

## AGIMENDO.calc

---

### Handbuch und Anwenderdokumentation

Produktversion	v3.0.2
Handbuchversion	v3.0.2
Status	Freigegeben
Datum	10.05.2017
Verantwortlich	Felix Weyde ( <a href="mailto:felix.weyde@ibsolution.de">felix.weyde@ibsolution.de</a> )

## 1. Inhaltsverzeichnis

<b>1.</b>	<b>Inhaltsverzeichnis .....</b>	<b>2</b>
<b>2.</b>	<b>Einführung .....</b>	<b>3</b>
2.1	<i>Funktionsweise .....</i>	4
2.2	<i>Knotentypen .....</i>	10
2.3	<i>Bedingungen im Baum.....</i>	11
<b>3.</b>	<b>Pflege Rechenregeln .....</b>	<b>13</b>
3.1	<i>Baumauswahl, Anlage und Pflege .....</i>	13
3.2	<i>Rechenbaumpflege.....</i>	14
3.2.1	<i>Baumpflege .....</i>	15
3.2.2	<i>Beziehungspflege .....</i>	16
3.2.3	<i>Bedingungspflege .....</i>	18
3.2.4	<i>Eigenschaftspflege.....</i>	18
3.2.5	<i>Testwerte .....</i>	24
<b>4.</b>	<b>Abbildung von Mapping-Regeln in AGIMENDO.calc.....</b>	<b>25</b>
<b>5.</b>	<b>Erweiterte Rechenregeln und UserExits .....</b>	<b>25</b>
5.1	<i>Allgemeine Konfiguration der Ausführungsplattform .....</i>	26
5.2	<i>Konfiguration und Erweiterung der Pflegeanwendung .....</i>	26
5.3	<i>Definition Knotentypen.....</i>	26
5.4	<i>Definition von BusinessContent Knoten zu einem Knotentyp .....</i>	31
5.5	<i>Definition Bedingungstypen .....</i>	31
<b>6.</b>	<b>Verwaltungstool zur Wiederverwendung von Knoten .....</b>	<b>32</b>
<b>7.</b>	<b>Ausführungsplattform Einbindungen und Aufruf .....</b>	<b>34</b>
7.1	<i>Integrationsmodul: Planungskontext .....</i>	34
7.2	<i>Integrationsmodul: Transformationskontext.....</i>	37
<b>8.</b>	<b>Berechtigungen.....</b>	<b>41</b>
<b>9.</b>	<b>Gruppen Pflege .....</b>	<b>42</b>
<b>10.</b>	<b>Transportwesen.....</b>	<b>42</b>
10.1	<i>Transportmanager.....</i>	42
10.1.1	<i>Ausgehende Transportitems.....</i>	43
10.1.2	<i>Eingehende Transportitems.....</i>	45
10.1.3	<i>Transportwesen als Backup .....</i>	46
10.2	<i>Altes Transportvorgehen / vollständiger Transport.....</i>	46

<b>11. Anhang</b> .....	<b>48</b>
11.1 Änderungshistorie.....	48
11.2 Verzeichnisse .....	48
11.2.1 Abbildungsverzeichnis .....	48
11.2.2 Tabellenverzeichnis.....	49

## 2. Einführung

AGIMENDO.calc ist eine Anwendung zur Pflege und Durchführung von Rechenregeln. Die Rechenregeln werden unabhängig vom Ausführungskontext definiert und anschließend in potentiell verschiedenen Ausführungskontexten angewendet. Auf diese Weise findet eine Trennung zwischen den Rechenregeln und der technischen Verwendung dieser statt, die es ermöglicht, dieselbe Rechenregel an unterschiedlichen Stellen wiederzuverwenden.

Aus dieser Trennung lässt sich direkt ableiten, dass die Anwendung AGIMENDO.calc grob aus zwei Teilen besteht. Zum einen gibt es eine Pflegeoberfläche (basierend auf WebDynpro4ABAP) und zum anderen eine Plattform zur Ausführung der Rechenregeln. Diese Ausführungsplattform wird derzeit mit zwei Integrationsmodulen ausgeliefert. Zum einen können Rechenregeln im Kontext der Integrierten Planung über einen Planungsfunktionstypen ausgeführt werden und zum anderen besteht die Möglichkeit innerhalb von Standard-Transformationen im BW die Rechenregeln auszuführen (Durch einen Aufruf innerhalb der Endroutine).

### Aufbau dieses Dokuments

Für ein Verständnis von AGIMENDO.calc wird im Folgenden zunächst die grundlegende Funktionsweise anhand eines einfachen Beispiels erklärt. Anschließend werden verschiedene Anwendungsszenarien und wie diese mit Hilfe von AGIMENDO.calc gelöst werden können, beschrieben.

Im folgenden Kapitel wird die Pflege der Rechenregeln über die Oberfläche erklärt. Eine Stärke von AGIMENDO.calc ist die Erweiterbarkeit und vielfältigen Möglichkeiten die „einfachen“ Berechnungen zu ergänzen. Im Kapitel 4 werden diese Möglichkeiten dargestellt und erklärt.

Kapitel 5 beinhaltet die Dokumentation der Integrationsmodule für AGIMENDO.calc. Dabei wird basierend auf dem allgemeinen Wissen über die Arbeitsweise dokumentiert, wie die Ausführungsplattform je Kontext konfiguriert werden muss.

### Voraussetzungen und Zielgruppe

Dieses Dokument richtet sich an versierte Fachanwender mit grundlegendem Verständnis für technische Abläufe. Grundsätzliches Verständnis von Kennzahlen- und Kontenmodell sowie der Arbeitsweise von Merkmalskombinationen zur Ermittlung von passenden Sätzen wird vorausgesetzt.

Für das vollständige Verständnis des Kapitels 4 und die Anwendbarkeit der dort beschriebenen Techniken wird Wissen über ABAP und ABAP-OO benötigt.



### Beispiel: Einfache Ergebnisrechnung

Das folgende Beispiel soll diese Grundprinzipien verdeutlichen.

Folgende Rechenregeln gelten:

- Ergebnis = Erträge – Aufwände
- Erträge = Umsätze + Zinserträge
- Aufwände = Lohnkosten + Rohstoffkosten + Herstellungskosten

Daraus ergibt sich der in Abbildung 2 dargestellte Rechenbaum.



Abbildung 2: Screenshot Pflege eines exemplarischen Rechenbaums einer Ergebnisrechnung

Als Basis für die Berechnung werden die in der folgenden Tabelle dargestellten Datensätze als Datentabelle verwendet.

Jahr	Gesellschaft	Kennzahltyp	Wert
2010	1000	UMSAETZE	10.000,-
2010	1000	ZINSERTRAEGE	700,-
2010	1000	LOHNKOSTEN	4.000,-
2010	1000	ROHSTOFFKOSTEN	2.000,-
2010	1000	HERSTELLKOSTEN	3.000,-

Tabelle 1: Datentabelle für Beispiel Ergebnisrechnung

Die Erwartungshaltung ist nun, dass nach der Berechnung drei weitere Datensätze erzeugt werden: „Erträge“, „Aufwände“ und „Ergebnis“. Um dies zu erreichen wird der Rechenbaum „durchgerechnet“. Am folgenden Beispiel soll verdeutlicht werden, wie dieses Durchrechnen des Baumes stattfindet. Der Rechenbaum wird rekursiv durchlaufen, die Werte der Blätter im Baum gelesen, beim wieder nach „oben laufen“ berechnet und in die Datentabelle geschrieben. Der oben dargestellte Rechenbaum wird in den folgenden Schritten durchlaufen:

1. Knoten Ergebnis (Start am Wurzelknoten, Durchlauf „nach unten“)
2. Knoten Erträge
3. Knoten Umsätze -> keine Kinder -> lese Wert 10.000,- // liefere diesen Wert zurück
4. Knoten Erträge -> merke 10.000,-
5. Knoten Zinserträge -> keine Kinder -> lese Wert 700,- // liefere diesen Wert zurück

6. Knoten Erträge -> verrechne 10.000,- und 700,- -> ergibt: 10.700 -> keine weiteren Kinder -> schreibe ERTRAEGE: 10.700,- // liefere diesen Wert zurück
7. Knoten Ergebnis -> merke 10.700,-
8. Knoten Aufwände
9. Knoten Lohnkosten -> keine Kinder -> lese Wert 4.000,- // liefere diesen Wert zurück
10. Knoten Aufwände -> merke 4.000,-
11. Knoten Rohstoffkosten -> keine Kinder -> lese Wert 2.000,- // liefere diesen Wert zurück
12. Knoten Aufwände -> verrechne 4.000,- und 2.000,- -> ergibt: 6.000,- -> merke 6.000
13. Knoten Herstellungskosten -> keine Kinder -> lese Wert 3.000,- // liefere diesen Wert zurück
14. Knoten Aufwände -> verrechne 6.000,- und 3.000,- -> ergibt 9.000,- -> keine weiteren Kinder -> schreibe AUFWAENDE: 9.000,- // liefere diesen Wert zurück
15. Knoten Ergebnis -> verrechne 10.700,- mit 9.000,- (hier ist ein „-“-Operator gepflegt) -> ergibt 1.700,- -> keine weiteren Kinder -> schreibe ERGEBNIS: 1.700,- // liefere diesen Wert zurück
16. Knoten Ergebnis des Wurzelknoten ist 1.700,- -> hier endet die Berechnung

Die Datentabelle enthält dann anschließend die Einträge, die in der Tabelle 2 dargestellt sind.

Jahr	Gesellschaft	Kennzahltyp	Wert
2010	1000	UMSAETZE	10.000,-
2010	1000	ZINSERTRAEGE	700,-
2010	1000	LOHNKOSTEN	4.000,-
2010	1000	ROHSTOFFKOSTEN	2.000,-
2010	1000	HERSTELLKOSTEN	3.000,-
2010	1000	ERTRAEGE	10.700,-
2010	1000	AUFWAENDE	9.000,-
2010	1000	ERGEBNIS	1.700,-

Tabelle 2: Ergebnis Datentabelle nach Ergebnisrechnung

Aus diesem Beispiel ergeben sich die folgenden Fragen, die in den nächsten Abschnitten geklärt werden:

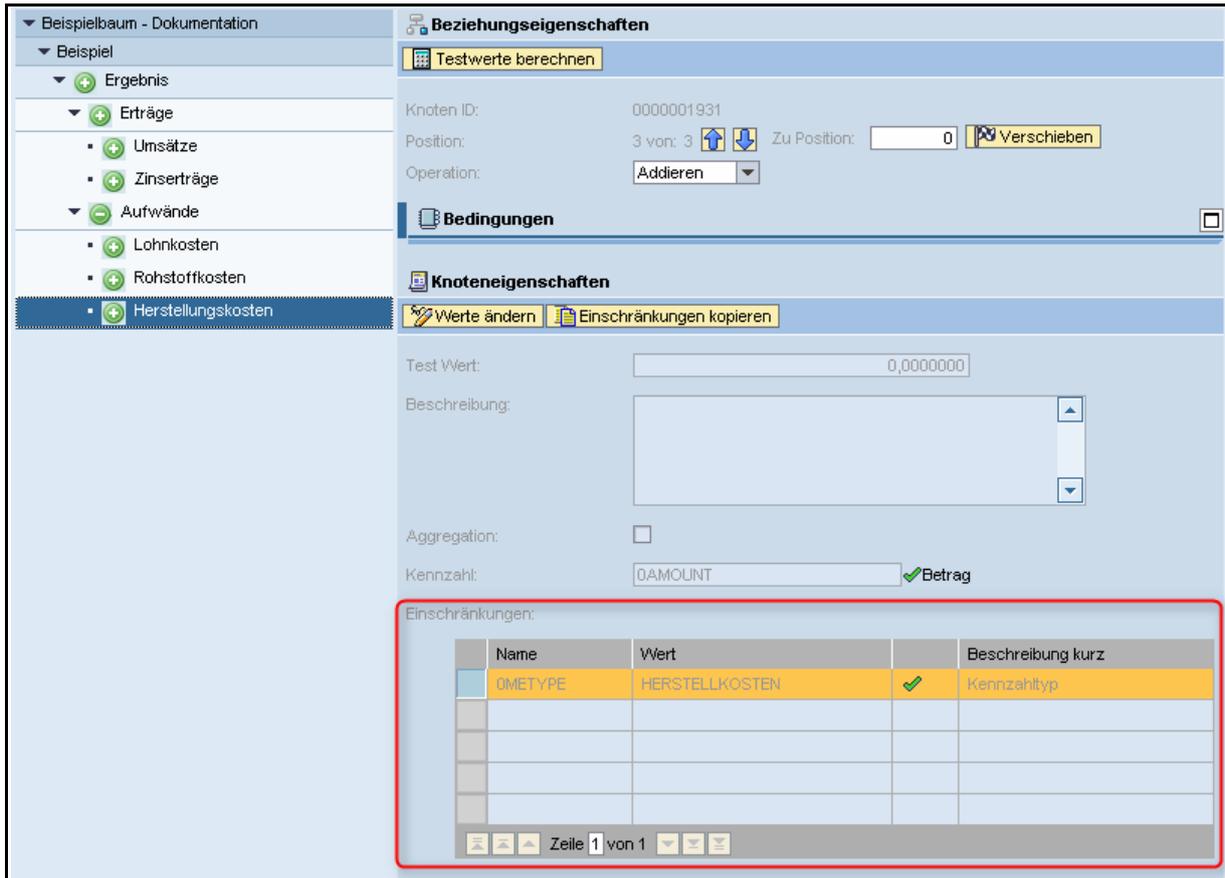
- Wie findet AGIMENDO.calc die „richtigen“ Sätze beim Lesen in der Datentabelle?
- Woher weiß AGIMENDO.calc auf welche Merkmalskombination ein Wert geschrieben werden soll?

Diese beiden Fragestellungen sind sehr eng miteinander verknüpft. Der Rechenbaum identifiziert sowohl beim Lesen aus als auch beim Schreiben in die Datentabelle den Zieldatensatz über Merkmalskombinationen. Im Kontext des Rechenkerns sprechen wir hierbei von einer Menge von **Einschränkungen**, die den Zieldatensatz (im Kontext der Datentabelle) eindeutig beschreiben.

**Hinweis:** Für das Verständnis der Arbeitsweise von AGIMENDO.calc bzw. zur Analyse von Problemen beim Einsatz ist dieser Punkt sehr wichtig. Wenn beim Lesen der Datensatz anhand der Einschränkungen nicht eindeutig bestimmt werden kann, wird zufällig der „erste passende“ zurückgeliefert. Wenn beim Schreiben der Datensatz nicht eindeutig identifiziert werden kann, wird je nach Situation entweder ein neuer Satz angelegt oder ebenfalls ein „zufällig passender“ Satz genutzt. Um erfolgreich mit AGIMENDO.calc zu arbeiten, ist es also wichtig, dass die Einschränkungen eindeutig sind. Mehr dazu im folgenden Kapitel.

### Einschränkungen

Im oben beschriebenen Beispiel „Einfache Ergebnisrechnung“ wurde der Rechenbaum vorgestellt, nicht aber die Definition der einzelnen Knoten. In der Pflegeanwendung von AGIMENDO.calc können zu allen lesenden und schreibenden Knoten Merkmalskombinationen als **explizite Einschränkung** direkt am Knoten gepflegt werden. In der Abbildung 3 ist zu sehen, dass für den lesenden Knoten „Herstellungskosten“ ein Kennzahltyp „HERSTELLKOSTEN“ im Bereich der expliziten Einschränkungen gepflegt ist.



The screenshot shows the configuration for the 'Herstellungskosten' node. The 'Einschränkungen' table is highlighted with a red border and contains the following data:

Name	Wert		Beschreibung kurz
OMETYPE	HERSTELLKOSTEN	✓	Kennzahltyp

Abbildung 3: Definition Einschränkungen Herstellungskosten

Aufbauend auf dem vorangegangenen Beispiel und der in Tabelle 1 dargestellten Datentabelle des Rechenbaumes ist diese Einschränkung eindeutig. Es kann also genau ein Datensatz in der Datentabelle ermittelt werden, der den explizit gepflegten Einschränkungen je Knoten entspricht und aus dem die definierte Kennzahl gelesen werden kann (im Beispiel die Kennzahl „OAMOUNT“).

Wenn die Datentabelle mehr Daten enthält, also beispielsweise Daten für mehr als eine Gesellschaft, reicht diese eine Einschränkung (OMETYPE == HERSTELLKOSTEN) für die Eindeutigkeit natürlich nicht mehr aus. (Die Tabelle 3 enthält zur Verdeutlichung Daten für zwei Gesellschaften).

Jahr	Gesellschaft	Kennzahltyp	Wert
2010	1000	UMSAETZE	10.000,-
2010	1000	ZINSERTRAEGE	700,-
2010	1000	LOHNKOSTEN	4.000,-
2010	1000	ROHSTOFFKOSTEN	2.000,-

2010	1000	HERSTELLKOSTEN	3.000,-
2010	2000	UMSAETZE	20.000,-
2010	2000	ZINSERTRAEGE	1.700,-
2010	2000	LOHNKOSTEN	7.000,-
2010	2000	ROHSTOFFKOSTEN	4.000,-
2010	2000	HERSTELLKOSTEN	6.000,-

Tabelle 3: Datentabelle mit Beispieldaten für mehr als eine Gesellschaft

Es gibt unterschiedliche Möglichkeiten darauf zu reagieren. Die einfachste wäre, im Rechenbaum für jede Gesellschaft einen eigenen Rechenbaum anzulegen. Das führt allerdings zu sehr hohem Pflegeaufwand und vielen redundant gepflegten Rechenregeln und ist damit im Allgemeinen keine Lösung für das Problem. Im Folgenden werden Möglichkeiten beschrieben, wie diesem Umstand begegnet werden kann.

#### Einschränkung durch Blockbildung

Eine Möglichkeit die Einschränkung aus dem Beispiel (OMETYPE = HERSTELLKOSTEN) eindeutig zu machen besteht nun darin, die Datentabelle zu verkleinern. Dies bedeutet, dass AGIMENDO.calc die Berechnungen einmal auf der ersten Hälfte der Tabelle durchführt (für die Gesellschaft 1000) und einmal auf der zweiten Hälfte (für die Gesellschaft 2000). Dies erfolgt beispielsweise, wenn AGIMENDO.calc innerhalb von Planungsfunktionen mit Blockbildung verwendet wird (Im Beispiel wäre die Gesellschaft in der Blockung, siehe hierzu Kapitel 7.1).

Eine zweite Möglichkeit besteht darin, die Ausführungsplattform von AGIMENDO.calc beim Aufruf zu parametrisieren und eine **globale Einschränkung** festzulegen. Damit werden alle expliziten Einschränkungen, die direkt in der Pflegeoberfläche definiert wurden, um eine Einschränkung erweitert. Über diesen Weg kann ein Rechenbaum mehrfach (im Beispiel für unterschiedliche Gesellschaften) verwendet werden, ohne die Regeldefinition anzupassen.

**Hinweis:** Die Art und Weise, wie die globalen Einschränkungen definiert werden können, ist abhängig von dem gewählten Integrationsmodul. In Kapitel 6 werden diese Konfigurationen beschrieben.

**Hinweis:** Wenn in den globalen Einschränkungen und in den an einem Knoten gepflegten Einschränkungen ein Merkmal doppelt auftaucht, kann AGIMENDO.calc nicht entscheiden, welche gültig ist und bricht die Berechnung mit einem Fehler ab. Der Fehler der hierbei erfolgt, sieht im Planungskontext wie in der folgenden Abbildung 4 dargestellt aus.

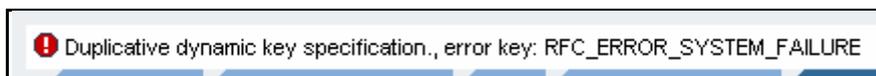
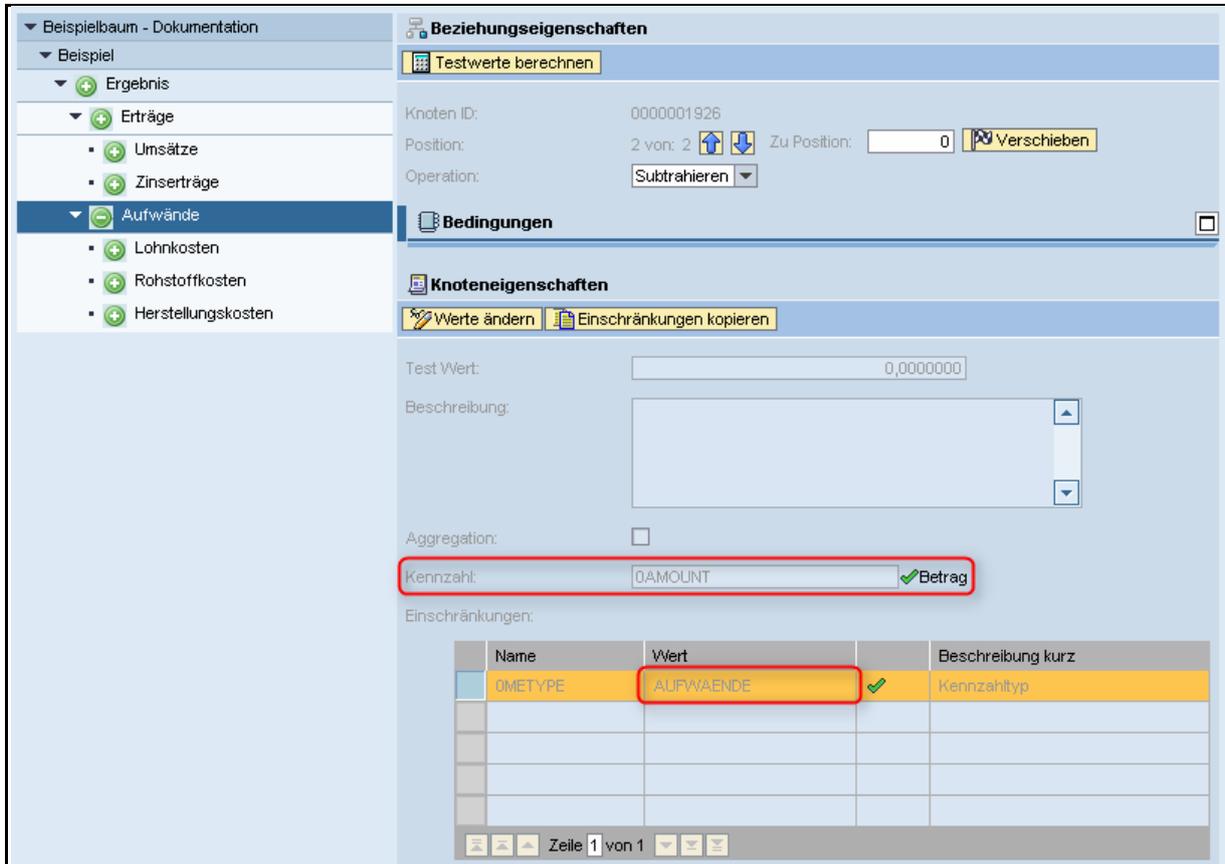


Abbildung 4: Screenshot - Fehlermeldung bei Überschneidung der Einschränkungen

#### Allgemeines Verhalten beim Schreiben von Werten

AGIMENDO.calc schreibt einen an einem Knoten errechneten Wert wieder in die Datentabelle (Siehe bspw. Schritt 6 des Berechnungsflusses im oben beschriebenen Beispiel). Dabei wird zunächst über denselben Mechanismus wie beim Lesen der Zieldatensatz über die Einschränkungen aus der Datentabelle ermittelt. Anschließend wird der im Berechnungsfluss errechnete Wert auf die in der Pflegeoberfläche definierte Kennzahl geschrieben. In der Abbildung 5 ist die Konfiguration des Knotens „Aufwände“ aus dem vorangegangenen Beispiel dargestellt.

**Hinweis:** Lesende und Schreibende Knoten werden identisch konfiguriert. Ein Knoten der keine untergeordneten Kind-Knoten besitzt wird als Blatt und somit als nur zu lesender Knoten interpretiert. Alle Knoten die untergeordnete Kind-Knoten besitzen werden beschrieben. In beiden Fällen steht im Feld „Kennzahl“ die zu lesende oder schreibende Kennzahl.



The screenshot shows the 'Beziehungseigenschaften' (Relationship Properties) window for a node. The left sidebar shows a tree structure with 'Aufwände' selected. The main window has several sections:

- Beziehungseigenschaften:** Includes 'Testwerte berechnen', 'Knoten ID: 0000001926', 'Position: 2 von: 2', 'Operation: Subtrahieren', and a 'Verschieben' button.
- Bedingungen:** A section with a checkbox.
- Knoteneigenschaften:** Includes 'Werte ändern' and 'Einschränkungen kopieren' buttons.
- Test Wert:** A text input field containing '0,0000000'.
- Beschreibung:** A large text area.
- Aggregation:** A checkbox.
- Kennzahl:** A dropdown menu showing '0AMOUNT' with a green checkmark and the label 'Betrag'.
- Einschränkungen:** A table with columns 'Name', 'Wert', and 'Beschreibung kurz'. The first row is highlighted in yellow and contains '0METYPE', 'AUFWAENDE', and 'Kennzahltyp'.

Abbildung 5: Definition Schreibender Knoten "Aufwände"

Beim Schreiben von Daten zurück in die Datentabelle sind die folgenden Fälle zu unterscheiden:

- Im einfacheren Fall wird ein Datensatz gefunden, der zu den Einschränkungen des schreibenden Knotens passt. In diesem Fall wird der vorhandene Satz geändert. Die Änderung betrifft immer nur die angegebene Kennzahl, im Beispiel von Abbildung 5 wird also der neue Wert auf die Kennzahl 0AMOUNT geschrieben. Dies wäre zum Beispiel der Fall, wenn auf der Ergebnistabelle des Beispiels (Tabelle 2) AGIMENDO.calc erneut, beispielsweise nach einer Korrektur der Lohnkosten, ausgeführt werden würde.
- Der etwas komplexere Fall tritt ein, wenn beim Schreiben kein passender Zieldatensatz gefunden wird. In diesem Fall muss der Zieldatensatz neu angelegt werden. Dazu müssen alle Merkmale des neu anzulegenden Satzes mit den „richtigen“ Werten belegt werden. Im Folgenden wird beschrieben, wie diese „richtigen Werte“ ermittelt werden.

#### Anlegen eines neuen Datensatzes

Wenn ein Datensatz neu angelegt wird, werden alle Merkmale mit den Einschränkungen vorbelegt, die zum Zeitpunkt der Anlage gültig sind. Hierzu zählen

- explizite Einschränkungen, also die Einschränkungen, die am Knoten gepflegt sind

- die globalen Einschränkungen, die für den aktuellen Durchlauf gültig sind,
- sowie die Merkmalskombinationen, die für die Initialwerte konfiguriert wurden.

Die globalen Einschränkungen und die Initialwerte werden über den Aufruf der Ausführungsplattform von AGIMENDO.calc definiert. Dies ist in Kapitel 6 genauer beschrieben.

**Hinweis:** Wichtig ist hierbei, dass sich die ersten beiden Gruppen von Einschränkungen nicht überschneiden, da ansonsten die Ausführungsplattform nicht korrekt zuordnen kann, mit welchem Wert der neue Satz angelegt werden soll und mit einem Fehler abbricht. Der Fehler hierzu ist in Abbildung 4 dargestellt. Die Initialwerte können von den anderen beiden Gruppen der Einschränkungen überschrieben werden. Dies sollte aus Übersichtsgründen allerdings nach Möglichkeit vermieden werden.

Die Arbeitsweise von AGIMENDO.calc kann man auf die vorangegangenen Grundsätze des Lesens, Verrechnens und das anschließende Schreiben vereinfachen. Um in komplexen Berechnungsszenarien einfach und schnell zum Ziel zu kommen, sind Spezialisierungen sowie Hilfestellungen zu diesem sehr einfachen und sehr generischen Modell notwendig. Diese werden in den folgenden Kapiteln beschrieben.

## 2.2 Knotentypen

Im vorangegangenen Kapitel wurde das grundsätzliche Verhalten von AGIMENDO.calc beschrieben. Dabei wurde lediglich von Knoten im Baum gesprochen, die lesend oder schreibend auf die Datentabelle zugreifen. AGIMENDO.calc kennt neben diesen Wert-Knoten zum Zugriff auf die Datentabelle die drei folgenden weiteren Knotentypen.

- Festwertknoten
- Temporäre Knoten
- User-Exit-Knoten / BusinessContent Knoten

**Festwertknoten** ermöglichen es, einen festen Wert zur Definitionszeit des Rechenbaumes abzulegen („Konstante“). Beispielsweise den Mehrwertsteuersatz. Festwertknoten innerhalb des Baumes (also mit eigenen Kindknoten) verhalten sich wie temporäre Knoten. **Temporäre Knoten** haben die Eigenschaft, dass sie zwar einen berechneten Wert nach weiter oben im Baum „weiterleiten“, aber kein Ergebnis zurück in die Datentabelle schreiben. Ein Temporärer Knoten als Blatt im Baum liefert den Wert 0 zurück.

**Hinweis:** Bei der Neuanlage eines Baumes wird als Wurzel ein temporärer Knoten generiert. Die Werte die bei der Berechnung in diesen Knoten weitergeleitet werden, werden nicht weiter behandelt.

Im Unterschied zu diesen beiden Knotentypen bringt der dritte Knotentyp, der **User-Exit-Knoten**, keine eigene Logik mit, sondern erlaubt es, komplett benutzerdefinierte Berechnungslogiken im Baum zu hinterlegen. Über den User-Exit-Knoten kann das Verhalten, beim Lesen bzw. Schreiben von Werten, beeinflusst oder ganz neugestaltet werden. Über die Definition eines eigenen Pflegeformulars (über eine BusinessContent Definition) für einen UserExit Knoten kann die Pflege vereinfacht werden. Mehr dazu finden Sie im Kapitel 4.

## 2.3 Bedingungen im Baum

Bedingungen im Berechnungsbaum ermöglichen es, einzelne Teile des Rechenbaumes in die Berechnung ein- oder davon auszuschließen. Die Pflege der Bedingung findet am Knoten statt. Wenn die Bedingung zur Laufzeit auf „FALSCH“ ausgewertet wird, wird die Berechnung mit dem nächsten Knoten auf derselben Ebene fortgesetzt. Es findet weder eine Berechnung der Unterknoten statt, noch wird ein Wert (bei einem Knoten innerhalb des (Unter-)Baumes) in die Datentabelle geschrieben.

Über Bedingungen können zum Beispiel innerhalb der Rechenregeln Ausnahmen bzw. Sonderfälle modelliert werden.

Ähnlich der Logik zum Erstellen von User-Exit-Knoten können auch selbstdefinierte Bedingungstypen definiert werden. Beispielsweise könnte ein Bedingungstyp implementiert werden, der in einem Infoobjekt zur aktuellen Gesellschaft überprüft, ob diese dort als berichtend gekennzeichnet ist.

Details zur Implementierung von eigenen Bedingungen sind in Kapitel 4 beschrieben.

### Pflege der Bedingungen

In der Pflegeanwendung von AGIMENDO.calc können diese am Knoten im Bereich Bedingungen gepflegt werden. Jede Zeile entspricht einem Bedingungsteil. Jeder Bedingungsteil hat einen Typ (bspw. Gleichheitsprüfung) und kann mit den anderen Bedingungen in der Liste mit „UND“ oder „ODER“ verknüpft werden.



Abbildung 6: Screenshot - Pflege Bedingungen am Knoten

Für eine Bedingung können bis zu fünf Werte gepflegt werden. Der Operator (1) ist UND (A), UND NICHT (AN) oder ODER (O), ODER NICHT (ON). Die Spalte Merkmal beinhaltet den Namen des Merkmals (2), das im Bedingungstyp verwendet wird, im Beispiel wird das Merkmal Buchungskreis (0COMP\_CODE) überprüft ob er einen bestimmten Wert hat oder in einem Intervall liegt. Dabei überprüft der angegebene Bedingungstyp (3) ob der Wert gleich Wert 1 (4) ist oder zwischen Wert 1 und Wert 2 (5) liegt.

**Hinweis:** In der v1.3 ist keine Klammerung der Bedingungsteile möglich, allerdings ist die UND/ODER-Implementierung „logisch korrekt“. UND bindet stärker als ODER.

**Beispiel:** für die Bindung von UND und ODER – Die Bedingung

0COM\_CODE = 1000 ODER 0COMP\_CODE = 2000 UND 0FISCYEAR = 2010

ist genau dann gültig, wenn der Buchungskreis 1000 vorliegt oder der Buchungskreis 2000 vorliegt und gleichzeitig das Geschäftsjahr 2010 aktuell ist. Die Bedingung ist also auch für Buchungskreis 1000 und Geschäftsjahr 2009 gültig. Die oben genannte Bedingung ist logisch identisch mit

0COMP\_CODE = 2000 UND 0FISCYEAR = 2010 ODER 0COM\_CODE = 1000

### Ausgelieferte Bedingungstypen

In v2.1.1 sind vier Bedingungstypen enthalten.

- Eine Gleichheitsüberprüfung („x = v1“)
- Eine Ungleichheitsüberprüfung („x <> v1“)
- Eine nicht „nicht-initial“ Überprüfung („x != 0“)
- Eine Zwischenüberprüfung („v1 <= x <=v2“)
- Eine Größer Überprüfung („x > v1“) (KYF.)
- Eine Kleiner Überprüfung („x < v2“) (KYF.)
- Eine Gleichheitsüberprüfung („x = v1“) (KYF.)
- Eine UserExit Überprüfung („UE-Cond“) (KYF.)

Bedingungstypen ohne die Bezeichnung (KYF.) prüfen „gegen“ die globalen Einschränkungen bzw. Initialwerte des aktuellen Berechnungsdurchlaufs. Wenn ein Rechenbaum also für eine Menge von Gesellschaften durchgerechnet wird, werden diese Gesellschaften in den Einschränkungen geführt (pro Baumdurchlauf genau eine). Diese Gesellschaft kann dann für die Bedingungsprüfung herangezogen werden.

Bedingungstypen mit der Bezeichnung (KYF.) können gegen alle Spalten der Tabelle prüfen. Zusätzlich besteht auch die Möglichkeit gegen einen gespeicherten Wert aus dem Speicher zu prüfen. Dazu muss vor dem Variablennamen ein „\$“ eingegeben werden. Sowohl in das Merkmal Feld als auch die Wert Felder können Variablennamen eingegeben werden.

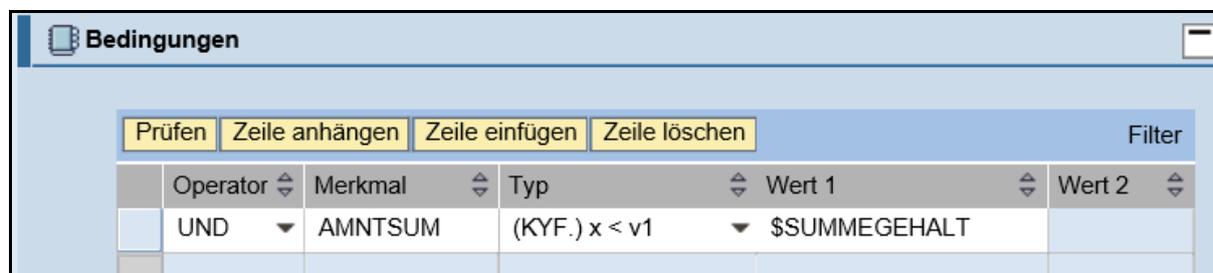


Abbildung 7 Screenshot - (KYF.) Bedingung

Bei der UserExit Überprüfung muss man eine Klasse im Feld "Merkmal" eingeben. Diese Klasse muss von der Klasse /ISV/CL\_UE\_COND\_CLASS erben und die Methode "TEST" redefinieren. Der UE Klasse wird die komplette Zeile, in der sich der Durchlauf befindet, übergeben. Zusätzlich wird noch Wert 1 und Wert 2 mit übergeben, dabei können auch Variablen mit "\$" referenziert werden.

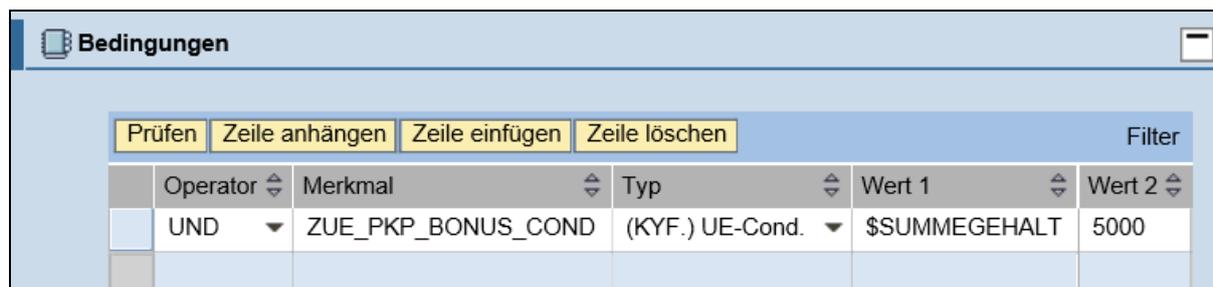


Abbildung 8 Screenshot - UserExit Bedingung

**Hinweis:** Die „nicht initial“ Überprüfung führt ein „is initial“ im ABAP durch.

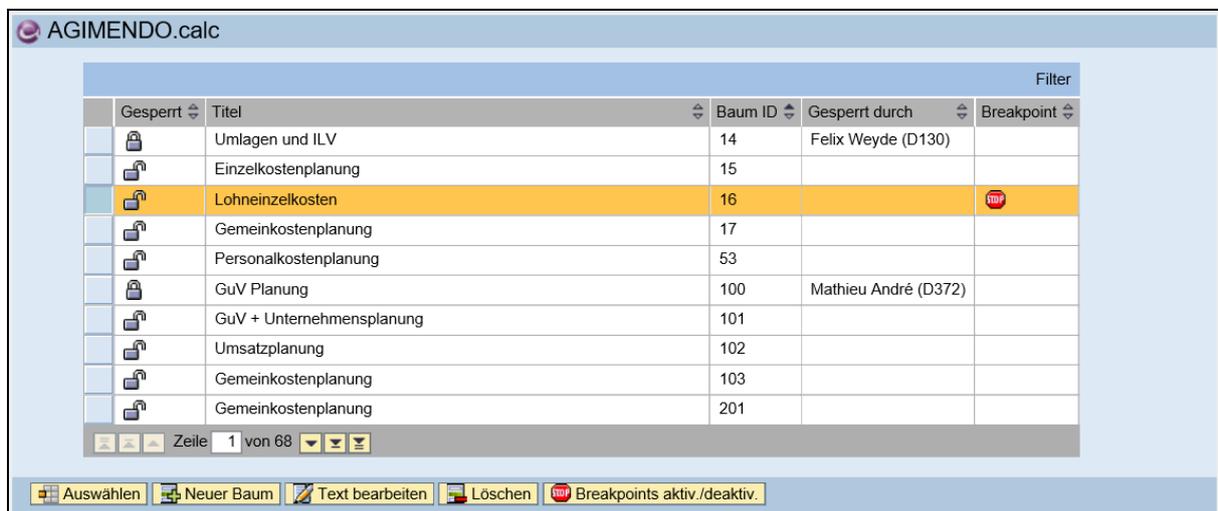
**Hinweis:** Die (Un-)Gleichheitsprüfung und die Initialprüfung arbeitet sowohl mit zahlenwertigen als auch mit zeichenbasierten Merkmalen, die Zwischenüberprüfung nur mit zahlenwertigen Merkmalen.

### 3. Pflege Rechenregeln

Die Pflege der Rechenregeln erfolgt über die WebDynpro-Anwendung /ISV/PCE\_TREE. Diese ist auf dem System unter dem Pfad [http://<server>:<port>/sap/bc/webdynpro/isv/pce\\_tree](http://<server>:<port>/sap/bc/webdynpro/isv/pce_tree) zu erreichen.

#### 3.1 Baumauswahl, Anlage und Pflege

Über den Startbildschirm der Anwendung ist es möglich einen Rechenbaum zur Pflege auszuwählen, den Namen eines vorhanden zu ändern, Breakpoints für einen Baum zu aktivieren, einen vorhandenen Rechenbaum zu löschen oder einen neuen Rechenbaum anzulegen.



Gesperrt	Titel	Baum ID	Gesperrt durch	Breakpoint
	Umlagen und ILV	14	Felix Weyde (D130)	
	Einzelkostenplanung	15		
	Lohneinzelkosten	16		
	Gemeinkostenplanung	17		
	Personalkostenplanung	53		
	GuV Planung	100	Mathieu André (D372)	
	GuV + Unternehmensplanung	101		
	Umsatzplanung	102		
	Gemeinkostenplanung	103		
	Gemeinkostenplanung	201		

Abbildung 9: Screenshot – Baumauswahl

In der Abbildung 9 ist der Startbildschirm von AGIMENDO.calc dargestellt. Die Liste der Bäume ist im Standard anhand des Titels (in Anmeldesprache) sortiert. Mit Ausnahme der Bäume, die durch den angemeldeten Benutzer im Moment in Bearbeitung sind. Diese stehen am Anfang der Tabelle.

Die Sortierreihenfolge kann durch Anklicken der Spaltenköpfe beeinflusst werden. Über den Link „Filter“ rechts oben kann eine Filterleiste zur einfachen Suche eingeblendet werden. Um einen Ausgewählten Baum zu pflegen, kann über den Button „Auswählen“ zur Pflegeanwendung gesprungen werden. Die Pflege des Baumtitels kann über den Button „Text bearbeiten“ für den ausgewählten Baum geöffnet werden.

Bei der Neuanlage von Rechenbäumen (über den Button „Neuer Baum“) muss ein Baumtitel und der Titel des Wurzelknotens (der implizit mit angelegt wird) angegeben werden.

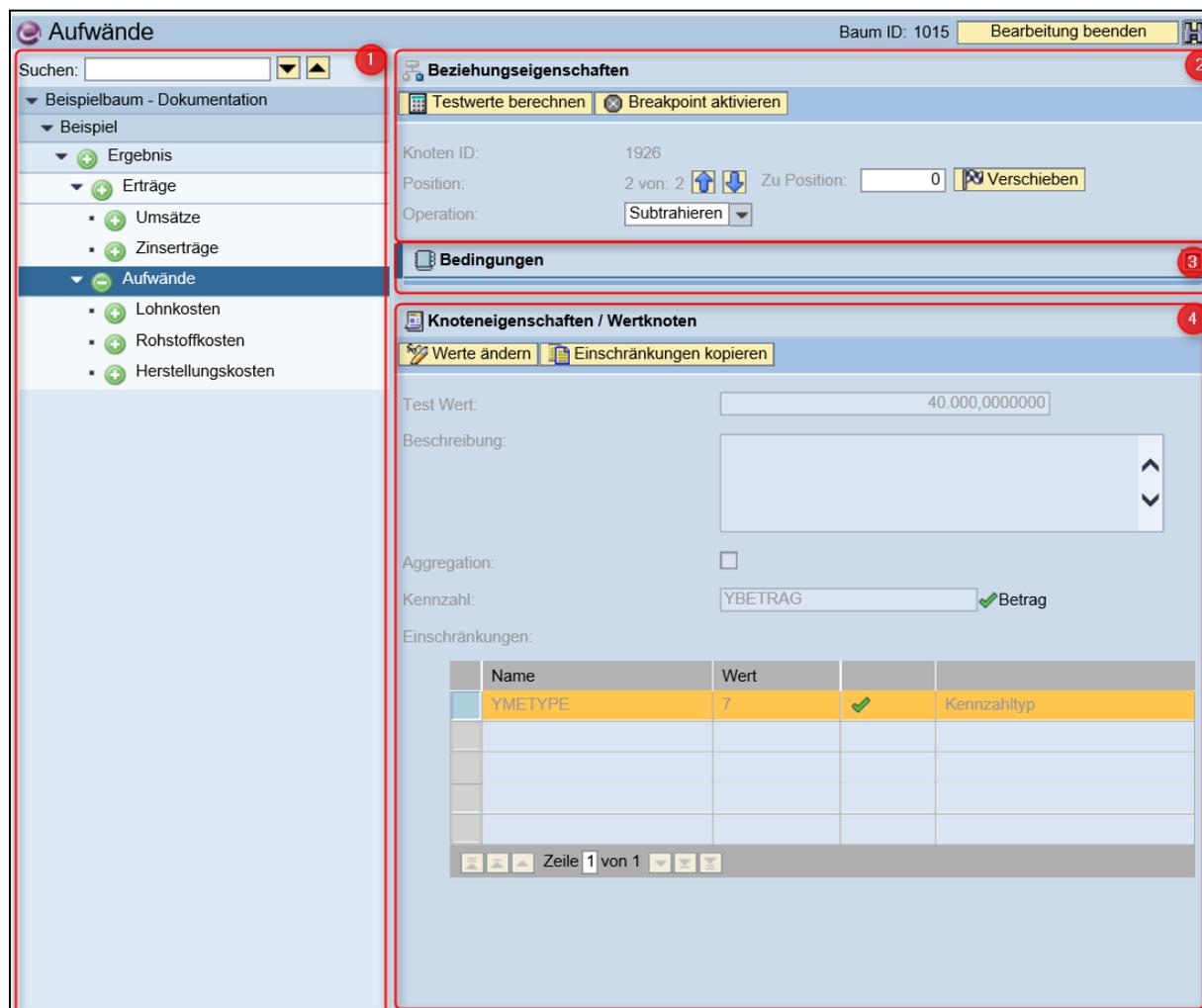
Das Löschen eines Rechenbaums ist nur für besonders berechtigte Benutzer möglich. Wenn der angemeldete Benutzer berechtigt ist, wird nach Rückfrage nach einem Klick auf „Löschen“ der ausgewählte Rechenbaum gelöscht. Die dafür benötigten Berechtigungsobjekte sind in Kapitel 0 beschrieben.

**Hinweis:** Beim Löschen wird lediglich die Baum-Definition gelöscht. Alle Knoten inkl. Struktur bleiben erhalten.

### 3.2 Rechenbaumpflege

Die Pflege des Rechenbaumes erfolgt entlang der Baumstruktur. Die Pflegeoberfläche ist in vier Bereiche gegliedert. Diese sind in Abbildung 10 dargestellt.

Im ersten Bereich (Baumpflege) ist der Baum dargestellt. Jeder Knoten repräsentiert einen Berechnungsschritt. Das Icon mit der Rechenoperation (in der Abbildung „+“ und „-“) signalisiert, wie das Ergebnis des Berechnungsschrittes in die übergeordnete Berechnung einfließt. Über diesen Bereich erfolgt die Navigation zwischen den Knoten, deren Eigenschaften dann in den Bereichen zwei bis vier dargestellt werden. Das Suchfeld kann durch einen Klick auf das sich im rechten oberen Rand befindliche Fernglas eingblendet werden. Mithilfe der Suchmaschine kann der Baum nach Knoten ID, Knoten Name, Beschreibung und Einschränkungen durchsucht werden.



The screenshot shows the 'Aufwände' (Expenses) section of the AGIMENDO.calc software. The interface is divided into four numbered red boxes:

- 1:** A search field and a tree view showing the hierarchy: 'Beispielbaum - Dokumentation' > 'Beispiel' > 'Ergebnis' > 'Erträge' > 'Umsätze', 'Zinserträge', 'Aufwände' > 'Lohnkosten', 'Rohstoffkosten', 'Herstellungskosten'.
- 2:** 'Beziehungseigenschaften' (Relationship Properties) panel for Knoten ID 1926. It shows 'Position: 2 von 2' and 'Operation: Subtrahieren'. Buttons include 'Testwerte berechnen' and 'Breakpoint aktivieren'.
- 3:** 'Bedingungen' (Conditions) panel, currently empty.
- 4:** 'Knoteneigenschaften / Wertknoten' (Node Properties / Value Nodes) panel. It shows 'Test Wert: 40.000,000000', 'Beschreibung:' (empty text area), 'Aggregation:' (checkbox), 'Kennzahl: YBETRAG' (checked), and 'Einschränkungen:' (restrictions table).

Name	Wert		
YMETYPE	7	✓	Kennzahltyp

Abbildung 10: Screenshot - Übersicht Pflegeoberfläche von AGIMENDO.calc

Im zweiten Bereich (Beziehungspflege) besteht die Möglichkeit die Beziehungseigenschaften vom aktuell gewählten Knoten zum übergeordneten Knoten zu pflegen.

Der dritte Bereich (Bedingungspflege) ist in der Abbildung 10 miniert. Hier werden die gepflegten Bedingungen für den ausgewählten Knoten angezeigt.

Der vierte Bereich (Eigenschaftspflege) enthält die Knoteneigenschaften. Dieser Bereich unterscheidet sich je Knotentyp. Aktuell gibt es die vier Knotentypen Wertknoten, Temporäre Knoten, Festwertknoten und User-Exit-Knoten. Die Funktion dieser Knotentypen sind in Kapitel 2.2 beschrieben.

Im Folgenden werden die vier Bereiche mit Ihren Möglichkeiten vorgestellt.

### 3.2.1 Baumpflege

In diesem Bereich werden die Knoten des Rechenbaumes in ihrer hierarchischen Struktur dargestellt. Diese Struktur dient vor allem zur Navigation im Baum. Über das Kontextmenü dargestellt in der Abbildung 11, kann der Baum gepflegt werden. Dabei stehen die folgenden Operationen zur Verfügung:

- Neuer Knoten – Über diese Funktion kann ein neuer Knoten eingefügt werden. Dabei wird, nach der Auswahl des Knotentyps, der neue Knoten angelegt und unterhalb des aktuell ausgewählten Knoten eingefügt.
- Existierenden Knoten einfügen – Im Unterschied zum Einfügen eines neuen Knoten, wird über diese Funktion eine Auswahlliste angeboten, für das Einfügen eines beliebigen existierenden Knoten. Dabei werden alle Kind-Elemente unterhalb des eingefügten Knotens mit eingefügt. Es wird also eine Referenz auf den eingefügten Knoten erzeugt. Eine Änderung an den Eigenschaften wirkt sich damit auch auf die anderen Verwendungen dieses Knotens aus.
- Knoten klonen – Beim Klonen eines Knotens wird der ausgewählte Knoten inklusive aller untergeordneten Knoten kopiert. Es findet also eine „tiefe“ Kopie statt. Vor dem Klonen kann ausgewählt werden, ob der erzeugte neue Knoten unterhalb des ausgewählten Knoten eingefügt werden soll.
- Knoten entfernen – Beim Entfernen eines Knotens wird der aktuell gewählte Knoten aus dem Baum entfernt. Die Knotendefinition wird davon aber nicht berührt. Der Knoten existiert weiterhin.
- Knoten löschen – Anders als beim Entfernen wird beim Löschen der Knoten gelöscht. Allerdings wird zunächst ein Dialog geöffnet, in dem der zu löschende Knoten ermittelt werden kann. Beim Löschen wird lediglich der Knoten gelöscht, nicht aber alle ihm untergeordneten Knoten.
- Knoten deaktivieren – Beim Deaktivieren eines Knotens, wird der Knoten beim Berechnungsdurchlauf nicht beachtet. Falls der deaktivierte Knoten Kinderknoten enthält, werden die ebenfalls nicht mit einbezogen. Ein deaktivierter Knoten wird nicht aus dem Baum entfernt, sondern ausgegraut dargestellt. Somit kann dieser jederzeit wieder aktiviert werden.
- Verwendungsnachweis – Beim Verwendungsnachweis wird eine neue Applikation im neuen Reiter geöffnet, die alle Informationen über den ausgewählten Knoten kompakt auf einer Seite darstellt. Darunter auch alle Vorkommen des Knotens.

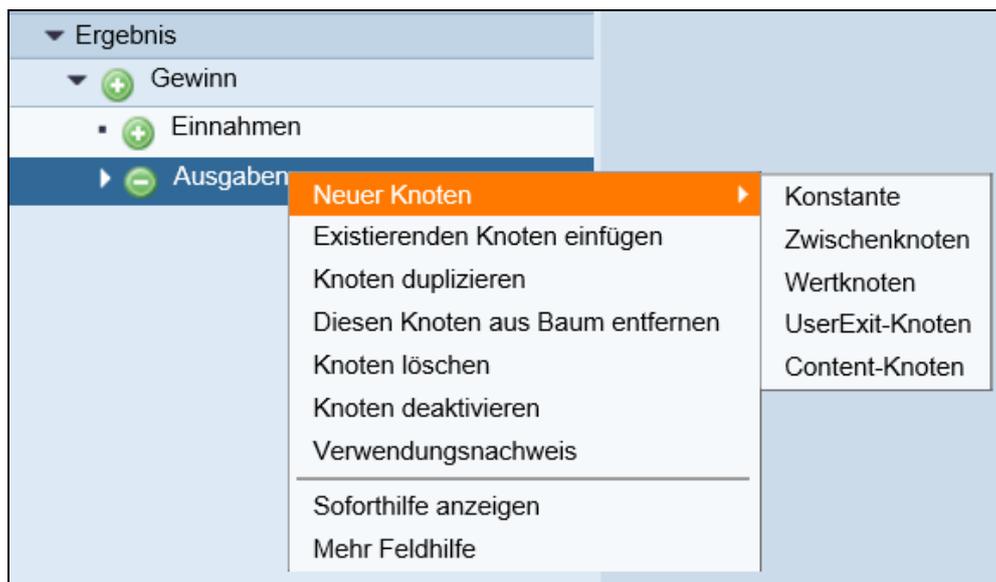


Abbildung 11: Screenshot - Kontextmenü Baumpflege

Ab einer AGIMENDO.calc Version 2.0.3 und BW Version 7.3 besteht die Möglichkeit, Knoten in der Baumpflege durch Drag and Drop zu verschieben. Dazu selektiert man den zu verschiebenden Knoten und zieht ihn an die gewünschte Position. Es besteht die Möglichkeit, einen Knoten in einen anderen zu verschieben und zusätzlich an die gewünschte Position.

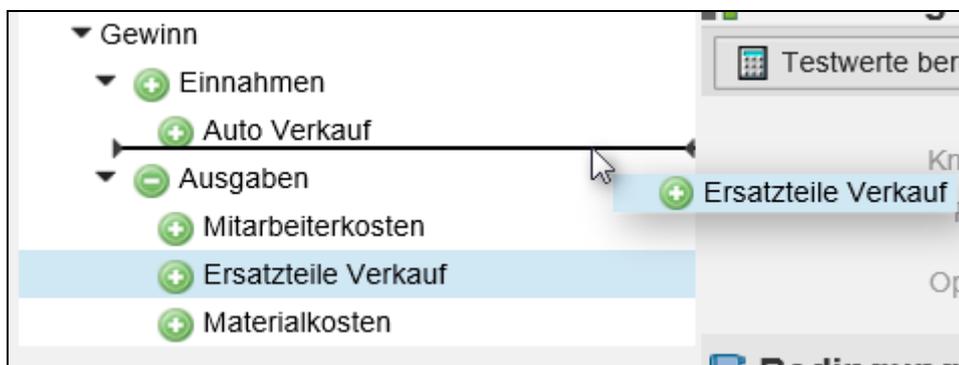


Abbildung 12: Screenshot - Drag and Drop

### 3.2.2 Beziehungspflege

Im Bereich der Beziehungseigenschaften werden die Beziehungen des ausgewählten Knoten zum übergeordneten Knoten gepflegt. Im Kontext der Abbildung 10 wird also die Beziehung zwischen den Knoten „Ergebnis“ und „Aufwände“ bearbeitet. Die Eigenschaften sind nochmals in Abbildung 13 dargestellt. (Die nachfolgend in Klammern geschriebenen Nummern beziehen sich auf die Nummern in Abbildung 13)

Die technische ID des Knotens der ausgewählt ist wird bei (1) dargestellt. Diese ID kann beispielsweise zur eindeutigen Identifikation von Knoten in den Suchdialogen verwendet werden.

Die Position (2) gibt an, an welcher Stelle der Kinderliste des übergeordneten Knoten sich der aktuelle Knoten befindet. Über die Pfeile kann der aktuell ausgewählte Knoten nach „oben“ und

„unten“ verschoben werden. Alternativ ist es möglich, den Knoten durch Eingabe einer neuen Position direkt an die gewünschte Stelle zu verschieben. (4)

Die Operation (3) gibt an, mit welcher Operation der aktuelle Knoten mit dem Ergebnis des vorangegangenen Knoten auf selber Ebene verrechnet wird. Im Beispiel von Abbildung 10 ist das für den ausgewählten Knoten „Aufwände“ die Operation „Subtrahieren“. Das bedeutet, dass von den „Erträgen“ die „Aufwände“ abgezogen werden.

Die Berechnung von Testwerten (5) bzw. deren Interpretation ist im Kapitel 3.2.5 beschrieben.

Es besteht die Möglichkeit einen Breakpoint (6) für einen Knoten zu aktivieren. Der Breakpoint wird beim Durchlauf des Baumes im Code angesprungen. Dazu müssen zusätzlich Breakpoints für den Baum aktiviert sein und der Baum muss vom User gesperrt werden.

Die Breakpoints befinden sich im Code vorm schreiben und lesen eines Werts. Falls Bedingungen vorhanden sind, hält das Programm auch vor Auswertung der Bedingung.

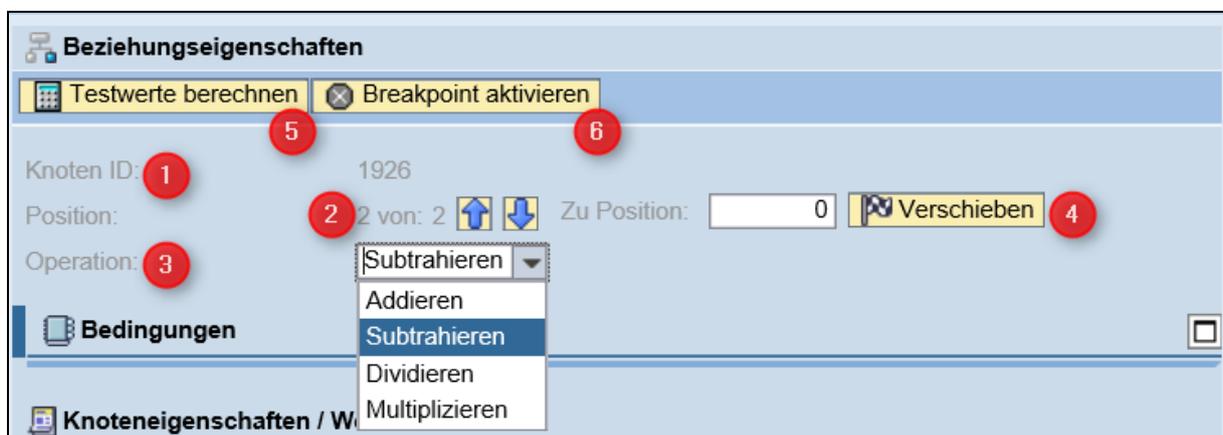


Abbildung 13: Screenshot - Pflege der Knotenbeziehungen

**Hinweis:** Die Berechnung in AGIMENDO.calc erfolgt linear entlang des Baumes. Das bedeutet, dass hier **nicht** „Punkt vor Strich“ gerechnet wird. Um eine „Punkt vor Strich“ Rechnung umzusetzen, muss ein temporärer Knoten als Zwischenknoten genutzt werden. Im nachfolgenden Beispiel ist dies dargestellt.

#### Beispiel: Nutzung von temporären Knoten zur Klammerung

Die Zinserträge sollen im Beispiel aus Abbildung 2 nicht voll, sondern nur zu 50% in die Berechnung einfließen. Dazu wird ein Festwertknoten („50% Anteil“) mit der Konstante 0,5 definiert. In der Abbildung 14 ist ein erster Berechnungsbaum ausschnittsweise dargestellt. Unter Beachtung des vorangegangenen Hinweises führt diese Berechnungsvorschrift allerdings im Ergebnis zu folgender Berechnung:

$$\text{Ertrag} = (\text{Umsätze} + \text{Zinserträge}) * 50\%$$



Abbildung 14: Screenshot - Ausschnitt Berechnung Erträge ohne Zwischenknoten

Erst durch das Einfügen eines Temporären Knotens („Anteiliger Zinsertrag“) dessen Ergebnis also nur in die Berechnung des übergeordneten Knoten einfließt, aber nicht gespeichert wird, dargestellt in Abbildung 15, führt zu der gewünschten Berechnung:

$$\text{Erträge} = \text{Umsätze} + (\text{Zinsertrag} * 50\%)$$



Abbildung 15: Screenshot - Ausschnitt Berechnung mit korrekter Klammerung

### 3.2.3 Bedingungspflege

Die Pflege der Bedingungen ist möglich, wenn der aktuelle Knoten im Änderungsmodus ist, also nachdem unter Knoteneigenschaften der Button „Werte ändern“ gedrückt wurde. Den Bereich der Bedingungspflege kann über das Tray-Symbol auf der rechten Seite ein- und ausgeblendet werden. Weitere Informationen zum Aufbau und zur Pflege der Bedingungen sind im Kapitel 2.3 zu finden.

### 3.2.4 Eigenschaftspflege

Im Bereich der Knoteneigenschaften kann über den Button „Werte ändern“ von der Ansicht in den Änderungsmodus gewechselt werden. Anschließend existieren für die vier Knotentypen vier Oberflächen zur Pflege der Eigenschaften.

#### Festwertknoten

Ein Festwertknoten stellt eine Konstante dar, die in der Berechnung verwendet wird. An einem Festwertknoten gibt es drei Felder. Den **Festwert**, einen zahlenbasierten Wert, der konstant in die Berechnung einfließt. Außerdem die **Kurzbeschreibung** (wird im Baum dargestellt als Knotenname) und **Beschreibung** haben lediglich informativen Charakter. Beide Felder werden abhängig von der Anmeldesprache gepflegt. In der folgenden Abbildung 16 ist diese Oberfläche dargestellt.

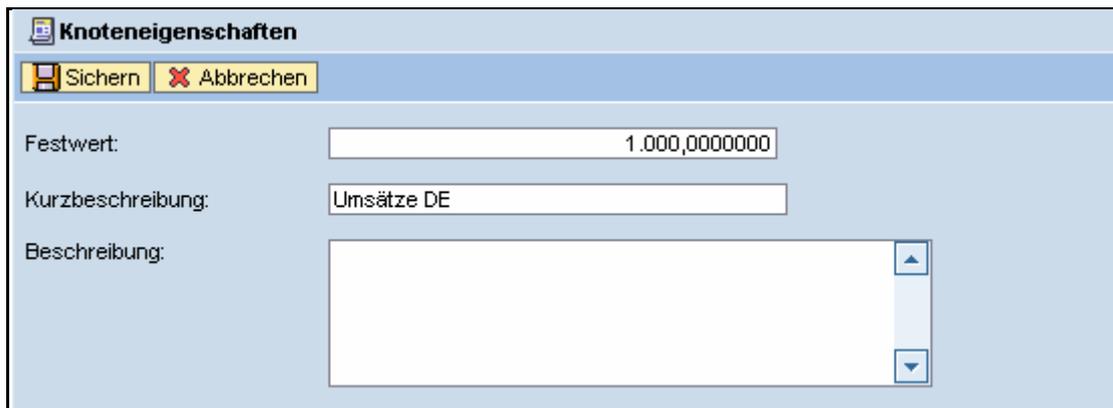


Abbildung 16: Screenshot - Knoteneigenschaften Festwertknoten

### Temporärer Knoten

Die Eigenschaften des temporären Knoten beinhalten neben den bekannten Feldern Kurzbeschreibung und Beschreibung (siehe Festwertknoten) das Feld Test Wert. Dieser Test Wert wird in der Testberechnung verwendet. Diese ist erklärt im Kapitel 3.2.5.



Abbildung 17: Screenshot - Knoteneigenschaften Temporärer Knoten

### Wertknoten

Bei Wertknoten und UserExit-Knoten gibt es neben dem Button „Werte ändern“ noch den Button „Einschränkungen kopieren“ (Abbildung 18). Dieser kopiert die expliziten Einschränkungen des gewählten Knoten in die Zwischenablage. Anschließend kann man auf einem anderen Wertknoten oder UserExit-Knoten diese wieder einfügen (nach einem Wechseln in den Änderungsmodus).



Abbildung 18: Screenshot - Werte ändern und Einschränkungen kopieren

In Abbildung 19 ist die Oberfläche zum Ändern der Knoteneigenschaften eines Wertknotens dargestellt. Neben den Feldern Test Wert, Kurzbeschreibung und Beschreibung stehen die folgenden Felder zur Verfügung.

Das Feld **Kennzahl** erfasst die Kennzahl in die das Ergebnis der Berechnung geschrieben oder der zu lesende Wert gelesen werden soll. Nach dem Feld wird über ein Kreuz oder Haken dargestellt ob der



### BusinessContent Knoten

Der Knotentyp „Business-Content“ ermöglicht es, Formulare gezielt für einzelne Szenarien auf Basis von eigenen UserExit Implementierungen zu definieren. Als Basis für jedes Formular steht das Formular des User-Exit-Knotens zur Verfügung. In diesem Formular werden dann einzelne Felder ausgeblendet. Des Weiteren ist es möglich, neue Felder hinzuzufügen.

**Hinweis:** Die hinzugefügten Felder werden technisch in die Einschränkungen gespeichert. Dies ist bei der Implementierung der zugrundeliegenden Implementierungsklasse zu beachten, da diese Einschränkungen ggf. sonderbehandelt werden müssen.

Ziel des Business-Content Knotentyps ist es, für projektspezifische Implementierungen einfach zu bedienende Oberflächen zu definieren. Folgende Business-Content Knotentypen sind im Auslieferungsumfang enthalten und können gemäß Customizing-Guide installiert werden.

- Erweiterte Validierung mit Toleranz (/ISV/CL\_AC\_VALIDATION\_ADV)  
Mit der „Erweiterten Validierung“ können in den Daten Inhalte geprüft werden und bestimmt durch das Ergebnis Werte gesetzt werden. Letztendlich wird abhängig von einer einfachen Wertprüfung auf eine Merkmalskombination ein Marker geschrieben. Im Tool steht ein ausführlicher Hilfetext zur Verwendung zur Verfügung.

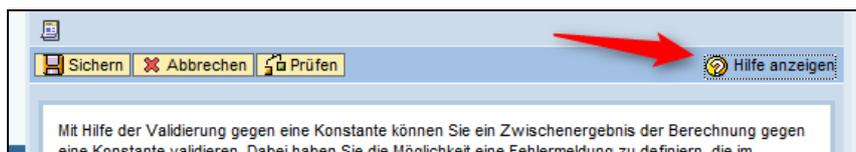


Abbildung 20: Screenshot - Hilfetext zur Erweiterten Validierung mit Toleranz

- Mapping-Vorschrift (/ISV/CL\_AC\_MAPPING\_DATA)
- Mapping-Gruppierung (/ISV/CL\_AC\_MAPPING\_GROUP)

**Hinweis:** Die Knotentypen Mapping-Gruppierung und Mapping-Vorschrift werden im Kapitel 4 beschrieben.

**Hinweis:** Die Definition eigener Business-Content Knoten ist in Kapitel 5.4 beschrieben.

### UserExit-Knoten

Der Knotentyp UserExit-Knoten besitzt im Unterschied zum Wertknoten nur das zusätzliche Feld **Klassenname**. In diesem Feld kann die Klasse eingetragen werden, die den eigenen Knotentyp darstellt. Die Interpretation der übrigen Felder (mit Ausnahme des Test Werts und der Beschreibungsfelder) ist dann abhängig von der jeweiligen Implementierung. Weitere Informationen zum Erstellen einer eigenen Implementierung sind im Kapitel 5.3 zu finden.



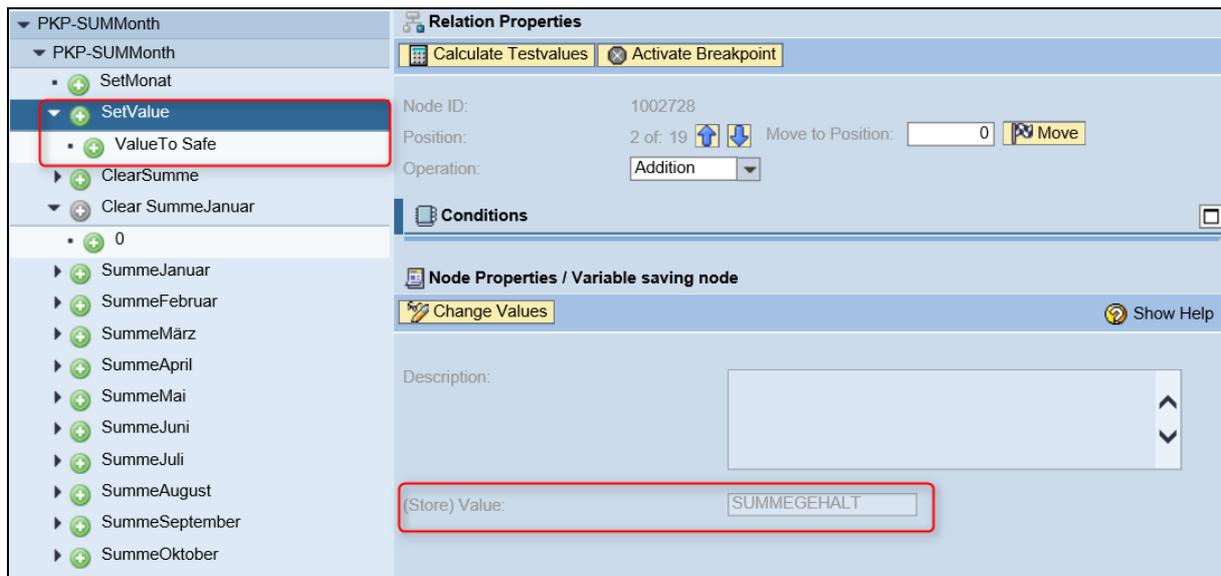


Abbildung 22 Screenshot - Value Saving Knoten

### Variable Reference Content Knoten

Eine weitere Möglichkeit ist der Variable Reference Knoten. Mithilfe diesem kann man die Kennzahl in einer Variable speichern. Dazu fügt man den Knoten als Blatt ein und trägt die Kennzahl in das Feld "Kennzahl" und den Variablen Namen in "(Speicher) Referenz Variable". Wenn der Variable Reference Knoten ein schreibender ist, dann lässt man nun die Kennzahl frei und trägt nur den Variablen Namen ein. Zusätzlich kann man auch Einschränkungen einfügen.

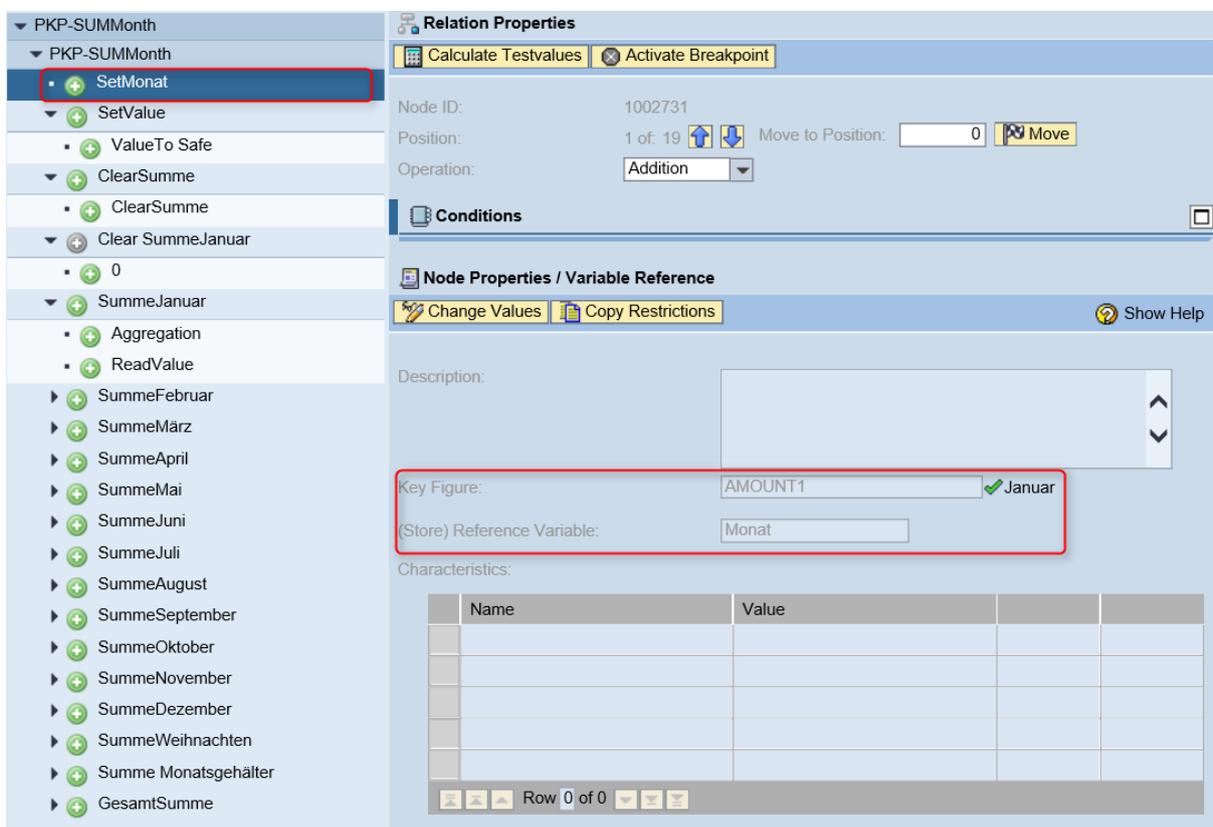


Abbildung 23 Screenshot - Variable Reference Content Knoten

### 3.2.5 Testwerte

AGIMENDO.calc bietet die Möglichkeit, Testwerte zu berechnen. Dies bedeutet, dass die Testwerte die im Bereich der Eigenschaftspflege im Baum eingetragen wurden, analog der Berechnungsvorschriften des Baumes miteinander verrechnet und verglichen werden.

Über den Button „Testwerte berechnen“ (Siehe Abbildung 13, Nummer (5)) wird die Testberechnung gestartet. Die Testwerteberechnung wird dann „ab dem ausgewählten Knoten“ gestartet.

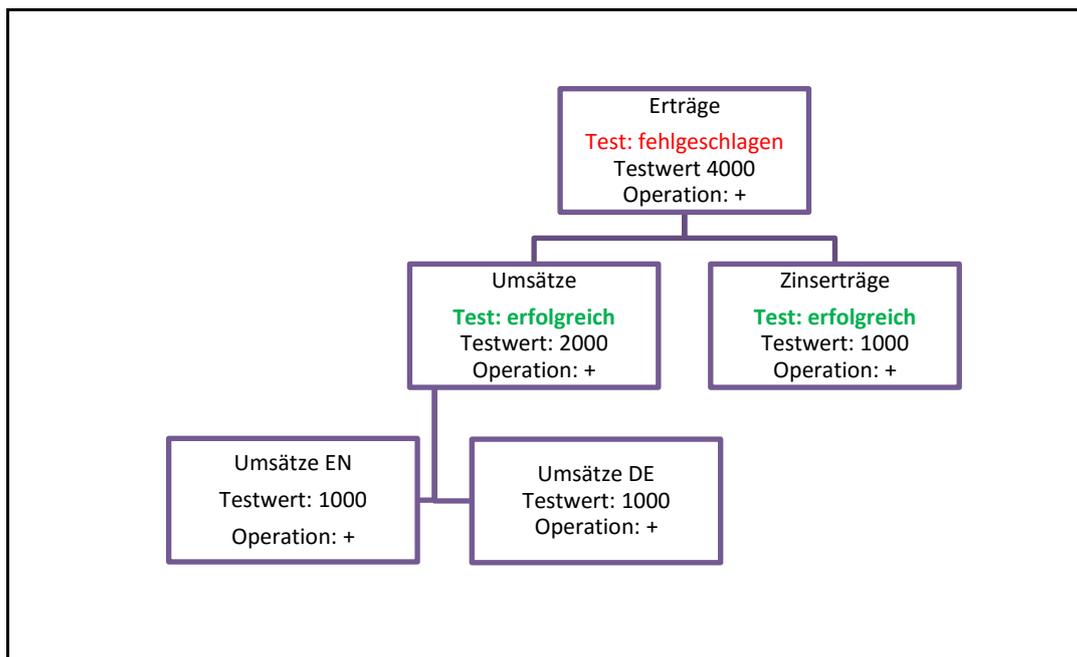


Abbildung 24: Beispiel – Testwertberechnung

Das in Abbildung 24 dargestellte und an den bisher genutzten Berechnungsbaum angelehnte Beispiel verdeutlicht die Testwertberechnung. Die Testwerte (bspw. 1000 bei Umsätze EN oder 4000 bei Erträge) wurden pro Knoten in der Pflegeanwendung eingegeben. Im Zuge der Testberechnung wird an den Blättern begonnen und die Testwerte von „Umsätze EN“ und „Umsätze DE“ laut Berechnungsvorschrift addiert und mit dem im Knoten „Umsätze“ gepflegten Testwert (2000) verglichen. Hier ist der Test erfolgreich. Auf Ebene Erträge ist dieser Test nicht erfolgreich. In Abbildung 25 ist die Ausgabe nach der durchgespielten Testwertberechnung dargestellt. Über die Liste der aufgetretenen Fehler kann so die „Spur“ zurück zum verursachenden Knoten verfolgt werden.



Abbildung 25: Screenshot - Ergebnis Testwertberechnung

## 4. Abbildung von Mapping-Regeln in AGIMENDO.calc

Wenn AGIMENDO.calc im Zuge der Expertenroutine ausgeführt wird (siehe Abschnitt Ausführungskontext), können Mappingregeln durchgeführt werden, die Zeilenweise auf den Eingabedaten ausgeführt werden. Im Folgenden wird beschrieben, wie diese Regeln konfiguriert und ausgeführt werden.

Ein Mappingbaum wird in der Regel beim Fortschreiben über eine Expertenroutine verwendet. Hierbei werden dann alle 1:1 abbildbaren Felder aus der Quelle übernommen und anschließend über Mappingvorschriften einzelne Felder neu gesetzt oder abgebildet. Hierbei können entweder Festwerte gesetzt werden oder Berechnungen durchgeführt werden. Typischerweise wird hier dann auch keine „eigene“ Aufteilung der Quelltable in Berechnungsdurchläufe vorgenommen, sondern die über die Eigenschaft EXPL\_LOOP (Kapitel 7.2) jede Zeile einzeln abgebildet. Über einen Mappingknoten kann hierbei ein Quell- auf ein Zielfeld übernommen werden. Über eine Mappinggruppe können mehrere Mappingvorschriften innerhalb einer „Ergebniszeile“ vorgenommen werden. Dadurch ist es dann auch möglich aus einer Quellzeile mehrere Zielzeilen zu erzeugen (Mehrere Mappinggruppen).

Hierfür stehen die folgenden Knotentypen zur Verfügung, die laut Customizing-Guide installiert werden können:

- `/ISV/CL_AC_MAPPING_GROUP` Group for a mapping definition
- `/ISV/CL_AC_MAPPING_DATA` Mapping Value Handler

## 5. Erweiterte Rechenregeln und UserExits

AGIMENDO.calc bietet eine Reihe von Möglichkeiten, über UserExits auf das Verhalten der Software Einfluss zu nehmen. Dabei existieren die folgenden Einstiegsmöglichkeiten, die ein unterschiedlich starkes Verständnis von AGIMENDO.calc, sowie verschiedenes Wissen über ABAP und Implementierungskonzepte voraussetzen.

- Allgemeine Konfiguration der Ausführungsplattform
- Konfiguration und Erweiterung der Pflegeanwendung
- Definition von eigenen Knotentypen
- Definition von eigenen Bedingungsstypen
- Implementierung eigener / Erweiterung bestehender Integrationsmodule

### Begriffsklärung Knoten, Knotentyp und „ValueHandler“

In der Pflegeanwendung können über den Knotentyp „User-Exit“ selbstdefinierte oder über erweiterte Inhalte ausgelieferte Knotentypen verwendet werden. Technisch gesehen wird in der Pflege ein Baum aus Knoten aufgebaut und jedem Knoten je eine Klasse zum Lesen und zum Schreiben von Werten zugeordnet. Zur Laufzeit wird pro Knoten beim Baumdurchlauf diese Klasse instanziiert und für das Lesen von Quellwerten (Methode `GET_VALUE`) bzw. Schreiben von Ergebnissen (`SET_VALUE`) genutzt. Diese Klasse muss das Interface `/ISV/IF_PCE_VALUE_HANDLER` implementieren, sollte aber zur Sicherstellung der Kompatibilität und für eine einfachere Implementierung von der abstrakten Klasse `/ISV/CL_PCE_ABSTRACT_VH` erben. Im Kapitel 5.3 wird auf die Arbeitsweise und das Erstellen von eigenen Knotentypen eingegangen.

**Hinweis:** Durch die technische Bezeichnung ist der Begriff „ValueHandler“ entstanden, der eine Klasse bezeichnet, die zur Wertebehandlung eingesetzt wird. In diesem Dokument wird diese Klasse synonym als Knotentyp bezeichnet.

**Hinweis:** In der Pflegeoberfläche wird aktuell immer dieselbe Klasse zum Lesen und Schreiben zugeordnet. Prinzipiell könnten aber auf Datenbankebene auch unterschiedliche Klassen zugeordnet werden. Dies ist aber m.E. nicht notwendig, da jeder Knoten entweder liest oder schreibt.

## 5.1 Allgemeine Konfiguration der Ausführungsplattform

Neben den im folgenden Kapitel beschriebenen Customizing-Einstellungen gibt es eine technische Customizing-Einstellung, die sich auf das Berechnungsverhalten von AGIMENDO.calc auswirkt. Mit einem Customizing-Eintrag `ALWAYS_SET_FOR_CLASS` besteht die Möglichkeit, der Ausführungsplattform mitzuteilen, dass auch auf den Blättern im Baum die schreibende Methode (`SET_VALUE`) aufgerufen werden soll. Dies macht vor allem im Kontext der Transformationen Sinn, wenn mit Referenzdaten gearbeitet wird. (Alle aus den Referenzdaten gelesenen Werte werden dann auch in das Ziel mit übernommen, somit ist im Ergebnis ersichtlich, auf welchen Basisdaten die Berechnung erfolgte). Der Wert für `ALWAYS_SET_FOR_CLASS` (Spalte `VALUE` in der Tabelle `/ISV/PCE_SETTING`) entspricht einem Pattern, das auf die Klassennamen der Knotentypen angewandt wird. Also beispielsweise `„/ISV/*“` um auf alle Knotentypen aus dem ISV-Namensraum diese Funktion anzuwenden.

**Hinweis:** Aktuell gilt diese Einstellung für alle Bäume eines Systems. Zukünftig wird diese Einstellung auf Baumebene bzw. im Aufruf der Ausführungsebene konfigurierbar sein.

## 5.2 Konfiguration und Erweiterung der Pflegeanwendung

Für die Pflegeanwendung von AGIMENDO.calc können die in der folgenden Tabelle dargestellten Einstellungen in der Datenbanktabelle `/ISV/PCE_SETTING` vorgenommen werden. Siehe auch Customizing-Guide

Customizingeintrag	Defaultwert	Beschreibung
<code>AUTH_CHECK_ENABLED</code>	X	Wenn gesetzt (X), wird die Berechtigungsprüfung durchgeführt, wenn nicht, können alle Benutzer alle Operationen durchführen
<code>EXPERT</code>	true	Wenn gesetzt (true) werden IDs in der Oberfläche mit angezeigt und erweiterte Pflegemöglichkeiten angeboten
<code>TREE_DEFAULT_LANGUAGE</code>	D	Standardsprache, wenn aktuelle Sprache nicht gepflegt, wird diese angezeigt, einstelliger (!) SAP-Ländercode
<code>TREE_FILTER_CLASS</code>	<code>/ISV/CL_PCE_CHANGE_TREE</code>	User-Exit um die angebotenen Bäume in der Auswahlliste nachzubearbeiten.
<code>ICON_CALCULATION_OPERATION_ADD</code>	add.png	Icon für das „+“ Symbol
<code>ICON_CALCULATION_OPERATION_DIV</code>	divide.png	Icon für das „/“ Symbol
<code>ICON_CALCULATION_OPERATION_MUL</code>	multiply.png	Icon für das „*“ Symbol
<code>ICON_CALCULATION_OPERATION_SUB</code>	substrate.png	Icon für das „-“ Symbol

## 5.3 Definition Knotentypen

Bei der Implementierung eigener Knotentypen geht es in aller Regel um das Ermitteln von Werten, die anschließend in die Berechnung einfließen und auf dem „normalen“ Weg in eine Kennzahl geschrieben werden. Bei der Ermittlung von Werten sind unterschiedliche Anforderungen denkbar, die dann ein sehr unterschiedliches Vorgehen bei der Implementierung zur Folge haben.

Allen Definitionen eigener Knotentypen gemein ist, dass eine Klasse im ABAP implementiert werden muss die entweder von einem bestehenden Knotentyp erbt und gezielt einzelne Methoden redefiniert oder von der allgemeinen Implementierungsvorlage /ISV/CL\_PCE\_VHANDLE erbt und die beiden Methoden GET\_VALUE und SET\_VALUE vollständig selbst implementiert. Von der Ausführungsplattform wird im Verlauf der Berechnung eines Baumes pro Knoten entweder die GET\_VALUE-Methode (Knoten ist ein Blatt und hat keine Kinder muss also „gelesen“ werden) oder die SET\_VALUE-Methode (Knoten ist innerhalb des Baumes, hat Kinder die einen Wert geliefert haben, muss also „geschrieben“ werden) aufgerufen.

Im Folgenden werden anhand verschiedener Beispiele Szenarien dargestellt, wie die Implementierung der beiden genannten Methoden erfolgen kann.

**Hinweis:** Die folgenden Beispiele beziehen sich auf die gängige Anforderung, dass Werte gelesen werden müssen. Analog können natürlich auch Werte geschrieben werden, allerdings werden die Ergebnisse der Berechnung meist auf dem „normalen“ Weg in die Datentabelle geschrieben, da die berechneten Werte im BW weiterverwendet werden sollen.

#### Beispiel: Nachlesen aus Datenbanktabelle

Der für das Verständnis einfachste Fall ist, dass der Wert, der durch einen eigenen Knotentyp ermittelt werden soll, von „irgendwo“ außerhalb von AGIMENDO.calc ermittelt werden muss. Beispielsweise soll eine Konstante aus einer transparenten Datenbanktabelle nachgelesen werden.

Um einen Knotentyp zu erstellen, der aus einer Datenbanktabelle einen Wert nachliest, ist eine Klasse zu implementieren, die von der abstrakten Klasse /ISV/PCE\_ABSTRACT\_VH erbt und die beiden Methoden SET\_VALUE und GET\_VALUE implementiert. In unserem Beispiel kann die SET\_VALUE Methode leer implementiert werden.

In der GET\_VALUE Methode wird der Wert aus der Datenbanktabelle gelesen und in den Ergebnisparameter EV\_VALUE kopiert. Im folgenden Beispiel wird eine Klasse ZCL\_CONSTANT\_READ\_VH implementiert, die von der Klasse /ISV/PCE\_ABSTRACT\_VH erbt. Im einfachsten Beispiel wird direkt mit einem konstanten SELECT-Statement aus der Datenbank gelesen.

```
METHOD /ISV/IF_PCE_VALUE_HANDLER~GET_VALUE.  
    SELECT SINGLE constant_value  
        FROM ZMY_CONSTANT  
        INTO ev_value  
        WHERE constant_name = 'HERSTELLKONSTANTE'  
ENDMETHOD.
```

Die Einbindung des neu erstellten Knotentyps erfolgt über einen User-Exit Knoten. Die Konfiguration sieht dann beispielsweise wie in der folgenden Abbildung dargestellt aus. Für ein echtes Szenario müssten jetzt zumindest noch die drei Herstellkosten-Knoten über Bedingungen bspw. verschiedenen Gesellschaften zugewiesen werden.

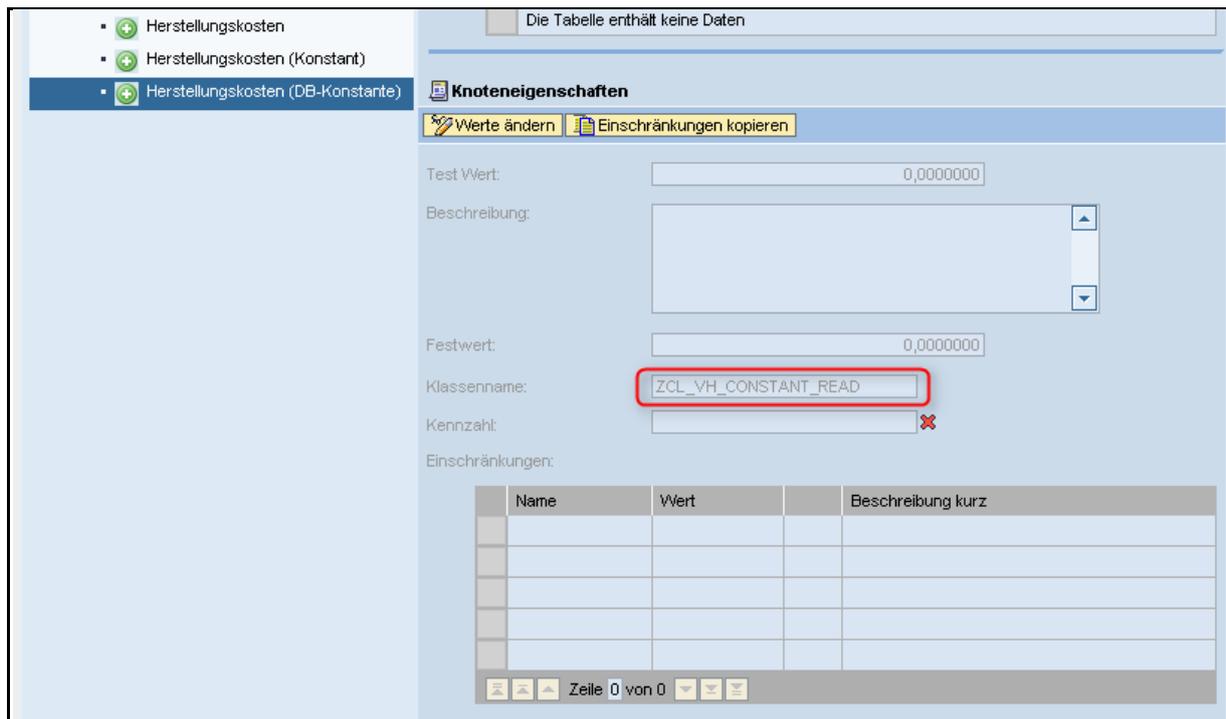


Abbildung 26: Screenshot - Einbindung der angelegten Klasse in die Berechnungslogik

Dieser Knotentyp ist natürlich sehr statisch und aufgrund der SELECT-Bedingung wahrscheinlich nicht beliebig einsetzbar. Trotzdem kann es sinnvoll sein, wiederkehrende „fast-Konstanten“ auf diesem Weg auszulesen. Eine einfache aber sehr wirkungsvolle Möglichkeit diesen Knotentyp zu erweitern, wäre, in die SELECT-Bedingung beispielsweise die aktuelle Gesellschaft mit aufzunehmen. Diese können entweder in den globalen Einschränkungen, in den expliziten Einschränkungen oder den Initialwerten vorliegen (vgl. Kapitel 2.1). Folgende beispielhafte Code-Strecke ermittelt die Gesellschaft und liest dann analog zum vorangegangenen Beispiel aus der Datenbanktabelle nach.

```

METHOD /isv/if_pce_value_handler~get_value.

    DATA lv_fieldname      TYPE /isv/pce_characteristic_name.
    DATA lv_value          TYPE /isv/pce_characteristic_value.
    DATA ls_data_selection LIKE LINE OF it_data_selection.

    * translate the iobject name into field name if necessary
    me->/isv/if_pce_value_handler~translate_field_name(
        EXPORTING
            iv_iobjname = '0COMP_CODE'
            iv_translate_to_fieldname = me->gv_translate_to_fieldname
        IMPORTING
            ev_fieldname = lv_fieldname
    ).

    * get the current restriction if configured
    lv_value = me->/isv/if_pce_value_handler~get_restriction_by_name( lv_fieldname ).

    * test if there was a restriction, if not check it_data_selection -> implicit restrictions
    IF lv_value IS INITIAL.
        READ TABLE it_data_selection INTO ls_data_selection WITH KEY iobjnm = lv_fieldname.
        IF sy-subrc = 0.
            MOVE ls_data_selection-low TO lv_value.
        ELSE.

```

```

        lv_value = 'HERSTELLKONSTANTE'.
    ENDIF.
ENDIF.

* perform DB-select
SELECT SINGLE constant_value
  FROM zmy_constant
  INTO ev_value
  WHERE constant_name = lv_value.

ENDMETHOD.

```

**Hinweis:** Der Aufruf der Methode **translate\_field\_name** bewirkt, dass wenn sich die Ausführungsplattform in einem Modus für Feldnamen anstatt InfoObjektnamen befindet, der fest vorgegebenen InfoObjektnamen (OCOM\_CODE) in den korrespondierenden Feldnamen übersetzt wird. Hintergrund ist, dass im Transformationskontext mit Feldnamen gearbeitet wird, im Planungsfunktionskontext mit InfoObjekt-Namen. Die Pflege im Rechenbaum erfolgt aber immer mit InfoObjektnamen.

**Hinweis:** Sollte der erstellte Knotentyp im Baum nicht als Blatt verwendet werden, und so eigentlich einen Wert schreiben, so wird, wenn die SET\_VALUE-Methode leer implementiert wurde, der Wert einfach nicht weggeschrieben. In diesem Szenario ist also das Verwenden des Knotens im Baum nur an den Blättern des Baumes sinnvoll, da nur hier die GET\_VALUE-Methode aufgerufen wird. Eine Verwendung innerhalb des Baumes würde aber nicht zu einem Fehler führen. Der Knoten hätte dann den Charakter eines temporären Knotens.

#### Beispiel: Lesen aus der Vorperiode

Ein anderes Szenario wäre, dass der gesuchte Wert direkt in der Datentabelle vorliegt, aber nicht direkt über die aktuell gültigen Einschränkungen gefunden wird. Beispielsweise muss der gesuchte Wert aus der Vorperiode gelesen werden. Im Kapitel 2.1 wurde erklärt, wie im Standard die Ermittlung des Wertes aus der Datentabelle erfolgt. Dieser Logik folgend, muss für einen Zugriff auf die Vorperiode lediglich zur Laufzeit die Einschränkung für die Periode richtig geändert werden.

Da es sich beim Ändern der Einschränkungen für den Zugriff auf die Datentabelle um eine gängige Anforderung handelt, gibt es die Möglichkeit nur hier Einfluss zu nehmen ohne die Implementierung der GET\_VALUE bzw. SET\_VALUE Methode für den Zugriff auf die Datentabelle anpassen zu müssen.

Um das genannte Szenario „Lesen aus der Vorperiode“ zu implementieren, muss zunächst eine Klasse erstellt werden, die vom Standard-Wertknotentyp (/ISV/CL\_PCE\_VH\_BI\_DATA) erbt. Diese Klasse beinhaltet zwei Methoden, die einen Zugriff auf die Einschränkungen **vor** dem Zugriff auf die Datentabelle ermöglichen.

- Vor dem Lesen (MODIFY\_SELECTION\_CRITERIA\_GET)
- Vor dem Schreiben (MODIFY\_SELECTION\_CRITERIA\_SET)

In unserem Fall müssen wir die Methode (MODIFY\_SELECTION\_CRITERIA\_GET) redefinieren und hier die Periode auf die Vorperiode setzen. Dies geschieht beispielhaft wie folgt.

```

METHOD MODIFY_SELECTION_CRITERIA_GET.

DATA lv_fieldname          TYPE /isv/pce_characteristic_name.
DATA lv_iobjnm             TYPE /isv/pce_characteristic_name.

```

```

DATA ls_selection_criteria LIKE LINE OF ct_selection_criteria.

lv_iobjnm = 'OFISCPER'.

* for read operations we need probably a translated field name (transformation context)
me->/isv/if_pce_value_handler~translate_field_name(
  EXPORTING
    iv_iobjname = lv_iobjnm
    iv_translate_to_fieldname = me->gv_translate_to_fieldname
  IMPORTING
    ev_fieldname = lv_fieldname
).

READ TABLE ct_selection_criteria WITH KEY iobjnm = lv_fieldname INTO ls_selection_criteria.
IF sy-subrc = 0.

* remove old selection criteria
DELETE ct_selection_criteria WHERE iobjnm = ls_selection_criteria-iobjnm.
* set the period to the period before, subtract one month in example

* !! be careful: this is a example implementation and will only work in special cases !!
ls_selection_criteria-low = ls_selection_criteria-low - 1.
APPEND ls_selection_criteria TO ct_selection_criteria.

ENDIF.

ENDMETHOD.

```

**Hinweis:** Die vorangegangene Beispielimplementierung betrachtet aus Übersichtsgründen nicht die Sonderfälle an Jahresgrenzen. Es wird lediglich der Monat um einen herunter gezählt. Dies führt an der Jahresgrenze natürlich zu einem Fehler. Auch ist zu beachten, dass ggf. mehr als eine Einschränkung geändert werden muss, um erfolgreich zu lesen. Beispielsweise könnte OFISCPER und OFISCPER3 in den globalen Einschränkungen enthalten sein. Für einen Zugriff auf die Datentabelle reicht allerdings OFISCPER. OFISCPER3 müsste also entweder parallel angepasst werden oder aus der Menge der Selektionskriterien (Einschränkungen) entfernt werden. Dies könnte durch das folgende Code-Stück geschehen:

```

* . . .

lv_iobjnm = 'OFISCPER3'.

* for read operations we need probably a translated field name (transformation context)
me->/isv/if_pce_value_handler~translate_field_name(
  EXPORTING
    iv_iobjname = lv_iobjnm
    iv_translate_to_fieldname = me->gv_translate_to_fieldname
  IMPORTING
    ev_fieldname = lv_fieldname
).

DELETE ct_selection_criteria WHERE iobjnm = lv_fieldname.

* . . .

```

**Hinweis:** Über die genannten MODIFY-Methoden können nur die globalen und expliziten Einschränkungen geändert werden, nicht aber die Initialwerte.

## 5.4 Definition von BusinessContent Knoten zu einem Knotentyp

Einen BusinessContent stellt ein Formular zu einem eigenen Knotentyp (Implementierungsklasse) dar. Diese Formulare können auch manuell angelegt werden. Dazu muss allerdings manuell technischer Tabelleninhalt gepflegt werden.

**Hinweis:** *Es wird nachdrücklich empfohlen diese Tabelleninhalte im Entwicklungssystem zu pflegen und anschließend zu transportieren.*

Folgende Tabellen müssen hierbei gepflegt werden:

- /ISV/PCE\_UI\_CONF – Zentrale Tabelle / Basiskonfiguration / Sichtbarkeiten der Standardfelder. Alle folgenden Tabellen hängen über den Primärschlüssel (CONFIG\_ID) zusammen
- /ISV/PCE\_UI\_DYN – Definition der Eingabefelder für Einschränkungen in der Darstellung – Jeder Eintrag steht für ein zusätzliches Eingabefeld, das oberhalb der Einschränkungen angezeigt wird. Texte werden mehrsprachig in der Beschreibungstabelle (/ISV/PCE\_DESCR) abgelegt und über die ID im Feld DESCRIPTION verknüpft. Typen für die Eingabe sind heute INPUT\_FIELD und CHECKBOX (/ISV/PCE\_UI\_DYN -FIELD\_TYPE)
- /ISV/PCE\_UI\_PRE – Vordefinierte Werte für die Standardfelder
- /ISV/PCE\_UI\_RES – Liste von Einschränkungen mit Vorbelegung, hierüber können zum Beispiel Festwerte in den Einschränkungen definiert werden, die dann auch gesetzt sind wenn die Einschränkungen gar nicht angezeigt werden (/ISV/PCE\_UI\_CONF~CHARACTERISTIC).
- /ISV/PCE\_UI\_TEXT – Mehrsprachigen Texte für die Darstellung

*Hinweis: Die Tabelle /ISV/PCE\_UI\_TYPE ist obsolet.*

## 5.5 Definition Bedingungstypen

### Entwicklung eigener Bedingungstypen

Bei der Auswertung einer Bedingung kann auf alle Informationen zum Berechnungszeitpunkt zugegriffen werden.

Um einen eigenen Bedingungstyp zu entwickeln muss eine Klasse implementiert werden, die von der Klasse /ISV/CL\_PCE\_CONDITION erbt. Damit diese Klasse über die Oberfläche gepflegt werden kann, muss sie in die Tabelle /ISV/PCE\_COND eingetragen werden. Hier müssen die folgenden Werte eingetragen werden:

- COND\_ID: Frei wählbarer eindeutiger Schlüssel
- COND\_SYMBOL: Angezeigtes Symbol in der Dropdownbox (semantischer Schlüssel, eindeutig)
- COND\_CLASS: Implementierte Klasse für den Bedingungstyp (erbt von /ISV/IF\_PCE\_CONDITION)
- DESCRIPTION: ID einer Beschreibung (vorerst nicht implementiert)
- COND\_TYPE: "1" (für spätere Entwicklungen vorgesehen)
- ENABLE\_VALUE1: Soll der „Wert 1“ in der Pflege eingabebereit sein („X“: ja „“: nein)
- ENABLE\_VALUE2: Soll der „Wert 2“ in der Pflege eingabebereit sein („X“: ja „“: nein)

In der eigenen Implementierung müssen in der Regel folgende Methoden überschrieben werden:

- /ISV/IF\_PCE\_CONDITION~TEST: Implementierung der Bedingung

```

METHOD /isv/if_pce_condition~test.
  DATA ls_selection LIKE LINE OF it_data_selection.
  READ TABLE it_data_selection INTO ls_selection WITH KEY iobjnm = gs_db_row-cond_char.
  IF ls_selection IS INITIAL.
    " restriction is not within the restrictions -> search within the it_initial values
    READ TABLE it_initial_values INTO ls_selection WITH KEY iobjnm = gs_db_row-cond_char.
  ENDIF.
  IF ls_selection IS NOT INITIAL.
    IF ls_selection-low EQ gs_db_row-cond_valuel.
      ev_result = 'X'.
    ELSE.
      ev_result = '-'.
    ENDIF.
  ELSE.
    ev_result = '-'.
  ENDIF.
ENDMETHOD.

```

**Hinweis:** Der Rückgabetyt von der Methode `/isv/if_pce_condition~test` ist ein CHAR1. Das Ergebnis wird als WAHR interpretiert genau dann wenn das Ergebnis „X“ beinhaltet. Alle anderen Werte werden als FALSCH interpretiert.

- `/ISV/IF_PCE_CONDITION~TO_STRING_LOCAL` : String-Darstellung der Bedingung

```

METHOD /ISV/IF_PCE_CONDITION~TO_STRING_LOCAL.
  CONCATENATE me->gs_db_row-cond_char
    '='
    me->gs_db_row-cond_valuel
    INTO rv_string SEPARATED BY space.
ENDMETHOD.

```

**Hinweis:** Technisch gesehen muss lediglich das Interface `/ISV/IF_PCE_CONDITION` implementiert werden. Allerdings werden eine Reihe von technisch bedingten Methoden benötigt (zur korrekten Auswertung und Instanziierung der Objekte) die in fast allen Fällen wohl immer gleich sind und in `/ISV/PCE_CL_CONDITION` bereits implementiert wurden.

## 6. Verwaltungstool zur Wiederverwendung von Knoten

Nachdem innerhalb von AGIMENDO.calc ein Knoten in mehr als einem Baum verwendet werden kann und hierbei auch alle darunter verknüpften Knoten mit wiederverwendet werden, entsteht eine gewisse Komplexität. Um Baumknoten auch ohne den Kontext eines Rechenbaumes zu verwalten und einen Verwendungsnachweis für einzelne Knoten zu ermöglichen gibt es eine erweiterte Pflegeanwendung.

In dieser Anwendung kann zum einen je Knoten ein Verwendungsnachweis (liste der Bäume die einen Knoten Verwenden) erstellt werden (Inkl. Pfad zum knoten) als auch einzelne Knoten gelöscht werden.

Die Anwendung ist unter folgender URL erreichbar:

[http\(s\)://<host>:<port>/sap/bc/webdynpro/isv/agim\\_ce\\_node\\_maint](http(s)://<host>:<port>/sap/bc/webdynpro/isv/agim_ce_node_maint)

In der Abbildung 28 ist die Anwendung dargestellt. In der oberen Tabelle kann ein Knoten ausgewählt werden (über die Filterzeile können hier einzelne Knoten gefunden werden). Ein ausgewählter Knoten kann gelöscht werden. Hierbei wird vor einem Löschen mit Kurzinformationen zur aktuellen Verwendung rückgefragt.

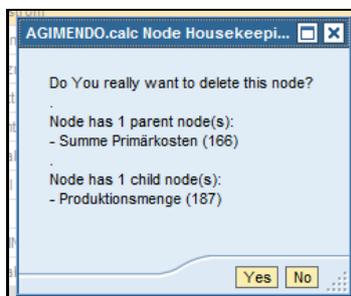


Abbildung 27: Screenshot - Rückfrage bei Löschen eines Knoten

Knoten die aktuell nirgendwo referenziert werden, haben in der Spalte „in use“ keinen Haken. In der Spalte „Usages“ stehen die anzahlen der Referenzen „im Baum nach oben“, in der Spalte Child Nodes die anzahl der Kindknoten. Ein Knoten nicht verwendet werden und trotzdem Kindknoten besitzen.

In der Spalte „Used in trees“ werden alle Baum-IDs aufgelistet, in denen der Knoten verwendet wird.

AGIMENDO.calc - Node Maintenance								
List of all available tree nodes								
Node ID	In Use	Short Description	Usages	Child Nodes	Used in trees	User Name	Filter Settings	
170	<input checked="" type="checkbox"/>	Fremdstrom	1	1	101, 103	D155	30.10.2009 00:00:00	
185	<input checked="" type="checkbox"/>	Mengenunterschied	2	6	14, 101, 206	D038	20.11.2009 00:00:00	
186	<input checked="" type="checkbox"/>	Absatzmenge	1	0	14, 101, 206	D038	20.11.2009 00:00:00	
187	<input checked="" type="checkbox"/>	Produktionsmenge	4	0	14, 101, 103, 206, 1010	D038	20.11.2009 00:00:00	
188	<input type="checkbox"/>	Gesamteinzelkosten	0	2		D049	30.10.2009 00:00:00	
189	<input checked="" type="checkbox"/>	Materialsummenkosten	1	0		D212	30.10.2009 00:00:00	
195	<input checked="" type="checkbox"/>	Anzahl Mitarbeiter (Kostenstelle)	1	0	101	D130	30.10.2009 00:00:00	
200	<input type="checkbox"/>	<leer>	2	2	14, 15, 101, 206	D130	13.08.2014 07:43:28	
203	<input checked="" type="checkbox"/>	EUR/MIN	4	0	14, 15, 101, 206	D212	20.11.2009 00:00:00	
207	<input type="checkbox"/>	Sozialabgaben 21%	0	0		D130	30.10.2009 00:00:00	

Abbildung 28: Screenshot – Housekeeping Anwendung für Knoten

Wenn ein Knoten in der Liste ausgewählt wird, erscheint ein Detailbericht (Abbildung 29) für diesen Knoten. Hier wird für jede Verwendung angezeigt, ausgehend von welchem Baum ein Knoten wo in diesem Baum verwendet wird. Zum einen wird der Pfad der KnotenIDs angezeigt und zum anderen auch der Pfad der Knotennamen. Außerdem steht eine Liste der Elternknoten und Kindknoten zur Verfügung.

List of all usages within trees of the selected node						
Tree ID	Treename	Node ID Path	Text Path	Node ID	Node Description	
101	GuV + Unternehmensplanung	-> 100 -> 163 -> 224 -> 166 ->	-> PACE Rechenengine -> Gemeinkostenplanung -> Summe Gesamtkosten -> Summe Primärkosten ->	170	Fremdstrom	
103	POS 0002	-> 163 -> 224 -> 166 ->	-> Gemeinkostenplanung -> Summe Gesamtkosten -> Summe Primärkosten ->	170	Fremdstrom	

List parent nodes of the selected node		List child nodes of the selected node	
Node ID	Node Description	Node ID	Node Description
166	Summe Primärkosten	187	Produktionsmenge

Abbildung 29: Screenshot – Knotendetails

## 7. Ausführungsplattform Einbindungen und Aufruf

Wie eingangs eingeführt besteht AGIMENDO.calc aus einer Ausführungsplattform die mit zwei Integrationsmodulen ausgeliefert wird. Diese Ausführungsplattform ermöglicht es, die Durchführung der Rechenregeln und damit die Pflege der Rechenregeln unabhängig von dem jeweiligen Kontext zu betrachten. Beispielsweise wird im Kontext einer Planungsfunktion auf Basis von InfoObjekt-Namen gearbeitet, während im Kontext von Transformationen der Basis von Feldnamen gearbeitet wird. Diese Umsetzung nimmt die Ausführungsplattform von AGIMENDO.calc vor um ein einheitliches Pflegen der Rechenregeln auf InfoObjekt-Namen zu ermöglichen.

Grundsätzlich müssen drei Eigenschaften gesetzt werden, damit die Ausführungsplattform eine Berechnung durchführen kann.

- **TreeID:** Die eindeutige Nummer identifiziert einen Rechenbaum, der in der Pflege erstellt wurde.
- **Globale Einschränkungen:** Die globalen Einschränkungen sind eine Menge von Merkmalen mit zugehörigen Werten, die während der Berechnung zusätzlich zu den explizit definierten Einschränkungen gelten.
- **Initialwerte:** die Initialwerte sind ebenfalls eine Menge von Merkmalen mit zugehörigen Werten, die bei einer Neuanlage eines Datensatzes gesetzt werden.

Wichtig ist, dass je Konfiguration des Rechenbaums (abhängig vom Integrationsmodul) die eigentliche Berechnung bei einem Aufruf unterschiedlich oft durchgeführt wird. Während einer Berechnung darf jedem Merkmal in den globalen Einschränkungen sowie in den Initialwerten immer nur genau ein Wert zugewiesen sein.

Im Folgenden werden die beiden Integrationsmodule mit den angebotenen Konfigurationsmöglichkeiten vorgestellt.

### 7.1 Integrationsmodul: Planungskontext

Im Planungskontext wird AGIMENDO.calc über Planungsfunktionen aufgerufen. Die Datentabelle wird in diesem Fall mit allen Daten der Planungsfunktion gefüllt. Im Lieferumfang von

AGIMENDO.calc sind zwei Planungsfunktionstypen für das Erstellen von Planungsfunktionen enthalten. Der erste unterstützt keine Blockung, der zweite unterstützt die Blockung innerhalb der Planungsfunktionen.

Die beiden Planungsfunktionstypen können über die Transaktion RSPLAN, wie in Abbildung 30 dargestellt, eingesehen werden.

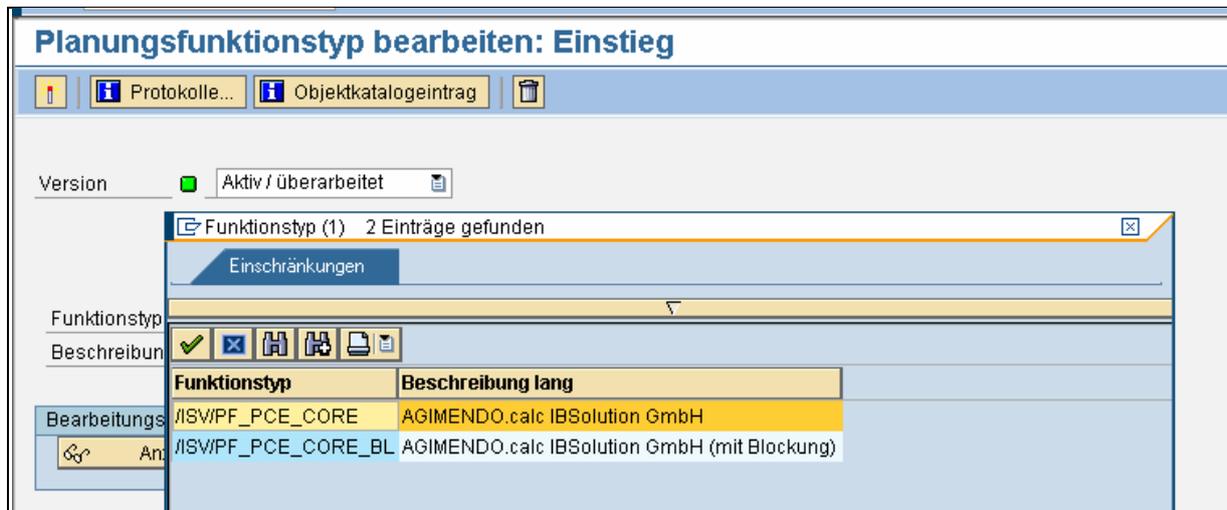


Abbildung 30: Screenshot - Transaktion RSPLAN

Mit Hilfe des „Modelers“ (zu starten über die Transaktion RSPLAN) können auf bestehenden Aggregationsebenen neue Planungsfunktionen angelegt werden. Dabei muss im Typ entweder der „blockende“ oder der „nicht blockende“ Planungsfunktionstyp ausgewählt werden. (Dargestellt in der folgenden Abbildung 31)

Grundsätzlich führt eine Planungsfunktion ohne Blockung den Planungsfunktionstyp (in unserem Fall den Aufruf des Rechenbaumes) einmal aus. Im Unterschied dazu wird der Rechenbaum durch eine blockende Planungsfunktion mehrfach aufgerufen. Dabei wird die Datentabelle anhand der Blockmerkmale in Teile zerlegt und nur in diesen Teilen an AGIMENDO.calc übergeben. Das hat zur Folge, dass nur die Daten innerhalb eines Blockes für die Berechnung zur Verfügung stehen.

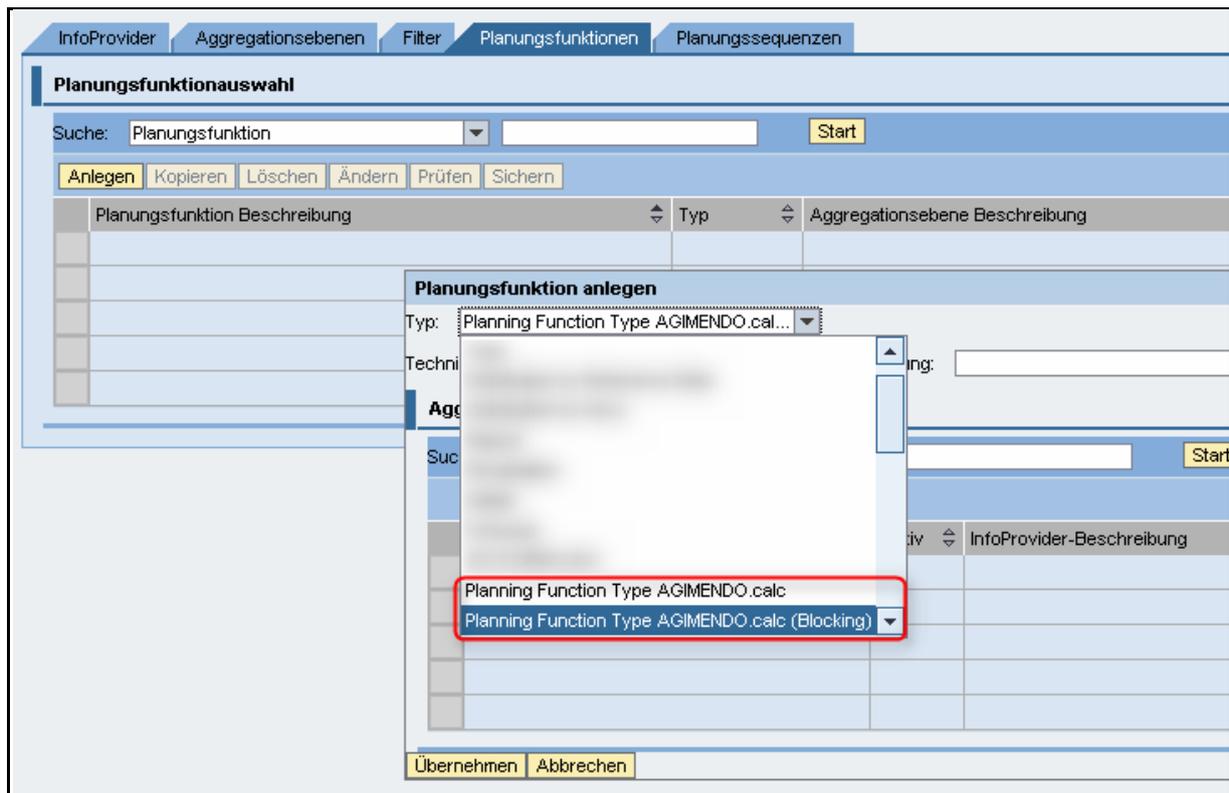


Abbildung 31: Screenshot - Neuanlage einer Planungsfunktion

Anschließend kann über die Eigenschaften der Planungsfunktion die Ausführung des Rechenbaumes beeinflusst werden. Folgende Eigenschaften müssen hierbei definiert werden:

- IV\_TREEID: Auswahl des Rechenbaumes, der zur Ausführung gebracht wird. Die eindeutige ID eines Rechenbaumes kann über die Pflegeoberfläche eingesehen werden (siehe Abbildung 9).
- DS\_TIME: Globale Einschränkungen, die für die gesamte Berechnung gelten sollen. Hierbei ist zu beachten, dass wenn einem Merkmal mehr als ein Wert zugeordnet ist, die Berechnung für jeden dieser Werte durchgeführt wird. Am Beispiel aus Kapitel 2.1 bedeutet das, wenn in DS\_TIME dem Merkmal Gesellschaft die Werte 1000 und 2000 zugewiesen wird, wird die Berechnung einmal für die Gesellschaft 1000 und einmal für die Gesellschaft 2000 durchgeführt. **Achtung:** Wenn für mehrere Merkmale mehrere Werte angegeben werden, wird das **vollständige Kreuzprodukt** der Kombinationen gebildet. Also 3 Gesellschaften für 5 Jahre für 3 Buchungskreise ergibt  $3 \times 5 \times 3 = 45$  Durchläufe. Außerdem muss darauf geachtet werden, dass **mindestens eine globale Einschränkung** definiert wird, da ansonsten die Anzahl der Baumdurchläufe null ist und nichts berechnet wird.
- DS\_INITIAL\_VALUES: Initialwerte, die für die Neuanlage von Werten genutzt werden sollen.

Über die Eigenschaft IV\_MOD\_SEL\_USER\_EXIT kann ein UserExit definiert werden, der es ermöglicht, auf die globalen Einschränkungen vor der Berechnung einzuwirken. Dies ermöglicht beispielsweise die nicht relevanten erzeugten Kombinationen (durch das gebildete vollständige Kreuzprodukt) zu entfernen. Der Wert des Parameters IV\_MOD\_SEL\_USER\_EXIT muss einem Klassennamen entsprechen, der das Interface /ISV/IF\_PCE\_SELECTION\_CRITERIA implementiert.

**Hinweis:** Die Blockmerkmale (einer blockenden Planungsfunktion) werden zur Laufzeit automatisch als Initialwerte in die Konfiguration eingefügt, dadurch wird sichergestellt, dass innerhalb eines Blocks nur Daten für diesen Block angelegt werden.

**Hinweis:** Die zusätzlichen Eigenschaften `IV_TREETAG` ist für zukünftige Funktionen vorgesehen und besitzt im heutigen Auslieferungszustand keine Funktion.

## 7.2 Integrationsmodul: Transformationskontext

AGIMENDO.calc kann über das Integrationsmodul für Transformationen innerhalb der Experten- oder der Endroutine aufgerufen werden. Dazu sind zwei Schritte notwendig. Im ersten Schritt muss innerhalb der Experten-/Endroutine ein Aufruf einer statischen Methode platziert werden und im zweiten Schritt dieser Aufruf konfiguriert werden.

**Hinweis:** Dadurch, dass die Konfiguration des Aufrufs von AGIMENDO.calc nicht innerhalb der Experten-/Endroutine stattfindet, muss bei einer Konfigurationsänderung nicht die Experten-/Endroutine neu aktiviert (und ggf. transportiert) werden

Welcher Aufruf zu nutzen ist, richtet sich nach dem Anwendungsfall von AGIMENDO.calc. AGIMENDO.calc kann beim Fortschreiben einerseits zum Abbilden eines Mappings verwendet werden (Änderung der Struktur, Expertenroutine) und andererseits zur Abbildung von Berechnungsvorschriften (innerhalb einer Struktur, Endroutine). Grundsätzlich können die Varianten auch vermischt werden, es bietet sich aber der Einfachheit halber an, die Schritte getrennt durchzuführen.

Der Aufruf innerhalb der Expertenroutine ist im folgenden Code-Stück dargestellt.

```
* call convenience method of AGIMENDO.calc to execute the tree-based
* calculation on the start and result package
/ISV/CL_PCE_CORE_TEXW=>TEXW_EXECUTE_TREE(
  EXPORTING
    iv_transformation = 'YOUR_UNIQUE_CALL_ID'
    IT_SOURCE_PACKAGE = SOURCE_PACKAGE
    IR_TRANS = me "optional"
    IR_LOG = log "optional"
  CHANGING
    ct_data_table = RESULT_PACKAGE
).
```

Der Aufruf innerhalb der Endroutine ist im folgenden Code-Stück dargestellt.

```
* call convenience method of AGIMENDO.calc to execute the tree-based
* calculation on the result package
/ISV/CL_PCE_CORE_TERW=>TERW_EXECUTE_TREE(
  EXPORTING
    iv_transformation = 'YOUR_UNIQUE_CALL_ID'
    IR_TRANS = me "optional"
  CHANGING
    ct_data_table = RESULT_PACKAGE
    ct_monitor = MONITOR
).
```

Der statische Parameter in dem dargestellten Aufruf (`IV_TRANSFORMATION`) stellt eine eindeutige ID dar, die die Referenz zur Konfigurationstabelle herstellt. Diese ID ist zeichenbasiert und frei wählbar. Ein Vorschlag zum Vorgehen der Wahl der ID ist, hier den technischen Namen der

Transformation (32 stellige GUID) zu verwenden um eine Rückkopplung aus der Konfiguration zur Transformation zu gewährleisten.

**Hinweis:** Der Parameter `IR_TRANS` wird in neueren Systemen (ab 7.4) benötigt, damit die Vergabe von `RECORD`-Nummern korrekt erfolgen kann. Falls dieser nicht übergeben wird, muss dies manuell im Coding nachgelagert vorgenommen werden.

**Hinweis:** Im Prinzip ist es möglich, innerhalb der Endroutine die eindeutige ID aus den bearbeiteten Daten dynamisch zu erzeugen. Damit kann bspw. anhand der Planungsrunde der zugehörige Rechenbaum geladen werden. In diesem Fall ist aber darauf zu achten, dass die Konfiguration korrekt vorbereitet ist.

Table: /ISV/PCE\_CT\_PROP  
Displayed Fields: 6 of 6 Fixed Columns: | 3 List Width 0250

	MANDT	TRANSFORMATION	PROP_ID	PROP_TYPE	NAME	VALUE
<input type="checkbox"/>	100	05COUNSCZ9IWFK1TL7HVXSUXH0WZ961	0000000001	TREE_ID		204
<input type="checkbox"/>	100	05COUNSCZ9IWFK1TL7HVXSUXH0WZ961	0000000004	SEL_BLOCK	0FISCPER	
<input type="checkbox"/>	100	05COUNSCZ9IWFK1TL7HVXSUXH0WZ961	0000000002	INIT_VAL	0FISCVARNT	K4

Abbildung 32: Screenshot - Konfigurationstabelle /ISV/PCE\_CT\_PROP

Die Konfiguration des Aufrufes erfolgt über eine transparente Tabelle (/ISV/PCE\_CT\_PROP). In dieser Tabelle müssen für jeden Aufruf (also jede in der Endroutine genutzte ID) Parameter hinterlegt werden. Die Tabelle hat die folgende Struktur:

- **TRANSFORMATION:** gewählte ID
- **PROP\_ID:** frei wählbare ID (um mehrere Werte für einen Parametertyp zu ermöglichen)
- **PROP\_TYPE:** Parametertyp, Konstanten (s.u.)
- **NAME:** Parametername
- **VALUE:** Parameterwert

In der folgenden Tabelle sind die möglichen Werte für die Spalte `PROP_TYPE` und die Beschreibung der Parametertypen zusammengestellt.

PROP_TYPE	Anzahl	Beschreibung
<b>TREE_ID</b>	1..1	Legt den durchzuführenden Rechenbaum fest. Die Spalte <code>NAME</code> muss leer sein, die Spalte <code>VALUE</code> muss die ID des gewählten Rechenbaumes enthalten, zu finden in der Verwaltung von AGIMENDO.calc
<b>INIT_VAL</b>	0..n	Für jeden Initialwert muss genau ein Eintrag für <code>INIT_VAL</code> eingefügt werden. Die Spalte <code>NAME</code> enthält hierbei den InfoObjekt-Namen, die Spalte <code>VALUE</code> den Initialwert für dieses InfoObjekt. Beispielsweise <code>0FISCVARNT</code> als InfoObjektname und <code>K4</code> als Initialwert.
<b>SELECTION</b>	0..n	Über diesen Parameter können einzelne globale Einschränkungen definiert werden. Analog zu <code>INIT_VAL</code> muss in der Spalte <code>NAME</code> InfoObjektname und in der Spalte <code>VALUE</code> der Wert gesetzt werden. Zu einem InfoObjektname können mehrere Einträge erzeugt werden. Dadurch werden dann analog zum Planungskontext mehrere Berechnungsdurchläufe erzeugt. Hierbei ist zu beachten, dass dabei alle Kombinationen durch das vollständige Kreuzprodukt gebildet werden und dadurch ggf. sehr viele Berechnungsdurchläufe entstehen können. Bei 3 Gesellschaften und 5 Jahre für 3 Buchungskreise ergeben sich bereits

		3x5x3=45 Durchläufe.
<b>SEL_BLOCK</b>	0..n	Über den Parameter SEL_BLOCK können zur Laufzeit globale Einschränkungen generiert werden. In der Spalte NAME wird ein InfoObjektname erwartet, die Spalte VALUE bleibt leer. Für das angegebene InfoObjekt werden alle in der Datentabelle vorhandenen Werte ermittelt und für jeden Wert ein Baumdurchlauf mit einer globalen Einschränkung erzeugt. Wenn beispielsweise in der Datentabelle Werte für 2009 und 2010 enthalten sind, und ein SEL_BLOCK Eintrag mit OFISCYEAR existiert, werden zwei Berechnungsdurchläufe für die beiden Jahre 2009 und 2010 durchgeführt. SEL_BLOCK kann mit SELECTION Einträgen kombiniert werden.
<b>IMPL_LOOP</b>	0..1	Ein Parameter vom Typ IMPL_LOOP und dem Wert ACTIVATED in der Spalte VALUE sorgt dafür, dass für jede Zeile des Result-Package ein Berechnungsdurchlauf von AGIMENDO.calc vorgenommen wird. Dazu wird aus jeder Zeile eine Liste von globalen Einschränkungen erzeugt. Allerdings steht in jedem Berechnungsdurchlauf trotzdem die gesamte Datentabelle zur Verfügung. Dieser Parameter sollte nicht im Zusammenspiel mit SELECTION und SEL_BLOCK verwendet werden um Dopplungen in den globalen Einschränkungen zu vermeiden. Dieser Modus kann mit Parametern vom Typ IM_EX_CHAR kombiniert werden.
<b>IM_EX_CHAR</b>	0..n	Ein Parameter IM_EX_CHAR kann nur in Kombination mit IMPL_LOOP verwendet werden. Ein Eintrag für IM_EX_CHAR (in der Spalte NAME den InfoObjektnamen, die Spalte VALUE leer) wird die Menge der globalen Einschränkungen, die durch IMPL_LOOP erzeugt wird um das jeweilige InfoObjekt reduziert.  Dadurch kann beispielsweise ein Konto von den globalen Parametern ausgeschlossen werden. Für das Beispiel aus Kapitel 2.1 könnte für die Tabelle 3 als Datentabelle die folgende Konfiguration sinnvoll sein: <i>IMPL_LOOP – ACTIVATED und IM_EX_CHAR – OMETYPE</i> Das würde bedeuten, dass die Berechnung für jede Gesellschaft und jedes Jahr einmal durchgeführt wird. Nicht aber für jeden Kennzahltyp.
<b>REF_DATA</b>	0..1	Ein Parameter vom Typ REF_DATA mit dem Wert ACTIVATED in der Spalte VALUE sorgt dafür, dass nach der Berechnung im RESULT_PACKAGE nur die Datensätze enthalten sind, die neu erzeugt wurden. Es werden also die Datensätze aus der Datentabelle nur als Referenzdaten verwendet.
<b>EXPL_LOOP</b>	0..1	Ein Parameter vom Typ EXPL_LOOP und dem Wert ACTIVATED in der Spalte VALUE sorgt dafür, dass für jede Zeile der der Quelle ein Berechnungsdurchlauf erzeugt wird. In der Datentabelle ist dann vor dem Ausführen des Rechenbaumes genau eine Zeile enthalten.
<b>SMART_TKC</b>	0..1	Ein Parameter vom Typ SMART_TKC und dem Wert ACTIVATED in der Spalte VALUE sorgt dafür, dass die Datentabelle auf der gearbeitet wird, eine interne Schlüsselstruktur erhält die optimiert ist auf die im Rechenbaum verwendeten Einschränkungen. Diese Option führt zu einer drastisch erhöhten Performance, wenn der Rechenbaum gleichartige Knoten enthält also strukturell mit gleichen Einschränkungen auf die Tabelle zugegriffen wird.

Tabelle 4: Parametertypen für die Konfiguration des Transformationsintegrationsmoduls

**Hinweis:** Beim Abbilden eines Mappings (oder einem anderen Aufruf in der Experten Routine) über die zugehörigen Mapping-Knotentypen muss die Konfigurationstabelle wie folgt aussehen. (Expliziter Loop / Blockung über ORECORD) Dadurch wird jede Zeile in der Quelle einzeln bearbeitet.

Data Browser: Table /ISV/PCE_CT_PROP 4 Hits					
Table: /ISV/PCE_CT_PROP					
Displayed Fields: 6 of 6 Fixed Columns: [ 3 ] List Width 0250					
MANDT	TRANSFORMATION	PROP_ID	PROP_TYPE	NAME	VALUE
<input type="checkbox"/>		0000000001	TREE_ID		
<input type="checkbox"/>		0000000005	EXPL_LOOP		ACTIVATED
<input type="checkbox"/>		0000000006	SEL_BLOCK	ORECORD	

Abbildung 33:Screenhsot - Konfiguration Mapping-Aufruf Expertenroutine

**Hinweis:** Aus technischen Gründen sollte bei der Verwendung von IMPL\_LOOP in Transformationen immer das fiktive InfoObjekt ORECORD, also das Feld RECORD von den globalen Einschränkungen ausgeschlossen werden. Dies geschieht durch die Konfiguration eines Parameters IM\_EX\_CHAR mit Wert ORECORD in der Spalte NAME.

## 8. Berechtigungen

Über das Berechtigungsobjekt /ISV/ACALC kann der Zugriff auf die Pflegeoberfläche gesteuert werden. In der Abbildung 34 sind die zur Verfügung stehenden und berücksichtigten Aktivitäten dargestellt.

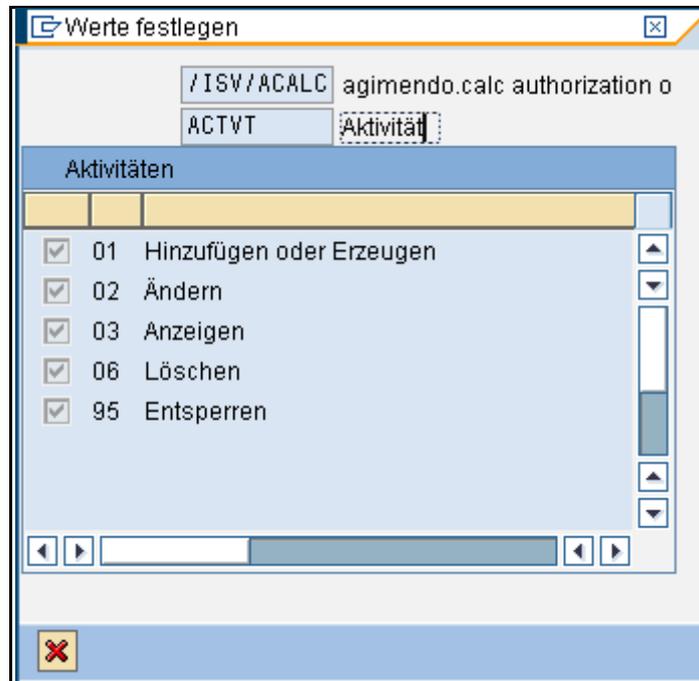


Abbildung 34: Aktivitäten des Berechtigungsobjekts /ISV/ACALC

Die Aktivität 01 wird beim Anlegen eines neuen Rechenbaumes geprüft, die Aktivität 02 bei Öffnen und beim Ändern des Titels eines Rechenbaumes. Die Aktivität 03 wird beim Öffnen der Pflegeanwendung überprüft. Beim Löschen eines Rechenbaumes wird die Aktivität 06 vorausgesetzt. Wenn ein Rechenbaum durch einen anderen Benutzer gesperrt ist, wird die Aktivität 95 überprüft um ein Öffnen und damit das Entsperren zu ermöglichen.

## 9. Gruppen Pflege

Mit der Produktversion 3.0 wurde ein Gruppensystem eingeführt. Mithilfe des Gruppensystems sollen Bäume Konsistent transportiert werden können.

Auf der Startseite der AGIMENDO.calc Konfigurationsoberfläche wurde dafür ein neuer Button angelegt.

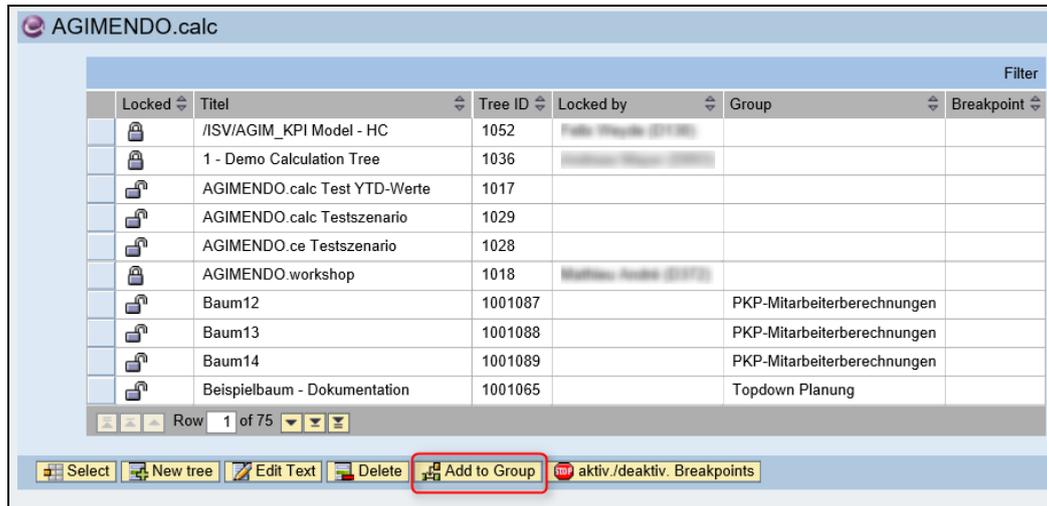


Abbildung 35: Add to Group Button

Mit dem Button „Zu Gruppe hinzufügen“ oder im Englischen „Add to Group“ besteht die Möglichkeit einen Button und alle enthaltenen Knoten zu einer Gruppe hinzuzufügen. Dabei können sie den Baum zu einer Bereits vorhandenen Gruppe hinzufügen oder eine neue Gruppe erzeugen. Nach Auswahl der Gruppe, werden die Verbindungen des Baums und aller enthaltener Knoten mit allen existierenden Knoten kontrolliert. Falls ein Knoten im Baum existiert, der auch in einem anderen Baum verwendet wird, so muss der Baum auch in die Gruppe eingefügt werden. Nach Auswahl der Gruppe, wird der Benutzer benachrichtigt, welche Bäume miteingefügt werden.

Zusätzlich wurde die Anwendung /ISV/PCE\_VERSION\_MANAGE erzeugt. Mithilfe dieser kann der Benutzer Gruppen löschen und somit die Bäume und Knoten wieder aus der Gruppe entlassen. Diese Anwendung bietet auch die Möglichkeit die Beschreibung und den Namen der Gruppe zu bearbeiten.

Erreichbar ist sie unter der URL:

[http://<server>:<port>/sap/bc/webdynpro/isv/pce\\_version\\_manage](http://<server>:<port>/sap/bc/webdynpro/isv/pce_version_manage)

## 10. Transportwesen

Mit der Produktversion 3.0 wurde ein neuer Transportanschluss eingeführt. Damit verfügt AGIMENDO.calc über die Möglichkeit, Gruppen von Bäumen separat zu transportieren. Der bisher empfohlene Weg, alle Tabellen in Summe zu transportieren ist damit nicht mehr der präferierte.

### 10.1 Transportmanager

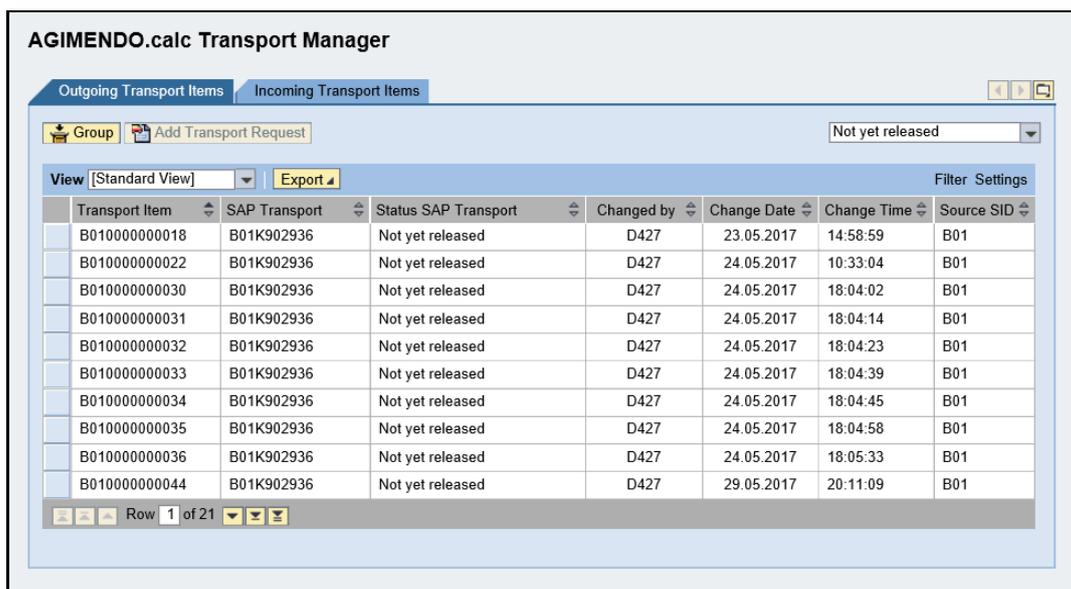
Der Transportmanager besteht als separate WebDynpro-Anwendung die auch separat berechtigt werden kann. Der Aufruf des Transportmanagers erfolgt über folgende URL

[http://<server>:<port>/sap/bc/webdynpro/isv/pce\\_transport\\_manager](http://<server>:<port>/sap/bc/webdynpro/isv/pce_transport_manager)

Die Oberflächen des Transportmanagers gliedern sich in die zwei Bereiche „Ausgehende Transportitems“ und „Eingehende Transportitems“.

### 10.1.1 Ausgehende Transportitems

Im Bereich der Ausgehenden Transportitems können die vorhandenen Transportitems eingesehen werden und neue Transportitems angelegt werden. In der folgenden Abbildung ist die Sicht der Ausgehenden Transporte dargestellt.



The screenshot shows the 'AGIMENDO.calc Transport Manager' window with the 'Outgoing Transport Items' tab selected. The interface includes a 'Group' button, an 'Add Transport Request' button, and a filter dropdown set to 'Not yet released'. Below this is a table with columns for Transport Item, SAP Transport, Status SAP Transport, Changed by, Change Date, Change Time, and Source SID. The table contains 11 rows of data, all with a status of 'Not yet released' and a source SID of 'B01'. At the bottom, there is a 'Row 1 of 21' indicator.

Transport Item	SAP Transport	Status SAP Transport	Changed by	Change Date	Change Time	Source SID
B010000000018	B01K902936	Not yet released	D427	23.05.2017	14:58:59	B01
B010000000022	B01K902936	Not yet released	D427	24.05.2017	10:33:04	B01
B010000000030	B01K902936	Not yet released	D427	24.05.2017	18:04:02	B01
B010000000031	B01K902936	Not yet released	D427	24.05.2017	18:04:14	B01
B010000000032	B01K902936	Not yet released	D427	24.05.2017	18:04:23	B01
B010000000033	B01K902936	Not yet released	D427	24.05.2017	18:04:39	B01
B010000000034	B01K902936	Not yet released	D427	24.05.2017	18:04:45	B01
B010000000035	B01K902936	Not yet released	D427	24.05.2017	18:04:58	B01
B010000000036	B01K902936	Not yet released	D427	24.05.2017	18:05:33	B01
B010000000044	B01K902936	Not yet released	D427	29.05.2017	20:11:09	B01

Abbildung34: Ausgehende Transporte im AGIMENDO.calc Transportmanager

In der Liste der Transportitems werden nur Transportitems aufgelistet, die in im aktuellen System erstellt wurden. Über die Auswahlbox oben rechts kann zwischen vier Sichten umgeschaltet werden.

- **Noch nicht freigegeben** – Listet alle Transportitems auf, die einem SAP Transportauftrag zugeordnet sind, der noch nicht freigegeben ist.
- **Nur manuelle Transportitems** – Listet alle Transportitems auf, die noch keinem SAP Transportauftrag zugeordnet sind.
- **Alle** – Listet alle vorhandenen Transportitems auf
- **Freigegeben** – Listet alle Transportitems auf, die einem SAP Transportauftrag zugeordnet sind, der bereitfreigegeben wurde.

Um ein neues Transportitem im Transportmanager anzulegen, klickt auf den Button  um eine oder mehrere Gruppen für den Transport auszuwählen.

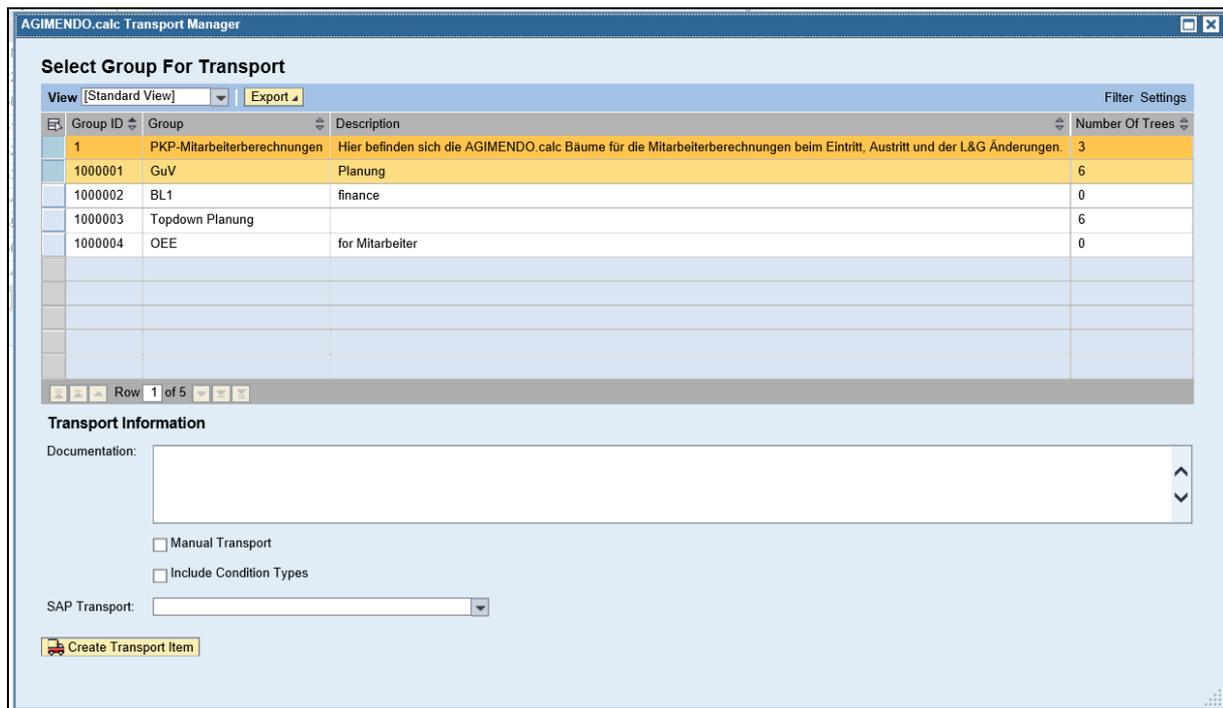


Abbildung 36: Gruppen Auswahl für den Transport

Die Dokumentation dient zur Information und kann als Freitext genutzt werden. Dieser Text wird im Transportmanager des Zielsystems wieder angezeigt.

Beim Transportieren kann neben der Auswahl der Gruppen und der Angabe des Transportauftrags zusätzlich definiert werden, ob neue Bedingungstypen mittransportiert werden sollen oder nicht. Dabei muss beachtet werden, dass neue Klassen, die für die Bedingungen angelegt worden sind, nicht mitübertragen werden.

Um selbst erzeugte UI Knoten zu transportieren, müssen folgende Tabellen (vollumfänglich, also \*) zum Transport hinzugefügt werden:

```

/ISV/PCE_UI_CONF
/ISV/PCE_UI_DYN
/ISV/PCE_UI_PRE
/ISV/PCE_UI_RES
/ISV/PCE_UI_TEXT

```

Wie auch bei den Bedingungstypen muss hier beachtet werden, dass neu erzeugte Klassen auf dem Zielsystem vorhanden ist.

Wenn für ein Transportitem beim Erstellen als „Manueller Transport“ markiert wurde, kann es nachträglich im Transport Manager einem SAP Transportauftrag zugewiesen werden. Dieser Vorgang ist in der Abbildung 34 dargestellt.

Während der Transportauftrag zugewiesen wird, kann die Dokumentation des Transportitems angepasst werden, bspw. kann eine Notiz eingefügt werden, warum dieser Auftrag gewählt wurde.

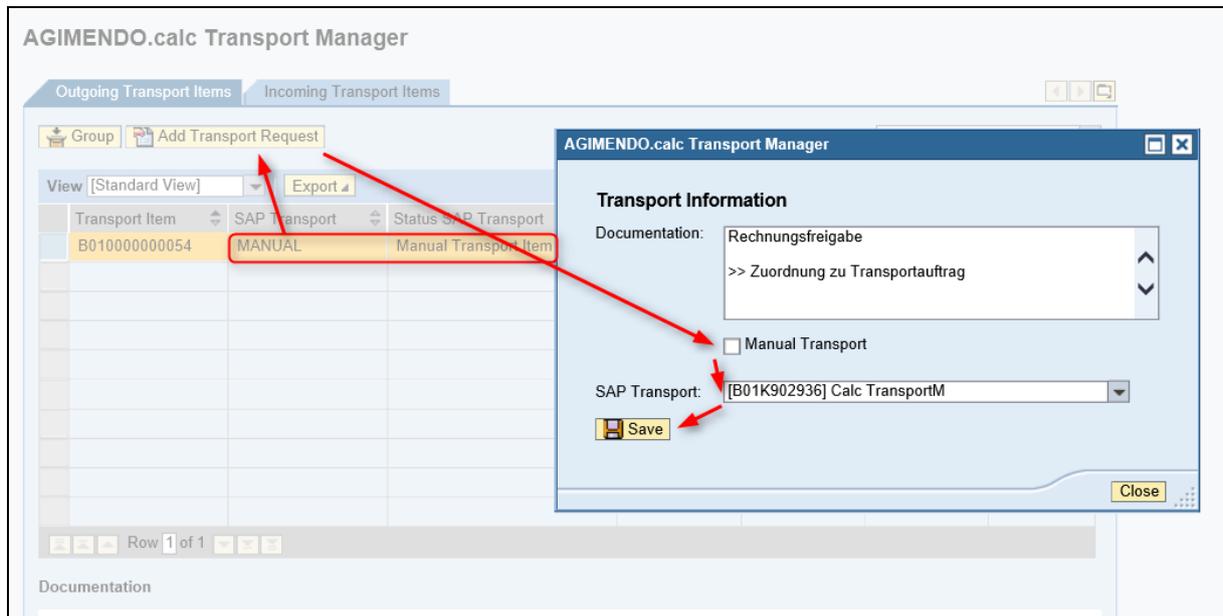


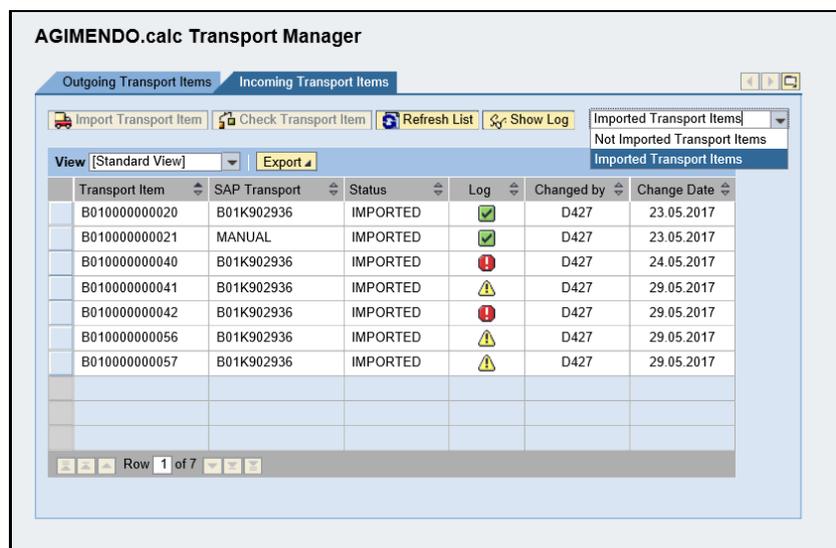
Abbildung 37: Transportmanager - Manueller Transport: Zuweisung eines Transportauftrags

### 10.1.2 Eingehende Transportitems

Auf der zweiten Registerkarte können die eingehenden Transportitems eingesehen werden. Dabei kann die Liste gefiltert werden auf entweder bereits importierten Transportitems oder alle noch nicht importierten Transportitems.

Ein ausgewähltes Transportitem kann entweder geprüft (  **Transportitem prüfen** ) oder geprüft und importiert (  **Transportitem importieren** ) werden. Zu einem bereits importierten Transportitem kann das Prüflog des Imports (  **Log anzeigen** ) eingesehen werden.

In der Tabelle wird im Feld „Log“ ein Icon angezeigt, das die schwerwiegendste Meldung des letzten Imports anzeigt.



The screenshot shows the 'AGIMENDO.calc Transport Manager' interface with the 'Incoming Transport Items' tab active. The table displays a list of transport items with columns for Transport Item, SAP Transport, Status, Log, Changed by, and Change Date. The 'Log' column contains icons representing the status of the last import: a green checkmark for successful imports and a red exclamation mark for failed imports. The 'Status' column shows 'IMPORTED' for all items.

Transport Item	SAP Transport	Status	Log	Changed by	Change Date
B01000000020	B01K902936	IMPORTED	✓	D427	23.05.2017
B01000000021	MANUAL	IMPORTED	✓	D427	23.05.2017
B01000000040	B01K902936	IMPORTED	!	D427	24.05.2017
B01000000041	B01K902936	IMPORTED	!	D427	29.05.2017
B01000000042	B01K902936	IMPORTED	!	D427	29.05.2017
B01000000056	B01K902936	IMPORTED	!	D427	29.05.2017
B01000000057	B01K902936	IMPORTED	!	D427	29.05.2017

Abbildung39: Transportmanager - Eingehende Transporte

### 10.1.3 Transportwesen als Backup

Mit einem kleinen Zusatzprogramm im Backend kann die Transportfunktionalität von AGIMENDO.calc dazu verwendet werden, über ein Transportitem von gepflegten Objekten ein Backup anzulegen und wieder einzuspielen. Dazu muss man zunächst ein Transportitem erstellen (mit dem Inhalt der gesichert und im selben System wieder importiert werden soll) und anschließend die Quellsystem-ID ändern damit das Transportitem in der Liste der Eingehenden Transportitems angezeigt wird. Dazu gibt es das Programm /ISV/CHANGE\_SID\_OF\_TRANSPORT. Dieses kann über die SE38 im Backend ausgeführt werden.

In Abbildung40 und in Abbildung41 ist dargestellt wie das Programm in der SE38 aufgerufen und ausgeführt werden kann. Beim Programmlauf können drei Optionen angegeben werden.

- Das Transportitem, das geändert werden soll
- Das zu schreibende Quellsystem (Idealerweise wird hier ein Quellsystem angegeben, das nicht wirklich existiert, es wird damit nur markiert, dass es sich nicht um das aktuelle System handelt.)

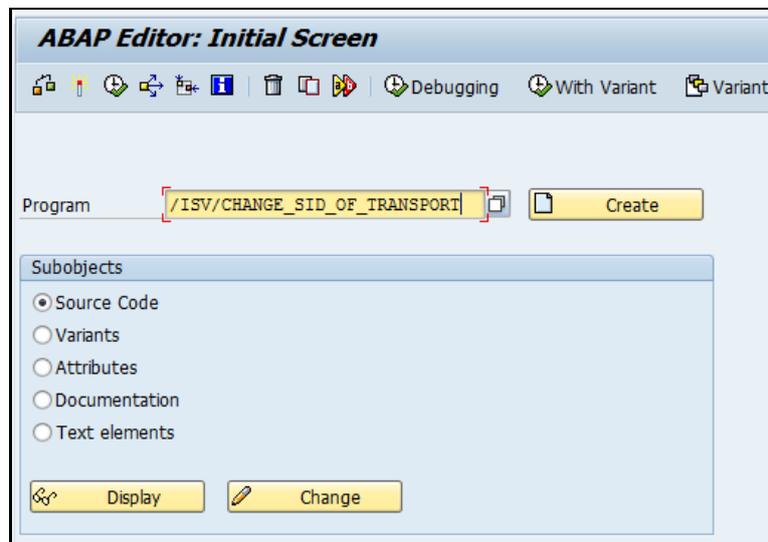


Abbildung40: Transportmanager - Programm um Reimport vorzubereiten

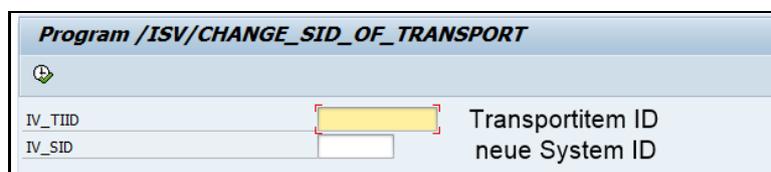


Abbildung41: Transportmanager - Optionen für Reimportvorbereitung

Nachdem die Quellsystem-ID mit dem genannten Programm geändert wurde, wird das Transportitem in der Liste der eingehenden Transporte angezeigt und kann normal importiert werden.

## 10.2 Altes Transportvorgehen / vollständiger Transport

Generell sollen Änderungen an den Rechenregeln nur im Entwicklungssystem durchgeführt und nach erfolgreichem Test ins Produktivsystem transportiert werden. Der Transport muss hierbei die folgenden Tabellen (vollumfänglich, also \*) beinhalten:

/ISV/PCE\_COND  
/ISV/PCE\_COND\_VA  
/ISV/PCE\_CT  
/ISV/PCE\_CT\_NODE  
/ISV/PCE\_CT\_PROP  
/ISV/PCE\_CT\_RELA  
/ISV/PCE\_CT\_TAG  
/ISV/PCE\_C\_RESTR  
/ISV/PCE\_DESCR  
/ISV/PCE\_IDS  
/ISV/PCE\_SETTING  
/ISV/PCE\_UI\_CONF  
/ISV/PCE\_UI\_DYN  
/ISV/PCE\_UI\_PRE  
/ISV/PCE\_UI\_RES  
/ISV/PCE\_UI\_TEXT  
/ISV/PCE\_VHANDLE

**Hinweis:** Bei einem Teiltransport muss vor einem Transport aus diesen Tabellen im Zielsystem alle relevanten Inhalte gelöscht werden („\*-Transport“), da ansonsten Fehler im Zusammenspiel mit den vorhandenen Daten entstehen. Ein Teil-Transport der Rechenbäume ist nur in extremen Einzelfällen sinnvoll und praktikabel.

**Hinweis:** Die Tabellen /ISV/PCE\_CTN\_TAG, /ISV/PCE\_UI\_TYPE werden aktuell nicht genutzt.

## 11. Anhang

### 11.1 Änderungshistorie

Version	Autor	Änderung	Datum
1.3	Felix Weyde	Erstellung und Zusammenführung	11.01.2011
1.3.2	Felix Weyde	Änderungen nach Review	31.01.2011
1.3.3	Felix Weyde	Korrekturen / Rechtschreibfehler	12.02.2011
1.4	Felix Weyde	Überarbeitung / neue Funktionalitäten	17.12.2015
1.4.1	Felix Weyde	Korrektur Aufruf Mapping Expertenroutine	26.01.2016

### 11.2 Verzeichnisse

#### 11.2.1 Abbildungsverzeichnis

Abbildung 1: Schematischer Baumdurchlauf (Rekursivität).....	4
Abbildung 2: Screenshot Pflege eines exemplarischen Rechenbaums einer Ergebnisrechnung.....	5
Abbildung 3: Definition Einschränkungen Herstellungskosten .....	7
Abbildung 4: Screenshot - Fehlermeldung bei Überschneidung der Einschränkungen.....	8
Abbildung 5: Definition Schreibender Knoten "Aufwände" .....	9
Abbildung 6: Screenshot - Pflege Bedingungen am Knoten.....	11
Abbildung 7 Screenshot - (KYF.) Bedingung .....	12
Abbildung 8 Screenshot - UserExit Bedingung .....	12
Abbildung 9:Screenshot – Baumauswahl.....	13
Abbildung 10: Screenshot - Übersicht Pflegeoberfläche von AGIMENDO.calc.....	14
Abbildung 11: Screenshot - Kontextmenü Baumpflege .....	16
Abbildung 12: Screenshot - Drag and Drop.....	16
Abbildung 13: Screenshot - Pflege der Knotenbeziehungen.....	17
Abbildung 14: Screenshot - Ausschnitt Berechnung Erträge ohne Zwischenknoten.....	18
Abbildung 15: Screenshot - Ausschnitt Berechnung mit korrekter Klammerung .....	18
Abbildung 16:Screenshot - Knoteneigenschaften Festwertknoten .....	19
Abbildung 17:Screenshot - Knoteneigenschaften Temporärer Knoten .....	19
Abbildung 18: Screenshot - Werte ändern und Einschränkungen kopieren.....	19
Abbildung 19: Screenshot - Knoteneigenschaften Wertknoten .....	20
Abbildung 20: Screenshot - Hilfetext zur Erweiterten Validierung mit Toleranz .....	21
Abbildung 21:Screenshot - Knoteneigenschaften UserExit-Knoten.....	22
Abbildung 22 Screenshot - Value Saving Knoten .....	23
Abbildung 23 Screenshot - Variable Reference Content Knoten .....	23
Abbildung 24: Beispiel – Testwertberechnung .....	24
Abbildung 25:Screenshot - Ergebnis Testwertberechnung.....	24
Abbildung 26: Screenshot - Einbindung der angelegten Klasse in die Berechnungslogik.....	28
Abbildung 27: Screenshot - Rückfrage bei Löschen eines Knoten .....	33
Abbildung 28: Screenshot – Housekeeping Anwendung für Knoten .....	33
Abbildung 29: Screenshot – Knotendetails .....	34
Abbildung 30: Screenshot - Transaktion RSPLAN.....	35

Abbildung 31: Screenshot - Neuanlage einer Planungsfunktion.....	36
Abbildung 32: Screenshot - Konfigurationstabelle /ISV/PCE_CT_PROP .....	38
Abbildung 33:Screenhsot - Konfiguration Mapping-Aufruf Expertenroutine .....	40
Abbildung 34:Aktivitäten des Berechtigungsobjekts /ISV/ACALC.....	41
Abbildung 35: Add to Group Button.....	42
Abbildung 36: Gruppen Auswahl für den Transport .....	44
Abbildung 37: Transportmanager - Manueller Transport: Zuweisung eines Transportauftrags .....	45

### 11.2.2 Tabellenverzeichnis

Tabelle 1: Datentabelle für Beispiel Ergebnisrechnung .....	5
Tabelle 2: Ergebnis Datentabelle nach Ergebnisrechnung .....	6
Tabelle 3: Datentabelle mit Beispieldaten für mehr als eine Gesellschaft .....	8
Tabelle 4: Parametertypen für die Konfiguration des Transformationsintegrationsmoduls.....	39

### Index

Ausführungsplattform .....	3	<b>Einschränkung</b> .....	6	Knotentyp.....	23
Bedingungen .....	10	explizite Einschränkung.....	6	Kreuzprodukt .....	34
Bedingungstyp		globale Einschränkung .....	8	Mappingregeln .....	22
„nicht-initial“ Überprüfung... ..	11	Endroutine .....	35	Operator .....	11
eigenen .....	29	Expertenroutine.....	35	Planungsfunktion .....	33
Gleichheitsprüfung .....	11	Integrationsmodulen .....	3	Planungsfunktionstyp.....	33
Ungleichheitsüberprüfung.....	11	Knoten		Referenzdaten .....	37
Zwischenprüfung .....	11	BusinessContent .....	10	Transformation .....	35
Bedingungstypen.....	11	Festwertknoten.....	10	Transport.....	39
Berechtigungsobjekt .....	38	lesender .....	8	Validierung .....	19
BusinessContent.....	19, 29	schreibender .....	8	Verwendungsnachweis .....	30
<b>Datentabelle</b> .....	4	tempärer .....	10	Wert-Knoten .....	10
Duplicate dynamic key .....	8	User-Exit.....	10		