Methode von VcNet

Mit dieser Methode werden die im Netzdiagramm markierten Knoten und Verbindungen ausgeschnitten und in den Zwischenspeicher abgelegt. Verbindungen werden allerdings nur berücksichtigt, wenn sowohl Vorgänger als auch Nachfolger markiert sind. Siehe auch **CopyNodesIntoClipboard** und **PasteNodesFromClipboard**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.CutNodesIntoClipboard
```

DeleteLinkRecord

Methode von VcNet

Mit dieser Methode können Sie eine Verbindung löschen, indem Sie den Datensatz der Verbindung angeben (s. auch Objekt **VcLink**, Methode **DeleteLink**).

	Datentyp	Beschreibung
Parameter: ⇒ linkRecord	Variant	Datensatz der Verbindung
Rückgabewert	Boolean	Datensatz der Verbindung erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
VcNet1.DeleteLinkRecord "A100;A105;;"
```

DeleteNodeRecord

Methode von VcNet

Mit dieser Methode können Sie einen Knoten löschen. Der Knoten wird durch die ID im Datensatz festgelegt. Welches Datenfeld als Identifizierung verwendet wird, wird auf der Eigenschaftenseite **Datendefinition** festgelegt.

	Datentyp	Beschreibung
Parameter: ⇒ nodeRecord	Variant	Datensatz des Knotens
Rückgabewert	Boolean	Datensatz des Knotens erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
VcNet1.DeleteNodeRecord "A100;;;;;"
```

DetectDataTableFieldName

Methode von VcNet

Mit dieser Eigenschaft 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 = VcNet1.DetectDataTableFieldName(0)
```

DetectDataTableName

Methode von VcNet

Mit dieser Eigenschaft 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 = VcNet1.DetectDataTableName(0)
```

DetectFieldIndex**Methode von VcNet**

Mit dieser Eigenschaft können Sie über den Namen einer Datentabelle und den Feldnamen den Index eines Tabellendatenfeldes erfragen.

	Datentyp	Beschreibung
Parameter: ⇒ tableName	String	Name der Datentabelle, in der sich das Feld befindet, dessen Index ermittelt werden soll
⇒ dataTableFieldName	String	Name des Datentabellenfeldes, dessen Index ermittelt werden soll
Rückgabewert	String	Zurückgegebener Index des Datentabellenfeldes

Code-Beispiel

```
'Find the index of a DataTableField
Dim fieldIndex As Integer

fieldIndex = VcNet1.DetectFieldIndex("Maindata", "Name")
```

DumpConfiguration**Methode von VcNet**

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	String	Dateiname, ggf. mit Pfad.

⇒ encoding	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	Datei erfolgreich (True)/nicht erfolgreich (False) abgespeichert.

EditLink

Methode von VcNet

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 VcNet1_OnLinkLClickCltn(ByVal linkCltn As VcNetLib.VcLinkCollection,
-
                                ByVal x As Long, ByVal y As Long, returnStatus As
Variant)
    If linkCltn.Count = 1 Then
        VcNet1.EditLink linkCltn.FirstLink
    End If
End Sub
```

EditNode

Methode von VcNet

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 VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
                               ByVal x As Long, ByVal y As Long, returnStatus As
Variant)
    VcNet1.EditNode node
End Sub
```

EndLoading**Methode von VcNet**

Hiermit wird das Ende des Ladevorgangs via **InsertNodeRecord** oder **InsertLinkRecord** angezeigt. Dadurch wird eine Aktualisierung der Grafik ausgelöst.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ladevorgang beendet {True}

Code-Beispiel

```
VcNet1.EndLoading
```

ExportGraphicsToFile**Methode von VcNet**

Mit dieser Methode können Sie ein Netzdiagramm 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 XNet 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.
⇒ PrintOutputFormat	PrintOutputFormat	Format der abzuspeichernden Datei
	Mögliche Werte:	
	vcBMP 2	Datei wird im Format BMP geschrieben.
	vcEMF 9	Datei wird im Format EMF geschrieben.
	vcEMFPlus 12	Datei wird als *.EMF-Datei geschrieben, beinhaltet aber nur EMF+-Format.
	vcEMFWithEMFPlusIncluded 11	Datei wird als *.EMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF+-Format Wird nicht mehr unterstützt.
	vcEPS 3	Datei wird im Format GIF geschrieben.
	vcGIF 4	Datei wird im Format JPG geschrieben.
	vcJPG 5	Wird nicht mehr unterstützt.
	vcPCX 6	Datei wird im Format PNG geschrieben.
	vcPNG 7	Datei wird im Format TIF geschrieben.
	vcTIF 8	Datei wird im Format VMF geschrieben.
	vcVMF 0	Datei wird im Format WMF geschrieben.
	vcWMF 1	Datei wird als *.WMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF-Format.
	vcWMFWithEMFIncluded 10	
⇒ 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	Höhe des exportierten Diagramms in Pixeln. Nur bei Pixelformaten möglich. Bei Angabe von 0 wird der Wert unter Beachtung des Seitenverhältnisses berechnet.
Rückgabewert	Boolean	Datei erfolgreich (True) / nicht erfolgreich (False) abgespeichert.

Code-Beispiel

```
VcNet1.ExportGraphicsToFile"C:\temp\export", vcVMF, 0, 0
```

GetAValueFromARGB

Methode von VcNet

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 = VcNet1.MakeARGB(alpha, red, green, blue)
alpha = VcNet1.GetAValueFromARGB(argb)
```

GetBValueFromARGB

Methode von VcNet

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 = VcNet1.MakeARGB(alpha, red, green, blue)
blue = VcNet1.GetBValueFromARGB(argb)
```


GetGValueFromARGB

Methode von VcNet

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 = VcNet1.MakeARGB(alpha, red, green, blue)
green = VcNet1.GetGValueFromARGB(argb)
```

GetLinkByID

Methode von VcNet

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 link As VcLink
```

```
Set link = VcNet1.GetLinkByID(" 5")
```

GetLinkByIDs

Methode von VcNet

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	String	Identifikation des Vorgängerknotens
⇒ successorID	String	Identifikation des Nachfolgerknotens
Rückgabewert	VcLink	Verbindung

Code-Beispiel

```
Dim link As VcLink
```

```
Set link = VcNet1.GetLinkByIDs(" 2", " 3")
```

GetNodeByID

Methode von VcNet

Mit dieser Methode können Sie auf einen einzelnen Knoten über seine ID 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 = VcNet1.GetNodeByID("10")
```

GetRValueFromARGB**Methode von VcNet**

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 = VcNet1.MakeARGB(alpha, red, green, blue)
red = VcNet1.GetRValueFromARGB(argb)
```

IdentifyFormatField**Methode von VcNet**

Mit dieser Methode können Sie das aktuell verwendete Format des angegebenen Knotens sowie den Index des an der bezeichneten Position befindlichen Formatfeldes erfragen. Falls sich an der bezeichneten Position ein Feld befindet, wird **True** zurückgegeben, andernfalls **False**.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **IdentifyFormatFieldAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate der Position
⇒ y	Long	Y-Koordinate der Position
⇒ node	VcNode	Referenzknoten
⇐ format	VcNodeFormat	Identifiziertes Format
⇐ formatFieldIndex	Integer	Format-Feldindex
Rückgabewert	Boolean	Ein Formatfeld befindet sich/befindet sich nicht an der angegebenen Position

Code-Beispiel

```
Private Sub VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
                               ByVal x As Long, ByVal y As Long, _
                               returnStatus As Variant)

    Dim foundFlag As Boolean
    Dim format As VcNodeFormat
    Dim formatFieldIndex As Integer

    foundFlag = VcNet1.IdentifyFormatField(x, y, node, format, formatFieldIndex)
    If foundFlag Then
        MsgBox "You hit the field with the index "+CStr(formatFieldIndex)
    End If
End Sub
```

IdentifyFormatFieldAsVariant

Methode von VcNet

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyFormatField**. 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.

IdentifyObjectAt

Methode von VcNet

Mit dieser Methode können Sie das Objekt, das sich an einer bestimmten Position des Diagramms befindet, identifizieren. Der Typ des Objekts wird zurückgegeben.

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 vcObjTypeGroup 7 vcObjTypeLinkCollection 3 vcObjTypeNode 2 vcObjTypeNodeInLegend 17 vcObjTypeNone 0	Objekttyp Box Objekttyp Gruppe Objekttyp LinkCollection Objekttyp Knoten Objekttyp Knoten im Legendbereich kein Objekt
Rückgabewert	Boolean	Objekt identifiziert/Objekt nicht identifiziert

Code-Beispiel

```
Private Sub VcNet1_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 VcNet1.IdentifyObjectAt(x, y, identifiedObject, identifiedObjectType)

    Select Case identifiedObjectType
        Case VcObjectTypeEnum.vcObjTypeNodeInDiagram,
VcObjectTypeEnum.vcObjTypeNodeInTable
        Set node = identifiedObject
        Labell1.Caption = node.DataField(1)
    Case Else
        Labell1.Caption = ""
    End Select
End Sub
```

IdentifyObjectAtAsVariant

Methode von VcNet

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 VcNet

Mit dieser Methode wird eine neue Verbindung erzeugt. Die Daten werden als CSV-String (mit Trennzeichen Semikolon) gemäß der auf der Eigenschaftenseite **Datendefinition** festgelegten Struktur übergeben. Die Methode **EndLoading** sollte am Ende des kompletten Ladevorgangs (Verbindungen und Knoten) einmal aufgerufen werden.

	Datentyp	Beschreibung
Parameter:		
⇒ linkRecord	data field/string	Datensatz der Verbindung
Rückgabewert	VcLink	Verbindung

Code-Beispiel

```
VcNet1.InsertNodeRecord "A100;Activity 1;12.09.14;17.09.14;5;Planning"
VcNet1.InsertNodeRecord "A105;Activity 5;13.09.14;18.09.14;7;Testing"
VcNet1.InsertLinkRecord "A100;A105;FS;0"
```

```
VcNet1.EndLoading
```

InsertNodeRecord

Methode von VcNet

Mit dieser Methode wird ein neuer Knoten erzeugt. Die Daten werden als CSV-String (mit Trennzeichen Semikolon) gemäß der auf der Eigenschaftenseite **Datendefinition** festgelegten Struktur übergeben. Die Methode **EndLoading** sollte am Ende des kompletten Ladevorgangs (Verbindungen und Knoten) einmal aufgerufen werden.

	Datentyp	Beschreibung
Parameter:		
⇒ nodeRecord	data field/string	Datensatz des Knotens
Rückgabewert	VcNode	Knoten

Code-Beispiel

```
' data format: "Number;Name;Start date;Finish date;Group code;Group name"
VcNet1.InsertNodeRecord "A100;Activity 1;12.09.14;17.09.14;5;Planning"
VcNet1.InsertNodeRecord "A105;Activity 5;13.09.14;18.09.14;7;Testing"
VcNet1.InsertLinkRecord "A100;A105;FS;0"
```

```
VcNet1.EndLoading
```

MakeARGB

Methode von VcNet

Mit dieser Methode können Sie aus den vier Einzelwerten einer Farbe einen ARGB-Wert bilden.

	Datentyp	Beschreibung
Parameter:		
⇒ alpha	Integer	Alpha-Wert
⇒ red	Integer	Rot-Wert
⇒ green	Integer	Grün-Wert
⇒ blue	Integer	Blau-Wert
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 = VcNet1.MakeARGB(alpha, red, green, blue)
```

Open

Methode von VcNet

Mit dieser Methode werden die Daten aus der angegebenen Datei geladen. In der Datei müssen die Daten gemäß der auf der Eigenschaftenseite **Datendefinition** festgelegten Struktur im CSV-Format (mit Semikolon als Trennzeichen) gespeichert sein. Zuerst werden Knotendaten gelesen, nach einer Zeile mit 4 Sternchen (****) werden Verbindungsdaten gelesen.

	Datentyp	Beschreibung
Parameter:		
⇒ fileName	String	Dateiname
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcNet1.Open "C:\ProjectData.net"
```

PageLayout

Methode von VcNet

Mit dieser Methode rufen Sie den Dialog **Seite einrichten** auf.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcNet1.PageLayout
```

PasteNodesFromClipboard

Methode von VcNet

Mit dieser Methode werden die im Zwischenspeicher vorhandenen Knoten und Verbindungen in das Diagramm eingefügt. Siehe auch **CopyNodesIntoClipboard** und **CutNodesIntoClipboard**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.PasteNodesFromClipboard
```

PixelsToRaster

Methode von VcNet

Mit dieser Methode konvertieren Sie die angegebenen Fensterkoordinaten, wie sie z. B. von Ereignissen zurückgegeben werden, in Bandnummern für horizontale und vertikale Richtung. Falls die Bandnummern außerhalb des Diagramms liegen, wird der Funktionswert **False** zurückgegeben. Siehe auch **RasterToPixels**.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **PixelsToRasterAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate in Pixeln
⇒ y	Long	Y-Koordinate in Pixeln

↔ xBandNo	Long	X-Koordinate in Bandnummern
↔ yBandNo	Long	Y-Koordinate in Bandnummern
Rückgabewert	Boolean	Konvertierung erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Private Sub VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
                               ByVal x As Long, ByVal y As Long, _
                               returnStatus As Variant)

    Dim lineNo As Long
    Dim columnNo As Long

    'change a data field of the node to the line number
    VcNet1.PixelsToRaster x, y, columnNo, lineNo
    node.DataField(19) = bandY
End Sub
```

PixelsToRasterAsVariant**Methode von VcNet**

Diese Methode ist bis auf die Parameter identisch mit der Methode **PixelsToRaster**. 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.

PrintDirectEx**Methode von VcNet**

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	<p>Mögliche Werte:</p> <p>vcPrintingSucceeded 0: Druck verlief erfolgreich</p> <p>vcNoPrinterInstalled 1 Es wurde weder der über VcPrinter.PrinterName angegebene noch ein im Windows-Betriebssystem als Standarddrucker gekennzeichnete Drucker gefunden.</p> <p>vcPrintingAbortedByUser 2: Der Druck wurde durch den Anwender abgebrochen.</p> <p>vcPrintingAbortedByDriver 3: Der Druck wurde durch den Windows-Druckertreiber abgebrochen.</p> <p>vcUnprintablePageLayout 4: Es konnte nicht gedruckt werden, weil das Seitenlayout zusammen mit den Druckereigenschaften wie Papiergröße und Ränder zu einem nicht druckbaren Layout führten.</p>

Code-Beispiel

```
PrintStatusResultEnum status = VcNet1.PrintDirectEx()
If status <> vcPrintingSucceeded Then
    Debug.Print "Printing failed: " & status & vbCrLf
End If
```

PrinterSetup

Methode von VcNet

Mit dieser Methode rufen Sie den Windows-Dialog **Drucker einrichten** auf.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcNet1.PrinterSetup
```

PrintIt

Methode von VcNet

Mit dieser Methode wird das Diagramm gedruckt. Dabei werden die im Dialog **Seite einrichten** verwendeten Parameter verwendet.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcNet1.PrintIt
```

PrintPreview

Methode von VcNet

Mit dieser Methode wird die Druckvorschau aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcNet1.PrintPreview
```

PrintToFile

Methode von VcNet

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
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.PrintToFile
```

RasterToPixels

Methode von VcNet

Mit dieser Methode konvertieren Sie die angegebenen Bandnummern in Fensterkoordinaten, wie sie z. B. von Ereignissen zurückgegeben werden. Falls die Koordinaten außerhalb des Diagramms liegen, wird der Funktionswert **False** zurückgegeben. Siehe auch **PixelsToRaster**.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **RasterToPixelsAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇒ xBandNo	Long	X-Koordinate in Bandnummern
⇒ yBandNo	Long	Y-Koordinate in Bandnummern
⇐ x	Long	X-Koordinate in Pixeln
⇐ y	Long	Y-Koordinate in Pixeln
Rückgabewert	Boolean	Konvertierung erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Private Sub VcNet1_OnDiagramLDb1Click(ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

' change a line number into a node data field
Dim lineNo, columnNo As Long
VcNet1.RasterToPixels columnNo, lineNo, x, y
Call MsgBox(columnNo & lineNo, vbOK)

End Sub
```

RasterToPixelsAsVariant

Methode von VcNet

Diese Methode ist bis auf die Parameter identisch mit der Methode **RasterToPixels**. 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.

Reset

Methode von VcNet

Mit dieser Methode wird je nach eingestelltem Wert von `resetAction` der komplette Inhalt sämtlicher Datentabellen gelöscht bzw. der zur Entwurfszeit auf den Eigenschaftenseiten eingestellte Zustand wiederhergestellt

	Datentyp	Beschreibung
Parameter: ⇒ <code>resetAction</code>	ResetActionEnum Mögliche Werte: <code>vcEmptyAllDataTables</code> 4 <code>vcReloadConfiguration</code> 2	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.
Rückgabewert	Boolean	Objekte im Diagramm erfolgreich gelöscht (True)

Code-Beispiel

```
VcNet1. Reset(vcReloadConfiguration) = True
```

SaveAsEx

Methode von VcNet

Mit dieser Methode werden die aktuellen Daten in die angegebene Datei im CSV-Format (Trennzeichen: Semikolon) 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: ⇒ <code>fileName</code> ⇒ <code>encoding</code>	String EncodingEnum Mögliche Werte: <code>vcANSIEncoding</code> 1	Dateiname 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.

	vcUnicodeEncoding 2	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 erfolgreich (True)/nicht erfolgreich (False) erfolgt

Code-Beispiel

```
VcNet1.SaveAs "C:\ProjectData.net"
```

ScheduleProject**Methode von VcNet**

Mit dieser Methode können Sie eine Vorwärts- und Rückwärtsberechnung des aktuellen Projekts durchführen. Bei Übergabe des Starttermins wird zunächst eine Vorwärtsberechnung, dann eine Rückwärtsberechnung durchgeführt. Bei Übergabe des Endtermins wird zunächst eine Rückwärtsberechnung, dann eine Vorwärtsberechnung durchgeführt. Es können auch Anfangs- und Enddatum übergeben werden, die Vorgänge erhalten dann entsprechende Pufferzeiten. Das Fehlen beider Daten führt zu einer Fehlermeldung. 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 oder 0
⇒ endDate	Date/Time	Enddatum oder 0
Rückgabewert	Boolean	Berechnung erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
VcNet1.ScheduleProject "21.06.14", 0
```

ScrollToNodePosition**Methode von VcNet**

Mit dieser Methode können Sie zu der Zeile eines bestimmten Knotens scrollen, so dass dieser sichtbar wird.

	Datentyp	Beschreibung
Parameter: ⇒ node	VcNode	Knoten, zu dessen Zeile gescrollt werden soll
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Private Sub VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
                               ByVal x As Long, ByVal y As Long, _
                               returnStatus As Variant)
    'scroll the diagram so that the node is completely on screen
    VcNet1.ScrollToNodePosition node
End Sub
```

ShowAlwaysCompleteView**Methode von VcNet**

Mit dieser Methode können Sie die Darstellung so einstellen, dass stets das komplette Diagramm angezeigt wird. Der Zoomfaktor paßt sich bei jeder Änderung des Charts automatisch an. Der maximale Zoomfaktor von 100% wird nicht überschritten, die Knoten werden also höchstens in Originalgröße dargestellt. Siehe auch Eigenschaft **ZoomFactor** und Methode **Zoom**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.ShowAlwaysCompleteView
```

ShowExportGraphicsDialog**Methode von VcNet**

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

```
VcNet1.ShowExportGraphicsDialog
```

SuspendUpdate**Methode von VcNet**

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
Rückgabewert	Boolean	SuspendUpdate(True): Beginn der SuspendUpdate Methode/ SuspendUpdate(False): Ende der SuspendUpdate Methode

Code-Beispiel

```
VcNet1.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

```

```
VcNet1.SuspendUpdate (False)
```

UpdateLinkRecord

Methode von VcNet

Mit dieser Methode können die Daten eines bestehenden Datensatzes einer Verbindung verändert werden. Der Datensatz wird durch die auf der Eigenschaftenseite **Datendefinition** festgelegte ID identifiziert. Diese Methode wird verwendet, wenn externe Änderungen in der Grafik auf dem Bildschirm nachvollzogen werden sollen.

	Datentyp	Beschreibung
Parameter: ⇒ linkRecord	Variant	Datensatz der Verbindung
Rückgabewert	VcLink	Aktualisierte Verbindung

Code-Beispiel

```
VcNet1.UpdateLinkRecord "A100;A105;FS;0"
```

UpdateNodeRecord

Methode von VcNet

Mit dieser Methode können die Daten eines bestehenden Datensatzes eines Knotens verändert werden. Der Datensatz wird durch die auf der Eigenschaftenseite **Datendefinition** festgelegte ID identifiziert. Diese Methode wird verwendet, wenn externe Änderungen in der Grafik auf dem Bildschirm nachvollzogen werden sollen.

	Datentyp	Beschreibung
Parameter: ⇒ nodeRecord	Variant	Datensatz des Knotens
Rückgabewert	VcNode	Aktualisierter Knoten

Code-Beispiel

```
VcNet1.UpdateNodeRecord "A100;Activity 1;12.09.14;18.09.14;6;Planning"
```

Zoom

Methode von VcNet

Mit dieser Methode kann die Bildschirmdarstellung um den angegebenen Prozent-Faktor vergrößert (`zoomFactor > 100`) oder verkleinert (`zoomFactor < 100`) werden.

	Datentyp	Beschreibung
Parameter: ⇒ zoomFactor	Integer	Zoomfaktor {11...999}, andere Werte bleiben unberücksichtigt
Rückgabewert	Boolean	Zoomen erfolgreich durchgeführt {True}

Code-Beispiel

```
VcNet1.Zoom 120
```

ZoomOnMarkedNodes

Methode von VcNet

Mit dieser Methode können Sie auf die markierten Knoten zoomen.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcNet1.ZoomOnMarkedNodes
```

Ereignisse**Error**

Ereignis von VcNet

Dieses Ereignis tritt nur dann auf, wenn ein unvorhergesehener Fehler im Code des VARCHART XNet 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

⇒ Scode	Long	&h800a402f (konstant)
⇒ Source	String	Name des Controls (konstant)
⇒ HelpFile	String	Hilfedatei: "" (konstant)
⇒ 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.

Code-Beispiel

```
Private Sub VcNet1_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 VcNet

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 VcNet

Dieses Ereignis tritt ein, wenn der Anwender eine Taste drückt, während VARCHART XNet 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)

⇒ 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.
---------	---------	---

Code-Beispiel

```
Private Sub VcNet1_KeyDown(KeyCode As Integer, Shift As Integer)
    MsgBox "key pressed"
End Sub
```

KeyPress**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn der Anwender eine ANSI-Taste drückt und wieder loslässt, während VARCHART XNet 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 Tastenschlag auf, d.h. das Objekt erhält kein Zeichen.

Code-Beispiel

```
Private Sub VcNet1_KeyPress(KeyAscii As Integer)
    MsgBox "Key pressed and released."
End Sub
```

KeyUp**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn der Anwender eine Taste loslässt, während der Fokus auf VARCHART XNet ist. 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)
⇒ 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.

Code-Beispiel

```
Private Sub VcNet1_KeyUp(KeyCode As Integer, Shift As Integer)
    MsgBox "key released"
End Sub
```

OLECompleteDrag**Ereignis von VcNet**

Dieses Ereignis tritt auf, wenn ein OLE Drag & Drop-Vorgang beendet wurde und die Quellkomponente darüber informiert wird, dass eine Drag-Aktion entweder erfolgreich durchgeführt oder abgebrochen wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ effect	vcOLEDropEffectEnum	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet. Effect ist zunächst ein Wert, der die von der Drag-Quelle unterstützten OLE-Drag & Drop-Operationen bezeichnet. Die Zielkomponente sollte diesen Wert prüfen und beim Ablegevorgang festlegen.

OLEDragDrop**Ereignis von VcNet**

Dieses Ereignis tritt auf, wenn bei einem OLE Drag & Drop-Vorgang Daten auf dem Drop-Ziel abgelegt werden und die **OLEDropMode**-Eigenschaft des Drop-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	Das OLE Drag & Drop-DataObject, in welchem die Daten importiert werden
⇒ effect	vcOLEDropEffectEnum	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet.
⇒ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: 1 (links), 2 (rechts) oder 4 (Mitte).
⇒ 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 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".
⇒ y	Long	Y-Koordinate des Mauszeigers
⇒ x	Long	X-Koordinate des Mauszeigers

OLEDragOver

Ereignis von VcNet

Dieses Ereignis tritt auf, wenn Daten über ein Drop-Ziel gezogen und die **OLEDropMode**-Eigenschaft des Drop-Ziels auf **vcOLEDropManual** gesetzt ist.

	Datentyp	Beschreibung
Parameter:		
⇒ data	DataObject	Das OLE-Drag & Drop-DataObject, in das die Daten importiert werden
⇔ effect	OLEDropEffectEnum	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet.
	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	Enthält eine Zahl, die angibt, welche Maustaste gedrückt ist: 1 (links), 2 (rechts) oder 4 (Mitte).
⇒ shift	Integer	Enthält eine Zahl, die den Zustand der Modifizierungstasten. 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 sowohl die STRG- als auch die ALT-Tasten gedrückt würden, wäre der Wert von Shift 6.
⇒ x	Long	Enthält die horizontale Position des Mauszeigers
⇒ y	Long	Enthält die vertikale Position des Mauszeigers
⇒ state	OLEDragStateEnum	Enthält eine Konstante, die die Richtung angibt, in die die Daten gezogen werden.
	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 VcNet

Dieses Ereignis tritt nach jedem OLEDragOver-Ereignis auf der Drop-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	vcOLEDropEffectEnum	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet. Effect ist anfangs ein Wert, der die von der Drag-Quelle unterstützten OLE-Drag & Drop-Operationen bezeichnet. Die Zielkomponente sollte diesen Wert prüfen und beim Ablegevorgang festlegen.
⇐ defaultCursors	Boolean	Verwendung des Standard-Cursors (True) oder eines benutzerdefinierten Cursors (False)

Code-Beispiel

```

Private Sub VcTree1_OLEGiveFeedback(ByVal Effect As Long, _
                                   DefaultCursors As Boolean)
    If Effect <> vcOLEDropEffectNone Then
        'activate own mouse cursor
        MousePointer = vbCustom
        MouseIcon = LoadPicture("h_point.cur")
        DefaultCursors = False
    End If
End Sub

```

OLESetData**Ereignis von VcNet**

Dieses Ereignis tritt bei einer Drag-Quelle auf, wenn ein Drop-Ziel die **GetData**-Methode aufruft und es in dem OLE Drag & Drop-DataObject keine Daten in einem festgelegten Format gibt.

	Datentyp	Beschreibung
Parameter:		
⇒ data	DataObject	DataObject, das bei der SetData-Methode verwendet wird, um die Daten dort abzulegen. Die Komponente ruft die SetData-Methode auf, um das entsprechende Format zu laden.
⇒ dataFormat	Integer	Ein numerischer oder Zeichenwert, der das Format der Daten bezeichnet, die die GetData-Methode verlangt. Die Drag-Quelle verwendet diesen Wert zur Bestimmung des Formats der im DataObject unterzubringenden Daten. Eine Tabelle, die die numerischen oder Zeichenwerte für das jeweilige Datenformat zusammen mit einer Beschreibung des Datenformats anzeigt, finden Sie unter GetData -Methode.

OLEStartDrag**Ereignis von VcNet**

Dieses Ereignis tritt auf, wenn die Methode **OLEDrag** ausgeführt wird oder wenn eine VARCHART-XNet-Komponente eine OLE Drag & Drop-Operation auslöst und die **OLEDragMode**-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 eventuell 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	Objekt vom Typ DataObject , das Formate, die von der Quelle bereitgestellt werden, und möglicherweise die Daten für diese Formate enthält. Wenn das Objekt keine Daten enthält, werden sie aus der Quellkomponente geladen, wenn das Steuerelement die GetData-Methode aufruft.
⇔ allowedEffect	vcOLEDropEffectEnum	Eine lange Integerzahl, die die von der Quellkomponente unterstützten Effekte enthält. Die Werte für diesen Parameter in dem Ereignis sollten vom Programmierer bereitgestellt werden.

Code-Beispiel

```
Private Sub VcTree1_OLEStartDrag(ByVal data As VcNetLib.DataObject, _
                                allowedEffects As Long)
    allowedEffects = vbDropEffectCopy

    ' make sure that dragging is allowed only from one XTree control
    ' into another one
    data.SetData Empty, myOLEDragFormat
End Sub
```

OnBoxLClick

Ereignis von VcNet

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 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 VcNet1_OnBoxLClick(ByVal box As VcNetLib.VcBox, _
    ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    Text1.Text = box.FieldText(1)

End Sub
```

OnBoxLDbIcIck**Ereignis von VcNet**

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 VcNet1_OnBoxLDbIcIck(ByVal box As VcNetLib.VcBox, _
    ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    box.FieldText(0) = Text1.Text

End Sub
```

OnBoxModifyComplete**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn die Modifizierung der Box abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇒ box	VcBox	Veränderte Box

Code-Beispiel

```
Private Sub VcNet1_OnBoxModifyComplete(ByVal box As _
    VcNetLib.VcBox)

    MsgBox "The box has been modified."

End Sub
```

OnBoxModifyCompleteEx**Ereignis von VcNet**

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 Mögliche Werte: vcBMTAnything 1 vcBMTNothing 0 vcBMTTextModified 4 vcBMTXYOffsetModified 2	Art der Veränderung beliebige Veränderung keine Veränderung Text der Box verändert Offset verändert

Code-Beispiel

```
Private Sub VcNet1_OnBoxModifyCompleteEx(ByVal box As _
    VcNetLib.VcBox)

    MsgBox "The box has been modified."

End Sub
```

OnBoxRClick**Ereignis von VcNet**

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. Durch das Setzen des Rückgabestatus können Sie an der übergebenen Position ein eignes Kontextmenü anzeigen.

	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 VcNet1_OnBoxRClick(ByVal box As VcNetLib.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
```

OnDataRecordCreate**Ereignis von VcNet**

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 VcNet1_OnDataRecordCreate(ByVal node As VcNetLib.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 VcNet

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
	vcNodesAndLinksCloned 4	selektierte Knoten wurden durch Ziehen mit der Maus bei gleichzeitigem Drücken der Strg-Taste kopiert
	vcNodeWithLinkCreated 3	Knoten zusammen mit Verbindung 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**Code-Beispiel**

```
Private Sub VcNet1_OnDataRecordCreateComplete(ByVal dataRecord As _
                                           VcNetLib.VcDataRecord, ByVal creationType As _
                                           VcNetLib.CreationTypeEnum, _
                                           ByVal isLastDataRecordInSeries As Boolean)
    addDataRecord dataRecord.AllData
End Sub
```

OnDataRecordDelete**Ereignis von VcNet**

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 VcNet1_OnDataRecordDelete(ByVal node As VcNetLib.VcNode, _
                                     returnStatus As Variant)
    'deny the deletion of the last data record in the chart
    If VcNet1.DataRecordCollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last data record cannot be deleted.")
    End If
End Sub
```

OnDataRecordDeleteComplete**Ereignis von VcNet**

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

OnDataRecordModify

Ereignis von VcNet

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 vcMoved 8 vcNothing 0	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Objekt wurde verschoben keine Änderung
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0	Die Veränderung wird rückgängig gemacht.

vcRetStatOK 1	Die Veränderung wird durchgeführt.
---------------	------------------------------------

OnDataRecordModifyComplete

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn die Modifizierung des Datensatzes abgeschlossen ist.

	Datentyp	Beschreibung
Parameter: ⇒ dataRecord	VcDataRecord	Veränderter Datensatz

Code-Beispiel

```
Private Sub VcNet1_OnDataRecordModifyComplete(ByVal box As _
    VcNetLib.VcBox)

    MsgBox "The data record has been modified."

End Sub
```

OnDataRecordNotFound

Ereignis von VcNet

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

OnDiagramLClick

Ereignis von VcNet

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 VcNet1_OnDiagramLClick(ByVal x As Long, _
                                   ByVal y As Long, returnStatus As Variant)
    Dim zoomfactor As Integer

    zoomfactor = VcNet1.Zoomfactor + 10
    VcNet1.Zoomfactor = zoomfactor
End Sub
```

OnDiagramLDbIcIck

Ereignis von VcNet

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 VcNet1_OnDiagramLDbIcIck(ByVal x As Long, _
                                       ByVal y As Long, returnStatus As Variant)
    Dim zoomfactor As Integer

    zoomfactor = VcNet1.Zoomfactor - 10
    VcNet1.Zoomfactor = zoomfactor
End Sub
```

OnDiagramRClick

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste im Diagrammbereich klickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten) wird als Parameter übergeben. Durch das Setzen des Rückgabestatus können Sie an der übergebenen Position ein eignes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnDiagramRClick(ByVal x As Long, ByVal y As Long, _
    returnStatus As Variant)

    'start a popup menu at the current mouse cursor position
    PopupMenu mnuDiagramPopup

    'revoke the VARCHART XNet context menu
    returnStatus = vcRetStatNoPopup

End Sub
```

OnGiveFeedbackForNodeCreating

Ereignis von VcNet

Dieses Ereignis wird aufgerufen, wenn der Knotenerzeugemodus eingeschaltet ist. X und Y bezeichnen die Position des Mauszeigers relativ zum Koordinatenursprung des Steuerelements. Wird der Defaultwert **1** von **creationAllowed** nicht geändert, können Knoten an dieser Mausposition erzeugt werden. Wenn **creationAllowed** auf **0** gesetzt wird, ist das Erzeugen von Knoten nicht möglich. Dies kann genutzt werden, um eine Knotenerzeugung in bestimmten Teilen des Diagramms (z.B. in Bereichen ohne Gruppen wie im Codebeispiel unten) von vornherein zu unterbinden.

	Datentyp	Beschreibung
Parameter:		
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ creationAllowed	Long	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGiveFeedbackForNodeCreating(ByVal X As Long, ByVal Y As
Long, creationAllowed As Long)
    Dim obj As Object
    Dim objType As VcObjectTypeEnum
    VcNet1.IdentifyObjectAt X, Y, obj, objType
    If objType = vcObjTypeNone Then
        creationAllowed = False
    End If
End Sub
```

OnGroupCreate

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn eine Gruppe erzeugt wird, also wenn der erste Knoten mit einem neuen Gruppencode im ActiveX-Steuerelement erzeugt wird. Das neu erzeugte Objekt wird als Parameter übergeben, so dass eine Validierung und ggf. ein Datenbank-Eintrag vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Angelegte Gruppe
⇔ returnStatus	Variant	Rückgabestatus: z.Z. ohne Funktion

OnGroupDelete

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender interaktiv den letzten Knoten einer Gruppe löscht oder verschiebt, so dass diese leer wird und daher gelöscht wird. Dabei wird das Gruppenobjekt als Parameter übergeben. Derzeit kann das Löschen einer Gruppe nicht durch Setzen des Rückgabestatus verhindert werden.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGroupDelete(ByVal group As VcNetLib.VcGroup, _
                                returnStatus As Variant)
    MsgBox ("The last node of the group is deleted.")
End Sub
```

OnGroupLClick

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender auf eine Gruppe mit der linken Maustaste klickt. Das getroffene Group-Objekt wird zusammen mit der Position (x,y-Koordinaten) als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGroupLClick(ByVal group As VcNetLib.VcGroup, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)
    'change the color of the group to a light yellow
    group.BackColor = &HC0FFFF
End Sub
```

OnGroupLDbIcClick**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn der Anwender auf eine Gruppe mit der linken Maustaste doppelklickt. Das getroffene Group-Objekt wird zusammen mit der Position (x,y-Koordinaten) als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGroupLDbIcClick(ByVal group As VcNetLib.VcGroup, _
                                       ByVal x As Long, ByVal y As Long, _
                                       returnStatus As Variant)
    MsgBox group.Name
End Sub
```

OnGroupModify**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn eine Gruppe im Modus Clusterung interaktiv kollabiert (modificationType = vcGMTCollapsing) oder expandiert (vcGMTExpanding) wird. Das getroffene Group-Objekt, die Art der Veränderung und der Rückgabestatus werden als Parameter mitgegeben.

Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Aktion abgebrochen.

	Datentyp	Beschreibung
Parameter:		
⇒ Group	VcGroup	Veränderte Gruppe
⇒ modificationType	GroupModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcGMTCollapsing 2 vcGMTExpanding 4 vcGMTMoved 32	Gruppe kollabiert Gruppe expandiert Objekt wurde verschoben
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGroupModify(ByVal Group As VcNetLib.VcGroup, _
                                ByVal modificationType As _
                                VcNetLib.GroupModificationTypeEnum, _
                                returnStatus As Variant)

    Select Case modificationType
        Case vcGMTCollapsing
            MsgBox "Group is collapsed."
        Case vcGMTExpanding
            MsgBox "Group is expanded."
    End Select

End Sub
```

OnGroupModifyComplete

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn das interaktive Kollabieren oder Expandieren einer Gruppe im Modus Clusterung abgeschlossen wurde.

	Datentyp	Beschreibung
Parameter:		
⇒ Group	VcGroup	Veränderte Gruppe
⇒ modificationType	GroupModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcGMTCollapsing 2 vcGMTExpanding 4 vcGMTMoved 32	Gruppe kollabiert Gruppe expandiert Objekt wurde verschoben

Code-Beispiel

```
Private Sub VcNet1_OnGroupModifyComplete(ByVal group As VcNetLib.VcGroup, _
                                          ByVal modificationType As _
                                          VcNetLib.GroupModificationTypeEnum)
```

```
MsgBox "The group has been modified successfully."
```

```
End Sub
```

OnGroupRClick

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender auf eine Gruppe mit der rechten Maustaste klickt. Das getroffene Gruppenobjekt wird zusammen mit der Position (x,y-Koordinaten) als Parameter übergeben. Durch das Setzen des Rückgabestatus können Sie an der übergebenen Position ein eignes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇒ group	VcGroup	Getroffene Gruppe
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
↔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnGroupRClick(ByVal group As VcNetLib.VcGroup, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    ' start a popup menu at the current mouse cursor position
    PopupMenu mnuGroupPopup

    returnStatus = vcRetStatNoPopup

End Sub
```

OnHelpRequested

Ereignis von VcNet

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

vcPageSetupDialog 4097	Hilfe wurde für den Seite einrichten Dialog angefordert
vcPrintPreviewDialog 4096	Hilfe wurde für den Druckvorschau Dialog angefordert

OnLegendViewClosed

Ereignis von VcNet

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Legendenansicht geschlossen wird.

	Datentyp	Beschreibung
Parameter: ↔ (no parameter)		

Code-Beispiel

```
Private Sub VcNet1_OnLegendViewClosed()
    MsgBox "Do you want to close the legend view window?", vbOKCancel
End Sub
```

OnLinkCreate

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Verbindung zwischen zwei Knoten erzeugt hat. Das neu erzeugte Objekt wird als Parameter übergeben, so dass eine Validierung und ggf. ein Datenbank-Eintrag vorgenommen werden kann. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Verbindung wieder gelöscht.

Dieses Ereignis sollte nur verwendet werden, um Daten der aktuellen Verbindung auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnLinkCreateComplete**.

	Datentyp	Beschreibung
Parameter: ⇒ link	VcLink	Angelegte Verbindung
↔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnLinkCreate(ByVal link As VcNetLib.VcLink, _
    returnStatus As Variant)
    'show own edit dialog for the new link
    ' (EditNewLinks attribute must be set to off!)
```



```

On Error GoTo CancelError
frmEditLinkDialog.setLink link
frmEditLinkDialog.Show Modal, Me
Exit Sub

CancelError:
returnStatus = vcRetStatFalse
End Sub

```

OnLinkCreateComplete

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn das interaktive Anlegen einer Verbindung zwischen zwei Knoten beendet ist. Das Verbindungs-Objekt, der Typ des Anlegens der Verbindung und die Information, ob die angelegte Verbindung die einzige Verbindung oder die letzte Verbindung einer Menge ist, werden als Parameter übergeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	angelegte Verbindung
⇒ creationType	CreationTypeEnum	Typ des Knotenanlegens
	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
	vcNodesAndLinksCloned 4	selektierte Knoten wurden durch Ziehen mit der Maus bei gleichzeitigem Drücken der Strg-Taste kopiert
	vcNodeWithLinkCreated 3	Knoten zusammen mit Verbindung angelegt
⇒ isLastLinkInSeries	Boolean	angelegte Verbindung ist/ist nicht die einzige bzw. die letzte Verbindung einer Menge

Code-Beispiel

```

Private Sub VcNet1_OnLinkCreateComplete(ByVal link As VcNetLib.VcLink, _
                                        ByVal creationType As VcNetLib.CreationTypeEnum, _
                                        ByVal isLastLinkInSeries As Boolean)

    'create a record in the underlying database of the application
    addLinkRecordToDatabase link.AllData

End Sub

```

OnLinkDelete

Ereignis von VcNet

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. verhindern können. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Löschvorgang verhindert; durch Setzen auf **vcRetStatOK** wird die Löschung durchgeführt; bei **vcRetStatDefault** bleibt das vordefinierte Default-Verhalten unverändert und die Löschung wird ebenfalls durchgeführt, und bei **vcRetStatPopup** wird das Kontext-Menü aufgerufen, über das Sie dem Benutzer weitere Interaktionen anbieten können.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	Gelöschte Verbindung
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnLinkDelete(ByVal link As VcNetLib.VcLink, _
                               returnStatus As Variant)
    'deny deletion of link with a certain predecessor
    If link.PredecessorNode.DataField(3) = "X" Then
        returnStatus = vcRetStatFalse
    End If
End Sub
```

OnLinkDeleteComplete

Ereignis von VcNet

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.

OnLinkLClickCltn

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Verbindung oder mehrere sich überlagernde Verbindungen klickt. Ein LinkCollection-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCltn	VcLinkCollection	getroffenes LinkCollection-Objekt
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnLinkLClickCltn(ByVal linkCltn As _
                                   VcNetLib.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
```

OnLinkLDbClickCltn

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Verbindung oder mehrere sich überlagernde Verbindungen doppelt klickt. Ein LinkCollection-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCltn	VcLinkCollection	getroffenes LinkCollection-Objekt
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

Private Sub VcNet1_OnLinkLDb1ClickCltn(ByVal linkCltn As _
                                         VcNetLib.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

CancelError:
    'deny the "Edit Link Data" dialog
    returnStatus = vcRetStatFalse
End Sub

```

OnLinkModifyComplete**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn die Modifizierung der angegebenen Verbindung abgeschlossen ist.

Das Knoten-Objekt 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:		
⇒ link	VcLink	angelegte Verbindung
⇒ isLastLinkInSeries	Boolean	angelegte Verbindung ist/ist nicht die einzige bzw. die letzte Verbindung einer Menge

Code-Beispiel

```

Private Sub VcNet1_OnLinkModifyComplete(ByVal link As _
                                         VcNetLib.VcLink, ByVal isLastLinkInSeries _
                                         As Boolean)
    ' modify a record in the underlying database of the application
    modifyDataRecord link.AllData
End Sub

```

OnLinkModifyEx**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn der Anwender interaktiv eine Verbindung verändert hat. Dabei kann die Verbindung verschoben oder ein Wert im Dialog **Daten bearbeiten** verändert worden sein. Wenn Sie den

Rückgabestatus auf **vcRetStatFalse** setzen, wird die Veränderung rückgängig gemacht.

Dieses Ereignis sollte nur verwendet werden, um Daten der aktuellen Verbindung auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnLinkModifyComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ link	VcLink	Verbindung nach der Veränderung
⇒ oldlink	VcLink	Verbindung vor der Veränderung
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnLinkModify(ByVal link As VcNetLib.VcLink, _
                               returnStatus As Variant)
    'deny any modification
    returnStatus = vcRetStatFalse
End Sub
```

OnLinkRClickCltn

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Verbindung oder mehrere sich überlagernde Verbindungen klickt. Ein LinkCollection-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben. 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 LinkCollection-Objekt
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnLinkRClickCltn(ByVal linkCltn As _
                                     VcNetLib.VcLinkCollection, _
                                     ByVal x As Long, ByVal y As Long, _
                                     returnStatus As Variant)
    ' start a popup menu at the current mouse cursor position
    PopupMenu mnuLinkPopup
```

```

    ' revoke the VARCHART XNet context menu
    returnStatus = vcRetStatNoPopup
End Sub

```

OnLinksMark

Ereignis von VcNet

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer Verbindungen zum Markieren oder Demarkieren ausgewählt hat. Die Parameter **button** und **shift** geben an, welche Steuer- und Maustasten gedrückt wurden. Wenn man den Rückgabestatus auf **vcRetStatFalse** setzt, wird das Markieren bzw. die Aufhebung der Markierung verhindert.

Dieses Ereignis sollte nur verwendet werden, um Daten der aktuellen Verbindung auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnLinksMarkComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ linkCollection	VcLinkCollection	Verbindungsauflistung (LinkCollection), die die vom Anwender selektierten Verbindungen enthält. Wenn in das Diagramm geklickt wurde, ist die Auflistung leer.
⇒ button	Integer	Zahl, die angibt, auf welche Weise markiert wurde: 0 : über die Tastatur, 1 : linke Maustaste, 2 : rechte Maustaste, 4 : mittlere Maustaste
⇒ 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".
↔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

Private Sub VcNet1_OnLinkMark()
    If MsgBox("Mark this link?", vbYesNo, "Marking links") = _
        vbNo Then returnStatus = vcRetStatFalse
End Sub

```

OnLinksMarkComplete

Ereignis von VcNet

Mit diesem Ereignis wird das Ende einer Markier- oder Demarkieroperation von Verbindungen angezeigt.

	Datentyp	Beschreibung
Parameter: ↵ (no parameter)		Kein Parameter

Code-Beispiel

```
Private Sub VcNet1_OnLinkMarkComplete()
    MsgBox "Links have been successfully marked."
End Sub
```

OnModifyComplete

Ereignis von VcNet

Dieses Ereignis tritt immer dann auf, wenn Daten interaktiv im Chart verändert werden, also explizit nach den folgenden Ereignissen:

- OnBoxModifyComplete
- 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

OnMouseDownClick

Ereignis von VcNet

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).
⇒ 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.
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseDown

Ereignis von VcNet

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).

⇒ 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.
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseMove

Ereignis von VcNet

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).
⇒ 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.
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnMouseUp

Ereignis von VcNet

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).
⇒ 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.
⇒ x	Long	X-Koordinate des Mauszeigers
⇒ y	Long	Y-Koordinate des Mauszeigers

OnNodeCreate

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten erzeugt hat. Das Knotenobjekt wird als Parameter übergeben, so dass eine Datenvalidierung vorgenommen werden kann. Dies kann wichtig sein, wenn der Benutzer bei aktiviertem Dialog **Daten bearbeiten** eigene Daten eingegeben hat. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten wieder gelöscht.

Dieses Ereignis sollte nur verwendet werden, um Daten des aktuellen Knotens auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeCreateCompleteEx**.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	anzulegender Knoten
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeCreate(ByVal node As VcNetLib.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 VcNet

Dieses Ereignis tritt ein, wenn das interaktive Anlegen eines Knotens abgeschlossen ist. Das Knotenobjekt, der Typ des Knotenanlegens und die Information, ob der angelegte Knoten der einzige Knoten oder der letzte Knoten einer Menge ist, werden als Parameter übergeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	angelegter Knoten
⇒ creationType	CreationTypeEnum	Typ des Knotenanlegens
	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
	vcNodesAndLinksCloned 4	selektierte Knoten wurden durch Ziehen mit der Maus bei gleichzeitigem Drücken der Strg-Taste kopiert
	vcNodeWithLinkCreated 3	Knoten zusammen mit Verbindung angelegt
⇒ isLastNodeInSeries	Boolean	angelegter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge

Code-Beispiel

```

Private Sub VcNet1_OnNodeCreateCompleteEx(ByVal node As _
    VcNetLib.VcNode, ByVal creationType As _
    VcNetLib.CreationTypeEnum, _
    ByVal isLastNodeInSeries As Boolean)
'create a record in the underlying database of the application
addDataRecord node.AllData
End Sub

```

OnNodeDelete

Ereignis von VcNet

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. verhindern können. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Löschvorgang verhindert; durch Setzen auf **vcRetStatOK** wird die Löschung durchgeführt; bei **vcRetStatDefault** bleibt das vordefinierte Default-Verhalten unverändert und die Löschung wird ebenfalls durchgeführt, und bei **vcRetStatPopup** wird das Kontext-Menü aufgerufen, über das Sie dem Benutzer weitere Interaktionen anbieten können.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knotenobjekt
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeDelete(ByVal node As VcNetLib.VcNode, _
                               returnStatus As Variant)
    ' deny the deletion of the last node in the chart
    If VcNet1.NodeCollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last node in the chart cannot be deleted.")
    End If
End Sub
```

OnNodeDeleteCompleteEx

Ereignis von VcNet

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

OnNodeLClick

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der linken Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) und dem Diagrammbereich übergeben.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knotenobjekt
⇒ location	LocationEnum	Lokalisierung im Chart
	Mögliche Werte: vcInDiagram 1	im Knotenbereich
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeLClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
                               ByVal x As Long, ByVal y As Long, _
                               returnStatus As Variant)
    'change data field of the node
    node.DataField(10) = 1 - CInt(node.DataField(10))
End Sub
```

OnNodeLDbIClick

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der linken Maustaste doppelt klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers und dem Diagrammbereichsparameter übergeben. Nach der Rückkehr wird der Dialog **Vorgänge bearbeiten** für diesen Knoten aufgerufen. Wenn Sie den Rückgabestatus auf **vcRetStatFalse** setzen, wird dieser Aufruf unterdrückt.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knotenobjekt
⇒ location	LocationEnum	Lokalisierung im Chart
	Mögliche Werte: vcInDiagram 1	im Knotenbereich
⇒ x	Long	X-Koordinate

⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeLDbClick(ByVal node As VcNetLib.VcNode, _
                                ByVal location As VcNetLib.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 VcNet**

Dieses Ereignis tritt ein, wenn die Modifizierung des angegebenen Knotens abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	angelegter Knoten
⇒ isLastNodeInSeries	Boolean	angelegter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge

Code-Beispiel

```
Private Sub VcNet1_OnNodeModifyComplete(ByVal node As VcNetLib.VcNode, ByVal
isLastNodeInSeries As Boolean)
    'modify a record in the underlying database of the application
    modifyDataRecord node.AllData
End Sub
```

OnNodeModifyCompleteEx**Ereignis von VcNet**

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer die Knotenhierarchie geändert hat.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	veränderter Knoten
⇒ isLastNodeInSeries	Boolean	veränderter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge
⇒ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcAnything 1 vcChangedGroup 16 vcMoved 8 vcNothing 0	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Objekt wurde verschoben keine Änderung

OnNodeModifyEx

Ereignis von VcNet

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten verändert. Dabei kann der Knoten 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
	Mögliche Werte: vcAnything 1 vcChangedGroup 16 vcMoved 8 vcNothing 0	Änderungstyp nicht näher bestimmt Zuordnung des Knotens zu einer Gruppe verändert Objekt wurde verschoben keine Änderung
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeModifyEx(ByVal oldNode As _
                                VcNetLib.VcNode, ByVal node As _
```

```

VcNetLib.VcNode, ByVal modificationType As _
VcNetLib.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 VcNet

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der rechten Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

Dieses Ereignis sollte nur verwendet werden, um Daten des aktuellen Knotens auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodesMarkComplete**.

	Datentyp	Beschreibung
Parameter:		
⇒ node	VcNode	Knotenobjekt
⇒ location	LocationEnum	Lokalisierung im Chart
	Mögliche Werte: vclnDiagram 1	im Knotenbereich
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```

Private Sub VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
                                ByVal location As VcNetLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    ' start a popup menu at the current mouse cursor position
    PopupMenu mnuNodePopup

    returnStatus = vcRetStatNoPopup

End Sub

```


OnNodesMarkComplete

Ereignis von VcNet

Mit diesem Ereignis wird das Ende einer Markier- oder Demarkieroperation von Knoten angezeigt.

	Datentyp	Beschreibung
Parameter: ⇨ (no parameter)		Kein Parameter

Code-Beispiel

```
Private Sub VcNet1_OnNodesMarkComplete()  
    MsgBox "Nodes have been successfully marked."  
End Sub
```

OnNodesMarkEx

Ereignis von VcNet

Mit diesem Ereignis wird bekannt gegeben, dass der Benutzer einen oder mehrere Knoten zum Markieren oder zum Aufheben der Markierung ausgewählt hat. In der Knotenaufistung (nodeCollection) sind diese Knoten verzeichnet. Die Parameter **button** und **shift** geben an, welche Steuer- und Maustasten gedrückt wurden. Wenn man den Rückgabestatus auf **vcRetStatFalse** setzt, wird das Markierung oder die Aufhebung der Markierung verhindert.

	Datentyp	Beschreibung
Parameter: ⇨ nodeCollection	VcNodeCollection	NodeCollection, die die vom Anwender selektierten Knoten enthält. Wenn in das Diagramm geklickt wurde, ist die Collection leer.
⇨ button	Integer	Zahl, die angibt, auf welche Weise markiert wurde: 0: über die Tastatur, 1: linke Maustaste, 2: rechte Maustaste, 4: mittlere Maustaste
⇨ 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".
⇨ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodesMarkEx (ByVal NodeCollection As _
    VcNetLib.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
```

OnSelectField**Ereignis von VcNet**

Dieses Ereignis tritt ein, wenn das Feld einer Box selektiert wird. Die Selektion kann durch setzen des Rückgabestatus verhindert werden.

	Datentyp	Beschreibung
Parameter:		
editObject	Object in	
editObjectType	VcObjectTypeEnum in	
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeGroup 7 vcObjTypeLinkCollection 3 vcObjTypeNode 2 vcObjTypeNodeInLegend 17 vcObjTypeNone 0	Objekttyp Box Objekttyp Gruppe Objekttyp LinkCollection Objekttyp Knoten Objekttyp Knoten im Legendbereich kein Objekt
fieldIndex	Long in	
objRectComplete	VcRect in	
objRectVisible	VcRect in	
fldRectComplete	VcRect in	
fldRectVisible	VcRect in	
returnStatus	Variant	

OnShowInPlaceEditor**Ereignis von VcNet**

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	Objekttyp Box
	vcObjTypeGroup 7	Objekttyp Gruppe
	vcObjTypeLinkCollection 3	Objekttyp LinkCollection
	vcObjTypeNode 2	Objekttyp Knoten
	vcObjTypeNodeInLegend 17	Objekttyp Knoten im Legendebereich
	vcObjTypeNone 0	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	

Code-Beispiel

```
Private Sub VcNet1_OnShowInPlaceEditor(ByVal editObject As Object, _
    ByVal editObjectType As VcNetLib.VcObjectTypeEnum, _
    ByVal fieldIndex As Long, ByVal objRectComplete As _
    VcNetLib.VcRect, ByVal objRectVisible As _
    VcNetLib.VcRect, ByVal fldRectComplete As _
    VcNetLib.VcRect, ByVal fldRectVisible As _
    VcNetLib.VcRect, returnStatus As Variant)

    Dim oldScaleMode As Long

    If editObjectType = vcObjTypeNode 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 + VcNet1.Left
                Text1.Top = fldRectVisible.Top + VcNet1.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 + VcNet1.Left
                MonthView1.Top = fldRectVisible.Top + VcNet1.Top
                MonthView1.Value = editObject.DataField(fieldIndex)
        End Select
    End If
End Sub
```

```

        MonthView1.Visible = True
        MonthView1.SetFocus

    Case 13 'Employee
        Combo1.Left = fldRectVisible.Left + VcNet1.Left
        Combo1.Top = fldRectVisible.Top + VcNet1.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 VcNet

Dieses Ereignis teilt Information über einen mit dem Mauscursor berührten Knoten mit. Sie können das Ereignis verwenden, um diese Information z. B. in einer Statusleiste anzuzeigen. Die Information selbst wird aus einem Datenfeld des Knotens entnommen. Welches Datenfeld übergeben wird, ist über die Konfigurationsdatei (IFD, Feld IF_ID2) einstellbar und ist normalerweise das Feld mit dem Index 4.

	Datentyp	Beschreibung
Parameter: ⇒ text	String	Text

Code-Beispiel

```

Private Sub VcNet1_OnStatusLineText(ByVal Text As String)
    'show text on status bar
    txtStatusBar.Text = Text
End Sub

```

OnSupplyTextEntry

Ereignis von VcNet

Dieses Ereignis tritt nur auf, wenn Sie die VcNet-Eigenschaft **Enable-SupplyTextEntryEvent** 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.

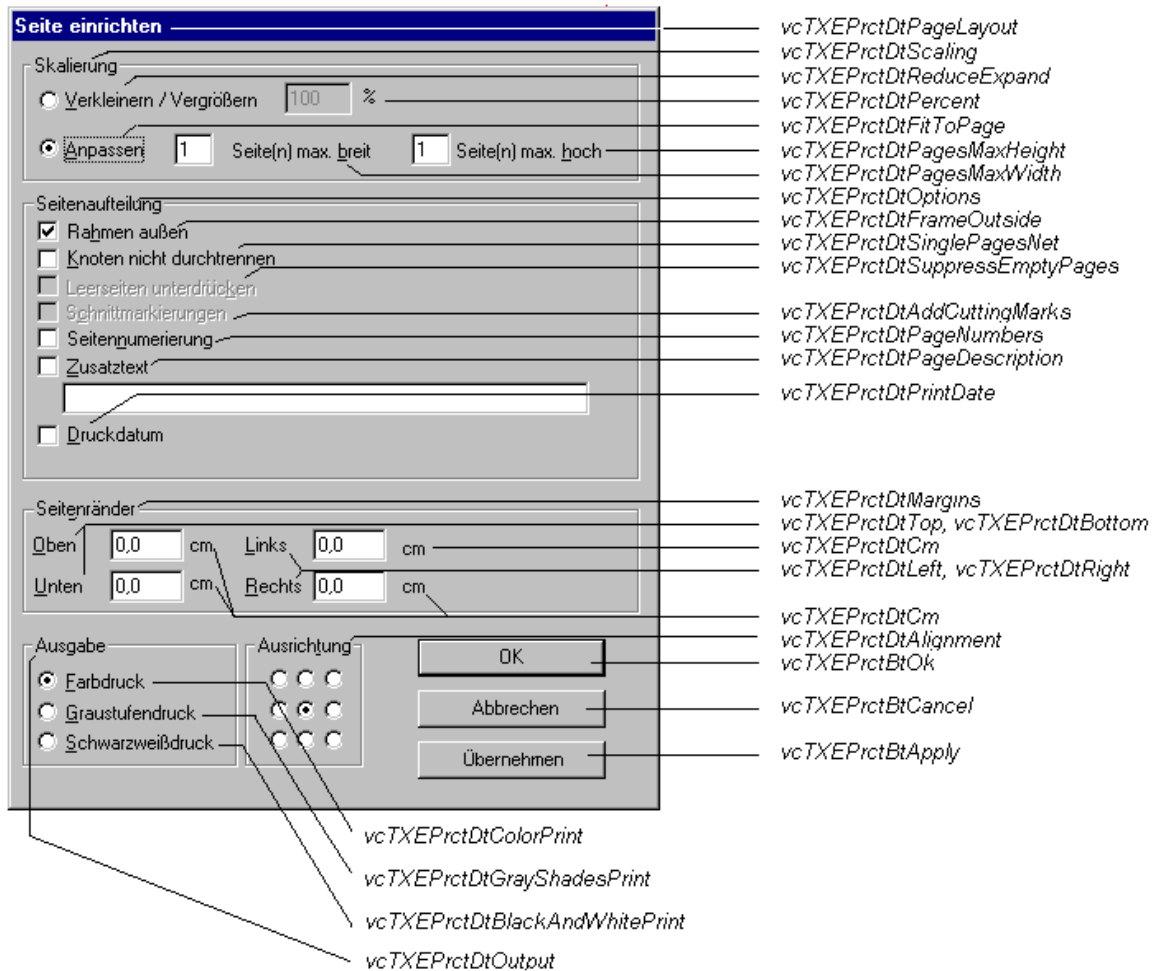
	Datentyp	Beschreibung
Parameter:		
⇒ controllIndex	TextEntryIndexEnum	Zu ersetzender Text
	<p>Mögliche Werte:</p> <p>vcTXEctxmenArrange 2150</p> <p>vcTXEctxmenArrowMode 2116</p> <p>vcTXEctxmenCopyNodes 2152</p> <p>vcTXEctxmenCreateNodesAndLinksMode 2117</p> <p>vcTXEctxmenCutNodes 2151</p> <p>vcTXEctxmenDeleteLink 2102</p> <p>vcTXEctxmenDeleteNode 2101</p> <p>vcTXEctxmenEditLink 2154</p> <p>vcTXEctxmenEditNode 2100</p> <p>vcTXEctxmenFilePrint 2122</p> <p>vcTXEctxmenFilePrintPreview 2121</p> <p>vcTXEctxmenFilePrintSetup 2120</p> <p>vcTXEctxmenFullDiagram 2156</p> <p>vcTXEctxmenGraphicExport 2123</p> <p>vcTXEctxmenPageLayout 2119</p> <p>vcTXEctxmenPasteNodes 2153</p> <p>vcTXEctxmenShowLegendView 2158</p> <p>vcTXEctxmenShowWorldView 2157</p> <p>vcTXEctxmenSubDiagram 2155</p> <p>vcTXEdateAM 2225</p> <p>vcTXEdateCW 2223</p> <p>vcTXEdateDay0 2212</p> <p>vcTXEdateDay1 2213</p> <p>vcTXEdateDay2 2214</p> <p>vcTXEdateDay3 2215</p> <p>vcTXEdateDay4 2216</p> <p>vcTXEdateDay5 2217</p> <p>vcTXEdateDay6 2218</p> <p>vcTXEdateMonth0 2200</p> <p>vcTXEdateMonth1 2201</p> <p>vcTXEdateMonth10 2210</p> <p>vcTXEdateMonth11 2211</p> <p>vcTXEdateMonth2 2202</p> <p>vcTXEdateMonth3 2203</p> <p>vcTXEdateMonth4 2204</p> <p>vcTXEdateMonth5 2205</p> <p>vcTXEdateMonth6 2206</p> <p>vcTXEdateMonth7 2207</p> <p>vcTXEdateMonth8 2208</p> <p>vcTXEdateMonth9 2209</p> <p>vcTXEdateOClock 2224</p> <p>vcTXEdatePM 2226</p> <p>vcTXEdateQuarter0 2219</p> <p>vcTXEdateQuarter1 2220</p>	<p>Text im Kontextmenü: Knoten anordnen</p> <p>Text im Kontextmenü: Selektier-Modus</p> <p>Text im Kontextmenü: Knoten kopieren</p> <p>Text im Kontextmenü: Knoten und Verbindungen anlegen</p> <p>Text im Kontextmenü: Knoten ausschneiden</p> <p>Text im Kontextmenü: Verbindung löschen</p> <p>Text im Kontextmenü: Knoten löschen</p> <p>Text im Kontextmenü: Verbindung bearbeiten</p> <p>Text im Kontextmenü: Daten bearbeiten</p> <p>Text im Kontextmenü: Drucken</p> <p>Text im Kontextmenü: Druckvorschau</p> <p>Text im Kontextmenü: Drucker einrichten</p> <p>Text im Kontextmenü Gesamtnetz wiederherstellen</p> <p>Text im Kontextmenü: Grafik exportieren</p> <p>Text im Kontextmenü: Seite einrichten</p> <p>Text im Kontextmenü: Knoten einfügen</p> <p>Text im Kontextmenü Legendenansicht anzeigen</p> <p>Text im Kontextmenü Komplettansicht anzeigen</p> <p>Text im Kontextmenü Teilnetz erstellen</p> <p>Ausgabertext für vormittags</p> <p>Ausgabertext für Kalenderwoche</p> <p>Ausgabertext für Montag</p> <p>Ausgabertext für Dienstag</p> <p>Ausgabertext für Mittwoch</p> <p>Ausgabertext für Donnerstag</p> <p>Ausgabertext für Freitag</p> <p>Ausgabertext für Samstag</p> <p>Ausgabertext für Sonntag</p> <p>Ausgabertext für Januar</p> <p>Ausgabertext für Februar</p> <p>Ausgabertext für November</p> <p>Ausgabertext für Dezember</p> <p>Ausgabertext für März</p> <p>Ausgabertext für April</p> <p>Ausgabertext für Mai</p> <p>Ausgabertext für Juni</p> <p>Ausgabertext für Juli</p> <p>Ausgabertext für August</p> <p>Ausgabertext für September</p> <p>Ausgabertext für Oktober</p> <p>Ausgabertext für Uhr</p> <p>Ausgabertext für nachmittags</p> <p>Ausgabertext für 1. Quartal</p> <p>Ausgabertext für 2. Quartal</p>

vcTXEDateQuarter2 2221	Ausgabertext für 3. Quartal
vcTXEDateQuarter3 2222	Ausgabertext für 4. Quartal
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
vcTXEDlgLegIddcancel 2042	Schaltfläche im Dialog Legendenattribute: Abbrechen
vcTXEDlgLegIdd 2040	Dialog Legendenattribute Beschriftung der Titelzeile
vcTXEDlgLegIddok 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"
vcTXEDlgNedIddapply 2027	Dialog Vorgänge bearbeiten , "Übernehmen"-Schaltfläche
vcTXEDlgNedIddcancel 2016	Text im Dialog Vorgänge bearbeiten: Abbrechen
vcTXEDlgNedIddclose 2029	Dialog Vorgänge bearbeiten: Schließen -Schaltfläche
vcTXEDlgNedIdd 2014	Überschrift des Dialogs Vorgänge bearbeiten
vcTXEDlgNedIddhelp 2028	Dialog Vorgänge bearbeiten: Hilfe -Schaltfläche
vcTXEDlgNedIddok 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
vcTXEErrTxtEntryTooLong 2730	Meldungstext: "Eintrag ist zu lang, %s Zeichen sind möglich."
vcTXEErrTxtWrongLongInteger 2729	Meldungstext: "Eintrag ist kein Integer oder zu lang."

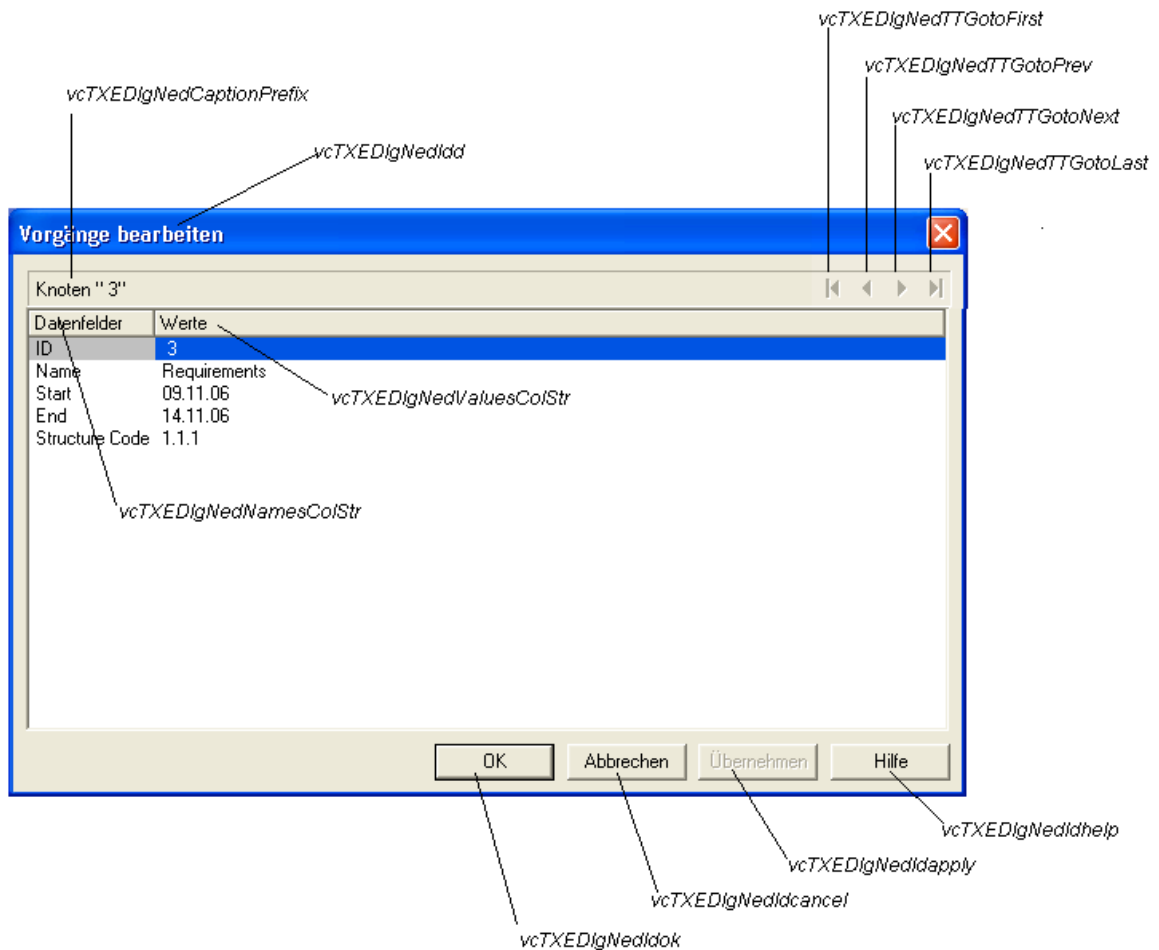
vcTXEPrctBtAll 2306	Schaltflächen-Text des Druckvorschau -Dialogs: Übersicht
vcTXEPrctBtApply 2318	Schaltflächen-Text im Seite einrichten -Dialog: Anwenden
vcTXEPrctBtCancel 2302	Schaltflächen-Text im Druck-Info -Fenster: Abbrechen
vcTXEPrctBtClose 2303	Schaltflächen-Text des Druckvorschau -Dialogs: 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
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
vcTXEPrctDtCurrentValues 2581	Text des Seite einrichten -Dialogs: Aktuell
vcTXEPrctDtEnableBoth 2561	inaktiv
vcTXEPrctDtEnableDiagram 2559	Text des Seite einrichten -Dialogs: Diagramm anzeigen
vcTXEPrctDtEnableTable 2558	inaktiv
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: Form A Form B Form C
vcTXEPrctDtFoldingMarksText 2576	Text des Seite einrichten -Dialogs: "&Faltmarkierungen (DIN 824)
vcTXEPrctDtFooterGroup 2584	Text des Seite einrichten -Dialogs: Fußzeile
vcTXEPrctDtFrameOutside 2515	Text des Seite einrichten -Dialogs: Rahmen außen
vcTXEPrctDtInch 2588	Text des Seite einrichten -Dialogs: Zoll
vcTXEPrctDtLeft 2520	Text des Seite einrichten -Dialogs: Links

vcTXEPrctDtMargins 2529	Text des Seite einrichten -Dialogs: Mindestgrößen für die Seitenränder
vcTXEPrctDtMaxPages 2580	Text des Seite einrichten -Dialogs: Seiten
vcTXEPrctDtOff 2557	Text Aus Dialog
vcTXEPrctDtOptions 2528	Text des Seite einrichten -Dialogs: Seitenaufteilung
vcTXEPrctDtPageDescription 2562	Text des Seite einrichten -Dialogs: Text
vcTXEPrctDtPageLayout 2532	Fenstertitel des Seite einrichten -Dialogs
vcTXEPrctDtPageNumberingItems 2582	Text des Seite einrichten -Dialogs: Zeile.Spalte Spalte.Zeile Seite/Anzahl
vcTXEPrctDtPageNumbers 2518	Text des Seite einrichten -Dialogs: Seitennummerierung
vcTXEPrctDtPagePadding 2585	Text des Seite einrichten -Dialogs: Seiten mit Leerraum auff&üllen
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: %
vcTXEPrctDtPrint 2506	Text im Druck-Info-Fenster: Drucke
vcTXEPrctDtPrintDate 2564	Text des Seite einrichten -Dialogs: &Druckdatum
vcTXEPrctDtPrintingPage 2556	Text im Druck-Info-Fenster: Seite %1 von %2 wird gedruckt auf
vcTXEPrctDtProjectName 2502	Text im Druck-Info-Fenster: Projektname
vcTXEPrctDtReduceExpand 2507	Text des Seite einrichten -Dialogs: Zoomfaktor
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)
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. \r\nAußerdem 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. \r\nAußerdem wird der rechte Rand auf %2 cm reduziert.

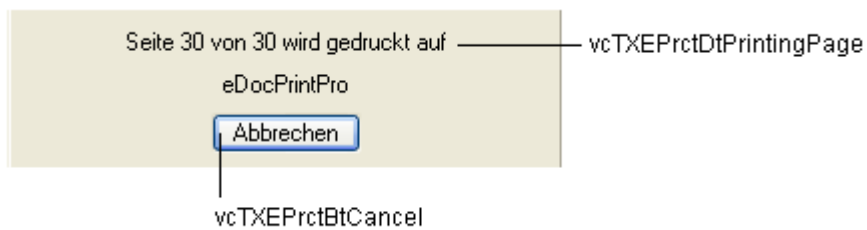
	vcTXEPrctMtAdjustRightAndLeftMargin 2435	Meldungstext: Der rechte Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. \r\nAußerdem wird der linke Rand auf %2 cm reduziert.
	vcTXEPrctMtAdjustTopAndBottomMargin 2436	Meldungstext: Der obere Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert. \r\nAußerdem 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)
⇒ textEntry	String	Text, der den Standardtext ersetzen soll
⇔ returnStatus	Variant	Rückgabestatus



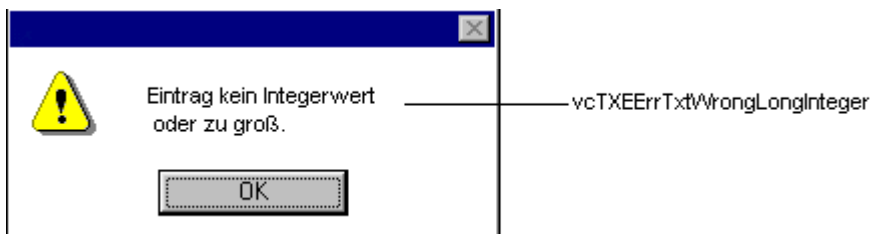
Konstanten des Dialogs **Seite einrichten**



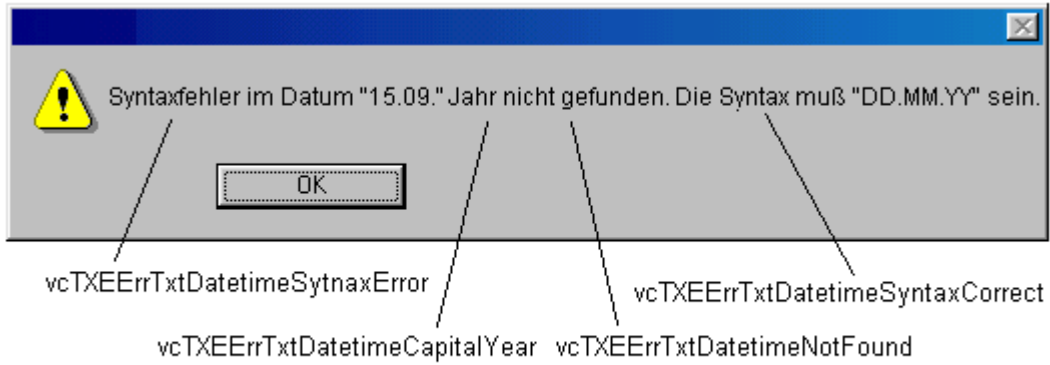
Konstanten der Dialoge **Vorgänge bearbeiten** und **Verbindung bearbeiten**, hier am Beispiel des Dialogs **Vorgänge bearbeitend** dargestellt



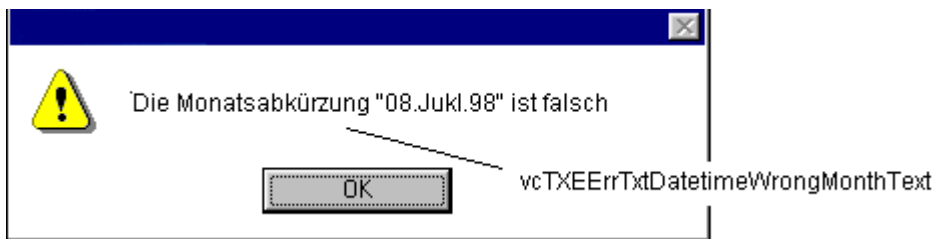
Konstanten der Infobox **Drucken**



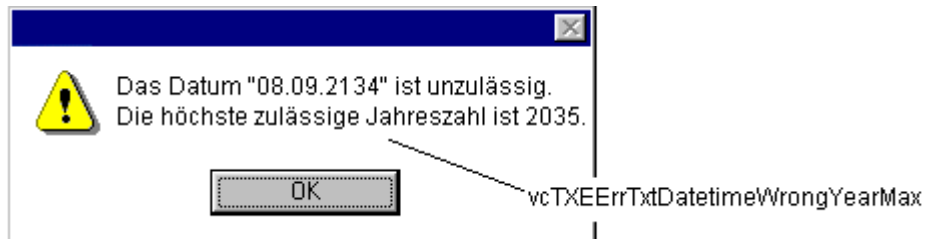
Konstanten der Fehlermeldung **Wert kein Integer**



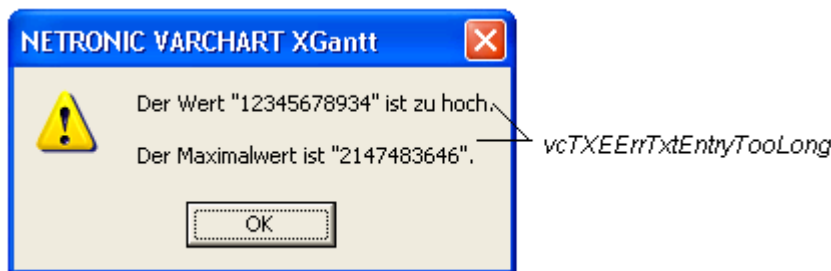
Konstanten der Fehlermeldung **Syntaxfehler im Datum**



Konstanten der Fehlermeldung **Fehler im Datum, Monat falsch**



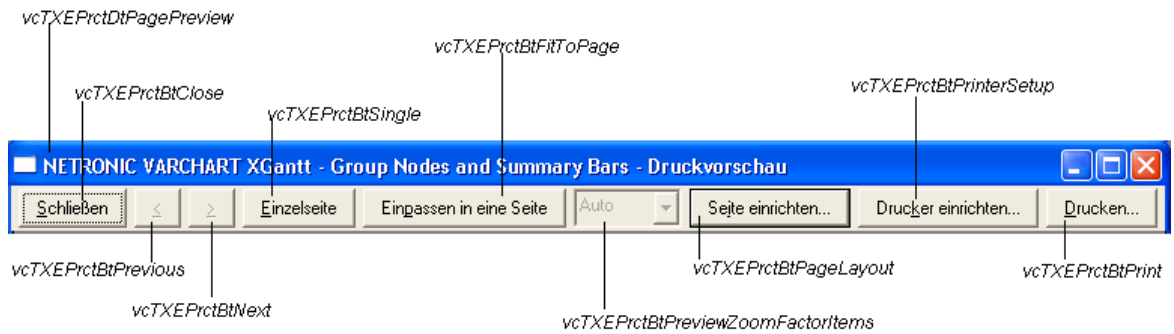
Konstante der Fehlermeldung **Fehler im Datum, Jahr zu groß**



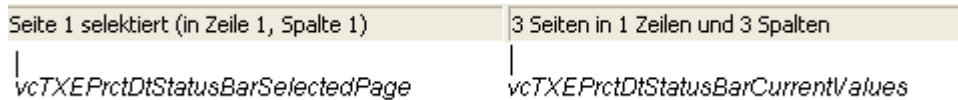
Konstanten der Fehlermeldung **Wert zu groß**



Konstanten der Tastenbeschriftungen in der **Druckvorschau im Einzelansichtsmodus**



Konstanten der Tastenbeschriftungen in der **Druckvorschau im Übersichtsmodus**



Konstanten der Statuszeile im Dialog **Druckvorschau**

Code-Beispiel

```
Private Sub VcNet1_OnSupplyTextEntry(ByVal controlIndex As _
                                   VcNetLib.TextEntryIndexEnum, _
                                   textEntry As String, _
                                   returnStatus As Variant)
    'change the texts of the context meun items
    Select Case controlIndex
        Case vcTXEPrctBtArrange
            textEntry = "Position all nodes automatically"
    End Select
End Sub
```

OnSupplyTextEntryAsVariant

Ereignis von VcNet

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.

OnToolTipText

Ereignis von VcNet

Dieses Ereignis tritt nur auf, wenn Sie die Eigenschaft **ShowToolTip** auf **True** gesetzt haben. Das Ereignis tritt auf, sobald der Cursor auf ein VcNet-Objekt bewegt wird. Es liefert Informationen über das Objekt, den Objekttyp und die Koordinaten des Cursors. Sie können mit Hilfe dieses Ereignisses die vorgegebenen Texte durch eigene Texte ersetzen, z. B. um sie in

unterschiedliche Sprachen zu übersetzen. Durch Setzen des ReturnStatus auf **vcRetStatFalse** oder Leerlassen des Textstrings "" können Sie den Tooltip an dieser Stelle unterdrücken.

	Datentyp	Beschreibung
Parameter:		
⇒ hitObject	Object	Objekt
⇒ hitObjectType	VcObjectTypeEnum	Objekttyp
	Mögliche Werte:	
	vcObjTypeBox 15	Objekttyp Box
	vcObjTypeGroup 7	Objekttyp Gruppe
	vcObjTypeLinkCollection 3	Objekttyp LinkCollection
	vcObjTypeNode 2	Objekttyp Knoten
	vcObjTypeNodeInLegend 17	Objekttyp Knoten im Legendebereich
	vcObjTypeNone 0	kein Objekt
⇒ x	Long	X-Koordinate
⇒ y	Long	Y-Koordinate
⇒ ToolTipText	String	Anzuzeigender Text, kann maximal 1024 Zeichen lang sein
↔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnToolTipText(ByVal hitObject As Object, _
                                ByVal hitObjectType As _
                                VcNetLib.VcObjectTypeEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                tooltipText As String, _
                                returnStatus As Variant)
    If hitObjectType = vcObjTypeNode Then
        tooltipText = "Moving over node!"
    End If
End Sub
```

OnToolTipTextAsVariant

Ereignis von VcNet

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 ↔) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

OnWorldViewClosed

Ereignis von VcNet

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Komplettansicht geschlossen wird.

	Datentyp	Beschreibung
Parameter: ↔ (no parameter)		

Code-Beispiel

```
Private Sub VcNet1_OnWorldViewClosed()
    MsgBox "Do you want to close the worldview window?", vbOKCancel
End Sub
```

OnZoomFactorModifyComplete

Ereignis von VcNet

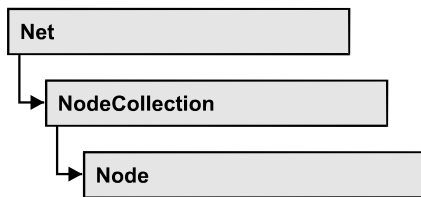
Dieses Ereignis tritt ein, wenn der Anwender in der Komplettansicht (WorldView) die Größe des Rechtecks verändert hat oder markierte Objekte gezoomt hat. Sie können stufenlos zoomen, indem Sie bei gedrückter Strg-Taste das Mausrad drehen. In bestimmten Schritten können Sie zoomen, indem Sie bei gedrückter Strg-Taste die Plus- bzw. Minus-Tasten des Ziffernblocks der Tastatur drücken.

	Datentyp	Beschreibung
Parameter: ↔ (no parameter)		

Code-Beispiel

```
Private Sub VcNet1_OnZoomFactorModifyComplete()
    MsgBox "Zoomfactor: " & Net1.ZoomFactor
End Sub
```

7.43 VcNode



Knoten sind Grundelemente eines Netzdiagramms. Sie lassen sich über Verbindungen zu einer Struktur verknüpfen. Das Aussehen eines Knotens wird über diejenigen NodeAppearance-Objekte bestimmt, deren Filter auf den Knoten zutreffen. Erzeugt werden Knoten über die Methode **VcNet.InsertNodeRecord** oder interaktiv.

Eigenschaften

- AllData
- DataField
- ID
- IncomingLinks
- MarkNode
- OutgoingLinks

Methoden

- DataRecord
- DeleteNode
- RelatedDataRecord
- 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 VcNet1_OnNodeModify(ByVal node As VcNetLib.VcNode, _
                               ByVal modificationType As _
                               VcNetLib.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
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

```
Private Sub VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
                                ByVal location As VcNetLib.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

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 VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.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 **Knotenmarkierung** nicht **Ohne** ausgewählt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knoten markiert/nicht markiert

Code-Beispiel

```
Private Sub VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
                               ByVal location As VcNetLib.LocationEnum, _
```

658 API-Referenz: VcNode

```
ByVal x As Long, ByVal y As Long, _
returnStatus As Variant)

Dim nodeMarked As Boolean

nodeMarked = node.MarkNode
MsgBox (nodeMarked)
returnStatus = vcRetStatNoPopup

End Sub
```

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 VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
ByVal location As VcNetLib.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
```

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 VcNet1_OnNodeRClick(ByVal node As VcNetLib.VcNode, _
    ByVal location As _
    VcNetLib.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
```

RelatedDataRecord

Methode von VcNode

Mit dieser Eigenschaft können Sie einen Datensatz aus einer verknüpften Tabelle erfragen, der dem Datensatz der Knotentabellensuche 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
Rückgabewert	VcDataRecord	Zurückgegebener zugeordneter Datensatz

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

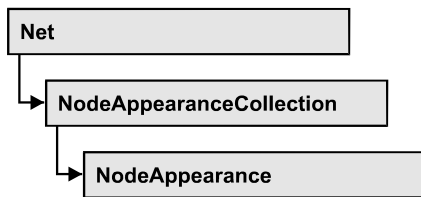
Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcNet1.NodeCollection
Set node = nodeCltn.FirstNode

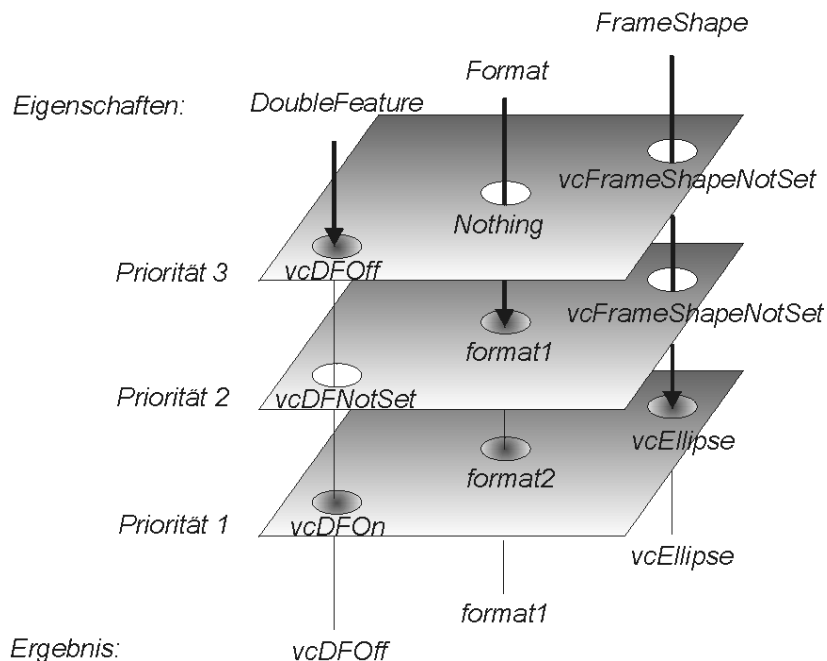
node.DataField(12) = "Group A"
node.UpdateNode
```

7.44 VcNodeAppearance



Ein NodeAppearance-Objekt bestimmt das Aussehen aller Knoten, deren Daten die dem NodeAppearance-Objekt zugeordneten Filterbedingungen erfüllen. Verschiedene NodeAppearance-Objekte können im Dialogfeld **Knotenaussehen verwalten**, das Sie über die Eigenschaftenseite **Knoten** erreichen, voreingestellt werden.

Die Skizze zeigt, wie sich die Eigenschaften von NodeAppearance-Objekten auf das Aussehen eines Knotens auswirken. Die auf den Knoten zutreffenden NodeAppearances sind nach Priorität absteigend dargestellt. Nicht gesetzte Eigenschaften bei NodeAppearance-Objekten führen dazu, daß die Eigenschaft des nächsttieferen NodeAppearance-Objektes übernommen wird.



Eigenschaften

- BackColorAsARGB
- BackColorDataFieldIndex
- BackColorMapName
- DoubleFeature
- FilterName
- FormatName

- FrameAroundFieldsVisible
- FrameShape
- LegendText
- LineColor
- LineColorDataFieldIndex
- LineColorMapName
- LineThickness
- LineType
- Name
- Pattern
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternDataFieldIndex
- PatternMapName
- Piles
- Shadow
- ShadowColorAsARGB
- Specification
- StrikeThrough
- StrikeThroughColor
- ThreeDEffect
- VisibleInLegend

Methoden

- PutInOrderAfter

Eigenschaften

BackColorAsARGB

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Hintergrundfarbe eines Knotens einstellen 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.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {(0...255),0...255},{0...255},{0...255}}

Code-Beispiel

```
Dim nodeAppearanceCltn As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCltn = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCltn.FirstNodeAppearance

nodeAppearance.BackColor = RGB(100, 100, 100)
```

BackColorDataFieldIndex

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der Verbindung mit der Eigenschaft **BackColorMapName** benötigt wird. Wenn Sie hier **-1** angeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

BackColorMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhnungstabelle für die Hintergrundfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **BackColorDataFieldIndex -1** angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhnungstabelle

DoubleFeature

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie eine doppelte Umrahmung eines Knotens einstellen oder erfragen. Bei **vcDFNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcDFNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceDoubleFeatureEnum Mögliche Werte: vcDFNotSet -1 vcDFOff 0 vcDFOn 1	Verschiedene Formen von Doppellinien Schalter für DoubleFeature nicht gesetzt Schalter für DoubleFeature aus Schalter für DoubleFeature an

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.DoubleFeature = vcDFOn
```

FilterName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Filter des NodeAppearance-Objekts setzen oder erfragen. Es gibt Sonderfilter, die unveränderlich sind:

- <ALWAYS>: gilt immer (beim Standard-Aussehen immer gesetzt)
- <NEVER>: gilt niemals
- <INTERFACE-COLLAPSED>: gilt für Teildiagramm-Anschlussknoten

	Datentyp	Beschreibung
Eigenschaftswert	String	Filtername

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim filtername As String
```

```

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

filtername = nodeAppearance.filtername

```

FormatName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Format des NodeAppearance-Objekts setzen oder erfragen. Ist keine Eigenschaft gesetzt, wird die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes übernommen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht leer ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Knotenformat-Objekts oder leere Zeichenkette

Code-Beispiel

```

Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim format1 As VcNodeFormat

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

Set format1 = nodeAppearance.format
MsgBox (format1.name)

```

FrameAroundFieldsVisible

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft kann festgelegt werden, ob der Rahmen um die innenliegenden Felder sichtbar ist oder nicht. Die Außenrandlinie der Form ist davon nicht betroffen, daher wirkt sich diese Eigenschaft bei den möglichen Rahmenformen unterschiedlich aus und hat z.B. beim Typ **vcRectangle** keine Auswirkung.

Diese Eigenschaft kann auch im Dialog **Knotenaussehen bearbeiten** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceFrameAroundFieldsVisibleEnum	Rahmen um Feld
	Mögliche Werte: vcFFVNotSet -1 vcFFVOff 0	Feldumrandung nicht gesetzt Schalter für Feldumrandung aus

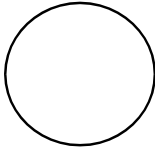

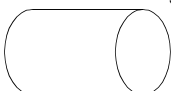

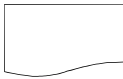
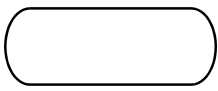

vcFFVOn 1

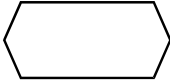
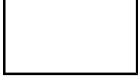
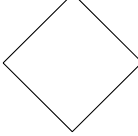


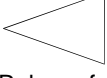

Schalter für **Feldumrandung** an

FrameShape

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt die Rahmenform zuweisen oder erfragen. Bei **vcFrameShapeNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcFrameShapeNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

Eigenschaftswert	Datentyp	Beschreibung
	AppearanceFrameShapeEnum	Rahmenform
	Mögliche Werte: vcCircle 11	Rahmenform kreisförmig 
	vcEllipse 12	Rahmenform elliptisch 
	vcFile 19	Rahmenform liegender Zylinder 
	vcFrameShapeNotSet -1 vcLeftArrow 17	Rahmenform nicht gesetzt Rahmenform pfeilförmig, nach links zeigend 
	vcListing 20	Rahmenform Dokument 
	vcNoFrameShape 1 vcOval 4	keine Rahmenform Rahmenform oval 
	vcParallelogram 9	Rahmenform Parallelogram 

vcPointed 7	Rahmenform an den senkrechten Kanten spitz ausgezogen
	
vcRectangle 2	Rahmenform rechteckig
	
vcRhombus 21	Rahmenform Raute
	
vcRightArrow 18	Rahmenform pfeilförmig, nach rechts zeigend
	
vcRounded 3	Rahmenform rechteckig gerundet
	
vcTriangleLeft 10	Rahmenform dreieckig, Spitze links
	
vcTriangleUp 13	Rahmenform dreieckig, Spitze oben
	

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.FrameShape = vcEllipse
```

LegendText**Eigenschaft von VcNodeAppearance**

Mit dieser Eigenschaft können Sie einem Knotenaussehen einen Text zuweisen oder erfragen, der in der Legende für das jeweilige Knotenaussehen angezeigt wird. Steht hier "", dann wird der Inhalt der Eigenschaft **Name** angezeigt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendertext des Knotenaussehens Standardwert: "" (content of the property Name)

LineColor

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt eine Linienfarbe zuweisen oder erfragen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte oder -1 ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.LineColor = RGB(256, 0, 100)
```

LineColorDataFieldIndex

Eigenschaft von VcNodeAppearance

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

LineColorMapName

Eigenschaft von VcNodeAppearance

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

LineThickness

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Linienstärke eines NodeAppearance-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.

Wenn Sie diese Eigenschaft auf **-1** setzen, kommt die gleichnamige Eigenschaft des in der Priorität nächst niedrigeren NodeAppearance-Objekts zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	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 nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")

nodeAppearance.LineThickness = 3
```

LineType

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt einen Linientyp zuweisen oder erfragen. Bei **vcNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	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

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.LineType = vcDotted
```

Name**Eigenschaft von VcNodeAppearance**

Mit dieser Eigenschaft können Sie den Namen eines NodeAppearance-Objekts erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim nodeAppName As String

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

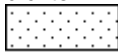


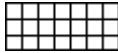


nodeAppName = nodeAppearance.name
```



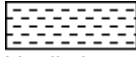

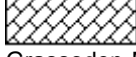
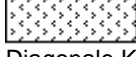

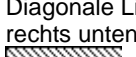
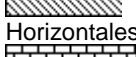
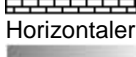

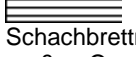


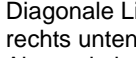

Pattern

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Muster des Knotens festlegen oder erfragen. Wenn in der Eigenschaft **PatternMapName** eine Zuordnungstabelle angegeben ist, steuert diese das Muster in Abhängigkeit von den Daten. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

Werte von vc05PercentPattern bis vc90PercentPattern lauten richtigerweise 2001 bis 2011.

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 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
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 vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern 

674 API-Referenz: VcNodeAppearance

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
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 VcNodeAppearance

Mit dieser Eigenschaft können Sie die Musterfarbe des Knotens 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.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

Wenn in der Eigenschaft **PatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe datenabhängig.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {(0...255),0...255},{0...255},{0...255}}

PatternColorDataFieldIndex

Eigenschaft von VcNodeAppearance

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

PatternColorMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Farbzuhnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuhnungstabelle 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 **PatternColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuhnungstabelle

PatternDataFieldIndex

Eigenschaft von VcNodeAppearance

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

PatternMapName

Eigenschaft von VcNodeAppearance

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 Farbzusordnungstabelle 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

Piles

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Anzahl von Knotenstapeln im Diagramm erfragen oder festlegen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren `NodeAppearance`-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei `VcNodeAppearance`).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl gestapelter Knoten oder -1

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Piles = 2
```

Shadow

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem `NodeAppearance`-Objekt einen Schatten zuweisen oder erfragen. Bei **vcShNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren `NodeAppearance`-Objektes

zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcShNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceShadowEnum Mögliche Werte: vcShNotSet -1 vcShOff 0 vcShOn 1	Art der Schattensetzung Schalter für Schatten nicht gesetzt Schalter für Schatten aus Schalter für Schatten an

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Shadow = vcShOn
```

ShadowColorAsARGB

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Farbe des Schattens am Knoten einstellen 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.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte {0...255},0...255},{0...255},{0...255} Standardwert: &hFFD8D8D8 (grau)

Code-Beispiel

```
Dim nodeAppearanceCltn As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCltn = VcNet1.NodeAppearanceCollection
```

```
Set nodeAppearance = nodeAppearanceCltn.FirstNodeAppearance
nodeAppearance.ShadowColor = MakeARGB(100, 100, 100, 100)
```

Specification

Nur-Lese-Eigenschaft von VcNodeAppearance

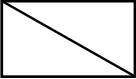
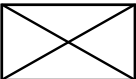
Mit dieser Eigenschaft können Sie die Spezifikation dieses Knotenaussehens auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Knotenaussehens mit der Methode **VcNodeAppearanceCollection.AddBySpecification** benutzt werden.

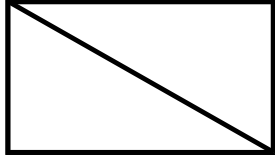
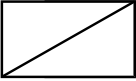
	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Knotenaussehens

StrikeThrough

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Durchstreichmuster eines NodeAppearance-Objekts setzen oder erfragen. Bei **vcStrikeThroughNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcStrikeThroughNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceStrikeThroughEnum	Durchstreichmuster
	Mögliche Werte: vcBackslashed 3	Knoten von links oben nach rechts unten durchgestrichen 
	vcCrossed 4	Knoten gekreuzt durchgestrichen 

vcDoubleBackslashed 8	Knoten doppelt von links oben nach rechts unten durchgestrichen 
vcDoubleSlashed 7	Knoten doppelt von links unten nach rechts oben durchgestrichen
vcNoStrikeThrough 0	Knoten nicht durchgestrichen
vcSlashed 2	Knoten von links unten nach rechts oben durchgestrichen 
vcStrikeThroughNotSet -1	kein Durchstreichmuster gesetzt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Strikethrough = vcBackslashed
```

StrikeThroughColor

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Farbe des Durchstreichmusters eines NodeAppearance-Objekts setzen oder erfragen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte oder -1 ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.StrikeThroughColor = RGB(255, 0, 0)
```

ThreeDEffect

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den 3D-Effekt für das NodeAppearance-Objekt erfragen oder festlegen. Bei **vc3DNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vc3DNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceThreeDEffectEnum Mögliche Werte: vc3DNotSet -1 vc3DOff 0 vc3DOn 1	Art der 3D-Effekt-Setzung Schalter für 3D-Effekt nicht gesetzt Schalter für 3D-Effekt aus Schalter für 3D-Effekt an

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.ThreeDEffect = vc3DOn
```

VisibleInLegend

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das NodeAppearance-Objekt in der Legende sichtbar ist. Diese Eigenschaft kann auch im Dialog **Knotenaussehen verwalten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knotenaussehen in Legende sichtbar (True)/nicht sichtbar (False) Standardwert: True

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")

nodeAppearance.VisibleInLegend = False
```

Methoden

PutInOrderAfter

Methode von VcNodeAppearance

Mit dieser Methode können Sie dieses Knotenaussehen in der Auflistung aller Knotenaussehen hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Knotenaussehen an die erste Stelle gesetzt. Die Reihenfolge der Knotenaussehen in der Auflistung entscheidet darüber, in welcher Reihenfolge sie auf die Knoten angewendet werden.

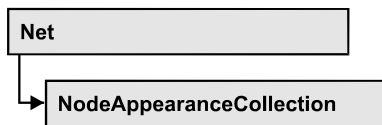
	Datentyp	Beschreibung
Parameter: refNodeAppearanceName	String	Name des Knotenaussehens, hinter das das aktuelle Knotenaussehen gesetzt werden soll.
Rückgabewert	Void	

Code-Beispiel

```
Dim nodeAppCltn As VcNodeAppearanceCollection
Dim nodeApp1 As VcNodeAppearance
Dim nodeApp2 As VcNodeAppearance

nodeAppCltn = VcGantt1.NodeAppearanceCollection()
nodeApp1 = nodeAppCltn.Add("nodeApp1")
nodeApp2 = nodeAppCltn.Add("nodeApp2")
nodeApp1.PutInOrderAfter("nodeApp2")
nodeAppCltn.Update()
```

7.45 VcNodeAppearanceCollection



In einem Objekt vom Typ `VcNodeAppearanceCollection` sind automatisch alle definierten `NodeAppearance`-Objekte zusammengefasst. Sie haben Zugriff auf ein Knotenaussehen über die Methode **`NodeAppearanceByName`**. Mit der Eigenschaft **`Count`** können Sie die Anzahl der `NodeAppearance`-Objekte ermitteln.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstNodeAppearance`
- `NextNodeAppearance`
- `NodeAppearanceByIndex`
- `NodeAppearanceByName`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcNodeAppearanceCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenaussehen-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 nodeApp As VcNodeAppearance

For Each nodeApp In VcNet1.NodeAppearanceCollection
    Debug.Print nodeApp.Name
Next
```

Count**Nur-Lese-Eigenschaft von VcNodeAppearanceCollection**

Mit dieser Eigenschaft können Sie die Anzahl der NodeAppearance-Objekte erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der NodeAppearance-Objekte

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim numberNodeAppColl As Integer

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection

numberNodeAppColl = nodeAppearanceCollection.Count
```

Methoden**Add****Methode von VcNodeAppearanceCollection**

Mit dieser Methode können Sie ein neues Knotenaussehen in der NodeAppearanceCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcNodeAppearance-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Bei dem neuen Knotenaussehen sind standardmäßig alle Eigenschaften auf transparent gesetzt.

	Datentyp	Beschreibung
Parameter: ⇒ newName	String	Name des NodeAppearance-Objekts

Rückgabewert	VcNodeAppearance	Neues NodeAppearance-Objekt
---------------------	------------------	-----------------------------

Code-Beispiel

```
Set newNodeAppearance = VcNet1.NodeAppearanceCollection.Add("nodeapp1")
```

AddBySpecification**Methode von VcNodeAppearanceCollection**

Mit dieser Methode können Sie ein Knotenaussehen über eine Knotenaussehen-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Knotenaussehen-Objekten. Die Spezifikation eines Knotenaussehens kann erfragt (siehe VcNodeAppearance-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Knotenaussehen mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter:		
⇒ nodeAppearanceSpecification	String	Knotenaussehen-Spezifikation
Rückgabewert	VcNodeAppearance	Neues Knotenaussehen-Objekt

Copy**Methode von VcNodeAppearanceCollection**

Mit dieser Methode können Sie ein Knotenaussehen kopieren. Wenn das Knotenaussehen mit dem angegebenen Namen existiert und der Name des neuen Knotenaussehens noch nicht verwendet wird, wird das neue Knotenaussehen-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇒ fromName	String	Name des zu kopierenden Knotenaussehens
⇒ newName	String	Name des neuen Knotenaussehens
Rückgabewert	VcNodeAppearance	NodeAppearance-Objekt

FirstNodeAppearance

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie auf das erste NodeAppearance-Objekt der Auflistung zugreifen, um anschließend mit der Methode **NextNodeAppearance** über die nachfolgenden Objekte zu iterieren. Existiert kein NodeAppearance-Objekt in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeAppearance	Erstes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance
```

NextNodeAppearance

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden NodeAppearance-Objekte der Auflistung zugreifen, nachdem Sie mit der Methode **FirstNodeAppearance** den Initialwert erfasst haben. Sind alle Objekte durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeAppearance	Nachfolgendes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

While Not nodeAppearance Is Nothing
    Listbox.AddItem nodeAppearance.Name
    Set nodeAppearance = nodeAppearanceCollection.NextNodeAppearance
Wend
```

NodeAppearanceByIndex

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie auf ein einzelnes NodeAppearance-Objekt über seinen Index zugreifen. Existiert kein NodeAppearance-Objekt unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ index	Integer	Index des Knotenaussehens
Rückgabewert	VcNodeAppearance	Ermitteltes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCltn As VcNodeAppearanceCollection

Set nodeAppearanceCltn = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCltn.NodeAppearanceByIndex(2)
nodeAppearance.LineThickness = 2
```

NodeAppearanceByName

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein NodeAppearance-Objekt über den Namen erfragen. Existiert kein NodeAppearance-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ nodeAppearanceName	String	Name des NodeAppearance-Objekts
Rückgabewert	VcNodeAppearance	Zurückgegebenes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcNet1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")
```

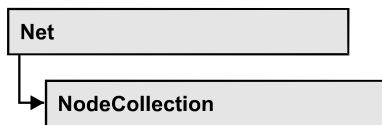

Remove

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein NodeAppearance-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	Name des Knotenaussehens
Rückgabewert	Boolean	Knotenaussehen gelöscht (True)/nicht gelöscht (False)

7.46 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 InNodeCollection** 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 VcNet1.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 = VcNet1.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 = VcNet1.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 = VcNet1.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 vcMarked 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

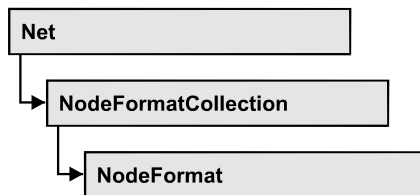
```
Dim nodeCltn As VcNodeCollection
```

692 API-Referenz: VcNodeCollection

```
Dim node As VcNode
```

```
Set nodeCltn = VcNet1.NodeCollection  
nodeCltn.SelectNodes vcSelected
```

7.47 VcNodeFormat



Ein Objekt vom Typ VcNodeFormat legt Inhalt und Erscheinungsbild eines Knotens fest. Knotenformate werden zur Designzeit im Dialogfeld **Knotenformate verwalten**, das Sie über die Eigenschaftenseite **Knoten** erreichen, verwaltet und bearbeitet.

Eigenschaften

- _NewEnum
- FieldsSeparatedByLines
- FormatField
- FormatFieldCount
- Name
- Specification
- WidthOfExteriorSurrounding

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNodeFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenformatfeld-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 VcNodeFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

FieldsSeparatedByLines**Eigenschaft von VcNodeFormat**

Mit dieser Eigenschaft können Sie festlegen, ob innenliegende Felder durch sichtbare Linien getrennt werden (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	innenliegende Felder durch sichtbare Linien getrennt (True)/ nicht getrennt (False)

Code-Beispiel

```
Dim format As VcNodeFormat

Set format = VcNet1.NodeFormatCollection.FormatByName("format1")
format.FieldsSeparatedByLines = True
```

FormatField**Nur-Lese-Eigenschaft von VcNodeFormat**

Mit dieser Eigenschaft können Sie ein VcNodeFormatField-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	Index des Knotenformatfeldes 0FormatFieldCount-1
Eigenschaftswert	VcNodeFormatField	Knotenformatfeld

FormatFieldCount

Nur-Lese-Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Anzahl der Felder eines Knotenformats ermitteln.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Felder im Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat
Dim nameofFormat As String

Set formatCollection = VcNet1.NodeFormatCollection
Set format = formatCollection.FormatByName("Standard")

numberOfFormatField = format.FormatFieldCount
```

Name

Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie den Namen des Knotenformats erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Knotenformatname

Code-Beispiel

```
Dim format As VcNodeFormat
Dim formatName As String

Set format = VcNet1.NodeFormatCollection.FirstFormat
formatName = format.Name
```

Specification

Nur-Lese-Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Knotenformats 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 Knotenformats mit der Methode **VcNodeFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Knotenformats

WidthOfExteriorSurrounding

Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Breite (in mm) des Außenbereichs des Knotenfeldes setzen, d. h. den Abstand in Millimetern, den Knoten mit diesem Knotenformat zu benachbarten Knoten und zum Rand der Darstellung halten sollen. Standardmäßig beträgt die Breite des Außenbereichs 3 mm. Bei kleineren Werten kann es gelegentlich zu Überlagerungen von grafischen Elementen kommen. Daher sollten Sie den Standardwert nur in begründeten Fällen unterschreiten.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des Außenbereichs des Knotenfeldes (in mm) 0 ... 9

Methoden

CopyFormatField

Methode von VcNodeFormat

Mit dieser Methode können Sie ein Knotenformatfeld kopieren. Das neue VcNodeFormatField-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 Knotenformatfeldes oberhalb unterhalb links von außerhalb, oberhalb außerhalb, unterhalb außerhalb, links von außerhalb, rechts von rechts von

⇒ refIndex	Integer	Index des Referenz-Knotenformatfeldes
Rückgabewert	VcNodeFormatField	Knotenformatfeld-Objekt

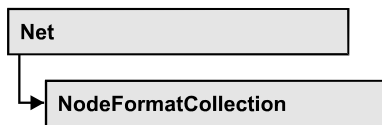
RemoveFormatField

Methode von VcNodeFormat

Mit dieser Methode können Sie ein Knotenformatfeld 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	Index des zu löschenden Knotenformatfeldes

7.48 VcNodeFormatCollection



In einem Objekt vom Typ `VcNodeFormatCollection` sind alle verfügbaren Knotenformate zusammengefasst. Über **For Each nodeFormat In NodeFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Knotenformate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaft **FormatByName**. Die Anzahl der im Auflistungsobjekt vorhandenen Knotenformate kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- `_NewEnum`
- `Count`

Methoden

- `Add`
- `AddBySpecification`
- `Copy`
- `FirstFormat`
- `FormatByIndex`
- `FormatByName`
- `NextFormat`
- `Remove`

Eigenschaften

`_NewEnum`

Nur-Lese-Eigenschaft von `VcNodeFormatCollection`

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface `IEnumVariant` implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenformat-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 VcNodeFormat

For Each format In VcNet1.NodeFormatCollection
    Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcNodeFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Knotenformatobjekte in der NodeFormat-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Knotenformate

Code-Beispiel

```
Dim formatCltn As VcNodeFormatCollection
Dim numberOfFormats As Long

Set formatCltn = VcNet1.NodeFormatCollection
numberOfFormats = formatCltn.Count
```

Methoden

Add

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein neues Knotenformat in der NodeFormatCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcNodeFormat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

Das Knotenformat besitzt standardmäßig folgende Eigenschaften:

- ein einziges Feld

- WidthOfExteriorSurrounding: 3 mm

Das Feld hat folgende Eigenschaften:

- Type: vcFFTText
- TextDataFieldIndex: in der Eigenschaftenseite **Allgemeines** festgelegte IDMinimumWidth: 3000
- Alignment: vcFFACenter
- BackColor: -1 (transparent)
- TextFontColor: RGB(0,0,0) (schwarz)
- TextFont: Arial, 10, normal
- LeftMargin, RightMargin, TopMargin, BottomMargin: 0,3 mm
- MinimumTextLineCount, MaximumTextLineCount: 1

	Datentyp	Beschreibung
Parameter: ⇒ newName	String	Name des Knotenformats
Rückgabewert	VcNodeFormat	Knotenformat-Objekt

Code-Beispiel

```
Set newNodeFormat = VcNet1.NodeFormatCollection.Add("nodeformat1")
```

AddBySpecification

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat über eine Knotenformat-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Knotenformat-Objekten. Die Spezifikation eines Knotenformats kann erfragt (siehe VcNodeFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Knotenformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter: ⇒ formatSpecification	String	Knotenformat-Spezifikation
Rückgabewert	VcNodeFormat	Neues Knotenformat-Objekt

Copy

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat kopieren. Wenn das Knotenformat mit dem angegebenen Namen existiert und der Name des neuen Knotenformats noch nicht verwendet wird, wird das neue Knotenformat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter: ⇒ fromName	String	Name des zu kopierenden Knotenformats
⇒ newName	String	Name des neuen Knotenformats
Rückgabewert	VcNodeFormat	NodeFormat-Objekt

FirstFormat

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie auf das erste Knotenformat der NodeFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Knotenformate zu iterieren. Existiert kein Knotenformat in der NodeFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeFormat	Erstes Knotenformat

Code-Beispiel

```
Dim format As VcNodeFormat
Set format = VcNet1.NodeFormatCollection.FirstFormat
```

FormatByIndex

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie auf ein einzelnes Knotenformat über ihren Index zugreifen. Existiert kein Knotenformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	Integer	Index des Knotenformats

Code-Beispiel

```
Dim nodeFormatCltn As VcNodeFormatCollection

Set nodeFormatCltn = VcNet1.NodeFormatCollection
Set nodeFormat = nodeFormatCltn.NodeFormatByIndex(2)
nodeFormat.WidthOfExteriorSurrounding = 2
```

FormatByName

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Knotenformat zugreifen. Existiert kein Knotenformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter: ⇒ formatName	String	Name des Knotenformats
Rückgabewert	VcNodeFormat	Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat

Set formatCollection = VcNet1.NodeFormatCollection
Set format = formatCollection.FormatByName("Standard")
```

NextFormat

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Knotenformate der NodeFormat-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	VcNodeFormat	Nachfolgendes Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat

Set formatCollection = VcNet1.NodeFormatCollection
Set format = formatCollection.FirstFormat

While Not format Is Nothing
    List1.AddItem format.Name
    Set format = formatCollection.NextFormat
Wend
```

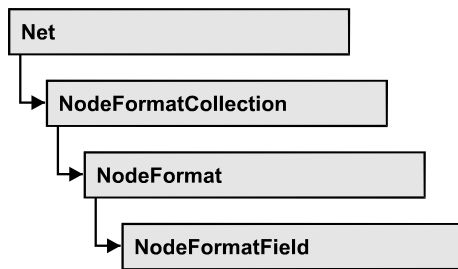
Remove

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat löschen. Wenn das Knotenformat noch irgendwo verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter: ⇒ name	String	Name des Knotenformats
Rückgabewert	Boolean	Knotenformat gelöscht (True)/nicht gelöscht (False)

7.49 VcNodeFormatField



Ein Objekt vom Typ VcNodeFormatField stellt ein Knotenformatfeld, also ein Feld eines VcNodeFormat-Objekts dar. Ein Knotenformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Knotenformat untergebracht ist.

Eigenschaften

- Alignment
- BottomMargin
- CombiField
- ConstantText
- FormatName
- GraphicsFileName
- GraphicsFileNameDataFieldIndex
- GraphicsFileNameMapName
- GraphicsHeight
- Index
- LeftMargin
- MaximumTextLineCount
- MinimumTextLineCount
- MinimumWidth
- PatternBackgroundColorAsARGB
- PatternBackgroundColorDataFieldIndex
- PatternBackgroundColorMapName
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternEx
- PatternExDataFieldIndex
- PatternExMapName
- RightMargin
- TextDataFieldIndex

- TextFont
- TextFontColor
- TextFontDataFieldIndex
- TextFontMapName
- TopMargin
- Type

Eigenschaften

Alignment

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Knotenformatfeld 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 VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des unteren Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des unteren Randes des Knotenformatfeldes in mm
		0 ... 9

CombiField

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Knotenfeld ein Kombifeld ist. (Vgl. Dialog **Knotenformat bearbeiten**).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kombifeld (True)/ kein Kombifeld (False)

ConstantText

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Knotenformatfeld ausgeben, falls der Typ des Knotenformatfeldes auf **vcFFTText** und falls die Eigenschaft **TextDataFieldIndex** auf **-1** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text

FormatName

Nur-Lese-Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen des Knotenformats erfragen, zu dem dieses Knotenformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Knotenformats

GraphicsFileName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Namen einer Grafikdatei setzen oder erfragen, deren Inhalt in dem Knotenformatfeld ausgegeben wird. Der Name muss eine gültige Grafikdatei bezeichnen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei

GraphicsFileNameDataFieldIndex

Eigenschaft von VcNodeFormatField

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 Knotenformatfeld die Grafikdatei ausgegeben, die im entsprechenden Knotenformat 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

GraphicsFileNameMapName

Eigenschaft von VcNodeFormatField

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

GraphicsHeight

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** die Höhe der Grafik in dem Knotenformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe der Grafik in mm 0 ... 99

Index

Nur-Lese-Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Index des Knotenformatfelds im zugehörigen Knotenformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Knotenformatfeldes

LeftMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des linken Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des linken Randes des Knotenformatfeldes in mm 0 ... 9

MaximumTextLineCount

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die maximale Anzahl der Zeilen in dem Knotenformatfeld setzen oder erfragen, falls das Knotenformatfeld vom Typ **vcFFText** ist. Bitte sehen Sie auch die Eigenschaft **MinimumTextLineCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	maximale Zeilenzahl 0 ... 9

MinimumTextLineCount

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die minimale Anzahl der Zeilen in dem Knotenformatfeld setzen oder erfragen, falls der Typ des Knotenformatfeldes auf **vcFFText** 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 0 ... 9

MinimumWidth

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Knotenformatfeldes 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 Knotenformatfeldes in mm 0 ... 99

PatternBackgroundColorAsARGB

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Knotenformatfeldes 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 Knotenformats besitzen soll.

Wenn in der Eigenschaft **PatternBackgroundColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	OLE_COLOR	Hintergrundfarbe des Knotenformatfelds
Eigenschaftswert	Long	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}

PatternBackgroundColorDataFieldIndex

Eigenschaft von VcNodeFormatField

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
Parameter: ⇒ Rückgabewert	Integer	Datenfeldindex
Eigenschaftswert	Long	Datenfeldindex

PatternBackgroundColorMapName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzusordnungstabelle (Typ vcColorMap) für die Hintergrundfarbe setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn der Name einer Farbzusordnungstabelle 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 **PatternBackgroundColor** benutzt.

	Datentyp	Beschreibung
Parameter: ⇒ Rückgabewert	String	Name der Farbzordnungstabelle
Eigenschaftswert	String	Name der Farbzordnungstabelle

PatternColorAsARGB

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Knotenformatfeldes 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	ARGB-Farbwerte {0...255},{0...255},{0...255},{0...255}

PatternColorDataFieldIndex

Eigenschaft von VcNodeFormatField

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

PatternColorMapName

Eigenschaft von VcNodeFormatField





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

PatternEx

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Knotenformatfeldes ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FieldFillPatternEnum	Mustertyp Standardwert: Wie im Dialog definiert
	Mögliche Werte: vcFieldNoPattern 1276 vcAeroGlassPattern 44	Kein Füllmuster Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcFieldVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
	vcFieldVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
	vcFieldVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 



PatternExDataFieldIndex

Eigenschaft von VcNodeFormatField

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 VcNodeFormatField

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
Parameter: ⇒ Rückgabewert	String	Name der Musterzuordnungstabelle
Eigenschaftswert	String	Name der Musterzuordnungstabelle

RightMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des rechten Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des rechten Randes des Knotenformatfeldes in mm 0 ... 9

TextDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Index des Datenfelds, dessen Inhalt in dem Tabellenformatfeld dargestellt werden soll, erfragen oder setzen, sofern es sich um ein Feld des Datentyps **vcFFTTText** handelt. Falls der Index **-1** ist, wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes

TextFont

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Knotenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTTText** gesetzt wurde. Wenn in der Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Schriftart in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Knotenformatfeldes

TextFontColor

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Knotenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTTText** 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 Knotenformatfelds Standardwert: -1

TextFontDataFieldIndex

Eigenschaft von VcNodeFormatField

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

TextFontMapName

Eigenschaft von VcNodeFormatField

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

TopMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des oberen Randes des Knotenformatfeldes in mm festlegen oder erfragen.

716 API-Referenz: VcNodeFormatField

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des oberen Randes des Knotenformatfeldes in mm 0 ... 9

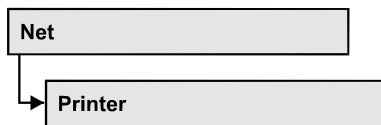
Type

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Typ des Knotenformatfelds erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldTypeEnum Mögliche Werte: vcFFTGraphics 64 vcFFTText 36	Typ des Knotenformatfeldes Grafik Text

7.50 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
- CurrentHorizontalPagesCount
- CurrentVerticalPagesCount
- CurrentZoomFactor
- CuttingMarks
- DefaultPrinterName
- DocumentName
- FitToPage
- FoldingMarksType
- MarginsShownInInches
- MaxHorizontalPagesCount
- MaxVerticalPagesCount
- Orientation
- PageDescription
- PageDescriptionString
- PageFrame
- PageNumberMode

- PageNumbers
- PagePaddingEnabled
- PaperSize
- PrintDate
- PrinterName
- RepeatTitleAndLegend
- StartUpSinglePage
- 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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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 **RepeatTitleAndLegend** 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

```
VcNet1.Printer.Alignment = vcPTopLeft
```

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 tatsächlichen Zoomfaktor in Prozent für die Einstellung **FitToPage = False** erfragen (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

Code-Beispiel

```
VcNet1.Printer.CuttingMarks = True
```

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

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: " "

FitToPage

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die über die Eigenschaften **MaxHorizontalPagesCount** und **MaxVerticalPagesCount** definierte Anzahl von Seiten gedruckt werden soll (True) oder ob das Diagramm in der mit der Eigenschaft **ZoomFactor** eingestellten Größe ausgegeben werden soll (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Diagramm wird auf eine definierte Anzahl von Seiten verteilt/wird in der voreingestellten Größe ausgegeben.

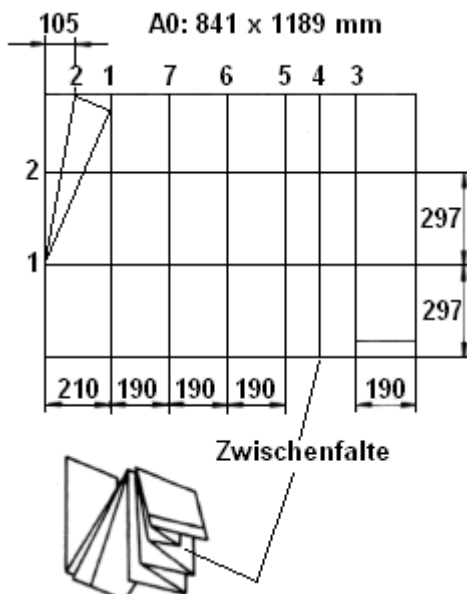
Code-Beispiel

```
VcNet1.Printer.FitToPage = True
```

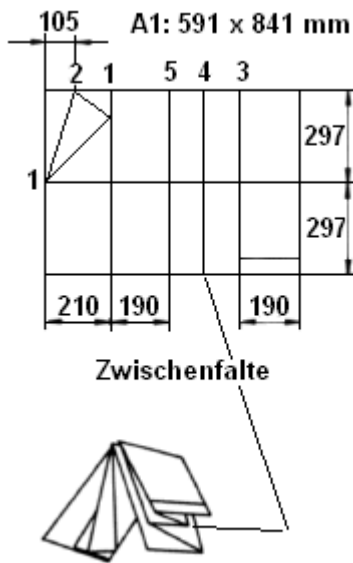
FoldingMarksType

Nur-Lese-Eigenschaft von VcPrinter

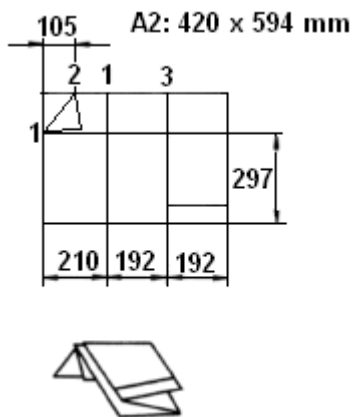
Mit dieser Eigenschaft können Sie folgende Faltmarkierungen nach DIN 824 für den Ausdruck festlegen oder erfragen. Diese ermöglichen das standardisierte Falten für DIN-A-Blattgrößen:



Faltung des DIN-A-0 Formats

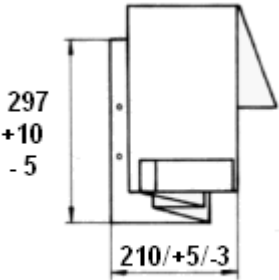
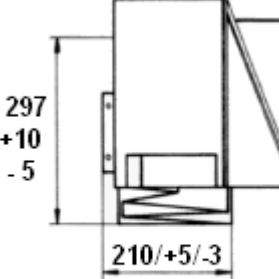
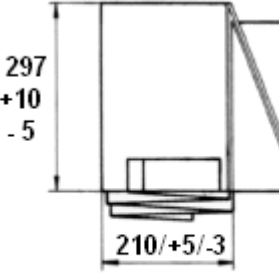


Faltung des DIN-A-1 Formats



Faltung des DIN-A-2 Formats

	Datentyp	Beschreibung
Eigenschaftswert	FoldingMarksTypeEnum	Faltmarkierungen Standardwert: vcFMTNone
	Mögliche Werte:	

vcFMTDIN824FormA 65	Ausgabe von Faltmarkierungen nach DIN824-A: Die gefaltete Zeichnung kann gelocht und ohne Heftstreifen abgeheftet werden.
	
	Faltung nach DIN 824-A
vcFMTDIN824FormB 66	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.
	
	Faltung nach DIN 824-C
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)

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

```
VcNet1.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

```
VcNet1.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 Mögliche Werte: vcLandscape 42 vcPortrait 41	Ausrichtung Standardwert: VcPortrait Querformat Hochformat

Code-Beispiel

```
VcNet1.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

Code-Beispiel

```
VcNet1.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 ""

Code-Beispiel

```
VcNet1.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

Code-Beispiel

```
VcNet1.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	"Seite N von M Seiten"

vcPRowColumn 1596	"x.y" (Zeilennummer.Spaltennummer)
-------------------	------------------------------------

Code-Beispiel

```
Dim printer As VcPrinter

Set printer = VcNet1.printer

With printer
    .Orientation = vcLandscape
    .PageNumberMode = vcPageNOfM
    .PageNumbers = True
    .FitToPage = False
End With

VcNet1.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

Code-Beispiel

```
VcNet1.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

Code-Beispiel

```
VcNet1.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

```
VcNet1.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

Code-Beispiel

```
VcNet1.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

RepeatTitleAndLegend

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Titel und Legende auf jeder Seite erscheinen sollen (True) oder nicht (False). Außerdem wird hiermit festgelegt, ob die Seitenaufteilung automatisch berechnet wird, so dass die Knoten nicht durchgeschnitten werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Titel und Legende werden auf jeder Seite wiederholt (True)/ Titel und Legende werden nur einmal ausgegeben und ggf. beim Seitenumbruch durchtrennt (False) Standardwert: False

Code-Beispiel

```
VcNet1.Printer.RepeatTitleAndLegend = True
```

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)

Code-Beispiel

```
Dim printer As VcPrinter
Set printer = VcNet1.printer
With printer
    .Orientation = vcLandscape
    .StartUpSinglePage = True
    .FitToPage = False
End With
VcNet1.PrintPreview
```

ZoomFactorAsDouble**Eigenschaft von VcPrinter**

Mit dieser Eigenschaft können Sie den Zoomfaktor in Prozent für die Einstellung **FitToPage = False** setzen oder erfragen (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Zoomfaktor für das Diagramm Standardwert: 100

Code-Beispiel

```
VcNet1.Printer.ZoomFactorAsDouble = 150
```

7.51 VcRect

Rect

Ein Objekt vom Typ **VcRect** bezeichnet ein Rechteck-Objekt und wird nur im Ereignis VcNet.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 VcNet1_OnShowInPlaceEditor(ByVal editObject As Object, _
    ByVal editObjectType As _
    VcNetLib.VcObjectTypeEnum, _
    ByVal fieldIndex As Long, ByVal objRectComplete As _
    VcNetLib.VcRect, ByVal objRectVisible As _
    VcNetLib.VcRect, ByVal fldRectComplete As _
    VcNetLib.VcRect, ByVal fldRectVisible As _
    VcNetLib.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 + VcNet1.Left
                Text1.Top = fldRectVisible.Top + VcNet1.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 + VcNet1.Left
                MonthView1.Top = fldRectVisible.Top + VcNet1.Top

                MonthView1.Value = editObject.DataField(fieldIndex)
                MonthView1.Visible = True
                MonthView1.SetFocus

            Case 13 'Employee
                Combol.Left = fldRectVisible.Left + VcNet1.Left
                Combol.Top = fldRectVisible.Top + VcNet1.Top
                Combol.Width = fldRectVisible.Width

                Combol.Text = editObject.DataField(fieldIndex)
                Combol.Visible = True
                Combol.SetFocus

        End Select

        Me.ScaleMode = oldScaleMode
    End Sub
```


736 API-Referenz: VcRect

```
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 + VcNet1.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
```

7.52 VcScheduler

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

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

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.

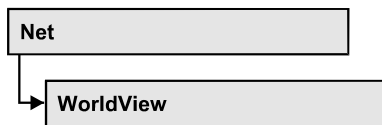
744 API-Referenz: VcScheduler

	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)
```

7.53 VcWorldView



Ein Objekt vom Typ **VcWorldView** bezeichnet das Komplettansicht-Fenster.

Eigenschaften

- Border
- 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

Code-Beispiel

```
VcNet1.WorldView.Mode = vcNotFixed
VcNet1.WorldView.Border = True
```

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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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. 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. 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. 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. 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.
vcFixedAtRight 2	
vcFixedAtTop 3	
vcNotFixed 5	
vcPopupWindow 6	

Code-Beispiel

```
VcNet1.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 (VcNet1.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	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig.
	vcHorizontalScrollBar 1	Anzeige einer horizontalen Bildlaufleiste, wenn nötig.
	vcNoScrollBar 0	Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt.
	vcVerticalScrollBar 2	Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcNet1.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

```
VcNet1.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

```
VcNet1.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

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

Code-Beispiel

```
VcNet1.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

```
VcNet1.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

```
VcNet1.LegendView.WidthActualValue = 600
```

8 Index

_NewEnum

Eigenschaft von

- DataObjectFiles* 299
- VcBoxCollection* 322
- VcBoxFormat* 328
- VcBoxFormatCollection* 333
- VcCalendarCollection* 357
- VcCalendarProfileCollection* 367
- VcDataDefinitionTable* 373
- VcDataRecordCollection* 384
- VcDataTableCollection* 393
- VcDataTableFieldCollection* 405
- VcFilter* 415
- VcFilterCollection* 421
- VcGroupCollection* 438
- VcIntervalCollection* 450
- VcLinkAppearanceCollection* 476
- VcLinkCollection* 482
- VcLinkFormat* 486
- VcLinkFormatCollection* 490
- VcMap* 500
- VcMapCollection* 507
- VcNodeAppearanceCollection* 683
- VcNodeCollection* 689
- VcNodeFormat* 693
- VcNodeFormatCollection* 698

A

AboutBox

Methode von

- VcNet* 574

About-Box 575

AbsoluteBottomMarginInCM

Eigenschaft von

- VcPrinter* 718

AbsoluteBottomMarginInInches

Eigenschaft von

- VcPrinter* 718

AbsoluteLeftMarginInCM

Eigenschaft von

- VcPrinter* 719

AbsoluteLeftMarginInInches

Eigenschaft von

- VcPrinter* 719

AbsoluteRightMarginInCM

Eigenschaft von

- VcPrinter* 719

AbsoluteRightMarginInInches

Eigenschaft von

- VcPrinter* 720

AbsoluteTopMarginInCM

Eigenschaft von

- VcPrinter* 720

AbsoluteTopMarginInInches

Eigenschaft von

- VcPrinter* 721

Active

Eigenschaft von

- VcCalendarCollection* 358

ActiveNodeFilter

Eigenschaft von

- VcNet* 527

ActualEndDateDataFieldIndex

Eigenschaft von

- VcScheduler* 737

ActualStartDateDataFieldIndex

- Eigenschaft von
 - VcScheduler 738
- Add**
 - Methode von
 - DataObjectFiles 301
 - VcBoxCollection 323
 - VcBoxFormatCollection 334
 - VcCalendarCollection 359
 - VcCalendarProfileCollection 367
 - VcDataRecordCollection 385
 - VcDataTableCollection 394
 - VcDataTableFieldCollection 406
 - VcFilterCollection 423
 - VcIntervalCollection 450
 - VcLinkAppearanceCollection 477
 - VcLinkFormatCollection 491
 - VcMapCollection 508
 - VcNodeAppearanceCollection 684
 - VcNodeFormatCollection 699
- AddBySpecification**
 - Methode von
 - VcBoxCollection 324
 - VcBoxFormatCollection 335
 - VcCalendarCollection 359
 - VcCalendarProfileCollection 368
 - VcFilterCollection 423
 - VcIntervalCollection 451
 - VcLinkAppearanceCollection 478
 - VcLinkFormatCollection 492
 - VcMapCollection 508
 - VcNodeAppearanceCollection 685
 - VcNodeFormatCollection 700
- AddDuration**
 - Methode von
 - VcCalendar 351
- AddSubCondition**
 - Methode von
 - VcFilter 418
- Alignment**
 - Eigenschaft von
 - VcBoundingBox 303
 - VcBoxFormatField 339
 - VcLinkFormatField 496
 - VcNodeFormatField 705
 - VcPrinter 721
- AllData**
 - Eigenschaft von
 - VcDataRecord 378
 - VcLink 462
 - VcNode 655
- AllowMultipleBoxMarking**
 - Eigenschaft von
 - VcNet 527
- AllowNewNodesAndLinks**
 - Eigenschaft von
 - VcNet 528
- Anordnen 273**
- Anschlussknoten 553**
- Arbeitsperiode**
 - Ende 353
- Arbeitszeitelemente**
 - berechnen 352
- Arrange**
 - Methode von
 - VcNet 575
- AssignCalendarToNodes**
 - Eigenschaft von
 - VcNet 528
- Ausgabeformat double 536**
- auslesen 732**
- Auslieferung 14**
- AutomaticSchedulingEnabled**
 - Eigenschaft von
 - VcScheduler 738

B

BackColor

Eigenschaft von
VcGroup 431

BackColorAsARGB

Eigenschaft von
VcNodeAppearance 662

BackColorDataFieldIndex

Eigenschaft von
VcNodeAppearance 663

BackColorMapName

Eigenschaft von
VcNodeAppearance 663

Bandnummern

in Fensterkoordinaten konvertieren
595
konvertieren aus Fensterkoordinaten
591, 592

Border

Eigenschaft von
VcLegendView 454
VcWorldView 745

BorderArea

Eigenschaft von
VcNet 529
siehe auch
VcBorderArea 302

BorderBox

Ausrichtung 303
Methode von
VcBorderArea 302
siehe auch
VcBorderBox 303

Borland Delphi 281

Bottom

Eigenschaft von
VcRect 734

BottomMargin

Eigenschaft von
VcNodeFormatField 705

Box

Farbe der Randlinie 312
Linienstärke 312, 433
markieren 314
Offset 319, 320
Priorität 316
Referenzpunkt 316
sichtbar 318
siehe auch
VcBox 310
Spezifikation 317
Typ der Randlinie 313
über Index 324
Ursprungspunkt 315
verschiebbar 314

Box Format

über Index 336

BoxByIndex

Methode von
VcBoxCollection 324

BoxByName

Methode von
VcBoxCollection 325

BoxCollection

Eigenschaft von
VcNet 529
siehe auch
VcBoxCollection 322

Boxen 85

Ausdehnung 318
Boxen bearbeiten 207
Boxen verwalten 204
Boxformat bearbeiten 211
Name 315

Name UpdateBehavior 317
Offset 319
Offset in Pixel umrechnen 319
Pixel in Offset umrechnen 321
Textfeld 311

BoxFormat

siehe auch
VcBoxFormat 328

BoxFormatCollection

Eigenschaft von
VcNet 529
siehe auch
VcBoxFormatCollection 333

Boxformate

Name 330
verwalten 209

Boxformatfeld

Ausrichtung 339
Hintergrundfarbe 343
Höhe Grafik 340
Index 341
maximale Zeilenzahl 341
Mindestbreite 342
minimale Zeilenzahl 342
Muster 344
Musterfarbe 343
Name des Formats 340
Schriftart 347
Schriftfarbe 347
Typ 348

BoxFormatField

siehe auch
VcBoxFormatField 339

Browser 11, 21

C

CalcDuration

Methode von
VcCalendar 352

Calendar

siehe auch
VcCalendar 349

CalendarByIndex

Methode von
VcCalendarCollection 360

CalendarByName

Methode von
VcCalendarCollection 360

CalendarCollection

Eigenschaft von
VcNet 530
siehe auch
VcCalendarCollection 357

CalendarProfile

siehe auch
VcCalendarProfile 363

CalendarProfileByIndex

Methode von
VcCalendarProfileCollection 368

CalendarProfileByName

Methode von
VcCalendarProfileCollection 369

CalendarProfileCollection

Eigenschaft von
VcCalendar 350
VcNet 530
siehe auch
VcCalendarProfileCollection 366

CalendarProfileName

Eigenschaft von
VcInterval 443

Clear

Methode von
DataObject 294

- DataObjectFiles* 301
- VcCalendar* 352
- VcNet* 575
- Clustering** 109, 162, 544
- Collapsed**
 - Eigenschaft von
 - VcGroup* 432
- ColorAsARGB**
 - Eigenschaft von
 - VcMapEntry* 513
- CombiField**
 - Eigenschaft von
 - VcNodeFormatField* 706
- ComparisonValueAsString**
 - Eigenschaft von
 - VcFilterSubCondition* 427
- ConfigurationName**
 - Eigenschaft von
 - VcNet* 530
- ConnectionOperator**
 - Eigenschaft von
 - VcFilterSubCondition* 428
- ConsiderFilterEntries**
 - Eigenschaft von
 - VcMap* 501
- ConstantText**
 - Eigenschaft von
 - VcLinkFormatField* 497
 - VcNodeFormatField* 706
- Copy**
 - Methode von
 - VcBoxCollection* 325
 - VcBoxFormatCollection* 335
 - VcCalendarCollection* 360
 - VcCalendarProfileCollection* 369
 - VcDataTableCollection* 395
 - VcDataTableFieldCollection* 407
- VcFilterCollection* 424
- VcIntervalCollection* 451
- VcLinkAppearanceCollection* 478
- VcLinkFormatCollection* 493
- VcMapCollection* 509
- VcNodeAppearanceCollection* 685
- VcNodeFormatCollection* 701
- CopyFormatField**
 - Methode von
 - VcBoxFormat* 331
 - VcLinkFormat* 488
 - VcNodeFormat* 696
- CopyNodesIntoClipboard**
 - Methode von
 - VcNet* 575
- CopySubCondition**
 - Methode von
 - VcFilter* 418
- Count**
 - Eigenschaft von
 - DataObjectFiles* 300
 - VcBoxCollection* 323
 - VcBoxFormatCollection* 334
 - VcCalendarCollection* 359
 - VcCalendarProfileCollection* 367
 - VcDataDefinitionTable* 374
 - VcDataRecordCollection* 384
 - VcDataTableCollection* 394
 - VcDataTableFieldCollection* 406
 - VcFilterCollection* 422
 - VcGroupCollection* 439
 - VcIntervalCollection* 450
 - VcLinkAppearanceCollection* 477
 - VcLinkCollection* 483
 - VcLinkFormatCollection* 491
 - VcMap* 501
 - VcMapCollection* 507

VcNodeAppearanceCollection 684

VcNodeCollection 690

VcNodeFormatCollection 699

CreateDataField

Methode von

VcDataDefinitionTable 374

CreateEntry

Methode von

VcMap 503

CSV-Dateien

Aufbau 15

Verwendung 16

CtrlCXVProcessing

Eigenschaft von

VcNet 531

CurrentHorizontalPagesCount

Eigenschaft von

VcPrinter 722

CurrentVersion

Eigenschaft von

VcNet 532

CurrentVerticalPagesCount

Eigenschaft von

VcPrinter 722

CurrentZoomFactor

Eigenschaft von

VcPrinter 722

CutNodesIntoClipboard

Methode von

VcNet 576

CuttingMarks

Eigenschaft von

VcPrinter 723

D

DataDefinition

Eigenschaft von

VcNet 532

siehe auch

VcDataDefinition 372

DataDefinitionTable

Eigenschaft von

VcFilter 415

siehe auch

VcDataDefinitionTable 373

DataField

Eigenschaft von

VcDataRecord 379

VcLink 463

VcNode 656

DataFieldIndex

Eigenschaft von

VcFilterSubCondition 429

DataFieldValue

Eigenschaft von

VcMapEntry 514

DataObject 293

Clear 294

Files 293

GetData 294

GetFormat 296

SetData 297

DataObjectFiles 299

_NewEnum 299

Add 301

Clear 301

Count 300

Item 300

Remove 301

DataRecord

Methode von

VcLink 465

VcNode 658

siehe auch

VcDataRecord 378

DataRecordByID

Methode von

VcDataRecordCollection 386

DataRecordCollection

Eigenschaft von

VcDataTable 390

siehe auch

VcDataRecordCollection 383

DataTable

siehe auch

VcDataTable 390

DataTableByIndex

Methode von

VcDataTableCollection 396

DataTableByName

Methode von

VcDataTableCollection 396

DataTableCollection

Eigenschaft von

VcNet 533

siehe auch

VcDataTableCollection 393

DataTableField

siehe auch

VcDataTableField 399

DataTableFieldByIndex

Methode von

VcDataTableFieldCollection 407

DataTableFieldByName

Methode von

VcDataTableFieldCollection 408

DataTableFieldCollection

Eigenschaft von

VcDataTable 391

siehe auch

VcDataTableFieldCollection 405

DataTableName

Eigenschaft von

VcDataRecord 380

VcDataTableField 400

DateFormat

Eigenschaft von

VcDataTableField 400

VcDefinitionField 410

Dateiliste 294

Dateinamen

Anzahl 300

entfernen 301

hinzufügen 301

Index 300

löschen 301

Daten 89

aus Datei einlesen 42

bearbeiten 259

laden 590

speichern 596

zum DataObject hinzufügen 298

Datenausgabeformat 534

Datenaustausch mit VARCHAR XNet 15

Datendefinitionstabellen 372

Anzahl der Felder 374

Datumsformat eines Feldes 411

Feld zur Laufzeit hinzufügen 374

Index eines Feldes 412

Name eines Feldes 412

Typ eines Feldes 413

Zugriff auf Feld über Index 375

Zugriff auf Feld über Namen 376

Datenfeld

editierbar 411

für Toolliptext 166, 563

versteckt 411

Datenfelder

- Knoten 259
- Verbindung 261

Datensatz

- abhängiger Datensatz nicht gefunden 616
- aktualisieren 382
- Aktualisierung 388
- alle Daten 379
- Anzahl in Collection 384
- aus Collection entfernen 388
- datenbasiertes Objekt 381
- Datenfeld 379
- eindeutige ID 387
- Ereignis Veränderung 615
- Ereignis Veränderung abgeschlossen 616
- erzeugen 612, 613
- ID 380
- Iteration, Enumerator Objekt 384
- Iteration, Erstwert 386
- Iteration, Folgewert 387
- löschen 380, 614, 615
- Name der zugehörigen Tabelle 380
- über ID** 386
- zu Collection hinzufügen 385
- zugeordneter Datensatz 382

Datentabelle

- Aktualisierung 398
- Anzahl in Collection 394
- Auflistungsobjekt 533
- Beschreibung 391
- Datensatz-Collection 390
- Erweiterte Datentabellen setzen 539
- für Knoten 165
- innerhalb der Collection kopieren 395
- Iteration, Enumerator Objekt 394

- Iteration, Erstwert 397
- Iteration, Folgewert 397
- Name 577
- Name 392
- Tabellendatenfeld-Collection 391
- über Index 396
- über Name 396
- zu Collection hinzufügen 395

Datentabellen 90

- verwalten 180

Datentabellenfeld

- Datentyp 404
- Datumsformat 400
- editierbar 401
- Index 578
- Index 402
- Index des Bezugfeldes 403
- Name 577
- Name 402
- Primärschlüssel 402
- versteckt 401
- zugehöriger Tabellenname 400

DateOutputFormat

- Eigenschaft von*
- VcNet 533

DatesWithHourAndMinute

- Eigenschaft von*
- VcFilter 416

DayInEndMonth

- Eigenschaft von*
- VcInterval 443

DayInStartMonth

- Eigenschaft von*
- VcInterval 443

DefaultPrinterName

- Eigenschaft von*
- VcPrinter 723

DefinitionField

siehe auch

VcDefinitionField 410

DefinitionTable*Eigenschaft von*

VcDataDefinition 372

DeleteDataRecord*Methode von*

VcDataRecord 380

DeleteEntry*Methode von*

VcMap 503

DeleteLink*Methode von*

VcLink 465

DeleteLinkRecord*Methode von*

VcNet 576

DeleteNode*Methode von*

VcNode 659

DeleteNodeRecord*Methode von*

VcNet 577

Description*Eigenschaft von*

VcDataTable 391

DetectDataTableFieldName*Methode von*

VcNet 577

DetectDataTableName*Methode von*

VcNet 577

DetectFieldIndex*Methode von*

VcNet 578

DiagramBackColor*Eigenschaft von*

VcNet 535

Diagramm

alle Objekte löschen 575

Ausrichtung 266

exportieren 83, 274

Hintergrundfarbe 535

in Datei speichern 582

Layoutberechnung 575

speichern 599

stets komplett darstellen 598

Dialogfeld

Druckvorschau 269

Seite einrichten 265

Verbindung bearbeiten 261

Vorgänge bearbeiten 259

Zuordnung einstellen 194

Dialogfelder

Schriftattribute 535

DialogFont*Eigenschaft von*

VcNet 535

DocumentName*Eigenschaft von*

VcPrinter 723

DoubleFeature*Eigenschaft von*

VcNodeAppearance 664

DoubleOutputFormat*Eigenschaft von*

VcNet 535

Druckdatum 268**Drucken 82, 273**Absolute Breite des linken
Seitenrandes in cm 719Absolute Breite des linken
Seitenrandes in Zoll 719

Absolute Breite des rechten
 Seitenrandes in cm 719
 Absolute Breite des rechten
 Seitenrandes in Zoll 720
 Absolute Höhe des oberen
 Seitenrandes in cm 720
 Absolute Höhe des oberen
 Seitenrandes in Zoll 721
 Absolute Höhe des unteren
 Seitenrandes in cm 718
 Absolute Höhe des unteren
 Seitenrandes in Zoll 718
 aktueller Drucker 723
 an Seitenzahlvorgabe anpassen 266
 Art der Seitennummerierung 729
 Ausdruck des Diagramms auf
 definierte Anzahl von Seiten 724
 auslösen 594
 Ausrichtung 721
 direkt 592
 Dokumentenname 723
 Druckdatum 731
 Drucker einrichten 273, 593
 Druckereigenschaften setzen bzw.
 ausfragen 567
 Druckernamen festlegen 732
 Druckvorschau 594
 Faltmarkierungen 725
 Hoch- oder Querformat 728
 in eine Datei 594
 max. Anzahl von Seiten (horizontal)
 727
 max. Anzahl von Seiten (vertikal) 727
 Papiergröße 731
 Probleme 286
 Rahmen 729
 Schnittmarkierungen 723
 Seitenbeschriftung 728, 729
 Seitennummern 730

Titel und Legende auf jeder Seite
 732

Zoomfaktor 266, 722, 733

Druckvorschau 269, 273, 732

DumpConfiguration

Methode von

VcNet 578

DurationDataFieldIndex

Eigenschaft von

VcScheduler 738

E

EarlyEndDateDataFieldIndex

Eigenschaft von

VcScheduler 739

EarlyStartDateDataFieldIndex

Eigenschaft von

VcScheduler 739

Editable

Eigenschaft von

VcDataTableField 401

VcDefinitionField 411

Editieren

direkt in Knotenfeldern 553

EditLink

Methode von

VcNet 579

EditNewLink

Eigenschaft von

VcNet 536

EditNewNode

Eigenschaft von

VcNet 536

EditNode

Methode von

VcNet 579

Eigenschaften

- _NewEnum*
 - DataObjectFiles* 299
 - VcBoxCollection* 322
 - VcBoxFormat* 328
 - VcBoxFormatCollection* 333
 - VcCalendarCollection* 357
 - VcCalendarProfileCollection* 367
 - VcDataDefinitionTable* 373
 - VcDataRecordCollection* 384
 - VcDataTableCollection* 393
 - VcDataTableFieldCollection* 405
 - VcFilter* 415
 - VcFilterCollection* 421
 - VcGroupCollection* 438
 - VcIntervalCollection* 450
 - VcLinkAppearanceCollection* 476
 - VcLinkCollection* 482
 - VcLinkFormat* 486
 - VcLinkFormatCollection* 490
 - VcMap* 500
 - VcMapCollection* 507
 - VcNodeAppearanceCollection* 683
 - VcNodeCollection* 689
 - VcNodeFormat* 693
 - VcNodeFormatCollection* 698
- AbsoluteBottomMarginInCM*
 - VcPrinter* 718
- AbsoluteBottomMarginInInches*
 - VcPrinter* 718
- AbsoluteLeftMarginInCM*
 - VcPrinter* 719
- AbsoluteLeftMarginInInches*
 - VcPrinter* 719
- AbsoluteRightMarginInCM*
 - VcPrinter* 719
- AbsoluteRightMarginInInches*
 - VcPrinter* 720
- AbsoluteTopMarginInCM*
 - VcPrinter* 720
- AbsoluteTopMarginInInches*
 - VcPrinter* 721
- Active*
 - VcCalendarCollection* 358
- ActiveNodeFilter*
 - VcNet* 527
- ActualEndDateDataFieldIndex*
 - VcScheduler* 737
- ActualStartDateDataFieldIndex*
 - VcScheduler* 738
- Alignment*
 - VcBoundingBox* 303
 - VcBoxFormatField* 339
 - VcLinkFormatField* 496
 - VcNodeFormatField* 705
 - VcPrinter* 721
- AllData*
 - VcDataRecord* 378
 - VcLink* 462
 - VcNode* 655
- AllowMultipleBoxMarking*
 - VcNet* 527
- AllowNewNodesAndLinks*
 - VcNet* 528
- AssignCalendarToNodes*
 - VcNet* 528
- AutomaticSchedulingEnabled*
 - VcScheduler* 738
- BackColor*
 - VcGroup* 431
- BackColorAsARGB*
 - VcNodeAppearance* 662
- BackColorDataFieldIndex*
 - VcNodeAppearance* 663
- BackColorMapName*

- VcNodeAppearance* 663
- Border*
 - VcLegendView* 454
 - VcWorldView* 745
- BorderArea*
 - VcNet* 529
- Bottom*
 - VcRect* 734
- BottomMargin*
 - VcNodeFormatField* 705
- BoxCollection*
 - VcNet* 529
- BoxFormatCollection*
 - VcNet* 529
- CalendarCollection*
 - VcNet* 530
- CalendarProfileCollection*
 - VcCalendar* 350
 - VcNet* 530
- CalendarProfileName*
 - VcInterval* 443
- Collapsed*
 - VcGroup* 432
- ColorAsARGB*
 - VcMapEntry* 513
- CombiField*
 - VcNodeFormatField* 706
- ComparisonValueAsString*
 - VcFilterSubCondition* 427
- ConfigurationName*
 - VcNet* 530
- ConnectionOperator*
 - VcFilterSubCondition* 428
- ConsiderFilterEntries*
 - VcMap* 501
- ConstantText*
 - VcLinkFormatField* 497
- VcNodeFormatField* 706
- Count*
 - DataObjectFiles* 300
 - VcBoxCollection* 323
 - VcBoxFormatCollection* 334
 - VcCalendarCollection* 359
 - VcCalendarProfileCollection* 367
 - VcDataDefinitionTable* 374
 - VcDataRecordCollection* 384
 - VcDataTableCollection* 394
 - VcDataTableFieldCollection* 406
 - VcFilterCollection* 422
 - VcGroupCollection* 439
 - VcIntervalCollection* 450
 - VcLinkAppearanceCollection* 477
 - VcLinkCollection* 483
 - VcLinkFormatCollection* 491
 - VcMap* 501
 - VcMapCollection* 507
 - VcNodeAppearanceCollection* 684
 - VcNodeCollection* 690
 - VcNodeFormatCollection* 699
- CtrlCXVProcessing*
 - VcNet* 531
- CurrentHorizontalPagesCount*
 - VcPrinter* 722
- CurrentVersion*
 - VcNet* 532
- CurrentVerticalPagesCount*
 - VcPrinter* 722
- CurrentZoomFactor*
 - VcPrinter* 722
- CuttingMarks*
 - VcPrinter* 723
- DataDefinition*
 - VcNet* 532
- DataDefinitionTable*

- VcFilter* 415
- DataField*
 - VcDataRecord* 379
 - VcLink* 463
 - VcNode* 656
- DataFieldIndex*
 - VcFilterSubCondition* 429
- DataFieldValue*
 - VcMapEntry* 514
- DataRecordCollection*
 - VcDataTable* 390
- DataTableCollection*
 - VcNet* 533
- DataTableFieldCollection*
 - VcDataTable* 391
- DataTableName*
 - VcDataRecord* 380
 - VcDataTableField* 400
- DateFormat*
 - VcDataTableField* 400
 - VcDefinitionField* 410
- DateOutputFormat*
 - VcNet* 533
- DatesWithHourAndMinute*
 - VcFilter* 416
- DayInEndMonth*
 - VcInterval* 443
- DayInStartMonth*
 - VcInterval* 443
- DefaultPrinterName*
 - VcPrinter* 723
- DefinitionTable*
 - VcDataDefinition* 372
- Description*
 - VcDataTable* 391
- DiagramBackColor*
 - VcNet* 535
- DialogFont*
 - VcNet* 535
- DocumentName*
 - VcPrinter* 723
- DoubleFeature*
 - VcNodeAppearance* 664
- DoubleOutputFormat*
 - VcNet* 535
- DurationDataFieldIndex*
 - VcScheduler* 738
- EarlyEndDateDataFieldIndex*
 - VcScheduler* 739
- EarlyStartDateDataFieldIndex*
 - VcScheduler* 739
- Editable*
 - VcDataTableField* 401
 - VcDefinitionField* 411
- EditNewLink*
 - VcNet* 536
- EditNewNode*
 - VcNet* 536
- Enabled*
 - VcNet* 537
- EnableSupplyTextEntryEvent*
 - VcNet* 537
- EndDateForAutomaticScheduling*
 - VcScheduler* 739
- EndDateNotLaterThanDataFieldIndex*
 - VcScheduler* 740
- EndTime*
 - VcInterval* 444
- EndMonth*
 - VcInterval* 444
- EndTime*
 - VcInterval* 444
- EndWeekday*
 - VcInterval* 445

- EventReturnStatus*
 - VcNet* 537
- EventText*
 - VcNet* 538
- ExtendedDataTables*
 - VcNet* 538
- FieldsSeparatedByLines*
 - VcBoxFormat* 329
 - VcNodeFormat* 694
- FieldText*
 - VcBox* 311
- FilePath*
 - VcNet* 539
- Files*
 - DataObject* 293
- FilterCollection*
 - VcNet* 540
- FilterName*
 - VcFilterSubCondition* 429
 - VcLinkAppearance* 467
 - VcNodeAppearance* 664
- FitToPage*
 - VcPrinter* 724
- FoldingMarksType*
 - VcPrinter* 724
- FontAntiAliasingEnabled*
 - VcNet* 540
- FontBody*
 - VcMapEntry* 514
- FontName*
 - VcMapEntry* 515
- FontSize*
 - VcMapEntry* 515
- FormatField*
 - VcBoxFormat* 329
 - VcLinkFormat* 487
 - VcNodeFormat* 694
- FormatFieldCount*
 - VcBoxFormat* 330
 - VcLinkFormat* 487
 - VcNodeFormat* 695
- FormatName*
 - VcBox* 311
 - VcBoxFormatField* 340
 - VcLinkAppearance* 468
 - VcLinkFormatField* 497
 - VcNodeAppearance* 665
 - VcNodeFormatField* 706
- FrameAroundFieldsVisible*
 - VcNodeAppearance* 665
- FrameShape*
 - VcNodeAppearance* 666
- FreeFloatDataFieldIndex*
 - VcScheduler* 740
- GraphicsFileName*
 - VcBorderBox* 304
 - VcMapEntry* 516
 - VcNodeFormatField* 706
- GraphicsFileNameDataFieldIndex*
 - VcNodeFormatField* 707
- GraphicsFileNameMapName*
 - VcNodeFormatField* 707
- GraphicsHeight*
 - VcBoxFormatField* 340
 - VcNodeFormatField* 707
- GroupCollection*
 - VcNet* 541
- GroupDescriptionName*
 - VcNet* 541
- GroupField*
 - VcNet* 542
- GroupHorizontalMargin*
 - VcNet* 542
- Grouping*

- VcNet* 542
- GroupingTitlesFullyVisible*
 - VcNet* 543
- GroupInteractionsAllowed*
 - VcNet* 543
- GroupMode*
 - VcNet* 544
- GroupMovingAllowed*
 - VcNet* 544
- GroupSortField*
 - VcNet* 545
- GroupSortMode*
 - VcNet* 545
- GroupTitleField*
 - VcNet* 546
- GroupVerticalMargin*
 - VcNet* 546
- Height*
 - VcLegendView* 455
 - VcRect* 734
 - VcWorldView* 746
- HeightActualValue*
 - VcLegendView* 455
 - VcWorldView* 746
- Hidden*
 - VcDataTableField* 401
 - VcDefinitionField* 411
- hWnd*
 - VcNet* 547
- ID*
 - VcDataRecord* 380
 - VcDefinitionField* 412
 - VcLink* 463
 - VcNode* 656
- IncomingLinks*
 - VcNode* 657
- Index*
 - VcBoxFormatField* 341
 - VcDataTableField* 402
 - VcFilterSubCondition* 429
 - VcLinkFormatField* 497
 - VcNodeFormatField* 708
 - InFlowGroupDescriptionName*
 - VcNet* 547
 - InFlowGroupField*
 - VcNet* 548
 - InFlowGroupingEnabled*
 - VcNet* 548
 - InFlowGroupSeparationLineColor*
 - VcNet* 548
 - InFlowGroupSeparationLineType*
 - VcNet* 549
 - InFlowGroupTimeInterval*
 - VcNet* 550
 - InFlowGroupTitleField*
 - VcNet* 550
 - InFlowGroupTitlesBackColor*
 - VcNet* 551
 - InFlowGroupTitlesFont*
 - VcNet* 551
 - InFlowGroupTitlesVisibleAtBottomOrRight*
 - VcNet* 551
 - InFlowGroupTitlesVisibleAtTopOrLeft*
 - VcNet* 551
 - InFlowGroupTitleTimeFormat*
 - VcNet* 552
 - InFlowGroupVerticalCaptionWidth*
 - VcNet* 552
 - InPlaceEditingAllowed*
 - VcNet* 552
 - InteractionMode*
 - VcNet* 553
 - InterfaceNodesShown*

- VcNet 553
- IntervalCollection*
 - VcCalendar 350
 - VcCalendarProfile 363
- Item*
 - DataObjectFiles 300
- LateEndDateDataFieldIndex*
 - VcScheduler 740
- LateStartDateDataFieldIndex*
 - VcScheduler 740
- Left*
 - VcLegendView 456
 - VcRect 735
 - VcWorldView 747
- LeftActualValue*
 - VcLegendView 456
 - VcWorldView 747
- LeftMargin*
 - VcNodeFormatField 708
- LegendElementsArrangement*
 - VcBoundingBox 305
- LegendElementsBottomMargin*
 - VcBoundingBox 305
- LegendElementsMaximumColumnCount*
 - VcBoundingBox 305
- LegendElementsMaximumRowCount*
 - VcBoundingBox 306
- LegendElementsTopMargin*
 - VcBoundingBox 306
- LegendFont*
 - VcBoundingBox 306
- LegendText*
 - VcNodeAppearance 667
- LegendTitle*
 - VcBoundingBox 306
- LegendTitleFont*
- VcBoundingBox 307
- LegendTitleVisible*
 - VcBoundingBox 307
- LegendView*
 - VcNet 554
- LineColor*
 - VcBox 312
 - VcGroup 432
 - VcLinkAppearance 468
 - VcNodeAppearance 668
- LineColorDataFieldIndex*
 - VcNodeAppearance 668
- LineColorMapName*
 - VcNodeAppearance 668
- LineThickness*
 - VcBox 312
 - VcGroup 433
 - VcLinkAppearance 469
 - VcNodeAppearance 669
- LineType*
 - VcBox 313
 - VcGroup 434
 - VcLinkAppearance 470
 - VcNodeAppearance 670
- LinkAnnotationColumnNumberDataFieldIndex*
 - VcNet 554
- LinkAnnotationRowNumberDataFieldIndex*
 - VcNet 554
- LinkAppearanceCollection*
 - VcNet 555
- LinkCollection*
 - VcNet 555
- LinkDurationDataFieldIndex*
 - VcScheduler 741
- LinkFormatCollection*

- VcNet* 555
- LinkPredecessorDataFieldIndex*
 - VcNet* 556
- LinksDataTableName*
 - VcNet* 556
- LinkSuccessorDataFieldIndex*
 - VcNet* 557
- LinkTypeDataFieldIndex*
 - VcNet* 558
- MapCollection*
 - VcNet* 558
- MarginsShownInInches*
 - VcPrinter* 726
- MarkBox*
 - VcBox* 314
- MarkedNodesFilter*
 - VcFilterCollection* 422
- MarkingColor*
 - VcWorldView* 748
- MarkLink*
 - VcLink* 464
- MarkNode*
 - VcNode* 657
- MaxHorizontalPagesCount*
 - VcPrinter* 727
- MaximumTextLineCount*
 - VcBoxFormatField* 341
 - VcNodeFormatField* 708
- MaxVerticalPagesCount*
 - VcPrinter* 727
- MinimumColumnWidth*
 - VcNet* 559
- MinimumRowHeight*
 - VcNet* 559
- MinimumTextLineCount*
 - VcBoxFormatField* 342
 - VcNodeFormatField* 709
- MinimumWidth*
 - VcBoxFormatField* 342
 - VcLinkFormatField* 498
 - VcNodeFormatField* 709
- Mode*
 - VcWorldView* 748
- MouseProcessingEnabled*
 - VcNet* 560
- Moveable*
 - VcBox* 314
- MultiplePrimaryKeysAllowed*
 - VcDataTable* 392
- Name*
 - VcBox* 315
 - VcBoxFormat* 330
 - VcCalendar* 350
 - VcCalendarProfile* 364
 - VcDataTable* 392
 - VcDataTableField* 402
 - VcDefinitionField* 412
 - VcFilter* 416
 - VcGroup* 435
 - VcInterval* 445
 - VcLinkAppearance* 471
 - VcLinkFormat* 488
 - VcMap* 502
 - VcNodeAppearance* 671
 - VcNodeFormat* 695
- NodeAppearanceCollection*
 - VcNet* 560
- NodeCalendarNameDataFieldIndex*
 - VcNet* 560
- NodeChangeRankToPredecessorRankDataFieldIndex*
 - VcNet* 561
- NodeCollection*
 - VcGroup* 435

- VcNet* 561
- NodeColumnNumberDataFieldIndex*
 - VcNet* 561
- NodeFormatCollection*
 - VcNet* 562
- NodeRowNumberDataFieldIndex*
 - VcNet* 562
- NodesDataTableName*
 - VcNet* 563
- NodeTooltipTextField*
 - VcNet* 563
- ObliqueTracksOnLinks*
 - VcNet* 564
- OLEDragMode*
 - VcNet* 564
- OLEDragWithOwnMouseCursor*
 - VcNet* 565
- OLEDragWithPhantom*
 - VcNet* 566
- OLEDropMode*
 - VcNet* 566
- Operator*
 - VcFilterSubCondition* 429
- Orientation*
 - VcNet* 567
 - VcPrinter* 728
- Origin*
 - VcBox* 315
- OutgoingLinks*
 - VcNode* 658
- PageDescription*
 - VcPrinter* 728
- PageDescriptionString*
 - VcPrinter* 728
- PageFrame*
 - VcPrinter* 729
- PageNumberMode*
 - VcPrinter* 729
- PageNumbers*
 - VcPrinter* 730
- PagePaddingEnabled*
 - VcPrinter* 730
- PaperSize*
 - VcPrinter* 731
- ParentHWnd*
 - VcLegendView* 457
 - VcWorldView* 749
- Pattern*
 - VcMapEntry* 517
 - VcNodeAppearance* 672
- PatternBackgroundColorAsARGB*
 - VcBoxFormatField* 343
 - VcNodeFormatField* 709
- PatternBackgroundColorDataFieldIndex*
 - VcNodeFormatField* 710
- PatternBackgroundColorMapName*
 - VcNodeFormatField* 710
- PatternColorAsARGB*
 - VcBoxFormatField* 343
 - VcNodeAppearance* 675
 - VcNodeFormatField* 711
- PatternColorDataFieldIndex*
 - VcNodeAppearance* 676
 - VcNodeFormatField* 711
- PatternColorMapName*
 - VcNodeAppearance* 676
 - VcNodeFormatField* 712
- PatternDataFieldIndex*
 - VcNodeAppearance* 676
- PatternEx*
 - VcBoxFormatField* 344
 - VcNodeFormatField* 712
- PatternExDataFieldIndex*

- VcNodeFormatField* 713
- PatternExMapName*
 - VcNodeFormatField* 713
- PatternMapName*
 - VcNodeAppearance* 677
- Piles*
 - VcNodeAppearance* 677
- PredecessorNode*
 - VcLink* 464
- PrePortSymbol*
 - VcLinkAppearance* 472
- PrimaryKey*
 - VcDataTableField* 402
- PrintDate*
 - VcPrinter* 731
- Printer*
 - VcNet* 567
- PrinterName*
 - VcPrinter* 732
- Priority*
 - VcBox* 316
- ReferencePoint*
 - VcBox* 316
- RelationshipFieldIndex*
 - VcDataTableField* 403
- RepeatTitleAndLegend*
 - VcPrinter* 732
- Right*
 - VcRect* 736
- RightMargin*
 - VcNodeFormatField* 713
- RoundedLinkSlantsEnabled*
 - VcNet* 568
- RoutingType*
 - VcLinkAppearance* 472
- ScheduledProjectEndDate*
 - VcScheduler* 741
- ScheduledProjectStartDate*
 - VcScheduler* 741
- Scheduler*
 - VcNet* 568
- ScheduleSuccessorsOnlyEnabled*
 - VcScheduler* 742
- ScrollBarMode*
 - VcLegendView* 457
 - VcWorldView* 750
- ScrollOffsetX*
 - VcNet* 568
- ScrollOffsetY*
 - VcNet* 569
- SecondsPerWorkday*
 - VcCalendar* 351
- Shadow*
 - VcNodeAppearance* 677
- ShadowColorAsARGB*
 - VcNodeAppearance* 678
- ShortenedLinks*
 - VcNet* 569
- ShowToolTip*
 - VcNet* 570
- Specification*
 - VcBox* 317
 - VcBoxFormat* 330
 - VcCalendar* 351
 - VcCalendarProfile* 364
 - VcFilter* 416
 - VcInterval* 445
 - VcLinkAppearance* 473
 - VcLinkFormat* 488
 - VcMap* 502
 - VcNodeAppearance* 679
 - VcNodeFormat* 695
- StartDateForAutomaticScheduling*
 - VcScheduler* 742

- StartDateNotEarlierThanDataFieldIndex*
 - VcScheduler* 742
- StartDateTime*
 - VcInterval* 446
- StartMonth*
 - VcInterval* 446
- StartTime*
 - VcInterval* 446
- StartUpSinglePage*
 - VcPrinter* 732
- StartWeekday*
 - VcInterval* 447
- StraightLinkDrawing*
 - VcNet* 570
- StrikeThrough*
 - VcNodeAppearance* 679
- StrikeThroughColor*
 - VcNodeAppearance* 680
- StringsCaseSensitive*
 - VcFilter* 417
- SubCondition*
 - VcFilter* 417
- SubConditionCount*
 - VcFilter* 417
- SuccessorNode*
 - VcLink* 464
- SuccPortSymbol*
 - VcLinkAppearance* 474
- Text*
 - VcBoundingBox* 308
- TextDataFieldIndex*
 - VcLinkFormatField* 498
 - VcNodeFormatField* 714
- TextFont*
 - VcBoundingBox* 308
 - VcBoxFormatField* 347
- VcLinkFormatField* 498
- VcNodeFormatField* 714
- TextFontColor*
 - VcBoxFormatField* 347
 - VcLinkFormatField* 499
 - VcNodeFormatField* 714
- TextFontDataFieldIndex*
 - VcNodeFormatField* 715
- TextFontMapName*
 - VcNodeFormatField* 715
- TextLineCount*
 - VcLinkFormatField* 499
- ThreeDEffect*
 - VcNodeAppearance* 681
- TimeUnit*
 - VcNet* 570
- Title*
 - VcGroup* 436
- TitleLineCount*
 - VcGroup* 436
- ToolTipChangeDuration*
 - VcNet* 571
- ToolTipDuration*
 - VcNet* 571
- ToolTipPointerDuration*
 - VcNet* 572
- ToolTipShowAfterClick*
 - VcNet* 572
- Top*
 - VcLegendView* 458
 - VcRect* 736
 - VcWorldView* 750
- TopActualValue*
 - VcLegendView* 458
 - VcWorldView* 751
- TopMargin*
 - VcNodeFormatField* 715

- TotalFloatDataFieldIndex*
- VcScheduler* 743
- Type*
- VcBoundingBox* 309
- VcBoxFormatField* 348
- VcCalendarProfile* 364
- VcDataTableField* 404
- VcDefinitionField* 413
- VcInterval* 447
- VcMap* 502
- VcNodeFormatField* 716
- UngroupedNodesAllowed*
- VcNet* 572
- UpdateBehaviorName*
- VcBox* 317
- VcWorldView* 751
- Visible*
- VcBox* 318
- VcLegendView* 459
- VcLinkAppearance* 474
- VcWorldView* 751
- VisibleInLegend*
- VcNodeAppearance* 681
- WaitCursorEnabled*
- VcNet* 573
- Width*
- VcLegendView* 459
- VcRect* 736
- VcWorldView* 752
- WidthActualValue*
- VcLegendView* 460
- VcWorldView* 752
- WidthOfExteriorSurrounding*
- VcNodeFormat* 696
- WindowMode*
- VcLegendView* 460
- WorldView*
- VcNet* 573
- X
- VcGroup* 437
- Y
- VcGroup* 437
- ZoomFactor*
- VcNet* 574
- ZoomFactorAsDouble*
- VcPrinter* 733
- ZoomingPerMouseWheelAllowed*
- VcNet* 574
- Eigenschaftenseite**
- Allgemeines 159
- Außenbereich 160
- Gruppierung 162
- Knoten 165
- Objekte 174
- Verbindungen 176
- Zeitrechnung 178
- Zusätzliche Ansichten 170
- Enabled**
- Eigenschaft von*
- VcNet* 537
- EnableSupplyTextEntryEvent**
- Eigenschaft von*
- VcNet* 537
- EndDateForAutomaticScheduling**
- Eigenschaft von*
- VcScheduler* 739
- EndDateNotLaterThanDataFieldIndex**
- Eigenschaft von*
- VcScheduler* 740
- EndDateTime**
- Eigenschaft von*
- VcInterval* 444
- EndLoading**
- Methode von*

- VcNet 580
- EndMonth**
 - Eigenschaft von
 - VcInterval 444
- EndTime**
 - Eigenschaft von
 - VcInterval 444
- EndWeekday**
 - Eigenschaft von
 - VcInterval 445
- Ereignis**
 - Rückgabewert 538
 - Tool Tip Text 538
- Ereignisse 100**
 - Error*
 - VcNet 602
 - ErrorAsVariant*
 - VcNet 603
 - KeyDown*
 - VcNet 603
 - KeyPress*
 - VcNet 604
 - KeyUp*
 - VcNet 604
 - OLECompleteDrag*
 - VcNet 605
 - OLEDragDrop*
 - VcNet 605
 - OLEDragOver*
 - VcNet 606
 - OLEGiveFeedback*
 - VcNet 607
 - OLESetData*
 - VcNet 608
 - OLEStartDrag*
 - VcNet 608
 - OnBoxLClick*
 - VcNet 609
 - OnBoxLDbfClick*
 - VcNet 610
 - OnBoxModifyComplete*
 - VcNet 610
 - OnBoxModifyCompleteEx*
 - VcNet 611
 - OnBoxRClick*
 - VcNet 611
 - OnDataRecordCreate*
 - VcNet 612
 - OnDataRecordCreateComplete*
 - VcNet 613
 - OnDataRecordDelete*
 - VcNet 614
 - OnDataRecordDeleteComplete*
 - VcNet 614
 - OnDataRecordModify*
 - VcNet 615
 - OnDataRecordModifyComplete*
 - VcNet 616
 - OnDataRecordNotFound*
 - VcNet 616
 - OnDiagramLClick*
 - VcNet 616
 - OnDiagramLDbfClick*
 - VcNet 617
 - OnDiagramRClick*
 - VcNet 617
 - OnGiveFeedbackForNodeCreating*
 - VcNet 618
 - OnGroupCreate*
 - VcNet 619
 - OnGroupDelete*
 - VcNet 619
 - OnGroupLClick*
 - VcNet 619

<i>OnGroupLDbIClick</i>	VcNet 631
VcNet 620	<i>OnMouseDown</i>
<i>OnGroupModify</i>	VcNet 631
VcNet 620	<i>OnMouseMove</i>
<i>OnGroupModifyComplete</i>	VcNet 632
VcNet 621	<i>OnMouseUp</i>
<i>OnGroupRClick</i>	VcNet 632
VcNet 622	<i>OnNodeCreate</i>
<i>OnHelpRequested</i>	VcNet 633
VcNet 622	<i>OnNodeCreateCompleteEx</i>
<i>OnLegendViewClosed</i>	VcNet 634
VcNet 623	<i>OnNodeDelete</i>
<i>OnLinkCreate</i>	VcNet 635
VcNet 623	<i>OnNodeDeleteCompleteEx</i>
<i>OnLinkCreateComplete</i>	VcNet 635
VcNet 624	<i>OnNodeLClick</i>
<i>OnLinkDelete</i>	VcNet 636
VcNet 625	<i>OnNodeLDbIClick</i>
<i>OnLinkDeleteComplete</i>	VcNet 636
VcNet 625	<i>OnNodeModifyComplete</i>
<i>OnLinkLClickCltn</i>	VcNet 637
VcNet 626	<i>OnNodeModifyCompleteEx</i>
<i>OnLinkLDbIClickCltn</i>	VcNet 637
VcNet 626	<i>OnNodeModifyEx</i>
<i>OnLinkModifyComplete</i>	VcNet 638
VcNet 627	<i>OnNodeRClick</i>
<i>OnLinkModifyEx</i>	VcNet 639
VcNet 627	<i>OnNodesMarkComplete</i>
<i>OnLinkRClickCltn</i>	VcNet 640
VcNet 628	<i>OnNodesMarkEx</i>
<i>OnLinksMark</i>	VcNet 640
VcNet 629	<i>OnSelectField</i>
<i>OnLinksMarkComplete</i>	VcNet 641
VcNet 630	<i>OnShowInPlaceEditor</i>
<i>OnModifyComplete</i>	VcNet 641
VcNet 630	<i>OnStatusLineText</i>
<i>OnMouseDbIClick</i>	VcNet 643

OnSupplyTextEntry
 VcNet 643
OnSupplyTextEntryAsVariant
 VcNet 652
OnToolTipText
 VcNet 652
OnToolTipTextAsVariant
 VcNet 653
OnWorldViewClosed
 VcNet 654
OnZoomFactorModifyComplete
 VcNet 654
Error
 Ereignis von
 VcNet 602
ErrorAsVariant
 Ereignis von
 VcNet 603
Erzeugemodus 272
Esker ActiveX Plug-In 21
Evaluate
 Methode von
 VcFilter 419
EventReturnStatus
 Eigenschaft von
 VcNet 537
EventText
 Eigenschaft von
 VcNet 538
Export 274
ExportGraphicsToFile
 Methode von
 VcNet 580
ExtendedDataTables
 Eigenschaft von
 VcNet 538

F

Faltmarkierungen 267
Fehlerbehebung 602
Fehlermeldungen 288
Fensterkoordinaten
 aus Bandnummern konvertieren 595
 in Bandnummern konvertieren 591,
 592
FieldByIndex
 Methode von
 VcDataDefinitionTable 375
FieldByName
 Methode von
 VcDataDefinitionTable 375
FieldsSeparatedByLines
 Eigenschaft von
 VcBoxFormat 329
 VcNodeFormat 694
FieldText
 Eigenschaft von
 VcBox 311
FilePath
 Eigenschaft von
 VcNet 539
Files
 Eigenschaft von
 DataObject 293
Filter 101
 Anzahl 422
 bearbeiten 185
 für Knoten 52
 Knoten selektieren 527
 markierte Knoten 422
 Name 416
 siehe auch
 VcFilter 414

- über Index 424
- Vergleichswert 186
- verwalten 183
- Zugriff über Filternamen 424
- FilterByIndex**
 - Methode von
 - VcFilterCollection 424
- FilterByName**
 - Methode von
 - VcFilterCollection 424
- FilterCollection**
 - Eigenschaft von
 - VcNet 540
 - siehe auch
 - VcFilterCollection 421
- FilterName**
 - Eigenschaft von
 - VcFilterSubCondition 429
 - VcLinkAppearance 467
 - VcNodeAppearance 664
- FilterSubCondition**
 - siehe auch
 - VcFilterSubCondition 427
- FirstBox**
 - Methode von
 - VcBoxCollection 326
- FirstCalendar**
 - Methode von
 - VcCalendarCollection 361
- FirstCalendarProfile**
 - Methode von
 - VcCalendarProfileCollection 369
- FirstDataRecord**
 - Methode von
 - VcDataRecordCollection 386
- FirstDataTable**
 - Methode von
 - VcDataTableCollection 397
- FirstDataTableField**
 - Methode von
 - VcDataTableFieldCollection 408
- FirstField**
 - Methode von
 - VcDataDefinitionTable 376
- FirstFilter**
 - Methode von
 - VcFilterCollection 425
- FirstFormat**
 - Methode von
 - VcBoxFormatCollection 336
 - VcLinkFormatCollection 493
 - VcNodeFormatCollection 701
- FirstGroup**
 - Methode von
 - VcGroupCollection 439
- FirstInterval**
 - Methode von
 - VcIntervalCollection 452
- FirstLink**
 - Methode von
 - VcLinkCollection 483
- FirstLinkAppearance**
 - Methode von
 - VcLinkAppearanceCollection 479
- FirstMap**
 - Methode von
 - VcMapCollection 509
- FirstMapEntry**
 - Methode von
 - VcMap 504
- FirstNode**
 - Methode von
 - VcNodeCollection 690
- FirstNodeAppearance**

- Methode von
 - VcNodeAppearanceCollection* 686
- FitToPage**
 - Eigenschaft von
 - VcPrinter* 724
- Flußrichtung 45**
- Flussrichtung 567**
- FoldingMarksType**
 - Eigenschaft von
 - VcPrinter* 724
- FontAntiAliasingEnabled**
 - Eigenschaft von
 - VcNet* 540
- FontBody**
 - Eigenschaft von
 - VcMapEntry* 514
- FontName**
 - Eigenschaft von
 - VcMapEntry* 515
- FontSize**
 - Eigenschaft von
 - VcMapEntry* 515
- FormatByIndex**
 - Methode von
 - VcBoxFormatCollection* 336
 - VcLinkFormatCollection* 494
 - VcNodeFormatCollection* 702
- FormatByName**
 - Methode von
 - VcBoxFormatCollection* 336
 - VcLinkFormatCollection* 494
 - VcNodeFormatCollection* 702
- Formatfeld**
 - Anzahl der Felder 330, 695
- FormatField**
 - Eigenschaft von
 - VcBoxFormat* 329
 - VcLinkFormat* 487
 - VcNodeFormat* 694
- FormatFieldCount**
 - Eigenschaft von
 - VcBoxFormat* 330
 - VcLinkFormat* 487
 - VcNodeFormat* 695
- FormatName**
 - Eigenschaft von
 - VcBox* 311
 - VcBoxFormatField* 340
 - VcLinkAppearance* 468
 - VcLinkFormatField* 497
 - VcNodeAppearance* 665
 - VcNodeFormatField* 706
- Formular**
 - anpassen 33
- FrameAroundFieldsVisible**
 - Eigenschaft von
 - VcNodeAppearance* 665
- FrameShape**
 - Eigenschaft von
 - VcNodeAppearance* 666
- FreeFloatDataFieldIndex**
 - Eigenschaft von
 - VcScheduler* 740

G

Gesamtnetz 274, 276

GetActualExtent

Methode von

- VcBox* 318

GetAValueFromARGB

Methode von

- VcNet* 582

GetBValueFromARGB

Methode von

- VcNet 583
- GetData**
 - Methode von
 - DataObject 294
- GetEndOfPreviousWorktime**
 - Methode von
 - VcCalendar 353
- GetFormat**
 - Methode von
 - DataObject 296
- GetGValueFromARGB**
 - Methode von
 - VcNet 584
- GetLinkByID**
 - Methode von
 - VcNet 584
- GetLinkByIDs**
 - Methode von
 - VcNet 585
- GetMapEntry**
 - Methode von
 - VcMap 505
- GetNewUniqueID**
 - Methode von
 - VcDataRecordCollection 387
- GetNextIntervalBorder**
 - Methode von
 - VcCalendar 353
- GetNodeByID**
 - Methode von
 - VcNet 585
- GetPreviousIntervalBorder**
 - Methode von
 - VcCalendar 354
- GetRValueFromARGB**
 - Methode von
 - VcNet 586
- GetStartOfInterval**
 - Methode von
 - VcCalendar 354
- GetStartOfNextWorktime**
 - Methode von
 - VcCalendar 355
- GetTopLeftPixel**
 - Methode von
 - VcBox 319
- GetXOffset**
 - Methode von
 - VcBox 319
- GetXOffsetAsVariant**
 - Methode von
 - VcBox 319
- Grafik**
 - exportieren 83
- Grafiken**
 - festlegen 245
- Grafikformat 103**
- GraphicsFileName**
 - Eigenschaft von
 - VcBorderBox 304
 - VcMapEntry 516
 - VcNodeFormatField 706
- GraphicsFileNameDataFieldIndex**
 - Eigenschaft von
 - VcNodeFormatField 707
- GraphicsFileNameMapName**
 - Eigenschaft von
 - VcNodeFormatField 707
- GraphicsHeight**
 - Eigenschaft von
 - VcBoxFormatField 340
 - VcNodeFormatField 707
- Group**
 - siehe auch

- VcGroup 431
- GroupByName**
 - Methode von
 - VcGroupCollection 440
- GroupCollection**
 - Eigenschaft von
 - VcNet 541
 - siehe auch
 - VcGroupCollection 438
- GroupDescriptionName**
 - Eigenschaft von
 - VcNet 541
- GroupField**
 - Eigenschaft von
 - VcNet 542
- GroupHorizontalMargin**
 - Eigenschaft von
 - VcNet 542
- Grouping**
 - Eigenschaft von
 - VcNet 542
- GroupingTitlesFullyVisible**
 - Eigenschaft von
 - VcNet 543
- GroupInteractionsAllowed**
 - Eigenschaft von
 - VcNet 543
- GroupMode**
 - Eigenschaft von
 - VcNet 544
- GroupMovingAllowed**
 - Eigenschaft von
 - VcNet 544
- GroupSortField**
 - Eigenschaft von
 - VcNet 545
- GroupSortMode**
 - Eigenschaft von
 - VcNet 545
- GroupTitleField**
 - Eigenschaft von
 - VcNet 546
- GroupVerticalMargin**
 - Eigenschaft von
 - VcNet 546
- Gruppe**
 - Anzahl 439
 - Hintergrundfarbe 431
 - kollabiert 432
 - Name 435
 - Titel 436
 - Titelzeilen 436
- Gruppen**
 - erzeugen 619
 - Hintergrundfarbe 163
 - interaktiv kollabieren/expandieren 543
 - Linienfarbe 432
 - Ränder 163
 - Randlinie 163
 - Titel 164
- Gruppencode 162**
- Gruppierung 108, 109, 162, 544**
 - an- bzw. abschalten 542
 - Datenfeld für Gruppensortierung 545
 - Datenfeld für Gruppierkriterium 542
 - Gruppe löschen 619
 - Gruppe verändern 621
 - Gruppentitel 546
 - horizontale Ränder 542
 - Linientyp 434
 - Sortierreihenfolge der Gruppen 545
 - vertikale Ränder 546

H

Height

- Eigenschaft von*
 - VcLegendView 455
 - VcRect 734
 - VcWorldView 746

HeightActualValue

- Eigenschaft von*
 - VcLegendView 455
 - VcWorldView 746

Hidden

- Eigenschaft von*
 - VcDataTableField 401
 - VcDefinitionField 411

Hierarchie

- ändern 637

Hilfe-Ereignis 622**Hilfsknoten 120**

- übersichtlich anordnen 69

Hintergrundfarbe

- des Diagramms 535

HTML 11**HTML-Seite 21****hWnd 547**

- Eigenschaft von*
 - VcNet 547

I

i 494**ID 111**

- Eigenschaft von*
 - VcDataRecord 380
 - VcDefinitionField 412
 - VcLink 463
 - VcNode 656

Identifikation 111**IdentifyFormatField**

- Methode von*
 - VcBox 320
 - VcNet 586

IdentifyFormatFieldAsVariant

- Methode von*
 - VcNet 587

IdentifyObject

- Methode von*
 - VcDataRecord 381

IdentifyObjectAt

- Methode von*
 - VcNet 587

IdentifyObjectAtAsVariant

- Methode von*
 - VcNet 588

IncomingLinks

- Eigenschaft von*
 - VcNode 657

Index

- Eigenschaft von*
 - VcBoxFormatField 341
 - VcDataTableField 402
 - VcFilterSubCondition 429
 - VcLinkFormatField 497
 - VcNodeFormatField 708

InFlowGroupDescriptionName

- Eigenschaft von*
 - VcNet 547

InFlowGroupField

- Eigenschaft von*
 - VcNet 548

InFlowGroupingEnabled

- Eigenschaft von*
 - VcNet 548

InFlowGroupSeparationLineColor

- Eigenschaft von*

- VcNet 548
- InFlowGroupSeparationLineType**
 - Eigenschaft von
 - VcNet 549
- InFlowGroupTimeInterval**
 - Eigenschaft von
 - VcNet 550
- InFlowGroupTitleField**
 - Eigenschaft von
 - VcNet 550
- InFlowGroupTitlesBackColor**
 - Eigenschaft von
 - VcNet 551
- InFlowGroupTitlesFont**
 - Eigenschaft von
 - VcNet 551
- InFlowGroupTitlesVisibleAtBottomOrRight**
 - Eigenschaft von
 - VcNet 551
- InFlowGroupTitlesVisibleAtTopOrLeft**
 - Eigenschaft von
 - VcNet 551
- InFlowGroupTitleTimeFormat**
 - Eigenschaft von
 - VcNet 552
- InFlowGroupVerticalCaptionWidth**
 - Eigenschaft von
 - VcNet 552
- In-Flow-Gruppierung 113, 167, 228**
- ini-Datei 280**
- InPlaceEditingAllowed**
 - Eigenschaft von
 - VcNet 552
- InsertLinkRecord**
 - Methode von
 - VcNet 589
- InsertNodeRecord**
 - Methode von
 - VcNet 589
- Installation 13**
- InteractionMode**
 - Eigenschaft von
 - VcNet 553
- Interaktion**
 - Markieren mehrerer Boxen 527
- Interaktionsmodi 553**
- InterfaceNodesShown**
 - Eigenschaft von
 - VcNet 553
- Internet 11, 83, 252, 274**
- Interval**
 - siehe auch
 - VcInterval 441
- IntervalByIndex**
 - Methode von
 - VcIntervalCollection 452
- IntervalByName**
 - Methode von
 - VcIntervalCollection 452
- IntervalCollection**
 - Eigenschaft von
 - VcCalendar 350
 - VcCalendarProfile 363
 - siehe auch
 - VcIntervalCollection 449
- Intervall**
 - Anzahl 450
 - Enddatum und -zeit 444
 - Endmonat 444
 - Endzeit 444
 - erster Wochentag 447
 - erstes Intervall 452
 - hinzufügen 450

Kalenderprofil 443
 kopieren 451
 letzter Wochentag 445
 löschen 453
 nächstes Intervall 453
 Name 445
 Reihenfolge 447
 Startdatum und -zeit 446
 Startmonat 446
 Startzeit 446
 Tag des ersten Monats 443
 Tag des letzten Monats 443
 Typ 447
 über Index 452
 Zugriff über Intervallnamen 452

Intervall-Collection

aktualisieren 453

Intervalle

bearbeiten 235, 239, 241, 242, 244

IsValid

Methode von

VcFilter 419

VcFilterSubCondition 430

IsWorktime

Methode von

VcCalendar 355

Item

Eigenschaft von

DataObjectFiles 300

K

Kalender

aktiver 358
 aktualisieren 355
 Anzahl 359
 Anzahl der Sekunden eines
 Arbeitstages 351

Dauer 351
 festlegen 233
 Name 350, 364
 über Index 360
 zu Knoten zuweisen 528

Kalenderprofil

Anzahl 367
 Reihenfolge 365
 Typ 364
 über Index 368
 Zugriff über Kalenderprofilnamen 369

Kalenderprofil verwalten

Dialogfeld: 237

KeyDown

Ereignis von

VcNet 603

KeyPress

Ereignis von

VcNet 604

KeyUp

Ereignis von

VcNet 604

Knoten 116

3D-Effekt 201
 aktualisieren 660
 alle Daten 655
 auf dem Rang ihrer Vorgänger
 anordnen 166
 aus Zwischenspeicher einfügen 591
 bearbeiten 259, 579
 Daten aktualisieren 601
 Daten bearbeiten 259
 Datenfeld 656
 Datensatz 658
 Doppelrahmen 200, 201
 durch Filter selektieren 527
 erzeugen 633, 634

- erzeugen und bearbeiten 48
- Format identifizieren 586, 587
- Gestaffelt 203
- gruppieren 74
- herausführende Verbindungen 658
- Hilfsknoten 120
- hineinführende Verbindungen 657
- Hintergrundfarbe 202
- ID 656
- in Zwischenspeicher kopieren 576
- in Zwischenspeicher verschieben 576
- interaktiv anlegen erlaubt 739
- interaktiv erzeugen 282, 284
- Knotenaussehen 54, 122
- Knotenaussehen bearbeiten 200
- Knotenaussehen verwalten 196
- Knotenform 200
- Knotenformat 58, 124
- Knotenformat bearbeiten 214
- laden 589
- löschen 577, 635, 659
- markieren 51, 637, 640, 657
- Markierungstyp 169
- Muster 201
- Musterfarbe 201
- neue bearbeiten 536
- neue zulassen 528, 544
- nicht durchtrennen 266
- Positionen 117
- Positionen mit Datenfeldern synchronisieren 166
- Rang 118
- Schatten 203
- verändern 638
- zugeordneter Datensatz 659

Knoten erzeugen

erlauben/verbieten 618

Knoten und Verbindungen

- erzeugen 262
- markieren und verschieben 264

Knotenaussehen

- 3D-Effekt 681
- Anzahl von Knotenstapeln 677
- doppelte Umrahmung 664
- Durchstreichmuster 679
- Farbe des Durchstreichmusters 680
- Filter 664
- Format 665
- Hintergrundfarbe 663
- Legendentext 667
- Linienfarbe 668
- Linienfarbenzuordnungstabelle 668
- Linienstärke 669
- Linientyp 670
- Name 671
- Rahmen um Felder 665
- Rahmenform 666
- Reihenfolge 682
- Schatten 678
- Schattenfarbe 678
- sichtbar in Legende 681

Knotenaussehen-Auflistung

- Anzahl 684
- Enumerator 683
- erstes Knotenaussehen 686
- hinzufügen 684
- hinzufügen über Spezifikation 685
- Kopieren 685
- löschen 688
- nächstes Knotenaussehen 686
- Zugriff über Index 687
- Zugriff über Name 687

Knotenformat

- Spezifikation 695
 - Knotenformat-Auflistung**
 - Anzahl 699
 - Enumerator 699
 - erstes Format 701
 - hinzufügen 700
 - hinzufügen über Spezifikation 700
 - kopieren 701
 - Löschen 703
 - nächstes Format 702
 - Zugriff über Index 702
 - Zugriff über Name 702
 - Knotenformate**
 - verwalten 209
 - Knotenformatfeld**
 - Füllmuster 712
 - maximale Zeilenzahl 708
 - minimale Zeilenzahl 709
 - Musterfarbe 711
 - Name 695
 - Knotenpositionen 66**
 - Komplettansicht 127, 274, 573**
 - schließen 623, 654
 - Konfiguration 84, 280, 531**
 - speichern 578
 - Kontextmenü**
 - abschalten 285
 - für das Diagramm 272
 - für Knoten 276
 - für Verbindungen 277
 - Kundendienst 27**
- L
- Ladevorgang**
 - Ende 580
 - LateEndDateDataFieldIndex**
 - Eigenschaft von
 - VcScheduler 740
 - LateStartDateDataFieldIndex**
 - Eigenschaft von
 - VcScheduler 740
 - Layoutberechnung 575**
 - Leerseiten unterdrücken 266**
 - Left**
 - Eigenschaft von
 - VcLegendView 456
 - VcRect 735
 - VcWorldView 747
 - LeftActualValue**
 - Eigenschaft von
 - VcLegendView 456
 - VcWorldView 747
 - LeftMargin**
 - Eigenschaft von
 - VcNodeFormatField 708
 - Legende**
 - Anordnung 249
 - Anordnung 250
 - Attribute 249
 - festlegen 245
 - Text 667
 - Titel 249
 - LegendElementsArrangement**
 - Eigenschaft von
 - VcBoundingBox 305
 - LegendElementsBottomMargin**
 - Eigenschaft von
 - VcBoundingBox 305
 - LegendElementsMaximumColumnCount**
 - Eigenschaft von
 - VcBoundingBox 305
 - LegendElementsMaximumRowCount**
 - Eigenschaft von

VcBoundingBox 306

LegendElementsTopMargin

Eigenschaft von

VcBoundingBox 306

Legendenansicht 129, 274, 554

LegendFont

Eigenschaft von

VcBoundingBox 306

LegendText

Eigenschaft von

VcNodeAppearance 667

LegendTitle

Eigenschaft von

VcBoundingBox 306

LegendTitleFont

Eigenschaft von

VcBoundingBox 307

LegendTitleVisible

Eigenschaft von

VcBoundingBox 307

Legendview 129, 554

LegendView

Eigenschaft von

VcNet 554

siehe auch

VcLegendView 454

LineColor

Eigenschaft von

VcBox 312

VcGroup 432

VcLinkAppearance 468

VcNodeAppearance 668

LineColorDataFieldIndex

Eigenschaft von

VcNodeAppearance 668

LineColorMapName

Eigenschaft von

VcNodeAppearance 668

LineThickness

Eigenschaft von

VcBox 312

VcGroup 433

VcLinkAppearance 469

VcNodeAppearance 669

LineType

Eigenschaft von

VcBox 313

VcGroup 434

VcLinkAppearance 470

VcNodeAppearance 670

Linie bearbeiten 231

Link

siehe auch

VcLink 462

LinkAnnotationColumnNumberDataFieldIndex

Eigenschaft von

VcNet 554

LinkAnnotationRowNumberDataFieldIndex

Eigenschaft von

VcNet 554

LinkAppearance

siehe auch

VcLinkAppearance 467

LinkAppearanceByIndex

Methode von

VcLinkAppearanceCollection 479

LinkAppearanceByName

Methode von

VcLinkAppearanceCollection 479

LinkAppearanceCollection

Eigenschaft von

VcNet 555

siehe auch
 VcLinkAppearanceCollection 476

LinkAppearance-Objekt
 Anzahl in Collection 477
 Enumerator-Objekt 477
 Iteration, Erstwert 479
 Iteration, Folgewert 480
 über Index 479
 über Name 479

LinkCollection
Eigenschaft von
 VcNet 555
 siehe auch
 VcLinkCollection 482

LinkDurationDataFieldIndex
Eigenschaft von
 VcScheduler 741

LinkFormat
 siehe auch
 VcLinkFormat 486

LinkFormatCollection
Eigenschaft von
 VcNet 555
 siehe auch
 VcLinkFormatCollection 490

Linkformatfeld
 Index 497

LinkFormatField
 siehe auch
 VcLinkFormatField 496

LinkPredecessorDataFieldIndex
Eigenschaft von
 VcNet 556

LinksDataTableName
Eigenschaft von
 VcNet 556

LinkSuccessorDataFieldIndex

Eigenschaft von
 VcNet 557

LinkTypeDataFieldIndex
Eigenschaft von
 VcNet 558

Lizenzierung 251
 Probleme 279

Lizenzinformationen
 anfordern 253

M

MakeARGB
Methode von
 VcNet 590

Map
 Anzahl der Einträge 501
 Anzahl der Maps 507
 durch Zuordnungstabelle bestimmte
 Vorgänge aktualisieren 512
 Eintrag erzeugen 503
 Eintrag löschen 504
 Name 502
 siehe auch
 VcMap 500

MapByIndex
Methode von
 VcMapCollection 509

MapByName
Methode von
 VcMapCollection 510

MapCollection
Eigenschaft von
 VcNet 558
 siehe auch
 VcMapCollection 506

Map-Eintrag
 Datenfeld-Inhalt 514

- Farbwert 514
- Grafikdatei 516
- Muster 517
- Schriftart 515
- Schriftgrad 514
- Schriftgröße 515
- MapEntry**
 - siehe auch
 - VcMapEntry 513
- Maps 558**
- MarginsShownInInches**
 - Eigenschaft von
 - VcPrinter 726
- MarkBox**
 - Eigenschaft von
 - VcBox 314
- MarkedNodesFilter**
 - Eigenschaft von
 - VcFilterCollection 422
- Markieren/Demarkieren**
 - Ende der Operation 630, 640
- Markiermodus 272**
- Markierungstyp**
 - Knoten 51
 - Verbindungen 51
- MarkingColor**
 - Eigenschaft von
 - VcWorldView 748
- MarkLink**
 - Eigenschaft von
 - VcLink 464
- MarkNode**
 - Eigenschaft von
 - VcNode 657
- MaxHorizontalPagesCount**
 - Eigenschaft von
 - VcPrinter 727
- MaximumTextLineCount**
 - Eigenschaft von
 - VcBoxFormatField 341
 - VcNodeFormatField 708
- MaxVerticalPagesCount**
 - Eigenschaft von
 - VcPrinter 727
- Methoden**
 - AboutBox
 - VcNet 574
 - Add
 - DataObjectFiles 301
 - VcBoxCollection 323
 - VcBoxFormatCollection 334
 - VcCalendarCollection 359
 - VcCalendarProfileCollection 367
 - VcDataRecordCollection 385
 - VcDataTableCollection 394
 - VcDataTableFieldCollection 406
 - VcFilterCollection 423
 - VcIntervalCollection 450
 - VcLinkAppearanceCollection 477
 - VcLinkFormatCollection 491
 - VcMapCollection 508
 - VcNodeAppearanceCollection 684
 - VcNodeFormatCollection 699
 - AddBySpecification
 - VcBoxCollection 324
 - VcBoxFormatCollection 335
 - VcCalendarCollection 359
 - VcCalendarProfileCollection 368
 - VcFilterCollection 423
 - VcIntervalCollection 451
 - VcLinkAppearanceCollection 478
 - VcLinkFormatCollection 492
 - VcMapCollection 508
 - VcNodeAppearanceCollection 685

- VcNodeFormatCollection* 700
- AddDuration*
 - VcCalendar* 351
- AddSubCondition*
 - VcFilter* 418
- Arrange*
 - VcNet* 575
- BorderBox*
 - VcBorderArea* 302
- BoxByIndex*
 - VcBoxCollection* 324
- BoxByName*
 - VcBoxCollection* 325
- CalcDuration*
 - VcCalendar* 352
- CalendarByIndex*
 - VcCalendarCollection* 360
- CalendarByName*
 - VcCalendarCollection* 360
- CalendarProfileByIndex*
 - VcCalendarProfileCollection* 368
- CalendarProfileByName*
 - VcCalendarProfileCollection* 369
- Clear*
 - DataObject* 294
 - DataObjectFiles* 301
 - VcCalendar* 352
 - VcNet* 575
- Copy*
 - VcBoxCollection* 325
 - VcBoxFormatCollection* 335
 - VcCalendarCollection* 360
 - VcCalendarProfileCollection* 369
 - VcDataTableCollection* 395
 - VcDataTableFieldCollection* 407
 - VcFilterCollection* 424
 - VcIntervalCollection* 451
 - VcLinkAppearanceCollection* 478
 - VcLinkFormatCollection* 493
 - VcMapCollection* 509
 - VcNodeAppearanceCollection* 685
 - VcNodeFormatCollection* 701
- CopyFormatField*
 - VcBoxFormat* 331
 - VcLinkFormat* 488
 - VcNodeFormat* 696
- CopyNodesIntoClipboard*
 - VcNet* 575
- CopySubCondition*
 - VcFilter* 418
- CreateDataField*
 - VcDataDefinitionTable* 374
- CreateEntry*
 - VcMap* 503
- CutNodesIntoClipboard*
 - VcNet* 576
- DataRecord*
 - VcLink* 465
 - VcNode* 658
- DataRecordByID*
 - VcDataRecordCollection* 386
- DataTableByIndex*
 - VcDataTableCollection* 396
- DataTableByName*
 - VcDataTableCollection* 396
- DataTableFieldByIndex*
 - VcDataTableFieldCollection* 407
- DataTableFieldByName*
 - VcDataTableFieldCollection* 408
- DeleteDataRecord*
 - VcDataRecord* 380
- DeleteEntry*
 - VcMap* 503
- DeleteLink*

- VcLink* 465
- DeleteLinkRecord*
 - VcNet* 576
- DeleteNode*
 - VcNode* 659
- DeleteNodeRecord*
 - VcNet* 577
- DetectDataTableFieldName*
 - VcNet* 577
- DetectDataTableName*
 - VcNet* 577
- DetectFieldIndex*
 - VcNet* 578
- DumpConfiguration*
 - VcNet* 578
- EditLink*
 - VcNet* 579
- EditNode*
 - VcNet* 579
- EndLoading*
 - VcNet* 580
- Evaluate*
 - VcFilter* 419
- ExportGraphicsToFile*
 - VcNet* 580
- FieldByIndex*
 - VcDataDefinitionTable* 375
- FieldByName*
 - VcDataDefinitionTable* 375
- FilterByIndex*
 - VcFilterCollection* 424
- FilterByName*
 - VcFilterCollection* 424
- FirstBox*
 - VcBoxCollection* 326
- FirstCalendar*
 - VcCalendarCollection* 361
- FirstCalendarProfile*
 - VcCalendarProfileCollection* 369
- FirstDataRecord*
 - VcDataRecordCollection* 386
- FirstDataTable*
 - VcDataTableCollection* 397
- FirstDataTableField*
 - VcDataTableFieldCollection* 408
- FirstField*
 - VcDataDefinitionTable* 376
- FirstFilter*
 - VcFilterCollection* 425
- FirstFormat*
 - VcBoxFormatCollection* 336
 - VcLinkFormatCollection* 493
 - VcNodeFormatCollection* 701
- FirstGroup*
 - VcGroupCollection* 439
- FirstInterval*
 - VcIntervalCollection* 452
- FirstLink*
 - VcLinkCollection* 483
- FirstLinkAppearance*
 - VcLinkAppearanceCollection* 479
- FirstMap*
 - VcMapCollection* 509
- FirstMapEntry*
 - VcMap* 504
- FirstNode*
 - VcNodeCollection* 690
- FirstNodeAppearance*
 - VcNodeAppearanceCollection* 686
- FormatByIndex*
 - VcBoxFormatCollection* 336
 - VcLinkFormatCollection* 494
 - VcNodeFormatCollection* 702
- FormatByName*

- VcBoxFormatCollection* 336
- VcLinkFormatCollection* 494
- VcNodeFormatCollection* 702
- GetActualExtent*
 - VcBox* 318
- GetAValueFromARGB*
 - VcNet* 582
- GetBValueFromARGB*
 - VcNet* 583
- GetData*
 - DataObject* 294
- GetEndOfPreviousWorktime*
 - VcCalendar* 353
- GetFormat*
 - DataObject* 296
- GetGValueFromARGB*
 - VcNet* 584
- GetLinkByID*
 - VcNet* 584
- GetLinkByIDs*
 - VcNet* 585
- GetMapEntry*
 - VcMap* 505
- GetNewUniqueID*
 - VcDataRecordCollection* 387
- GetNextIntervalBorder*
 - VcCalendar* 353
- GetNodeByID*
 - VcNet* 585
- GetPreviousIntervalBorder*
 - VcCalendar* 354
- GetRValueFromARGB*
 - VcNet* 586
- GetStartOfInterval*
 - VcCalendar* 354
- GetStartOfNextWorktime*
 - VcCalendar* 355
- GetTopLeftPixel*
 - VcBox* 319
- GetXYOffset*
 - VcBox* 319
- GetXYOffsetAsVariant*
 - VcBox* 319
- GroupName*
 - VcGroupCollection* 440
- IdentifyFormatField*
 - VcBox* 320
 - VcNet* 586
- IdentifyFormatFieldAsVariant*
 - VcNet* 587
- IdentifyObject*
 - VcDataRecord* 381
- IdentifyObjectAt*
 - VcNet* 587
- IdentifyObjectAtAsVariant*
 - VcNet* 588
- InsertLinkRecord*
 - VcNet* 589
- InsertNodeRecord*
 - VcNet* 589
- IntervalByIndex*
 - VcIntervalCollection* 452
- IntervalByName*
 - VcIntervalCollection* 452
- IsValid*
 - VcFilter* 419
 - VcFilterSubCondition* 430
- IsWorktime*
 - VcCalendar* 355
- LinkAppearanceByIndex*
 - VcLinkAppearanceCollection* 479
- LinkAppearanceByName*
 - VcLinkAppearanceCollection* 479
- MakeARGB*

- VcNet 590
- MapByIndex
 - VcMapCollection 509
- MapByName
 - VcMapCollection 510
- NextBox
 - VcBoxCollection 326
- NextCalendar
 - VcCalendarCollection 361
- NextCalendarProfile
 - VcCalendarProfileCollection 370
- NextDataRecord
 - VcDataRecordCollection 387
- NextDataTable
 - VcDataTableCollection 397
- NextDataTableField
 - VcDataTableFieldCollection 409
- NextField
 - VcDataDefinitionTable 376
- NextFilter
 - VcFilterCollection 425
- NextFormat
 - VcBoxFormatCollection 337
 - VcLinkFormatCollection 494
 - VcNodeFormatCollection 702
- NextGroup
 - VcGroupCollection 440
- NextInterval
 - VcIntervalCollection 453
- NextLink
 - VcLinkCollection 484
- NextLinkAppearance
 - VcLinkAppearanceCollection 480
- NextMap
 - VcMapCollection 510
- NextMapEntry
 - VcMap 505
- NextNode
 - VcNodeCollection 691
- NextNodeAppearance
 - VcNodeAppearanceCollection 686
- NodeAppearanceByIndex
 - VcNodeAppearanceCollection 687
- NodeAppearanceByName
 - VcNodeAppearanceCollection 687
- Open
 - VcNet 590
- PageLayout
 - VcNet 591
- PasteNodesFromClipboard
 - VcNet 591
- PixelsToRaster
 - VcNet 591
- PixelsToRasterAsVariant
 - VcNet 592
- PrintDirectEx
 - VcNet 592
- PrinterSetup
 - VcNet 593
- PrintIt
 - VcNet 594
- PrintPreview
 - VcNet 594
- PrintToFile
 - VcNet 594
- PutInOrderAfter
 - VcCalendarProfile 365
 - VcInterval 447
 - VcLinkAppearance 475
 - VcNodeAppearance 682
- RasterToPixels
 - VcNet 595
- RasterToPixelsAsVariant
 - VcNet 595

RelatedDataRecord

- VcDataRecord* 381
- VcLink* 466
- VcNode* 659

Remove

- DataObjectFiles* 301
- VcBoxCollection* 327
- VcBoxFormatCollection* 337
- VcCalendarCollection* 362
- VcCalendarProfileCollection* 370
- VcDataRecordCollection* 388
- VcFilterCollection* 426
- VcIntervalCollection* 453
- VcLinkAppearanceCollection* 480
- VcLinkFormatCollection* 495
- VcMapCollection* 511
- VcNodeAppearanceCollection* 688
- VcNodeFormatCollection* 703

RemoveFormatField

- VcBoxFormat* 331
- VcLinkFormat* 489
- VcNodeFormat* 697

RemoveSubCondition

- VcFilter* 419

Reset

- VcNet* 596

SaveAsEx

- VcNet* 596

ScheduleProject

- VcNet* 597
- VcScheduler* 743

ScrollToNodePosition

- VcNet* 597

SelectCalendarProfiles

- VcCalendarProfileCollection* 370

SelectLinks

- VcLinkCollection* 484

SelectMaps

- VcMapCollection* 511

SelectNodes

- VcNodeCollection* 691

SetData

- DataObject* 297

SetXY

- VcGroup* 437

SetXYOffset

- VcBox* 320

SetXYOffsetByTopLeftPixel

- VcBox* 320

ShowAlwaysCompleteView

- VcNet* 598

ShowExportGraphicsDialog

- VcNet* 598

SuspendUpdate

- VcNet* 600

Update

- VcBoxCollection* 327
- VcCalendar* 355
- VcCalendarCollection* 362
- VcCalendarProfileCollection* 371
- VcDataRecordCollection* 388
- VcDataTableCollection* 398
- VcIntervalCollection* 453
- VcLegendView* 461
- VcLinkAppearanceCollection* 481
- VcMapCollection* 512

UpdateDataRecord

- VcDataRecord* 382

UpdateLink

- VcLink* 466

UpdateLinkRecord

- VcNet* 601

UpdateNode

- VcNode* 660

- UpdateNodeRecord*
 - VcNet* 601
 - Zoom*
 - VcNet* 601
 - ZoomOnMarkedNodes*
 - VcNet* 602
 - MinimumColumnWidth**
 - Eigenschaft von*
 - VcNet* 559
 - MinimumRowHeight**
 - Eigenschaft von*
 - VcNet* 559
 - MinimumTextLineCount**
 - Eigenschaft von*
 - VcBoxFormatField* 342
 - VcNodeFormatField* 709
 - MinimumWidth**
 - Eigenschaft von*
 - VcBoxFormatField* 342
 - VcLinkFormatField* 498
 - VcNodeFormatField* 709
 - Mode**
 - Eigenschaft von*
 - VcWorldView* 748
 - MouseProcessingEnabled**
 - Eigenschaft von*
 - VcNet* 560
 - Moveable**
 - Eigenschaft von*
 - VcBox* 314
 - MultiplePrimaryKeysAllowed**
 - Eigenschaft von*
 - VcDataTable* 392
 - Muster 232**
-
- N
- Name**
 - Eigenschaft von*
 - VcBox* 315
 - VcBoxFormat* 330
 - VcCalendar* 350
 - VcCalendarProfile* 364
 - VcDataTable* 392
 - VcDataTableField* 402
 - VcDefinitionField* 412
 - VcFilter* 416
 - VcGroup* 435
 - VcInterval* 445
 - VcLinkAppearance* 471
 - VcLinkFormat* 488
 - VcMap* 502
 - VcNodeAppearance* 671
 - VcNodeFormat* 695
 - Navigation**
 - Tastatur 256
 - Net**
 - siehe auch
 - VcNet* 521
 - Netscape 21**
 - NextBox**
 - Methode von*
 - VcBoxCollection* 326
 - NextCalendar**
 - Methode von*
 - VcCalendarCollection* 361
 - NextCalendarProfile**
 - Methode von*
 - VcCalendarProfileCollection* 370
 - NextDataRecord**
 - Methode von*
 - VcDataRecordCollection* 387
 - NextDataTable**
 - Methode von*
 - VcDataTableCollection* 397

NextDataTableField*Methode von**VcDataTableFieldCollection* 409**NextField***Methode von**VcDataDefinitionTable* 376**NextFilter***Methode von**VcFilterCollection* 425**NextFormat***Methode von**VcBoxFormatCollection* 337*VcLinkFormatCollection* 494*VcNodeFormatCollection* 702**NextGroup***Methode von**VcGroupCollection* 440**NextInterval***Methode von**VcIntervalCollection* 453**NextLink***Methode von**VcLinkCollection* 484**NextLinkAppearance***Methode von**VcLinkAppearanceCollection* 480**NextMap***Methode von**VcMapCollection* 510**NextMapEntry***Methode von**VcMap* 505**NextNode***Methode von**VcNodeCollection* 691**NextNodeAppearance***Methode von**VcNodeAppearanceCollection* 686**Node**

siehe auch

VcNode 655**NodeAppearance**

siehe auch

VcNodeAppearance 661**NodeAppearanceByIndex***Methode von**VcNodeAppearanceCollection* 687**NodeAppearanceByName***Methode von**VcNodeAppearanceCollection* 687**NodeAppearanceCollection***Eigenschaft von**VcNet* 560

siehe auch

VcNodeAppearanceCollection 683**NodeCalendarNameDataFieldIndex***Eigenschaft von**VcNet* 560**NodeChangeRankToPredecessorRankDataFieldIndex***Eigenschaft von**VcNet* 561**NodeCollection***Eigenschaft von**VcGroup* 435*VcNet* 561

siehe auch

VcNodeCollection 689**NodeColumnNumberDataFieldIndex***Eigenschaft von**VcNet* 561**NodeFormat**

siehe auch

VcNodeFormat 693

NodeFormatCollection*Eigenschaft von*

VcNet 562

siehe auch

VcNodeFormatCollection 698

NodeFormatField

siehe auch

VcNodeFormatField 704

NodeRowNumberDataFieldIndex*Eigenschaft von*

VcNet 562

NodesDataTableName*Eigenschaft von*

VcNet 563

NodeTooltipTextField*Eigenschaft von*

VcNet 563

**Objekt**

identifizieren 587, 588

Objekte

DataObject 293

DataObjectFiles 299

VcBorderArea 302

VcBorderBox 303

VcBox 310

VcBoxCollection 322

VcBoxFormat 328

VcBoxFormatCollection 333

VcBoxFormatField 339

VcCalendar 349

VcCalendarCollection 357

VcCalendarProfile 363

VcCalendarProfileCollection 366

VcDataDefinition 372

VcDataDefinitionTable 373

VcDataRecord 378

VcDataRecordCollection 383

VcDataTable 390

VcDataTableCollection 393

VcDataTableField 399

VcDataTableFieldCollection 405

VcDefinitionField 410

VcFilter 414

VcFilterCollection 421

VcFilterSubCondition 427

VcGroup 431

VcGroupCollection 438

VcInterval 441

VcIntervalCollection 449

VcLegendView 454

VcLink 462

VcLinkAppearance 467

VcLinkAppearanceCollection 476

VcLinkCollection 482

VcLinkFormat 486

VcLinkFormatCollection 490

VcLinkFormatField 496

VcMap 500

VcMapCollection 506

VcMapEntry 513

VcNet 521

VcNode 655

VcNodeAppearance 661

VcNodeAppearanceCollection 683

VcNodeCollection 689

VcNodeFormat 693

VcNodeFormatCollection 698

VcNodeFormatField 704

VcPrinter 717

VcRect 734

VcScheduler 737

VcWorldView 745

ObliqueTracksOnLinks*Eigenschaft von*

VcNet 564

OLE Drag & Drop 131

beendet 605

Cursor während eines OLE-Drag-Vorgangs in Zielkomponente gesetzt 565

Daten auf Ziel abgelegt 606

Daten über Ziel gezogen 606

Drag-Vorgang ausgeführt 609

Ereignis des Drop-Ziels 608

Knoten aus anderer VARCHART-ActiveX-Komponente in die aktuelle ziehen erlaubt 566

OLEGiveFeedback 607

Phantom während eines OLE-Drag-Vorgangs 566

Ziehen über Steuerelement-Grenze hinaus erlaubt 565

OLECompleteDrag*Ereignis von*

VcNet 605

OLEDragDrop*Ereignis von*

VcNet 605

OLEDragMode*Eigenschaft von*

VcNet 564

OLEDragOver*Ereignis von*

VcNet 606

OLEDragWithOwnMouseCursor*Eigenschaft von*

VcNet 565

OLEDragWithPhantom*Eigenschaft von*

VcNet 566

OLEDropMode*Eigenschaft von*

VcNet 566

OLEGiveFeedback*Ereignis von*

VcNet 607

OLESetData*Ereignis von*

VcNet 608

OLEStartDrag*Ereignis von*

VcNet 608

OnBoxLClick*Ereignis von*

VcNet 609

OnBoxLDbIClick*Ereignis von*

VcNet 610

OnBoxModifyComplete*Ereignis von*

VcNet 610

OnBoxModifyCompleteEx*Ereignis von*

VcNet 611

OnBoxRClick*Ereignis von*

VcNet 611

OnDataRecordCreate*Ereignis von*

VcNet 612

OnDataRecordCreateComplete*Ereignis von*

VcNet 613

OnDataRecordDelete*Ereignis von*

VcNet 614

OnDataRecordDeleteComplete*Ereignis von*

- VcNet 614
- OnDataRecordModify**
 - Ereignis von
 - VcNet 615
- OnDataRecordModifyComplete**
 - Ereignis von
 - VcNet 616
- OnDataRecordNotFound**
 - Ereignis von
 - VcNet 616
- OnDiagramLClick**
 - Ereignis von
 - VcNet 616
- OnDiagramLDbIClick**
 - Ereignis von
 - VcNet 617
- OnDiagramRClick**
 - Ereignis von
 - VcNet 617
- OnGiveFeedbackForNodeCreating**
 - Ereignis von
 - VcNet 618
- OnGroupCreate**
 - Ereignis von
 - VcNet 619
- OnGroupDelete**
 - Ereignis von
 - VcNet 619
- OnGroupLClick**
 - Ereignis von
 - VcNet 619
- OnGroupLDbIClick**
 - Ereignis von
 - VcNet 620
- OnGroupModify**
 - Ereignis von
 - VcNet 620
- OnGroupModifyComplete**
 - Ereignis von
 - VcNet 621
- OnGroupRClick**
 - Ereignis von
 - VcNet 622
- OnHelpRequested**
 - Ereignis von
 - VcNet 622
- OnLegendViewClosed**
 - Ereignis von
 - VcNet 623
- OnLinkCreate**
 - Ereignis von
 - VcNet 623
- OnLinkCreateComplete**
 - Ereignis von
 - VcNet 624
- OnLinkDelete**
 - Ereignis von
 - VcNet 625
- OnLinkDeleteComplete**
 - Ereignis von
 - VcNet 625
- OnLinkLClickCltn**
 - Ereignis von
 - VcNet 626
- OnLinkLDbIClickCltn**
 - Ereignis von
 - VcNet 626
- OnLinkModifyComplete**
 - Ereignis von
 - VcNet 627
- OnLinkModifyEx**
 - Ereignis von
 - VcNet 627
- OnLinkRClickCltn**

- Ereignis von*
VcNet 628
- OnLinksMark**
Ereignis von
VcNet 629
- OnLinksMarkComplete**
Ereignis von
VcNet 630
- OnModifyComplete**
Ereignis von
VcNet 630
- OnMouseDbIClk**
Ereignis von
VcNet 631
- OnMouseDown**
Ereignis von
VcNet 631
- OnMouseMove**
Ereignis von
VcNet 632
- OnMouseUp**
Ereignis von
VcNet 632
- OnNodeCreate**
Ereignis von
VcNet 633
- OnNodeCreateCompleteEx**
Ereignis von
VcNet 634
- OnNodeDelete**
Ereignis von
VcNet 635
- OnNodeDeleteCompleteEx**
Ereignis von
VcNet 635
- OnNodeLClick**
Ereignis von
VcNet 636
- OnNodeLDbIClick**
Ereignis von
VcNet 636
- OnNodeModifyComplete**
Ereignis von
VcNet 637
- OnNodeModifyCompleteEx**
Ereignis von
VcNet 637
- OnNodeModifyEx**
Ereignis von
VcNet 638
- OnNodeRClick**
Ereignis von
VcNet 639
- OnNodesMarkComplete**
Ereignis von
VcNet 640
- OnNodesMarkEx**
Ereignis von
VcNet 640
- OnSelectField**
Ereignis von
VcNet 641
- OnShowInPlaceEditor**
Ereignis von
VcNet 641
- OnStatusLineText**
Ereignis von
VcNet 643
- OnSupplyTextEntry**
Ereignis von
VcNet 643
- OnSupplyTextEntryAsVariant**
Ereignis von
VcNet 652

OnSupplyTextEntry-Ereignis
aktivieren 537

OnToolTipText

Ereignis von
VcNet 652

OnToolTipTextAsVariant

Ereignis von
VcNet 653

OnWorldViewClosed

Ereignis von
VcNet 654

OnZoomFactorModifyComplete

Ereignis von
VcNet 654

Open

Methode von
VcNet 590

Operator

Eigenschaft von
VcFilterSubCondition 429

Orientation

Eigenschaft von
VcNet 567
VcPrinter 728

Origin

Eigenschaft von
VcBox 315

OutgoingLinks

Eigenschaft von
VcNode 658

P

PageDescription

Eigenschaft von
VcPrinter 728

PageDescriptionString

Eigenschaft von

VcPrinter 728

PageFrame

Eigenschaft von
VcPrinter 729

PageLayout

Methode von
VcNet 591

PageNumberMode

Eigenschaft von
VcPrinter 729

PageNumbers

Eigenschaft von
VcPrinter 730

PagePaddingEnabled

Eigenschaft von
VcPrinter 730

PaperSize

Eigenschaft von
VcPrinter 731

Papiergröße 731

ParentHWND

Eigenschaft von
VcLegendView 457
VcWorldView 749

PasteNodesFromClipboard

Methode von
VcNet 591

Pattern

Eigenschaft von
VcMapEntry 517
VcNodeAppearance 672

PatternBackgroundColorAsARGB

Eigenschaft von
VcBoxFormatField 343
VcNodeFormatField 709

PatternBackgroundColorDataFieldIndex

- Eigenschaft von*
 - VcNodeFormatField* 710
- PatternBackgroundColorMapName**
 - Eigenschaft von*
 - VcNodeFormatField* 710
- PatternColorAsARGB**
 - Eigenschaft von*
 - VcBoxFormatField* 343
 - VcNodeAppearance* 675
 - VcNodeFormatField* 711
- PatternColorDataFieldIndex**
 - Eigenschaft von*
 - VcNodeAppearance* 676
 - VcNodeFormatField* 711
- PatternColorMapName**
 - Eigenschaft von*
 - VcNodeAppearance* 676
 - VcNodeFormatField* 712
- PatternDataFieldIndex**
 - Eigenschaft von*
 - VcNodeAppearance* 676
- PatternEx**
 - Eigenschaft von*
 - VcBoxFormatField* 344
 - VcNodeFormatField* 712
- PatternExDataFieldIndex**
 - Eigenschaft von*
 - VcNodeFormatField* 713
- PatternExMapName**
 - Eigenschaft von*
 - VcNodeFormatField* 713
- PatternMapName**
 - Eigenschaft von*
 - VcNodeAppearance* 677
- PDF-Dateien**
 - Export 135
- Performance 287**
- Pfad 539**
- Piles**
 - Eigenschaft von*
 - VcNodeAppearance* 677
- PixelsToRaster**
 - Methode von*
 - VcNet* 591
- PixelsToRasterAsVariant**
 - Methode von*
 - VcNet* 592
- Positionen von Knoten und Verbindungsbeschriftungen**
 - speichern und laden 66
- PredecessorNode**
 - Eigenschaft von*
 - VcLink* 464
- PrePortSymbol**
 - Eigenschaft von*
 - VcLinkAppearance* 472
- Primärschlüssel**
 - zusammengesetzt 392
- PrimaryKey**
 - Eigenschaft von*
 - VcDataTableField* 402
- PrintDate**
 - Eigenschaft von*
 - VcPrinter* 731
- PrintDirectEx**
 - Methode von*
 - VcNet* 592
- Printer**
 - Eigenschaft von*
 - VcNet* 567
 - siehe auch
 - VcPrinter* 717
- PrinterName**
 - Eigenschaft von*

VcPrinter 732

PrinterSetup

Methode von

VcNet 593

PrintIt

Methode von

VcNet 594

PrintPreview

Methode von

VcNet 594

PrintToFile

Methode von

VcNet 594

Priorität 197

Boxen 206

Priority

Eigenschaft von

VcBox 316

Projektdatei

Datei für Design-Phase 47

Projektdatei berechnen 743

Publizieren im Internet 83, 252, 274

PutInOrderAfter

Methode von

VcCalendarProfile 365

VcInterval 447

VcLinkAppearance 475

VcNodeAppearance 682

R

Rahmen

außen 266

RasterToPixels

Methode von

VcNet 595

RasterToPixelsAsVariant

Methode von

VcNet 595

Rect

siehe auch

VcRect 734

ReferencePoint

Eigenschaft von

VcBox 316

RelatedDataRecord

Methode von

VcDataRecord 381

VcLink 466

VcNode 659

RelationshipFieldIndex

Eigenschaft von

VcDataTableField 403

Remove

Methode von

DataObjectFiles 301

VcBoxCollection 327

VcBoxFormatCollection 337

VcCalendarCollection 362

VcCalendarProfileCollection 370

VcDataRecordCollection 388

VcFilterCollection 426

VcIntervalCollection 453

VcLinkAppearanceCollection 480

VcLinkFormatCollection 495

VcMapCollection 511

VcNodeAppearanceCollection 688

VcNodeFormatCollection 703

RemoveFormatField

Methode von

VcBoxFormat 331

VcLinkFormat 489

VcNodeFormat 697

RemoveSubCondition

Methode von

VcFilter 419

RepeatTitleAndLegend

Eigenschaft von

VcPrinter 732

Reset

Methode von

VcNet 596

Return Status 100

Right

Eigenschaft von

VcRect 736

RightMargin

Eigenschaft von

VcNodeFormatField 713

RoundedLinkSlantsEnabled

Eigenschaft von

VcNet 568

RoutingType

Eigenschaft von

VcLinkAppearance 472

Rückgabewerte 100

S

SaveAsEx

Methode von

VcNet 596

ScheduledProjectEndDate

Eigenschaft von

VcScheduler 741

ScheduledProjectStartDate

Eigenschaft von

VcScheduler 741

ScheduleProject

Methode von

VcNet 597

VcScheduler 743

Scheduler

Eigenschaft von

VcNet 568

siehe auch

VcScheduler 737

ScheduleSuccessorsOnlyEnabled

Eigenschaft von

VcScheduler 742

Schnittmarkierungen 267

Schnittstelle

Definition 15

einrichten 34

Schriftattribute

Dialogfelder 535

Schriften

Anti-Aliasing 540

ScrollBarMode

Eigenschaft von

VcLegendView 457

VcWorldView 750

Scrollen

zu der Zeile eines Knotens 597

ScrollOffsetX

Eigenschaft von

VcNet 568

ScrollOffsetY

Eigenschaft von

VcNet 569

ScrollToNodePosition

Methode von

VcNet 597

SecondsPerWorkday

Eigenschaft von

VcCalendar 351

Seite einrichten 273, 591

Seitenansicht 732

Seitennummerierung 267

Seitenränder 268

SelectCalendarProfiles

Methode von
VcCalendarProfileCollection 370

SelectLinks

Methode von
VcLinkCollection 484

SelectMaps

Methode von
VcMapCollection 511

SelectNodes

Methode von
VcNodeCollection 691

SetData

Methode von
DataObject 297

SetXY

Methode von
VcGroup 437

SetXYOffset

Methode von
VcBox 320

SetXYOffsetByTopLeftPixel

Methode von
VcBox 320

Shadow

Eigenschaft von
VcNodeAppearance 677

ShadowColorAsARGB

Eigenschaft von
VcNodeAppearance 678

ShortenedLinks

Eigenschaft von
VcNet 569

ShowAlwaysCompleteView

Methode von
VcNet 598

ShowExportGraphicsDialog

Methode von
VcNet 598

ShowToolTip

Eigenschaft von
VcNet 570

Spalte

minimale Breite 559

Specification

Eigenschaft von
VcBox 317
VcBoxFormat 330
VcCalendar 351
VcCalendarProfile 364
VcFilter 416
VcInterval 445
VcLinkAppearance 473
VcLinkFormat 488
VcMap 502
VcNodeAppearance 679
VcNodeFormat 695

Sprachanpassung 137

StartDateForAutomaticScheduling

Eigenschaft von
VcScheduler 742

StartDateNotEarlierThanDataFieldIndex

Eigenschaft von
VcScheduler 742

StartDateTime

Eigenschaft von
VcInterval 446

StartMonth

Eigenschaft von
VcInterval 446

StartTime

Eigenschaft von
VcInterval 446

StartUpSinglePage

Eigenschaft von
VcPrinter 732

StartWeekday

Eigenschaft von
VcInterval 447

Statuszeilentext 138, 643**StraightLinkDrawing**

Eigenschaft von
VcNet 570

Strg+C, Strg+X und Strg+V 531**StrikeThrough**

Eigenschaft von
VcNodeAppearance 679

StrikeThroughColor

Eigenschaft von
VcNodeAppearance 680

StringsCaseSensitive

Eigenschaft von
VcFilter 417

SubCondition

Eigenschaft von
VcFilter 417

SubConditionCount

Eigenschaft von
VcFilter 417

SuccessorNode

Eigenschaft von
VcLink 464

SuccPortSymbol

Eigenschaft von
VcLinkAppearance 474

SuspendUpdate

Methode von
VcNet 600

T**Tabellendatenfeld**

Anzahl in Collection 406
Iteration, Enumerator Objekt 406
Iteration, Erstwert 408
Iteration, Folgewerte 409
kopieren 407
über Index 407
über Name 408
zu Collection hinzufügen 406

Taste

Ereignis beim Drücken 603
Ereignis beim Drücken und Loslassen
604
Ereignis beim Loslassen 604

Technische Voraussetzungen 12**Teilnetz 273, 276****Text**

Eigenschaft von
VcBoundingBox 308

Textausgabe 643, 652**TextDataFieldIndex**

Eigenschaft von
VcLinkFormatField 498
VcNodeFormatField 714

Texte

festlegen 245

**Texte, Grafiken und Legende
festlegen 245****TextFont**

Eigenschaft von
VcBoundingBox 308
VcBoxFormatField 347
VcLinkFormatField 498
VcNodeFormatField 714

TextFontColor

Eigenschaft von

- VcBoxFormatField* 347
- VcLinkFormatField* 499
- VcNodeFormatField* 714
- TextFontDataFieldIndex**
 - Eigenschaft von
 - VcNodeFormatField* 715
- TextFontMapName**
 - Eigenschaft von
 - VcNodeFormatField* 715
- TextLineCount**
 - Eigenschaft von
 - VcLinkFormatField* 499
- ThreeDEffect**
 - Eigenschaft von
 - VcNodeAppearance* 681
- TimeUnit**
 - Eigenschaft von
 - VcNet* 570
- Titel**
 - in Darstellung wiederholen 266
- Title**
 - Eigenschaft von
 - VcGroup* 436
- TitleLineCount**
 - Eigenschaft von
 - VcGroup* 436
- Tooltip 570, 653**
 - Datenfeld für Text 166, 563
- ToolTip**
 - Dauer bis zur Anzeige 572
 - Erscheinungsdauer 571
 - Verschwinden auf Klick 572
 - Wechseldauer 571
- ToolTipChangeDuration**
 - Eigenschaft von
 - VcNet* 571
- ToolTipDuration**
 - Eigenschaft von
 - VcNet* 571
- ToolTipPointerDuration**
 - Eigenschaft von
 - VcNet* 572
- Tooltips**
 - zur Laufzeit 139
- ToolTipShowAfterClick**
 - Eigenschaft von
 - VcNet* 572
- Top**
 - Eigenschaft von
 - VcLegendView* 458
 - VcRect* 736
 - VcWorldView* 750
- TopActualValue**
 - Eigenschaft von
 - VcLegendView* 458
 - VcWorldView* 751
- TopMargin**
 - Eigenschaft von
 - VcNodeFormatField* 715
- TotalFloatDataFieldIndex**
 - Eigenschaft von
 - VcScheduler* 743
- Type**
 - Eigenschaft von
 - VcBoundingBox* 309
 - VcBoxFormatField* 348
 - VcCalendarProfile* 364
 - VcDataTableField* 404
 - VcDefinitionField* 413
 - VcInterval* 447
 - VcMap* 502
 - VcNodeFormatField* 716

U

UngroupedNodesAllowed*Eigenschaft von**VcNet* 572**Unicode 140****Update***Methode von**VcBoxCollection* 327*VcCalendar* 355*VcCalendarCollection* 362*VcCalendarProfileCollection* 371*VcDataRecordCollection* 388*VcDataTableCollection* 398*VcIntervalCollection* 453*VcLegendView* 461*VcLinkAppearanceCollection* 481*VcMapCollection* 512**UpdateBehaviorName***Eigenschaft von**VcBox* 317*VcWorldView* 751**UpdateDataRecord***Methode von**VcDataRecord* 382**UpdateLink***Methode von**VcLink* 466**UpdateLinkRecord***Methode von**VcNet* 601**UpdateNode***Methode von**VcNode* 660**UpdateNodeRecord***Methode von**VcNet* 601**URL 531**

V

VARCHART XNet

automatisch skalieren 33

im Formular plazieren 31

zur Werkzeugsammlung hinzufügen
30**VcBorderArea 302***BorderBox* 302**VcBorderBox 303***Alignment* 303*GraphicsFileName* 304*LegendElementsArrangement* 305*LegendElementsBottomMargin* 305*LegendElementsMaximumColumnCo
unt* 305*LegendElementsMaximumRowCount*
306*LegendElementsTopMargin* 306*LegendFont* 306*LegendTitle* 306*LegendTitleFont* 307*LegendTitleVisible* 307*Text* 308*TextFont* 308*Type* 309**VcBox 310***FieldText* 311*FormatName* 311*GetActualExtent* 318*GetTopLeftPixel* 319*GetXYOffset* 319*GetXYOffsetAsVariant* 319*IdentifyFormatField* 320*LineColor* 312*LineThickness* 312

- LineType* 313
- MarkBox* 314
- Moveable* 314
- Name* 315
- Origin* 315
- Priority* 316
- ReferencePoint* 316
- SetXYOffset* 320
- SetXYOffsetByTopLeftPixel* 320
- Specification* 317
- UpdateBehaviorName* 317
- Visible* 318
- VcBoxCollection 322**
 - _NewEnum* 322
 - Add* 323
 - AddBySpecification* 324
 - BoxByIndex* 324
 - BoxByName* 325
 - Copy* 325
 - Count* 323
 - FirstBox* 326
 - NextBox* 326
 - Remove* 327
 - Update* 327
- VcBoxFormat 328**
 - _NewEnum* 328
 - CopyFormatField* 331
 - FieldsSeparatedByLines* 329
 - FormatField* 329
 - FormatFieldCount* 330
 - Name* 330
 - RemoveFormatField* 331
 - Specification* 330
- VcBoxFormatCollection 333**
 - _NewEnum* 333
 - Add* 334
 - AddBySpecification* 335
 - Copy* 335
 - Count* 334
 - FirstFormat* 336
 - FormatByIndex* 336
 - FormatByName* 336
 - NextFormat* 337
 - Remove* 337
- VcBoxFormatField 339**
 - Alignment* 339
 - FormatName* 340
 - GraphicsHeight* 340
 - Index* 341
 - MaximumTextLineCount* 341
 - MinimumTextLineCount* 342
 - MinimumWidth* 342
 - PatternBackgroundColorAsARGB* 343
 - PatternColorAsARGB* 343
 - PatternEx* 344
 - TextFont* 347
 - TextFontColor* 347
 - Type* 348
- VcCalendar 349**
 - AddDuration* 351
 - CalcDuration* 352
 - CalendarProfileCollection* 350
 - Clear* 352
 - GetEndOfPreviousWorktime* 353
 - GetNextIntervalBorder* 353
 - GetPreviousIntervalBorder* 354
 - GetStartOfInterval* 354
 - GetStartOfNextWorktime* 355
 - IntervalCollection* 350
 - IsWorktime* 355
 - Name* 350
 - SecondsPerWorkday* 351
 - Specification* 351

- Update* 355
- VcCalendarCollection 357**
 - _NewEnum* 357
 - Active* 358
 - Add* 359
 - AddBySpecification* 359
 - CalendarByIndex* 360
 - CalendarByName* 360
 - Copy* 360
 - Count* 359
 - FirstCalendar* 361
 - NextCalendar* 361
 - Remove* 362
 - Update* 362
- VcCalendarProfile 363**
 - IntervalCollection* 363
 - Name* 364
 - PutInOrderAfter* 365
 - Specification* 364
 - Type* 364
- VcCalendarProfileCollection 366**
 - _NewEnum* 367
 - Add* 367
 - AddBySpecification* 368
 - CalendarProfileByIndex* 368
 - CalendarProfileByName* 369
 - Copy* 369
 - Count* 367
 - FirstCalendarProfile* 369
 - NextCalendarProfile* 370
 - Remove* 370
 - SelectCalendarProfiles* 370
 - Update* 371
- VcDataDefinition 372**
 - DefinitionTable* 372
- VcDataDefinitionTable 373**
 - _NewEnum* 373
- Count* 374
- CreateDataField* 374
- FieldByIndex* 375
- FieldByName* 375
- FirstField* 376
- NextField* 376
- VcDataRecord 378**
 - AllData* 378
 - DataField* 379
 - DataTableName* 380
 - DeleteDataRecord* 380
 - ID* 380
 - IdentifyObject* 381
 - RelatedDataRecord* 381
 - UpdateDataRecord* 382
- VcDataRecordCollection 383**
 - _NewEnum* 384
 - Add* 385
 - Count* 384
 - DataRecordByID* 386
 - FirstDataRecord* 386
 - GetNewUniqueID* 387
 - NextDataRecord* 387
 - Remove* 388
 - Update* 388
- VcDataTable 390**
 - DataRecordCollection* 390
 - DataTableFieldCollection* 391
 - Description* 391
 - MultiplePrimaryKeysAllowed* 392
 - Name* 392
- VcDataTableCollection 393**
 - _NewEnum* 393
 - Add* 394
 - Copy* 395
 - Count* 394
 - DataTableByIndex* 396

- DataTableByName* 396
- FirstDataTable* 397
- NextDataTable* 397
- Update* 398
- VcDataTableField 399**
 - DataTableName* 400
 - DateFormat* 400
 - Editable* 401
 - Hidden* 401
 - Index* 402
 - Name* 402
 - PrimaryKey* 402
 - RelationshipFieldIndex* 403
 - Type* 404
- VcDataTableFieldCollection 405**
 - _NewEnum* 405
 - Add* 406
 - Copy* 407
 - Count* 406
 - DataTableFieldByIndex* 407
 - DataTableFieldByName* 408
 - FirstDataTableField* 408
 - NextDataTableField* 409
- VcDefinitionField 410**
 - DateFormat* 410
 - Editable* 411
 - Hidden* 411
 - ID* 412
 - Name* 412
 - Type* 413
- VcFilter 414**
 - _NewEnum* 415
 - AddSubCondition* 418
 - CopySubCondition* 418
 - DataDefinitionTable* 415
 - DatesWithHourAndMinute* 416
 - Evaluate* 419
 - IsValid* 419
 - Name* 416
 - RemoveSubCondition* 419
 - Specification* 416
 - StringsCaseSensitive* 417
 - SubCondition* 417
 - SubConditionCount* 417
- VcFilterCollection 421**
 - _NewEnum* 421
 - Add* 423
 - AddBySpecification* 423
 - Copy* 424
 - Count* 422
 - FilterByIndex* 424
 - FilterByName* 424
 - FirstFilter* 425
 - MarkedNodesFilter* 422
 - NextFilter* 425
 - Remove* 426
- VcFilterSubCondition 427**
 - ComparisonValueAsString* 427
 - ConnectionOperator* 428
 - DataFieldIndex* 429
 - FilterName* 429
 - Index* 429
 - IsValid* 430
 - Operator* 429
- VcGroup 431**
 - BackColor* 431
 - Collapsed* 432
 - LineColor* 432
 - LineThickness* 433
 - LineType* 434
 - Name* 435
 - NodeCollection* 435
 - SetXY* 437
 - Title* 436

- TitleLineCount* 436
- X* 437
- Y* 437
- VcGroupCollection 438**
 - _NewEnum* 438
 - Count* 439
 - FirstGroup* 439
 - GroupByName* 440
 - NextGroup* 440
- VcInterval 441**
 - CalendarProfileName* 443
 - DayInEndMonth* 443
 - DayInStartMonth* 443
 - EndDateTime* 444
 - EndMonth* 444
 - EndTime* 444
 - EndWeekday* 445
 - Name* 445
 - PutInOrderAfter* 447
 - Specification* 445
 - StartDateTime* 446
 - StartMonth* 446
 - StartTime* 446
 - StartWeekday* 447
 - Type* 447
- VcIntervalCollection 449**
 - _NewEnum* 450
 - Add* 450
 - AddBySpecification* 451
 - Copy* 451
 - Count* 450
 - FirstInterval* 452
 - IntervalByIndex* 452
 - IntervalByName* 452
 - NextInterval* 453
 - Remove* 453
 - Update* 453
- VcLegendView 454**
 - Border* 454
 - Height* 455
 - HeightActualValue* 455
 - Left* 456
 - LeftActualValue* 456
 - ParentHWND* 457
 - ScrollBarMode* 457
 - Top* 458
 - TopActualValue* 458
 - Update* 461
 - Visible* 459
 - Width* 459
 - WidthActualValue* 460
 - WindowMode* 460
- VcLink 462**
 - AllData* 462
 - DataField* 463
 - DataRecord* 465
 - DeleteLink* 465
 - ID* 463
 - MarkLink* 464
 - PredecessorNode* 464
 - RelatedDataRecord* 466
 - SuccessorNode* 464
 - UpdateLink* 466
- VcLinkAppearance 467**
 - FilterName* 467
 - FormatName* 468
 - LineColor* 468
 - LineThickness* 469
 - LineType* 470
 - Name* 471
 - PrePortSymbol* 472
 - PutInOrderAfter* 475
 - RoutingType* 472
 - Specification* 473

- SuccPortSymbol* 474
- Visible* 474
- VcLinkAppearanceCollection 476**
 - _NewEnum* 476
 - Add* 477
 - AddBySpecification* 478
 - Copy* 478
 - Count* 477
 - FirstLinkAppearance* 479
 - LinkAppearanceByIndex* 479
 - LinkAppearanceByName* 479
 - NextLinkAppearance* 480
 - Remove* 480
 - Update* 481
- VcLinkCollection 482**
 - _NewEnum* 482
 - Count* 483
 - FirstLink* 483
 - NextLink* 484
 - SelectLinks* 484
- VcLinkFormat 486**
 - _NewEnum* 486
 - CopyFormatField* 488
 - FormatField* 487
 - FormatFieldCount* 487
 - Name* 488
 - RemoveFormatField* 489
 - Specification* 488
- VcLinkFormatCollection 490**
 - _NewEnum* 490
 - Add* 491
 - AddBySpecification* 492
 - Copy* 493
 - Count* 491
 - FirstFormat* 493
 - FormatByIndex* 494
 - FormatByName* 494
 - NextFormat* 494
 - Remove* 495
- VcLinkFormatField 496**
 - Alignment* 496
 - ConstantText* 497
 - FormatName* 497
 - Index* 497
 - MinimumWidth* 498
 - TextDataFieldIndex* 498
 - TextFont* 498
 - TextFontColor* 499
 - TextLineCount* 499
- VcMap 500**
 - _NewEnum* 500
 - ConsiderFilterEntries* 501
 - Count* 501
 - CreateEntry* 503
 - DeleteEntry* 503
 - FirstMapEntry* 504
 - GetMapEntry* 505
 - Name* 502
 - NextMapEntry* 505
 - Specification* 502
 - Type* 502
- VcMapCollection 506**
 - _NewEnum* 507
 - Add* 508
 - AddBySpecification* 508
 - Copy* 509
 - Count* 507
 - FirstMap* 509
 - MapByIndex* 509
 - MapByName* 510
 - NextMap* 510
 - Remove* 511
 - SelectMaps* 511
 - Update* 512

VcMapEntry 513

ColorAsARGB 513
DataFieldValue 514
FontBody 514
FontName 515
FontSize 515
GraphicsFileName 516
Pattern 517

VcNet 521

AboutBox 574
ActiveNodeFilter 527
AllowMultipleBoxMarking 527
AllowNewNodesAndLinks 528
Arrange 575
AssignCalendarToNodes 528
BorderArea 529
BoxCollection 529
BoxFormatCollection 529
CalendarCollection 530
CalendarProfileCollection 530
Clear 575
ConfigurationName 530
CopyNodesIntoClipboard 575
CtrlCXVProcessing 531
CurrentVersion 532
CutNodesIntoClipboard 576
DataDefinition 532
DataTableCollection 533
DateOutputFormat 533
DeleteLinkRecord 576
DeleteNodeRecord 577
DetectDataTableFieldName 577
DetectDataTableName 577
DetectFieldIndex 578
DiagramBackColor 535
DialogFont 535
DoubleOutputFormat 535

DumpConfiguration 578
EditLink 579
EditNewLink 536
EditNewNode 536
EditNode 579
Enabled 537
EnableSupplyTextEntryEvent 537
EndLoading 580
Error 602
ErrorAsVariant 603
EventReturnStatus 537
EventText 538
ExportGraphicsToFile 580
ExtendedDataTables 538
FilePath 539
FilterCollection 540
FontAntiAliasingEnabled 540
GetAValueFromARGB 582
GetBValueFromARGB 583
GetGValueFromARGB 584
GetLinkByID 584
GetLinkByIDs 585
GetNodeByID 585
GetRValueFromARGB 586
GroupCollection 541
GroupDescriptionName 541
GroupField 542
GroupHorizontalMargin 542
Grouping 542
GroupingTitlesFullyVisible 543
GroupInteractionsAllowed 543
GroupMode 544
GroupMovingAllowed 544
GroupSortField 545
GroupSortMode 545
GroupTitleField 546
GroupVerticalMargin 546

- hWnd* 547
- IdentifyFormatField* 586
- IdentifyFormatFieldAsVariant* 587
- IdentifyObjectAt* 587
- IdentifyObjectAtAsVariant* 588
- InFlowGroupDescriptionName* 547
- InFlowGroupField* 548
- InFlowGroupingEnabled* 548
- InFlowGroupSeparationLineColor* 548
- InFlowGroupSeparationLineType* 549
- InFlowGroupTimeInterval* 550
- InFlowGroupTitleField* 550
- InFlowGroupTitlesBackColor* 551
- InFlowGroupTitlesFont* 551
- InFlowGroupTitlesVisibleAtBottomOrRight* 551
- InFlowGroupTitlesVisibleAtTopOrLeft* 551
- InFlowGroupTitleTimeFormat* 552
- InFlowGroupVerticalCaptionWidth* 552
- InPlaceEditingAllowed* 552
- InsertLinkRecord* 589
- InsertNodeRecord* 589
- InteractionMode* 553
- InterfaceNodesShown* 553
- KeyDown* 603
- KeyPress* 604
- KeyUp* 604
- LegendView* 554
- LinkAnnotationColumnNumberDataFieldIndex* 554
- LinkAnnotationRowNumberDataFieldIndex* 554
- LinkAppearanceCollection* 555
- LinkCollection* 555
- LinkFormatCollection* 555
- LinkPredecessorDataFieldIndex* 556
- LinksDataTableName* 556
- LinkSuccessorDataFieldIndex* 557
- LinkTypeDataFieldIndex* 558
- MakeARGB* 590
- MapCollection* 558
- MinimumColumnWidth* 559
- MinimumRowHeight* 559
- MouseProcessingEnabled* 560
- NodeAppearanceCollection* 560
- NodeCalendarNameDataFieldIndex* 560
- NodeChangeRankToPredecessorRankDataFieldIndex* 561
- NodeCollection* 561
- NodeColumnNumberDataFieldIndex* 561
- NodeFormatCollection* 562
- NodeRowNumberDataFieldIndex* 562
- NodesDataTableName* 563
- NodeTooltipTextField* 563
- ObliqueTracksOnLinks* 564
- OLECompleteDrag* 605
- OLEDragDrop* 605
- OLEDragMode* 564
- OLEDragOver* 606
- OLEDragWithOwnMouseCursor* 565
- OLEDragWithPhantom* 566
- OLEDropMode* 566
- OLEGiveFeedback* 607
- OLESetData* 608
- OLEStartDrag* 608
- OnBoxLClick* 609
- OnBoxLDbIClick* 610
- OnBoxModifyComplete* 610
- OnBoxModifyCompleteEx* 611
- OnBoxRClick* 611
- OnDataRecordCreate* 612
- OnDataRecordCreateComplete* 613

- OnDataRecordDelete* 614
- OnDataRecordDeleteComplete* 614
- OnDataRecordModify* 615
- OnDataRecordModifyComplete* 616
- OnDataRecordNotFound* 616
- OnDiagramLClick* 616
- OnDiagramLDbIClick* 617
- OnDiagramRClick* 617
- OnGiveFeedbackForNodeCreating* 618
- OnGroupCreate* 619
- OnGroupDelete* 619
- OnGroupLClick* 619
- OnGroupLDbIClick* 620
- OnGroupModify* 620
- OnGroupModifyComplete* 621
- OnGroupRClick* 622
- OnHelpRequested* 622
- OnLegendViewClosed* 623
- OnLinkCreate* 623
- OnLinkCreateComplete* 624
- OnLinkDelete* 625
- OnLinkDeleteComplete* 625
- OnLinkLClickCltn* 626
- OnLinkLDbIClickCltn* 626
- OnLinkModifyComplete* 627
- OnLinkModifyEx* 627
- OnLinkRClickCltn* 628
- OnLinksMark* 629
- OnLinksMarkComplete* 630
- OnModifyComplete* 630
- OnMouseDbIClick* 631
- OnMouseDown* 631
- OnMouseMove* 632
- OnMouseUp* 632
- OnNodeCreate* 633
- OnNodeCreateCompleteEx* 634
- OnNodeDelete* 635
- OnNodeDeleteCompleteEx* 635
- OnNodeLClick* 636
- OnNodeLDbIClick* 636
- OnNodeModifyComplete* 637
- OnNodeModifyCompleteEx* 637
- OnNodeModifyEx* 638
- OnNodeRClick* 639
- OnNodesMarkComplete* 640
- OnNodesMarkEx* 640
- OnSelectField* 641
- OnShowInPlaceEditor* 641
- OnStatusLineText* 643
- OnSupplyTextEntry* 643
- OnSupplyTextEntryAsVariant* 652
- OnToolTipText* 652
- OnToolTipTextAsVariant* 653
- OnWorldViewClosed* 654
- OnZoomFactorModifyComplete* 654
- Open* 590
- Orientation* 567
- PageLayout* 591
- PasteNodesFromClipboard* 591
- PixelsToRaster* 591
- PixelsToRasterAsVariant* 592
- PrintDirectEx* 592
- Printer* 567
- PrinterSetup* 593
- PrintIt* 594
- PrintPreview* 594
- PrintToFile* 594
- RasterToPixels* 595
- RasterToPixelsAsVariant* 595
- Reset* 596
- RoundedLinkSlantsEnabled* 568
- SaveAsEx* 596
- ScheduleProject* 597

- Scheduler* 568
- ScrollOffsetX* 568
- ScrollOffsetY* 569
- ScrollToNodePosition* 597
- ShortenedLinks* 569
- ShowAlwaysCompleteView* 598
- ShowExportGraphicsDialog* 598
- ShowToolTip* 570
- StraightLinkDrawing* 570
- SuspendUpdate* 600
- TimeUnit* 570
- ToolTipChangeDuration* 571
- ToolTipDuration* 571
- ToolTipPointerDuration* 572
- ToolTipShowAfterClick* 572
- UngroupedNodesAllowed* 572
- UpdateLinkRecord* 601
- UpdateNodeRecord* 601
- WaitCursorEnabled* 573
- WorldView* 573
- Zoom* 601
- ZoomFactor* 574
- ZoomingPerMouseWheelAllowed* 574
- ZoomOnMarkedNodes* 602
- VcNode 655**
 - AllData* 655
 - DataField* 656
 - DataRecord* 658
 - DeleteNode* 659
 - ID* 656
 - IncomingLinks* 657
 - MarkNode* 657
 - OutgoingLinks* 658
 - RelatedDataRecord* 659
 - UpdateNode* 660
- VcNodeAppearance 661**
 - BackColorAsARGB* 662
 - BackColorDataFieldIndex* 663
 - BackColorMapName* 663
 - DoubleFeature* 664
 - FilterName* 664
 - FormatName* 665
 - FrameAroundFieldsVisible* 665
 - FrameShape* 666
 - LegendText* 667
 - LineColor* 668
 - LineColorDataFieldIndex* 668
 - LineColorMapName* 668
 - LineThickness* 669
 - LineType* 670
 - Name* 671
 - Pattern* 672
 - PatternColorAsARGB* 675
 - PatternColorDataFieldIndex* 676
 - PatternColorMapName* 676
 - PatternDataFieldIndex* 676
 - PatternMapName* 677
 - Piles* 677
 - PutInOrderAfter* 682
 - Shadow* 677
 - ShadowColorAsARGB* 678
 - Specification* 679
 - StrikeThrough* 679
 - StrikeThroughColor* 680
 - ThreeDEffect* 681
 - VisibleInLegend* 681
- VcNodeAppearanceCollection 683**
 - _NewEnum* 683
 - Add* 684
 - AddBySpecification* 685
 - Copy* 685
 - Count* 684
 - FirstNodeAppearance* 686

- NextNodeAppearance* 686
- NodeAppearanceByIndex* 687
- NodeAppearanceByName* 687
- Remove* 688
- VcNodeCollection 689**
 - _NewEnum* 689
 - Count* 690
 - FirstNode* 690
 - NextNode* 691
 - SelectNodes* 691
- VcNodeFormat 693**
 - _NewEnum* 693
 - CopyFormatField* 696
 - FieldsSeparatedByLines* 694
 - FormatField* 694
 - FormatFieldCount* 695
 - Name* 695
 - RemoveFormatField* 697
 - Specification* 695
 - WidthOfExteriorSurrounding* 696
- VcNodeFormatCollection 698**
 - _NewEnum* 698
 - Add* 699
 - AddBySpecification* 700
 - Copy* 701
 - Count* 699
 - FirstFormat* 701
 - FormatByIndex* 702
 - FormatByName* 702
 - NextFormat* 702
 - Remove* 703
- VcNodeFormatField 704**
 - Alignment* 705
 - BottomMargin* 705
 - CombiField* 706
 - ConstantText* 706
 - FormatName* 706
 - GraphicsFileName* 706
 - GraphicsFileNameDataFieldIndex* 707
 - GraphicsFileNameMapName* 707
 - GraphicsHeight* 707
 - Index* 708
 - LeftMargin* 708
 - MaximumTextLineCount* 708
 - MinimumTextLineCount* 709
 - MinimumWidth* 709
 - PatternBackgroundColorAsARGB* 709
 - PatternBackgroundColorDataFieldIndex* 710
 - PatternBackgroundColorMapName* 710
 - PatternColorAsARGB* 711
 - PatternColorDataFieldIndex* 711
 - PatternColorMapName* 712
 - PatternEx* 712
 - PatternExDataFieldIndex* 713
 - PatternExMapName* 713
 - RightMargin* 713
 - TextDataFieldIndex* 714
 - TextFont* 714
 - TextFontColor* 714
 - TextFontDataFieldIndex* 715
 - TextFontMapName* 715
 - TopMargin* 715
 - Type* 716
- VcPrinter 717**
 - AbsoluteBottomMarginInCM* 718
 - AbsoluteBottomMarginInInches* 718
 - AbsoluteLeftMarginInCM* 719
 - AbsoluteLeftMarginInInches* 719
 - AbsoluteRightMarginInCM* 719
 - AbsoluteRightMarginInInches* 720
 - AbsoluteTopMarginInCM* 720

- AbsoluteTopMarginInInches* 721
- Alignment* 721
- CurrentHorizontalPagesCount* 722
- CurrentVerticalPagesCount* 722
- CurrentZoomFactor* 722
- CuttingMarks* 723
- DefaultPrinterName* 723
- DocumentName* 723
- FitToPage* 724
- FoldingMarksType* 724
- MarginsShownInInches* 726
- MaxHorizontalPagesCount* 727
- MaxVerticalPagesCount* 727
- Orientation* 728
- PageDescription* 728
- PageDescriptionString* 728
- PageFrame* 729
- PageNumberMode* 729
- PageNumbers* 730
- PagePaddingEnabled* 730
- PaperSize* 731
- PrintDate* 731
- PrinterName* 732
- RepeatTitleAndLegend* 732
- StartUpSinglePage* 732
- ZoomFactorAsDouble* 733
- VcRect 734**
 - Bottom* 734
 - Height* 734
 - Left* 735
 - Right* 736
 - Top* 736
 - Width* 736
- VcScheduler 737**
 - ActualEndDateDataFieldIndex* 737
 - ActualStartDateDataFieldIndex* 738
 - AutomaticSchedulingEnabled* 738
 - DurationDataFieldIndex* 738
 - EarlyEndDateDataFieldIndex* 739
 - EarlyStartDateDataFieldIndex* 739
 - EndDateForAutomaticScheduling* 739
 - EndDateNotLaterThanDataFieldIndex* 740
 - FreeFloatDataFieldIndex* 740
 - LateEndDateDataFieldIndex* 740
 - LateStartDateDataFieldIndex* 740
 - LinkDurationDataFieldIndex* 741
 - ScheduledProjectEndDate* 741
 - ScheduledProjectStartDate* 741
 - ScheduleProject* 743
 - ScheduleSuccessorsOnlyEnabled* 742
 - StartDateForAutomaticScheduling* 742
 - StartDateNotEarlierThanDataFieldIndex* 742
 - TotalFloatDataFieldIndex* 743
- VcWorldView 745**
 - Border* 745
 - Height* 746
 - HeightActualValue* 746
 - Left* 747
 - LeftActualValue* 747
 - MarkingColor* 748
 - Mode* 748
 - ParentHWnd* 749
 - ScrollBarMode* 750
 - Top* 750
 - TopActualValue* 751
 - UpdateBehaviorName* 751
 - Visible* 751
 - Width* 752
 - WidthActualValue* 752
- Verbindung**

- bearbeiten 261
- Datensatz 465
- ID 463
- markiert/nicht markiert 464
- Nachfolgerknoten 557
- Vorgängerknoten 556
- zugeordneter Datensatz 466
- Verbindungen 141, 176**
 - aktualisieren 466
 - Anzahl 483
 - anzeigen 224
 - bearbeiten 48, 579
 - Daten 462
 - Daten aktualisieren 601
 - Datenfeld 463
 - erzeugen 48, 623, 624
 - interaktiv erzeugen 283, 284
 - interaktiv verändern 627, 628
 - laden 589
 - löschen 465, 576, 625
 - markieren 51, 629
 - Markierungstyp 51, 177
 - Nachfolgerknoten 176, 464
 - neue bearbeiten 536
 - orthogonal 144, 564
 - Positionen von
 - Verbindungsbeschriftungen 117
 - schräg 144, 564
 - Verbindungsausssehen 61
 - Verbindungsausssehen verwalten 224
 - Verbindungsformat bearbeiten 221
 - Verbindungsformate verwalten 219
 - Verbindungstyp 177
 - verkürzt dargestellt 569
 - Vorgängerknoten 176, 464
- Verbindungsausssehen 147**
 - Filter 467
 - Formatname 468
 - Linienfarbe 468
 - Linienstärke 469
 - Linientyp 470
 - Name 471
 - Portsymbol für Nachfolgerknoten 474
 - Portsymbol für Vorgängerknoten 472
 - Reihenfolge 475
 - sichtbar 474
 - Typ der Verbindungsrouten 473
- Verbindungsausssehen-Auflistung**
 - hinzufügen 477
 - hinzufügen über Spezifikation 478
 - Kopieren 478
 - löschen 480
- Verbindungsbeschriftungspositionen 66**
- Verbindungsformat-Auflistung**
 - Anzahl 491
 - Enumerator 491
 - erstes Format 493
 - hinzufügen 492
 - hinzufügen über Spezifikation 492
 - kopieren 493
 - Löschen 495
 - nächstes Format 495
 - Zugriff über Name 494
- Versionsnummer**
 - anzeigen 532
- Verzeichnispfad 539**
- Visible**
 - Eigenschaft von*
 - VcBox 318
 - VcLegendView 459
 - VcLinkAppearance 474
 - VcWorldView 751
- VisibleInLegend**

Eigenschaft von

VcNodeAppearance 681

Visual Studio 6.0 mit Visual C++/MFC
18

Visualisierungsmodus 109, 162, 544

Vorgänge

bearbeiten 259

W

WaitCursorEnabled

Eigenschaft von

VcNet 573

Width

Eigenschaft von

VcLegendView 459

VcRect 736

VcWorldView 752

WidthActualValue

Eigenschaft von

VcLegendView 460

VcWorldView 752

WidthOfExteriorSurrounding

Eigenschaft von

VcNodeFormat 696

WindowMode

Eigenschaft von

VcLegendView 460

Worldview 127, 573, 654

WorldView

Eigenschaft von

VcNet 573

Name UpdateBehavior 751

siehe auch

VcWorldView 745

X

X

Eigenschaft von

VcGroup 437

Y

Y

Eigenschaft von

VcGroup 437

Z

Zeile

minimale Höhe 559

Zeitberechnung

automatisch 738

Zeiteinheit 571

Zeitrechnung 76, 149, 178, 568, 597

Aktuelles Anfangsdatum 738

Aktuelles Enddatum 737

Dauer 738

freier Puffer 740

frühester Projektanfang 742

frühestes Projektende 741

frühestmögliches Anfangsdatum 739

frühestmögliches Enddatum 739

geplantes Anfangsdatum 742

geplantes Enddatum 740

Gesamtpuffer 743

nur Knoten mit Vorgängern

berechnen 742

Projektanfang 742

spätestmögliches Anfangsdatum 740

spätestmögliches Enddatum 740

Verbindungsdauer 741

Zeitrechnungseingabe 178

Zeitrechnungsergebnis 178

Zeitumstellung 98

Zoom

Methode von

- VcNet 601
- Zoomen 257, 601**
 - auf markierte Knoten 602
 - per Mausehrad 574
 - Worldview 654
 - Zoomfaktor 574
- ZoomFactor**
 - Eigenschaft von VcNet 574
- ZoomFactorAsDouble**
 - Eigenschaft von VcPrinter 733
- ZoomingPerMouseWheelAllowed**
 - Eigenschaft von VcNet 574
- ZoomOnMarkedNodes**
 - Methode von VcNet 602
- Zuordnungstabelle**
 - Typ 502
 - über Index 510
 - Zuordnungstabelle bearbeiten 191
- Zuordnungstabellen 151, 558**
 - Angabe von Wertebereichen durch Filter 501
 - Zuordnungstabellen verwalten 189
- Zusatztext 268**
- Zwischenspeicher 576**