VARCHART XTree

ActiveX Edition 5.2
Benutzer- und Referenzhandbuch

NETRONIC

VARCHART XTree ActiveX Edition

Version 5.2

Benutzerhandbuch

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

© Copyright 2020 NETRONIC Software GmbH
Alle Rechte vorbehalten.

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

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

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

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

Bearbeitungsstand: 27 April 2020

Inhaltsverzeichnis

1 Einleitung

1.1 Allgemeines zu VARCHART XTree

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

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

Mit VARCHART XTree können Sie logische Daten in Form von Baum-Diagrammen visualisieren, editieren, exportieren und drucken. VARCHART XTree bietet sich für die Darstellung von hierarchischen Strukturen, wie z.B. Organigrammen, Dateisystemen oder Gliederungen allgemein an. Zusätzlich zur herkömmlichen Baumdarstellung kann auch eine dem Verzeichnisbaum im Microsoft Explorer ähnliche Darstellung angezeigt werden (TreeView-Stil).

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

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

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

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

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

> Leistungsmerkmale

- Knoten können durch die Zuweisung verschiedener Knotenaussehen unterschiedlicher Priorität frei gestaltet werden.
- Grafische Attribute können datengesteuert über Filter zugeordnet werden. (Z. B. können alle Knoten einer bestimmten Abteilung in Gelb dargestellt werden.)
- Über sog. Knotenformate kann die grafische Gestaltung der Knoten mit den darin enthaltenen Daten sehr flexibel festgelegt werden.
- Benutzerinteraktionen wie das Anlegen, Löschen und Verschieben von Knoten sind verfügbar.
- Knoten können im TreeView-Stil angeordnet werden. Dabei werden vertikale Ebenen wie in TreeView-Controls mit Plus- oder Minus-Zeichen dargestellt. Ein Mausklick auf eins der beiden Zeichen überführt den daneben stehenden Knoten in den jeweils anderen Kollabierzustand.
- Bei vertikaler Anordnung können Sie die Gesamthöhe eines Baum-Diagramms (in Zeilen) begrenzen.
- Sie können die Baum-Diagramme durch die Kombination horizontal und vertikal angeordneter Teilstrukturen optimieren. Dabei können Sie die Ebene angeben, ab der die Knoten vertikal angeordnet werden sollen.
- Baumstrukturen lassen sich kollabieren und expandieren.
- Sie können festlegen, ob die Baumstruktur über einen Strukturcode oder über die ID des Vaterknotens festgelegt werden soll.
- Das OLE Drag & Drop-Verhalten des VARCHART XTree ist kompatibel zu dem in Visual Basic üblichen, d. h. die Methoden, Eigenschaften und Ereignisse tragen dieselben Namen und haben dieselbe Bedeutung wie bei den Standardobjekten aus Visual Basic. Mit dem OLE Drag & Drop-Modus können Knoten oder Teilbäume verschoben oder kopiert werden.
- Sie können Ihre Diagramme stufenlos zoomen. Beliebige Ausschnitte Ihres Diagramms können Sie interaktiv bildschirmfüllend darstellen lassen. Mit Hilfe der Bildlaufleisten können Sie dann das Fenster wie eine Lupe über der Darstellung verschieben und so auch die anderen Bereiche der Darstellung in derselben Vergrößerung betrachten.
- Wenn Sie einen Knoten über den Rand des Steuerelements verschieben, wird die Darstellung automatisch nachgescrollt (Autoscrolling).
- Ihre Darstellungen können mit Titel und Legende gedruckt oder exportiert werden (Formate: VMF, WMF, JPG, BMP, EPS, GIF, PCX,

PNG, TIF). (Zum Format VMF siehe Kapitel "Wichtige Begriffe: Viewer Metafile (*.vmf)".)

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

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

1.2 Technische Voraussetzungen

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

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

1.3 Installation

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

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

- c:\windows\system32\regsvr32 "c\:program files\varchart\xtree\vctree.ocx"

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

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

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

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

- c:\windows\system32\regsvr32 -u "c\:program files\varchart\xtree\-vctree.ocx"

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

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

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

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

1.4 Auslieferung

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

VARCHART-XTree-Dateien:

- *vctree.ocx* (Version 5.0)
- *vctreed.dll*
- *vcpane32u.dll* (Version 5.0)
- *vcprct32u.dll* (Version 5.0)
- *vcwin32u.dll* (Version 5.0)
- *vxcsv32u.dll* (Version 1.320)

Microsoft Bibliotheken:

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

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

Die Installation der Bibliotheken *mfc100u.dll, msvcp100.dll, mfcm100u.dll* und *msvcr100.dll* kann entweder direkt durch Kopieren in das Windows-Systemverzeichnis oder über die im Unterverzeichnis **redist** mitgelieferte Setup-Datei *vcredist_vs2010_x86.exe* erfolgen.

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

- *vctree.lic* (enhält Ihre Entwicklerlizenz)
- *vctree.chm* (Online-Hilfe für Entwickler)

1.5 Datenaustausch mit VARCHART XTree

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

1.5.1 Schnittstellendefinition

Standardmäßig stehen 20 Datenfelder für die Knoten zur Verfügung (Maindata-Tabelle). Sie können auf der Eigenschaftenseite **Datendefinition** neue Datenfelder anlegen, indem Sie das Feld "Neu..." am Ende der Liste editieren. Sobald Sie dieses Feld nach dem Editieren verlassen, wird ein neues Feld angelegt.

Auf der Eigenschaftenseite **Datendefinition** können Sie diese Felder mit beliebigen Namen versehen und den gewünschten Datentyp festlegen (Alphanumerisch, Integer, Datum/Zeit). Bei Datumsfeldern müssen Sie auch noch das verwendete Datumsformat angeben (z.B. DD.MM.YY). Die Datentypen sollten Sie später möglichst nicht mehr verändern, da Formate und Knoten danach ggf. auf falschen Datentypen basieren. Dadurch könnte ein fehlerhaftes Verhalten entstehen.

1.5.2 Aufbau von CSV-Dateien

Geben Sie für die Knoten pro Zeile einen Datensatz ein und trennen Sie die Datenfelder jeweils durch ein Semikolon.

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

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

Code-Beispiel

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

1.5.3 CSV-Dateien verwenden

Die gewünschte Datei können Sie mit der Methode **Open** öffnen und mit der Methode **SaveAsEx** speichern. Wird bei **SaveAs** kein Name angegeben, so

wird der beim vorherigen Anwenden der Methode **Open** angegebene Name verwendet .

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

Code-Beispiel

```
VcTree1.Open "c:\daten\beispiel1.tre"
...
VcTree1.SaveAs ""
' oder
VcTree1.SaveAs "c:\daten\beispiel2.tre"
```

1.5.4 Knotendaten per API an VARCHART XTree übergeben

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

Code-Beispiel

```
Dim data As String
data = "1;1.;;;SWDevelopment;A;;GroupA;6;0;100;03.11.00;10.11.00;;0;;"
VcTree1.InsertNodeRecord data
VcTree1.EndLoading
```

1.5.5 Knotendaten aus VARCHART XTree ausfragen

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

Code-Beispiel

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

Set nodeCltn = VcTree1.NodeCollection
nodeCltn.SelectNodes vcAll
Set node = nodeCltn.FirstNode
Do Until node Is Nothing
   '
   ' Zugriff auf Feld 0 eines jeden Knotens
   '
```

```
   value = node.DataField(0)
   '
   ' Zugriff auf alle Daten
   '
   value = node.AllData
   Set node = nodeCltn.NextNode
Loop
```

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

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

Visual Studio 6.0:

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

Visual Studio 7.0:

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

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

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

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

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

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

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

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

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

Code-Beispiel

```
#include "xtree.h"
using namespace XTree;
```

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

Code-Beispiel

```
XTree::CVcTitle title = VcTree1.GetTitle();
```

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

Objekts an jenes anbinden können. Dabei ist das **Detach()** am Ende der Bearbeitung des Objekts nicht zu vergessen.

Sollten Sie bereits Projekte mit den generierten Dateien begonnen haben, sollte ein Umstieg nicht schwerfallen, weil die NETRONIC Software GmbH die vom Visual Studio 6.0 generierten Dateien als Grundlage verwendet, die daher kompatibel sind. Der einzige Unterschied ist die Verwendung von Namespaces, um die Namen von Unterobjekten eindeutig zu machen.

1.7 VARCHART ActiveX in HTML-Seiten

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

1.7.1 Einschränkungen

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

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

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

1.7.2 Implementierung mit direkter Einbettung

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

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

Code-Beispiel

```
<OBJECT ID="VcTree1" WIDTH=700 HEIGHT=350
 CLASSID="CLSID:CA393F28-3DE9-11D3-B22E-0080AD0058C7"
 CODEBASE="vctree.cab#version=4,000,0,0">
</OBJECT>
```

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

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

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

Code-Beispiel

```
VcTree1.ConfigurationName =
"http://www.netronic_test.com/xtree_sample.ini"
```

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

Eine kleine Webapplikation wird mit den Programmbeispielen ausgeliefert.

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

Code-Beispiel

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

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

1.7.3 Implementierung mit indirekter Einbettung

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

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

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

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

1.7.4 Problembehandlung

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

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

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

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

OBJECT Tag which specifies component FileVersion and #Version

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

How To Implement IObjectSafety in Visual Basic 6.0 Controls

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

Mozilla ActiveX Project

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

MeadCo Neptune

www.meadroid.com/neptune

Microsoft Cabinet SDK

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

Microsoft IExpress

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

Code Download Log Viewer (CDLLOGVW)

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

Microsoft Script Debugger

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

Code signing

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

Certification authorities

VeriSign: www.verisign.com/developer

Thawte: www.thawte.com

GeoTrust: www.trustcenter.de

GlobalSign: www.globalsign.net

Signcode tool

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

Signtool tool

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

1.8 Unterstützung und Beratung

Sie wissen nicht, ob VARCHART XTree die speziellen Anforderungen Ihres Baumdiagramms erfüllt?

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

Sie machen gerade den Anfangstest mit VARCHART XTree und möchten eine ganz spezifische Anforderung Ihres Baumdiagramms programmieren?

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

NETRONIC Software GmbH

Pascalstraße 15

52076 Aachen

Deutschland

Tel.: +49-2408-141-0

Fax: +49-2408-141-33

E-Mail: support@netronic.de

www.netronic.de

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

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

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

2 Tutorium

2.1 Überblick

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

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

- **Eigenschaftenseiten und Dialogfelder**

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

- **Elemente der Benutzerschnittstelle**

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

- **API-Referenz**

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

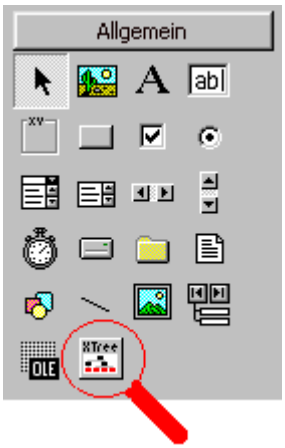
Als Entwicklungsumgebung für die Beispiele verwenden wir Visual Basic 6.0.

2.2 VARCHART XTree zur Werkzeugsammlung hinzufügen

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

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

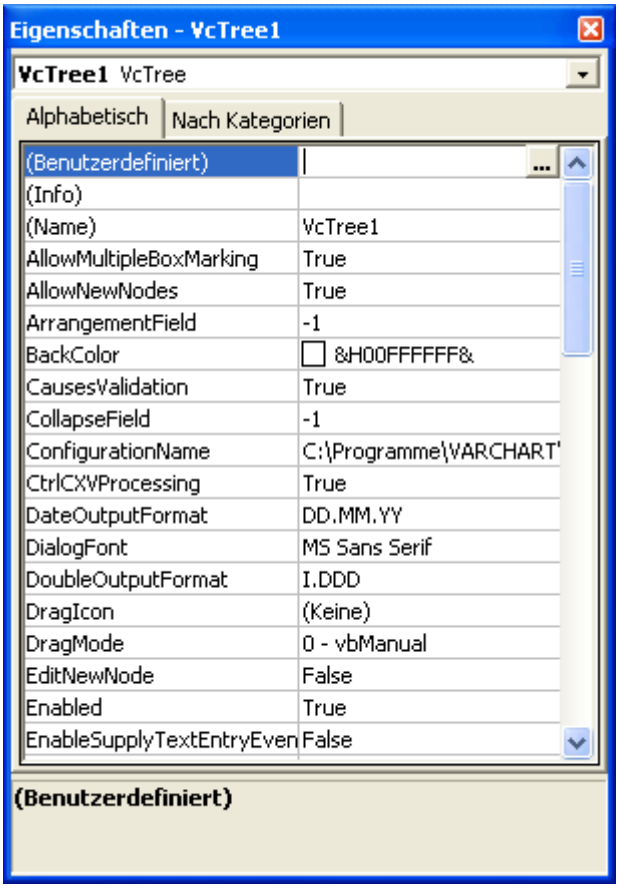
Nach erfolgreicher Aufnahme des VARCHART-XTree-Steuerelements in die Werkzeugsammlung erscheint seine Ikone in der Werkzeugsammlung:



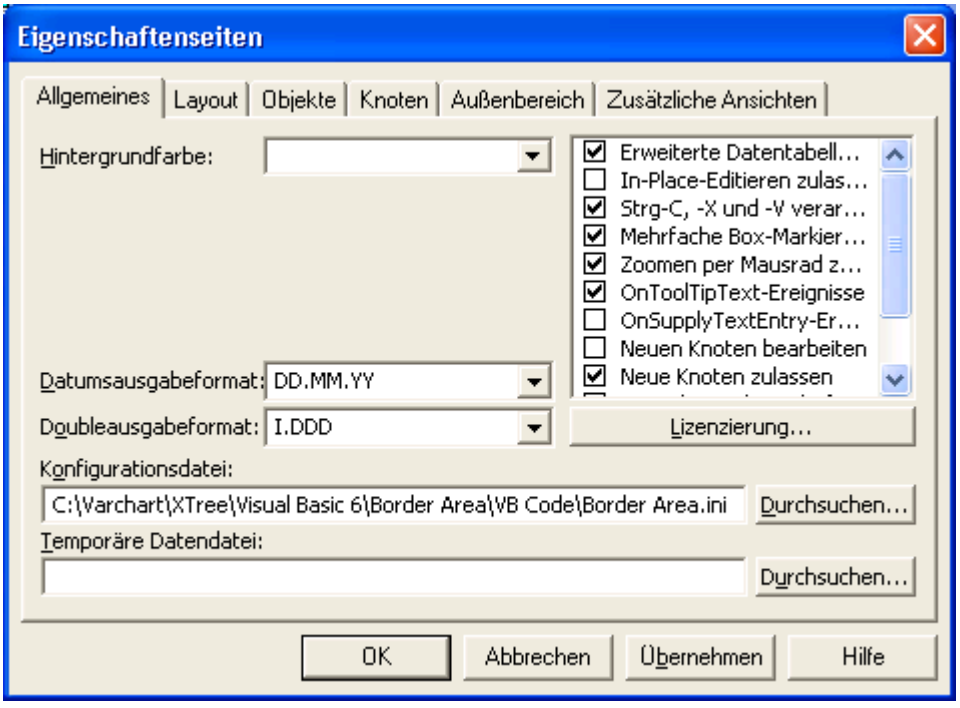
2.3 VARCHART XTree in einem Formular plazieren

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

Im Eigenschaftenfenster für VARCHART XTree können Sie mit Hilfe des Eintrags **Benutzerdefiniert** die VARCHART-XTree-Eigenschaftenseiten aktivieren.



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



Hinweis: Das eingefügte Steuerelement heißt hier und in allen Code-Beispielen **VcTree1**.

2.4 VARCHART XTree automatisch skalieren

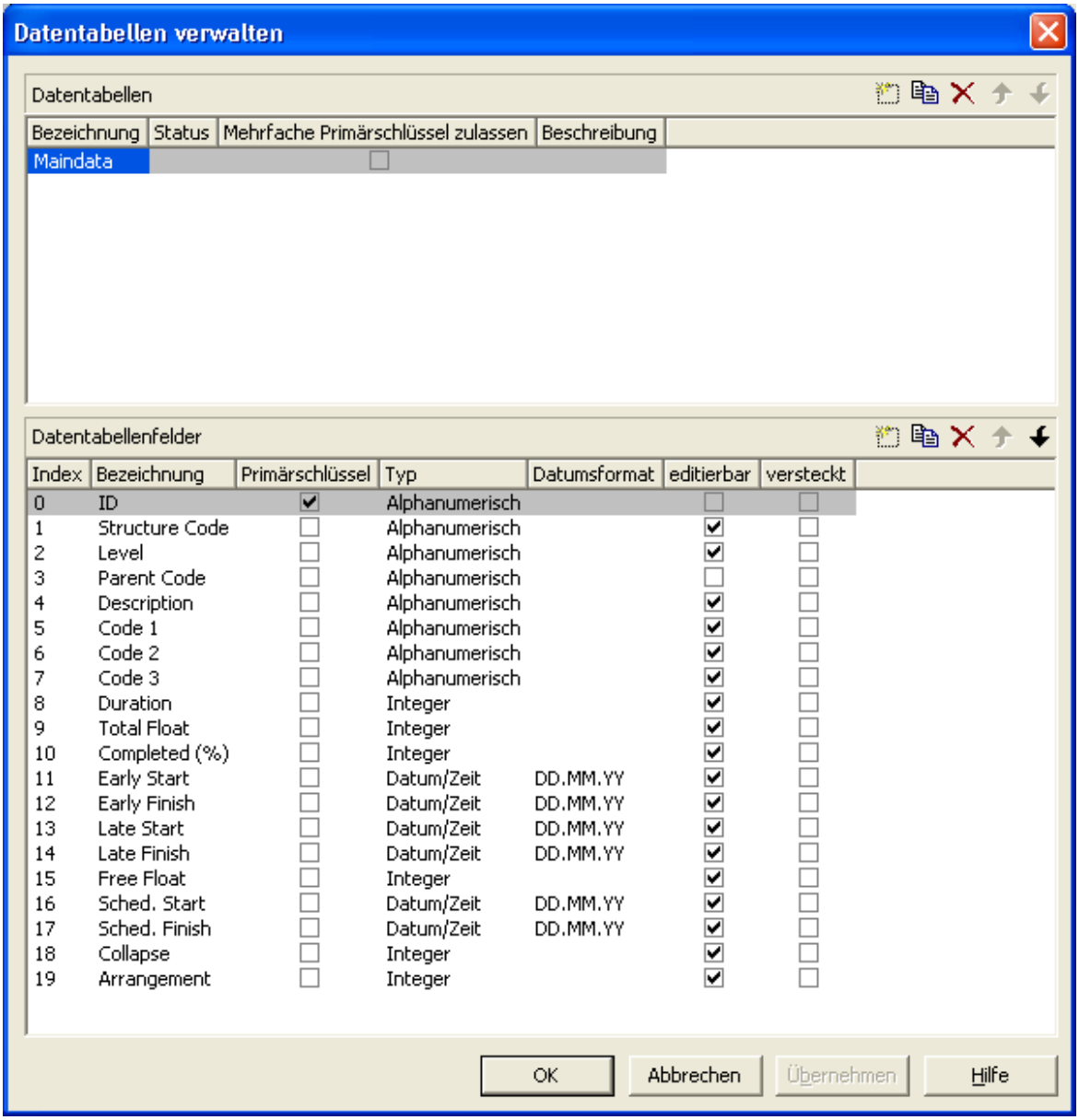
Wenn der rechte und der untere Rand des VARCHART-XTree-Steuerelement zur Laufzeit immer auf die Größe des gesamten Fensters angepasst werden sollen, können Sie das z. B. mit Hilfe des folgenden Codes erreichen:

Code-Beispiel

```
Private Sub Form_Resize()
   If ScaleWidth - VcTree1.Left > 0 And _
               ScaleHeight - VcTree1.Top > 0 Then
      VcTree1.Width = ScaleWidth - VcTree1.Left
      VcTree1.Height = ScaleHeight - VcTree1.Top
   End If
End Sub
```

2.5 Schnittstelle einrichten

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



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

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

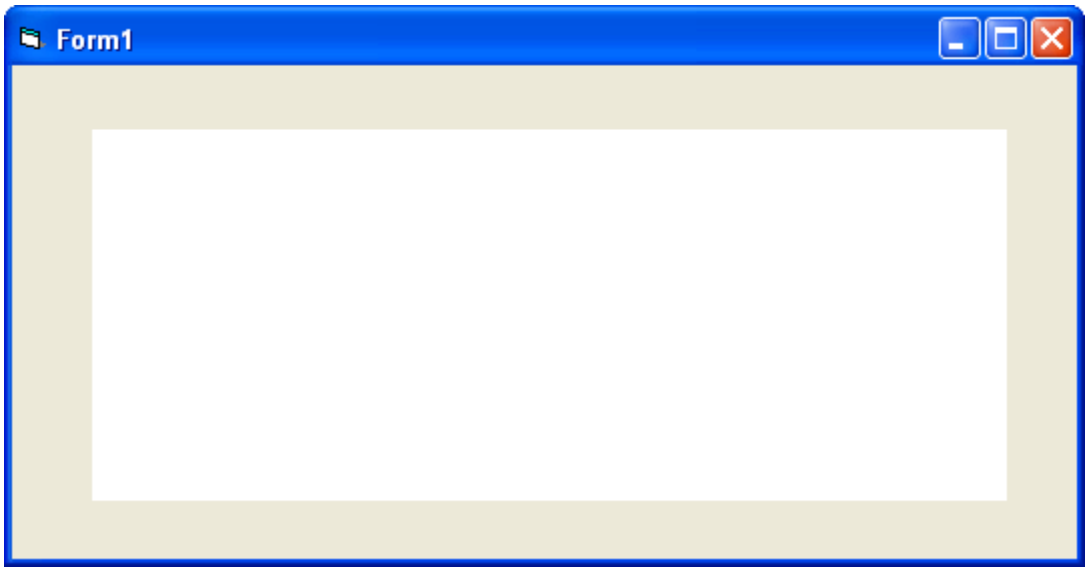
werden, benennen Sie das Feld bitte in "Nummer" um und wählen den Datentyp "Integer". Die ID soll nicht editerbar sein, damit sie im Standarddialog **Vorgänge bearbeiten** nicht überschrieben werden kann.

Felder der Maindata-Tabelle:

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

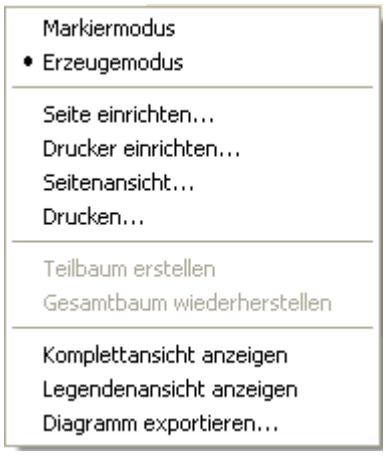
2.6 Der erste Lauf

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

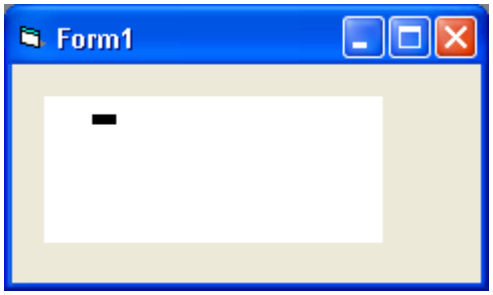


> Knoten anlegen

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

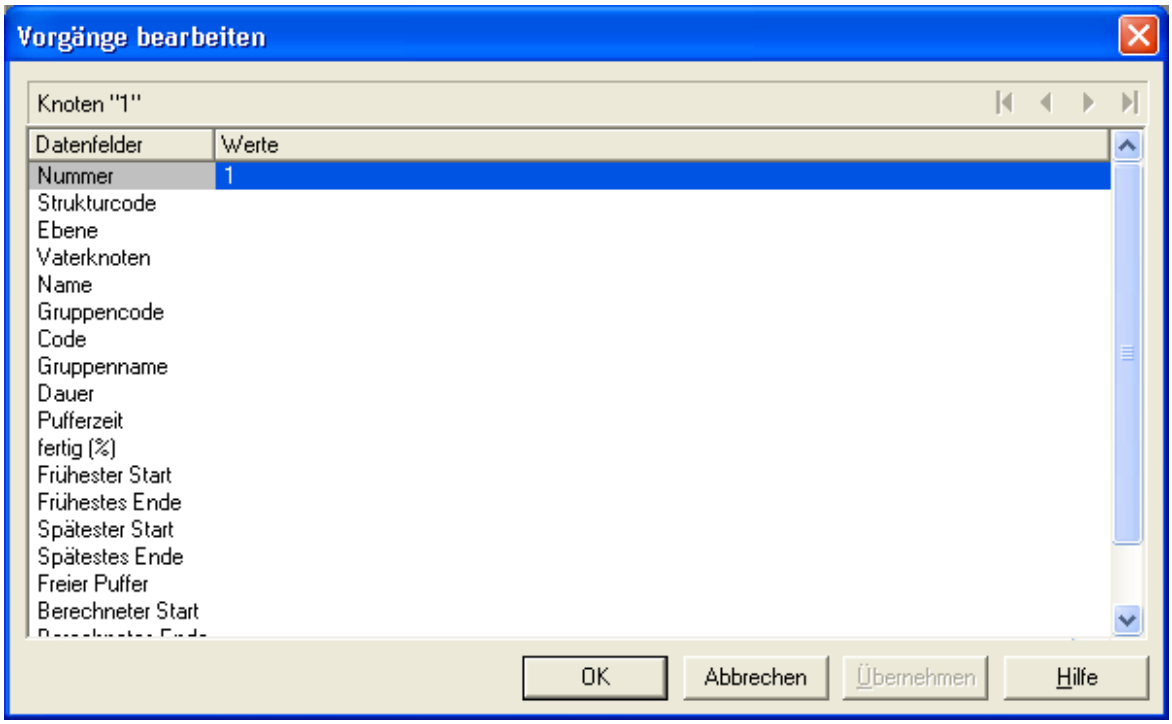


Im Erzeugemodus wird der Mauszeiger im leeren Diagrammbereich zu einem Knotenphantom in der Form eines kleinen schwarzen Rechtecks.

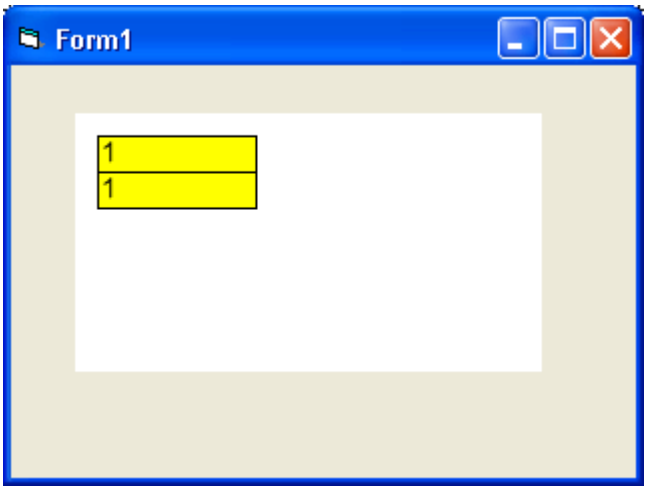


Klicken Sie nun einmal mit der linken Maustaste.

Wenn auf der Eigenschaftenseite **Allgemeines** die Option **Neuen Knoten bearbeiten** aktiviert ist, erscheint zunächst das Dialogfeld **Vorgänge bearbeiten**, in dem die Daten dieses Knotens angezeigt werden.

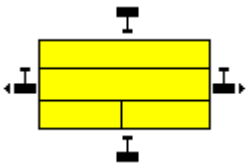


Zunächst ist nur das Datenfeld "Nummer" vorbesetzt (beim ersten Knoten mit dem Wert "1"). Sie können ggf. weitere Daten ergänzen, z.B. die Termindaten und eine Beschreibung. Sobald Sie die Daten mit **OK** bestätigen, wird der erste Knoten erzeugt.



War auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten bearbeiten** nicht aktiviert, wird der erste Knoten angelegt, sobald Sie im Erzeugemodus mit der linken Maustaste in den Diagrammbereich klicken. Das Dialogfeld **Vorgänge bearbeiten** erscheint dann nicht.

Weitere Knoten können Sie durch Anlagern an einen bereits vorhandenen Knoten erzeugen. Wenn Sie den Cursor im Erzeugemodus in die Nähe eines Knotens führen, verändert der Cursor seine Form und zeigt an, wo der neue Knoten angelegt würde (als Vater, Sohn oder linker bzw. rechter Bruder des Bezugsknotens).



> Knoten bearbeiten

Einen Knoten bearbeiten können Sie, indem Sie im Markiermodus mit einem Doppelklick auf den Knoten das Dialogfeld **Vorgänge bearbeiten** öffnen. Sie finden hier die Datenfelder wieder, die Sie im Dialog **Datentabellen verwalten** vereinbart haben. (Die Datenfelder, die dort als **versteckt** definiert worden sind, erscheinen nicht in diesem Dialogfeld. Die Datenfelder, die dort als **nicht editierbar** definiert worden sind, können in diesem Dialogfeld nicht bearbeitet werden.)

> Zurück zum Design-Modus

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

2.7 Daten aus einer Datei einlesen

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

Dazu reagieren Sie auf das Ereignis Form_Load:

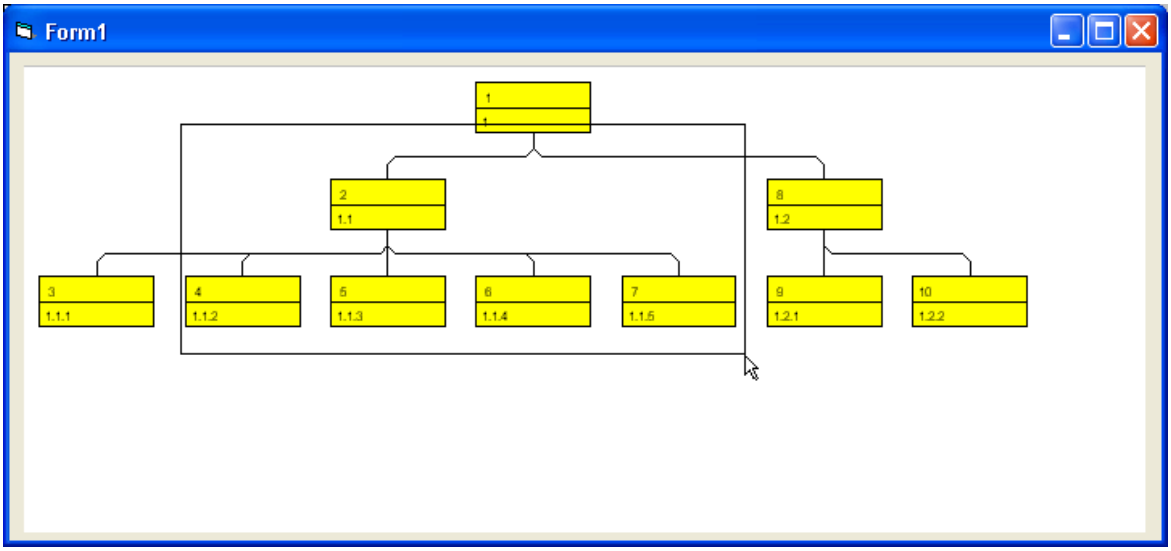
Code-Beispiel

```
Private Sub Form_Load()
    VcTree1.Open "C:\Programme\Varchart\xtree\tutorial.tre"
End Sub
```

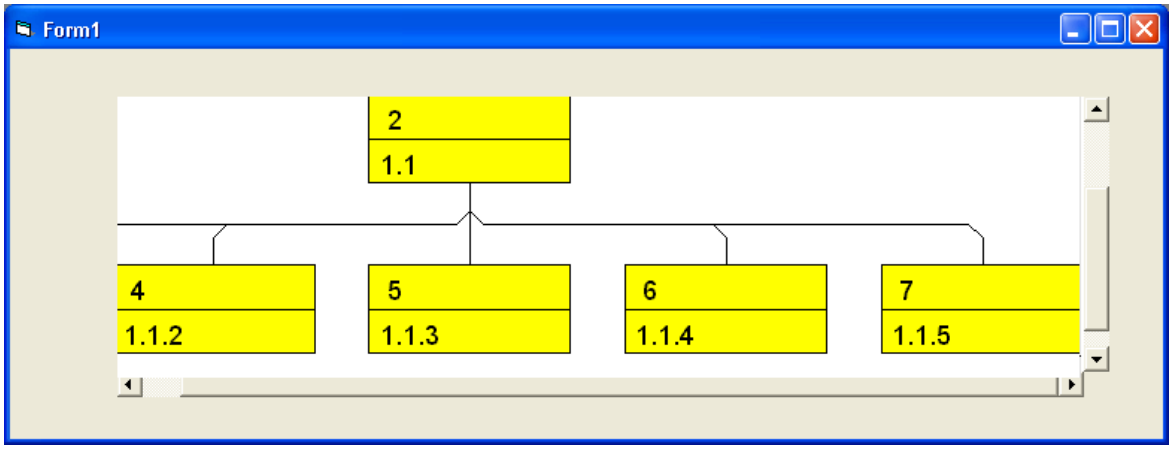
Die Pfadangabe ist abhängig von Ihrer Installation. Speichern Sie nun das Projekt. Wenn Sie nun das Programm starten, werden die Knoten des Projektes angezeigt.

VARCHART XTree stellt ein Baum-Diagramm komplett dar.

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



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



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

Code-Beispiel

```
Private Sub Form_Load()
   VcTree1.Zoomfactor = 100
End Sub
```

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

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

2.8 Markierung von Knoten festlegen

Auf der Eigenschaftenseite **Knoten** können Sie aus der Kombobox **Markierungstyp** auswählen, in welcher Weise Knoten markiert werden sollen.

Starten Sie das Programm und legen Sie im Erzeugemodus einige Knoten an. Klicken Sie nun im Markiermodus einmal auf einen Knoten, um ihn zu markieren.

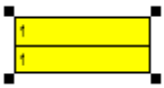
Mehrere Knoten können Sie markieren und toggeln, indem Sie sie bei gedrückter Strg-Taste mit der linken Maustaste anklicken. Jeder Klick auf einen markierten Knoten demarkiert diesen, jeder Klick auf einen nicht markierten Knoten markiert diesen.

Einen Teilbaum markieren Sie, indem Sie bei gedrückter Shift-Taste den Vaterknoten des Teilbaums anklicken.

Sie können alle markierten Knoten demarkieren, indem Sie mit der linken Maustaste in den leeren Diagrammbereich klicken.

Probieren Sie verschiedene Alternativen für den Knotenmarkierungstyp aus.

In der Abbildung sehen Sie einen durch Pickmarks markierten Knoten:

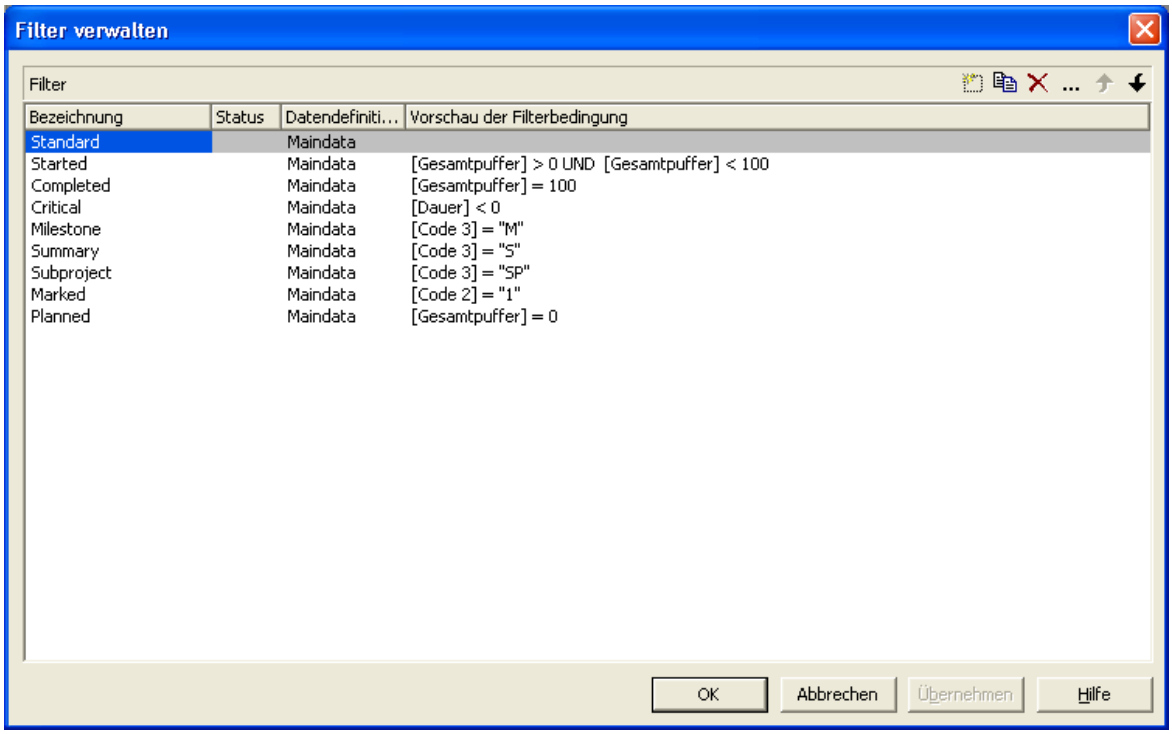


2.9 Filter für Knoten festlegen

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

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

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



> Schaltflächen im Dialogfeld "Filter verwalten"

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

> Filter erstellen und bearbeiten

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

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

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

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

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

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

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

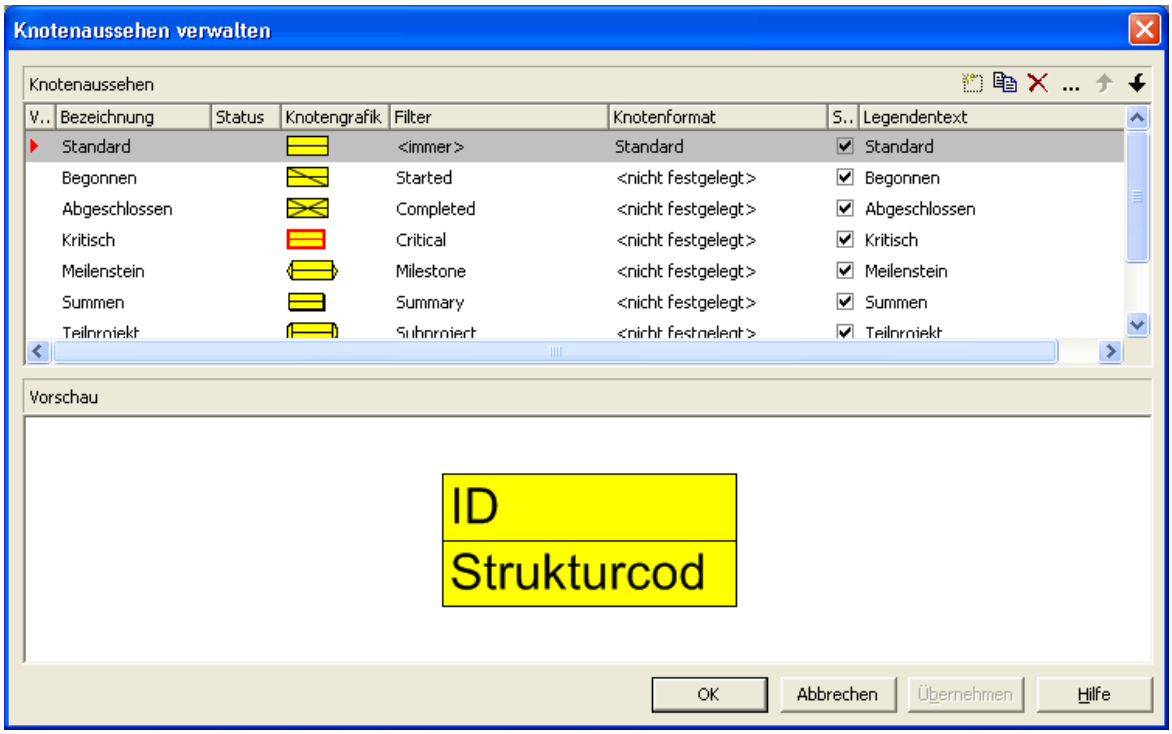
Definieren Sie den Filter wie in der Abbildung. Verlassen Sie dann das Dialogfeld **Filter bearbeiten** mit **OK**. Sie kehren in das Dialogfeld **Filter verwalten** zurück.

2.10 Knotenaussehen festlegen

VARCHART XTree bietet eine Vielzahl von Gestaltungsmöglichkeiten für das Aussehen von Knoten. Sie können das Aussehen von Knoten in Abhängigkeit von deren Daten festlegen. Beispielsweise können Sie festlegen, dass die Knoten für unterschiedliche Abteilungen jeweils unterschiedliche Farben erhalten.

Diese grafischen Attribute werden als Knotenaussehen bezeichnet.

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



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

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

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

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

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

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

> Knotenaussehen hinzufügen, kopieren, löschen und bearbeiten

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

 **Knotenaussehen hinzufügen**

 **Knotenaussehen kopieren**

 **Knotenaussehen löschen**

 **Knotenaussehen bearbeiten**

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

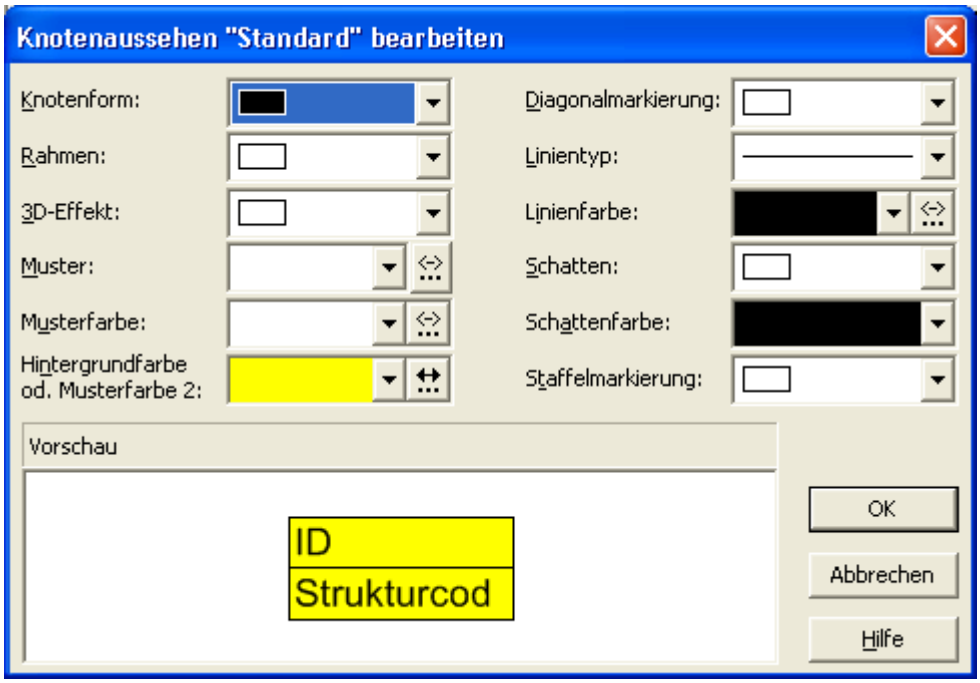
> Knotenaussehen und Filter verwenden

Lernen Sie nun den Umgang mit Knotenaussehen und Filtern an einem Beispiel kennen. Erzeugen Sie die neuen Knotenaussehen "Abteilung A" und "Abteilung B" als Kopien des Knotenaussehens "Standard".

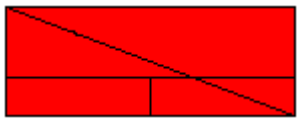
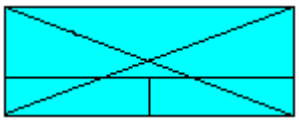
Weisen Sie dem Knotenaussehen "Abteilung A" die höchste und dem Knotenaussehen "Abteilung B" die zweithöchste Priorität zu.

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

können Sie hier festlegen, mit welchem Knotenformat und mit welchem Filter das Knotenaussehen kombiniert werden soll.



Legen Sie nun folgendes fest:

Knotenaussehen	Abteilung A	Abteilung B
Filter	Abteilung A	Abteilung B
Filterbedingung	Gruppenname gleich A	Gruppenname gleich B
Hintergrundfarbe	rot	blau
Diagonalmarkierung	abwärts	gekreuzt
Aussehen		

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

- Geben Sie für "Abteilung" den Wert "A" an: Der Knoten erscheint mit dem "Abteilung A"-Knotenaussehen, also in rot und abwärts durchgestrichen.
- Geben Sie nun für "Abteilung" den Wert "B" an: Der Knoten erscheint mit dem Knotenaussehen "Abteilung B", also in blau und mit gekreuzten Linien.

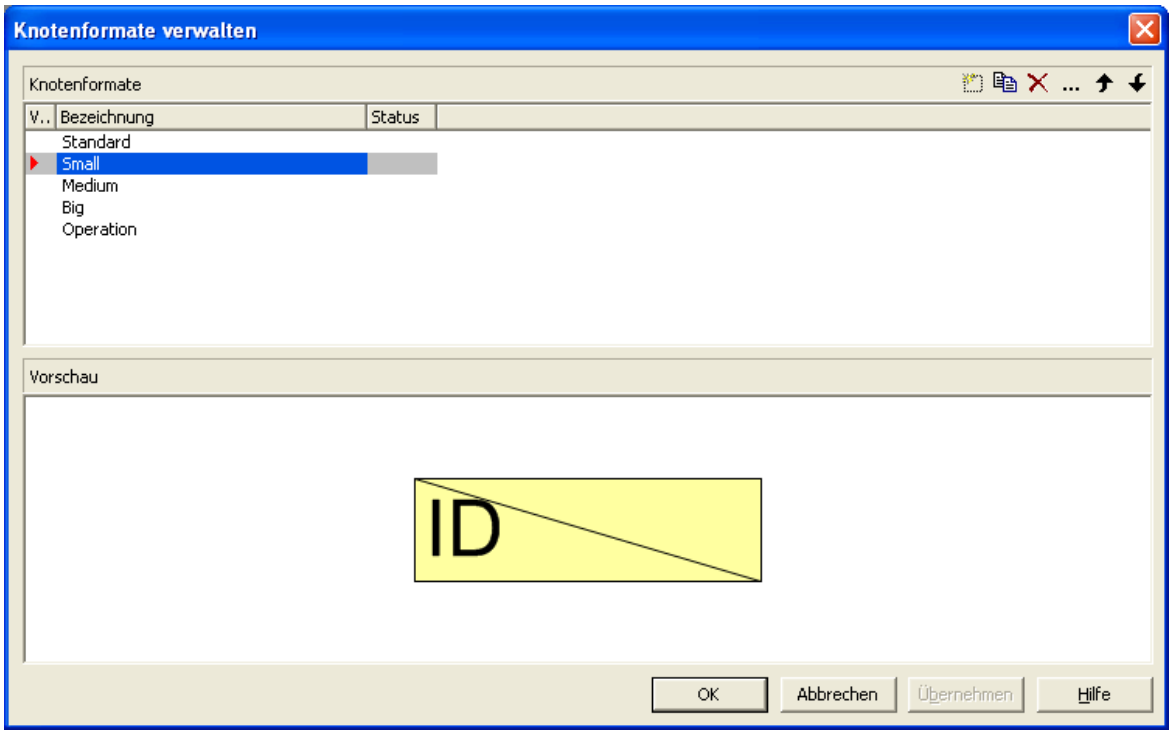
> Knotenaussehen datenabhängig festlegen

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

2.11 Knotenformate festlegen

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

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



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

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

Knotenformat hinzufügen

Knotenformat kopieren

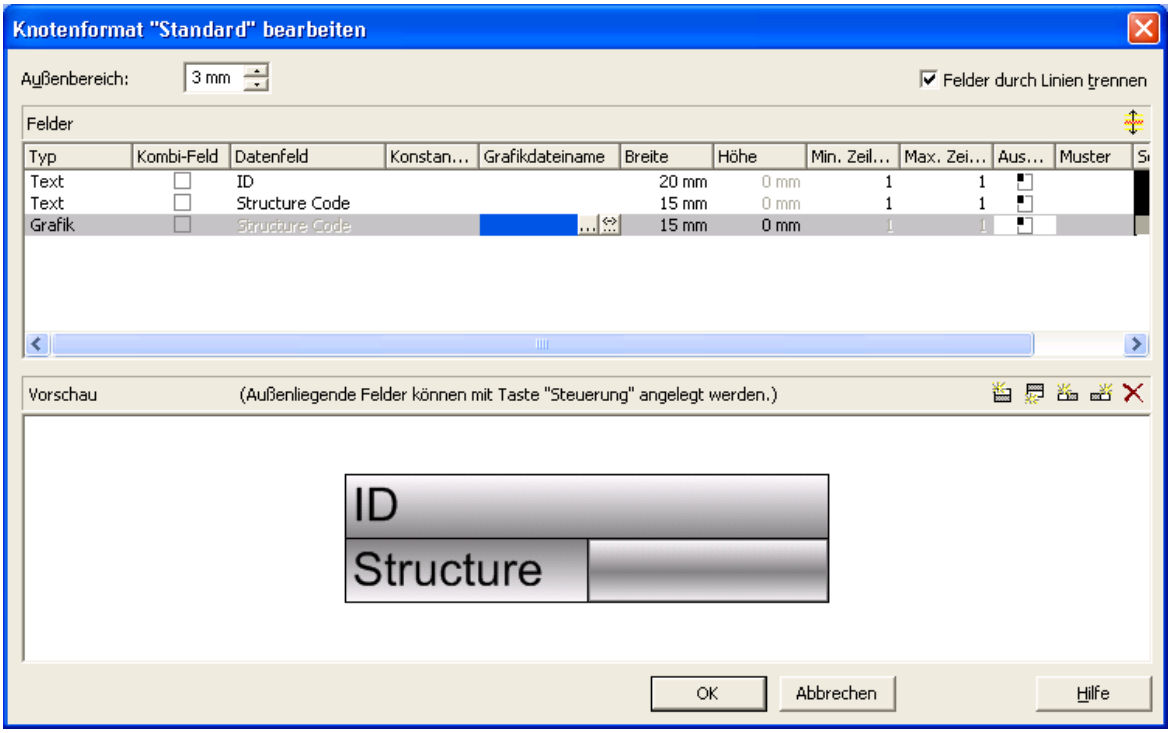
Knotenformat löschen

Knotenformat bearbeiten

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

> Knotenformat bearbeiten

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



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

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

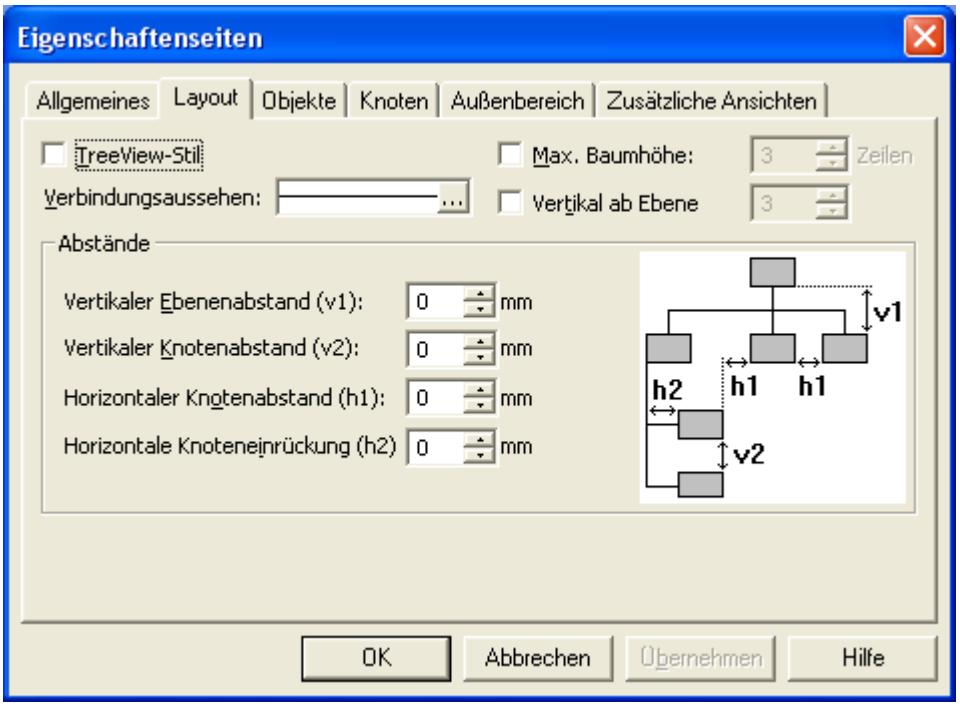
> Darstellung von Grafiken in Knotenfeldern

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

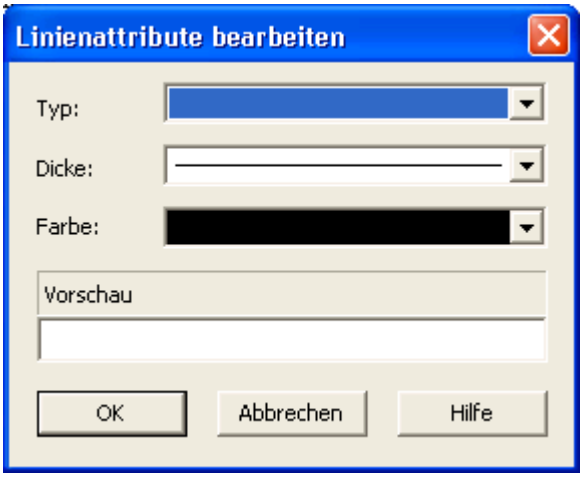
Sie können aber auch eine Zuordnung zwischen den Einträgen eines Datenfeldes und Grafikdateien herstellen. Klicken Sie dazu auf die Schaltfläche **Zuordnungen einstellen** (), um den gleichnamigen Dialog zu öffnen. Wenn eine Zuordnung vorgenommen worden ist, wird das durch ein Symbol neben dem Grafikdateinamen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

2.12 Das Aussehen von Verbindungen festlegen

Auf der Eigenschaftenseite **Layout** wird im Feld **Verbindungsaussehen** das aktuelle Verbindungsaussehen angezeigt.

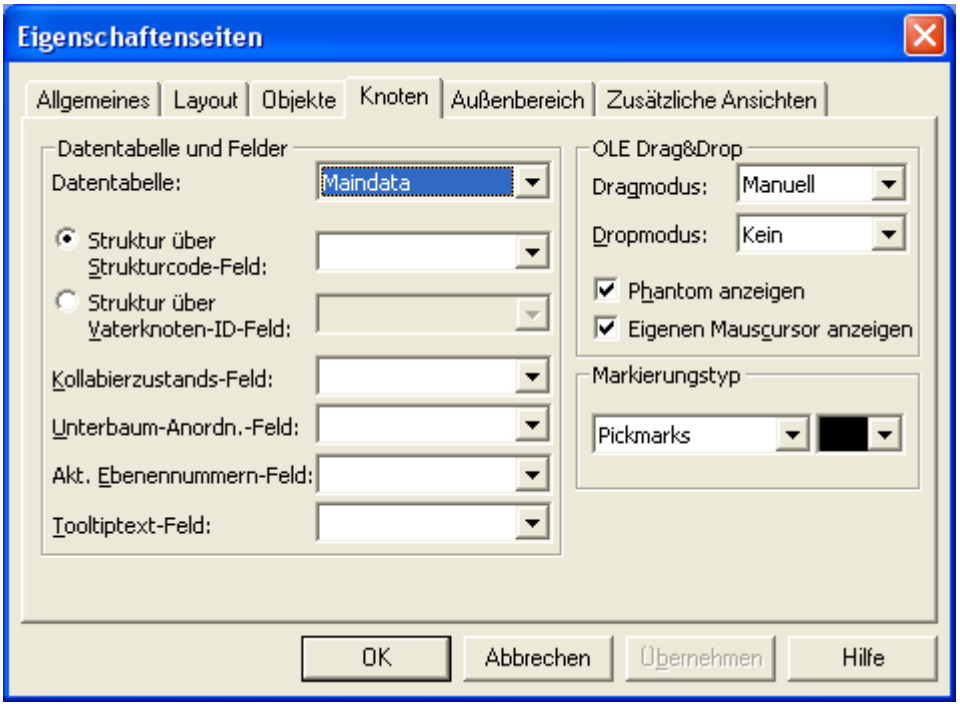


Um es zu ändern, klicken Sie auf die **Bearbeiten**-Schaltfläche. Sie gelangen dann in das Dialogfeld **Linienattribute bearbeiten**, in dem Sie Typ, Dicke und Farbe der Verbindungslinien festlegen können.



2.13 Baumstruktur festlegen

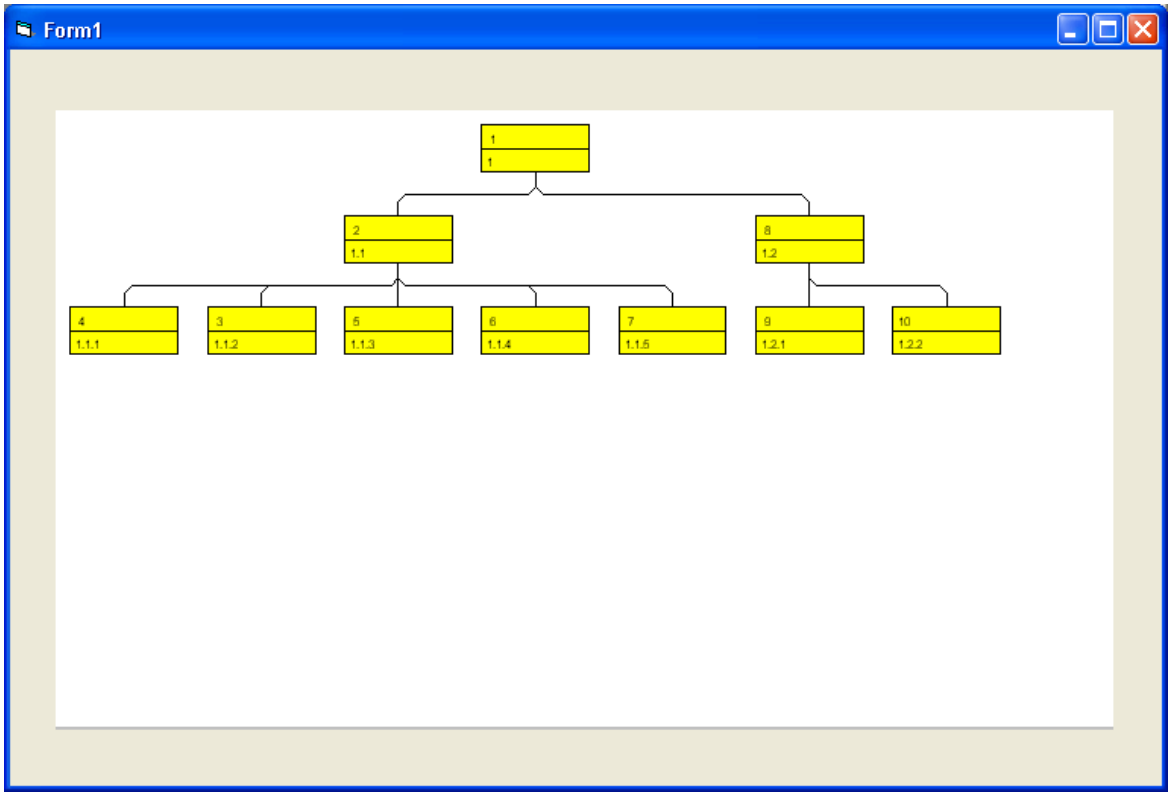
Auf der Eigenschaftenseite **Knoten** können Sie die Struktur der Baum-Diagramme festlegen.



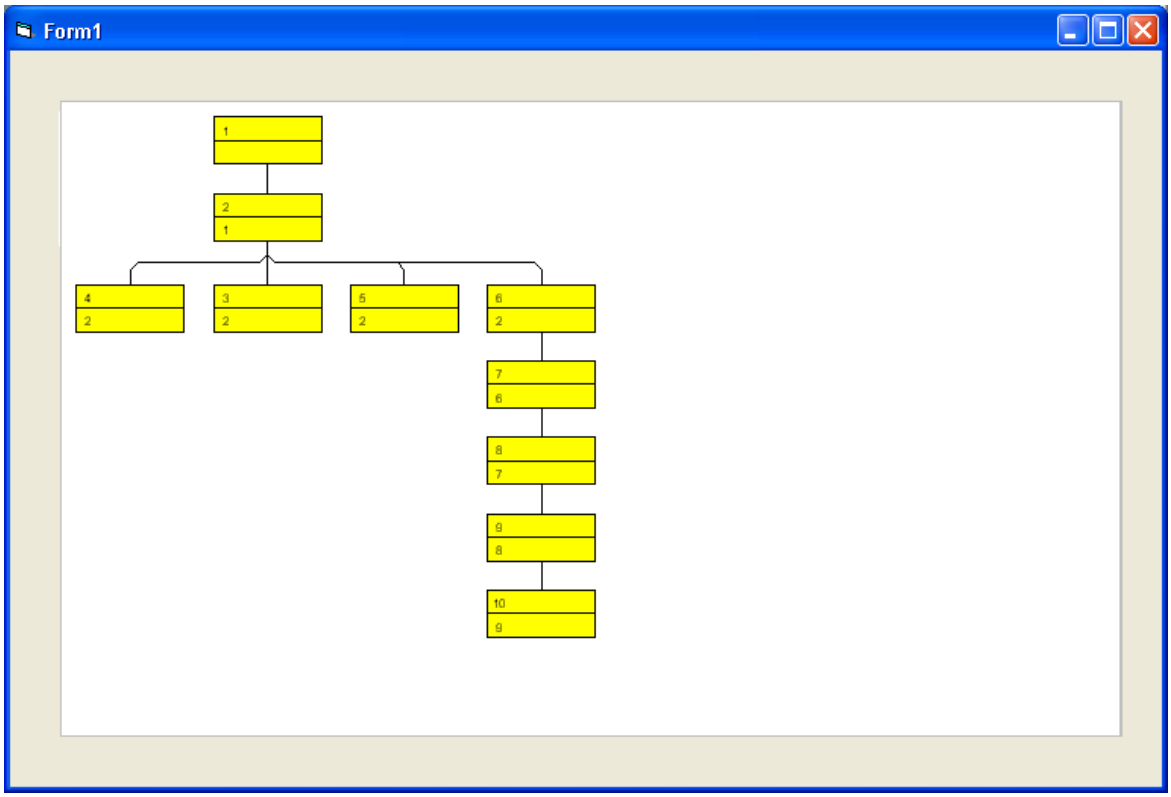
Dabei gibt es zwei grundsätzliche Alternativen:

1. **Struktur über Strukturcodefeld:** Der Baum wird gemäß eines Strukturcodes aufgebaut. Sie können dann ein Datenfeld auswählen, das den Wert des Strukturcodes enthält. Die Ebenen werden durch ein Trennzeichen (Punkt) voneinander getrennt.
2. **Struktur über Vaterknoten-ID-Feld:** Der Baum wird durch die ID des Vaterknotens jedes Knotens definiert. Sie können das Datenfeld wählen, das die ID des Vaterknotens enthält.

Wählen Sie zunächst die Option **Struktur über Strukturcodefeld:** und wählen Sie das Feld "Strukturcode" aus. Der Wert dieses Feldes bestimmt dann die Baumstruktur. Starten Sie nun das Programm. Sie erhalten folgendes Resultat:



Kehren Sie nun zum Design-Modus zurück und wählen Sie die Option **Struktur über Vaterknoten-ID-Feld**. Wählen Sie als Datenfeld, das die ID des Vaterknotens enthält, das Feld "Vaterknoten". Starten Sie nun das Programm. Sie erhalten folgendes Resultat:

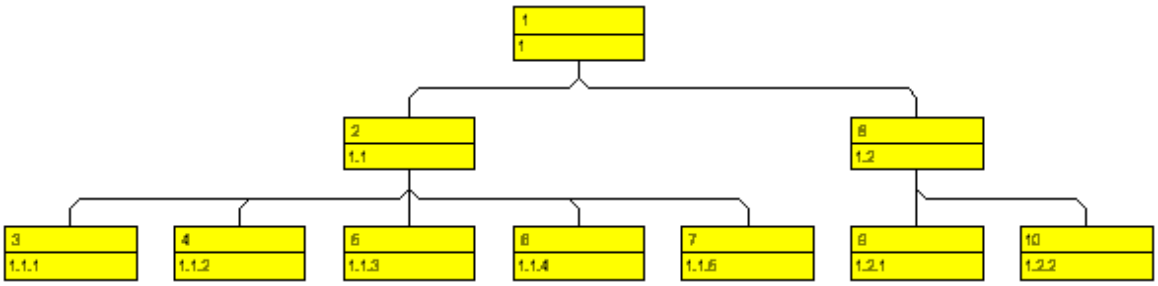


2.14 Vertikale und horizontale Anordnung von Baumstrukturen

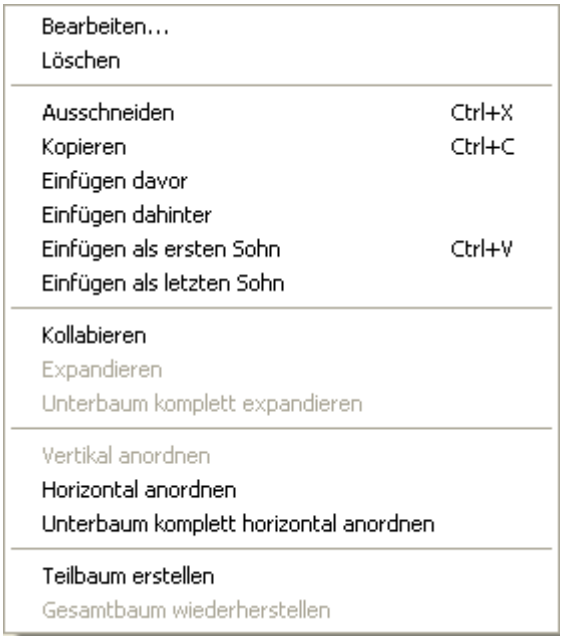
Lernen Sie nun, wie Sie Ihr Baum-Diagramm durch die Kombination horizontal und vertikal angeordneter Teilstrukturen optimieren können.

- *Horizontale Anordnung:* Durch eine horizontale Anordnung lässt sich die Höhe eines Baum-Diagramms verringern. Alle Knoten einer Ebene werden dabei nebeneinander angeordnet. Der Port (Anknüpfungspunkt) der Verbindungslinie liegt dann mittig am unteren Rand des Vaterknotens und mittig am oberen Rand des Sohnknotens.
- *Vertikale Anordnung:* Durch eine vertikale Anordnung lässt sich die Breite des Baum-Diagramms verringern. Alle Knoten einer Ebene und deren Unterebenen werden dabei untereinander angeordnet. Der Port der Verbindungslinie liegt dann in der unteren linken Ecke des Vaterknotens und in der Mitte des linken Randes des Sohnknotens.

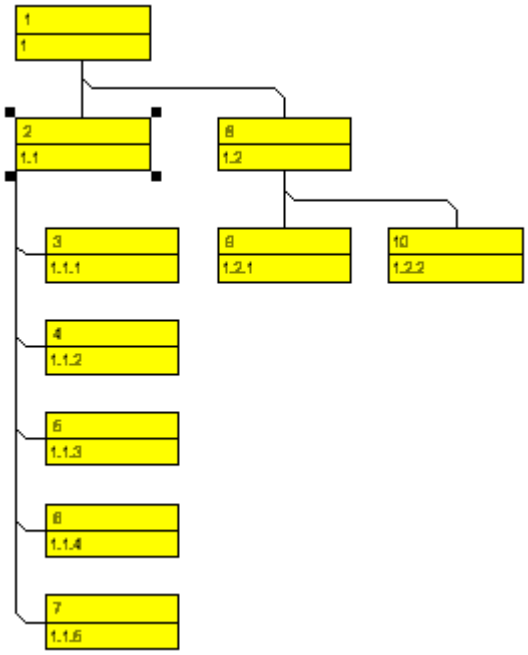
Legen Sie auf der Eigenschaftenseite **Knoten** fest: **Strukturcode in Feld:** "Strukturcode". Starten Sie dann das Programm mit den Daten der Datei *tutorial.tre* (vgl. "Tutorium: Daten aus einer Datei einlesen").



Markieren Sie nun den ersten Knoten der zweiten Ebene und klicken Sie auf die rechte Maustaste. Das folgende Kontextmenü erscheint, wobei jeweils nur die verfügbaren Befehle aktiviert sind.



Wählen Sie den Befehl **Vertikal anordnen**. Dadurch wird der Teilbaum unter dem markierten Knoten vertikal angeordnet.



Falls der Baum nicht vollständig vertikal angeordnet wird, wechseln Sie wieder zum Designmodus, öffnen Sie die Eigenschaftenseite **Layout** und prüfen Sie dort die vereinbarte maximale Baumhöhe. Die Gesamthöhe eines Baum-Diagramms (in Zeilen) bei der vertikalen Anordnung wird durch die Angabe unter **Max. Baumhöhe** begrenzt. Diese Einstellung wirkt sich nur bei vertikaler Anordnung aus. Falls in einem vertikal angeordneten Ast mehr Ebenen vorhanden sind, erfolgt ein Umbruch, und ein neuer Ast wird am Vaterknoten erzeugt.

Aktivieren Sie das Kontrollkästchen **Max. Baumhöhe** und wählen Sie den Wert "10".

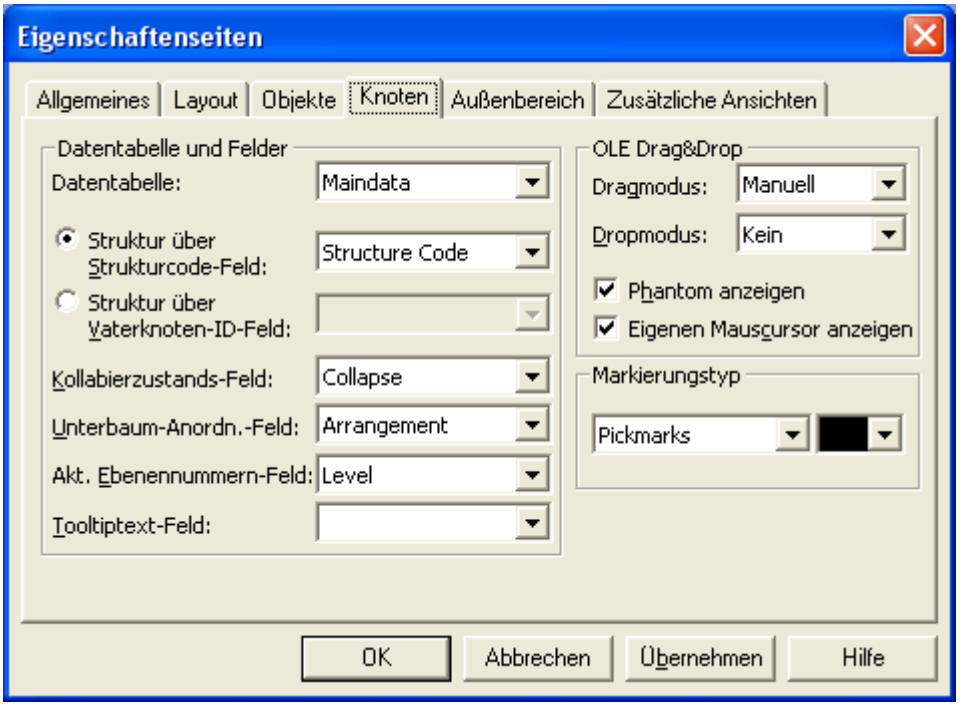
Um die vertikal angeordnete Teilstruktur wieder horizontal anordnen zu lassen, markieren Sie den obersten Knoten der vertikal angeordneten Teilstruktur und wählen Sie anschließend im Kontextmenü den Befehl **Horizontal anordnen**. Die Teilstrukturen des markierten Knotens werden dann horizontal angeordnet, aber nur eine Ebene tief. Auf die Anordnungen in den nächst tieferen Ebenen hat der Befehl **Horizontal anordnen** keine Auswirkung.

Mit dem Befehl **Unterbaum komplett horizontal anordnen** werden die Teilbäume unter den markierten Knoten vollständig, d. h. über alle Ebenen, horizontal angeordnet.

> Unterbaum-Anordnung in Datenfeld speichern

Ob Teilbäume vertikal oder horizontal angeordnet werden sollen, können Sie in einem Datenfeld speichern.

Kehren Sie dazu zum Designmodus zurück und öffnen Sie die Eigenschaftenseite **Knoten**.

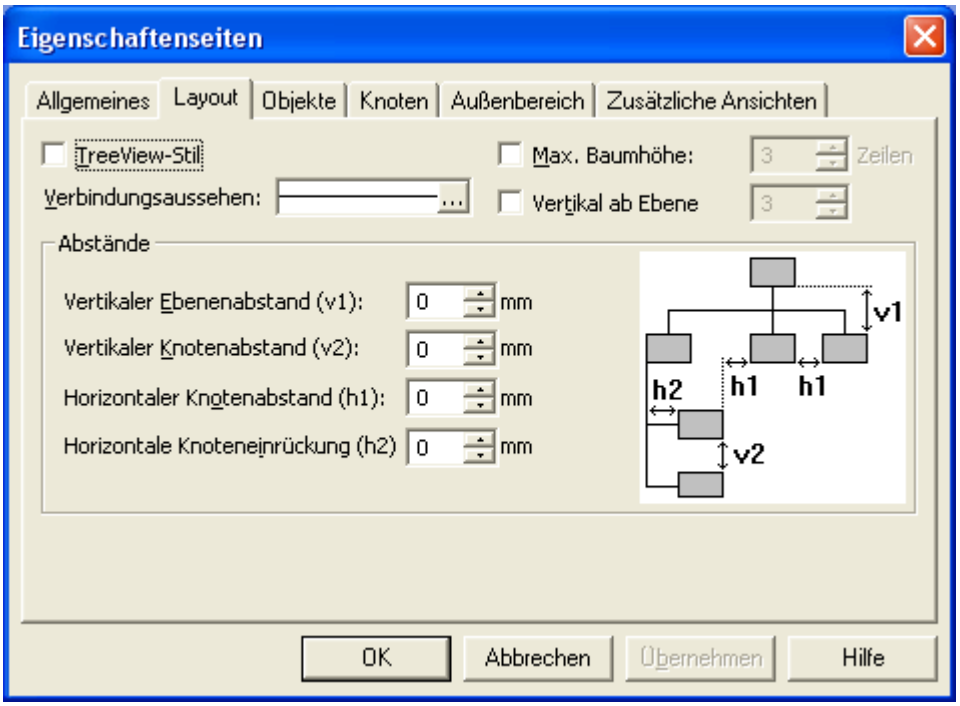


Bitte wählen Sie aus der Kombobox das Datenfeld "Anordnung" aus. Wenn dieses Datenfeld "0" enthält, wird der darunter sitzende Teilbaum horizontal angeordnet, bei "1" vertikal. Die horizontale Anordnung ist nur sichtbar, wenn der direkte oder indirekte Vaterknoten selbst eine vertikale Anordnung besitzt.

Starten Sie nun das Programm. Markieren Sie einen Knoten und klicken Sie doppelt auf die linke Maustaste, um den Dialog **Vorgänge bearbeiten** zu öffnen. Ist die Teilstruktur unterhalb des Knotens horizontal angeordnet, steht hier unter **Anordnung** der Wert "0". Ist die Teilstruktur unterhalb des Knotens vertikal angeordnet, steht hier unter **Anordnung** der Wert "1".

> Abstände

Auf der Eigenschaftenseite **Layout** können Sie außerdem folgende Abstände festlegen (Einheit: mm):



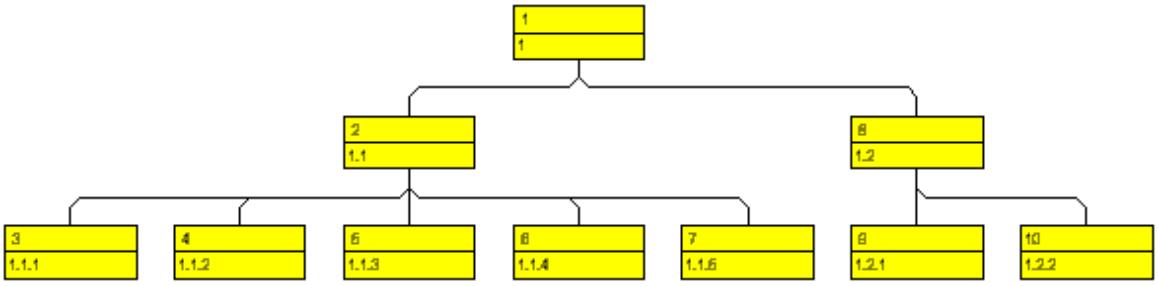
- **Vertikaler Ebenenabstand (v1):** Hier können Sie den vertikalen Abstand zwischen zwei horizontal angeordneten Knotenebenen festlegen.
- **Vertikaler Knotenabstand (v2):** Hier können Sie den vertikalen Abstand zwischen zwei vertikal angeordneten Knoten festlegen.
- **Horizontaler Knotenabstand (h1):** Hier können Sie den horizontalen Abstand zwischen zwei horizontal angeordneten Knoten festlegen.
- **Horizontale Knoteneinrückung (h2):** Hier können Sie die horizontale Einrückung vertikal angeordneter Knoten festlegen.

2.15 Baumstrukturen kollabieren und expandieren

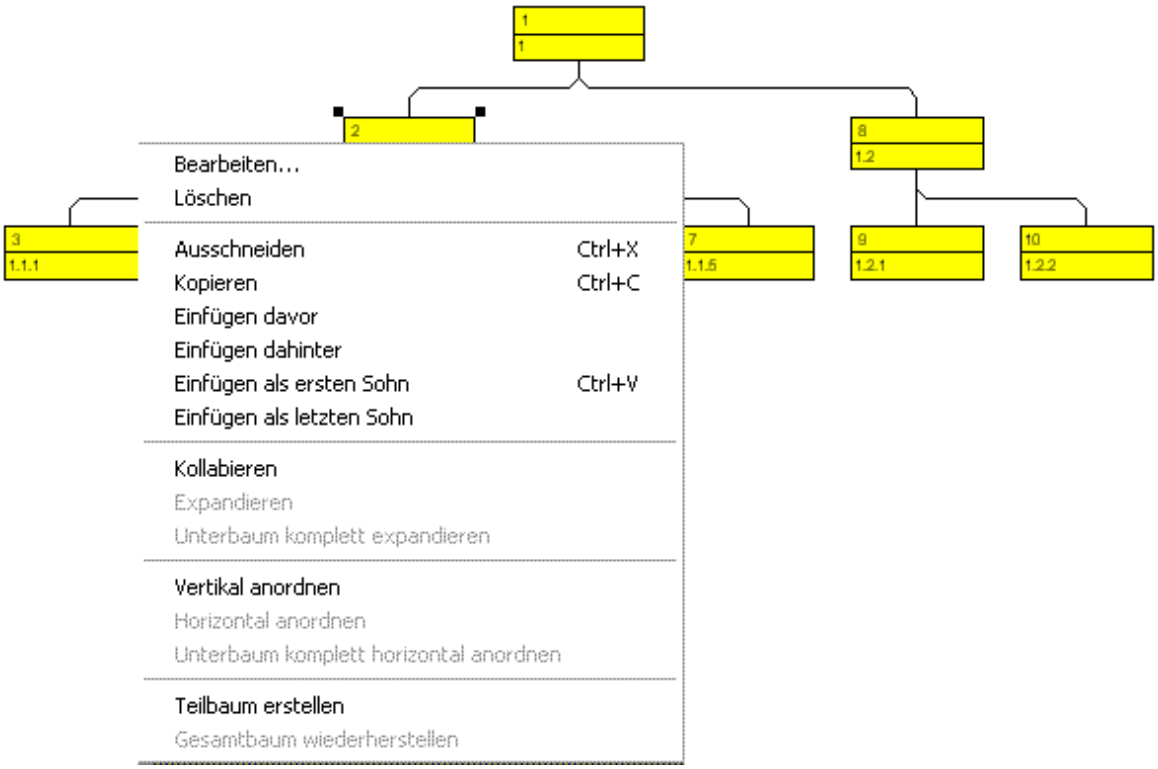
Hier erfahren Sie, wie Sie Baumstrukturen kollabieren und wieder expandieren können. Teilstrukturen von Baum-Diagrammen lassen sich kollabieren (wegklappen) und wieder expandieren (aufklappen). Jede beliebige Teilstruktur kann auf ihren obersten Knoten, den Gliederungsknoten, kollabiert werden. Als Gliederungsknoten wird jeweils der Wurzelknoten der Teilstruktur, die kollabiert werden soll, verwendet. Die kollabierte Teilstruktur kann ggf. wieder in ihre ursprüngliche Form expandiert werden.

Durch das Kollabieren von Teilstrukturen lassen sich auch sehr komplexe Strukturen übersichtlich gestalten. Wenn Sie beispielsweise nur bestimmte Teilstrukturen präsentieren möchten, kollabieren Sie einfach alle anderen Teilstrukturen. Dadurch, dass die kollabierten Teilstrukturen erhalten bleiben, geht die Information über die Gesamtstruktur nicht verloren.

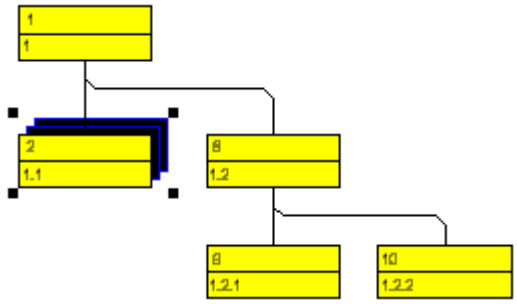
Starten Sie nun das Programm mit den Daten der Datei *tutorial.tre* (vgl. "Tutorium: Daten aus einer Datei einlesen").



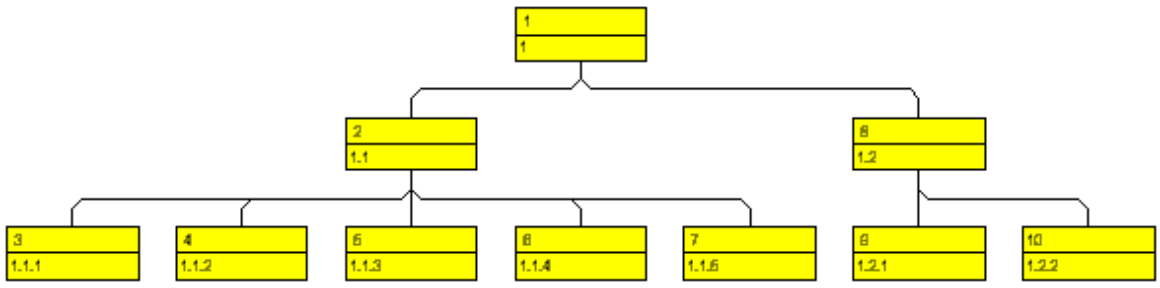
Markieren Sie nun den ersten Knoten der zweiten Ebene und klicken Sie auf die rechte Maustaste. Das Kontextmenü für Knoten erscheint.



Wählen Sie den Befehl **Kollabieren**. Nun werden alle Teilstrukturen, die zu dem markierten Wurzelknoten gehören, kollabiert. Die markierten Wurzelknoten verwandeln sich damit in Gliederungsknoten, die jeweils die nun verborgenen Teilstrukturen repräsentieren.

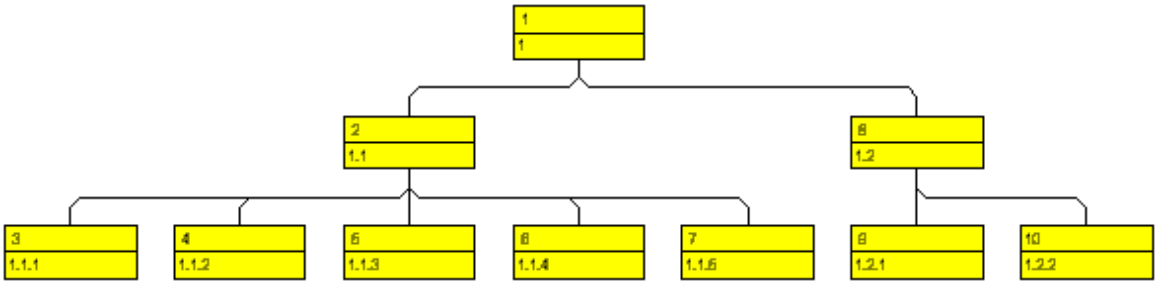


Wählen Sie nun den Befehl **Expandieren** aus dem Kontextmenü, um die Teilstrukturen wieder zu expandieren. Es werden jeweils nur die kollabierten Knoten expandiert. Befinden sich in den Unterbäumen weitere kollabierte Knoten, bleiben diese kollabiert.



Mit dem Befehl **Unterbaum komplett expandieren** aus dem Kontextmenü können Sie die kollabierten Teilstrukturen vollständig expandieren, d. h. dass dabei alle kollabierten Knoten in den Unterbäumen expandiert werden. Um diesen Befehl auszuprobieren, kollabieren Sie nun den ersten Knoten der zweiten Ebene und anschließend den Knoten der obersten Ebene.

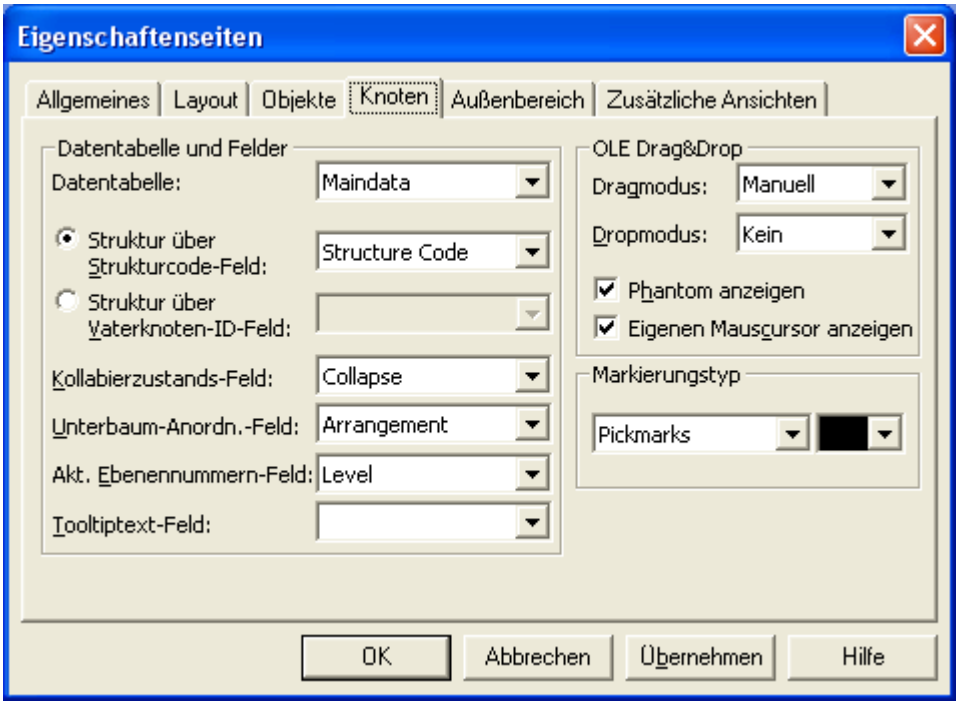
Wählen Sie nun den Befehl **Unterbaum komplett expandieren**. Nun wird der Baum wieder komplett expandiert.



> Kollabierzustand in Datenfeld speichern

Sie können den Kollabierzustand der Knoten in einem Datenfeld speichern.

Kehren Sie dazu zum Designmodus zurück und öffnen Sie die Eigenschaftenseite **Knoten**.



Wählen Sie aus der Kombobox hinter **Kollabierzustand in Feld** das Datenfeld "Kollabiert" aus. Ab jetzt wird der Kollabierstatus jedes Knotens in

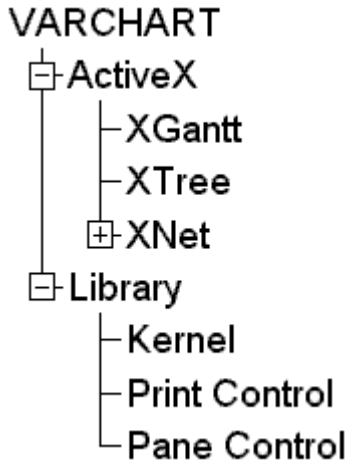
dem Datenfeld "Kollabiert" gehalten. Mögliche Inhalte des Datenfelds sind "0" (Knoten expandiert) und "1" (Knoten kollabiert).

Starten Sie nun das Programm. Markieren Sie einen Knoten und klicken Sie doppelt auf die linke Maustaste, um den Dialog **Vorgänge bearbeiten** zu öffnen. Ist die Teilstruktur unterhalb des Knotens kollabiert, steht hier unter **Kollabiert** der Wert "1". Ist die Teilstruktur unterhalb des Knotens expandiert, steht hier unter **Kollabiert** der Wert "0".

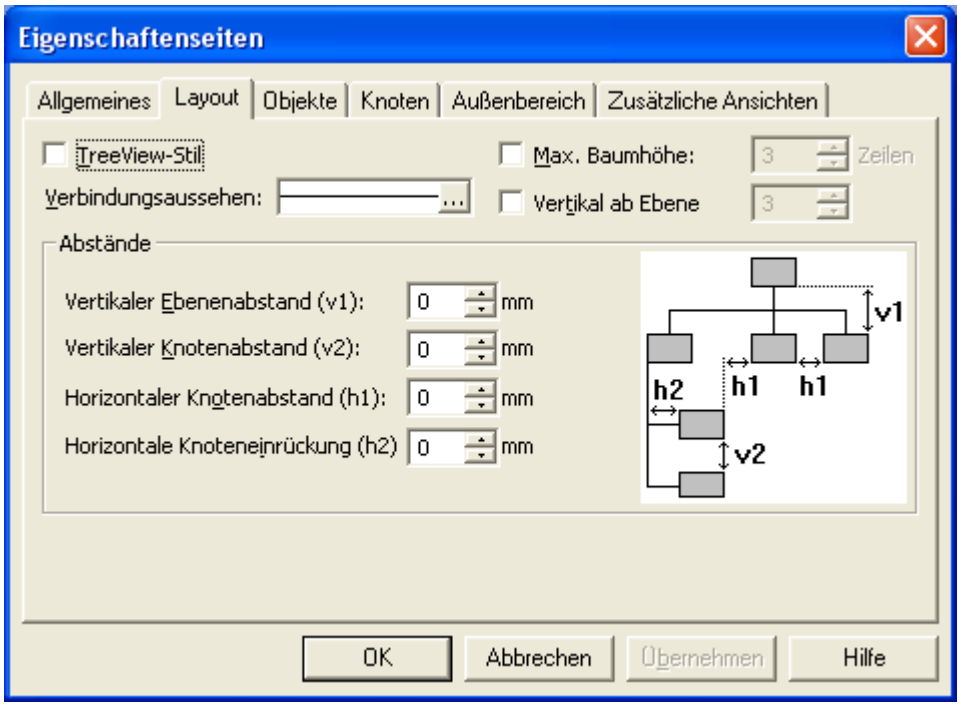
2.16 TreeView-Stil

Lernen Sie nun kennen, wie Sie Knoten im TreeView-Stil anordnen lassen können. Bei der TreeView-Ansicht werden vertikale Ebenen wie in TreeView-Controls (z. B. bekannt aus der Verzeichnisbaumansicht des Microsoft Explorers) mit Plus- oder Minus-Zeichen dargestellt. Dabei bedeutet ein Plus-Zeichen, dass der Knoten auf der gleichen Ebene kollabiert ist, und ein Minus-Zeichen, dass er expandiert ist. Die Zeichen werden nur bei Knoten angezeigt, die keine Blattknoten sind, d. h. die Sohnknoten besitzen. Ein Mausklick auf eines der beiden Zeichen überführt den daneben stehenden Knoten in den jeweils anderen Kollabierzustand.

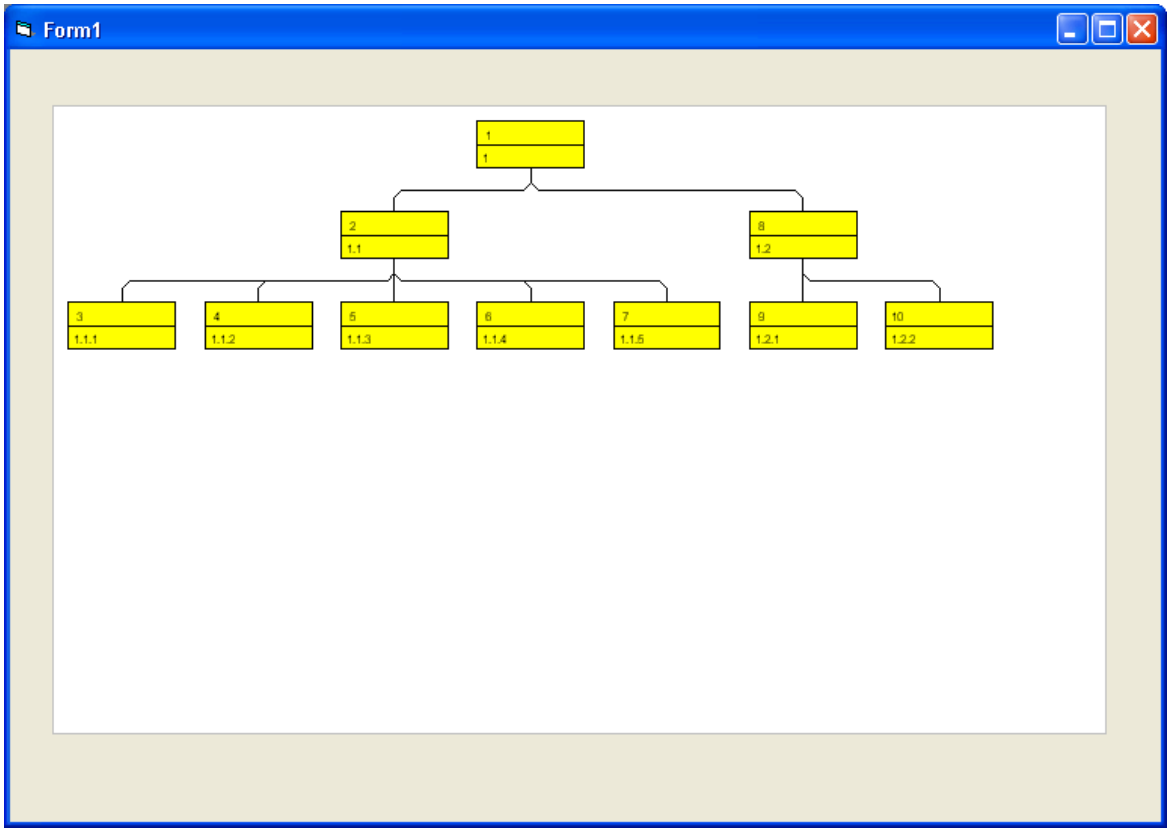
Ein Beispiel für den TreeView-Stil zeigt die folgende Abbildung:



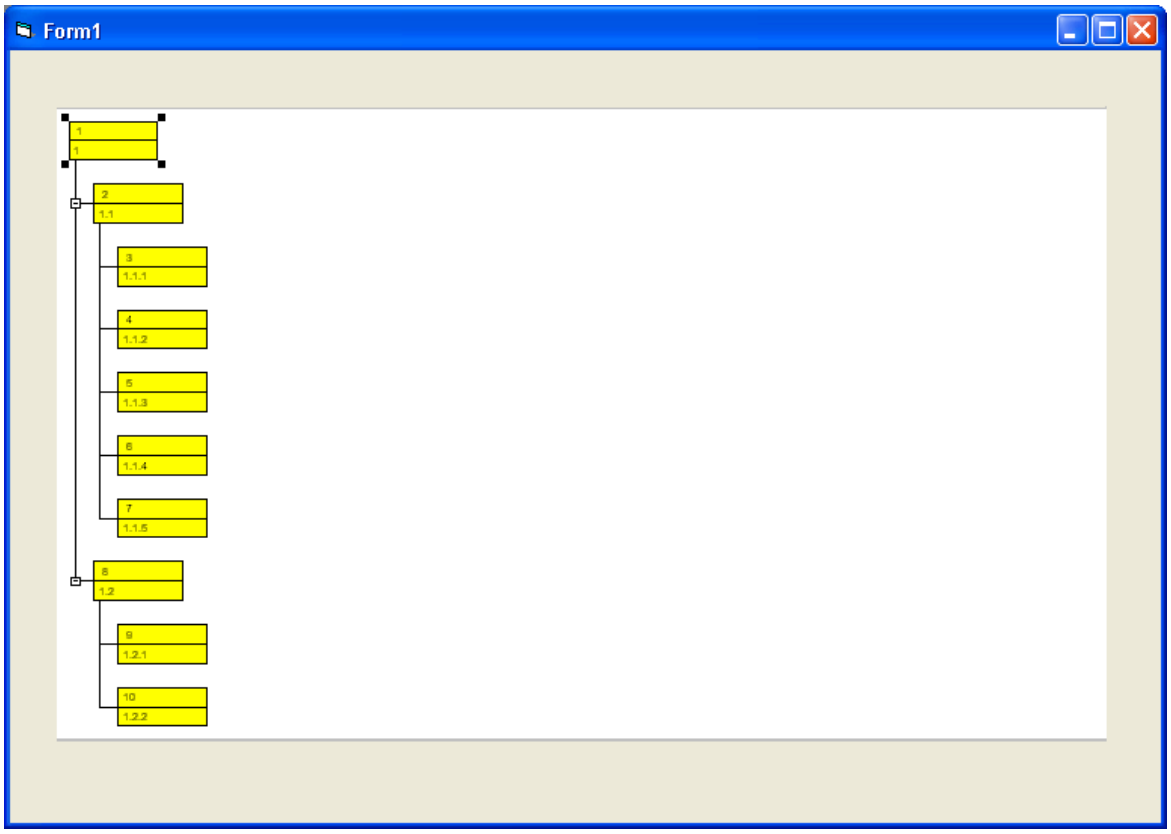
Das Kontrollkästchen **TreeView-Stil** auf der Eigenschaftenseite **Layout** bestimmt, ob die Knoten im TreeView-Stil angeordnet werden.



Deaktivieren Sie das Kontrollkästchen zunächst und starten Sie dann das Programm. Sie erhalten eine Anordnung ohne TreeView-Stil:



Wechseln Sie nun wieder zum Design-Modus und aktivieren Sie auf der Eigenschaftenseite **Layout** das Kontrollkästchen **TreeView-Stil**, um die Knoten jetzt im TreeView-Stil anordnen zu lassen:


Form1
1
1
2
1.1
3
1.1.1
4
1.1.2
5
1.1.3
6
1.1.4
7
1.1.5
8
1.2
9
1.2.1
10
1.2.2

2.17 Diagramm drucken

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

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

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

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

Wenn Sie zur Laufzeit die Seiteneinstellungen verändern möchten,können Sie aus dem Kontextmenü den Befehl **Seite einrichten** auswählen, oder auf

Seitenansicht klicken und dort auf die Schaltfläche **Seite einrichten**.

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

Im **Seite einrichten**-Dialogfeld können Sie Einstellungen zur Skalierung, zur Seitenaufteilung, zur Seitennummerierung, zu den Seitenrändern etc. vornehmen. Weitere Informationen hierzu finden Sie im Kapitel 5.14, "Seite einrichten".

2.18 Diagramm exportieren

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

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

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

2.19 Konfigurationseinstellungen speichern

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

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

> So speichern Sie Ihre aktuelle Konfiguration:

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

> So lesen Sie eine bereits gespeicherte Konfiguration wieder ein:

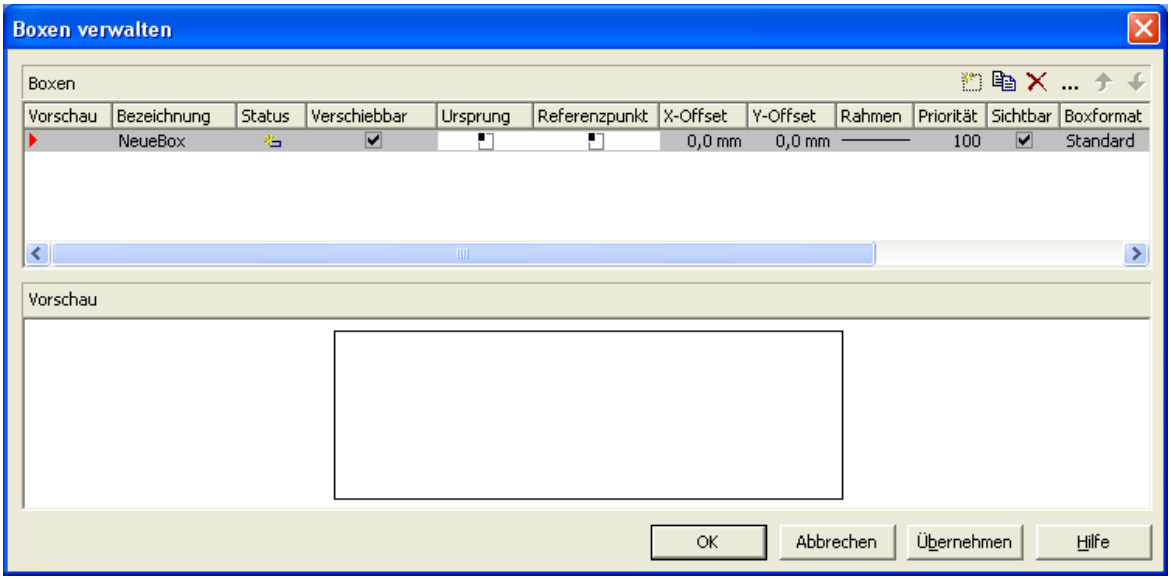
Der Eintrag im Feld **Konfigurationsdatei** wird verwendet, um den Namen einer Datei anzugeben, aus der die Einstellungen geladen werden sollen. Wenn Sie einen vorhandenen Dateinamen angeben und auf die **Übernehmen**-Schaltfläche klicken, wird die gewählte Konfiguration geladen und von nun an mit der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

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

3 Wichtige Konzepte

3.1 Boxen

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



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

Für jede Box können Sie hier folgendes festlegen:

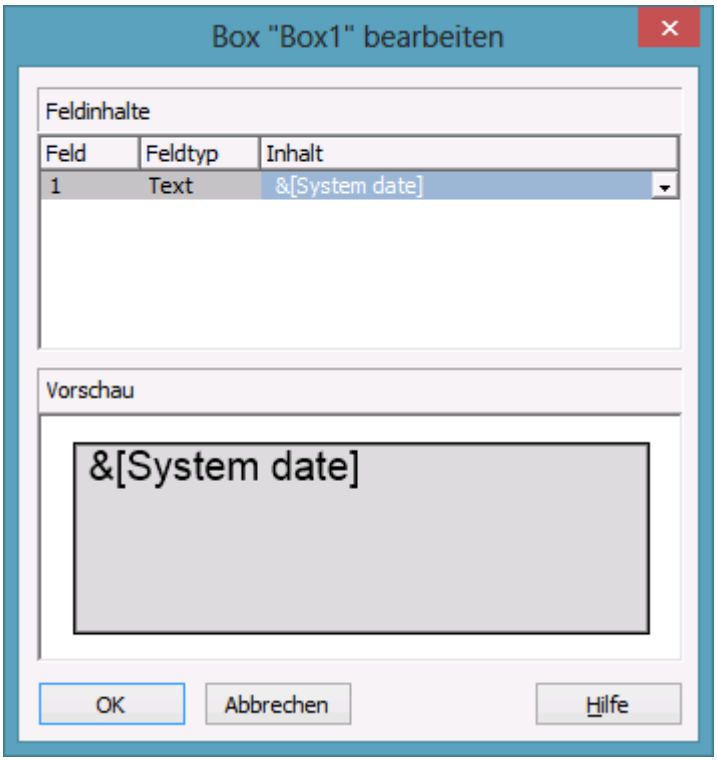
- ihre Bezeichnung
- ob sie zur Laufzeit frei im Diagrammbereich verschiebbar sein soll
- ihren Ursprung (den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird)
- ihren Referenzpunkt (Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird)
- ihren X- bzw. Y-Offset (Abstand zwischen Ursprung und Referenzpunkt in x- bzw. y-Richtung)

- Typ, Dicke und Farbe der Umrandungslinie der Box
- ihre relative Priorität gegenüber anderen Objekten im Diagramm
- ob die Box sichtbar ist
- das Boxformat

> Boxen bearbeiten

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

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



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

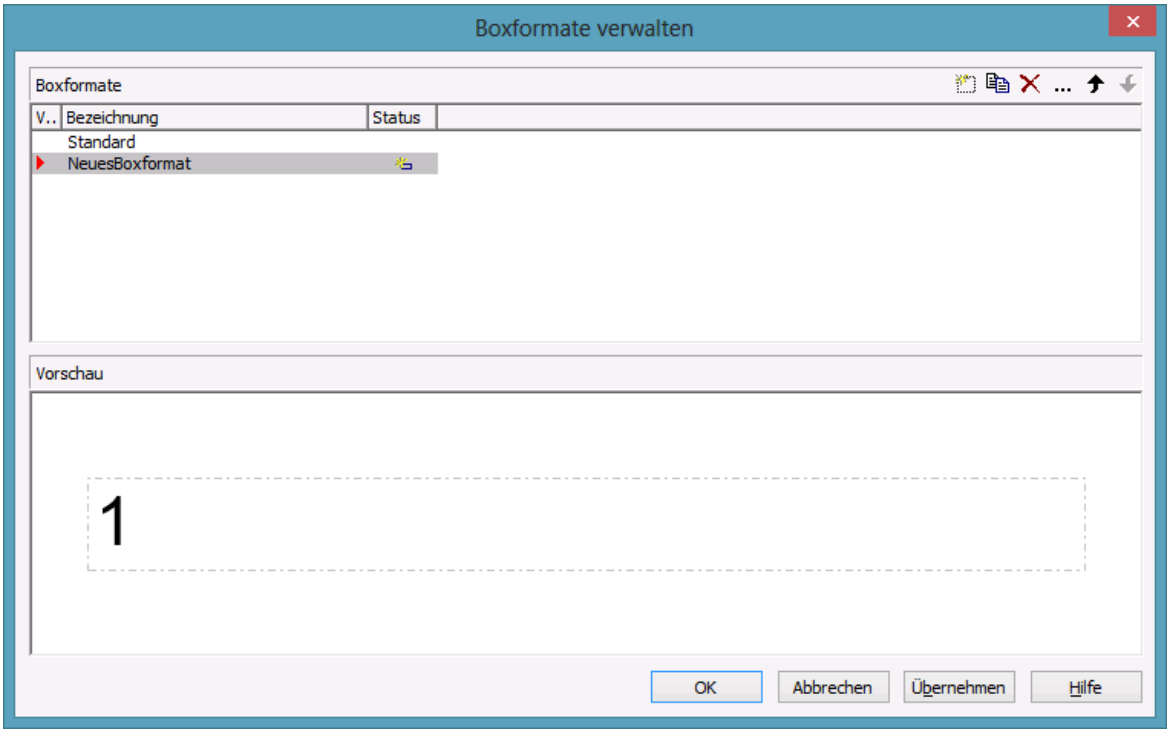
In der Spalte **Typ** wird der Feldtyp jedes Feldes angezeigt (Text oder Grafik).

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

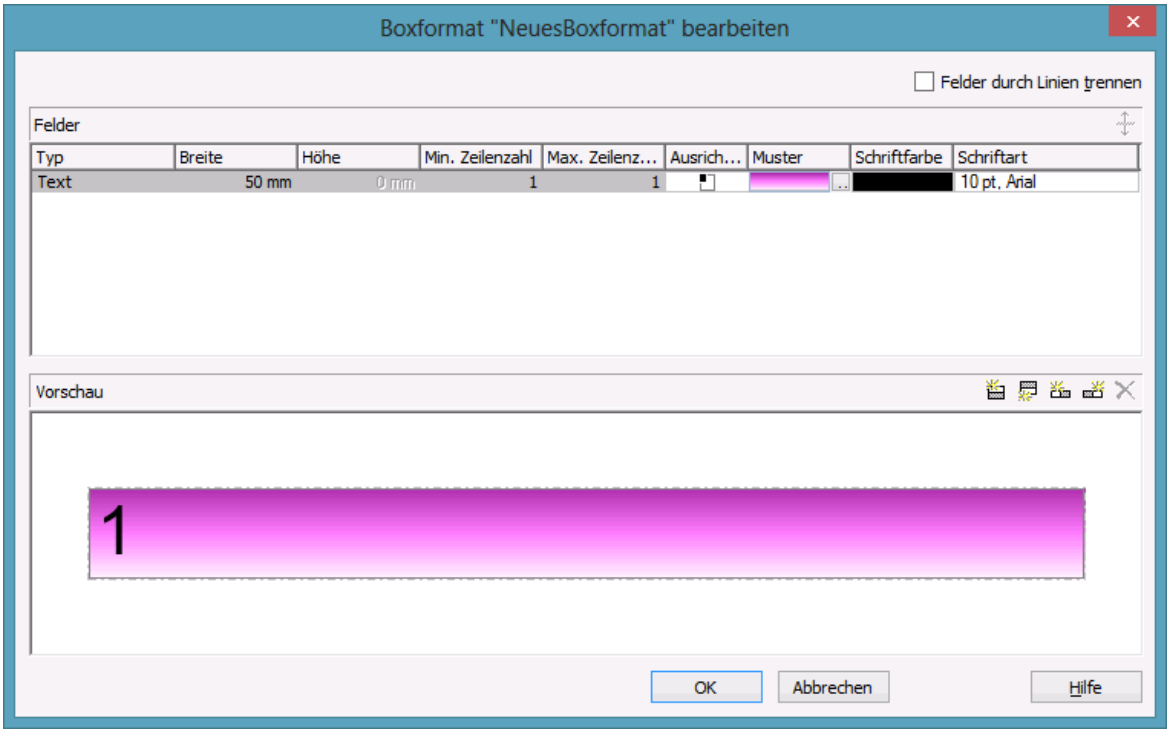
> Boxformate

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

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



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



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

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

3.2 Daten

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

Code-Beispiel

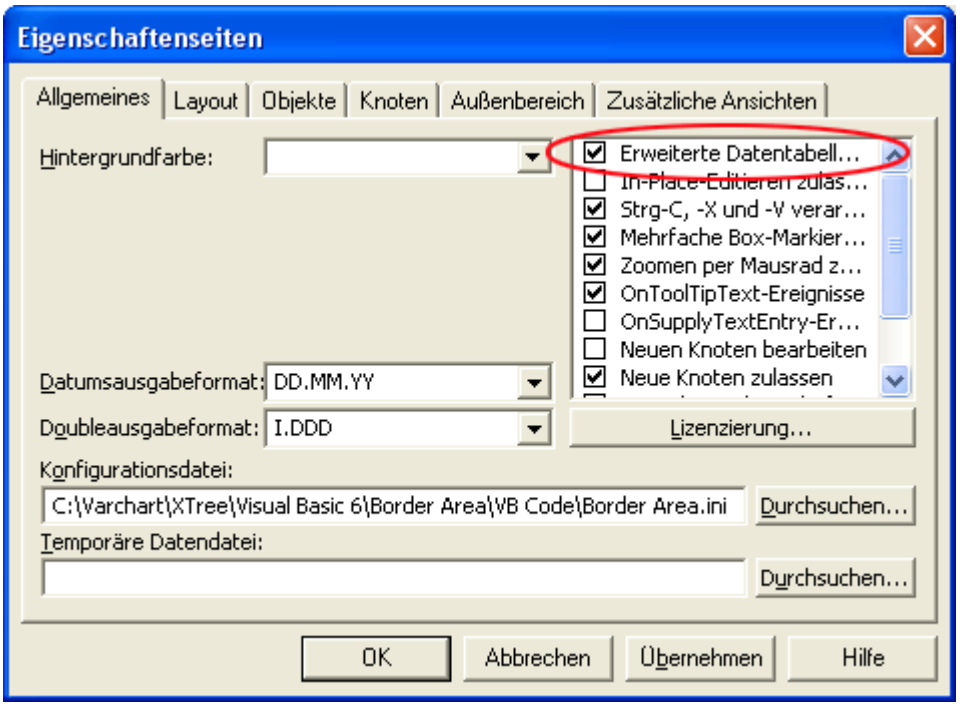
```
VcTree1.InsertNodeRecord "1;1.;;;" + Chr$(34) + " Firma A; Abteilung B "
+ Chr$(34) + ""
```

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

3.3 Datentabellen

Zur Darstellung von Baum-Diagrammen verwendet VARCHART XTree eine Standard-Datentabellen für Knoten, deren Felder individuell festgelegt werden können. In VARCHART XTree 4.0 wurde dieses Konzept erweitert. Es können jetzt bis zu 90 Datentabellen vereinbart und zusätzlich Beziehungen in Form von 1:n Relationen zwischen diesen Tabellen definiert werden. Indem - wie in Datenbanken üblich - die Daten in voneinander abhängigen Datensätzen organisert werden, werden Redundanzen vermieden.

Aus Kompatibilitätsgründen mit bestehenden Anwendungen arbeitet VARCHART XTree4.0 standardmäßig im bisherigen Modus. Erst durch Aktivieren der entsprechenden Option zur Design- oder Laufzeit können die neuen Möglichkeiten genutzt werden. Auf der Eigenschaftenseite **Allgemeines** finden Sie dazu die Option **Erweiterte Datentabellen**:



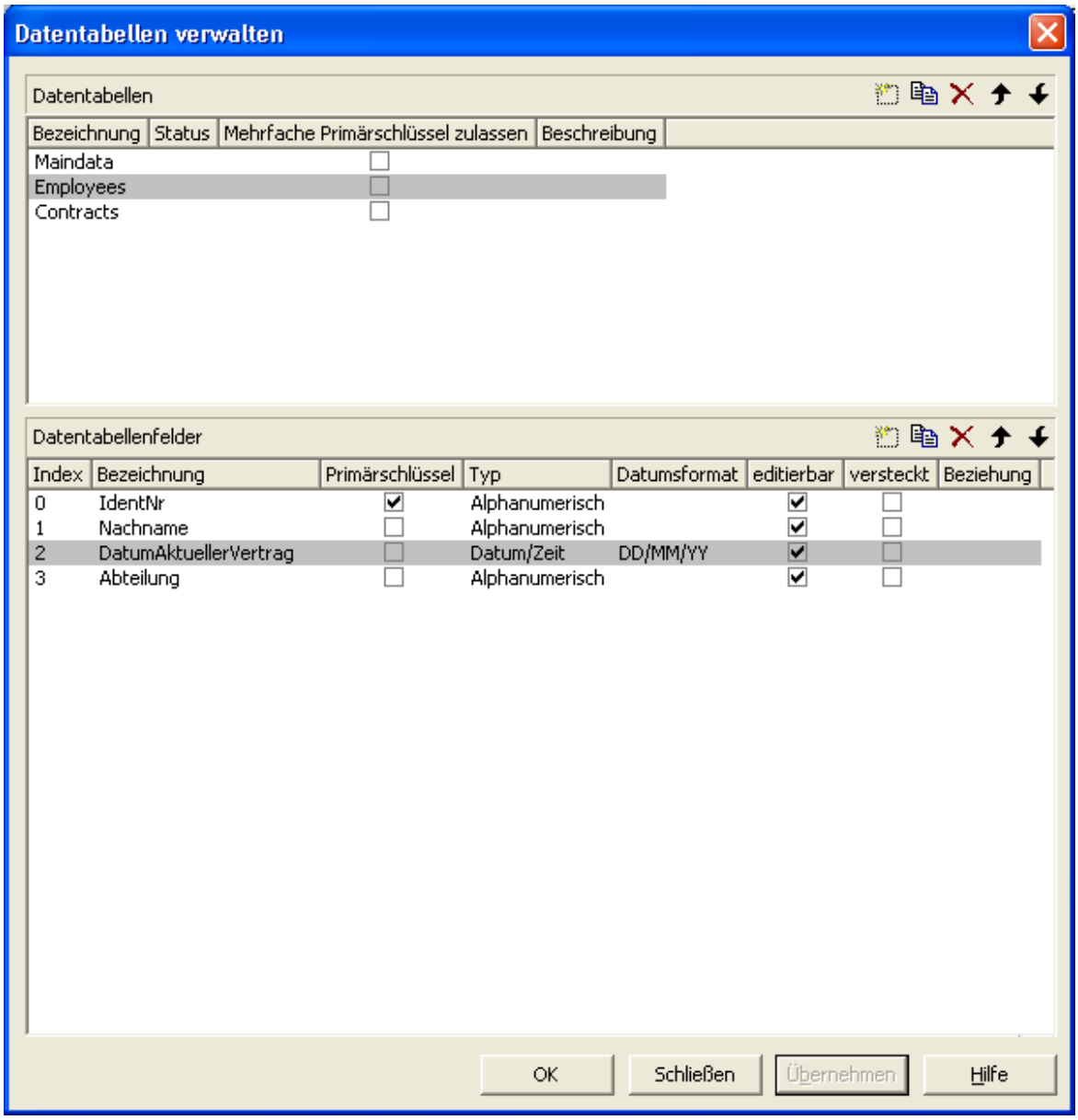
Alternativ können die erweiterten Datentabellen statt zur Designzeit auch zur Laufzeit über die API eingeschaltet werden, indem die Eigenschaft **ExtendedDataTables** des Objekts **VcTree** auf **True** gesetzt wird.

> Datentabellen verwalten

Standardmäßig ist die Datentabelle **Maindata** vorhanden. Auf der Eigenschaftenseite **Objekte** gelangen Sie über die Schaltfläche **Datentabellen…** in den Dialog **Datentabellen verwalten**. Nur im erweiterten Modus ist es

möglich, neue Datentabellen anzulegen. Im Bild unten wurden die Datentabellen **Employees** und **Contracts** durch Klick auf  neu angelegt.

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



Die Spalte **Index** besitzt eine zentrale Bedeutung, denn der Zugriff auf die Inhalte der Datenfelder innerhalb eines Datensatzes geschieht ausschließlich über den Index. Wenn Sie die Reihenfolge der Felder ändern, nachdem bereits Programmcode existiert, dann müssen Sie den Programmcode für den Datenfeldzugriff ebenso anpassen.

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

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

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

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

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

In dem Beispiel wird eine Verknüpfung zwischen den Tabellen **Employees** und **Contracts** durch die Angabe von **Employees:IdentNr** in der Spalte **Beziehung** vereinbart.

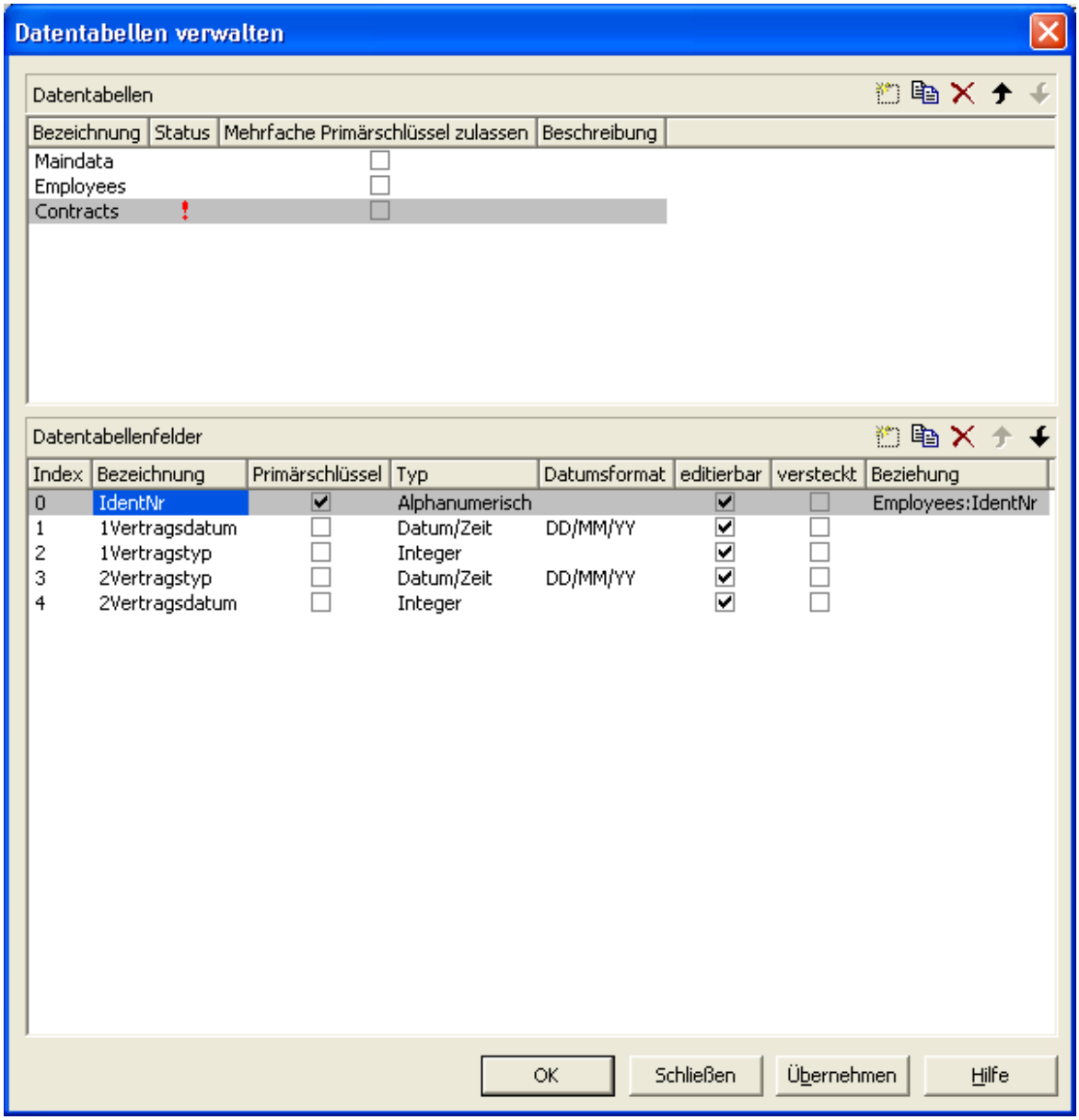


Tabelle Employees:

IdentNr	Nachname	Abteilung
1	Müller	Forschung und Entwicklung
2	Schmidt	Verwaltung

Tabelle Contracts:

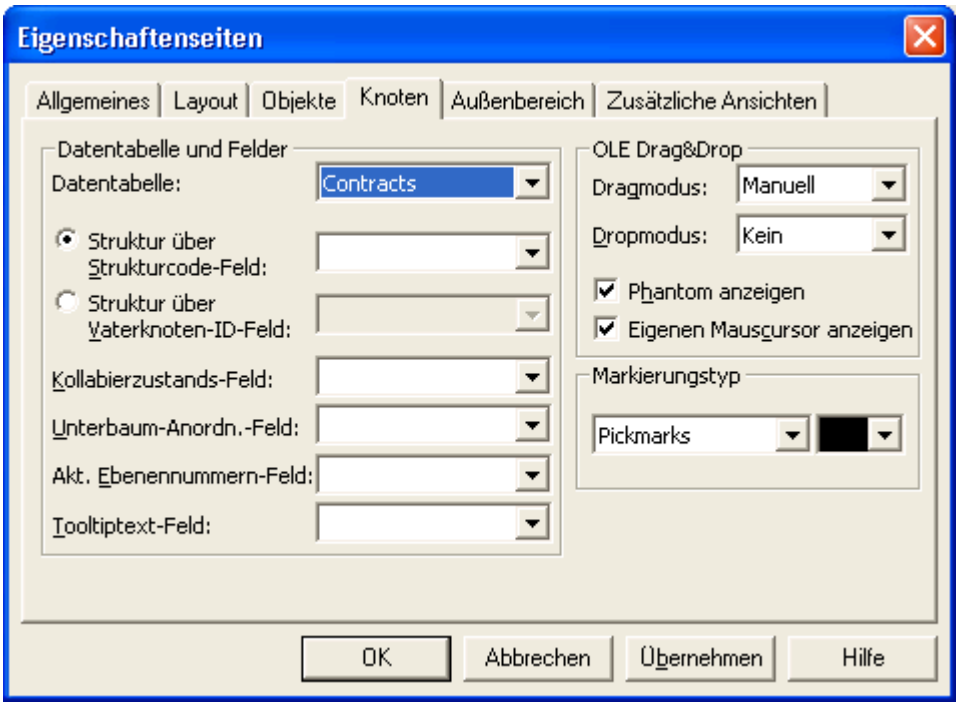
ID	Vertragsdatum	Gehaltsstufe	MitarbeiterIdentNo
1	1996/08/01	3	1
2	2006/06/01	4	1

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

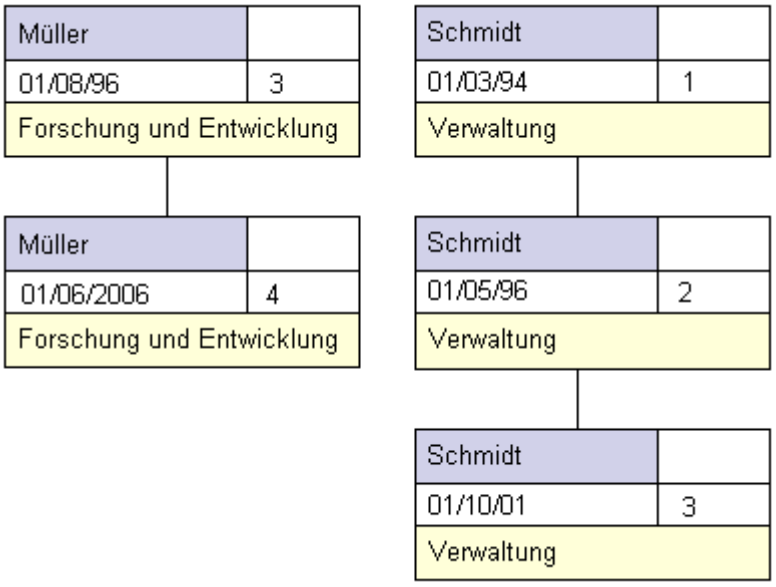
Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByName("Employees")

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

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

VcTree1.EndLoading
```

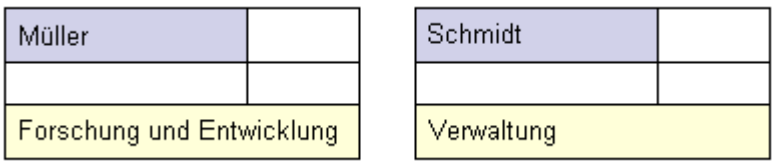
In Abhängigkeit davon, welche Tabelle unter **Datentabelle** bei der Eigenschaftenseite **Knoten** angegeben wird, ergibt sich eine andere Basis, aus der die grafische Darstellung der Knoten hergeleitet wird. Beim interaktiven Anlegen von Knoten ist es diejenige Tabelle, der automatisch neue Datensätze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



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



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



In der VARCHART XTree Version 4.0 sind aufgrund der erweiterten Datentabellen eine Reihe neuer Objekttypen entstanden, die bestehende Objekttypen ablösen werden. Aus Gründen der Kompatibilität sind in dieser Version die bisherigen Objekttypen noch enthalten. Für neue Anwendungen und beim Aktualisieren von bestehenden Anwendungen sollten nur noch die neuen Objekttypen eingesetzt werden.

Bisher	Ab Version 4.0
VcDataDefinition	VcDataTable
VcDefinitionTable	VcDataTableFieldCollection
VcDefinitionField	VcDataTableField
	VcDataRecord

> Datensätze anlegen und ändern

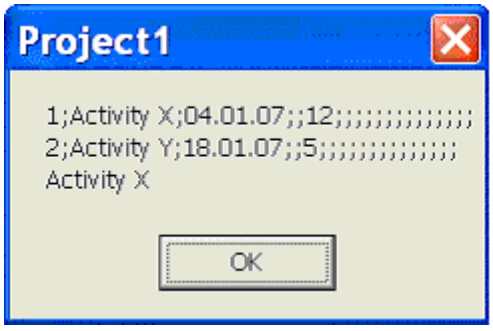
Nach der Definition der Datentabellenfelder können Sie einer Tabelle über die API Datensätze hinzufügen. Es gibt zwei Wege, auf denen die Datensätze mit Daten gefüllt werden können. Der normale und empfehlenswerte Weg besteht in der Vereinbarung eines Arrays vom Datentyp Variant, wobei dessen Elementeanzahl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

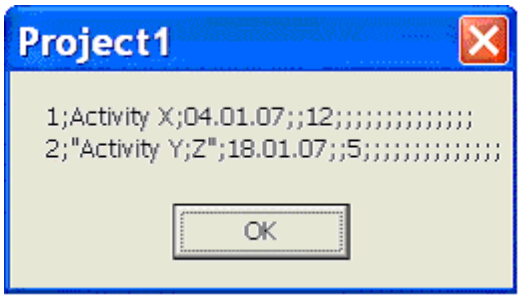
Code-Beispiel

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

'...

Dim dataRec1 As VcDataRecord
Dim dataRec2 As VcDataRecord

Dim content As String

VcTree1.ExtendedDataTablesEnabled =  True
Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection

ReDim dataRecVal(dataTable.DataTableFieldCollection.Count)

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

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

VcTree1.EndLoading

'...
```

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

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

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

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

'...

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

Erzeugte Ausgabe:





3.4 Datumsangaben und Zeitumstellung

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

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

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

Datumsangaben in der betroffenen Vergangenheit hier systembedingt falsch interpretiert.

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

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

3.5 Ereignisse

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

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

> Rückgabewerte

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

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

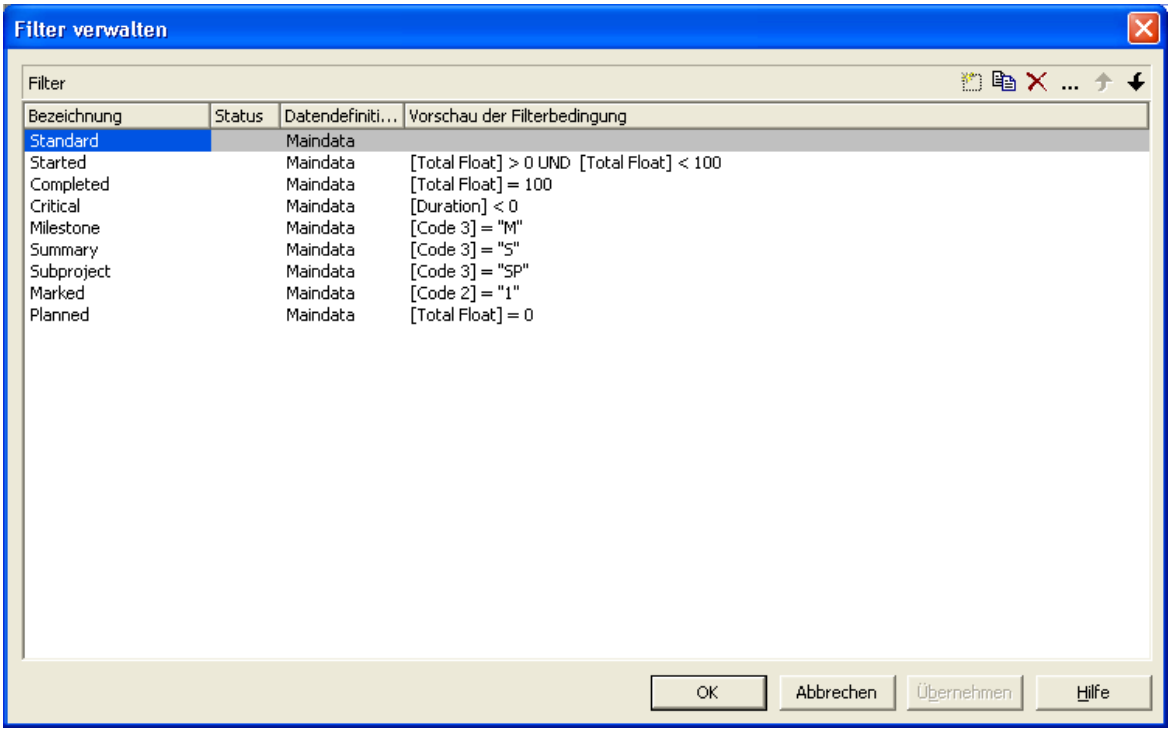
3.6 Filter

Ein Filter enthält Bedingungen für Knoten. Mit Hilfe eines Filters können Sie alle Knoten, die die Filterbedingungen erfüllen, auswählen, um sie beispielsweise grafisch hervorzuheben.

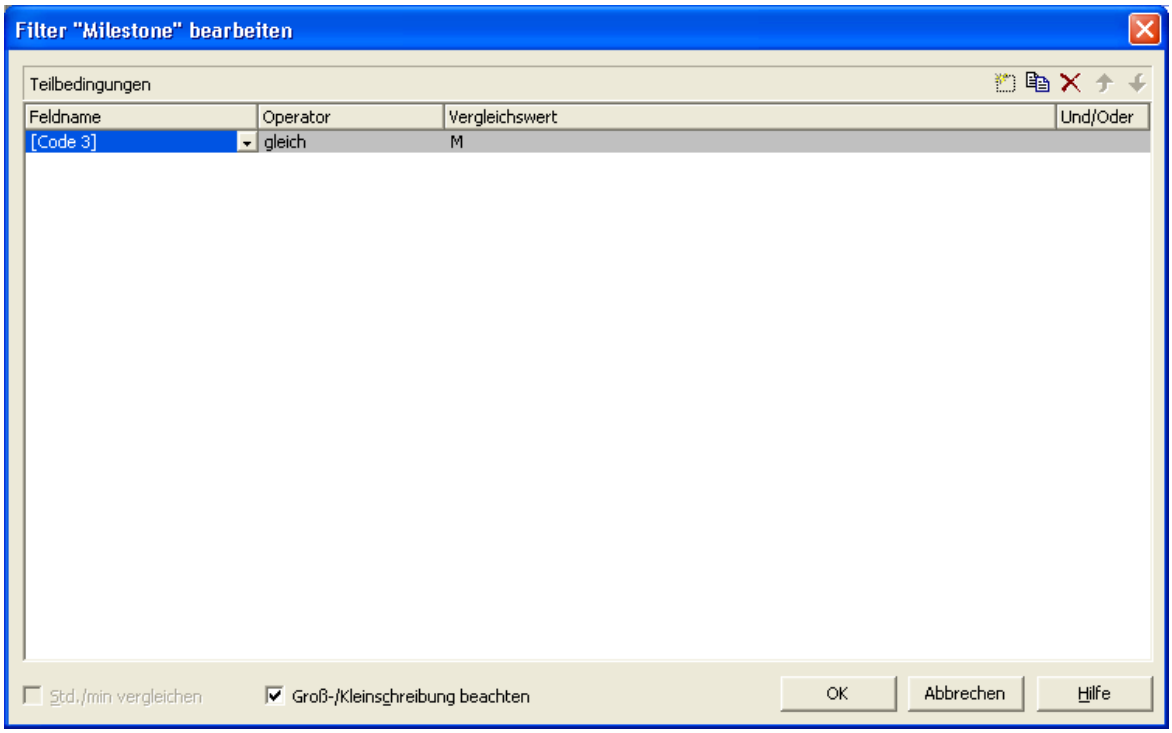
Wenn Sie einen Filter anwenden, werden die Informationen des Datensatzes eines Knotens mit den Kriterien aus dem Filter verglichen. Es werden die Knoten ausgewählt, die die Filterkriterien erfüllen. Beispielsweise können Sie den Filter "Alle Knoten von Abteilung A" definieren.

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

Sie erreichen das Dialogfeld **Filter verwalten** über die Eigenschaftenseite **Objekte**. Im Dialogfeld **Filter verwalten** können Sie Filter umbenennen, kopieren, löschen, bearbeiten oder neu definieren.



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


Filter "Milestone" bearbeiten
Teilbedingungen
Feldname
Operator
Vergleichswert
Und/Oder
[Code 3]
gleich
M
Std./min vergleichen
Groß-/Kleinschreibung beachten
OK
Abbrechen
Hilfe

3.7 Grafikformate

VARCHART unterstützt die folgenden Grafikformate, was für den Export von Grafiken für Bedeutung ist, vor allem für die Aufrufe **VcTree1.Show-GraphicsExportDialog** und **VcTree1.ExportGraphics**.

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

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

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

> WMF (Windows Metafile Format)

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

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

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

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

Formatbeschreibung siehe:

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

Beschränkungen des Formats siehe:

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

> EMF (Enhanced Metafile Format)

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

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

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

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

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

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

Formatbeschreibung siehe:

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

> EMF+ (Enhanced Metafile Format Plus)

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

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

Formatbeschreibung siehe:

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

> GIF (Graphics Interchange Format)

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

Die Unterart "Animated GIF" wird überhaupt nicht unterstützt.

> JPEG (Joint Photographic Experts Group)

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

> BMP (Windows Bitmap)

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

> TIFF (Tagged Image File Format)

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

> PNG (Portable Network Graphics)

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

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

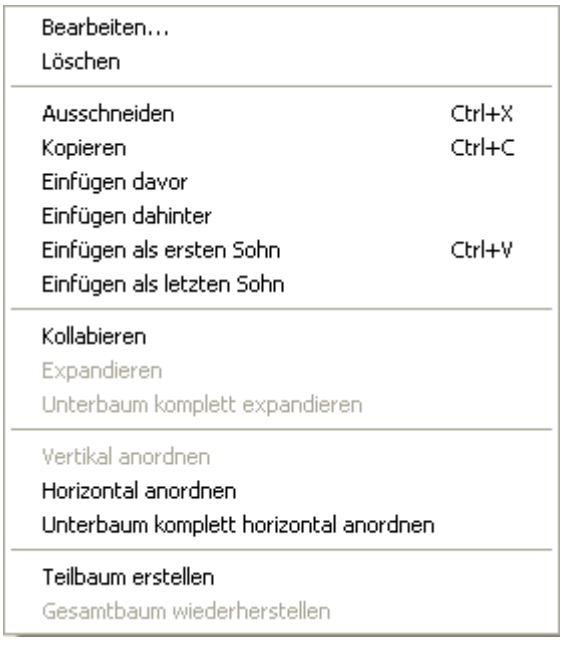
Formatbeschreibung siehe:
http://www.libpng.org/pub/png/spec/1.1/PNG-Contents.html

3.8 Horizontale/vertikale Anordnung

Wenn Sie Ihr Baum-Diagramm auf den Bildschirm holen, kann es sein, dass die Darstellung beim ersten Mal nicht optimal proportioniert wird. Durch eine geeignete Kombination horizontal und vertikal angeordneter Teilstrukturen lassen sich jedoch äußerst kompakte Baum-Diagramme erstellen.

- *Horizontale Anordnung:* Durch eine horizontale Anordnung lässt sich die Höhe eines Baum-Diagramms verringern. Alle Knoten einer Ebene werden dabei nebeneinander angeordnet. Die Ports (Anknüpfungspunkte) der Verbindungslinie liegen dann mittig am unteren Rand des Vaterknotens bzw. mittig am oberen Rand des Sohnknotens.
- *Vertikale Anordnung:* Durch eine vertikale Anordnung lässt sich die Breite des Baum-Diagramms verringern. Alle Knoten einer Ebene und deren Unterebenen werden dabei untereinander angeordnet. Die Ports der Verbindungslinie liegen dann in der unteren linken Ecke des Vaterknotens bzw. in der Mitte des linken Randes des Sohnknotens.

Um die Befehle zum horizontalen und vertikalen Anordnen von Bäumen zur Verfügung zu haben, markieren Sie einen Knoten und klicken Sie auf die rechte Maustaste. Es erscheint das folgende Kontextmenü, wobei jeweils nur die verfügbaren Befehle aktiviert sind.



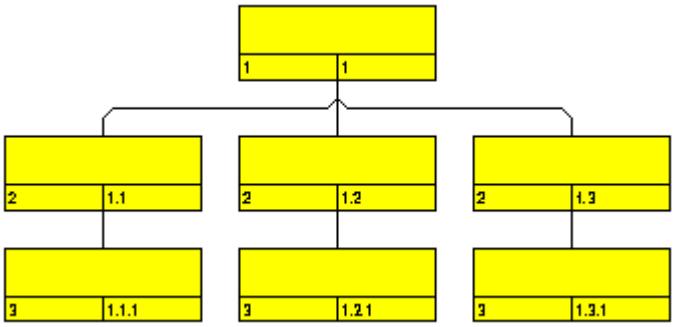
Um eine oder mehrere Teilstrukturen eines Baum-Diagramms horizontal anordnen zu lassen, markieren Sie den/die obersten Knoten jeder Teilstruktur und wählen Sie anschließend im Kontextmenü den Befehl **Horizontal anordnen**. Die Teilstrukturen der markierten Knoten werden dann horizontal

angeordnet, aber nur eine Ebene tief. Auf die Anordnungen in den nächst tieferen Ebenen hat der Befehl **Horizontal anordnen** keine Auswirkung.

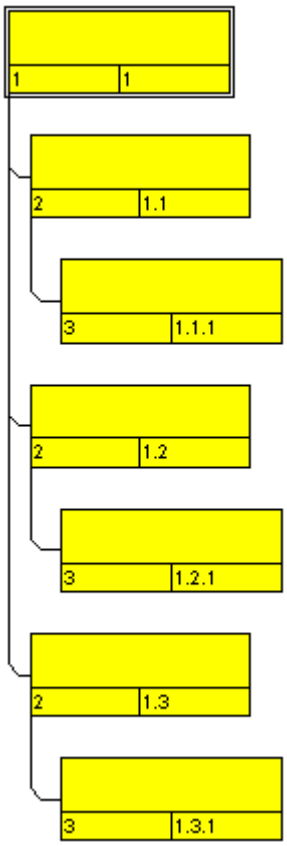
Mit dem Befehl **Unterbaum komplett horizontal anordnen** werden die Teilbäume unter den markierten Knoten vollständig, d. h. über alle Ebenen, horizontal angeordnet.

Mit dem Befehl **Vertikal anordnen** werden die Teilbäume unter den markierten Knoten vertikal angeordnet, und zwar beginnend beim höchsten ausgewählten Wurzelknoten.

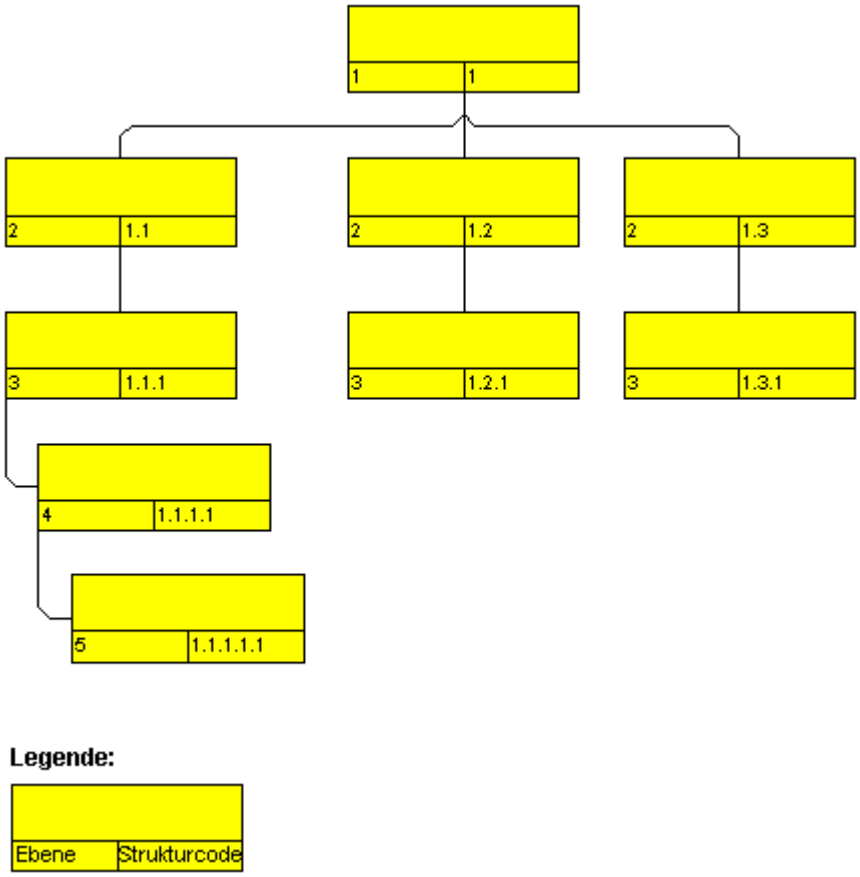
Hinweis: Falls ein Baum nicht vollständig vertikal angeordnet wird, prüfen Sie die vereinbarte maximale Baumhöhe auf der Eigenschaftenseite **Layout**. Die Anzahl der Ebenen bei der vertikalen Anordnung wird durch die Angabe unter **Max. Baumhöhe** begrenzt.



alle Ebenen horizontal angeordnet



alle Ebenen vertikal angeordnet



Beispiel für ein Baum-Diagramm mit horizontal und vertikal angeordneten Teilstrukturen

Hinweis: Änderungen in der Anordnung wirken sich auch auf kollabierte Teilbäume aus.

> Unterbaum-Anordnung in Datenfeld speichern

Aktivieren Sie auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Unterbaum-Anordnung in Feld**, um die Orientierung eines Teilbaums synchron in einem Datenfeld zu halten. Mögliche Inhalte des Datenfelds sind "0" (Teilbaum horizontal angeordnet) und "1" (Teilbaum vertikal angeordnet). Die horizontale Anordnung ist nur sichtbar, wenn der direkte oder indirekte Vaterknoten selbst eine vertikale Anordnung besitzt.

Alternativ können Sie die VcTree-Eigenschaft **ArrangementField** verwenden, um die Orientierung eines Teilbaums synchron in einem Datenfeld zu halten.

> Vertikal ab Ebene

Auf der Eigenschaftenseite **Layout** können Sie die Ebene angeben, ab der die Knoten vertikal angeordnet werden sollen. Aktivieren Sie dazu das Kontrollkästchen **Vertikal ab Ebene** und geben Sie an, ab welcher Ebene die

Knoten vertikal angeordnet werden sollen. Damit die Knoten tatsächlich vertikal angeordnet werden, muss die API-Methode **Arrange** aufgerufen werden.

Alternativ können Sie über die VcTree-Eigenschaft **FirstVerticalLevel** erfragen oder festlegen, ob ab einer bestimmten Ebene die Knoten vertikal angeordnet sind. Beim Wert "-1" ist diese Eigenschaft abgeschaltet.

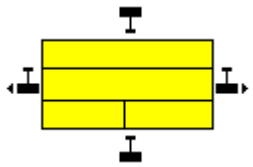
3.9 Knoten

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

> Knoten erzeugen

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

Nachdem Sie den ersten Knoten erzeugt haben, können Sie weitere Knoten nur durch Anlagern an einen bereits vorhandenen Knoten erzeugen. Wenn Sie den Cursor im Erzeugemodus in die Aura eines Knotens führen, verändert der Cursor seine Form und zeigt an, wo der neue Knoten angelegt würde (als Vater, Sohn oder Bruder des Bezugsknotens).



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

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

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

> Markieren von Knoten

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

Mögliche Alternativen sind:

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

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

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

> Knoten löschen

Einen oder mehrere Knoten können Sie löschen, indem Sie sie bei gedrückter Umschalt- oder Strg-Taste mit der linken Maustaste markieren und dann die rechte Maustaste drücken. Wählen Sie dann im Kontextmenü für Knoten den Befehl **Löschen** bzw. **Ausschneiden**.

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

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

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

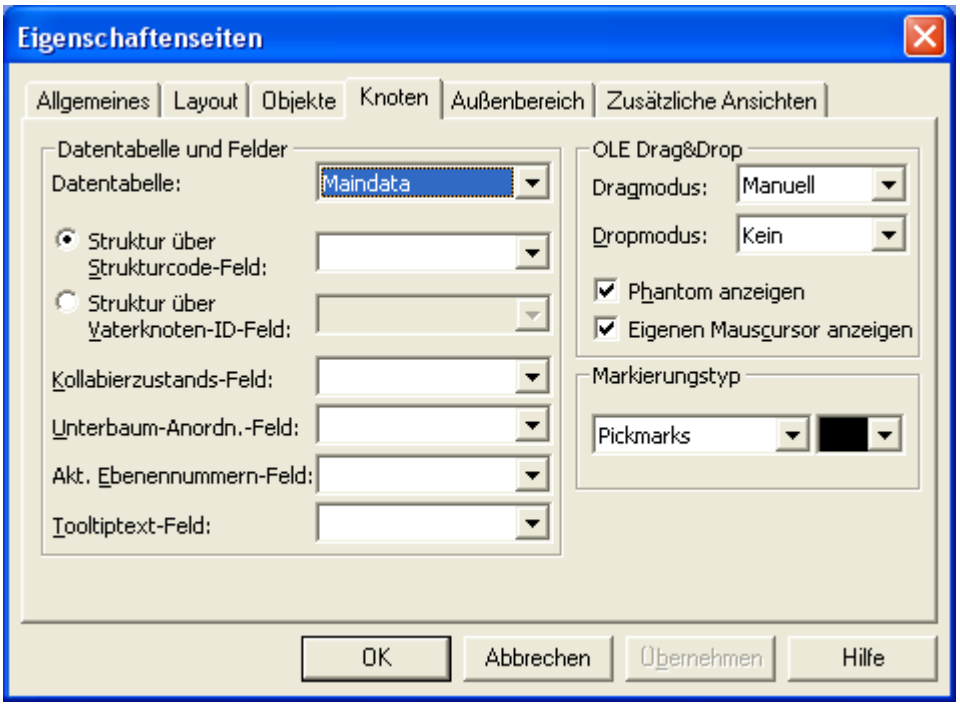
> Ereignisse

Auf folgende Ereignisse können Sie reagieren:

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

> Datenfelder für Baumstrukturen festlegen

Auf der Eigenschaftenseite **Knoten** können Sie Datenfelder für die Baumstruktur festlegen.



Entscheiden Sie zuerst, ob die Baumstruktur über einen Strukturcode oder über die ID des Vaterknotens festgelegt werden soll (**Struktur über Strukturcode-Feld** oder **Struktur über Vaterknoten-ID- Feld**). Legen Sie dann die entsprechenden Datenfelder fest:

- **Struktur über Strukturcode-Feld**: Wenn diese Checkbox aktiviert ist, können Sie das Datenfeld auswählen, das den Strukturcode festlegen soll.
- **Struktur über Vaterknoten-ID- Feld**: Wenn diese Checkbox aktiviert ist, können Sie das Datenfeld auswählen, das für die ID des Vaterknotens verwendet werden soll.

Legen Sie weiterhin folgende Datenfelder fest:

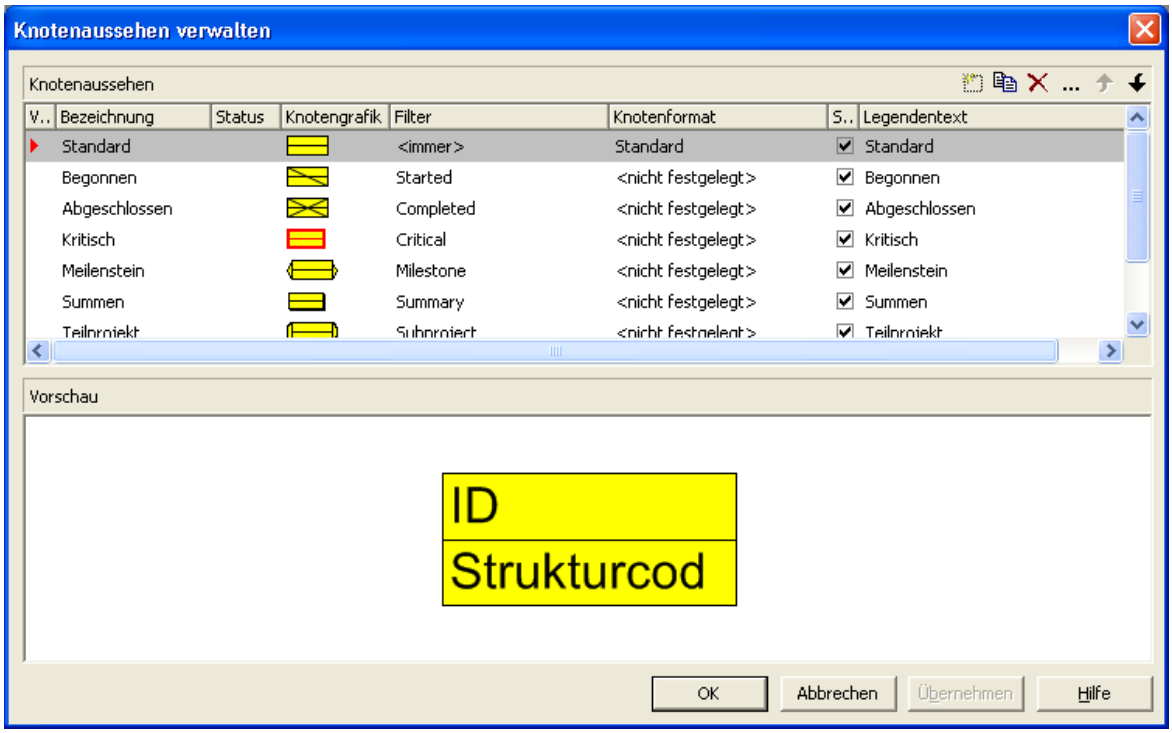
- **Kollabierzustands-Feld**: Wählen Sie hier ein Feld aus, um den Kollabierstatus eines Knotens synchron in dem hier gewählten Datenfeld zu halten. Mögliche Inhalte des Datenfelds sind "0" (Knoten expandiert) und "1" (Knoten kollabiert).
- **Unterbaum-Anordnungs-Feld**: Wählen Sie hier ein Feld aus, um die Orientierung eines Teilbaums synchron darin zu halten. Mögliche Inhalte des Datenfelds sind "0" (Teilbaum horizontal angeordnet) und "1" (Teilbaum vertikal angeordnet). Die horizontale Anordnung ist nur sichtbar, wenn der direkte oder indirekte Vaterknoten selbst eine vertikale Anordnung besitzt.
- **Aktuelles Ebenennummern-Feld:** Legen Sie hier fest, in welchem Datenfeld die Ebenennummer der Knoten abgelegt wird. Die Ebenennummern zählen von 0 an aufwärts. Zur Laufzeit können Sie nicht

die Ebene des Knotens verändern, indem Sie den Wert des Ebenennummer-Datenfeldes ändern.

3.10 Knotenaussehen

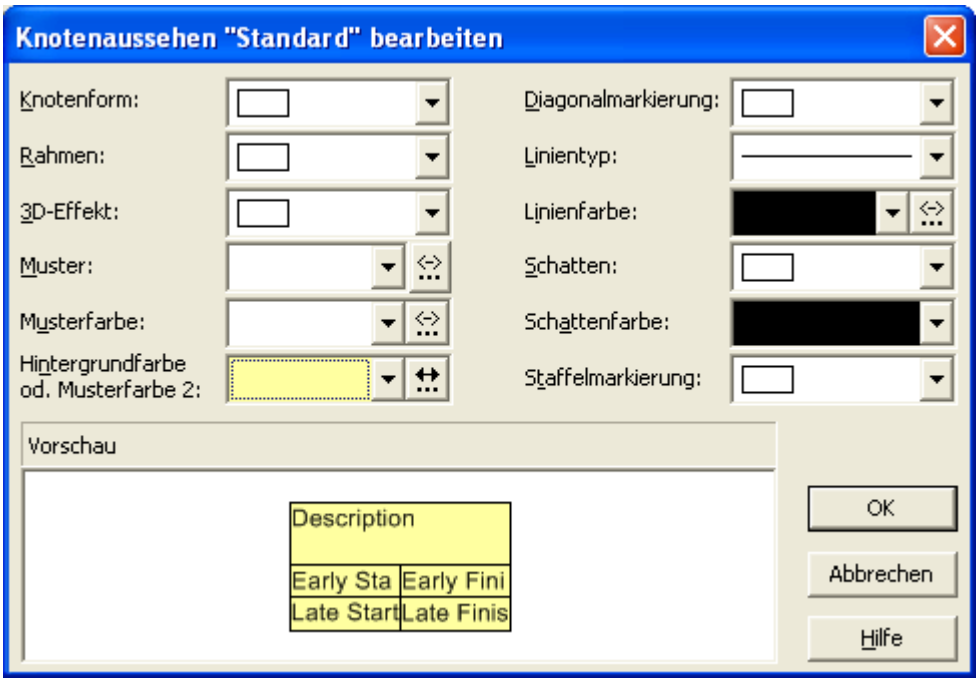
Das Aussehen von Knoten lässt sich in Abhängigkeit von deren Daten festlegen. Beispielsweise können alle Knoten der Abteilung A durch einen roten Hintergrund und eine doppelte schwarze Rahmenlinie gekennzeichnet werden, Knoten der Abteilung B durch einen blauen Hintergrund und eine einfache gelbe Rahmenlinie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

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



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

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



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

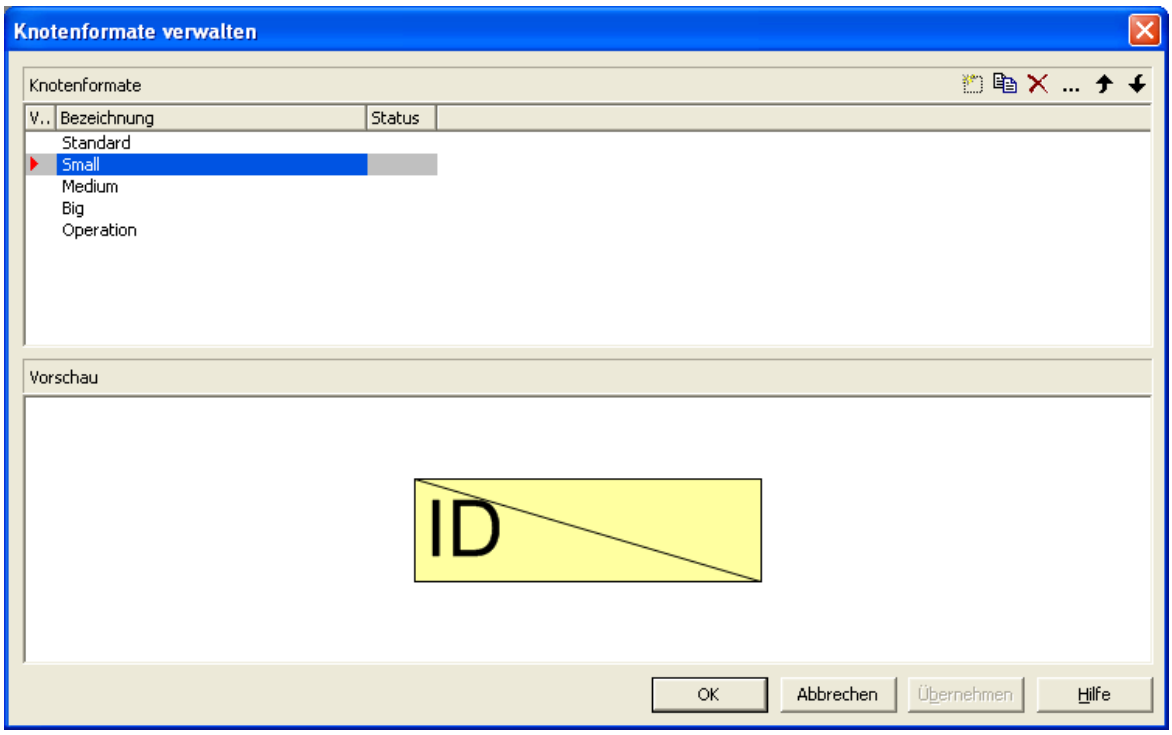
Sie können die Abarbeitungsreihenfolge der Knotenaussehen mit Hilfe der Pfeil-Schaltflächen verändern.

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

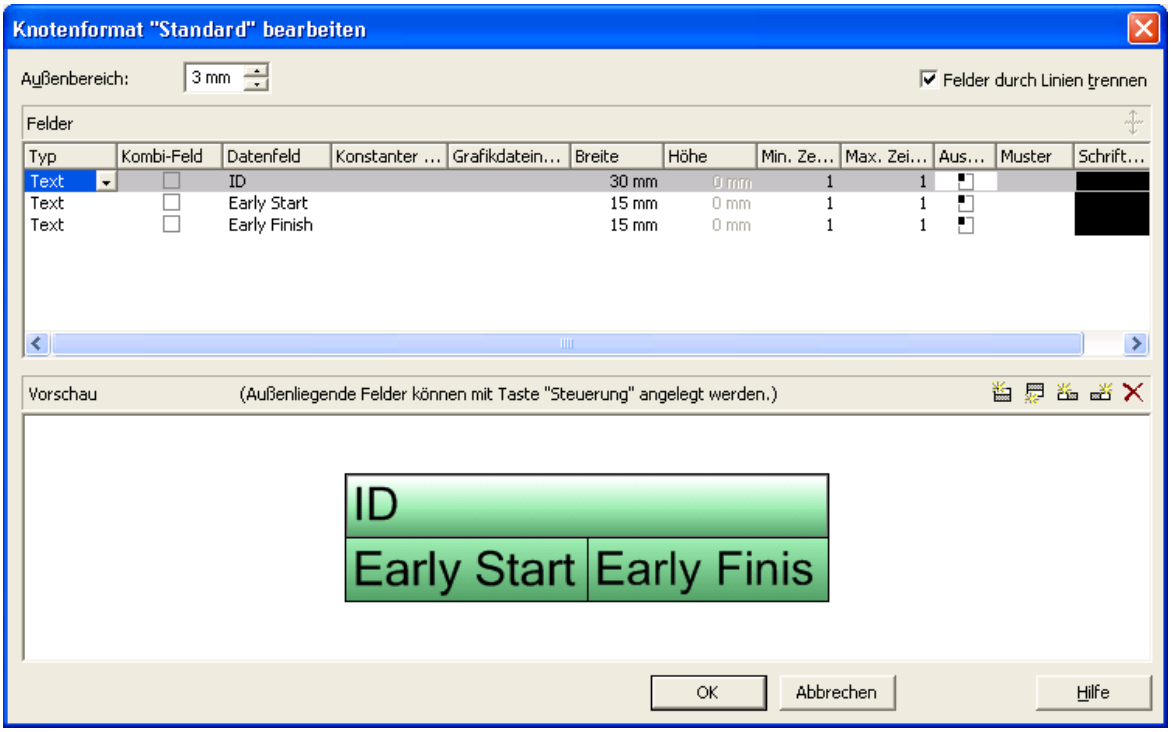
3.11 Knotenformat

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

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



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



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

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

> Datumsformat der Terminfelder von Knoten

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

> Darstellung von Grafiken in Knotenfeldern

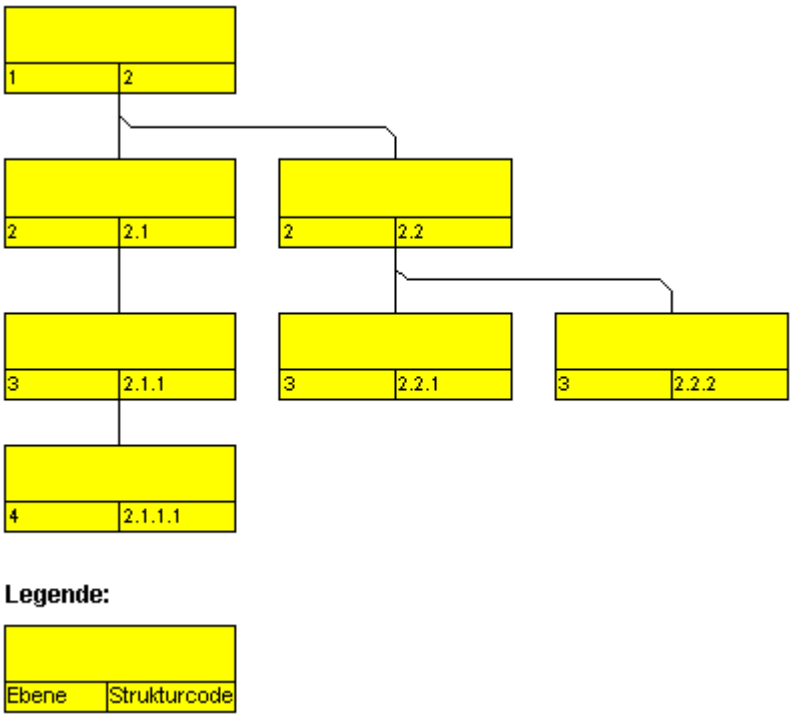
Sie können für ein Knotenfeld des Typs Grafik eine Grafikdatei wählen, indem Sie das Feld **Grafikdateiname** markieren, auf die dann erscheinende Schaltfläche **Grafikdatei auswählen** () klicken und anschließend im gleichnamigen Windows-Dialog eine Datei wählen.

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

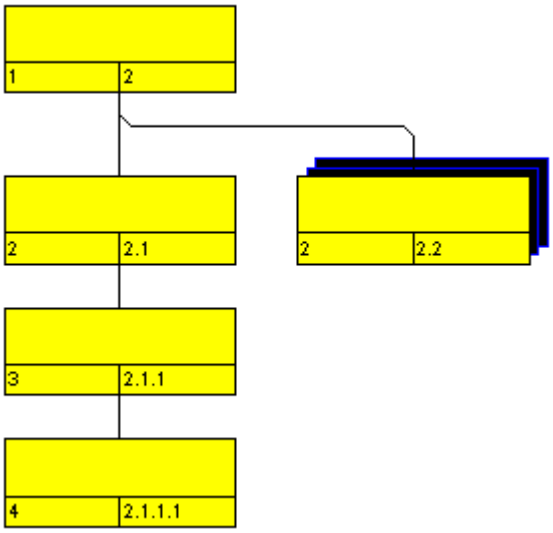
3.12 Kollabieren und Expandieren

Teilstrukturen von Baum-Diagrammen lassen sich kollabieren (wegklappen) und wieder expandieren (aufklappen). Jede beliebige Teilstruktur kann auf ihren obersten Knoten, den Gliederungsknoten, kollabiert werden. Als Gliederungsknoten wird jeweils der Wurzelknoten der Teilstruktur, die kollabiert werden soll, verwendet. Die kollabierte Teilstruktur kann ggf. wieder in ihre ursprüngliche Form expandiert werden.

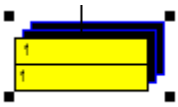
Durch das Kollabieren von Teilstrukturen lassen sich auch sehr komplexe Strukturen übersichtlich gestalten. Wenn Sie beispielsweise nur bestimmte Teilstrukturen präsentieren möchten, kollabieren Sie einfach alle anderen Teilstrukturen. Dadurch, dass die kollabierten Teilstrukturen erhalten bleiben, geht die Information über die Gesamtstruktur nicht verloren.



Expandierter Baum



In Form eines Gliederungsknotens kollabierte Teilstruktur



Vollständig kollabierter Baum

Mit dem Befehl **Kollabieren** aus dem Kontextmenü für Knoten können Sie die Teilstrukturen, die zu den markierten Wurzelknoten gehören, kollabieren. Die markierten Wurzelknoten verwandeln sich damit in Gliederungsknoten, die jeweils die nun verborgenen Teilstrukturen repräsentieren.

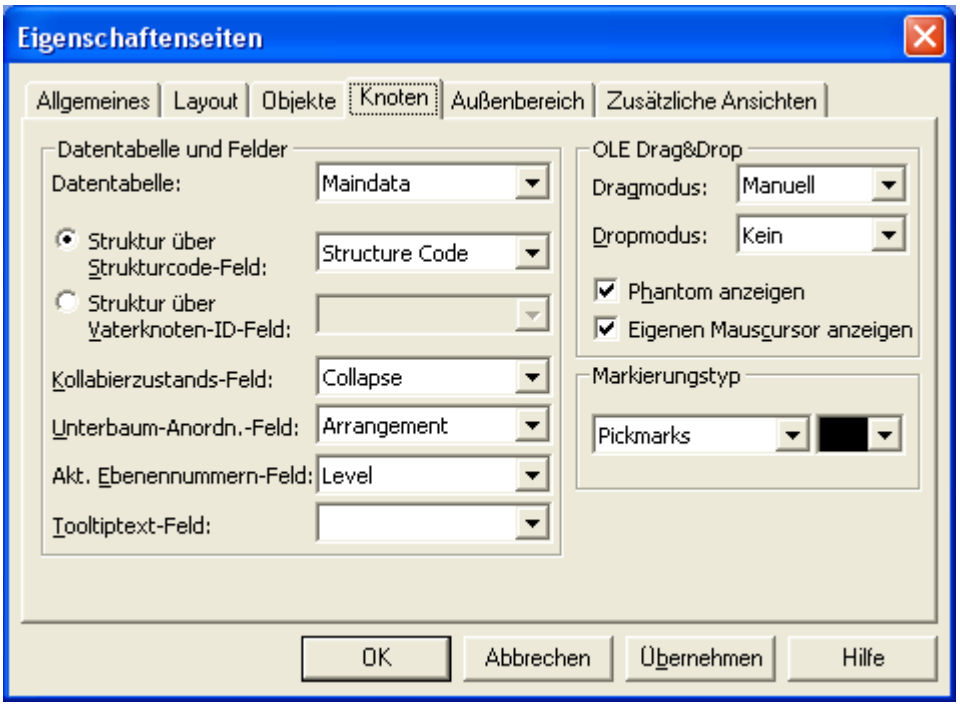
Mit dem Befehl **Expandieren** aus dem Kontextmenü für Knoten können Sie die Teilstrukturen wieder expandieren, die durch die markierten Gliederungsknoten repräsentiert werden. Es werden jeweils nur die kollabierten Knoten expandiert. Befinden sich in den Unterbäumen weitere kollabierte Knoten, bleiben diese kollabiert.

Mit dem Befehl **Unterbaum komplett expandieren** aus dem Kontextmenü für Knoten können Sie die kollabierten Teilstrukturen vollständig expandieren. Das heißt, dass hierbei auch alle kollabierten Knoten in den Unterbäumen expandiert werden.

> Kollabierzustand in Datenfeld speichern

Sie können den Kollabierzustand der Knoten in einem Datenfeld speichern.

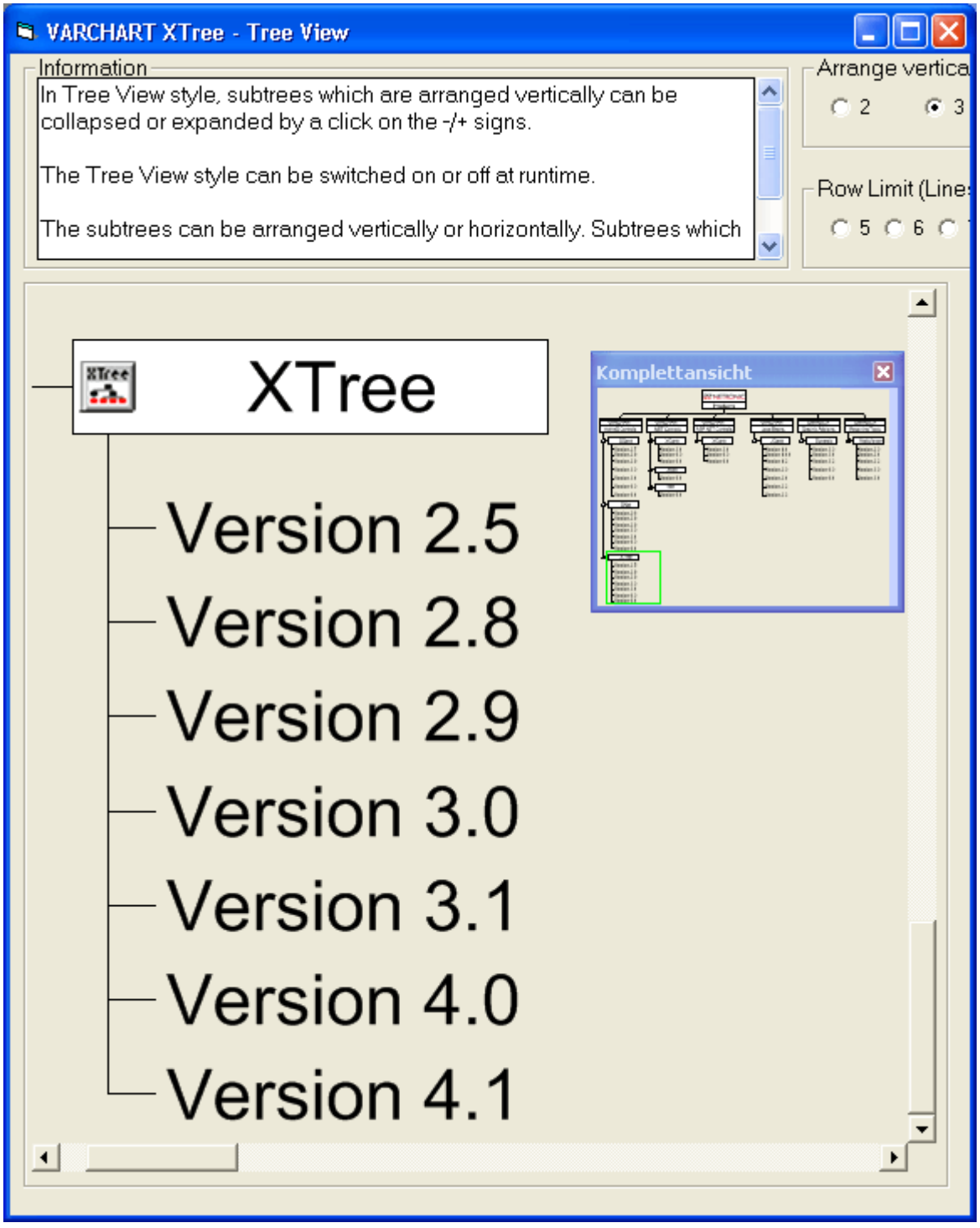
Aktivieren Sie dazu auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Kollabierzustands-Feld** und wählen Sie aus der Kombobox das gewünschte Datenfeld, z.B. "Kollabiert" aus.



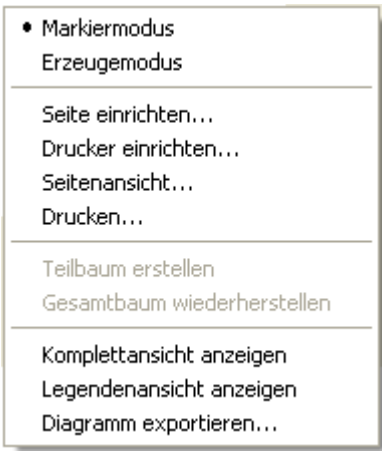
Ab jetzt wird der Kollabierstatus jedes Knotens synchron in dem Datenfeld "Kollabiert" gehalten. Mögliche Inhalte des Datenfelds sind "0" (Knoten expandiert) und "1" (Knoten kollabiert).

3.13 Komplettansicht (World View)

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



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

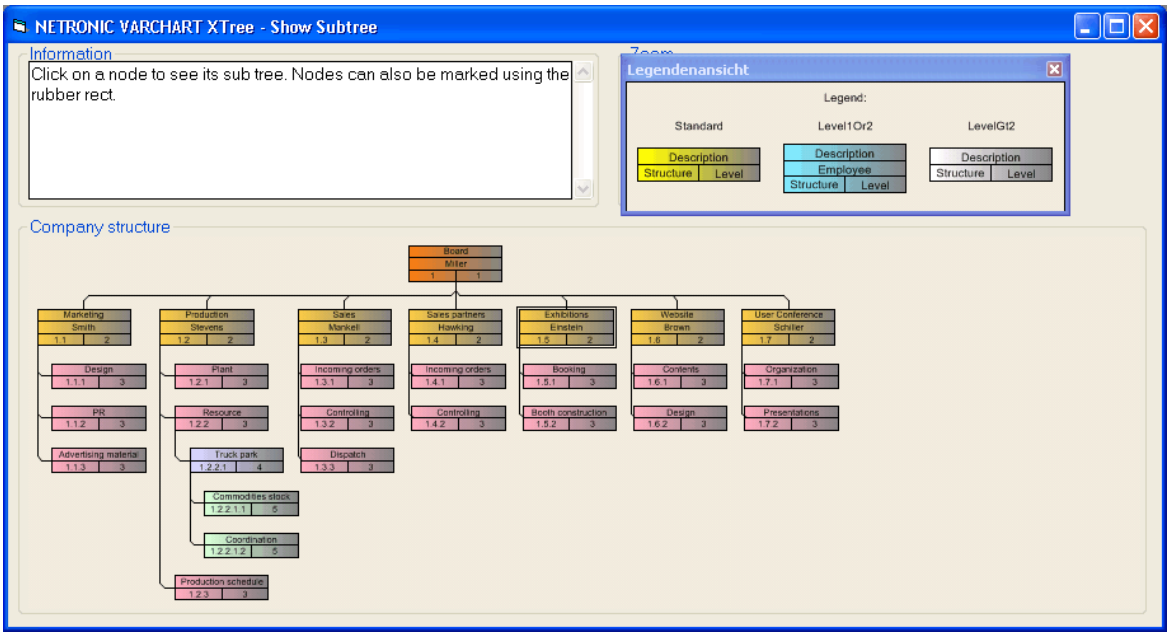


Auf der Eigenschaftenseite **Zusätzliche Ansichten** können Sie die Eigenschaften der Komplettansicht festlegen. Einzelheiten hierzu finden Sie im Kapitel **Eigenschaftenseiten und Dialogfelder**, Eigenschaftenseite **Zusätzliche Ansichten**.

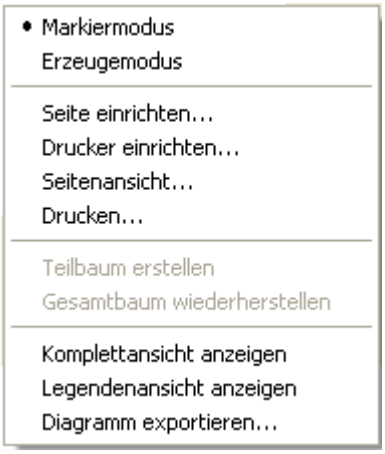
Alternativ können Sie die Optionen der Komplettansicht auch über die API-Eigenschaft **VcTree.VcWorldView** festlegen.

3.14 Legendenansicht (Legend View)

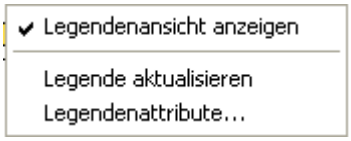
Die Legendenansicht ist ein zusätzliches Fenster zur Darstellung einer Legende auf dem Bildschirm. Das Aussehen der Legende wird festgelegt im Dialog **Legendenattribute**, der über die Eigenschaftenseite **Außenbereich** zu erreichen ist, oder über die Legendenattribute des Objektes **VcBorderBox**.



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



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



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

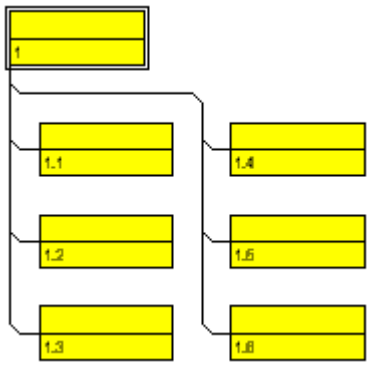
Eine Aktualisierung über das Menü kann notwendig sein, da nach Änderungen im Diagramm die Legende nicht automatisch aktualisiert wird. Werden also zum Beispiel Knoten hinzugefügt oder gelöscht, muss eine Aktualisierung entweder über das Kontextmenü oder durch Aus- und Einschalten der Legende durchgeführt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

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

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

3.15 Maximale Höhe des Baum-Diagramms

Die Gesamthöhe eines Baum-Diagramms (in Zeilen) kann begrenzt werden. Aktivieren Sie dazu das Kontrollkästchen **Max. Baumhöhe** auf der Eigenschaftenseite **Layout** und geben Sie die maximale Baumhöhe in Zeilen an. Diese Einstellung wirkt sich nur bei vertikaler Anordnung aus. Falls in einem vertikal angeordneten Ast mehr Ebenen vorhanden sind, erfolgt ein Umbruch, und ein neuer Ast wird am Vaterknoten erzeugt.



Vertikal angeordnete Baumstruktur, maximale Höhe: 4 Zeilen

Sie können die maximale Höhe einer Baumstruktur auch über die VcTree-Eigenschaft **RowLimit** erfragen oder festlegen.

3.16 OLE Drag & Drop

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

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

> OLE-Dragmodus

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

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

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

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

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

> OLE-Dropmodus

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

Mögliche Alternativen sind:

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

> Phantom während eines OLE-Drag-Vorgangs anzeigen

Mit Hilfe des Kontrollkästchens **Phantom anzeigen** können Sie festlegen, ob während eines OLE-Drag-Vorgangs ein Phantom erscheinen soll oder nicht. Das Abschalten des Phantoms ist für Anwendungen gedacht, die beim Hineindraggen eines Objekts kein neues Objekt erzeugen, sondern beispielsweise den Knoten, auf dem dann ein Objekt fallen gelassen wird, nur neu attributieren.

> Eigenen Mauscursor anzeigen

Mit Hilfe des Kontrollkästchens **Eigenen Mauscursor anzeigen** können Sie festlegen, ob während eines OLE-Drag-Vorgangs der Mauscursor in der Zielkomponente gesetzt werden soll.

Bei OLE Drag & Drop ist es möglich, den Cursor in der Quellkomponente über das Ereignis **OLEGiveFeedback** zu setzen. Daher würde ein Setzen durch die Zielkomponente zu einem Flimmern der konkurrierenden Cursor führen. Über das Kontrollkästchen **Eigenen Mauscursor anzeigen** kann man dieses Verhalten beeinflussen. Außerdem können bei eingeschaltetem Mauscursor und bei der auf **vcOLEDropManual** gesetzten Eigenschaft Objekte außerhalb der Anlagerungsstellen eines Knotens nicht fallen gelassen werden, während dies bei ausgeschaltetem Mauscursor möglich ist.

> Auftretende Ereignisse

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

Nach dem erfolgreichen Beginn einer OLE Drag & Drop-Operation wird das Ereignis **OLEStartDrag** auf dem Quellcontrol ausgelöst. Hiermit können Sie noch Änderungen am übergebenen **DataObject** vornehmen (d. h. andere

Datenformate hinzufügen) und auch die erlaubten Drop-Effekte (Kopieren und/oder Verschieben) festlegen. Das **DataObject** ist sozusagen die programmtechnische Verpackung des grafisch gezogenen Knotens. Beim Verschieben des Objekts werden dann bei dem jeweiligen Control unter dem Mauscursor (Zielcontrol) **OLEDragOver**-Ereignisse ausgelöst. Hier kann der jeweils an der aktuellen Mausposition erlaubte Drop-Effekt auf Kopieren, Verschieben oder Verboten gesetzt werden.

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

Bitte beachten Sie, dass Quell- und Zielcontrol identisch sein können und dass Quell- oder Zielcontrol natürlich auch andere Komponenten als VARCHART-ActiveX-Komponenten sein können, die evtl. sogar außerhalb Ihrer Applikation liegen. Wenn Sie sichergehen wollen, dass Quell- und Zielcontrol nur festgelegte Controls Ihrer Applikation sind, dann können Sie beim **OLEStartDrag**-Ereignis ein eigenes Format mittels der **DataObject**-Methode **SetData** hinzufügen, das Sie über den Windows-API-Befehl **RegisterClipboardFormat** registriert haben, und dieses Format bei den **OLEDragOver**- und **OLEDragDrop**-Ereignissen auf dem Zielcontrol auf Vorhandensein mittels der **DataObject**-Methode **GetFormat** prüfen.

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

dataObject.SetData Empty, myClipFormat

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

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

3.17 Schreiben von PDF-Dateien

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

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

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

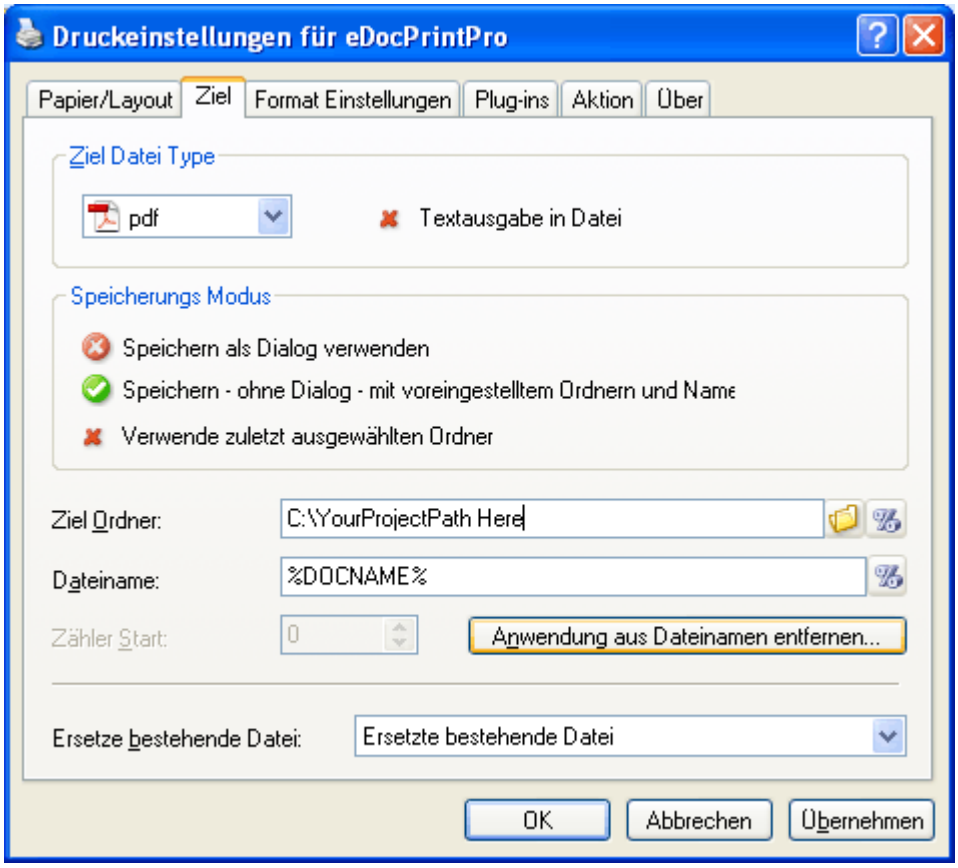
- Je nach Design der Anwendung kann es notwendig sein, den Treiber über die API so einzustellen, dass zur Laufzeit keine Dialoge und Messageboxen erscheinen. Dazu gehören insbesondere Dialoge zur Festlegung von Dateinamen und Pfaden.
- Soll das Setzen von Dateinamen und Pfad erst zur Laufzeit erfolgen,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

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

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

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

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



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

Code-Beispiel

```
VcTree1.Printer.PrinterName = "eDocPrintPro"
VcTree1.Printer.DocumentName = "abc.pdf"
VcTree1.PrintEx
```

Ganz wenige Druckertreiber erfordern eine andere Code-Sequenz:

Code-Beispiel

```
VcTree1.Printer.PrinterName = "Win2PDF"
VcTree1.PrintToFile "abc.pdf"
```

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

3.18 Sprachanpassung von Textausgaben

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

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

Code-Beispiel

```
VcTree1.EnableSupplyTextEntryEvent = True
```

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

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

Code-Beispiel

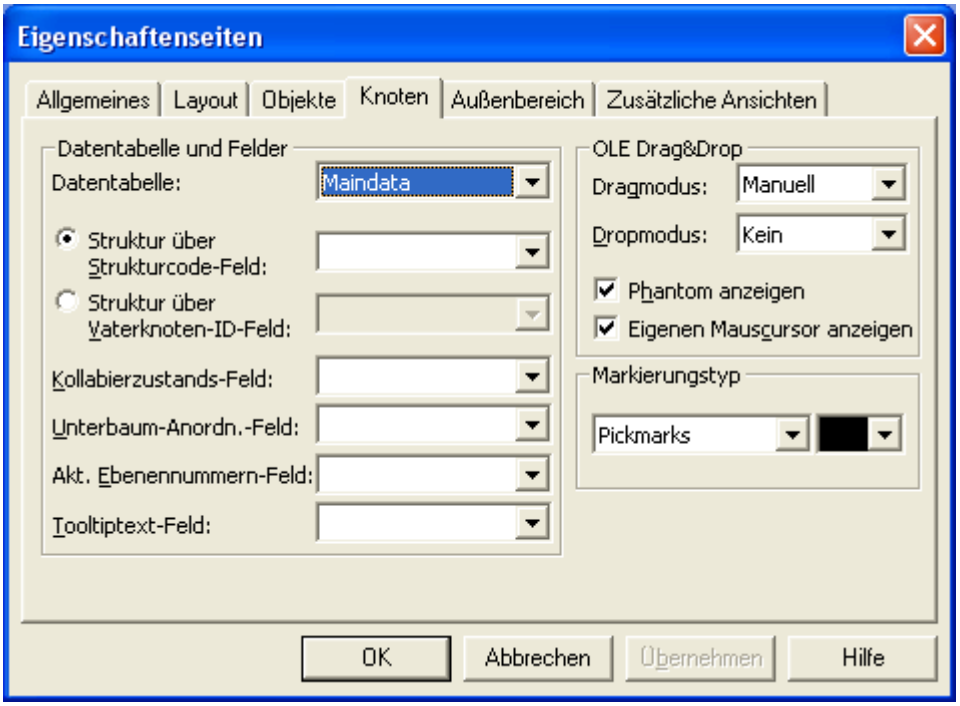
```
Private Sub VcTree1_OnSupplyTextEntry(ByVal controlIndex As _
                                VcTreeLib.TextEntryIndexEnum, _
                                TextEntry As String, _
                                returnStatus As Variant)
    Select Case controlIndex
        Case vcTXECtxmenCollapse
            TextEntry = "Knoten kollabieren"
        Case vcTXECtxmenExpand
            TextEntry = "Knoten expandieren"
        End Select
End Sub
```

3.19 Statuszeilentext

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

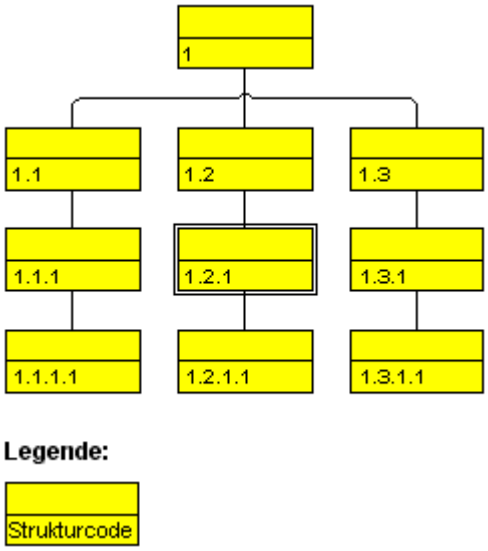
3.20 Struktur

Auf der Eigenschaftenseite **Knoten** können Sie die Struktur der Baum-Diagramme festlegen.

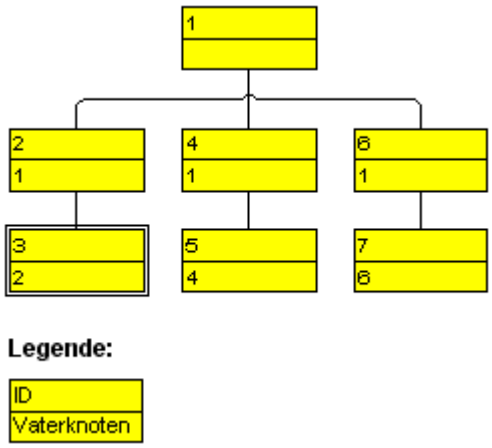


Dabei gibt es zwei grundsätzliche Alternativen:

1. **Struktur über Strukturcode-Feld:** Der Baum wird gemäß eines Strukturcodes aufgebaut. Sie können dann ein Datenfeld auswählen, das den Wert des Strukturcodes enthält. Als Trennzeichen wird ein Punkt verwendet.
2. **Struktur über Vaterknoten-ID-Feld:** Der Baum wird durch die ID des Vaterknotens jedes Knotens definiert. Sie können das Datenfeld wählen, das die ID des Vaterknotens enthält.



Beispiel für einen durch einen Strukturcode definierten Baum



Beispiel für einen durch die IDs der Vaterknoten definierten Baum

3.21 Tooltips zur Laufzeit

Sie können Tooltips verwenden, um Informationen über das mit der Maus berührte Objekt bereitzustellen. Mit Hilfe des Ereignisses **OnToolTipText** bzw. **OnToolTipTextAsVariant** können Sie die Texte aller zur Laufzeit erscheinenden Tooltips (Node, None) bearbeiten, z. B. um sie in unterschiedliche Sprachen zu übersetzen oder zu unterdrücken.

Das Ereignis **OnToolTipTextAsVariant** benötigen Sie, wenn Sie eine Skriptsprache benutzen, die keine Rückgabe von Strings erlaubt, z. B. VBScript.

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

Code-Beispiel

```
VcTree1.ShowToolTip = True
```

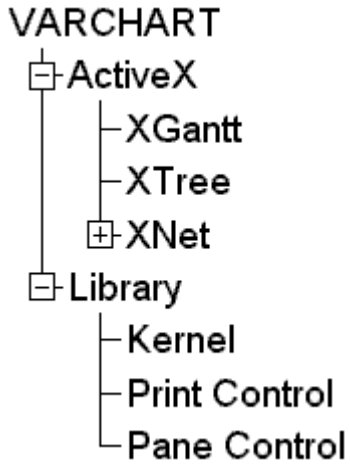
Alternativ können Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **OnToolTipText-Ereignisse** aktivieren.

Fangen Sie dann das Ereignis **OnToolTipText** bzw. **OnToolTipTextAs-Variant** ab und legen Sie fest, welcher Text erscheinen soll, oder ob an dieser Stelle kein Tooltip erscheinen soll.

3.22 TreeView-Stil

Knoten können im TreeView-Stil angeordnet werden. Bei dieser Ansicht werden vertikale Ebenen wie in TreeView-Controls (z. B. bekannt aus der Verzeichnisbaumansicht des Microsoft Explorers) mit Plus- oder Minus-Zeichen dargestellt. Dabei bedeutet ein Plus-Zeichen, dass der Knoten auf der gleichen Ebene kollabiert ist, und ein Minus-Zeichen, dass er expandiert ist. Die Zeichen werden nur bei Knoten angezeigt, die keine Blattknoten sind, d. h. Sohnknoten besitzen. Ein Mausklick auf eines der beiden Zeichen überführt den daneben stehenden Knoten in den jeweils anderen Kollabierzustand.

Ein Beispiel für den TreeView-Stil zeigt die folgende Abbildung:



Um die Knoten im TreeView-Stil anordnen zu lassen, aktivieren Sie auf der Eigenschaftenseite **Layout** das Kontrollkästchen **TreeView-Stil**.

3.23 Unicode-Zeichen

Damit zur Entwurfszeit in den Eigenschaftenseiten Unicode-Zeichen erscheinen, muss unter **Start / Systemsteuerung / Einstellungen / Anzeige / Darstellung** dem Bildelement **Fenster** ein geeigneter Schrifttyp zugewiesen werden.

Außerdem können trotz des Schrifttyps nur Zeichen angezeigt werden, deren zugehörige Sprache unter **Start / Systemsteuerung / Einstellungen / Region- und Spracheinstellungen** gewählt wurde.

Zur Laufzeit kann jedes Objekt in einer VARCHART-Komponente, das Texte enthält, Unicode-Zeichen darstellen, wenn ein geeigneter Schrifttyp bei der zugehörigen Eigenschaft **Font** gesetzt ist.

Den Kontextmenüs, Tooltips und Laufzeit-Dialogen kann über die Eigenschaft **DialogFont** des Objektes **DummyObject** ein Unicode-Schrifttyp zugewiesen werden.

Eine Übersicht über alle verfügbaren Fonts, die zumindest einen Teil aller Unicode-Zeichen enthalten, ist bei "Wazu Japan's Gallery of Unicode Fonts" zu finden (http:// www.wazu.jp/index.html). Eine weitere gute Informationsquelle für den Unicode-Standard bietet die Homepage des Unicode-Konsortiums (http:// www.unicode.org) sowie die Einführung in Unicode auf der GlobalDev-Homepage von Microsoft (http:// www.microsoft.com / globaldev / getwr / steps / wrg_unicode.mspx). Unter **Start / Programme / Zubehör / Systemprogramme / Zeichentabelle** kann man sich in Windows 2000 und XP darüber informieren, welcher installierte Schriftarttyp welche Zeichen enthält.

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

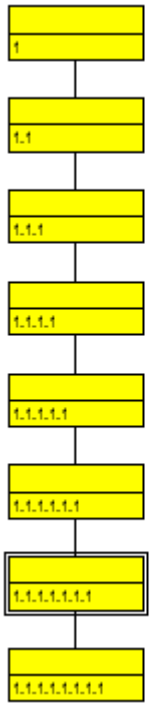
Hinweis: Die Entwicklungsumgebungen von Visual Studio 6 können in Quellcodedateien keine Unicode-Zeichen verwenden. Die interne Zeichendarstellung in Strings von Visual Basic 6 ist aber Unicode. Bei Visual C++ in Verbindung mit MFC muss man die Defines _UNICODE und UNICODE setzen, um Strings in Unicode zu verwenden. Ab Visual Studio .NET 2002 ist das Editieren von Quellcodedateien in Unicode-Kodierung möglich, beim Speichern muss man als Kodierung "Unicode" auswählen.

3.24 Vertikale Ebenen

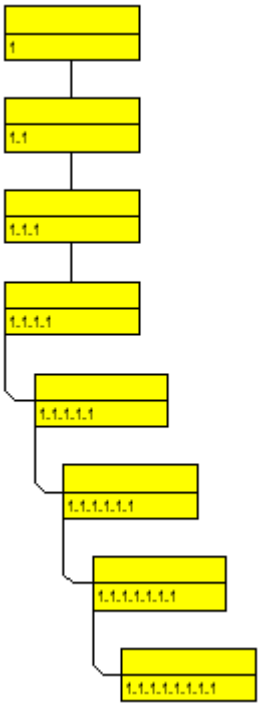
Bei der vertikalen Anordnung werden alle Knoten einer Ebene untereinander angeordnet. In vielen Fällen ist es sinnvoll, Knoten ab einer bestimmten Ebene vertikal anordnen zu lassen, um die Breite des Baum-Diagramms zu verringern.

Aktivieren Sie dazu auf der Eigenschaftenseite **Knoten** das Kontrollkästchen **Vertikal ab Ebene** und geben Sie an, ab welcher Ebene ggf. vertikal angeordnet werden soll.

Damit die Knoten tatsächlich vertikal angeordnet werden, muss die API-Methode **Arrange** aufgerufen werden.



vollständig horizontal angeordneter Baum



*nach dem Aufruf von **Arrange**: ab Ebene 4 vertikal angeordneter Baum*

3.25 Zuordnungstabellen

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

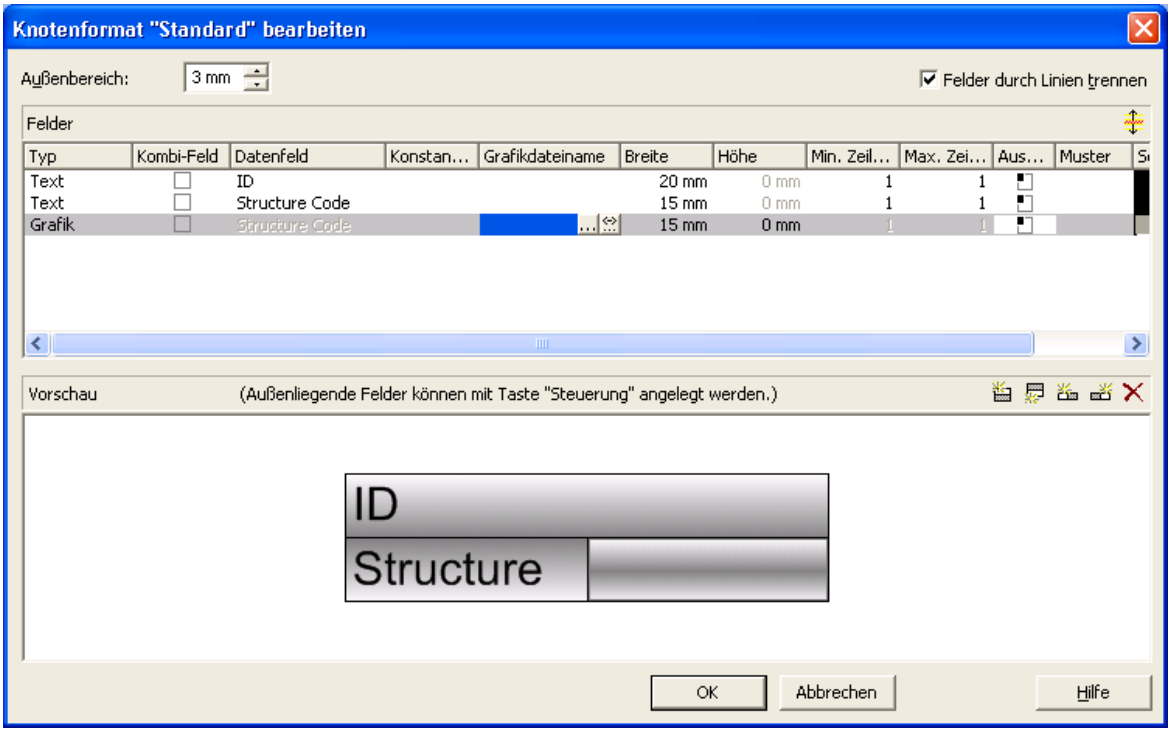
> Knotenaussehen datenabhängig festlegen

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

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

> Grafik eines Knotenformats datenabhängig festlegen

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



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

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

> Zuordnung einstellen

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

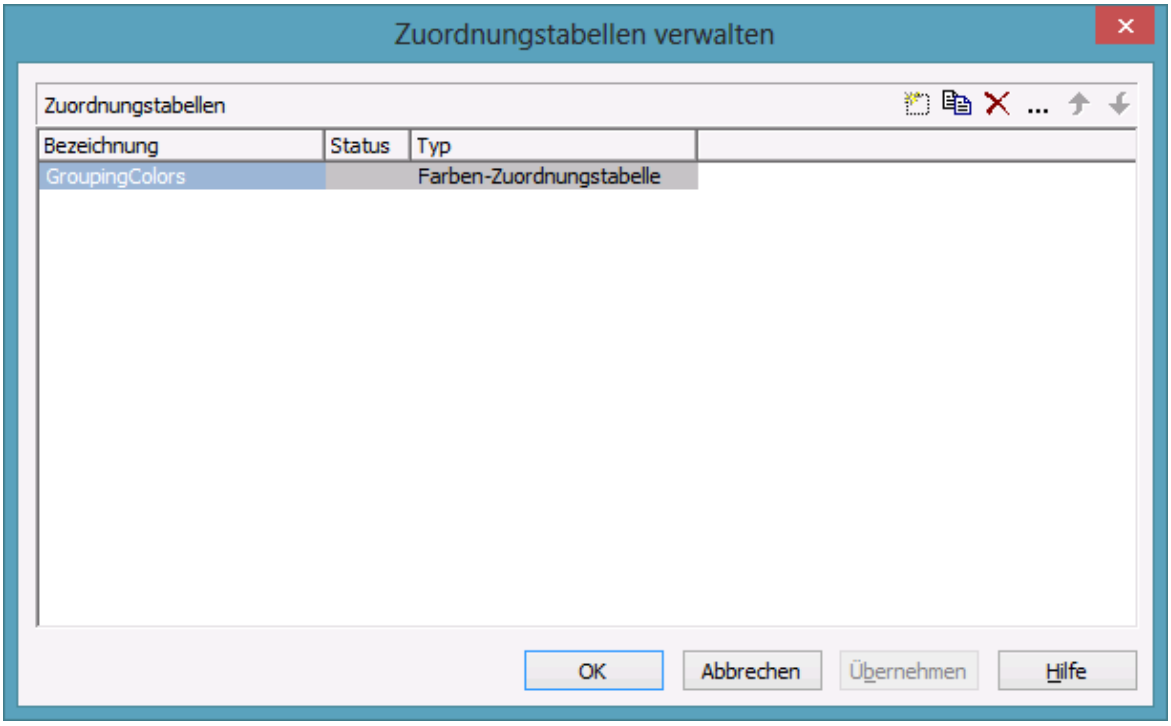
Wählen Sie dazu in der ersten Kombobox das **Datenfeld**, von dessen Einträgen die Grafikdatei des zu bearbeitenden Knotenformatfeldes bzw. die Hintergrundfarbe des zu bearbeitenden Knotenaussehens abhängen soll. Wählen Sie dann in der zweiten Kombobox die **Zuordnungstabelle**, die den einzelnen Datenfeldeinträgen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

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

> Zuordnungstabellen verwalten

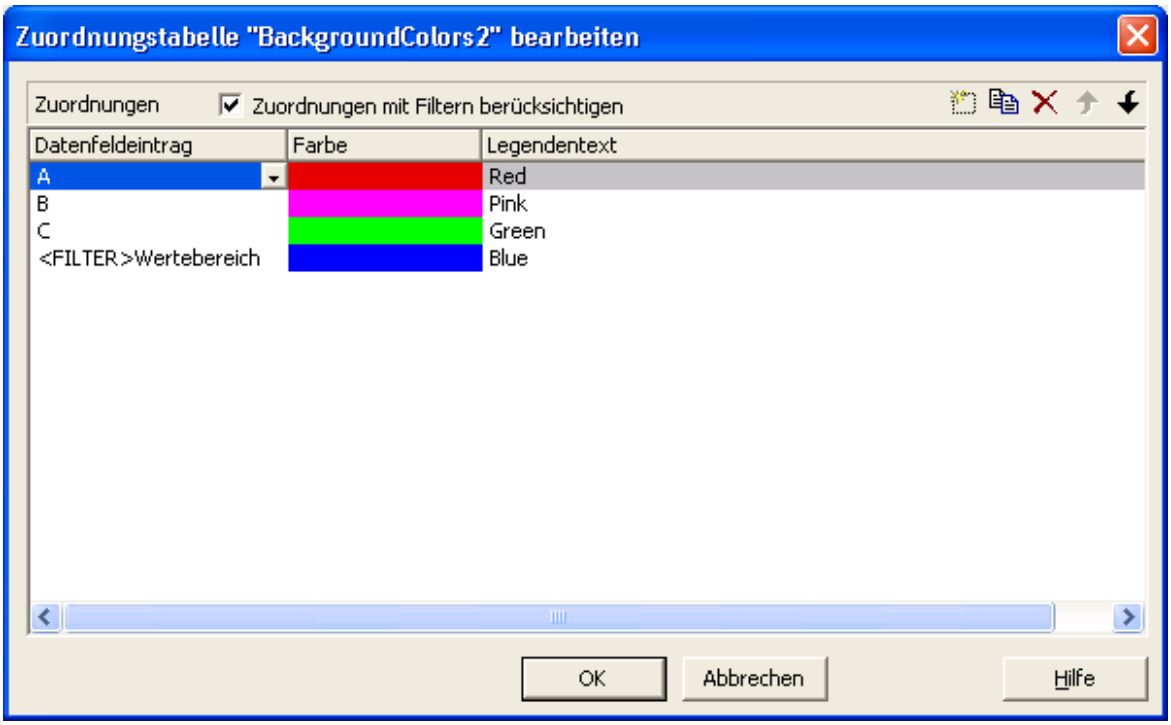
Im Dialogfeld **Zuordnungstabellen verwalten**, das Sie durch Klick auf die Schaltfläche **Zuordnungstabellen** oder über die Schaltfläche **Zuordnungstabellen** auf der Eigenschaftenseite **Objekte** erreichen, können Sie Namen und Typ einer Zuordnungstabelle durch direkte Eingabe verändern sowie über die entsprechenden Schaltflächen oben rechts im Fenster Zuordnungstabellen erstellen, kopieren, löschen oder bearbeiten.

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



> Zuordnungstabellen bearbeiten

Um eine Zuordnungstabelle zu bearbeiten, markieren Sie diese in der Tabelle und klicken Sie auf die Schaltfläche … oberhalb der Tabelle. Es erscheint das Dialogfeld **Zuordnungstabelle bearbeiten**.



In der **Zuordnungen**-Tabelle werden für jeden Schlüssel die entsprechenden Werte aufgelistet, in unserem Beispiel sind dies die Hintergrundfarbe und der Legendentext.

Über die Schaltflächen oben rechts können Sie Schlüssel (Zuordnungen) hinzufügen, kopieren oder löschen oder deren Reihenfolge verändern.

Wenn die Option **Zuordnungen mit Filtern berücksichtigen** ausgewählt ist, werden nicht nur die in der Liste der Datenfeldeinträge angegebenen festen Werte als Schlüssel berücksichtigt, sondern auch Filter, die aus der Dropdown-Liste ausgewählt werden können. Dadurch hängen die Ausführungswerte nicht mehr nur von einem konkreten Wert , sondern von komplexeren Kriterien ab.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie weitere Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

Einzelheiten zu den hier beschriebenen Dialogfeldern finden Sie im Kapitel "Eigenschaftenseiten und Dialogfelder".

> Anpassung der Zuordnungstabelle zur Laufzeit

Sie können die Zuordnungstabellen auch zur Laufzeit noch mit Hilfe der VcMap-Methoden anpassen. Damit geben Sie dem Anwender die Möglichkeit, Ihre Voreinstellungen über einen von Ihnen erstellten Dialog zu verändern.

4 Eigenschaftenseiten und Dialogfelder

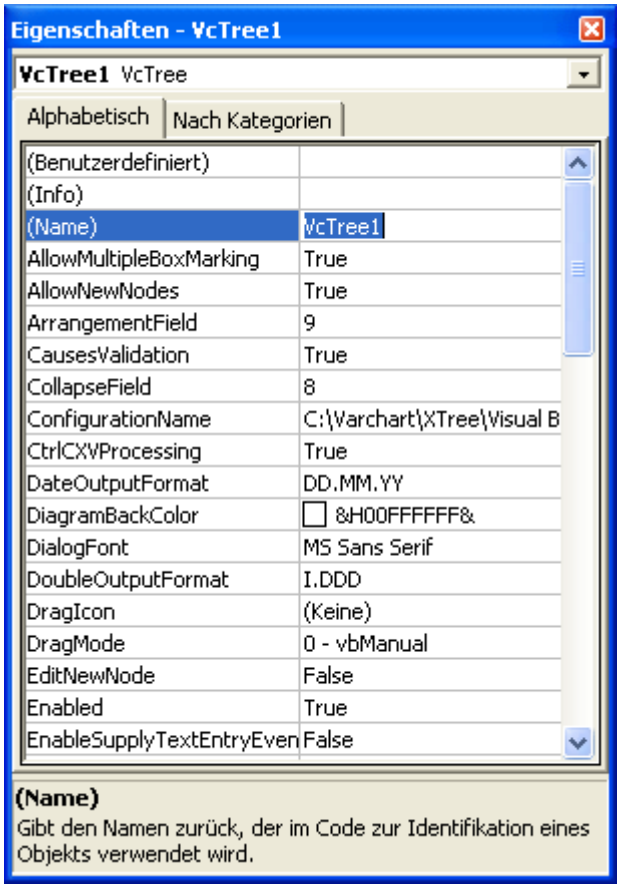
4.1 Allgemeines

Durch die Eigenschaftsseiten kann VARCHART XTree bereits zur Entwurfszeit konfiguriert werden. Es gibt zwei Möglichkeiten, zu den Eigenschaftsseiten zu gelangen:

- Drücken Sie die rechte Maustaste, wenn der Mauszeiger sich innerhalb des Steuerelements befindet, und wählen Sie im Kontextmenü den Befehl **Eigenschaften** aus.

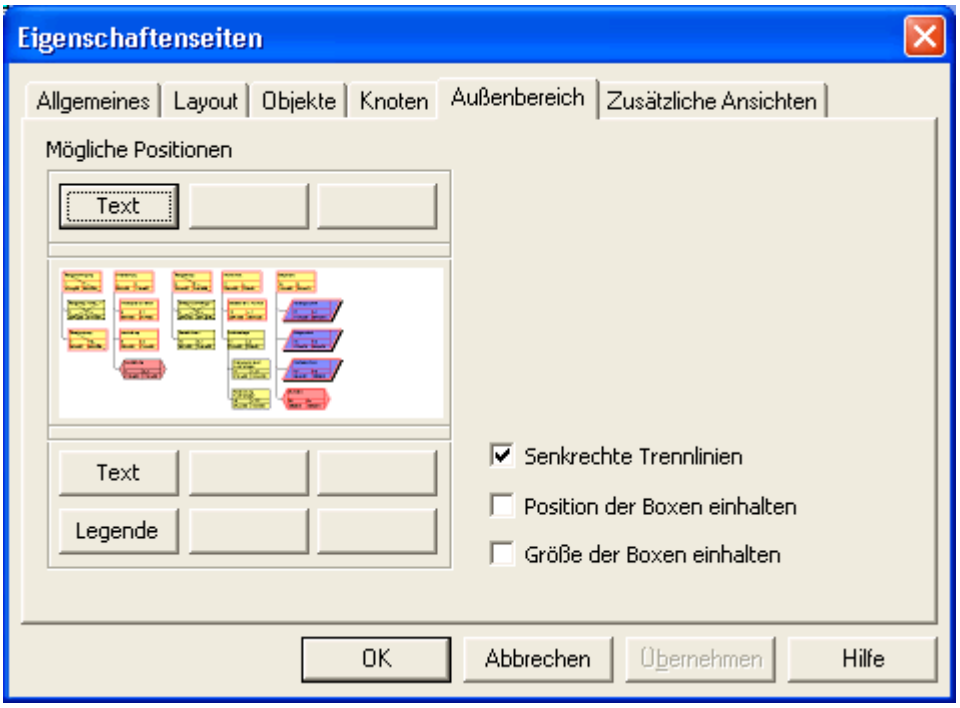
oder

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



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

4.2 Eigenschaftenseite "Außenbereich"



Mögliche Positionen

Oberhalb der Grafik stehen Ihnen drei und unterhalb der Grafik sechs Bereiche zur Verfügung, in denen Sie Texte, Grafiken oder eine Legende plazieren können. Jeder dieser Bereiche wird in diesem Dialog durch je eine Schaltfläche repräsentiert. Alle diese Bereiche werden nur in der Seitenansicht und im Ausdruck angezeigt. Klicken Sie auf eine der Schaltflächen ober- bzw. unterhalb der Grafik, um den Dialog **Texte, Grafiken und Legende festlegen** zu öffnen.

Senkrechte Trennlinien

Aktivieren Sie dieses Kontrollkästchen, wenn die Bereiche für Texte, Grafiken oder Legende durch senkrechte Trennlinien getrennt werden sollen.

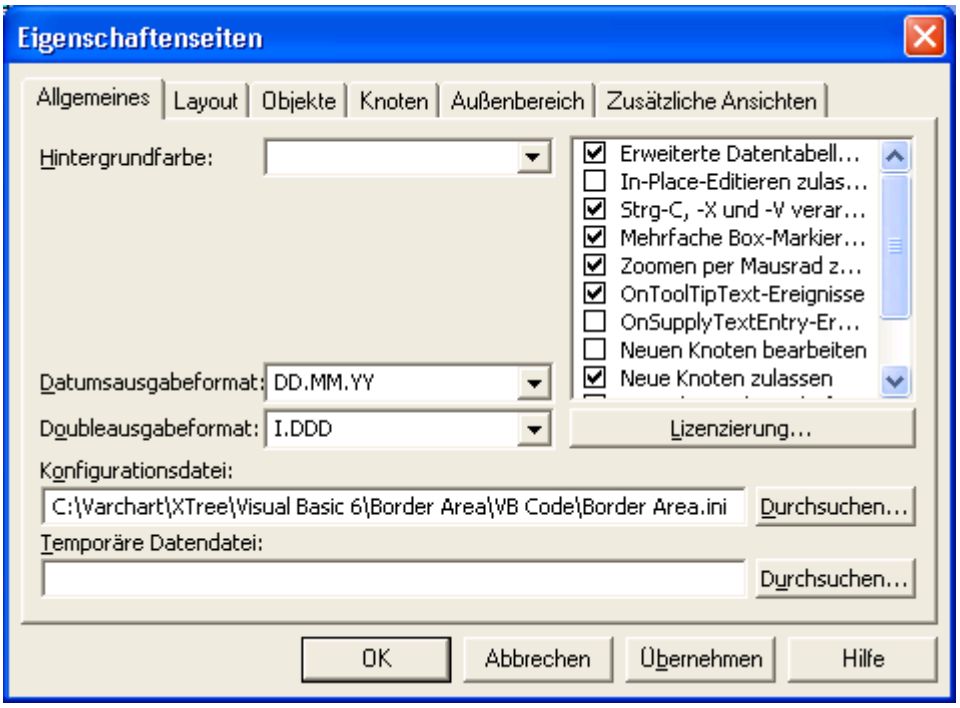
Position der Boxen einhalten

Aktivieren Sie dieses Kontrollkästchen, wenn die Position der Boxen möglichst genau eingehalten werden soll. Andernfalls wird der vorhandene Platz proportional auf die in der jeweiligen Zeile vorhandenen Elemente verteilt.

Größe der Boxen einhalten

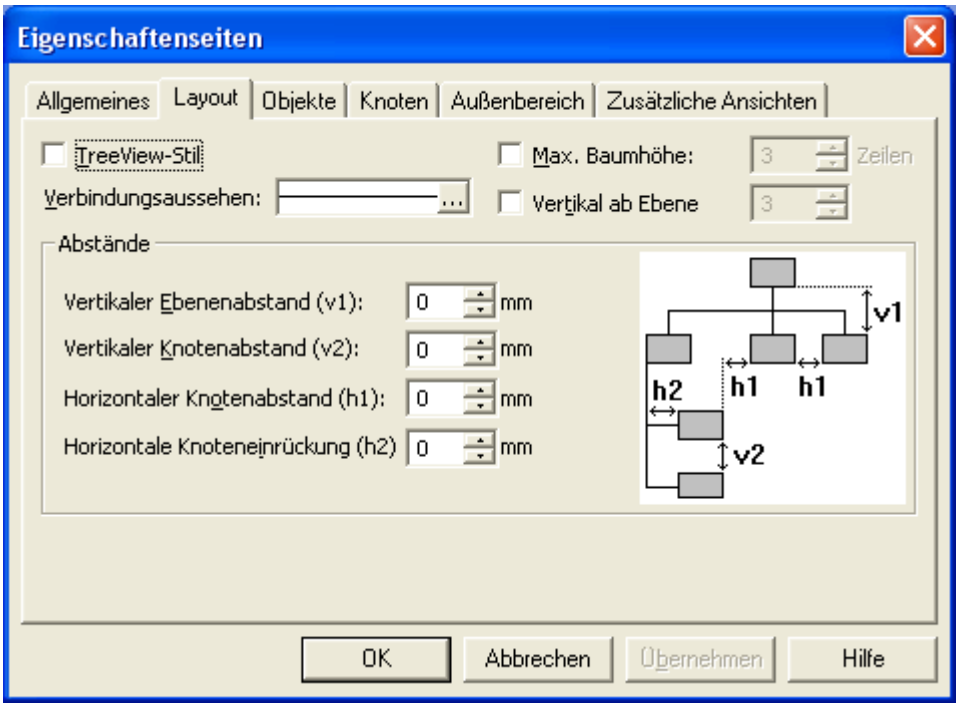
Aktivieren Sie dieses Kontrollkästchen, wenn die Größe der Boxen möglichst genau eingehalten werden soll. Gegebenenfalls wird das Diagramm vergrößert und/oder die Texte in den Boxen abgeschnitten.

4.3 Eigenschaftenseite "Allgemeines"



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

4.4 Eigenschaftenseite "Layout"



Auf dieser Eigenschaftenseite können Sie das Layout Ihres Diagramms festlegen.

TreeView-Stil

Aktivieren Sie dieses Kontrollkästchen, damit Knoten im TreeView-Stil angeordnet werden. Bei dieser Ansicht werden vertikale Ebenen wie in TreeView-Controls (z. B. bekannt aus der Verzeichnisbaumansicht von Microsoft Explorer) mit Plus- oder Minus-Zeichen dargestellt. Dabei bedeutet ein Plus-Zeichen, dass der Knoten auf der gleichen Ebene kollabiert ist, und ein Minus-Zeichen, dass er expandiert ist. Die Zeichen werden nur bei Knoten angezeigt, die keine Blattknoten sind, d. h. Sohnknoten besitzen. Ein Mausklick auf eines der beiden Zeichen überführt den daneben stehenden Knoten in den jeweils anderen Kollabierzustand.

Verbindungsaussehen

Hier wird Ihnen das aktuelle Verbindungsaussehen angezeigt. Um es zu ändern, klicken Sie auf die **Bearbeiten**-Schaltfläche. Sie gelangen dann in das Dialogfeld **Linienattribute**, in dem Sie Typ, Dicke und Farbe der Verbindungslinien festlegen können.

Max. Baumhöhe

Die Gesamthöhe eines Baum-Diagramms (in Zeilen) kann begrenzt werden. Wenn Sie dieses Kontrollkästchen aktivieren, können Sie hier die maximale Höhe der Baumstruktur vorgeben. Diese Einstellung wirkt sich nur bei vertikaler Anordnung aus. Falls in einem vertikal angeordneten Ast mehr Ebenen vorhanden sind, erfolgt ein Umbruch, und ein neuer Ast wird am Vaterknoten erzeugt.

Vertikal ab Ebene

Wenn dieses Kontrollkästchen aktiviert ist, werden die Knoten ab der Ebene, die Sie hier angegeben haben, vertikal angeordnet. Damit die Knoten tatsächlich vertikal angeordnet werden, muss die API-Methode **Arrange** aufgerufen werden.

Vertikaler Ebenenabstand (v1)

Hier können Sie den vertikalen Abstand zwischen zwei horizontal angeordneten Knotenebenen in Millimetern festlegen.

Vertikaler Knotenabstand (v2)

Hier können Sie den vertikalen Abstand zwischen zwei vertikal angeordneten Knoten in Millimetern festlegen.

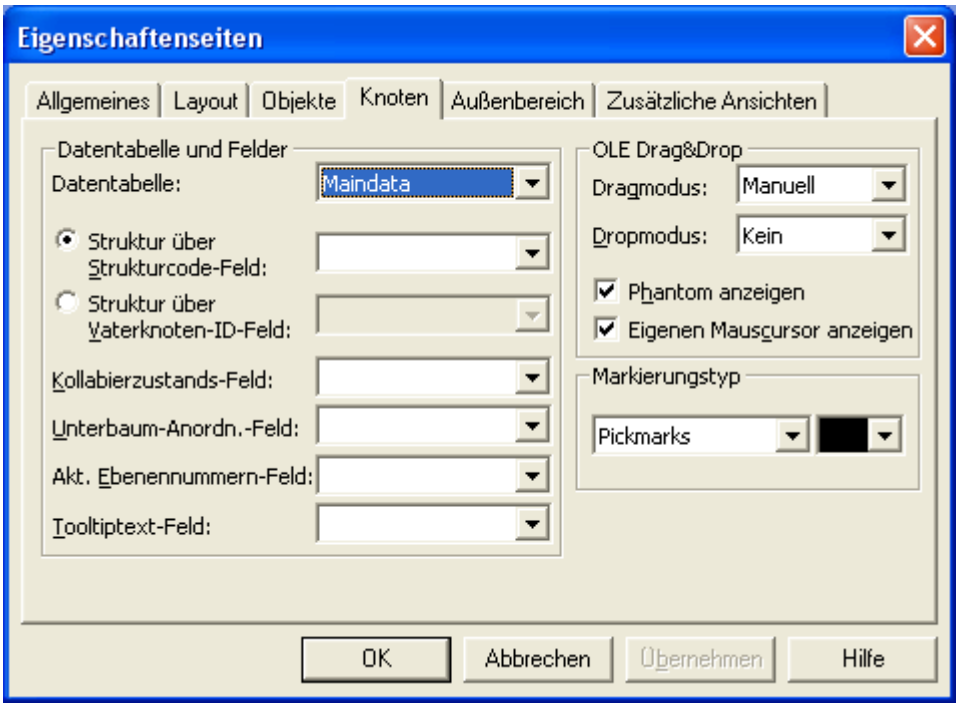
Horizontaler Knotenabstand (h1)

Hier können Sie den horizontalen Abstand zwischen zwei horizontal angeordneten Knoten in Millimetern festlegen.

Horizontale Knoteneinrückung (h2)

Hier können Sie die horizontale Einrückung vertikal angeordneter Knoten in Millimetern festlegen.

4.5 Eigenschaftenseite "Knoten"



Datentabelle

Wählen Sie hier die Datentabelle aus, die für die Darstellung der Knoten herangezogen werden soll.

Diese Option kann auch über die Eigenschaft **VcTree.NodesDataTable-Name** festgelegt werden.

Tooltiptext-Feld

Das Datenfeld, das Sie hier auswählen, wird als Tooltip angezeigt, wenn Sie eine VMF-Datei mit dem WebViewer ansehen und dort auf einen Knoten rechtsklicken. Es sind keine weiteren Einstellungen notwendig.

Das VMF- (Viewer Metafile) Format ist ein Vektorformat, in denen ein Diagramm auflösungsunabhängig gespeichert werden kann. Es kann mittels des als Java Applet ausgeführten GRANEDA WebViewers plattformunabhängig mit jedem Java-fähigen Internet-Browser angesehen werden.

Um in Ihrer Applikation in dem VARCHART ActiveX Tooltips anzuzeigen, müssen Sie auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **OnToolTipText-Ereignisse** aktivieren bzw. die Eigenschaft **ShowToolTip** auf **True** setzen und im **OnToolTipText**-Ereignis programmieren, welche Datenfelder angezeigt werden sollen.

Diese Option kann auch über die Eigenschaft **VcTree.NodeToolTipText-Field** gesetzt werden.

Dragmodus

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

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

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

VARCHART XTree verpackt die Daten u.a. in das Standard-Zwischenablage-Format CF_TEXT (für Visual-Basic-Benutzer: das vbCFText-Format), das mühelos gelesen werden kann.

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

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

Dropmodus

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

Mögliche Alternativen sind:

- **Kein**: Knoten aus einer anderen VARCHART-XNet-Komponente können nicht in die aktuelle Komponente herein gezogen werden.
- **Manuell**: Sie erhalten beim Dropping das Ereignis **OLEDragDrop**, so dass Sie die übertragenen Daten selbst weiterverarbeiten können, um z. B. einen Knoten zu erzeugen oder eine Datei einzulesen. Wenn Quell-

und Zielkomponente identisch sind, erhalten Sie wie bei abgeschaltetem OLE Drag&Drop eins der Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**.

- **Automatisch**: Der Dropping-Vorgang wird von der Komponente selbst verarbeitet, d. h. es wird, falls möglich, ein Knoten an entsprechender Mausposition erzeugt.

Phantom anzeigen

Mit Hilfe dieses Kontrollkästchens können Sie festlegen, ob während eines OLE-Drag-Vorgangs ein Phantom erscheinen soll oder nicht. Das Abschalten des Phantoms ist für Anwendungen gedacht, die beim Hineindraggen eines Objekts kein neues Objekt erzeugen, sondern beispielsweise den Knoten, auf dem dann ein Objekt fallen gelassen wird, nur neu attributieren.

Diese Option kann auch über die Eigenschaft **VcTree.OLEDragWith-Phantom** festgelegt werden.

Eigenen Mauscursor anzeigen

Mit Hilfe dieses Kontrollkästchens können Sie festlegen, ob während eines OLE-Drag-Vorgangs der Mauscursor in der Zielkomponente gesetzt werden soll. Bei OLE Drag & Drop ist es möglich, den Cursor in der Quellkomponente über das Ereignis **OLEGiveFeedback** zu setzen. Daher würde ein Setzen durch die Zielkomponente zu einem Flimmern der konkurrierenden Cursor führen. Über dieses Kontrollkästchen kann man dieses Verhalten beeinflussen. Auserdem können bei eingeschaltetem Mauscursor und bei der auf **vcOLEDropManual** gesetzten Eigenschaft Objekte außerhalb der Anlagerungsstellen eines Knotens nicht fallen gelassen werden, während dies bei eingeschaltetem Mauscursor möglich ist.

Diese Option kann auch über die Eigenschaft **VcTree.OLEDragWithOwn-MouseCursor** festgelegt werden.

Strukturcode in Feld

Wenn Sie diese Optionsschaltfläche aktiviert haben, können Sie hier das Datenfeld auswählen, das den Strukturcode bestimmen soll.

ID des Vaterknotens in Feld

Wenn Sie diese Optionsschaltfläche aktiviert haben, können Sie hier das Datenfeld auswählen, das für die ID des Vaterknotens verwendet werden soll.

Markierungstyp

Legen Sie hier den Markierungstyp für Knoten fest.

Mögliche Alternativen sind:

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

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

Weiterhin können Sie hier eine Farbe für die Markierung auswählen.

Kollabierzustand in Feld

Aktivieren Sie dieses Kontrollkästchen, um den Kollabierstatus eines Knotens synchron in dem hier gewählten Datenfeld zu halten. Mögliche Inhalte des Datenfelds sind "0" (Knoten expandiert) und "1" (Knoten kollabiert).

Unterbaum-Anordnung in Feld

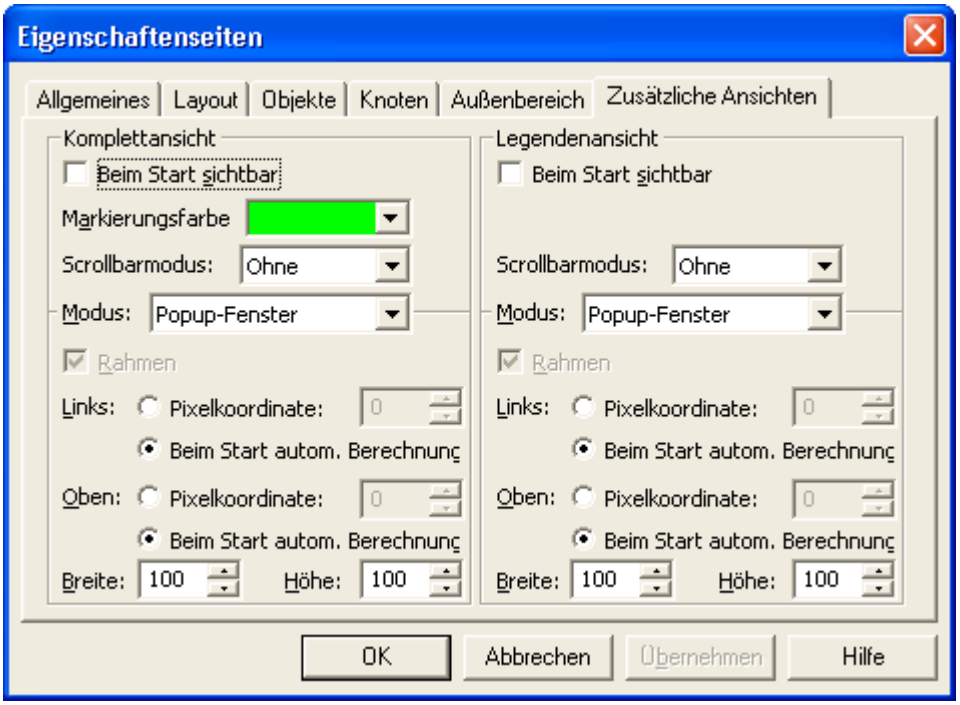
Aktivieren Sie dieses Kontrollkästchen, um die Orientierung eines Teilbaums synchron in einem Datenfeld zu halten. Mögliche Inhalte des Datenfelds sind "0" (Teilbaum horizontal angeordnet) und "1" (Teilbaum vertikal angeordnet). Die horizontale Anordnung ist nur sichtbar, wenn der direkte und alle indirekten Vaterknoten horizontal angeordnet sind.

Aktuelle Ebenennr. in Feld

Aktivieren Sie dieses Kontrollkästchen, um festzulegen, in welchem Datenfeld die Ebenennummer der Knoten abgelegt wird. Die Ebenennummern zählen von 1 an aufwärts. Diese Eigenschaft kann auch über die VcTree-Eigenschaft **LevelField** festgelegt werden.

Hinweis: Zur Laufzeit können Sie nicht die Ebene des Knotens verändern, indem Sie den Wert des Ebenennummer-Datenfeldes ändern.

4.6 Eigenschaftenseite "Zusätzliche Ansichten"



Auf dieser Eigenschaftenseite können Sie die Eigenschaften der Komplettansicht (World View) sowie der Legendenansicht (Legend View) festlegen. Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm angezeigt wird. Ein Rahmen darin zeigt an, welchen Ausschnitt des Diagramms das Hauptfenster gerade anzeigt.

Mithilfe der Legendenansicht lässt sich, ebenfalls in einem zusätzlichen Fenster, eine Legende auf dem Bildschirm darstellen.

Um die Ansichten anzeigen zu lassen, wählen Sie zur Laufzeit im Standard-Kontextmenü **Komplettansicht anzeigen** bzw. für die Legende **Legendenansicht anzeigen**. Über diese Menüpunkte können die Ansichten auch wieder ausgeschaltet werden (alternativ über die **Schließen**-Schaltfläche in der Titelleiste des jeweiligen Fensters).

im Folgenden sind die möglichen Einstellungen für beide Ansichten gleichzeitig beschrieben. Sollte ein Kriterium nicht für beide Ansichten gelten, so wird gesondert darauf hingewiesen.

Beim Start sichtbar

Aktivieren Sie dieses Kontrollkästchen, damit die Ansicht beim Start des Programms sichtbar ist.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Visible** bzw. **VcLegendView.Visible** der Programmierschnittstelle gesetzt werden.

Markierungsfarbe (nur für Komplettansicht)

Wählen Sie hier die Farbe der Linie des Rechtecks aus, das in der Komplettansicht den aktuell gewählten Ausschnitt anzeigt.

Diese Eigenschaft kann auch über die Aufrufe **VcWorld-View.MarkingColor** bzw. **VcLegendView.MarkingColor** der Programmierschnittstelle gesetzt werden.

Scrollbarmodus

Wählen Sie hier, ob und welche Bildlaufleisten in der Ansicht dargestellt werden sollen. Durch die Verwendung von Bildlaufleisten werden Leerbereiche vermieden und das Diagramm bzw. die Legende ist besser zu erkennen, weil es bzw. sie größer dargestellt wird. Folgende Möglichkeiten stehen zur Verfügung:

- **Ohne:** In der Ansicht wird immer alles vollständig dargestellt. Dadurch können Leerbereiche entstehen, wenn die Ansicht in ihren Proportionen nicht denen des Charts(oder der Legende) entspricht.
- **Horizontal:** Es wird, wenn notwendig, eine horizontale Bildlaufleiste dargestellt.
- **Vertikal:** Es wird, wenn notwendig, eine vertikale Bildlaufleiste dargestellt.
- **Automatisch:** Es wird, wenn notwendig, eine horizontale oder eie vertikale Bildlaufleiste dargestellt.

Diese Eigenschaft kann auch über die Aufrufe **VcWorld-View.ScrollBarMode** bzw. **VcLegendView.ScrollBarMode** der Programmierschnittstelle gesetzt werden.

Modus

Wählen Sie hier den Modus für die Ansicht aus. Es gibt folgende Möglichkeiten:

- **fest an linker Seite:** Die Ansicht wird links im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.

- **fest an rechter Seite:** Die Ansicht wird rechts im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
- **fest an oberer Seite:** Die Ansicht wird oben im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
- **fest an unterer Seite:** Die Ansicht wird unten im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
- **nicht fest positioniert:** Die Ansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des VARCHART ActiveX und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft **VcWorldView.ParentHWnd** geändert werden.
- **Popup-Fenster:** Die Ansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Es kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die **Schließen**-Schaltfläche in der Titelleiste ausgeschaltet werden.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Mode** bzw. **VcLegendView.Mode** der Programmierschnittstelle gesetzt werden.

Rahmen

*Nicht aktiviert, wenn der Modus **Popup-Fenster** gewählt wurde.* Aktivieren Sie dieses Kontrollkästchen, wenn die Ansicht einen Rahmen erhalten soll. Die Rahmenfarbe kann aus der Drop-Down-Liste gewählt werden.

Diese OPpionen können auch über die Aufrufe **VcWorldView.Border** und **VcWorldView.Border.Color** bzw. **VcLegendView.Border** und **VcLegend-View.Border.Color**der Programmierschnittstelle gesetzt werden.

Links

*Nur aktiviert, wenn der Modus **nicht fest positioniert** oder **Popup-Fenster** gewählt wurde.* Legen Sie hier die linke Position der Ansicht fest. Dabei gibt es zwei Möglichkeiten:

1. Geben Sie unter **Pixelkoordinate** einen Wert an. Dabei handelt es sich um Gerätekoordinaten.

2. Wählen Sie die Option **Beim Start automat. Berechnung**, damit die Position der Ansicht beim Programmstart automatisch berechnet wird.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Left** bzw. **VcLegendView.Left** der Programmierschnittstelle gesetzt werden.

Oben

*Nur aktiviert, wenn der Modus **nicht fest positioniert** oder **Popup-Fenster** gewählt wurde.* Legen Sie hier die obere Position der Ansicht fest. Dabei gibt es zwei Möglichkeiten:

1. Geben Sie unter **Pixelkoordinate** einen Wert an. Dabei handelt es sich um Gerätekoordinaten.
2. Wählen Sie die Option **Beim Start automat. Berechnung**, damit die Position der Ansicht beim Programmstart automatisch berechnet wird.

Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Top** bzw. **VcLegendView.Top** der Programmierschnittstelle gesetzt werden.

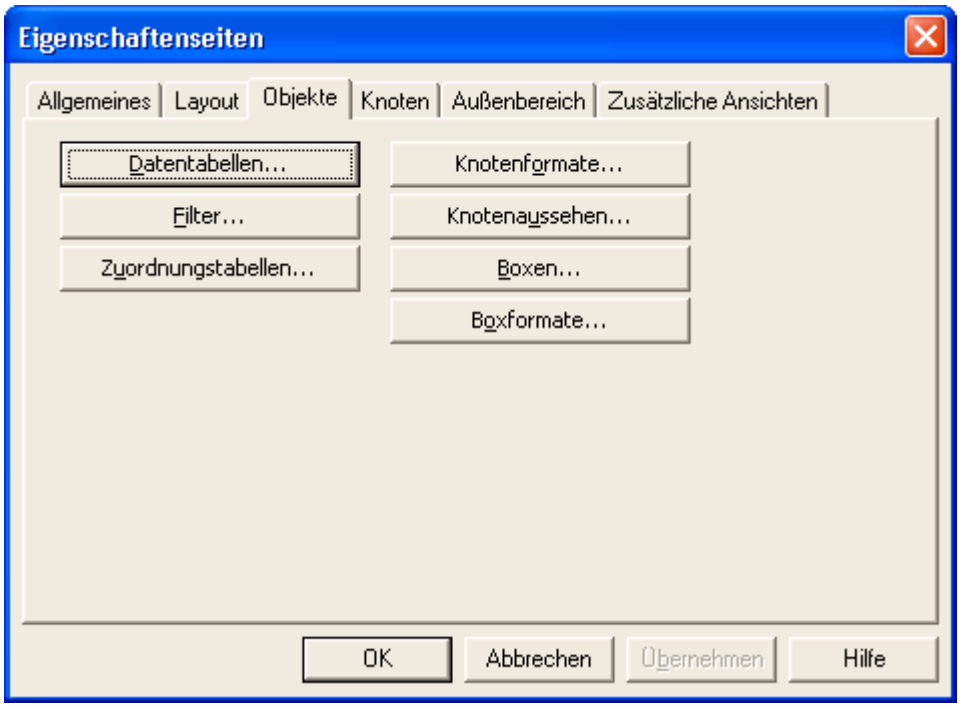
Breite

*Nicht aktiviert, wenn der Modus **fest an oberer/unterer Seite** gewählt wurde.* Legen Sie hier die horizontale Ausdehnung der Ansicht fest. Der Wert wird in Pixelkoordinaten (Gerätekoordinaten) angegeben. Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Width** bzw. **VcLegendView.Width** der Programmierschnittstelle gesetzt werden.

Höhe

*Nicht aktiviert, wenn der Modus **fest an linker/rechter Seite** gewählt wurde.* Legen Sie hier die vertikale Ausdehnung der Ansicht fest. Der Wert wird in Pixelkoordinaten (Gerätekoordinaten) angegeben. Diese Eigenschaft kann auch über die Aufrufe **VcWorldView.Height** bzw. **VcLegendView.Height** der Programmierschnittstelle gesetzt werden.

4.7 Eigenschaftenseite "Objekte"



Datentabellen

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

Filter

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

Zuordnungstabellen

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

Knotenformate

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

Knotenaussehen

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

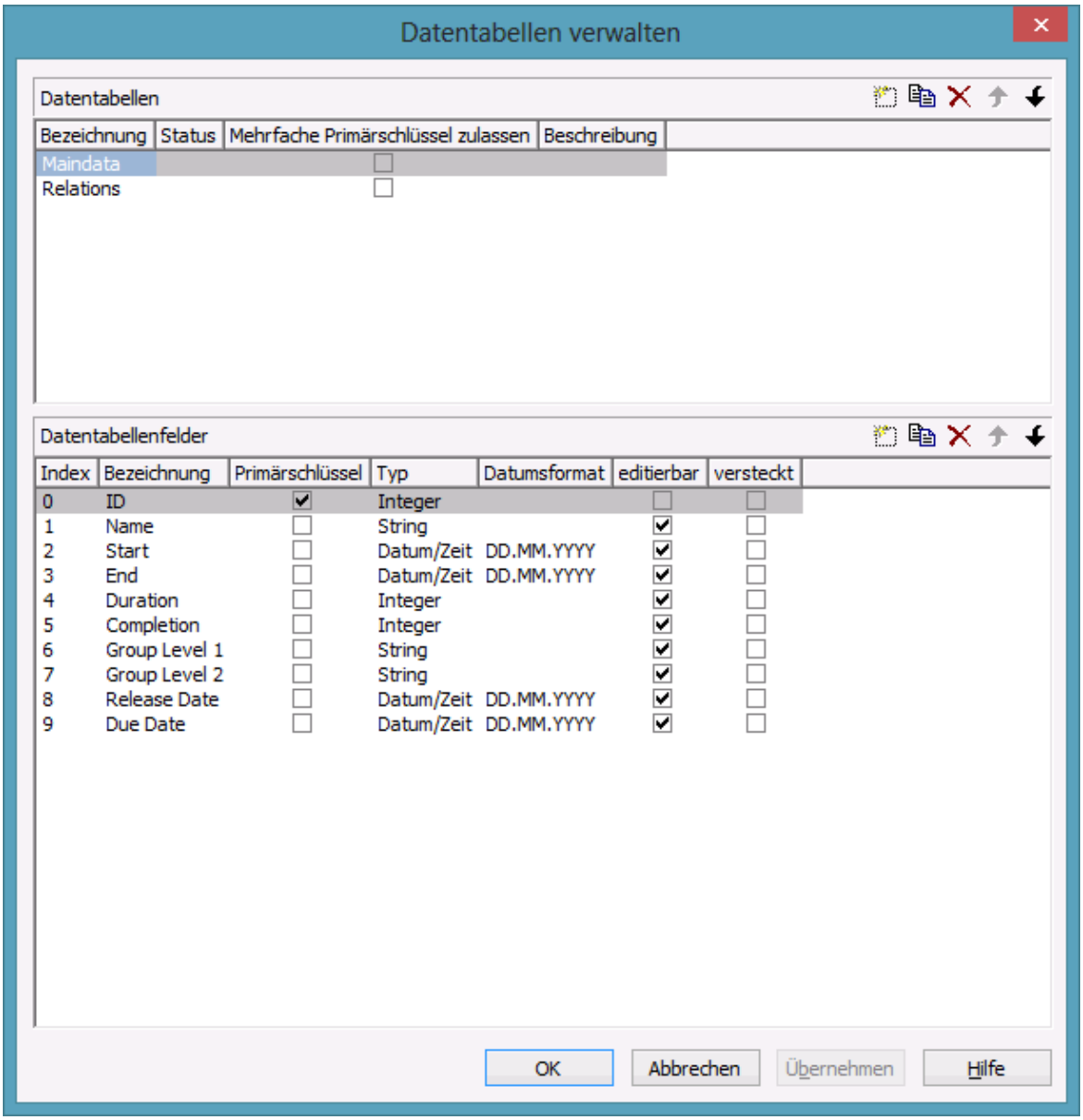
Boxen

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

Boxformate

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

4.8 Dialogfeld "Datentabellen verwalten"



Sie gelangen in diesen Dialog über die Eigenschaftenseite **Objekte**. Sie können hier Datentabellen sowie die zugehörigen Datenfelder anlegen und bearbeiten.

Datentabellen

- **Bezeichnung:** In dieser Spalte stehen die Namen aller vorhandenen Datentabellen. Die Namen sind editierbar.
- **Status:** In der Spalte **Status** wird jede Dateantabelle gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt ( ) und/oder geändert ( ) worden ist.

- **Mehrfache Primärschlüssel zulassen:** Bestimmen Sie hier, ob der Primärschlüssel der Tabelle aus **einem** oder **mehreren (maximal 3)** Feldern bestehen soll. Sobald Sie **Mehrfache Primärschlüssel zulassen** ausgewählt haben, sind im Bereich **Datentabellenfelder** bis zu 3 Felder für den Primärschlüssel wählbar. Die Einstellung **Mehrfache Primärschlüssel zulassen** kann erst dann wieder deaktiviert werden, wenn im Bereich **Datentabellenfelder** nur noch ein Feld für den Primärschlüssel ausgewählt ist.
- **Beschreibung:** Geben Sie hier eine Beschreibung für die Datentabelle ein.

Datentabelle hinzufügen / kopieren / löschen / nach oben / unten

 Mit Hilfe dieser Schaltflächen können Sie Datentabellen hinzufügen, kopieren, löschen und in der Liste nach oben oder unten verschieben.

Datentabellenfelder

Hier können Sie für die im Bereich **Datentabellen** ausgewählte Datentabelle Datentabellenfelder anlegen und bearbeiten.

- **Index:** Der Index der Datentabellenfelder ist nicht veränderbar, da er intern als Referenz dient. In der API werden Datenfelder über den Index angesprochen.
- **Bezeichnung:** Diese Spalte zeigt die Namen der Felder der Datentabelle. Nach Anklicken können Sie diese ändern.
- **Primärschlüssel:** Hier können Sie festlegen, welches Feld in der Spalte der Primärschlüssel der Datensätze dieser Tabelle sein soll.
- **Typ:** Hier können Sie den Datentyp für jedes Datentabellenfeld festlegen. Zur Auswahl stehen:

  Alphanumerisch

  Integer

  Datum/Zeit

  Double

- **Datumsformat:** Für die Datentabellenfelder vom Typ **Datum/Zeit** können Sie hier jeweils das Datumsformat an das Datumsformat Ihrer Knotendaten anpassen. Einige gebräuchliche Datumsformate stehen zur

Auswahl. Sie können außerdem ein eigenes Datumsformat definieren, z. B. "DD.MM.YY hh:mm".

Das Datumsformat wird aus den Kombinationen **YY** (zweistellige Jahreszahl), **YYYY** (vierstellige Jahreszahl), **MM** (zweistellige Monatszahl), **MMM** (dreistelliges Monatsnamenkürzel), **DD** (zweistellige Tageszahl), **hh** (zweistellige Stundenzahl), **mm** (zweistellige Minutenzahl) und **ss** (zweistellige Sekundenzahl) zusammengesetzt.

Beachten Sie bitte, dass das Datumsformat, das Sie hier festlegen, mit dem Datumsformat Ihrer Knotendaten übereinstimmen muss.

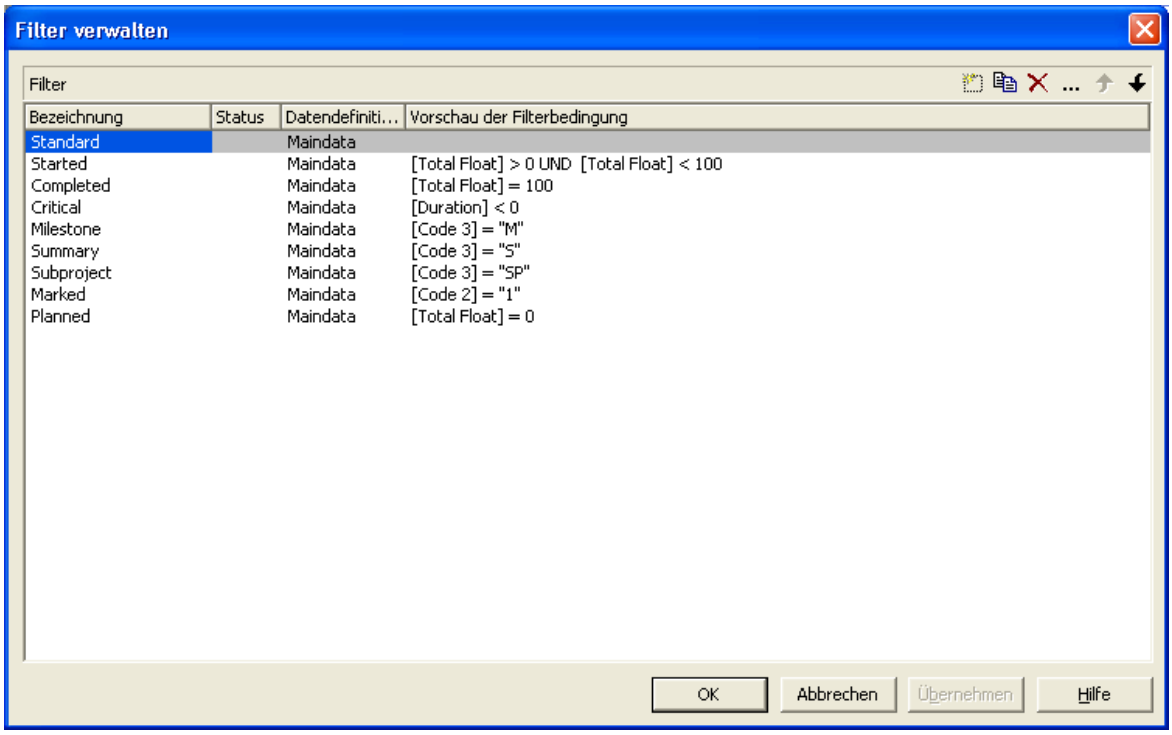
Dieses Datumsformat ist nur für die Dateneingabe, aber nicht für die Darstellung von Daten in Ihrer Grafik relevant.

- **Editierbar:** Aktivieren Sie dieses Kontrollkästchen für alle Datentabellenfelder, die der Anwender im Dialogfeld **Vorgänge bearbeiten** bearbeiten können soll.
- **Versteckt:** Aktivieren Sie dieses Kontrollkästchen für alle Datentabellenfelder, die dem Anwender im Dialogfeld **Vorgänge bearbeiten** verborgen bleiben sollen.
- **Beziehung:** Hier können Sie eine Verknüpfung zu einer anderen Tabelle festlegen, so dass die Datensätze dieser Tabelle über das als Primärschlüssel definierte Feld einer anderen verknüpft sind. Aus diesem Grund werden Ihnen zur Auswahl alle Tabellen angeboten, für die Sie einen Primärschlüssel definiert haben.

Datentabellenfeld hinzufügen / kopieren / löschen / nach oben / unten

 Mit Hilfe dieser Schaltflächen können Sie Datentabellenfelder hinzufügen, kopieren, löschen und in der Liste nach oben oder unten verschieben.

4.9 Dialogfeld "Filter verwalten"



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

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Filter. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jeder Filter gekennzeichnet, der seit dem Aufruf des Dialogs hinzugefügt ( ) oder geändert ( ) worden ist.

Datendefinitionstabelle

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

Vorschau der Filterbedingung

In dieser Spalte wird die Bedingung jedes Filters angezeigt. Sie kann hier nicht editiert werden. Um die Filterbedingung zu bearbeiten, klicken Sie auf die **Filter bearbeiten**-Schaltfläche.

Filter hinzufügen

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

Neue Filter werden kontextabhängig angelegt, d. h. es wird immer die jeweils passende Datendefinitionstabelle verwendet.

Filter kopieren

 Der markierte Filter wird kopiert.

Filter löschen

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

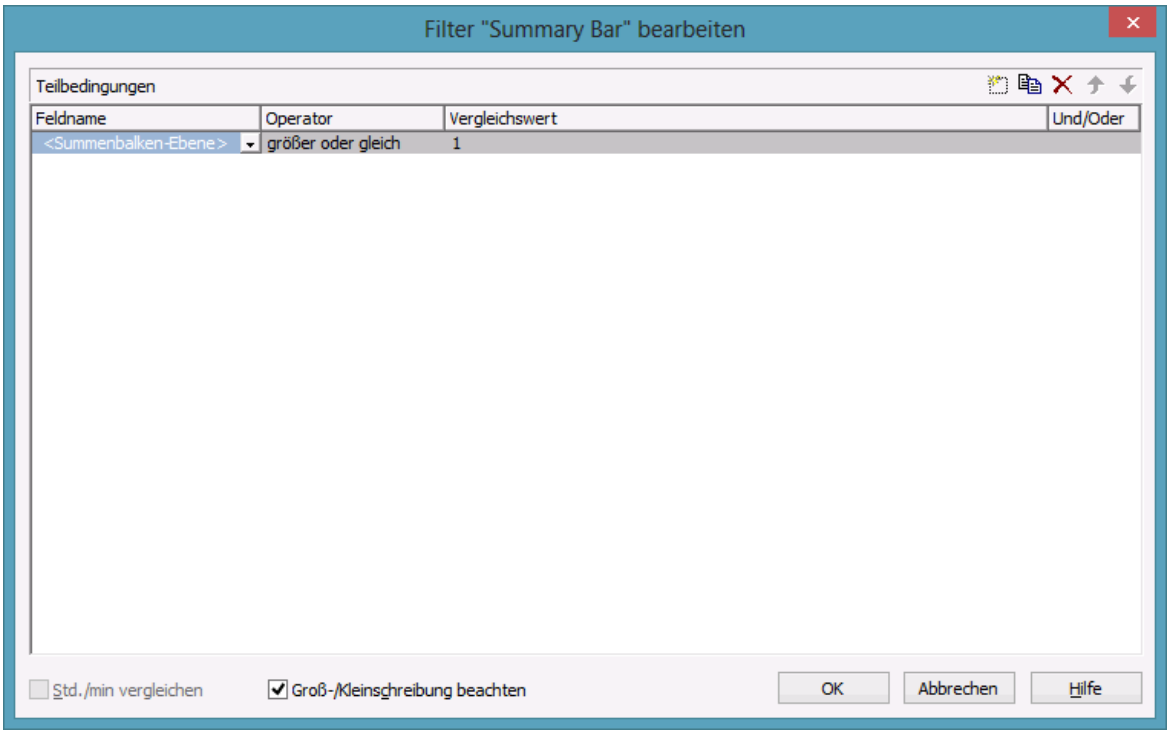
Filter bearbeiten

 Um die Bedingungen eines Filters anzusehen oder zu ändern, klicken Sie auf die Schaltfläche **Filter bearbeiten**. Es erscheint das Dialogfeld **Filter bearbeiten**. Nun können Sie die für den Filter formulierten Bedingungen bearbeiten.

Filter eine Zeile nach oben/unten

 Mit Hilfe dieser Schaltflächen können Sie den markierten Filter eine Zeile nach oben/unten schieben.

4.10 Dialogfeld "Filter bearbeiten"



Sie erreichen dieses Dialogfeld in dem Sie entweder

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

den **Filter verwalten**-Dialog aktivieren und anschließend die Schaltfläche **Filter bearbeiten** anklicken. In der Kopfzeile dieses Dialogfeldes steht der Name des aktuellen Filters.

Teilbedingung hinzufügen

Vor der markierten Zeile wird eine neue Zeile für eine Teilbedingung eingefügt.

Teilbedingung kopieren

Die markierte Teilbedingung wird kopiert und in der Liste eingefügt.

Teilbedingung löschen

Die markierte Teilbedingung wird gelöscht.

Teilbedingung früher/später abarbeiten

 Wenn ein Filter mehrere Teilbedingungen enthält, werden die einzelnen Teilbedingungen in der Reihenfolge, in der sie in der **Teilbedingungen**-Tabelle stehen, abgearbeitet.

Klicken Sie die Schaltfläche **Teilbedingung früher/später abarbeiten** an, um die markierte Teilbedingung in der Tabelle eine Position höher/tiefer zu setzen, damit sie früher/später abgearbeitet wird.

Feldname

Hier können Sie das Datenfeld auswählen, das mit dem Vergleichswert verglichen werden soll. Die Liste enthält alle verfügbaren Datenfelder.

Operator

Wählen Sie hier den Vergleichsoperator für den Vergleich des unter **Feldname** gewählten Datenfeldes mit dem unter **Vergleichswert** angegebenen Wertes bzw. Datenfeldes aus.

Vergleichswert

Hier wird der aktuelle Vergleichswert angezeigt. Im Feld **Vergleichswert** werden in eckigen Klammern alle Felder des passenden Datentyps angeboten, die jeweils als Vergleichswert eingesetzt werden können. Wurde unter **Feldname** beispielsweise das Feld "Frühester Anfang" eingetragen, werden als Vergleichswerte nur Terminfelder (z. B. "Frühestes Ende") und die Optionen <heute> und <Eingabe> angeboten.

Mit Hilfe des Vergleichswertes <Eingabe> können Sie einen variablen Filter definieren. In variablen Filtern sind nur der Feldname und der Operator, aber nicht der Vergleichswert definiert. Diesen können Sie bei Bedarf angeben. Sie können einen variablen Filter verwenden, wenn Sie ein Projekt öffnen und die darzustellenden Vorgänge begrenzen möchten.

Termine werden in dem Format, das auf der Eigenschaftenseite **Allgemeines** unter **Datumsausgabeformat** festgelegt wurde, eingegeben. Wenn Sie unter **Feldname** ein Terminfeld gewählt haben, erscheinen im Feld **Vergleichswert** zwei Pfeil-Schaltflächen, sobald Sie dieses Feld anklicken. Über die erste Pfeil-Schaltfläche öffnen Sie eine Kombobox, die Ihnen alle verfügbaren Termin-Datenfelder anzeigt. Über die zweite Pfeil-Schaltfläche öffnen Sie den Datumsdialog, aus dem Sie das gewünschte Datum per Mausklick auswählen können. Sie können das Datum aber auch direkt editieren.

Zahlenwerte oder Texte müssen direkt eingegeben werden.

Bei den Operatoren "gleich" und "ungleich" können Sie für Textfelder auch Wildcards verwenden:

*: kein oder mehrere beliebige Zeichen

?: genau ein beliebiges Zeichen

Wenn Sie die Zeichen * oder ? nicht als Wildcards verwenden, sondern auf diese Zeichen abfragen wollen, müssen Sie dies durch einen vorangestellten Backslash kenntlich machen:

*: *

\?: ?

Wenn dem Backslash kein * oder ? folgt, wird nach dem Vorhandensein von \ gefragt.

Beispiele:

Vorgang 1 : Name = "Baugenehmigung"

Vorgang 2 : Name = "*Baugenehmigung"

Mögliche Filter für Vorgang 1:

[Name] = B*

[Name] = B?ugenehmigung

Mögliche Filter für Vorgang 2:

[Name] = *B*

[Name] = **

[Name] = ?B*

Und/Oder

Hier wird die logische Verknüpfung der aktuellen mit der jeweils nächsten Teilbedingung in der Tabelle angezeigt.

Wenn Sie den UND-Operator wählen, werden nur die Objekte ausgewählt, die beide Teilbedingungen erfüllen. Wählen Sie den ODER-Operator, werden die Objekte ausgewählt, die mindestens eine der verknüpften Teilbedingungen erfüllen.

Haben Sie mehrere Teilbedingungen formuliert und diese zum Teil mit UND, zum Teil mit ODER verknüpft, werden erst die UND-Verknüpfungen abgearbeitet. (UND bindet stärker als ODER.)

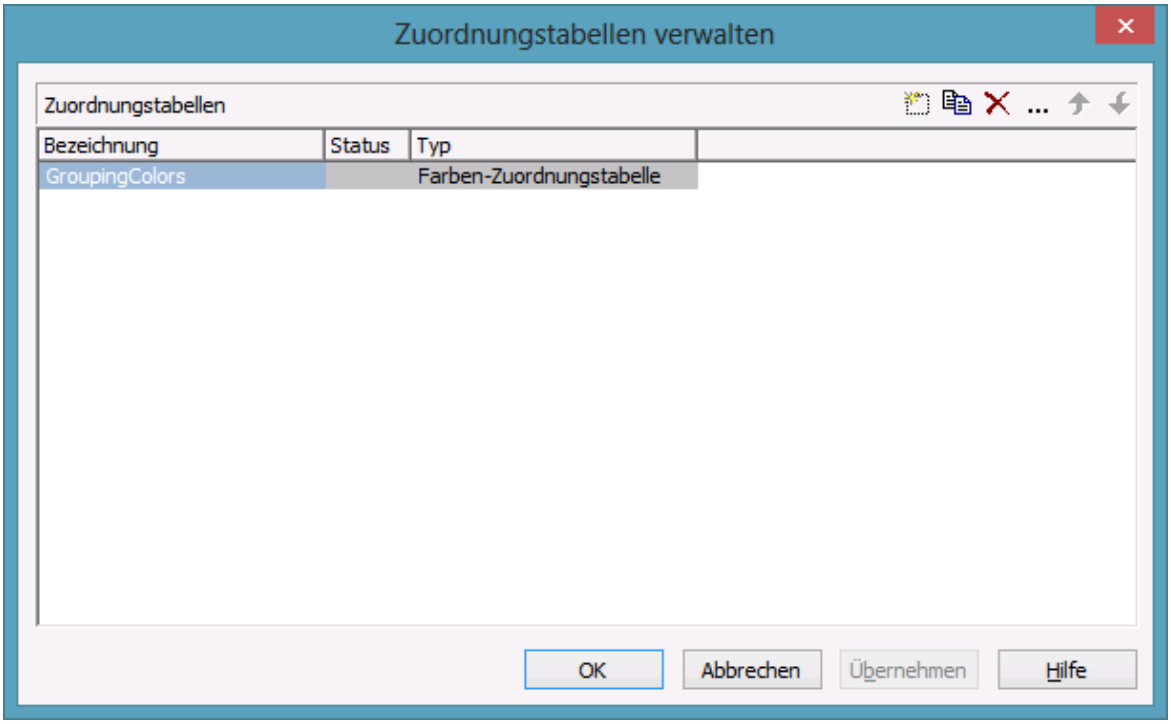
Std./min vergleichen

Aktivieren Sie dieses Kontrollkästchen, wenn beim Datumsvergleich auch Stunde und Minute berücksichtigt werden sollen.

Groß-/Kleinschreibung beachten

Aktivieren Sie dieses Kontrollkästchen, wenn beim Vergleich die Groß-/Kleinschreibung beachtet werden soll.

4.11 Dialogfeld "Zuordnungstabellen verwalten"



Sie gelangen über die Eigenschaftenseite **Objekte** in dieses Dialogfeld oder indem Sie im Dialogfeld **Zuordnung einstellen** auf die Schaltfläche **Zuordnungstabellen** klicken.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Zuordnungstabellen. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jede Zuordnungstabelle gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt ( ) und/oder geändert ( ) worden ist.

Typ

Wählen Sie hier den Typ der Zuordnungstabelle aus:

- Farben-Zuordnungstabelle
- Schraffuren-Zuordnungstabelle (für spätere Anwendungen)

- Grafikdateien-Zuordnungstabelle

Zuordnungstabelle hinzufügen

 Eine neue Zuordnungstabelle mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Zuordnungstabelle kopieren

 Die markierte Zuordnungstabelle wird kopiert.

Zuordnungstabelle löschen

 Die Zuordnungstabelle, die Sie in der Liste markiert haben, wird gelöscht. Es können nur die Zuordnungstabellen gelöscht werden, die zur Zeit nicht benutzt werden.

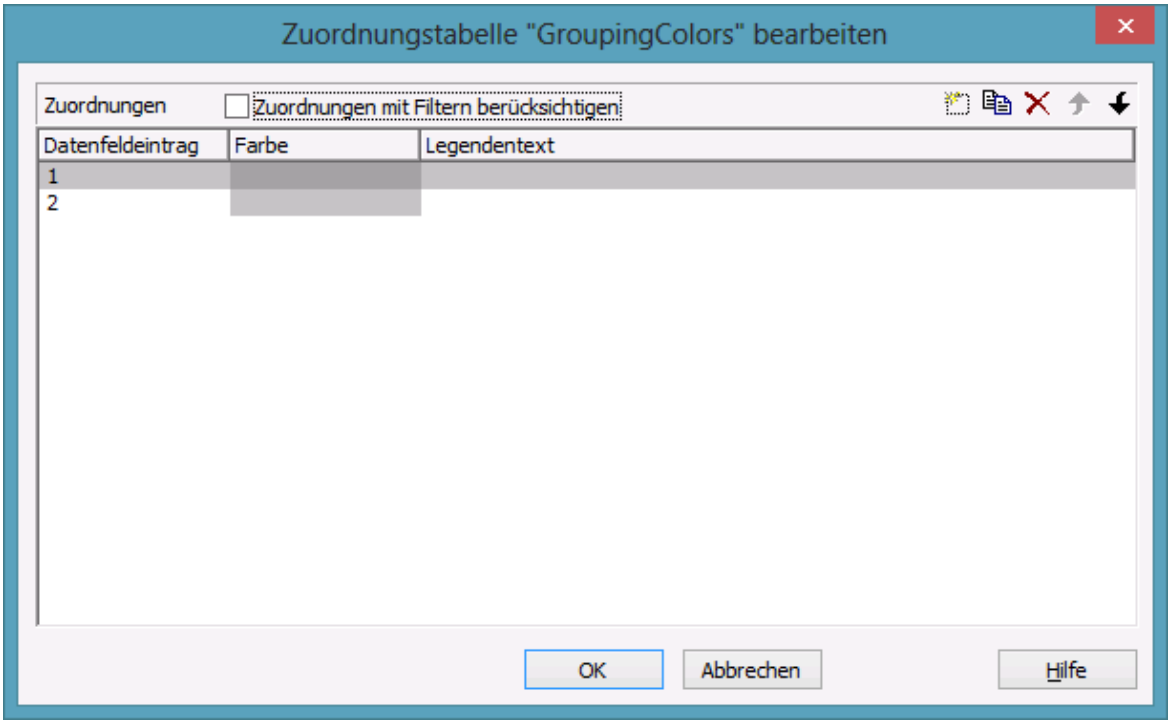
Zuordnungstabelle bearbeiten

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

Zuordnungstabelle eine Zeile nach oben / unten

 Mit Hilfe dieser Schaltfläche können Sie die markierte Zuordnungstabelle in der Tabelle eine Zeile nach oben/unten schieben.

4.12 Dialogfeld "Zuordnungstabelle bearbeiten"



Sie gelangen in dieses Dialogfeld, indem Sie im Dialogfeld **Zuordnungstabellen verwalten** auf die Schaltfläche **Zuordnungstabelle bearbeiten** (…) klicken.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie mehr Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

Zuordnungen mit Filtern berücksichtigen

Wenn diese Option ausgewählt ist, werden nicht nur die in der Liste der Datenfeldeinträge angegebenen festen Werte als Schlüsselwerte berücksichtigt, sondern auch Filter, die aus der Dropdown-Liste ausgewählt werden können. Dadurch hängen die Ausführungswerte nicht mehr nur von einem konkreten Wert ab, sondern von einem Wertebereich.

Datenfeldeintrag

In dieser Spalte werden die Einträge des gewählten Datenfeldes aufgeführt, für die Sie eine Farbe bzw. eine Grafikdatei und den Legendentext vereinbaren können.

Grafikdateiname

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

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

Farbe/Grafikdateiname

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

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

Legendentext

(nur für Farb-Zuordnungstabellen) Hier können Sie für jeden Datenfeldeintrag einen Legendentext festlegen.

Zuordnung hinzufügen

 Eine neue Zuordnung mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Zuordnung kopieren

 Die markierte Zuordnung wird kopiert.

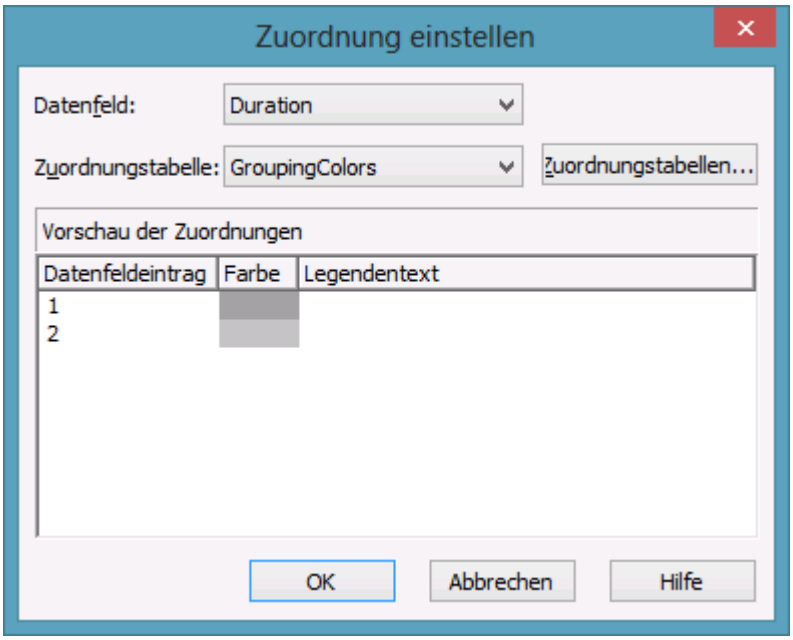
Zuordnung löschen

 Die Zuordnung, die Sie in der Liste markiert haben, wird gelöscht. Es können nur die Zuordnungen gelöscht werden, die zur Zeit nicht benutzt werden.

Zuordnung eine Zeile nach oben / unten

 Die markierte Zuordnung wird in der Tabelle eine Zeile nach oben/unten geschoben.

4.13 Dialogfeld "Zuordnung einstellen"



Verbinden Sie in diesem Dialogfeld das Datenfeld eines Knotens mit einer Zuordnungstabelle. Sie erreichen es über den Dialog **Layer bearbeiten**. Klicken Sie dort beim gewünschten Attribut auf die Schaltfläche  .

Datenfeld

Wählen Sie hier das Datenfeld aus, von dessen Einträgen die gewünschten Attribute des zu bearbeitenden Objekts abhängen soll.

Datenfeld

Wählen Sie hier das Datenfeld aus, von dessen Einträgen die Farbe oder das Muster des zu bearbeitenden Objekts abhängen soll.

Zuordnungstabelle

(nur aktiv, wenn ein Datenfeld gewählt wurde) Wählen Sie hier die Zuordnungstabelle aus, die in Abhängigkeit ihresTyps den einzelnen Datenfeldeinträgen die entsprechenden Attribute zuordnet.

Zuordnungstabelle

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

Zuordnungstabellen

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

Zuordnungstabellen

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

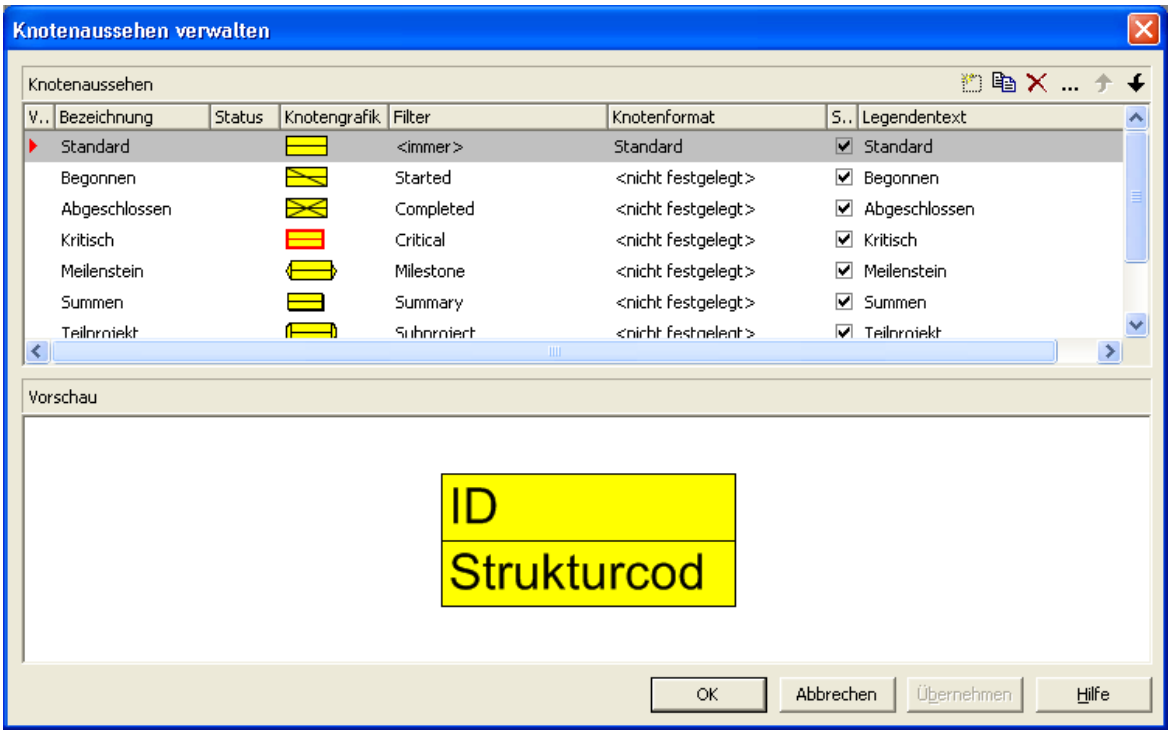
Vorschau der Zuordnungen

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

Vorschau der Zuordnungen

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

4.14 Dialogfeld "Knotenaussehen verwalten"



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

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

Vorschau

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

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

Bezeichnung

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

Status

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

Knotengrafik

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

Filter

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

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

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

Knotenformat

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

Sichtbar in Legende

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

Legendentext

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

Knotenaussehen hinzufügen

 Ein neues Knotenaussehen wird am Ende der Liste hinzugefügt.

Knotenaussehen kopieren

 Das markierte Knotenaussehen wird kopiert.

Knotenaussehen löschen

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

Knotenaussehen bearbeiten

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

Knotenaussehen früher/später abarbeiten

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

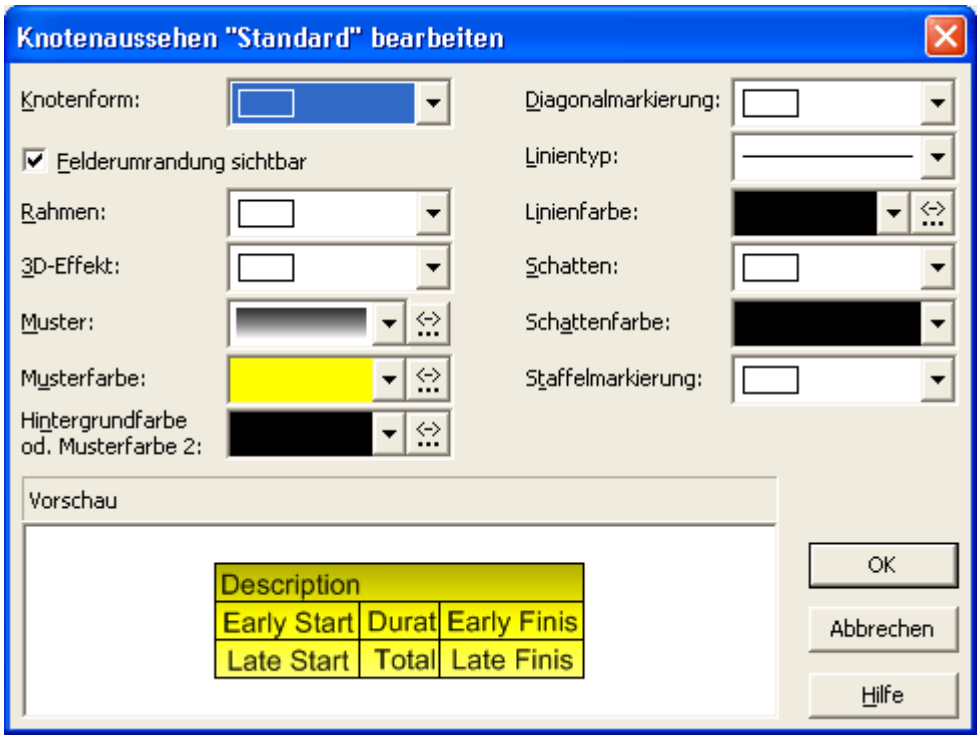
Wenn mehrere Knotenaussehen auf einen Knoten zutreffen, werden die Attribute jedes Knotenaussehens durch die Attribute aller Knotenaussehen, die später abgearbeitet werden, ersetzt. Nur die Attribute, deren Wert "nicht festgelegt" ist, ersetzen nicht die Attribute ihrer Vorgänger.

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

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

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

4.15 Dialogfeld "Knotenaussehen bearbeiten"



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

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

Knotenform

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

Felderumrandung sichtbar

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

Diese Option kann auch über die Eigenschaft **VcNodeAppearance.Frame-AroundFieldsVisible** gesetzt werden.

Rahmen

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

3D-Effekt

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

Muster

Hier wird das standardmäßige Muster des Knotenaussehens angezeigt.

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

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

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

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

Musterfarbe

Hier können Sie die standardmäßige Farbe des Musters festlegen.

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

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

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

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

Hintergrundfarbe oder Musterfarbe 2

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

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

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

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

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

Diagonalmarkierung

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

Linientyp

Wählen Sie hier die Linienart für die Umrandung der Knoten.

Linienfarbe

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

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

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

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

Schatten

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

Schattenfarbe

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

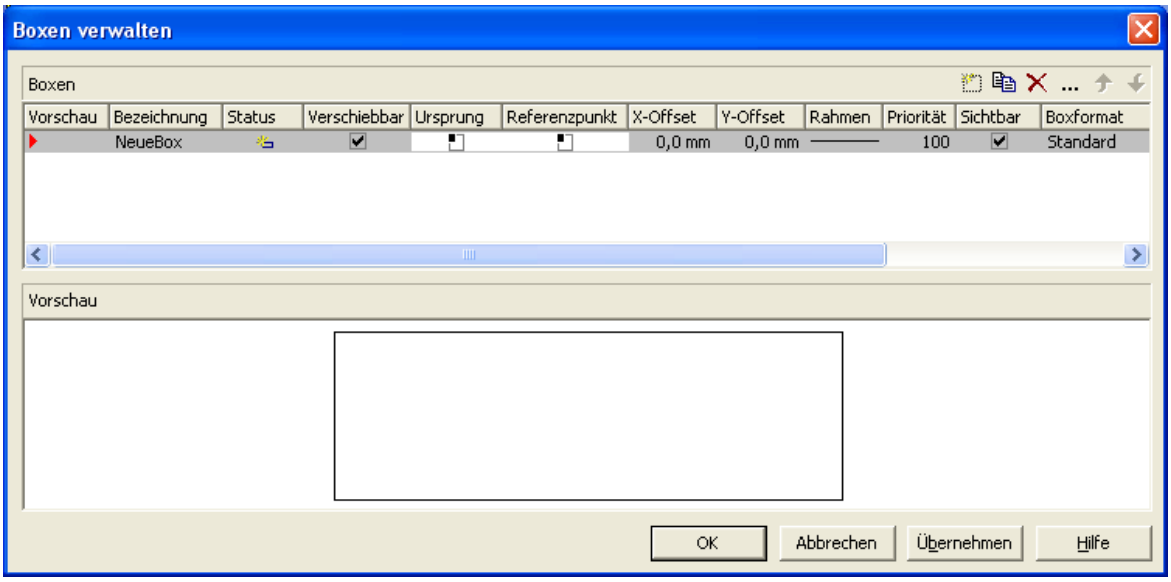
Staffelmarkierung

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

Vorschau

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

4.16 Dialogfeld "Boxen verwalten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte**. Im Diagrammbereich können beliebig viele Boxen dargestellt werden, die Sie in diesem Dialogfeld verwalten können.

Vorschau

Die in dieser Spalte markierte Box wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Boxen. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jede Box gekennzeichnet, die seit dem Aufruf des Dialogs hinzugefügt ( ) und/oder geändert ( ) worden ist.

Aktualisierungsverhalten

Wählen Sie hier das gewünschte Aktualisierungsverhalten aus. Bei der Einstellung <nicht festgelegt>, gilt das Aktualisierungsverhalten, das im Dialog **Aktualisierungsverhalten bearbeiten** für Boxen festgelegt wurde.

Verschiebbar

Durch Bewegen einer Box wird ihr Offset verändert. Legen Sie hier fest, ob die Box zur Laufzeit im Diagrammbereich frei beweglich sein soll. Wenn die Box nicht mehr verschiebbar sein soll, nachdem Sie sie positioniert haben, schalten Sie diese Option ab.

Ursprung

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

Legen Sie hier den Ursprung fest, d. h. den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird. Mögliche Werte des Ursprungs: links oben, mittig oben, rechts oben, links mittig, mittig mittig, rechts mittig, links unten, mittig unten, rechts unten.

Referenzpunkt

Legen Sie hier den Referenzpunkt der Box fest, d. h. den Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird. Mögliche Werte des Referenzpunkts: oben links, oben mittig, oben rechts, mittig links, mittig mittig, mittig rechts, unten links, unten mittig, unten rechts.

X-Offset

Legen Sie hier den Abstand zwischen Ursprung und Referenzpunkt in x-Richtung fest (in mm).

Y-Offset

Legen Sie hier den Abstand zwischen Ursprung und Referenzpunkt in y-Richtung fest (in mm).

Rahmen

Wenn Sie auf dieses Feld klicken, erscheint eine **Bearbeiten**-Schaltfläche, über die Sie in den Dialog **Linie bearbeiten** gelangen. Hier können Sie den Typ, die Dicke und die Farbe der Umrandungslinie der Box festlegen.

Priorität

Legen Sie hier die relative Zeichnungspriorität der Box zu anderen Objekten im Diagramm (Knoten, Liniengitter etc.) fest. Die Priorität von Knoten ist 0. Falls die Boxen eine höhere Priorität als die Knoten haben, überdecken sie die Knoten so, dass darauf interaktiv nicht mehr zugegriffen werden kann.

Sichtbar

Legen Sie fest, ob die Box zur Laufzeit sichtbar sein soll.

Boxformat

Hier wird das Boxformat der Box angezeigt. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Aus der Kombobox können Sie ein vorhandenes Boxformat wählen.

 Über die **Bearbeiten**-Schaltfläche gelangen Sie in den Dialog **Boxformate verwalten**.

Box hinzufügen

 Eine neue Box mit einem Standardnamen wird angelegt. Diesen Namen können Sie editieren, indem Sie darauf doppelklicken und ihn dann verändern.

Box kopieren

 Die markierte Box wird kopiert.

Box löschen

 Die Box, die Sie in der Liste markiert haben, wird gelöscht.

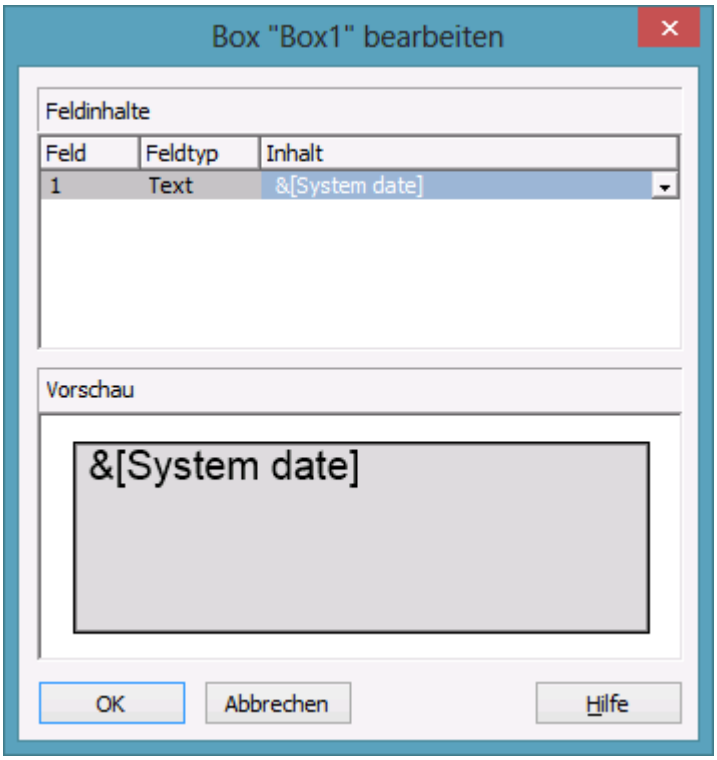
Box bearbeiten

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

Box eine Zeile nach oben / unten

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

4.17 Dialogfeld "Box bearbeiten"



Sie erreichen dieses Dialogfeld über die Eigenschaftenseite **Objekte** und das Dialogfeld **Boxen verwalten**, indem Sie dort auf die **Box bearbeiten**-Schaltfläche klicken. Dieser Dialog erscheint auch zur Laufzeit, wenn sie auf eine Box doppelklicken.

Feld

In dieser Spalte werden die Nummern aller Felder der Box aufgeführt. (Die Anzahl der Felder hängt vom gewählten Boxformat ab.)

Feldtyp

Hier wird der Feldtyp jedes Feldes angezeigt (Text oder Grafik).

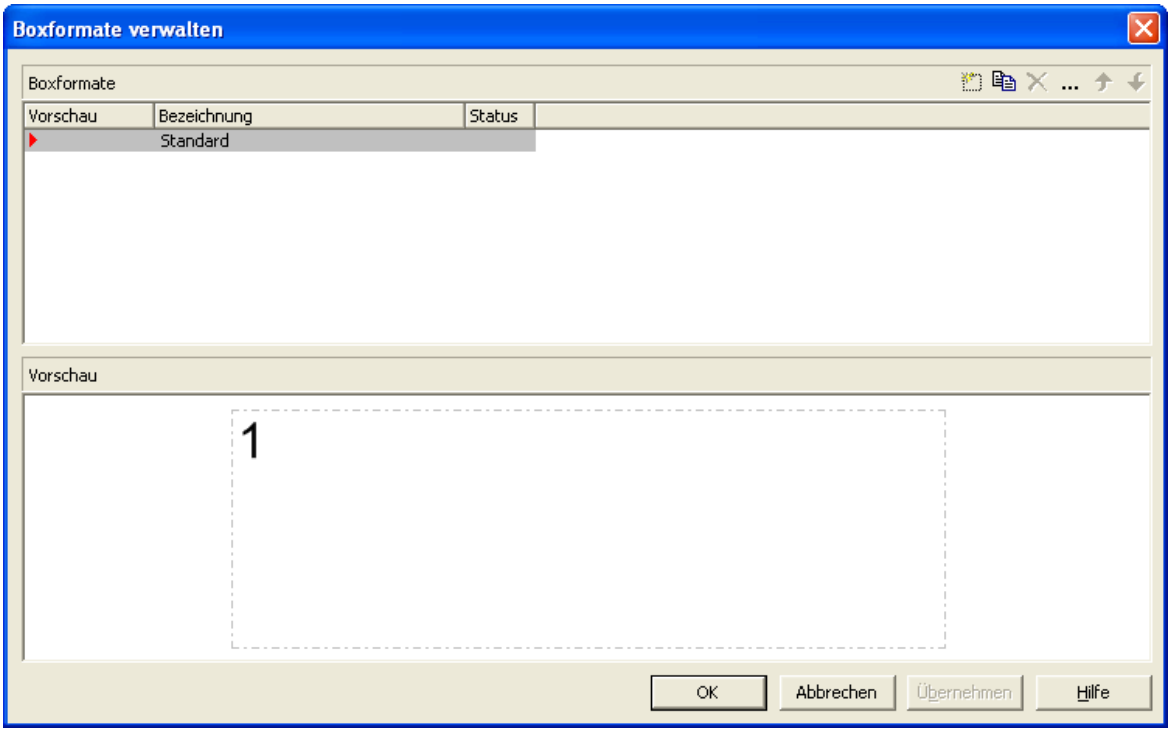
Inhalt

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

Bei mehrzeiligen Textfeldern müssen für einen erzwungenen Umbruch die einzelnen Zeilen des Textfelds mit "\n" im String getrennt sein (Beispiel: "Zeile1\nZeile2"). Ohne erzwungenen Umbruch wird automatisch an Leerzeichen umgebrochen.

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

4.18 Dialogfeld "Boxformate/Knotenformate verwalten"



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

Vorschau

Das in der Vorschauspalte markierte Format wird im Vorschaufenster angezeigt.

Bezeichnung

In dieser Spalte stehen die Namen aller vorhandenen Formate. Die Namen sind editierbar.

Status

In der Spalte **Status** wird jedes Format gekennzeichnet, das seit dem Aufruf des Dialogs hinzugefügt ( ) und/oder geändert ( ) worden ist.

Box/Knotenformat hinzufügen

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

Box/Knotenformat kopieren

 Das markierte Format wird kopiert.

Box/Knotenformat löschen

 Das Format, das Sie in der Liste markiert haben, wird gelöscht. Es können nur die Formate gelöscht werden, die zur Zeit nicht benutzt werden.

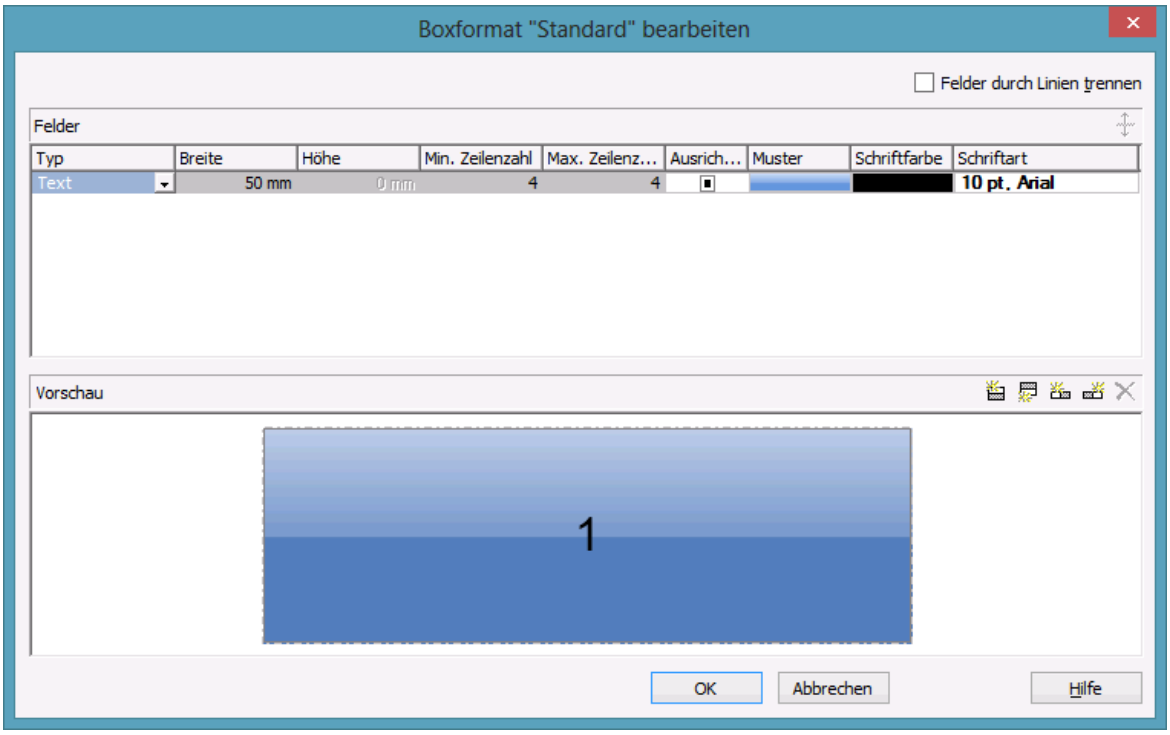
Box/Knotenformat bearbeiten

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

Box/Knotenformat eine Zeile nach oben/ unten

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

4.19 Dialogfeld "Boxformat bearbeiten"



Sie erreichen dieses Dialogfeld, indem Sie auf der Eigenschaftenseite **Objekte** das Dialogfeld **Boxformate verwalten** aktivieren und dort auf die **Boxformat bearbeiten**-Schaltfläche klicken.

Felder durch Linien trennen

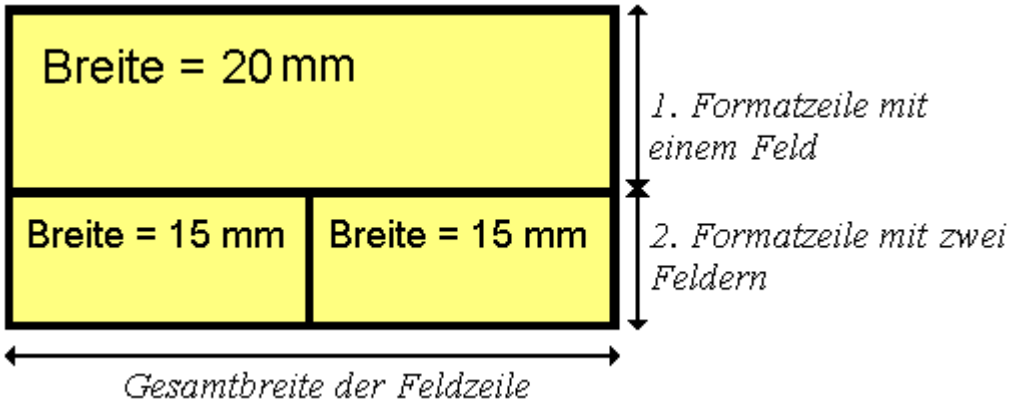
Aktivieren Sie dieses Kontrollkästchen, wenn die Felder der Box durch Linien getrennt werden sollen.

Typ

Wählen Sie hier den Feldtyp (Text oder Grafik).

Breite

Legen Sie hier die Breite in Millimetern für das markierte Feld fest. Die maximale Breite eines Feldes beträgt 200 mm. Ist eine Feldzeile in zwei oder mehr Felder unterteilt und die Gesamtbreite der einzelnen Feldzeilen unterschiedlich groß, so richtet sich die Gesamtbreite nach der insgesamt breitesten Feldzeile.



Höhe

(nur für den Typ Grafik) Legen Sie hier die minimale Höhe in Millimetern für das markierte Feld fest. Die maximale Höhe eines Feldes beträgt 200 mm.

Min./Max. Zeilenzahl

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

Ausrichtung

Wählen Sie hier die Ausrichtung des Inhalts in dem markierten Feld (9 Möglichkeiten).

Muster

Legen Sie hier die Hintergrundfarbe und -muster des Feldes fest. Durch Klick auf ... gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen können. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.

Schriftfarbe

(nur für den Typ Text) Die Schriftfarbe des Feldes wird hier angezeigt.

Die Farbzuordnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.

Schriftart

(nur für den Typ Text) Die Schriftart des Feldes wird hier angezeigt.

 Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**.

selektierte Eigenschaft in alle Felder übernehmen

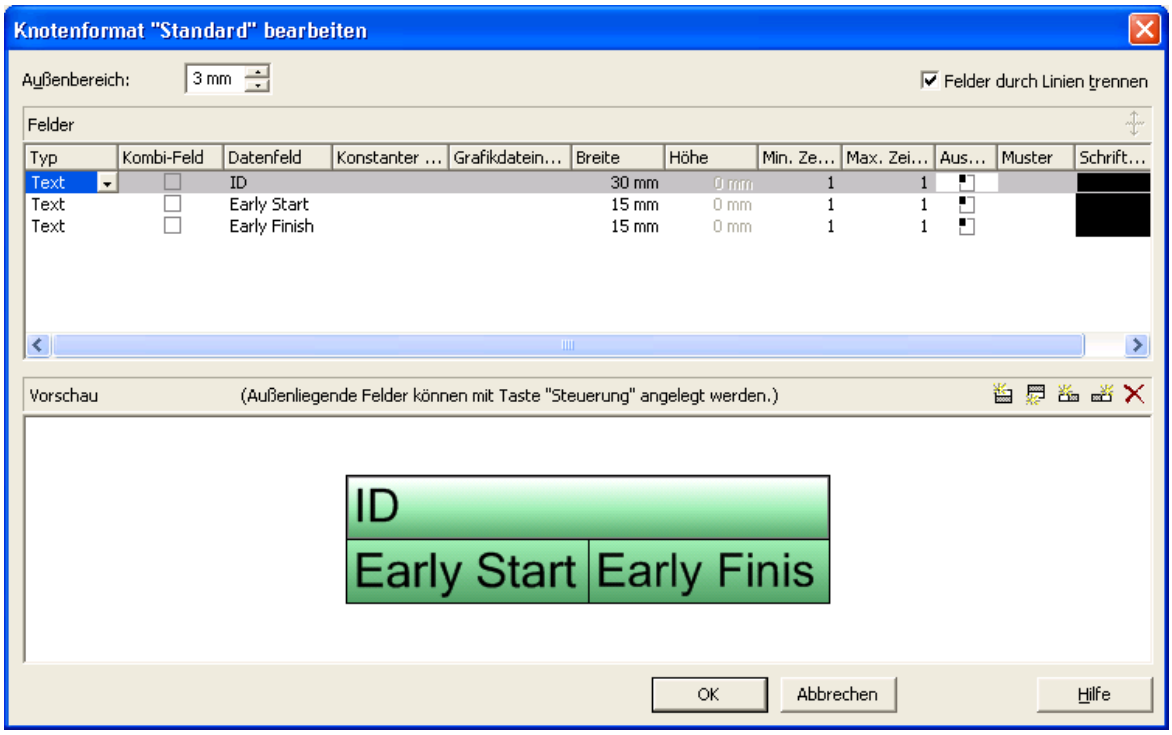
 Klicken Sie auf diese Schaltfläche, damit die selektierte Eigenschaft in alle Felder übernommen wird.

Vorschau

Hier werden die aktuellen Felder des Boxformats dargestellt. Wenn Sie im Vorschaufenster ein Feld anklicken, können Sie dessen Attribute in der **Felder**-Tabelle bearbeiten.

 Mit Hilfe der Schaltflächen oberhalb der Vorschau können Sie neue Felder anlegen oder das markierte Feld löschen. Zum Löschen von Feldern können Sie auch die Entf-Taste benutzen.

4.20 Dialogfeld "Knotenformat bearbeiten"



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

Außenbereich

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

Felder durch Linien trennen

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

Typ

Wählen Sie hier den Feldtyp (Text oder Grafik).

Kombi-Feld

Wenn dieses Kontrollkästchen aktiviert ist, können in dem Knotenfeld ein Text und eine Grafik kombiniert werden, und zwar folgendermaßen:

- **Typ**: Text, **Kombi-Feld**: nein: nur Text wird ausgegeben (wie unter **Datenfeld** oder unter **Konstanter Text** angegeben)
- **Typ**: Grafik, **Kombi-Feld**: nein: nur eine Grafik wird ausgegeben (wie unter **Grafikdateiname** angegeben)
- **Typ**: Text, **Kombi-Feld**: ja: Text (wie unter **Datenfeld** oder unter **Konstanter Text** angegeben) und Grafik (wie unter **Grafikdateiname** angegeben) werden ausgegeben
- **Typ**: Grafik, **Kombi-Feld**: ja: nur eine Grafik wird ausgegeben (wie unter **Grafikdateiname** angegeben). Text (unter **Datenfeld** angegeben) ist nur im Tooltip sichtbar; er kann aber, falls geeignet, als Hyperlink genutzt werden.)

Datenfeld

Wählen Sie hier das Datenfeld, dessen Inhalt in dem aktuellen Feld ausgegeben werden soll. Passt der Inhalt des Datenfeldes nicht in das Feld, wird der Überhang bei der Ausgabe abgeschnitten.

Konstanter Text

(nur wenn kein Datenfeld gewählt wurde) Sie können hier einen konstanten Text eingeben, der in dem Knotenfeld ausgeben werden soll.

Grafikdateiname

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

Wenn Sie auf ein **Grafikdateiname**-Feld klicken, erscheinen zwei Schaltflächen:

 Der Windows-Dialog **Grafikdatei auswählen** erscheint. Hier können Sie eine Grafikdatei auswählen, die in dem aktuellen Formatfeld erscheinen soll.

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

 Klicken Sie auf diese Schaltfläche (**Zuordnungen einstellen**), um eine Zuordnungstabelle zu verwenden, um Grafiken abhängig vom Inhalt eines Datenfelds in einem Knotenfeld erscheinen zu lassen.

Wird im Dialog **Zuordnung einstellen** nur ein Datenfeld, aber keine Zuordnungstabelle ausgewählt, so wird der Inhalt des eingestellten Datenfelds direkt als Name einer Grafikdatei interpretiert. Findet sich in dem Datenfeld bzw. in der Zuordnungstabelle kein gültiger Name einer Grafikdatei, dann wird der direkt in Spalte angegebene Dateiname verwendet.

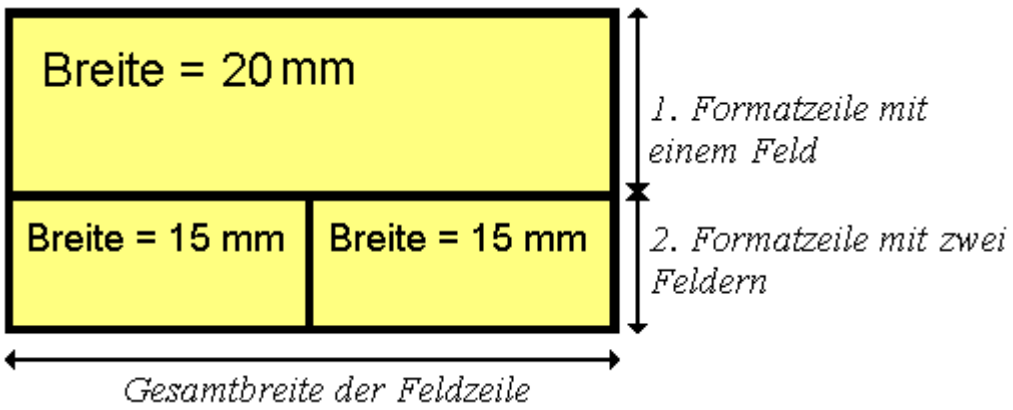
Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der 2. Schaltfläche fett dargestellt ().

 Sobald Sie die entsprechende Zeile verlassen, zeigt ein Symbol im Feld **Grafikdateiname** an, dass eine Zuordnung vorgenommen worden ist.

Bei der Darstellung der Grafik wird die Farbe des Pixels von ihrer Ecke links oben durch die Diagramm-Hintergrundfarbe ersetzt, d. h. alle Pixel der Grafik in dieser Farbe werden transparent dargestellt.

Breite

Legen Sie hier die Breite in Millimetern für das markierte Feld fest. Die maximale Breite eines Feldes beträgt 99 mm. Ist eine Feldzeile in zwei oder mehr Felder unterteilt und die Gesamtbreite der einzelnen Feldzeilen unterschiedlich groß, so richtet sich die Gesamtbreite nach der insgesamt breitesten Feldzeile.



Höhe

(nur für den Typ Grafik) Legen Sie hier die minimale Höhe in Millimetern für das markierte Feld fest. Die maximale Höhe beträgt 99 mm.

Min./Max. Zeilenzahl

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

Ausrichtung

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

Muster

Legen Sie hier Hintergrundfarbe und -muster des Feldes fest. Durch Klick auf  gelangen Sie in den Dialog **Musterattribute bearbeiten**, in dem Sie ein Muster, eine Hintergrundfarbe sowie eine 2. Musterfarbe auswählen können. Zusätzlich zu der vorgegebenen Farbpalette können Sie beliebige Farben definieren. Es stehen außerdem transparente Farben zur Verfügung.

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

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

Schriftfarbe

(nur für den Typ Text) Die Schriftfarbe des Feldes wird hier angezeigt, und Sie können sie hier bearbeiten. Wenn Sie auf dieses Feld klicken, erscheinen zwei Schaltflächen:

 Die Farbzuordnungstabelle öffnet sich, und Sie können daraus eine Schriftfarbe auswählen.

 Über die zweite Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**. Hier können Sie eine datenabhängige Zuordnung der Farbe vereinbaren. Wenn eine Zuordnung vorgenommen worden ist, wird der Pfeil auf der Schaltfläche fett dargestellt ().

Schriftart

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

selektierte Eigenschaft in alle Felder übernehmen

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

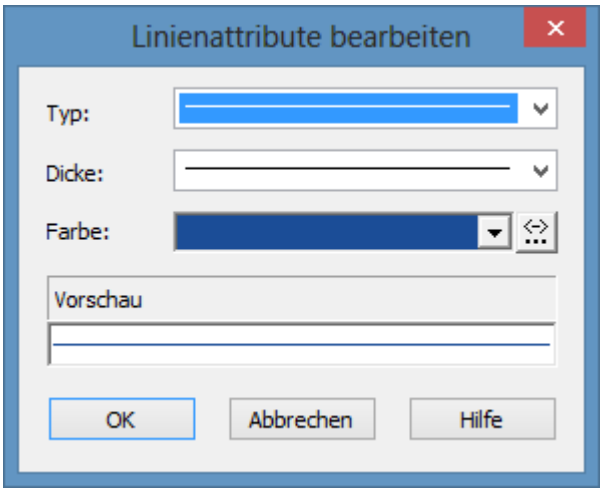
Vorschau

Hier wird das aktuelle Knotenformat dargestellt. Wenn Sie im Vorschaufenster ein Feld anklicken, können Sie dessen Attribute in der **Felder**-Tabelle bearbeiten.

 Mit Hilfe der Schaltflächen oberhalb der Vorschau können Sie neue Felder anlegen oder das markierte Feld löschen. Sie können auch die Entf-Tast benutzen, um Felder zu löschen.

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

4.21 Dialogfeld "Linienattribute bearbeiten"



Der Dialog zum Bearbeiten der Linienattribute, der jeweils über die Schaltfläche ... aufgerufen werden kann, steht zur Verfügung für das Verbindungsaussehen sowie für den Rahmen von Boxen.

Typ

Wählen Sie hier den Typ der Linie aus (durchgezogen, gestrichelt, etc.).

Dicke

Wählen Sie hier die Linienstärke aus.

Farbe

Wählen Sie hier die Farbe der Linien aus.

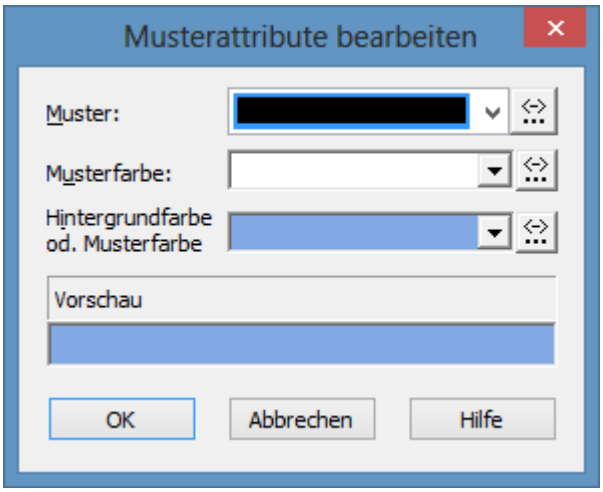
Über diese Schaltfläche gelangen Sie in den Dialog **Zuordnung einstellen**, in dem Sie eine datenabhängige Zuordnung der Linienfarbe vereinbaren können.

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

Vorschau

Hier wird das Aussehen der Linie mit den gewählten Einstellungen angezeigt.

4.22 Dialogfeld "Musterattribute bearbeiten"



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

Muster

Hier können Sie ein Hintergrundmuster auswählen.

Musterfarbe

Wählen Sie hier die Vordergrund-Farbe des Musters aus.

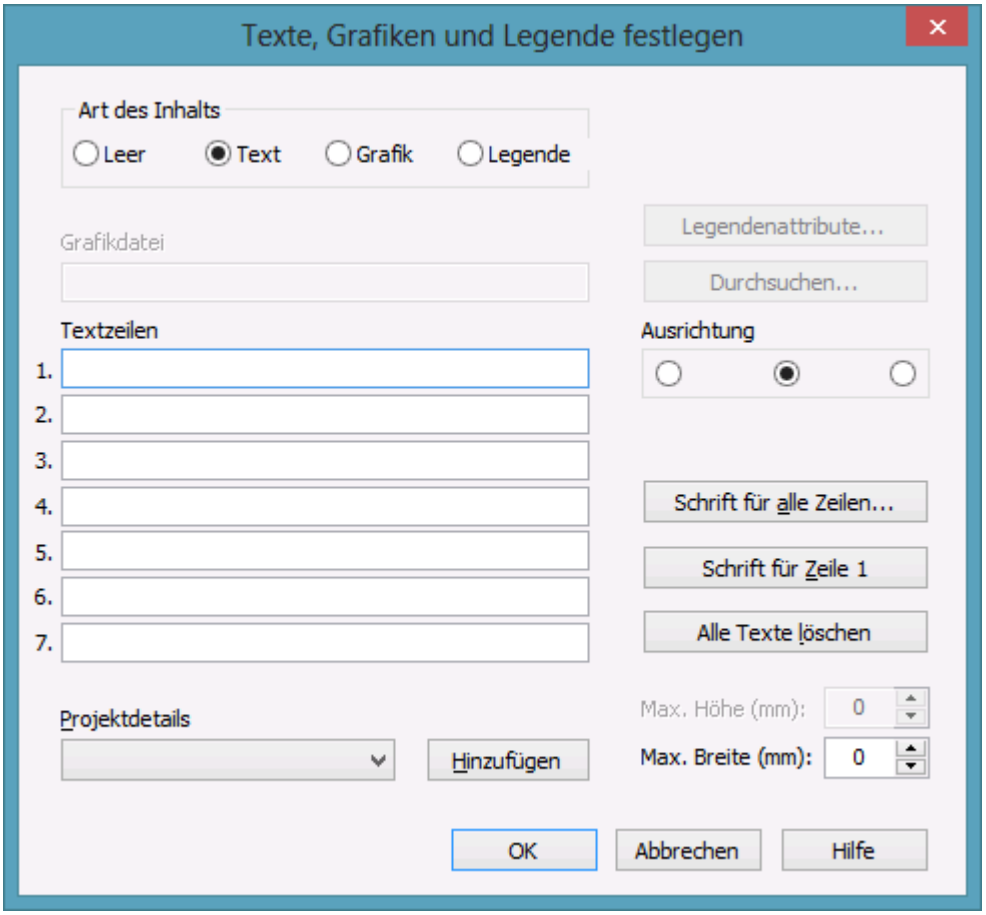
Hintergrundfarbe oder Musterfarbe 2

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

Vorschau

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

4.23 Dialogfeld "Texte, Grafiken und Legende festlegen"



Sie erreichen dieses Dialogfeld, indem Sie auf der Eigenschaftenseite **Außenbereich** auf eine der neun Schaltflächen ober- bzw. unterhalb der Grafik klicken.

Art des Inhalts

Wählen Sie hier die Art des Inhalts, der in dem zuvor gewählten Bereich der Darstellung ausgegeben werden soll:

- **Leer:** Der gewählte Diagrammbereich bleibt leer.
- **Text:** In dem gewählten Diagrammbereich wird der Text der sechs Textzeilen dargestellt.
- **Grafik:** Sie können in dem gewählten Bereich eine Grafik (z.B. Ihr Firmenlogo) plazieren. Grafiken werden immer mittig ausgerichtet.

- **Legende:** Eine Legende wird in dem gewählten Diagrammbereich ausgeben. Sie dokumentiert die Layer, die in der aktuellen Darstellung auftreten.

Die jeweils nicht benötigten Bereiche des Dialogs werden abhängig von Ihrer Auswahl deaktiviert. Dabei bleiben alle Angaben erhalten.

Legendenattribute

*Nur aktiv, wenn das Kontrollkästchen **Legende** angeklickt wurde.* Sie gelangen in den gleichnamigen Dialog, der für die Legende weitere Gestaltungsmöglichkeiten bietet.

Grafikdatei

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Tragen Sie hier den Namen der Grafikdatei ein. Falls sich die gewünschte Grafikdatei nicht im Installationsverzeichnis des VARCHART ActiveX befindet, müssen Sie das Laufwerk und den Pfad auch angeben.

Durchsuchen

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Wenn Sie auf diese Schaltfläche klicken, erscheint der Windows-Dialog **Grafikdatei auswählen**, mit dessen Hilfe Sie das Laufwerk, das Verzeichnis und den Dateinamen der Grafik festlegen können.

Textzeilen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Vereinbaren Sie den Text (max. 6 Zeilen), mit denen der gewählte Diagrammbereich beschriftet werden soll. Sie können einen beliebigen Text eintragen, aber auch Platzhalter (z.B. &[System-Datum]) für Projektdetails einsetzen. Sind alle sechs Textzeilen leer, wird dieser Bereich nicht dargestellt.

Projektdetails

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Hier können Sie dem Diagramm verschiedene Informationen (Anzahl der Seiten, Seitennummer, Systemdatum) hinzufügen, indem Sie aus der Kombobox den gewünschten Platzhalter auswählen und auf die Schaltfläche **Hinzufügen** klicken. Die Platzhalter werden in der Druckvorschau oder beim Ausdruck

durch die entsprechenden Daten ersetzt und stets auf dem aktuellen Stand gehalten.

Hinzufügen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Nachdem Sie aus der Liste ein Projektdetail ausgewählt haben, bestätigen Sie Ihre Wahl mit der Schaltfläche **Hinzufügen**. Die Projektdetails werden in die Zeile geschrieben, in der sich der Cursor gerade befindet.

Textausrichtung

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über die Kontrollkästchen können Sie die Textzeilen linksbündig, zentriert oder rechtsbündig ausrichten.

Schrift für alle Zeilen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**. Hier können Sie die Schriftart, Schriftgröße usw. für alle sechs Zeilen festlegen. Beim Ausführen dieser Aktion werden die Einstellungen für die Schrift der einzelnen Zeilen überschrieben.

Schrift für Zeile 1...6

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**. Hier können Sie die Schriftart, Schriftgröße usw. für die Zeile festlegen, in der sich der Cursor befindet.

Alle Texte löschen

*Nur aktiv, wenn das Kontrollkästchen **Text** angeklickt wurde.* Wenn Sie diese Schaltfläche anklicken, werden die Texte aller sechs Textzeilen gelöscht.

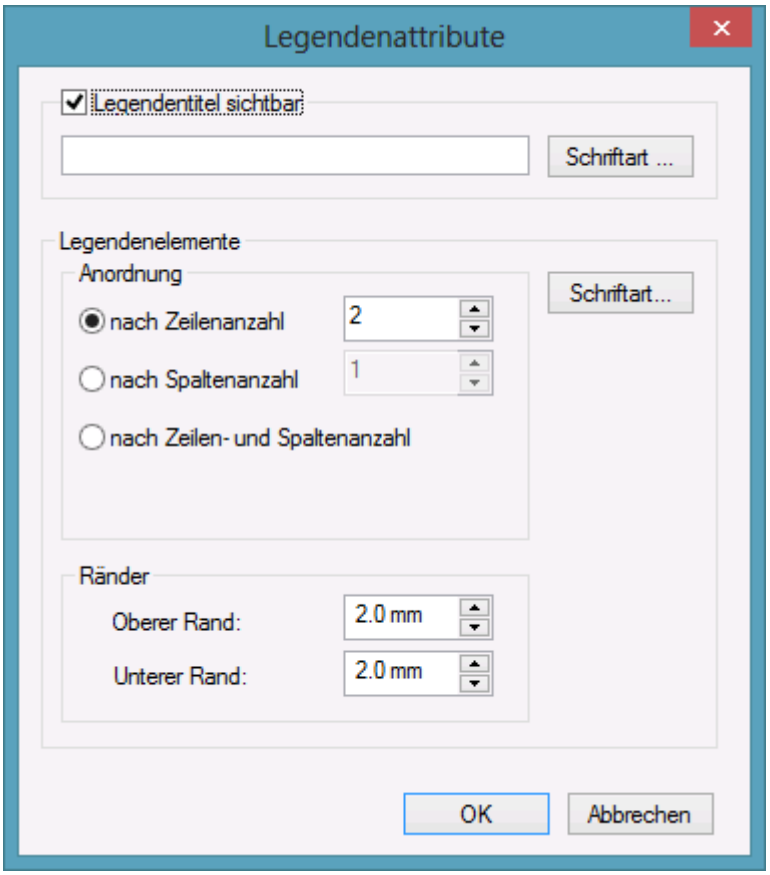
Max. Höhe (mm)

*Nur aktiv, wenn das Kontrollkästchen **Grafik** angeklickt wurde.* Falls mehrere Felder für Text, Grafik oder Legende vereinbart worden sind, können Sie hier die maximale Höhe des aktuellen Feldes festlegen. So können Sie verhindern, dass Feldinhalte abgeschnitten werden.

Max. Breite (mm)

*Nur aktiv, wenn das Kontrollkästchen **Text** oder **Grafik** angeklickt wurde.* Falls mehrere Felder für Text, Grafik oder Legende vereinbart worden sind, können Sie hier die max. Breite des aktuellen Feldes festlegen. So können Sie verhindern, dass Feldinhalte abgeschnitten werden.

4.24 Dialogfeld "Legendenattribute"



Sie erreichen dieses Dialogfeld zur Laufzeit über das Kontextmenü der Legende oder zur Designzeit über den Dialog **Texte, Grafiken und Legende festlegen** durch Klick auf die entsprechende Schaltfläche. Diese wird erst wählbar, nachdem Sie bei **Art des Inhalts Legende** ausgewählt haben.

Legendentitel sichtbar

Legen Sie hier fest, ob ein Legendentitel angezeigt werden soll und geben Sie einen Text ein. Durch Klick auf **Schriftart** öffnen Sie den gleichnamigen Windows-Dialog, in dem Sie die Schriftattribute für den Legendentitel festlegen können.

Anordnung

- nach Zeilenanzahl: Geben Sie hier an, ob und mit wie vielen Zeilen die Legende dargestellt werden soll.
- nach Spaltenanzahl: Geben Sie hier an, ob und mit wie vielen Spalten die Legende dargestellt werden soll.

- Nach Zeilen- und Spaltenanzahl: Die Legende wird in Spalten und Zeilen dargestellt. Ist die hier eingegebene Zahl niedriger als die vorhandenen Layern, so werden die überzähligen Layer nicht dargestellt.

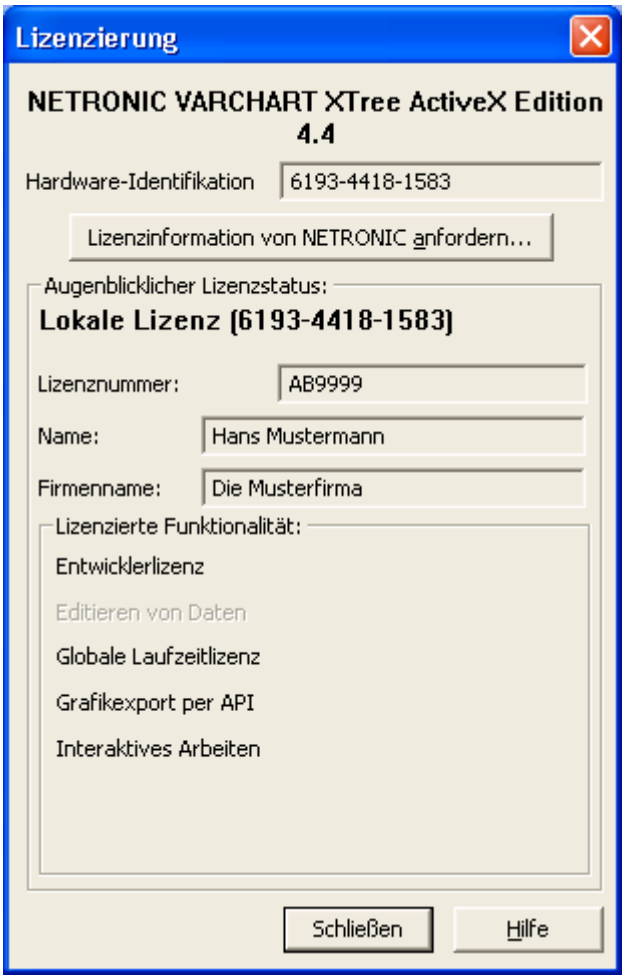
Ränder

- oberer Rand: Geben Sie ein Maß für den oberen Rand des Legendenelements an.
- unterer Rand: Geben Sie ein Maß für den unteren Rand des Legendenelements an.

Schriftart

Über diese Schaltfläche öffnen Sie den Windows-Dialog **Schriftart**, in dem Sie die Schriftattribute für die Legende festlegen können.

4.25 Dialogfeld "Lizenzierung"



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

Vor der Lizenzierung ist das Programm nur als Demoversion lizenziert. Folgende Einschränkungen gelten gegenüber der Vollversion: Die Nutzungsdauer zum Testen des Produkts ist auf 30 Tage begrenzt. Nach Ablauf dieses Zeitraums erscheint das Wort "Demo" im Diagramm.

Hardware-Identifikation

(nicht editierbar) Die Nummer, die in diesem Feld angezeigt wird, wird aus der Hardware-Konfiguration Ihres Rechners berechnet. Sie wird von NETRONIC Software GmbH für die Lizenzierung benötigt.

Bei Änderungen an Ihrer Hardware ist eine neue Lizenzierung erforderlich. Der Kundendienst von NETRONIC Software GmbH hilft Ihnen dann gern weiter.

Anfordern

Um die Lizenzierung vorzunehmen, klicken Sie auf diese Schaltfläche. Dann erscheint der Dialog **Lizenzinformationen anfordern**.

Lizenznummer/Name/Firmenname

(nicht editierbar) Hier werden Ihre Lizenznummer, Ihr Name und der Name Ihrer Firma angezeigt.

Lizenzierte Funktionalität

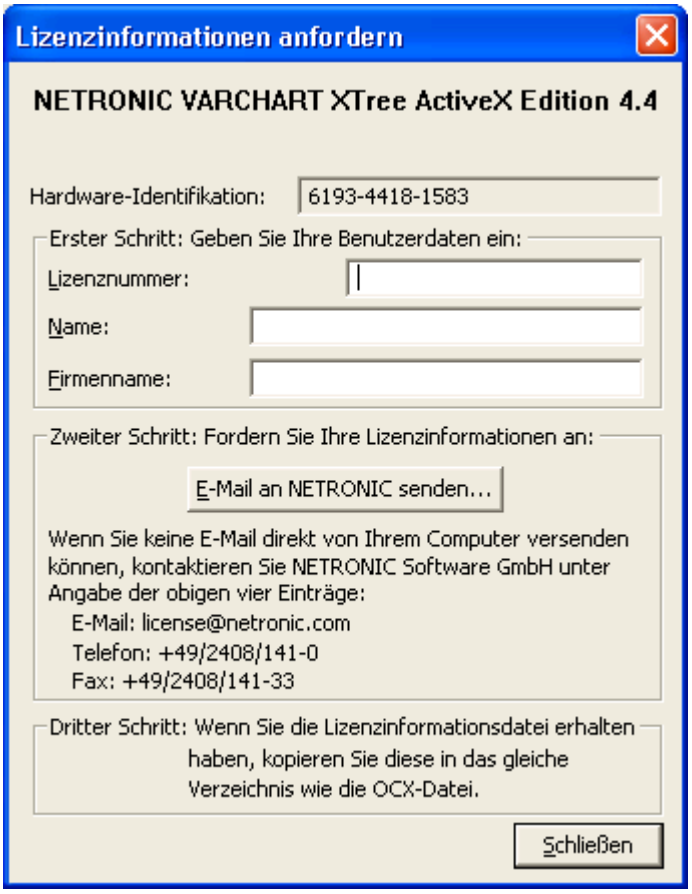
Hier wird angezeigt, welche Module für Sie freigegeben wurden. Wenn Sie die Lizenzierung vorgenommen haben, sind die freigegebenen Module aktiviert.

- **Entwicklerlizenz**
- **Globale Laufzeitlizenz** (das VARCHART ActiveX-Steuerelement läuft im Runtime-Modus auf jedem anderen Rechner.)
- **Einzelplatz-Laufzeitlizenzen** (das VARCHART ActiveX-Steuerelement muss auf jedem Rechner, auf dem es im Runtime-Modus laufen soll, einzeln lizenziert werden.)
- **Grafikexport per API**
- **Interaktives Arbeiten**

Schließen

Sie verlassen den Dialog.

4.26 Dialogfeld "Lizenzinformationen anfordern"



Geben Sie Ihre Lizenznummer, Ihren Namen und den Namen Ihrer Firma an und klicken Sie auf **E-Mail an NETRONIC senden**. Damit wird automatisch eine E-Mail generiert, die Sie nur noch absenden müssen. Sobald wir Ihre E-Mail erhalten haben, werden wir unverzüglich eine Datei mit Ihren Lizenzinformationen (**vctree.lic**) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei **vctree.ocx** steht.

Wenn Sie die neue Lizenzierung vorgenommen haben, müssen Sie diese in jedem Ihrer Projekte aktivieren. Öffnen Sie dazu in jedem Ihrer Projekte eine beliebige Eigenschaftenseite, nehmen Sie dort eine beliebige Änderung vor und speichern Sie diese. Nun ist die neue Lizenzierung aktiviert.

5 Benutzerschnittstelle

5.1 Übersicht

Die folgende Liste gibt einen Überblick über die Interaktionsmöglichkeiten für Anwender:

- Navigation im Diagramm
- Zoomen
- Knoten erzeugen
- Knoten markieren
- Knoten löschen, ausschneiden, kopieren und einfügen
- Knoten bearbeiten
- Verbindungsaussehen festlegen
- Knoten mit seinem Teilbaum umhängen
- Teilstrukturen horizontal und vertikal anordnen
- Teilstrukturen kollabieren und expandieren
- Legende bearbeiten
- Seite einrichten
- Druckvorschau verwenden

Kontextmenüs (rechte Maustaste):

- für das Diagramm
- für Knoten
- für die Legende

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

5.2 Navigation im Diagramm

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

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

Weitere Tastenkombinationen für die Navigation im Diagramm:

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

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

5.3 Zoomen

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

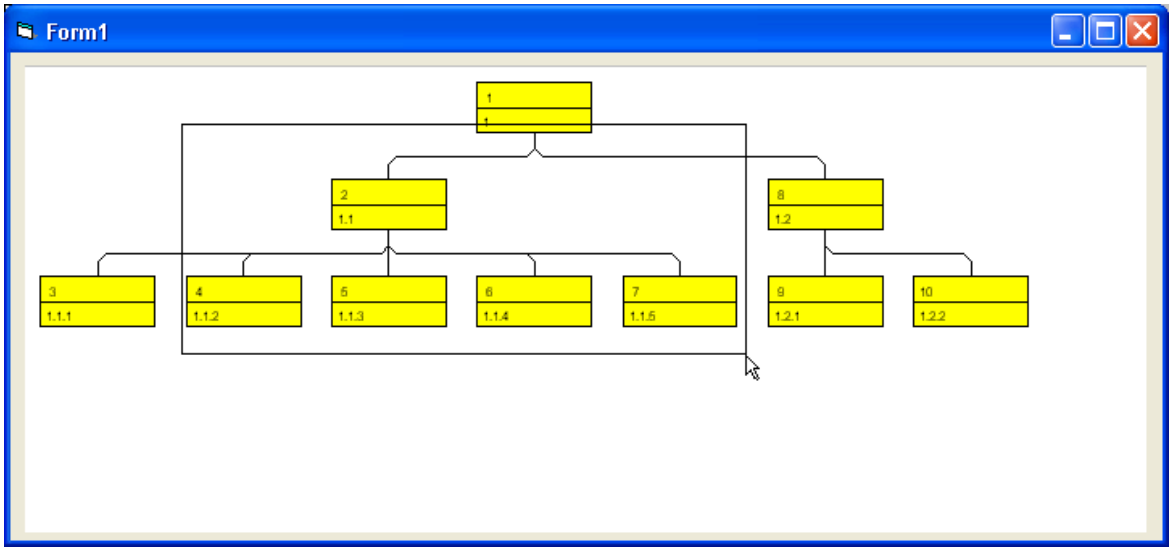
- **Strg** + **-** (Nummernblock): Verkleinern
- **Strg** + **+** (Nummernblock): Vergrößern

Auch die Maus kann zum Zoomen genutzt werden:

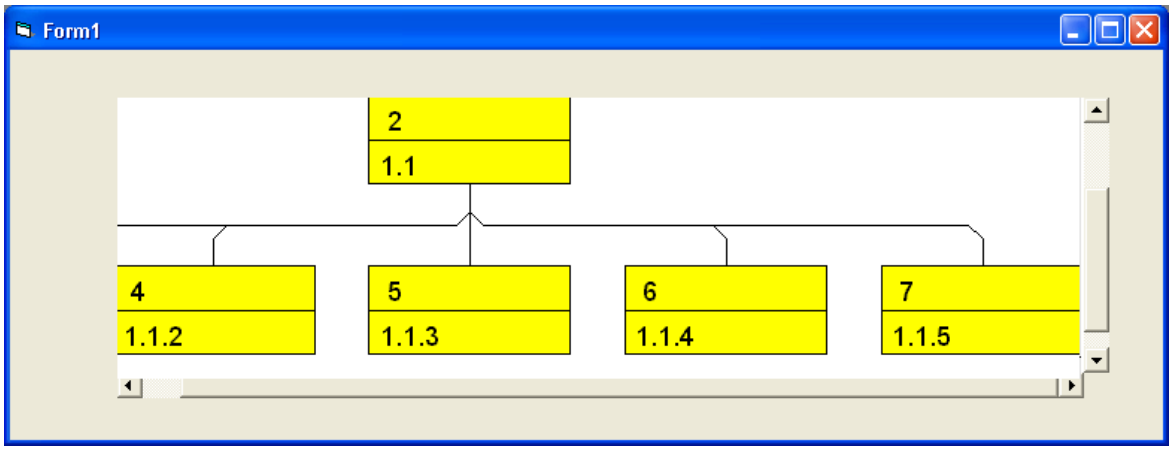
- Drehen Sie das Mausrad während Sie die Strg-Taste gedrückt halten. Dazu muss das Zoomen per Mausrad zugelassen sein. Dies geschieht entweder über die Option **Zoomen per Mausrad zulassen** auf der Eigenschaftenseite **Allgemeines** oder über die API-Eigenschaft **VcTree1.-ZoomingPerMouseWheelAllowed**. (Diese Eigenschaft st standardmäßig deaktiviert.)
- Sie können einen Ausschnitt Ihres Diagramms bildschirmfüllend darstellen lassen, indem Sie mit gedrückter linker Maustaste ein Rechteck um den zu vergrößernden Ausschnitt aufziehen und dann (bei noch gedrückter linker Maustaste) die rechte Maustaste drücken. Mit Hilfe der Bildlaufleisten können Sie dann das Fenster wie eine Lupe über der Darstellung verschieben und so auch die anderen Bereiche der Darstellung in derselben Vergrößerung betrachten.

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

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



vor dem Zoomen

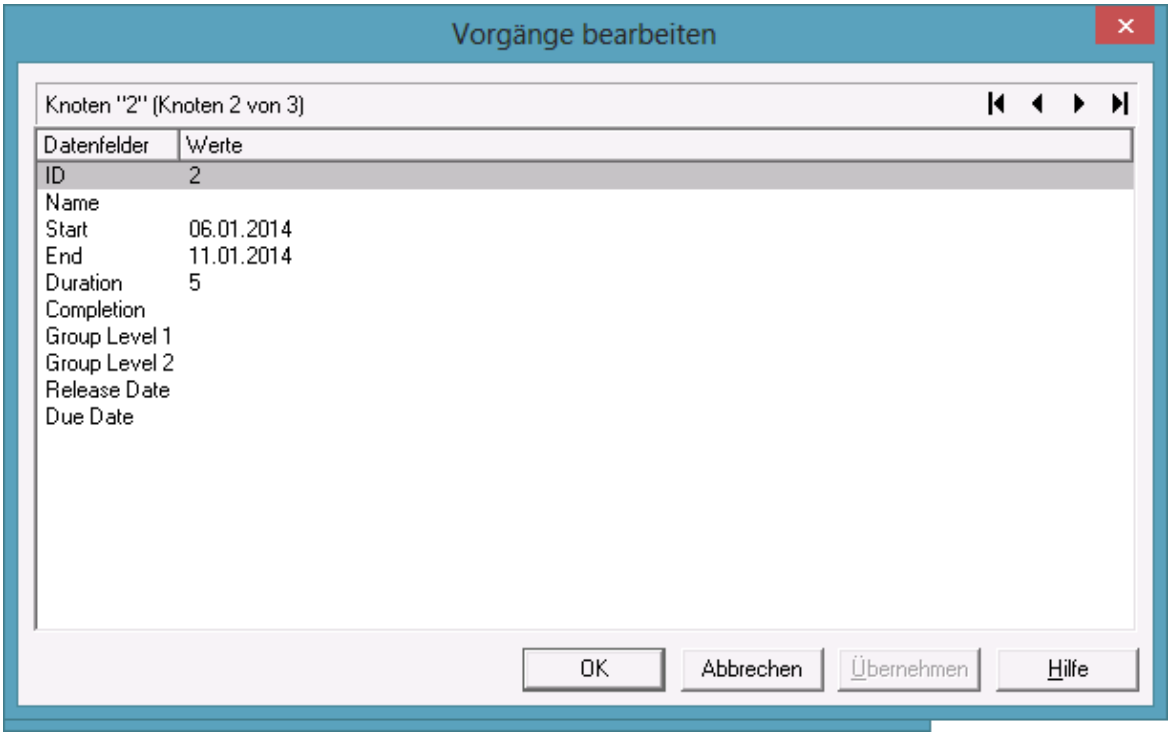


nach dem Zoomen

5.4 Knotendaten bearbeiten

Alle Daten eines Knotens können Sie im Dialogfeld **Vorgänge bearbeiten** bearbeiten. Sie erreichen das Dialogfeld durch einen Doppelklick auf einen Knoten oder über den Befehl <Bearbeiten> seines Kontextmenüs.

Um die Daten mehrerer Knoten zu bearbeiten, markieren Sie die gewünschten Knoten und wählen Sie aus dem Kontextmenü eines der markierten Knoten ebenfalls den Befehl **Bearbeiten** um das Dialogfeld **Vorgänge bearbeiten** zu öffnen. Jetzt können Sie nacheinander die Daten aller markierten Knoten bearbeiten



Durch einen Doppelklick auf einen Knoten wird das Ereignis **OnNodeLDblClick** ausgelöst.

Wenn ein Knoten interaktiv verändert worden ist (hier durch die Veränderung eines Wertes im Dialog **Vorgänge bearbeiten**), wird das Ereignis **OnNodeModify** ausgelöst. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird die Veränderung unterdrückt.

Datenfelder

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

als **versteckt** definiert wurden. Welche Datenfelder verfügbar sind, hängt von Ihrer Datendefinition ab.

Werte

Sie können in dieser Spalte alle Werte des markierten Knotens direkt bearbeiten, sofern sie im Dialog **Datentabellen verwalten** als **editierbar** definiert worden sind.

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

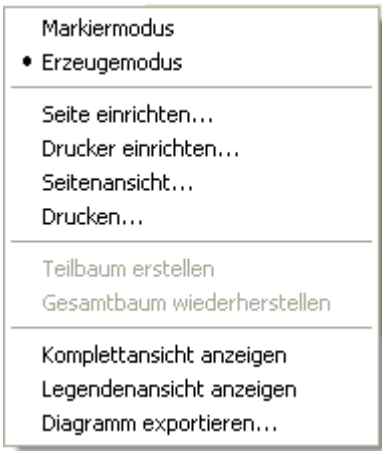
Das **Datumsausgabeformat** wird auf der Eigenschaftenseite **Allgemeines** festgelegt.

Wenn Sie ein Datenfeld vom Typ **Integer** bearbeiten, erscheint ein Spincontrol, mit dem Sie den gewünschten Wert einstellen können.

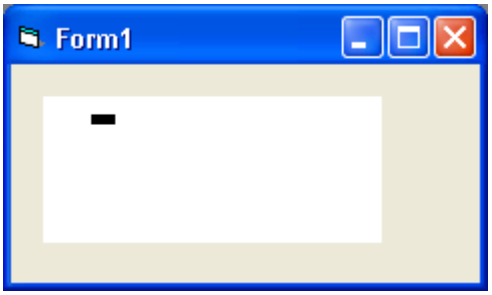
5.5 Knoten erzeugen

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

(Damit Sie Knoten interaktiv erzeugen können, muss auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten zulassen** aktiviert sein.)

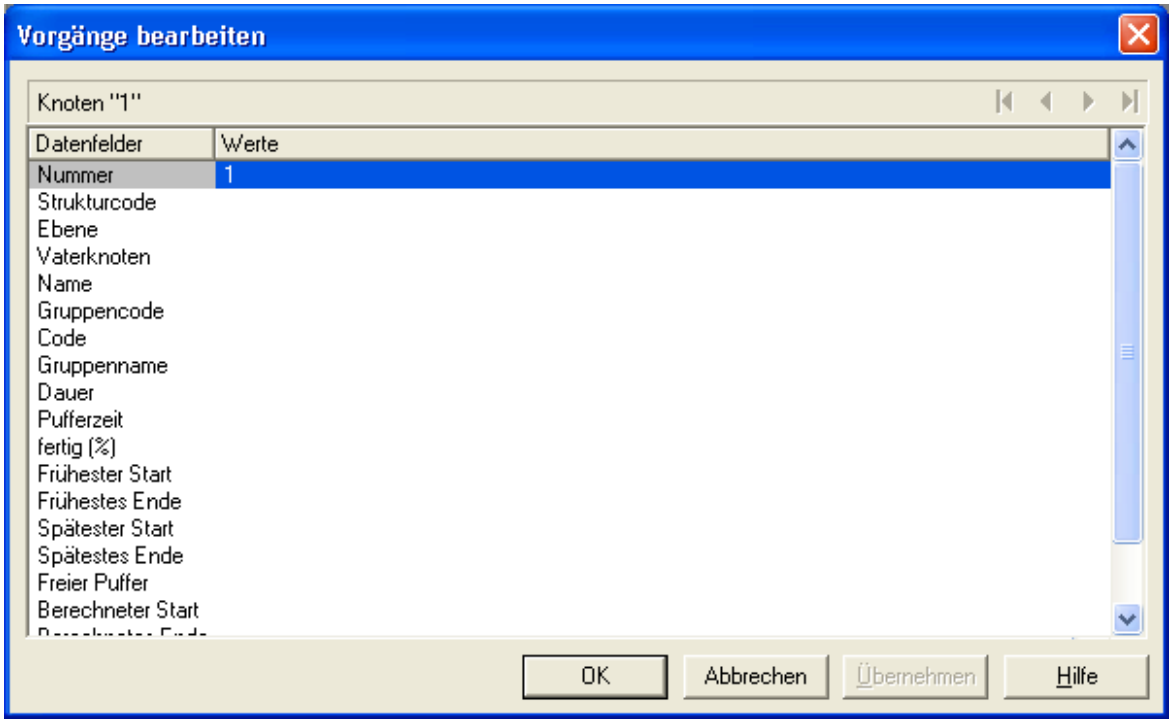


Im Erzeugemodus wird der Cursor im leeren Diagrammbereich zu einem Knotenphantom in der Form eines kleinen schwarzen Rechtecks.

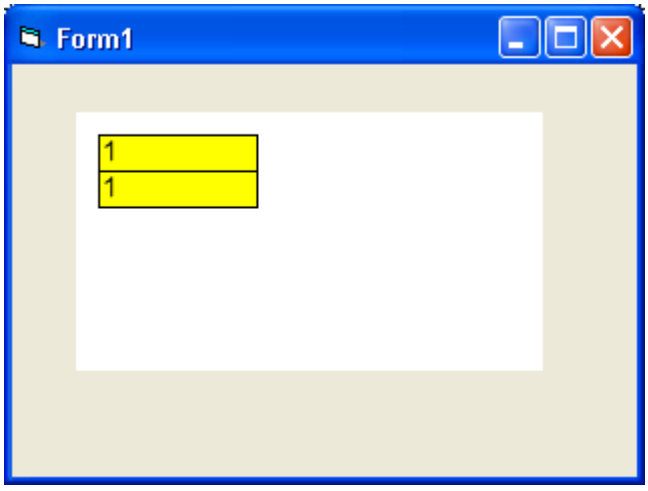


Klicken Sie nun einmal auf die linke Maustaste.

Wenn auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten bearbeiten** aktiviert ist, erscheint zunächst das Dialogfeld **Vorgänge bearbeiten**, in dem die Daten dieses Knotens angezeigt werden.

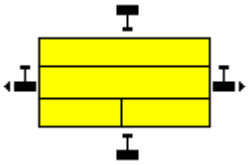


Zunächst ist nur das Datenfeld "Nummer" vorbesetzt (beim ersten Knoten mit dem Wert "1"). Sie können ggf. weitere Daten ergänzen, z. B. die Termindaten und eine Beschreibung. Sobald Sie die Daten mit **OK** bestätigen, wird der erste Knoten erzeugt.



War auf der Eigenschaftenseite **Allgemeines** die Option **Neue Knoten bearbeiten** nicht aktiviert, wird der erste Knoten angelegt, sobald Sie im Erzeugemodus mit der linken Maustaste in den Diagrammbereich klicken. Das Dialogfeld **Vorgänge bearbeiten** erscheint dann nicht.

Weitere Knoten können Sie durch Anlagern an einen bereits vorhandenen Knoten erzeugen. Wenn Sie den Cursor im Erzeugemodus in die Nähe eines Knotens führen, verändert der Cursor seine Form und zeigt an, wo der neue Knoten angelegt würde (als Vater, Sohn oder als rechter bzw. linker Bruder des Bezugsknotens).



5.6 Knoten markieren

Einzelnen Knoten markieren

Einen einzelnen Knoten markieren Sie, indem Sie ihn mit der linken Maustaste anklicken.

Knoten sammeln und toggeln

Sie können Knoten sammeln und toggeln, indem Sie sie bei gedrückter Strg-Taste mit der linken Maustaste anklicken. Jeder Klick auf einen markierten Knoten demarkiert diesen, jeder Klick auf einen nicht markierten Knoten markiert diesen.

Teilbaum markieren

Ein Teilbaum wird markiert, indem Sie bei gedrückter Umschalt-Taste den Vaterknoten des Teilbaums anklicken.

Alle Knoten demarkieren

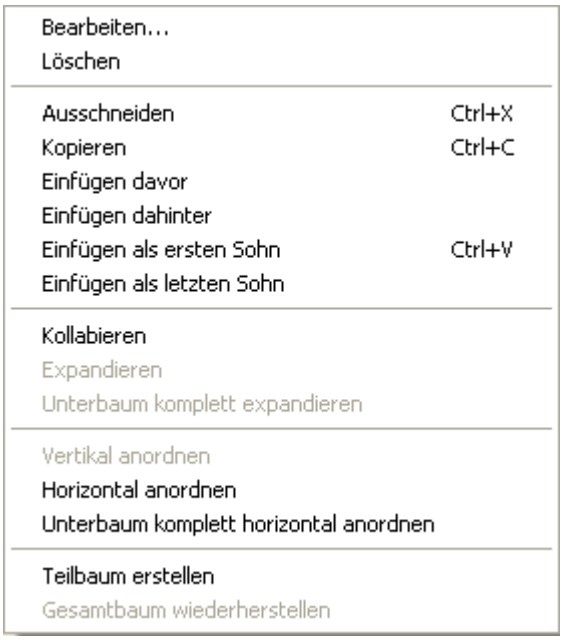
Sie können alle markierten Knoten demarkieren, indem Sie mit der linken Maustaste in den leeren Diagrammbereich klicken.

Markierungstyp festlegen

Auf der Eigenschaftenseite **Knoten** können Sie aus der Kombobox **Markierungstyp** auswählen, in welcher Weise Knoten markiert werden sollen.

5.7 Knoten löschen, ausschneiden, kopieren und einfügen

Mit Hilfe der entsprechenden Befehle des Kontextmenüs für Knoten können Sie die jeweils markierten Knoten löschen, ausschneiden, kopieren und die kopierten bzw. ausgeschnittenen Knoten einfügen.



Kontextmenü für Knoten

Damit ein Knoten eingefügt werden kann, muss ein Knoten markiert sein. Sie können dann wählen, ob der kopierte bzw. ausgeschnittene Knoten vor oder hinter dem markierten Knoten oder als erster oder letzter Sohn des markierten Knotens eingefügt werden soll.

Sie können markierte Knoten auch mit Hilfe der Entf-Taste löschen.

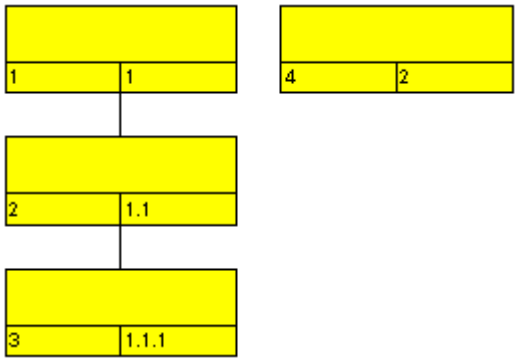
5.8 Knoten mit seinem Teilbaum umhängen

Sie können im Markiermodus einen Knoten mit dem gesamten darunter hängenden Teilbaum auf einmal per Drag & Drop umhängen. Es läßt sich immer nur ein einzelner Knoten mit seinem Teilbaum umhängen, auch wenn mehrere Knoten markiert sind.

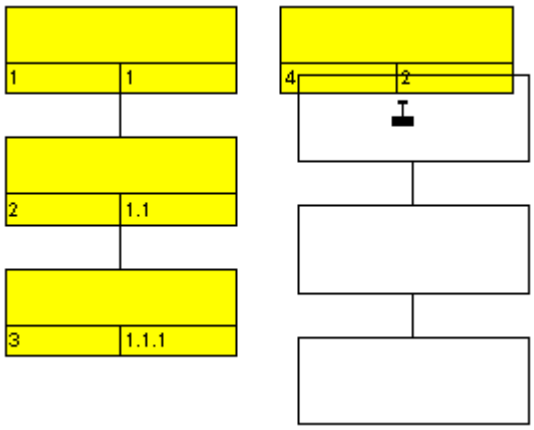
Markieren Sie den Knoten, den Sie zusammen mit seinen Sohnknoten umhängen möchten, und schieben Sie ihn mit gedrückter Maustaste an die gewünschte Position. Während Sie den Knoten verschieben, erscheint ein Phantom für den markierten Knoten mit seinem Teilbaum. Ziehen Sie das Phantom auf den Knoten, an dem Sie den verschobenen Knoten mit seinem Teilbaum anhängen möchten. Das Phantom des umgehängten Knotens muss dabei den Zielknoten teilweise überdecken. Sobald Sie die Maustaste loslassen, wird der Teilbaum umgehängt.

Beim Drag & Drop in Baum-Diagrammen wird der umgehängte Knoten stets dem Zielknoten als letzter Sohnknoten angehängt.

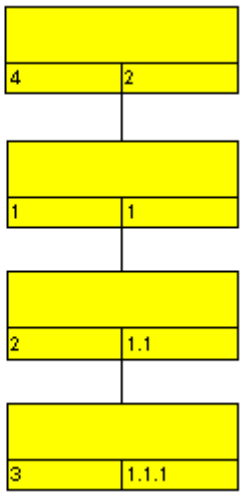
Beim Umhängen eines Knotens werden seine Verbindungen und Knotendaten automatisch angepasst.



Der linke Vaterknoten soll mit seinen Söhnen unter den rechten Knoten gehängt werden.



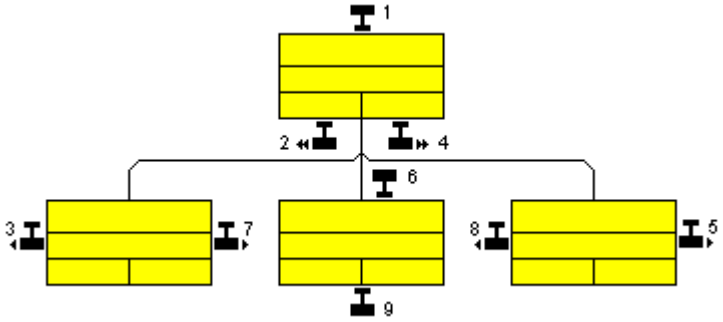
Während des Verschiebens erscheint ein Phantom, und der Cursor zeigt an, wo der verschobene Knoten mit seinem Teilbaum beim Loslassen der Maustaste angehängt würde.



Der verschobene Teilbaum wurde unter dem zuvor rechten Knoten angehängt.

Hinweis: Die Reihenfolge von Sohnknoten läßt sich nicht direkt, sondern nur indirekt durch gezieltes Umhängen der Sohnknoten unter denselben Vaterknoten verändern.

In der folgenden Abbildung sind die verschiedenen Anlagerungsmöglichkeiten bei horizontaler Anordnung dargestellt:



1: Neuer Vaterknoten für den gesamten Baum

2: Neuer Sohnknoten ganz links außen

4: Neuer Sohnknoten ganz rechts außen

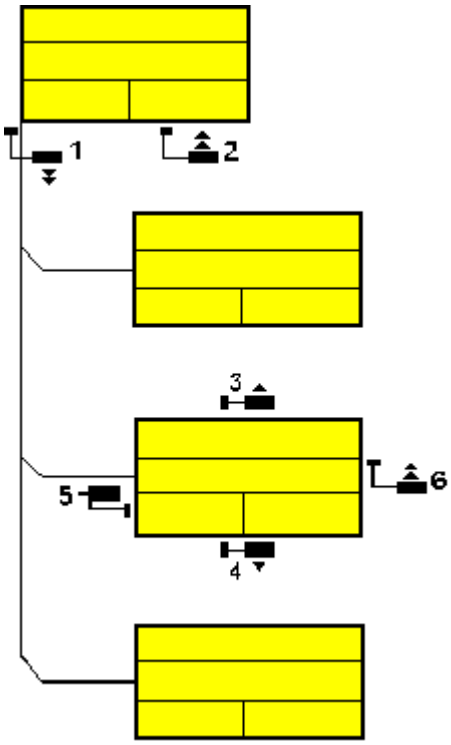
6: Neuer Vater zum Knoten

5, 7: Neuer Bruder rechts zum Knoten

3, 8: Neuer Bruder links zum Knoten

9: Neuer Sohn zum Knoten

Bei vertikaler Anordnung gibt es folgende Anlagerungsmöglichkeiten:



1: Neuer Sohnknoten ganz unten

2: Neuer Sohnknoten ganz oben

3: Neuer Bruder oberhalb

4: Neuer Bruder unterhalb

5: Neuer Vater zum Knoten

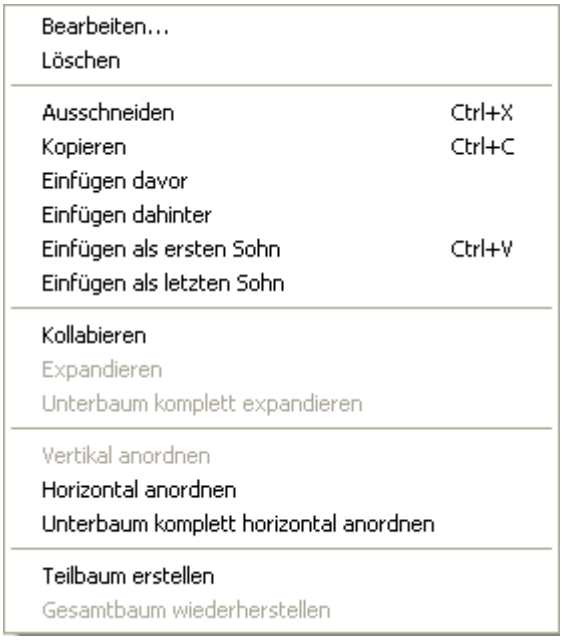
6: Neuer Sohn zum Knoten

5.9 Teilstrukturen horizontal und vertikal anordnen

Teilstrukturen von Baum-Diagrammen können ganz oder teilweise horizontal oder vertikal angeordnet werden.

- *Horizontale Anordnung:* Alle Knoten einer Ebene werden nebeneinander angeordnet. Der Port (Anknüpfungspunkt) der Verbindungslinie liegt mittig am unteren Rand des Vaterknotens und mittig am oberen Rand des Sohnknotens. Durch eine horizontale Anordnung beliebiger Teilstrukturen läßt sich die Höhe eines Baum-Diagramms verringern.
- *Vertikale Anordnung:* Alle Knoten einer Ebene werden untereinander angeordnet. Der Port der Verbindungslinie liegt in der unteren linken Ecke des Vaterknotens und in der Mitte des linken Randes des Sohnknotens. Durch eine vertikale Anordnung beliebiger Teilstrukturen läßt sich die Breite des Baum-Diagramms verringern.

Um die Befehle zum horizontalen und vertikalen Anordnen von Bäumen zur Verfügung zu haben, markieren Sie einen Knoten und klicken Sie auf die rechte Maustaste. Es erscheint das folgende Kontextmenü, wobei jeweils nur die verfügbaren Befehle aktiviert sind.

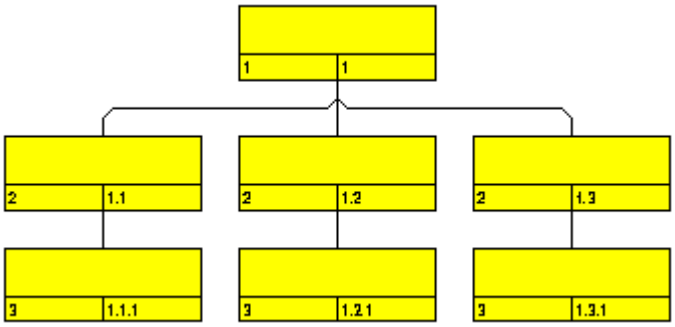


Um eine Teilstruktur eines Baum-Diagramms horizontal anordnen zu lassen, markieren Sie den obersten Knoten dieser Teilstruktur und wählen anschließend im Kontextmenü den Befehl **Horizontal anordnen**. Die Teilstrukturen des markierten Knotens werden dann horizontal angeordnet, aber nur eine Ebene tief. Auf die Anordnungen in den nächst tieferen Ebenen hat der Befehl **Horizontal anordnen** keine Auswirkung.

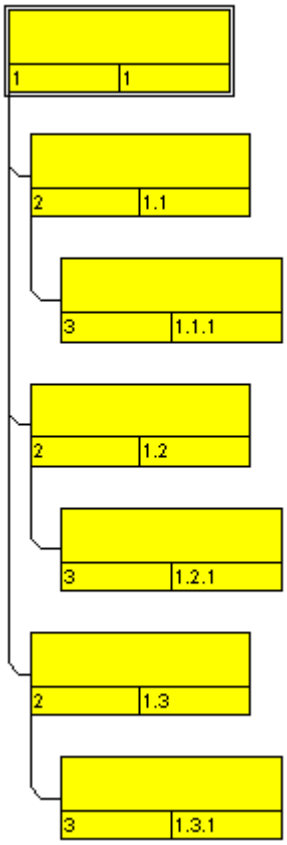
Mit dem Befehl **Unterbaum komplett horizontal anordnen** werden die Teilbäume unter den markierten Knoten vollständig, d. h. über alle Ebenen, horizontal angeordnet.

Mit dem Befehl **Vertikal anordnen** werden die Teilbäume unter den markierten Knoten vertikal angeordnet, und zwar beginnend beim höchsten ausgewählten Wurzelknoten.

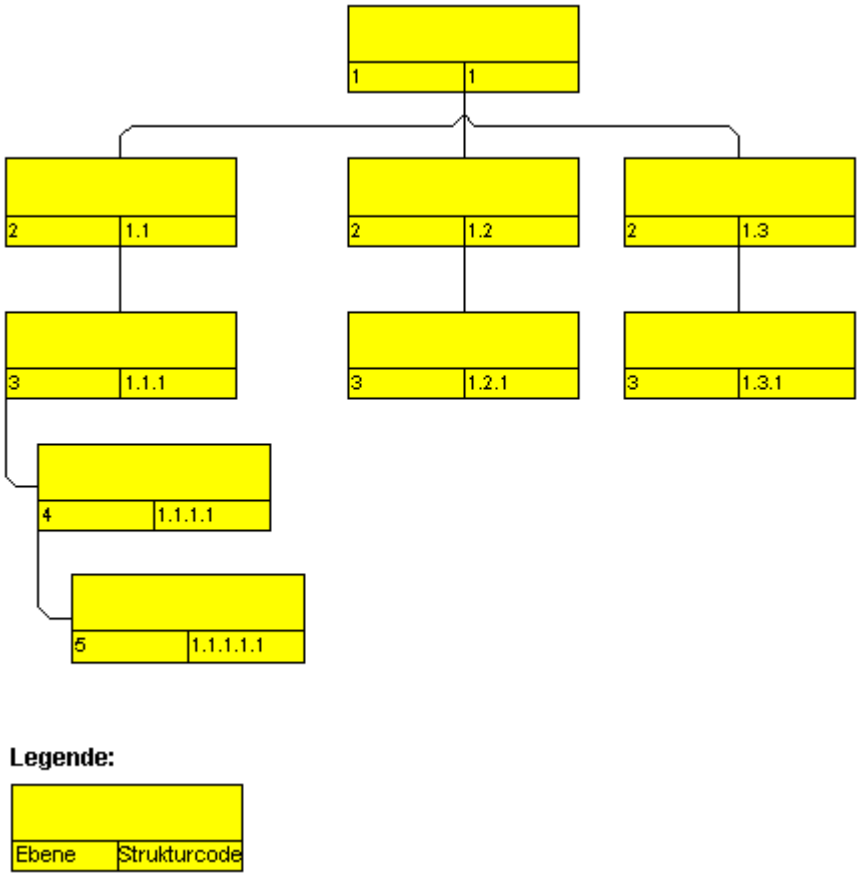
Hinweis: Falls ein Baum nicht vollständig vertikal angeordnet wird, prüfen Sie die vereinbarte maximale Baumhöhe auf der Eigenschaftenseite **Layout**. Die Anzahl der Ebenen bei der vertikalen Anordnung wird durch die Angabe unter **Max. Baumhöhe** begrenzt.



alle Ebenen horizontal angeordnet



alle Ebenen vertikal angeordnet



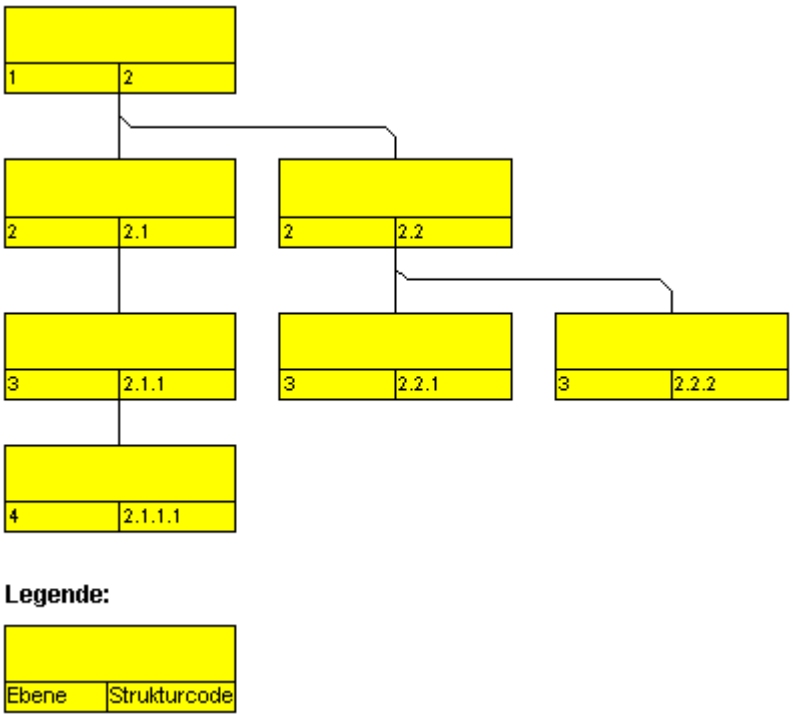
Beispiel für ein Baumdiagramm mit horizontal und vertikal angeordneten Teilstrukturen

Hinweis: Änderungen in der Anordnung wirken sich auch auf kollabierte Teilbäume aus.

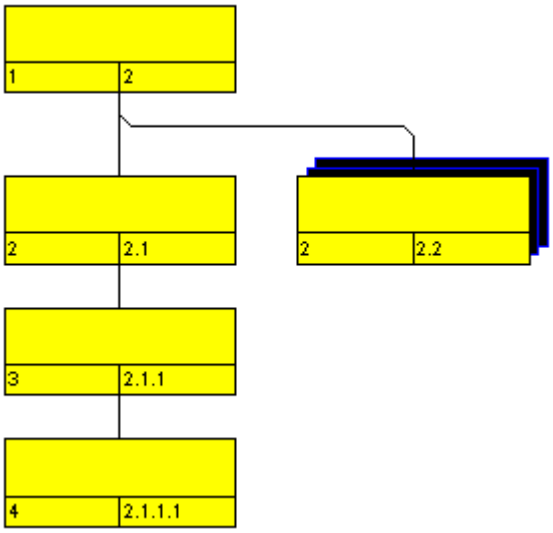
5.10 Teilstrukturen kollabieren und expandieren

Durch das Kollabieren von Teilstrukturen auf Gliederungsknoten lassen sich auch sehr komplexe Strukturen übersichtlich gestalten. Wenn Sie beispielsweise nur bestimmte Teilstrukturen präsentieren möchten, kollabieren Sie einfach alle anderen Teilstrukturen. Dadurch, dass die kollabierten Teilstrukturen erhalten bleiben, geht die Information über die Gesamtstruktur nicht verloren.

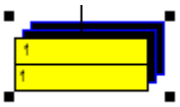
In Baum-Diagrammen läßt sich jede beliebige Teilstruktur auf einen einzigen Knoten, den Gliederungsknoten, kollabieren und auf Wunsch wieder in ihre ursprüngliche Form expandieren. Als Gliederungsknoten wird jeweils der Wurzelknoten der Teilstruktur, die kollabiert werden soll, verwendet.



Expandierter Baum



In Form eines Gliederungsknotens kollabierte Teilstruktur



Vollständig kollabierter Baum

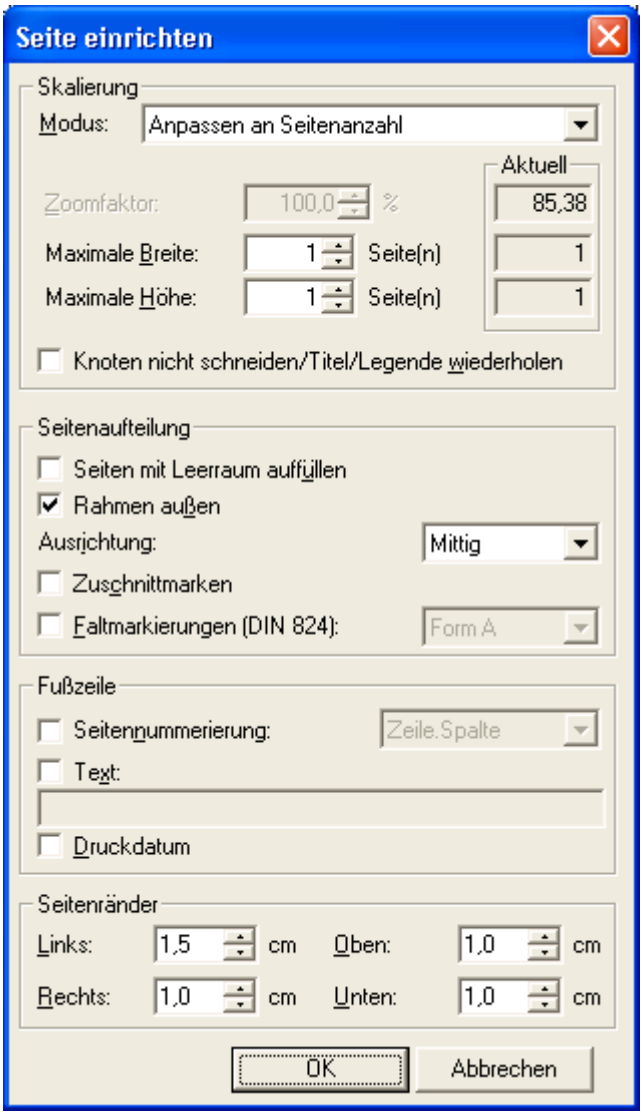
Mit dem Befehl **Kollabieren** aus dem Kontextmenü für Knoten können Sie die Teilstrukturen, die zu den markierten Wurzelknoten gehören, kollabieren. Die markierten Wurzelknoten verwandeln sich damit in Gliederungsknoten, die jeweils die nun verborgenen Teilstrukturen repräsentieren. Kollabierte Knoten werden automatisch mit dem Knotenaussehen "Kollabiert" versehen.

Mit dem Befehl **Expandieren** aus dem Kontextmenü für Knoten können Sie die Teilstrukturen wieder expandieren, die durch die markierten Gliederungsknoten repräsentiert werden. Es wird jeweils nur die nächsttieferliegende Ebene expandiert. Befinden sich in dieser Ebene weitere Gliederungsknoten, bleiben diese erhalten.

Mit dem Befehl **Unterbaum komplett expandieren** aus dem Kontextmenü für Knoten können Sie die markierten Gliederungsknoten vollständig expandieren.

5.11 Seite einrichten

Alle Einstellungen zum Seitenlayout können Sie im Dialog "Seite einrichten" vornehmen. Sie gelangen in diesen Dialog entweder über den entsprechenden Befehl im Diagramm-Kontextmenü über aus der Druckvorschau durch Klick auf die gleichnamige Schaltfläche.



Modus

Durch Auswahl einer Skalierungsart aus der Drop-Down-Liste und der entsprechenden Werte bei **Zoomfaktor** bzw. **Maximale Breite/Höhe** bestimmen Sie den Maßstab der Darstellung bei der Ausgabe. Nach Klick auf **Übernehmen** werden unter **Aktuell** die Werte angezeigt, die sich aus Ihren Einstellungen ergeben.

Zoomfaktor

Ein Skalierungsfaktor von 100 % entspricht der Originalgröße, ein kleinerer Wert bewirkt eine entsprechende Verkleinerung, ein größerer Wert eine Vergrößerung.

Anpassen an Seitenzahl

Durch Auswahl dieser Option können Sie die Anzahl der Seiten in Höhe und Breite vorgeben, auf die das Diagramm bei der Ausgabe maximal aufgeteilt werden soll (**Maximale Breite, Maximale Höhe**). Die Diagramme werden so groß wie möglich, aber ohne Verzerrungen dargestellt.

Knoten nicht durchtrennen/Titel/Legende wiederholen

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

Seiten mit Leerraum auffüllen

Mithilfe dieser Option können Sie festlegen, ob zwischen dem Diagramm und den Boxen für Titel und Legende so viel Platz gelassen wird, dass die Boxen auf jeder Druckseite immer in voller Breite gedruckt werden können und fest am Blattrand positioniert sind. Wenn diese Option nicht ausgewählt ist, werden die Boxen ohne Zwischenraum am Diagramm gedruckt und können dann je nach Diagramm auf den verschiedenen Druckseiten in der Breite variieren.

Rahmen außen

Aktivieren Sie dieses Kontrollkästchen, damit ein Rahmen um das Diagramm herum ausgegeben wird. Wenn die Option **Knoten nicht durchtrennen/ Titel wiederholen** ausgewählt ist, erhält jede Seite einen Rahmen, andernfalls wird ein Rahmen um das gesamte Diagramm gezogen.

Ausrichtung

Bestimmen Sie die Ausrichtung des Diagramms auf dem Blatt.

Zuschnittmarken

Wurde dieses Kontrollfeld aktiviert, wird das Diagramm mit Zuschnittmarken versehen, die das Zusammenkleben der ausgedruckten Einzelseiten zu einer Gesamtgrafik erleichtern.

Faltmarkierungen (DIN 824)

In der DIN Norm 824 ist für Bauzeichnungen eine ganz bestimmte Art der Faltung vorgeschrieben, mit der man die Zeichnung auf DIN A4-Größe zusammenfalten kann. Die Ausgabe von entsprechenden Faltmarkierungen auf Ihrem Diagramm erleichtert Ihnen die Faltung. Folgende Formate stehen zur Verfügung:

- **Form A:** mit Heftrand auf der linken Seite, damit die gefaltete Zeichnung gelocht und ohne Heftstreifen in einem Ordner abgeheftet werden kann.
- **Form B:** insgesamt etwas schmaler, damit ein Heftstreifen angebracht werden kann, der dann gemeinsam mit der Zeichnung die Breite von DIN A4 erreicht.
- **Form C:** die gefaltete Zeichnung wird nicht gelocht, sondern in eine Sichthülle gelegt

Die vorliegenden Faltmarkierungen können für jedwedes Zielformat ausgegeben werden, während die DIN 824 explizit nur die Formate DIN A0 bis A3 kennt.

Seitennummerierung

Ist dieses Kontrollkästchen aktiviert, wird auf jeder Seite unten links die Seitennummer ausgegeben. Folgende Möglichkeiten stehen dabei zur Auswahl:

- **Zeile.Spalte**: Sinnvoll, wenn das Diagramm sich auf mehr als eine Seite in der Länge als auch in der Breite erstreckt. Vor dem Punkt wird die Position der Seite in der vertikalen, dann die in der horizontalen Reihenfolge ausgegeben.
- **Spalte.Zeile**: Sinnvoll, wenn das Diagramm sich auf mehr als eine Seite in der Länge als auch in der Breite erstreckt. Vor dem Punkt wird die Position der Seite in der horizontalen, dann die in der vertikalen Reihenfolge ausgegeben.
- **Seite/Anzahl**: Zuerst erscheint die aktuelle Seitenzahl, danach die Anzahl der Gesamtseiten: 1/6, 2/6 etc.

Text

Aktivieren Sie dieses Kontrollkästchen, um jede Seite unten links mit einem beliebigen Text zu versehen. Dieser Text wird ggf. rechts von der Seitennummer ausgegeben.

Für die Seitennummerierung können Sie in die Zeile **Zusatztext** folgende Platzhalter eingeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{PAGE} = fortlaufende Seitennummer

{NUMPAGES} = Gesamtanzahl der Seiten

{ROW} = Zeilenposition des Ausschnitts im Gesamtdiagramm

{COLUMN} = Spaltenposition des Ausschnitts im Gesamtchart

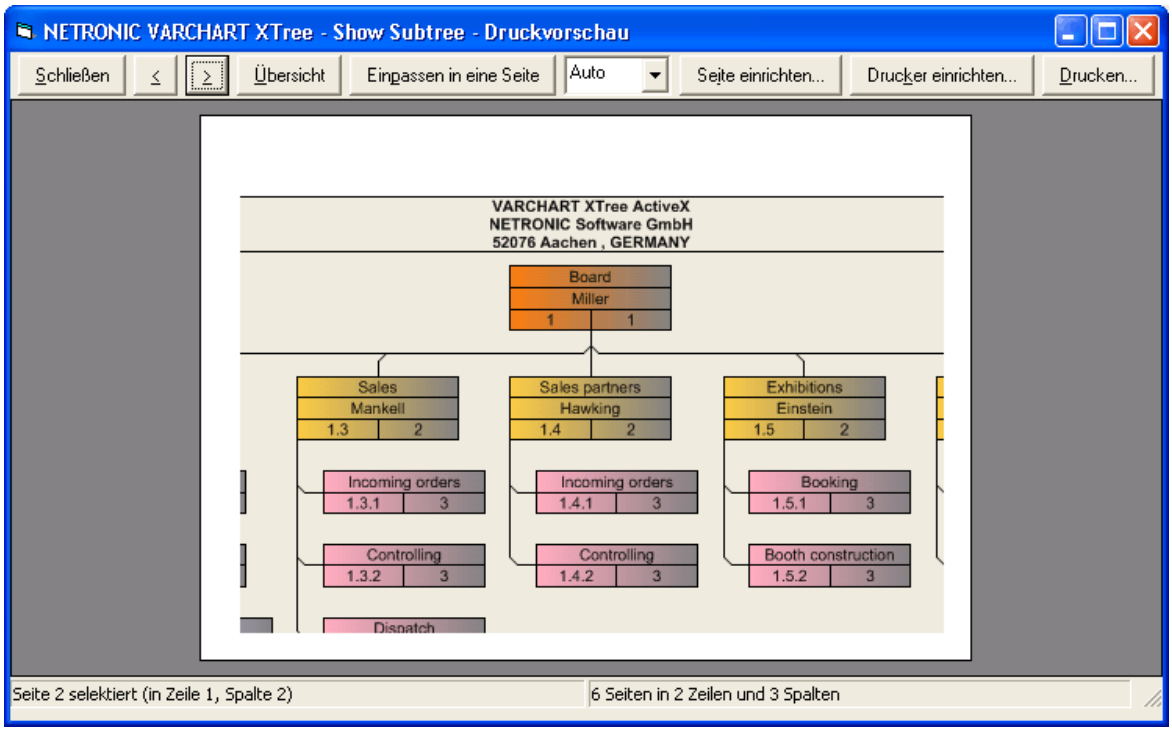
Druckdatum

Ist dieses Kontrollkästchen aktiviert, wird auf jeder Seite unten links das Druckdatum ausgegeben. Das Druckdatum wird ggf. rechts von der Seitennummer und dem Zusatztext ausgegeben.

Seitenränder

Über die Felder **Oben**, **Unten**, **Links** und **Rechts** legen Sie den Raum zwischen Papierrand und dem Diagramm in cm fest.

5.12 Druckvorschau



Vor dem Drucken können Sie das Diagramm in der Druckvorschau prüfen. Es wird so auf dem Bildschirm dargestellt, wie es im Dialogfeld **Seite einrichten** festgelegt ist und wie es gedruckt wird.

Sie können hier einzelne Seiten oder die Gesamtübersicht über alle Seiten Ihrer Darstellung ansehen oder einen Ausschnitt Ihres Diagramms interaktiv vergrößern und anschliessend drucken.

Schließen

Sie verlassen die Druckvorschau und gelangen zurück in die Darstellung.

<

Nur aktiv, wenn die Schaltfläche ***Einzel*** *gedrückt wurde*. Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten einzeln ansehen. Mit Hilfe dieser Schaltfläche gelangen Sie zur vorangehenden Seite. Sie bewegen sich über die Seiten von rechts nach links in aufsteigenden Zeilen.

>

Nur aktiv, wenn die Schaltfläche ***Einzel*** *gedrückt wurde*. Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten einzeln ansehen. Mit Hilfe dieser Schaltfläche gelangen Sie zur nächsten Seite. Sie bewegen sich über die Seiten von links nach rechts in absteigenden Zeilen.

Einzelseite/Übersicht

Wenn das Diagramm sich über mehrere Seiten erstreckt, können Sie sich die Seiten entweder einzeln oder in der Übersicht ansehen. In der **Übersicht** sehen Sie alle Seiten - je nach Seitenanzahl mehr oder weniger stark verkleinert, im Darstellungsmodus **Einzelseite**, wird zunächst die erste Seite des Diagramms einzeln und vergrößert dargestellt. Mit  oder  können Sie dann durch die Seiten blättern. Mit einem Doppelklick auf eine Seite wechseln Sie bequem zwischen den beiden Darstellungsarten **Einzelseite** und **Übersicht**.

Ein Ausschnitt Ihres Diagramms lässt sich im Darstellungsmodus **Einzelseite** interaktiv vergrößern, indem Sie bei gedrückter linker Maustaste ein Rechteck um den zu vergrößernden Bildausschnitt ziehen. Sobald Sie die linke Maustaste loslassen, wird der eingerahmte Bildausschnitt entsprechend vergrößert und statt der Schaltfläche **Drucken** erscheint die Schaltfläche 

über die Sie den Bildausschnitt dann in der aktuellen Vergrößerung drucken können. Beachten Sie bitte, dass der in der Druckvorschau interaktiv gewählte Vergrößerungsfaktor nicht den Skalierungsfaktor im Dialogfeld **Seite einrichten** verändert.

Einpassen in eine Seite

Mit dieser Schaltfläche läßt sich ein mehrseitiges Diagramm auf eine Seite verkleinern. Auch hier können Teile des Diagramms, wie unter **Einzelseite/Übersicht** beschrieben, interaktiv vergrößert und anschliessend gedruckt werden.

Zoomfaktor

Wählen Sie einen Zoomfaktor aus der Liste oder definieren einen individuellen, um die Darstellungsgröße ihres Diagramms für die Druckvorschau zu verändern. Dies ist nur im Modus "Einzelseite" möglich. Der Zoomfaktor lässt sich auch durch Drehen des Mausrades bei gedrückter <STRG>-Taste verändern. Er hat keinen Einfluss auf den späteren Druck. Je

nach gewählter Größe werden vertikale und/oder horizontale Bildlaufleisten angezeigt. Auch das Mausrad kann zum Bewegen des Bildes verwendet werden (ohne Umschalttaste vertikal, mit Umschalttaste horizontal).

Der Zoomfaktor **Auto** ist voreingestellt und verkleinert bzw. vergrößert das Blatt immer so, dass es bildschirmfüllend dargestellt wird.

Seite einrichten

Sobald Sie auf diese Schaltfläche klicken, gelangen Sie in das Dialogfeld **Seite einrichten** und können dort Änderungen am Seitenlayout vornehmen.

Drucker einrichten

*Nur sichtbar, wenn auf der Eigenschaftenseite **Allgemeines** die Option **PrintDlgEx-Dialog verwenden** nicht gewählt wurde.*

Sobald Sie auf diese Schaltfläche klicken, gelangen Sie in das Windows-Dialogfeld **Drucker einrichten** und können dort Änderungen an den Druckereinstellungen vornehmen.

Drucken/Ausschnitt drucken

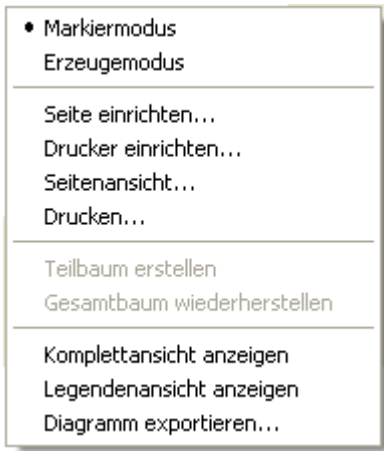
Über diese Schaltfläche gelangen Sie in den Windows-Dialog **Drucken** und können von dort aus den Druckvorgang einleiten.

Wenn Sie in der Druckvorschau einen Ausschnitt interaktiv gezoomt haben, ändert die Schaltfläche ihre Beschriftung zu **Ausschnitt drucken**. Wenn Sie sie anklicken, ist im Windows-Dialog **Drucken** die Option **Markierung** bereits ausgewählt. Durch Klick auf **OK** wird der am Bildschirm dargestellte Ausschnitt gedruckt.

Beachten Sie bitte, dass der in der Druckvorschau interaktiv gewählte Vergrößerungsfaktor nicht den Skalierungsfaktor im Dialogfeld **Seite einrichten** verändert.

5.13 Kontextmenü für das Diagramm

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



Markiermodus

Der Markiermodus ist der Standardmodus.

Erzeugemodus

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

Der Cursor wird im leeren Diagrammbereich zu einem Knotenphantom, das anzeigt, wo Sie durch einen Klick auf die linke Maustaste einen neuen Knoten erzeugen könnten.

Wenn auf der Eigenschaftenseite **Allgemeines** außerdem die Option **Neuen Knoten bearbeiten** aktiviert ist, öffnet sich beim Loslassen der Maustaste das Dialogfeld **Vorgänge bearbeiten**. Hier können Sie alle Daten des neu angelegten Knotens bearbeiten.

Der Erzeugemodus lässt sich auf zwei Arten aktivieren:

1. über das standardmäßige Kontextmenü für das Diagramm
2. über das Setzen der Eigenschaft **InteractionMode** auf den Wert **vcCreateNodes**.

Seite einrichten

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

Drucker einrichten

*Nur aktiv, wenn auf der Eigenschaftenseite **Allgemeines** die Option **PrintDlgEx Dialog verwenden** nicht gewählt wurde.*

Sie gelangen in das Windows-Dialogfeld **Drucker einrichten**.

Druckvorschau

Sie gelangen in das Dialogfeld **Druckvorschau**.

Drucken

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

Teilbaum erstellen

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

Gesamtbaum wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teilbaum erstellen** gewählt wurde)* Wählen Sie diese Option, um den Gesamtbaum wiederherzustellen.

Komplettansicht anzeigen

Über diesen Menüpunkt können Sie die Komplettansicht ein- oder ausschalten. Die Komplettansicht ist ein zusätzliches Fenster, in dem das komplette Diagramm angezeigt wird. Ein farbiger Rahmen markiert den aktuell im Hauptfenster dargestellten Diagrammausschnitt. Wenn Sie diesen Rahmen mit der Maus verschieben, wird auch im Hauptfenster der entsprechende Diagrammausschnitt angezeigt.

Die Komplettansicht lässt sich auch über die Eigenschaft **VcWorldView.Visible** an- oder abschalten.

Legendenansicht anzeigen

Über diesen Menüpunkt können Sie die Legendenansicht ein- oder ausschalten. Die Legende erscheint in einem zusätzlichen Fenster.

Sie können die Legendenansicht auch über die Eigenschaft **VcLegendView.Visible** an- oder abschalten.

Diagramm exportieren

Wenn Sie diese Option wählen, gelangen Sie in das Windows-Dialogfeld **Speichern unter**, in dem Sie das dargestellte Diagramm als Grafikdatei speichern können.

Sie können diesen Dialog auch mit der VcTree Methode **ShowExport-GraphicsDialog** aufrufen.

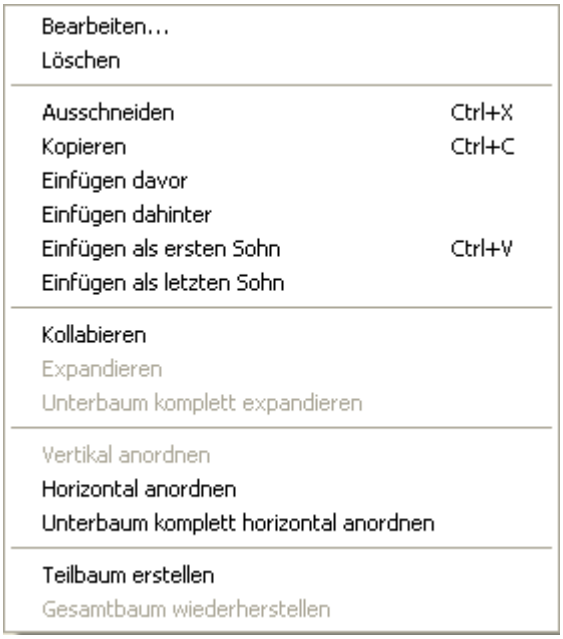
Beim Exportieren wird die Größe des exportierten Diagramms in Pixeln wie folgt berechnet:

- PNG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen.
- GIF, TIFF, BMP, JPEG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.
- WMF: Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.
- EMF/EMF+: Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand verwendet.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten finden Sie im Kapitel: "Wichtige Konzepte: Grafikformate".

5.14 Kontextmenü für Knoten

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



Bearbeiten

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

Löschen

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

Ausschneiden

Mit diesem Befehl können Sie die markierten Knoten ausschneiden.

Kopieren

Mit diesem Befehl können Sie die markierten Knoten kopieren.

Einfügen davor

Damit ein Knoten eingefügt werden kann, muss ein Knoten markiert sein. Wählen Sie diesen Befehl, um den kopierten bzw. ausgeschnittenen Knoten vor dem markierten Knoten einzufügen.

Einfügen dahinter

Wählen Sie diesen Befehl, um den kopierten bzw. ausgeschnittenen Knoten hinter dem markierten Knoten einzufügen.

Einfügen als ersten Sohn

Wählen Sie diesen Befehl, um den kopierten bzw. ausgeschnittenen Knoten als ersten Sohn des markierten Knotens einzufügen.

Einfügen als letzten Sohn

Wählen Sie diesen Befehl, um den kopierten bzw. ausgeschnittenen Knoten als letzten Sohn des markierten Knotens einzufügen.

Kollabieren

Mit diesem Befehl können Sie die Teilstrukturen, die zu den markierten Wurzelknoten gehören, wegklappen (kollabieren). Die markierten Wurzelknoten verwandeln sich damit in Gliederungsknoten, die jeweils die nun verborgenen Teilstrukturen repräsentieren.

Dadurch, dass die kollabierten Teilstrukturen erhalten bleiben, geht die Information über die Gesamtstruktur nicht verloren.

Expandieren

Mit diesem Befehl können Sie die Teilstrukturen wieder aufklappen (expandieren), die durch die markierten Gliederungsknoten repräsentiert werden. Es wird jeweils nur die nächsttieferliegende Ebene expandiert. Befinden sich in dieser Ebene weitere Gliederungsknoten, bleiben diese erhalten.

Unterbaum komplett expandieren

Mit diesem Befehl können Sie die markierten Gliederungsknoten vollständig expandieren.

Vertikal anordnen

Um die Breite des Baum-Diagramms zu verringern, können Sie beliebige Teilstrukturen vertikal anordnen. Bei vertikaler Anordnung werden alle Knoten einer Ebene untereinander angeordnet. Der Port der Verbindungslinie

liegt dann in der unteren linken Ecke des Vaterknotens und in der Mitte des rechten Randes des Sohnknotens.

Mit dem Befehl **Vertikal anordnen** werden die Teilbäume unter den markierten Knoten vertikal angeordnet, und zwar beginnend beim höchsten ausgewählten Wurzelknoten.

Hinweis: Falls ein Baum nicht vollständig vertikal angeordnet wird, prüfen Sie die vereinbarte maximale Baumhöhe auf der Eigenschaftenseite **Layout**. Die Anzahl der Ebenen bei der vertikalen Anordnung wird durch die Angabe unter **Max. Baumhöhe** begrenzt.

Horizontal anordnen

Um die Höhe eines Baum-Diagramms zu verringern, können Sie beliebige Teilstrukturen horizontal anordnen. Bei horizontaler Anordnung werden alle Knoten einer Ebene nebeneinander angeordnet. Die Verbindungslinie beginnt dann mittig am unteren Rand des Vaterknotens und endet mittig am oberen Rand des Sohnknotens.

Um eine Teilstruktur eines Baum-Diagramms horizontal anordnen zu lassen, markieren Sie den obersten Knoten dieser Teilstruktur und wählen Sie anschließend den Befehl **Horizontal anordnen**. Die Teilstrukturen des selektierten Knotens werden dann horizontal angeordnet, aber nur eine Ebene tief. Auf die Anordnungen in den nächst tieferen Ebenen hat der Befehl **Horizontal anordnen** keine Auswirkung.

Unterbaum komplett horizontal anordnen

Mit diesem Befehl werden die Teilbäume unter den markierten Knoten vollständig, d. h. über alle Ebenen, horizontal angeordnet.

Teilbaum erstellen

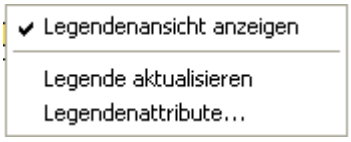
Wählen Sie diese Option, um einen Teilbaum aus den markierten Knoten darzustellen.

Gesamtbaum wiederherstellen

*(nur aktiv, wenn zuvor die Option **Teilbaum erstellen** gewählt wurde)* Wählen Sie diese Option, um den Gesamtbaum wiederherzustellen.

5.15 Kontextmenü für die Legende

Wenn Sie mit der rechten Maustaste auf die Legende klicken, erscheint das folgende Menü:



Legendenansicht anzeigen

Über diesen Menüpunkt können Sie die Legendenansicht ein- oder ausschalten.

Legende aktualisieren

Über diesen Menüpunkt können Sie die Legendenansicht aktualisieren.

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

Legendenattribute

Über diesen Menüpunkt öffnen Sie den gleichnamigen Dialog, in dem Sie Einstellungen zum Legendentitel, den Legendenelementen und zu den Rändern der Legende vornehmen können. Weitere Information zu diesem Dialog finden Sie in Kapitel 4.44 "Dialogfeld Legendenattribute".

6 Häufig gestellte Fragen

6.1 Wie kann das Steuerelement neu lizenziert werden? Was ist bei Problemen mit der Lizenzierung zu tun?

Wenn Sie die Nutzungsdauer des VARCHART ActiveX-Steuerelementes verlängern oder ein neu erworbenes Modul lizenzieren möchten, öffnen Sie bitte den Dialog **Lizenzierung**.

Sie erreichen diesen Dialog über die Eigenschaftenseite **Allgemeines**. Um die Lizenzierung vorzunehmen, klicken Sie auf die **Anfordern**-Schaltfläche. Dann erscheint der Dialog **Lizenzinformationen anfordern**.

Geben Sie dort Ihre Lizenznummer, Ihren Namen und den Namen Ihrer Firma an und klicken sie auf **E-Mail an NETRONIC senden**. Damit wird automatisch eine E-Mail generiert, die Sie nur noch absenden müssen. Sobald wir Ihre E-Mail erhalten haben, werden wir unverzüglich eine Datei mit Ihren Lizenzinformationen (vctree.lic) generieren und sie Ihnen zusenden. Bitte kopieren Sie dann diese Datei in das Verzeichnis, in dem die Datei *vctree.ocx* steht.

Wenn Sie die neue Lizenzierung vorgenommen haben, müssen Sie die neue Lizenzierung noch aktivieren. Öffnen Sie dazu eine beliebige Eigenschaftenseite, nehmen Sie dort eine beliebige Änderung vor und speichern Sie diese. Nun ist die neue Lizenzierung aktiviert.

Wenn Sie beim Lizenzieren des VARCHART ActiveX-Steuerelementes die Fehlermeldung "REGSVR32 Error Return: 0X0000007e" erhalten, ist die DLL *vcwin32u.dll* nicht vorhanden oder steht nicht in einem im PATH angegebenen Verzeichnis. Sollte die Datei fehlen, wenden Sie sich bitte an den Kundendienst der NETRONIC Software GmbH.

6.2 Wie kann erreicht werden, dass eine veränderte ini-Datei des VARCHART ActiveX-Steuerelementes verwendet wird?

Einige Einstellungen für das VARCHART ActiveX-Steuerelement können auf den Eigenschaftenseiten nicht eingestellt werden, lassen sich durch das Editieren der ini-Datei aber trotzdem verändern. Gehen Sie dazu folgendermaßen vor:

1. Öffnen Sie die Eigenschaftenseite **Allgemeines**. Dort wird im Feld **Konfigurationsdatei** die aktuell verwendete Konfigurationsdatei (z.B. *projekt.ini*) angezeigt.
2. Klicken Sie auf die Schaltfläche **Durchsuchen**. Das Dialogfeld **Laden bzw. Speichern** öffnet sich. Geben Sie hier unter **Dateiname** den Namen für eine Konfigurationsdatei an, die nur vorübergehend verwendet werden soll, z. B. *dummy.ini*. Klicken Sie auf **Speichern**.
3. Klicken Sie nun auf der Eigenschaftenseite **Allgemeines** auf **OK** bzw. **Übernehmen**. Die Konfigurationsdatei *dummy.ini* wird nun automatisch erzeugt und verwendet.
4. Nun können Sie Ihre ini-Datei (z. B. *projekt.ini*) in einem Editor (z. B. Microsoft WordPad) bearbeiten und Ihre Änderungen speichern.
5. Wählen Sie anschließend auf der Eigenschaftenseite **Allgemeines** unter **Konfigurationsdatei** Ihre bearbeitete Konfigurationsdatei (*projekt.ini*) wieder aus und klicken Sie auf **OK**. Nun wird die bearbeitete ini-Datei (*projekt.ini*) des VARCHART ActiveX-Steuerelementes verwendet.

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

Nach dem Upgrade oder Update von VARCHART XTree auf eine neue Version ist es notwendig, diese neue Version in dem Delphi Package Borland Anwenderkomponenten neu zu installieren. Dazu gehen Sie bitte wie folgt vor:

- Starten Sie Borland Delphi.
- Klicken Sie auf **Komponente** und **ActiveX importieren**.
- Selektieren Sie in der Liste der ActiveX Controls *NETRONIC VARCHART XTree* und entfernen Sie die Registrierung durch Klick auf die Schaltfläche **Entfernen**. Verlassen Sie dann den Dialog mit **Abbrechen**.
- Öffnen Sie nun den Dialog **Komponente > Packages installieren** und wählen Sie das Package *Borland Anwenderkomponenten*. (Dieses Package ist in der Datei *dclusr*0.bpl* gespeichert, wobei das '*' im Dateinamen abhängig von Ihrer Delphi-Version ist: 5, 6 oder 7.)
- Klicken Sie auf **Bearbeiten**. Damit wird die Datei *dclusrX0.dpk* geöffnet.
- Selektieren Sie nun nacheinander die Dateien *VcTreeLib_TLB.pas* und *VcTreeLib_TLB.dcr* und entfernen Sie sie jeweils durch einen Rechtsklick aus dem Projekt.
- Kompilieren Sie nun das Package und schließen Sie danach den Dialog. Dadurch werden die Änderungen in dem Projekt *dclusrX0* gespeichert.
- Öffnen Sie nun wieder den Dialog **Komponente > ActiveX importieren**.
- Klicken Sie auf **Hinzufügen**, wählen Sie *vctree.ocx* aus und klicken Sie auf **Öffnen**. Jetzt sehen Sie *NETRONIC VARCHART XTree* wieder in der Liste der registrierten ActiveX Controls.
- Klicken Sie auf **Installieren...**. Damit wird das Package *dclusrX0.bpl* neu kompiliert.
- Schließen Sie den Dialog. Damit wird das Projekt *dclusrX0* gespeichert.

6.4 Wieso können Knoten u. U. nicht interaktiv erzeugt werden?

Wenn Sie zur Laufzeit keine Knoten mit der Maus anlegen können, stellen Sie sicher, dass auf der Eigenschaftenseite **Allgemeines** das Kontrollkästchen **Neue Knoten zulassen** aktiviert ist.

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

6.5 Wie verhindert man das interaktive Erzeugen von Knoten?

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

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

Code-Beispiel

```
Sub Form_Load
   VcTree1.AllowNewNodes = False
End Sub
```

6.6 Wie lassen sich die Standard-Kontextmenüs abschalten?

Die vordefinierten Kontextmenüs können Sie durch Setzen des Rückgabestatus auf **vcRetStatNoPopup** unterdrücken.

Code-Beispiel

```
' Kontextmenü für das Diagramm abschalten
Private Sub VcTree1_OnDiagramRClick(ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)
    returnStatus = vcRetStatNoPopup
End Sub

' Kontextmenü für Knoten abschalten
Private Sub VcTree1_OnNodeRClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As
VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)
   returnStatus = vcRetStatNoPopup
End Sub
```

6.7 Was ist bei Problemen mit dem Drucken zu tun?

Wenn Sie Ihr Diagramm nicht drucken können oder den Drucker nicht einrichten können, prüfen Sie bitte, ob die Datei *vcprct32.dll* vorhanden ist. Prüfen Sie außerdem, ob sie in dem im PATH angegebenen Verzeichnis enthalten ist, und ob der Standard-Windows-Drucker eingerichtet ist.

Sollte die Datei *vcprct32.dll* fehlen, wenden Sie sich bitte an den Kundendienst der NETRONIC Software GmbH.

6.8 Wie lässt sich die Performance verbessern?

> SuspendUpdate

Bei größeren Datenmengen kann es unter Umständen sehr lange dauern, wenn bei einer großen Anzahl von Vorgängen dieselbe Aktion durchgeführt wird. Nicht jeder automatische Aktualisierungsvorgang im Diagramm ist notwendig; in einem solchen Fall können Sie Aktualisierungen für den Einzelfall unterdrücken und nach der Abarbeitung einer Code-Sequenz final einmal durchführen. Unterdrückung und Wiedereinsatz der Aktualisierung erfolgen mit der Methode **SuspendUpdate**, die zu Beginn der Code-Sequenz auf **True** bzw. am Ende auf **False** gesetzt wird. Auf diese Weise können Sie die Performance insgesamt beträchtlich erhöhen.

Code-Beispiel

```
VcTree1.SuspendUpdate (True)

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

VcTree1.SuspendUpdate (False)
```

> Grafiken

Eine Ursache für eine zu geringe Performance können Grafiken beispielsweise in Tabellen-, Knoten- oder Boxfeldern sein, die zu groß sind oder eine zu große Pixelzahl haben.

6.9 Fehlermeldungen

> Fehlermeldungen zur Laufzeit, vom Entwickler verursacht

Fehlerursache	Meldung
Lizenzierung fehlgeschlagen	Sie nutzen eine unlizenzierte Version von *. Bitte wenden Sie sich an NETRONIC, um eine vollständig lauffähige Version zu erhalten.
	Die Lizenzierung meldet einen Fehler. Bitte setzen Sie sich mit NETRONIC in Verbindung.
	Die Nutzungsberechtigung ist abgelaufen. Bitte setzen Sie sich mit NETRONIC in Verbindung.
	Ihre Identifizierung hat sich von * in * geändert, bitte wenden Sie sich an NETRONIC!
	Das in diesem Programm benutzte ActiveX-Steuerelement hat keine Runtime-Lizenz!
ActiveX-Installation unvollständig oder ältere DLL-Versionen im Systempfad	Die DLL * wurde nicht gefunden
	Das Interface (*) konnte nicht geladen werden
	Die Interface- DLL (Version **) ist zu alt. Dieses Programm benötigt Version * oder jünger.
Programminstallation unvollständig oder absoluter Pfad fehlerhaft	Die Gruppentiteldatei konnte nicht gefunden werden.
	Die Datei * ist keine gültige Grafikdatei.
	Grafikfile ist nicht angegeben oder existiert nicht.
Fehler bei der Zuweisung einer neuen INI-Datei	Konfigurationsfile * nicht gefunden, Programm erzeugt es mit Standardkonfiguration.
INI-Datei hat Fehler	Die Knotenkennzeichnung/Die Tabelle/Der Layer * verlangt den nicht existierenden Filter *. Der Filtereintrag wird auf <immer> korrigiert.
	Die Knotenkennzeichnung/Die Tabelle * verlangt die nicht existierende Knotenbeschriftung *. Der Knotenbeschriftungseintrag wird auf <nicht festgelegt> korrigiert.
	Der Name * für Layer ist doppelt vorhanden. Bitte Konfigurationsdatei prüfen.
	Highlight * existiert nicht
	Der Name * für Verbindungsaussehen ist doppelt vorhanden. Bitte Konfigurationsdatei(en) prüfen.
	Your configuration file * is corrupt. [*] must be unique.

.

7 API Referenz

7.1 Objekttypen

- DataObject
- DataObjectFiles
- VcBorderArea
- VcBorderBox
- VcBox
- VcBoxCollection
- VcBoxFormat
- VcBoxFormatCollection
- VcBoxFormatField
- VcDataDefinition
- VcDataDefinition
- VcDataDefinitionTable
- VcDataRecord
- VcDataRecordCollection
- VcDataTable
- VcDataTableCollection
- VcDataTableField
- VcDataTableFieldCollection
- VcDefinitionField
- VcFilter
- VcFilterCollection
- VcFilterSubCondition
- VcLegendView
- VcMap
- VcMapCollection
- VcMapEntry
- VcNode
- VcNodeAppearance
- VcNodeAppearanceCollection
- VcNodeCollection
- VcNodeFormat
- VcNodeFormatCollection
- VcNodeFormatField

- VcPrinter
- VcRect
- VcTree
- VcWorldView

7.2 DataObject

DataObject

Durch die OLE-Drag&Drop-Technik können Sie ausgewählte Knoten in ein anderes VARCHART-ActiveX-Steuerelement ziehen und dort ablegen. Der Transportspeicher für diese Daten ist das DataObject-Objekt. Zu diesem Zweck stellt das Objekt die passenden Methoden und Eigenschaften bereit: **Files**, **Clear**, **GetData**, **GetFormat** und **SetData**.

Sie können auch Daten mit anderen OLE-Drag&Drop-fähigen Steuerelementen und Anwendungen austauschen. Bedenken Sie bitte dabei, dass die VARCHART-ActiveX-Steuerelemente die Daten immer im CSV-Textformat ablegen und interpretieren.

Damit OLE Drag & Drop einsetzbar wird, müssen im Eigenschaften-Fenster die Eigenschaften **OLEDragMode** und **OLEDropMode** aktiviert werden. Auf der Eigenschaftsseite **Knoten** bestimmt die Option **Move all selected nodes**, ob nur ein einzelner oder gleichzeitig mehrere Knoten verschoben werden können.

Nähere Erläuterungen zu diesem Thema finden Sie im Kapitel **Wichtige Begriffe** unter der dem Thema **OLE-Drag&Drop**.

Eigenschaften

- DropInsertionPosition
- Files

Methoden

- Clear
- GetData
- GetFormat
- SetData

Eigenschaften

DropInsertionPosition

Nur-Lese-Eigenschaft von DataObject

Mit dieser Eigenschaft können Sie die aktuelle Anlagerungsposition bei OLE Drag & Drop im Ereignis **OLEDragOver** bzw. **OLEDragDrop** erfragen.

Mit Hilfe dieser Eigenschaft können Sie die Einfügeoperation manuell durchführen. Wenn gleichzeitig mit **IdentifyObjectAt** der Referenzknoten ermittelt wird, kann mittels **InsertNodeRecordEx** ein Knoten an der angegebenen Anlagerungsposition ins Chart eingefügt werden.

	Datentyp	Beschreibung
Eigenschaftswert	InsertionPositionEnum **Mögliche Werte:** vcIPFirstChild 3 vcIPLastChild 4 vcIPLeftBrother 31 vcIPNone 0 vcIPNormal 1 vcIPParent 6 vcIPRightBrother 32	Einfügeposition als ersten Sohnknoten des Referenzknotens einfügen als letzten Sohnknoten des Referenzknotens einfügen als linken Bruderknoten des Referenzknotens einfügen nicht zulässige Einfügeposition (gilt nur für DataObject.DropInsertionPosition) ohne Referenzknoten einfügen als Vaterknoten des Referenzknotens einfügen als rechten Bruderknoten des Referenzknotens einfügen

Files

Nur-Lese-Eigenschaft von DataObject

Diese Eigenschaft gibt eine DataObjectFiles-Auflistung zurück, die wiederum eine Liste aller Dateinamen enthält, die von einem DataObject-Objekt verwendet werden (wie beispielsweise Namen von Dateien, die ein Benutzer in oder aus dem Windows-Explorer zieht). Diese Eigenschaft kann nur benutzt werden, wenn das DataObject Daten im Format **15** (**Dateiliste**, s. Eigenschaft **GetFormat**) enthält.

	Datentyp	Beschreibung
Eigenschaftswert	DataObjectFiles	Liste der verfügbaren Dateien

Methoden

Clear

Methode von DataObject

Mit dieser Methode können Sie den Inhalt des DataObject-Objekts löschen. Diese Methode steht nur für Drag-Operationen zur Verfügung, das heißt, für

die Ereignisse **OLEStartDrag**, **OLESetData**, **OLEGiveFeedback** und **OLECompleteDrag**.

	Datentyp	Beschreibung
Rückgabewert	Void	

GetData

Methode von DataObject

Diese Methode gibt Daten vom Typ **Variant** aus einem DataObject zurück und steht nur für DataObject-Objekte der Ereignisse **OLEDragOver** und **OLEDragDrop** zur Verfügung.

Die Methode **GetData** kann auch andere als die unten aufgeführten Datenformate verwenden, wozu unter anderem auch benutzerdefinierte Formate gehören, die in Windows über die API-Funktion **RegisterClipboardFormat()** registriert sind. Hierbei sind jedoch einige Einschränkungen zu berücksichtigen:

Die **GetData**-Methode gibt Daten immer in Form eines Byte-Datenfelds zurück, wenn sie ein Format haben, das die Methode nicht erkennt.

Das von der **GetData**-Methode zurückgegebene Byte-Datenfeld ist ggf. größer als die tatsächlichen Daten, wobei sich am Ende des Datenfelds beliebige Bytes befinden. Die Ursache dafür liegt darin, dass VARCHART ActiveX das Format der Daten nicht kennt und lediglich weiß, wieviel Speicher das Betriebssystem für die Daten reserviert hat. Dieser reservierte Speicher ist häufig größer als der tatsächlich für die Daten erforderliche Speicher. Aus diesem Grund können sich am Ende des reservierten Speichersegments möglicherweise überflüssige Bytes befinden. Dies führt dazu, dass Sie die richtigen Funktionen für eine sinnvolle Interpretation der zurückgegebenen Daten verwenden müssen (z. B. in Visual Basic das Abschneiden einer Zeichenfolge an einer bestimmten Stelle mit der **Left**-Funktion, falls die Daten im Textformat vorliegen).

Hinweis: Nicht alle Anwendungen unterstützen die Formate **2** (Bitmap) oder **9** (Farbpalette), daher sollten Sie möglichst immer **8** (Geräteunabhängige Bitmap) verwenden.

	Datentyp	Beschreibung
Parameter:		
⇨ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vbCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF
Rückgabewert	Variant	geholte Daten

GetFormat

Methode von DataObject

Diese Methode gibt einen booleschen Wert zurück, der anzeigt, ob im DataObject Daten eines bestimmten Formats vorhanden sind. Sie steht nur für DataObject-Objekte der Ereignisse **OLEDragOver** und **OLEDragDrop** zur Verfügung.

	Datentyp	Beschreibung
Parameter:		
⇨ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vcCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF
Rückgabewert	Boolean	Die Methode **GetFormat** liefert **True** zurück, wenn ein Element des DataObject-Objektes dem spezifizierten Format entspricht, andernfalls wird **False** zurückgegeben.

SetData

Methode von DataObject

Diese Methode fügt Daten im angegebenen Datenformat zum DataObject hinzu und steht nur für DataObject-Objekte der Ereignisse **OLEStartDrag**, **OLESetData**, **OLEGiveFeedback** und **OLECompleteDrag** zur Verfügung.

Die Methode **SetData** kann auch andere als die unter **Format** aufgeführten Datenformate verwenden, wozu unter anderem auch benutzerdefinierte Formate gehören, die in Windows über die API-Funktion **RegisterClip-**

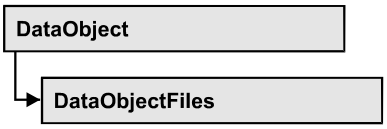
boardFormat() registriert sind. Hierbei sind jedoch einige Einschränkungen zu berücksichtigen:

Für die **SetData**-Methode müssen die Daten in Form eines Byte-Datenfelds vorliegen, wenn das angegebene Datenformat nicht erkannt wird.

Nicht alle Anwendungen unterstützen **2** (Bitmap) oder **9** (Farbpalette), daher sollten Sie möglichst immer **8** (geräteunabhängige Bitmap) verwenden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ data	Variant	Zu setzende Daten oder **Empty**, falls bekannt gegeben werden soll, dass das Format über das Ereignis **OLESetData** bei Bedarf gesetzt werden kann.
⇨ format	Integer	Identifikationsnummer des Formats (mit Beispielen aus Visual Basic und C): 1 - Text in ANSI-Codierung (.txt-Dateien) VB: vcCFText; C: CF_TEXT 2 - Bitmap (.bmp-Dateien) VB: vbCFBitmap; C: CF_BITMAP 3 - Metafile (.wmf-Dateien) VB: vbCFMetaFile; C: CF_METAFILE 8 - Geräteunabhängige Bitmap (DIB) VB: vbCFDIB; C: CF_DIB 9 - Farbpalette VB: vbCFPalette; C: CF_PALETTE 13 - Text in Unicode-Codierung (.txt-Dateien) VB: 13; C: CF_UNICODETEXT 14 - Enhanced Metafile (.emf-Dateien) VB: vbCFEMetaFile; C: CF_METAFILE 15 - Dateiliste VB: vbCFFiles; C: CF_FILES -16639 - Rich text format (.rtf-Dateien) VB: vbCFRTF; C: CF_RTF
Rückgabewert	Void	

7.3 DataObjectFiles



Dieses Objekt verwaltet eine Liste aller Dateinamen, die in einem DataObject gespeichert sind, wenn dieses Daten im Format **15** (Dateiliste) enthält. Über **For Each Item in DataObjectFiles** können Sie in einer Schleife auf alle Dateinamen zugreifen.

Eigenschaften

- _NewEnum
- Count
- Item

Methoden

- Add
- Clear
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von DataObjectFiles

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datenobjekt-Dateien iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Private Sub VcTree1_OLEDragOver(ByVal data As VcTreeLib.DataObject, effect As
Long, ByVal button As Integer, ByVal Shift As Integer, ByVal x As Long, ByVal y
As Long, ByVal state As VcTreeLib.OLEDragStateEnum)

Dim fileName as String
For Each fileName In DataObject.DataObjectFiles
   Debug.Print fileName
Next

End Sub
```

Count

Nur-Lese-Eigenschaft von DataObjectFiles

Diese Eigenschaft gibt die Anzahl der verwalteten Dateinamen zurück.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Dateinamen

Item

Eigenschaft von DataObjectFiles

Diese Eigenschaft setzt oder erfragt einen der verwalteten Dateinamen über den angegebenen Index. Da dies die definierte Standardeigenschaft des Objekts ist, kann in vielen Entwicklungsumgebungen (z. B. Visual Basic) der Eigenschaftenname weggelassen werden. Beispiel: DataObjectFiles(0) erfragt den ersten Dateinamen.

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

Methoden

Add

Methode von DataObjectFiles

Mit dieser Methode können Sie den angegebenen Dateinamen zur Liste der Dateinamen hinzufügen. Ist ein Index (Integer, Wertebereich 0 bis .Count-1) angegeben, dann wird der Dateiname an der entsprechenden Stelle eingefügt. Andernfalls wird er am Ende der Liste angefügt.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Variant	Index der Position in der Liste, an der der Dateiname eingefügt werden soll (optional)
⇨ fileName	String	Name der Datei
Rückgabewert	Void	

Clear

Methode von DataObjectFiles

Mit dieser Methode können Sie alle Dateinamen der Liste löschen.

	Datentyp	Beschreibung
Rückgabewert	Void	

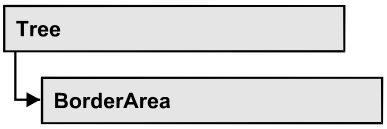
Remove

Methode von DataObjectFiles

Mit dieser Methode können Sie den Dateinamen an dem angegebenem Index (Wertebereich 0 bis .Count-1) entfernen.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Long	Index der Position in der Liste, von der der Dateiname gelöscht werden soll.
Rückgabewert	Void	

7.4 VcBorderArea



Ein Objekt vom Typ **VcBorderArea** bezeichnet den Titel- bzw. Legendenbereich der Grafik.

Methoden

- BorderBox

Methoden

BorderBox

Methode von VcBorderArea

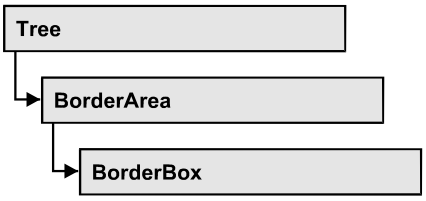
Diese Methode ermöglicht den Zugriff auf ein BorderBox-Objekt.

	Datentyp	**Beschreibung**
Parameter:		
boxPosition	BorderBoxPositionEnum	Position der Box
	Mögliche Werte:	
	vcBBXPBottomBottomCentered 8	zweite Zeile im unteren Bereich, mittig
	vcBBXPBottomBottomLeft 7	zweite Zeile im unteren Bereich, links
	vcBBXPBottomBottomRight 9	zweite Zeile im unteren Bereich, rechts
	vcBBXPBottomTopCentered 5	erste Zeile im unteren Bereich, mittig
	vcBBXPBottomTopLeft 4	erste Zeile im unteren Bereich, links
	vcBBXPBottomTopRight 6	erste Zeile im unteren Bereich, rechts
	vcBBXPLegend 51	Legende
	vcBBXPTopCentered 2	oben mittig
	vcBBXPTopLeft 1	oben links
	vcBBXPTopRight 3	oben rechts
Rückgabewert	VcBorderBox	Box des Titel- und Legendenbereichs

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
```

7.5 VcBorderBox



Ein Objekt vom Typ **VcBorderBox** bezeichnet eine der Boxen des Titel- bzw. Legendenbereichs der Grafik.

Eigenschaften

- Alignment
- GraphicsFileName
- LegendElementsArrangement
- LegendElementsBottomMargin
- LegendElementsMaximumColumnCount
- LegendElementsMaximumRowCount
- LegendElementsTopMargin
- LegendFont
- LegendTitle
- LegendTitleFont
- LegendTitleVisible
- Text
- TextFont
- Type

Eigenschaften

Alignment

Eigenschaft von VcBorderBox

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

	Datentyp	Beschreibung
Eigenschaftswert	BorderBoxAlignmentEnum	Ausrichtung der BorderBox
	Mögliche Werte:	
	vcBBXACentered -1	unten mittig
	vcBBXALeft -3	links

vcBBXARight -2	rechts

GraphicsFileName

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Namen der Grafikdatei angeben oder erfragen, die in dem aktuellen VcBorderBox-Objekt verwendet wird. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile)
- *.WMF mit eingebautem EMF

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxTR As VcBorderBox
```

```
Set borderArea = VcTree1.BorderArea
Set bBoxTR = borderArea.BorderBox(vcBBXPTopRight)
bBoxTR.Type = vcBBXTGraphics
bBoxTR.GraphicsFilename = "Asterix.jpg"
```

LegendElementsArrangement

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Anordnung der Elemente der Legende setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	LegendElementsArrangementEnum	Typ der Anordnung der Legendenelemente
	Mögliche Werte:	
	vcLEAFixedToColumns 1	Die Legendenelemente werden nur in Spalten ausgerichtet.
	vcLEAFixedToRows 0	Die Legendenelemente werden nur in Zeilen ausgerichtet.
	vcLEAFixedToRowsAndColumns 2	Die Legendenelemente werden in Zeilen und Spalten ausgerichtet.

LegendElementsBottomMargin

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Abstand der Elemente zum unteren Rand festlegen oder erfragen (Einheit: mm).

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des unteren Randes

LegendElementsMaximumColumnCount

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Anzahl der Spalten setzen oder erfragen, über die sich die Elemente der Legende verteilen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Spalten

LegendElementsMaximumRowCount

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Anzahl der Zeilen setzen oder erfragen, über die sich die Elemente der Legende verteilen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Zeilen

LegendElementsTopMargin

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Abstand der Elemente zum oberen Rand festlegen oder erfragen (Einheit: mm).

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des oberen Randes

LegendFont

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Schriftattribute der Legende angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute der Legende

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTLegend
logThis (bBoxBBL.LegendFont.Name)
```

LegendTitle

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Legendentitel angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Legendentitel

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitle = "Explanation"
```

LegendTitleFont

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Schriftattribute des Legendentitels angeben oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftattribute des Legendentitels

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTLegend
logThis (bBoxBBL.LegendTitleFont.Name)
```

LegendTitleVisible

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft legen Sie fest, ob der Legendentitel sichtbar ist oder nicht.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Legendentitel sichtbar (True)/nicht sichtbar (False)

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.LegendTitleVisible = False
```

Text

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie den Text einer Titelzeile (oberhalb oder unterhalb des Diagramms) angeben oder erfragen. Für die Seitennummerierung oder die Ausgabe des Systemdatums können Sie folgende Platzhalter angeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{COLUMN} = Seitennummer in der Breite (einer zweidimensionalen Seitenanordnung)

{NUMPAGES} = Gesamtanzahl der Seiten

{PAGE} = fortlaufende Seitennummer

{ROW} = Seitennummer in der Höhe (einer zweidimensionalen Seitenanordnung)

{SYSTEMDATE} = Systemdatum

	Datentyp	**Beschreibung**
Parameter:		
rowIndex	Integer	Zeilenindex {0...6}
Eigenschaftswert	String	Text in Textfeldern

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTText
bBoxBBL.Text(index) = "Department A"
```

TextFont

Eigenschaft von VcBorderBox

Mit dieser Eigenschaft können Sie die Schriftattribute einer Titelzeile (oberhalb oder unterhalb des Diagramms) angeben oder erfragen.

Dies ist eine indizierte Eigenschaft, die in C# über die beiden Methoden **set_TextFont (rowIndex, pvn)** und **get_TextFont (row-Index)** angesprochen wird.

	Datentyp	Beschreibung
Parameter:		
rowIndex	Integer	Zeilenindex {0...6}
Eigenschaftswert	StdFont	Schriftattribute des Textes

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxTL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)

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

Code Sample in C#

```
/ Text for Title
VcBorderBox borderBox =
VcTree1.BorderArea.BorderBox(VcBorderBoxPosition.vcBBXPTopCentered);
borderBox.Type = VcBorderBoxType.vcBBXTText;

Font titleFont1 = new Font("Arial", 20, FontStyle.Bold);

borderBox.set_Text(1, "Time Scheduler");
borderBox.set_TextFont(1, titleFont1);
```

Type

Eigenschaft von VcBorderBox

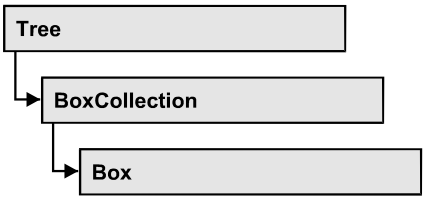
Mit dieser Eigenschaft können Sie den Typ des BorderBox-Objekts angeben oder erfragen.

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

Code-Beispiel

```
Dim borderArea As VcBorderArea
Dim bBoxBBL As VcBorderBox

Set borderArea = VcTree1.BorderArea
Set bBoxBBL = borderArea.BorderBox(vcBBXPBottomBottomLeft)
bBoxBBL.Type = vcBBXTGraphics
```

7.6 VcBox



Ein Objekt vom Typ **VcBox** beschreibt eine Box, in der Texte und Grafiken ausgegeben werden können.

Eigenschaften

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

Methoden

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

Eigenschaften

FieldText

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Inhalt eines Feldes einer Box erfragen oder festlegen. Diese Eigenschaft können Sie auch im Dialog **Box bearbeiten** festlegen.

Bei mehrzeiligen Textfeldern müssen für einen erzwungenen Umbruch die einzelnen Zeilen des Textfelds mit "\n" im String getrennt sein (Beispiel: "Zeile1\nZeile2"). Ohne erzwungenen Umbruch wird automatisch an Leerzeichen umgebrochen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ fieldIndex	Integer	Feldindex
Eigenschaftswert	String	Feldinhalt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.boxCollection
Set box = boxCltn.FirstBox
box.FieldText(0) = "User: "
```

FormatName

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen des Boxformats erfragen oder festlegen.

	Datentyp	**Beschreibung**
Eigenschaftswert	VcBoxFormat	BoxFormat-Objekt oder **Nothing**

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

boxCltn = VcTree1.BoxCollection
box = boxCltn.FirstBox
box.FormatName = "Standard"
```

LineColor

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Farbe der Randlinie der Box festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineColor = RGB(255, 0, 0)
```

LineThickness

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Stärke der Randlinie der Box erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm **Standardwert:** Wie im Dialog definiert

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineThickness = 2
```

LineType

Eigenschaft von VcBox

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

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp **Standardwert:** vcSolid
	Mögliche Werte:	
	vcDashed 4	Linientyp **gestrichelt**
	vcDashedDotted 5	Linientyp **gestrichelt-gepunktet**
	vcDotted 3	Linientyp **gepunktet**
	vcLineType0 100	Linientyp 0
	vcLineType1 101	Linientyp 1
	vcLineType10 110	Linientyp 10
	vcLineType11 111	Linientyp 11
	vcLineType12 112	Linientyp 12
	vcLineType13 113	Linientyp 13
	vcLineType14 114	Linientyp 14
	vcLineType15 115	Linientyp 15
	vcLineType16 116	Linientyp 16
	vcLineType17 117	Linientyp 17
	vcLineType18 118	Linientyp 18
	vcLineType2 102	Linientyp 2
	vcLineType3 103	Linientyp 3

vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp **durchgezogen**

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.LineType = vcDotted
```

MarkBox

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob eine Box markiert ist.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	**True**: Box markiert; **false**: Box nicht markiert

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.MarkBox = True
```

Moveable

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Box interaktiv verschiebbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	verschiebbar (True)/ nicht verschiebbar (False) **Standardwert:** True

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Moveable = False
```

Name

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen einer Box erfragen oder setzen. Diese Eigenschaft können Sie im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Box

Code-Beispiel

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

Set boxCltn = VcTree1.boxCollection
Set box = boxCltn.FirstBox
boxName = box.Name
MsgBox boxName
```

Origin

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Ursprungspunkt der Box festlegen oder erfragen, d. h. den Diagrammpunkt, von dem aus der Abstand zum Referenzpunkt der Box in x- bzw. y-Richtung angegeben wird.

Mithilfe der Eigenschaften **Origin** und **ReferencePoint** sowie der Methode **GetXYOffset** können Sie jede einzelne Box im Diagrammbereich positionieren, wobei die relative Position der Box zum Diagramm unabhängig von der Diagrammgröße ist.

	Datentyp	Beschreibung
Eigenschaftswert	BoxOriginEnum **Mögliche Werte:**	Ursprungspunkt der Box

vcBOBottomCenter 28	unten mittig
vcBOBottomLeft 27	unten links
vcBOBottomRight 29	unten rechts
vcBOCenterCenter 25	mittig mittig
vcBOCenterLeft 24	mittig links
vcBOCenterRight 26	mittig rechts
vcBOTopCenter 22	oben mittig
vcBOTopLeft 21	oben links
vcBOTopRight 23	oben rechts

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Origin = vcBOTopCenter
```

Priority

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Priorität der Box erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Prioritätsstufe

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.Priority = 3
```

ReferencePoint

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Referenzpunkt der Box festlegen oder erfragen, d. h. den Punkt der Box, von dem aus der Abstand zum Ursprung in x- bzw. y-Richtung angegeben wird.

	Datentyp	Beschreibung
Eigenschaftswert	BoxReferencePointEnum	Referenzpunkt der Box
	Mögliche Werte:	
	vcBRPBottomCenter 28	unten mittig
	vcBRPBottomLeft 27	unten links
	vcBRPBottomRight 29	unten rechts
	vcBRPCenterCenter 25	mittig mittig

vcBRPCenterLeft 24	mittig links
vcBRPCenterRight 26	mittig rechts
vcBRPTopCenter 22	oben mittig
vcBRPTopLeft 21	oben links
vcBRPTopRight 23	oben rechts

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
box.ReferencePoint = vcBRPCenterRight
```

Specification

Nur-Lese-Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie die Spezifikation dieser Box auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung einer Box mit der Methode **VcBox-Collection.AddBySpecification** benutzt werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Boxspezifikation

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(0)
MsgBox box.Specification
```

UpdateBehaviorName

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Name des Aktualisierungsverhaltens

Visible

Eigenschaft von VcBox

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob die Box sichtbar sein soll. Diese Eigenschaft können Sie auch im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Box sichtbar/unsichtbar **Standardwert:** True

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.FirstBox
box.Visible = False
```

Methoden

GetActualExtent

Methode von VcBox

Mit dieser Methode können Sie die Breite und Höhe der Box erfragen (Einheit: 1/100 mm).

Werden diese Werte beim XY-Offset berücksichtigt, kann man z.B. den Referenzpunkt der Verankerungslinie ändern, ohne die Position der Box zu verändern.

	Datentyp	Beschreibung
Parameter:		
⇦ width	Integer	Breite der Box
⇦ height	Integer	Höhe der Box
Rückgabewert	Boolean	Ausdehnung der Box wird zurückgegeben/wird nicht zurückgegeben

GetTopLeftPixel

Methode von VcBox

Mit dieser Methode können Sie den gespeicherten XY-Offset für die obere linke Ecke in Pixel umrechnen und ausgeben.

Der x-Wert kann dann z.B. mit der Methode **VcGantt.GetDate** weiter verwendet werden, um ein Datum zu erhalten

	Datentyp	Beschreibung
Parameter:		
⇦ x	Integer	x-Wert des Offsets
⇦ y	Integer	y-Wert des Offsets
Rückgabewert	Boolean	Offset wird zurückgegeben/nicht zurückgegeben

GetXYOffset

Methode von VcBox

Mit dieser Methode können Sie den Abstand zwischen Ursprung und Referenzpunkt in x- und y-Richtung erfragen (Einheit: 1/100 mm).

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **GetXYOffsetAsVariant** benutzen.

	Datentyp	Beschreibung
Parameter:		
⇦ xOffset	Integer	x-Wert des Offsets
⇦ yOffset	Integer	y-Wert des Offsets
Rückgabewert	Boolean	Offset wird zurückgegeben/nicht zurückgegeben

GetXYOffsetAsVariant

Methode von VcBox

Diese Methode ist bis auf die Parameter identisch mit der Methode **GetXYOffset**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

IdentifyFormatField

Methode von VcBox

Mit dieser Methode können Sie den Index des an der bezeichneten Position befindlichen Formatfeldes erfragen. Falls sich an der bezeichneten Position ein Feld befindet, wird **True** zurückgegeben, ansonsten **False**.

	Datentyp	Beschreibung
Parameter:		
⇨ x	Long	X-Koordinate der Position
⇨ y	Long	Y-Koordinate der Position
⇦ format	VcBoxFormat	Identifiziertes Format
⇦ formatFieldIndex	Integer	Format-Feldindex
Rückgabewert	Boolean	Ein Formatfeld befindet sich/befindet sich nicht an der angegebenen Position

SetXYOffset

Methode von VcBox

Mit dieser Methode können Sie den Abstand zwischen Ursprung und Referenzpunkt in x- und y-Richtung festlegen (Einheit: 1/100 mm).

Den Offset können Sie auch im Dialog **Boxen verwalten** festlegen.

	Datentyp	Beschreibung
Parameter:		
⇨ xOffset	Integer	x-Wert des Offsets
⇨ yOffset	Integer	y-Wert des Offsets
Rückgabewert	Boolean	Offset wird gesetzt (True)/nicht gesetzt (False)

Code-Beispiel

```
Dim OffsetSet As Boolean
OffsetSet = VcTree1.boxCollection.FirstBox.SetXYOffset(100, 100)
```

SetXYOffsetByTopLeftPixel

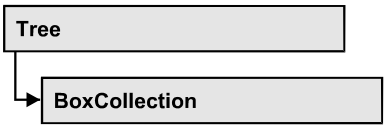
Methode von VcBox

Mit dieser Methode können Sie den angegebenen Pixelwert der oberen linken Ecke intern in einen XY-Offset umrechnen und speichern.

Damit kann man z.B. an einer XY-Koordinate aus einem Ereignis eine Box positionieren.

	Datentyp	**Beschreibung**
Parameter:		
⇨ x	Integer	x-Wert des Offsets
⇨ y	Integer	y-Wert des Offsets
Rückgabewert	Boolean	Offset wird gesetzt (True)/nicht gesetzt (False)

7.7 VcBoxCollection



In einem Objekt des Typs VcBoxCollection sind alle verfügbaren Boxen zusammengefasst. Über **For Each box In BoxCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Boxen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **BoxByName** und **BoxByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Boxen kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Boxen.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- BoxByIndex
- BoxByName
- Copy
- FirstBox
- NextBox
- Remove
- Update

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcBoxCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Box-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim box As VcBox

For Each box In VcTree1.BoxCollection
   Debug.Print box.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcBoxCollection

Mit dieser Eigenschaft können Sie die Anzahl der Boxobjekte in der Box-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Boxen

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim numberOfBoxes As Long

Set boxCltnn = VcTree1.BoxCollection
Dim numberOfBoxes = boxCltn.Count
```

Methoden

Add

Methode von VcBoxCollection

Mit dieser Methode können Sie eine neue Box in der BoxCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Boxobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die neu angelegte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter:		
⇨ boxName	String	Name der Box
Rückgabewert	VcBox	Neues Boxobjekt

Code-Beispiel

```
Set newBox = VcTree1.BoxCollection.Add("box1")
```

AddBySpecification

Methode von VcBoxCollection

Mit dieser Methode können Sie eine Box über eine Box-Spezifikation erzeugen. Dies dient der Persistenz von Box-Objekten. Die Spezifikation einer Box kann erfragt (siehe VcBox-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Box mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden. Um die neu angelegte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter:		
⇨ Specification	String	Boxspezifikation
Rückgabewert	VcBox	Neues Boxobjekt

BoxByIndex

Methode von VcBoxCollection

Mit dieser Methode können Sie auf eine einzelne Box über ihren Index zugreifen. Existiert keine Box unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index der Box
Rückgabewert	VcBox	Ermitteltes Boxobjekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection

Set boxCltn = VcTree1.BoxCollection
```

```
Set box = boxCltn.BoxByIndex(2)
box.LineThickness = 2
```

BoxByName

Methode von VcBoxCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf eine bestimmte Box zugreifen. Existiert keine Box unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ boxName	String	Name der Box
Rückgabewert	VcBox	Box

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByName("Box 1")
```

Copy

Methode von VcBoxCollection

Mit dieser Methode können Sie eine Box kopieren. Wenn die Box mit dem angegebenen Namen existiert und der Name der neuen Box noch nicht verwendet wird, wird das neue Boxobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Um die kopierte Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter:		
⇨ boxName	String	Name der zu kopierenden Box
⇨ newBoxName	String	Name der neuen Box
Rückgabewert	VcBox	Boxobjekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.Copy("BoxOne", "NewBox")
boxCltn.Update
```

FirstBox

Methode von VcBoxCollection

Mit dieser Methode können Sie auf die erste Box der Box-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextBox** über die nachfolgenden Boxen zu iterieren. Existiert keine Box in der Box-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBox	erste Box

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.FirstBox
```

NextBox

Methode von VcBoxCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Boxen der Box-Auflistung zugreifen, nachdem Sie mit der Methode **FirstBox** den Initialwert erfasst haben. Sind alle Boxen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBox	Nachfolgendes Box-Objekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.FirstBox

While Not box Is Nothing
    Listbox.AddItem box.Name
    Set box = boxCltn.NextBox
Wend
```

Remove

Methode von VcBoxCollection

Mit dieser Methode können Sie eine Box löschen. Um das Löschen der Box im Diagramm sichtbar werden zu lassen, muss die Auflistung (Collection) mit **Update** aktualisiert werden.

	Datentyp	Beschreibung
Parameter:		
⇨ boxName	String	Name der Box
Rückgabewert	Boolean	Box gelöscht (True)/nicht gelöscht (False)

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(2)
boxCltn.Remove (box.Name)
boxCltn.Update
```

Update

Methode von VcBoxCollection

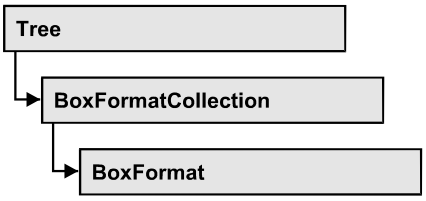
Mit dieser Methode können Sie eine BoxCollection aktualisieren, nachdem Sie sie verändert haben.

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

Code-Beispiel

```
Dim boxCltn As VcBoxCollection
Dim box As VcBox

Set boxCltn = VcTree1.BoxCollection
Set box = boxCltn.BoxByIndex(2)
boxCltn.Remove (box.Name)
boxCltn.Update
```

7.8 VcBoxFormat



Ein Objekt vom Typ **VcBoxFormat** beschreibt die Formate von Boxen.

Eigenschaften

- _NewEnum
- FieldsSeparatedByLines
- FormatField
- FormatFieldCount
- Name
- Specification

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcBoxFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Boxformatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim formatField As VcBoxFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

FieldsSeparatedByLines

Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie festlegen, ob die Felder durch sichtbare Linien getrennt werden (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Boxfelder werden durch Linien getrennt (True)/ nicht getrennt (False).

Code-Beispiel

```
Dim boxFormat As VcBoxFormat

Set boxFormat = VcTree1.BoxFormatCollection.FormatByIndex(2)
boxFormat.FieldsSeparatedByLines = True
```

FormatField

Nur-Lese-Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie ein VcBoxFormatField-Objekt per Index holen. Der Index muss im Bereich von 0 bis .FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht (wie in den Feldeigenschaften vorher) von 1 bis .FormatFieldCount!

	Datentyp	Beschreibung
Parameter:		
index	Integer	Index des Boxformatfeldes 0 ... .FormatFieldCount-1
Eigenschaftswert	VcBoxFormatField	Boxformatfeld

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcTree1.BoxFormatCollection.FirstFormat
Set formatField = boxFormat.formatField(0)
MsgBox formatField.FormatName
```

FormatFieldCount

Nur-Lese-Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie die Anzahl der Felder eines Boxformats ermitteln.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Felder im Boxformat

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcTree1.BoxFormatCollection.FirstFormat
MsgBox boxFormat.FormatFieldCount
```

Name

Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie den Namen eines Boxformats erfragen oder setzen. Diese Eigenschaft können Sie auch im Dialog **Boxformate verwalten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Boxformats

Code-Beispiel

```
Dim boxFormat As VcBoxFormat

For Each boxFormat In VcTree1.BoxFormatCollection
   List1.AddItem (boxFormat.Name)
Next
```

Specification

Nur-Lese-Eigenschaft von VcBoxFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Boxformats auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Boxformats mit der Methode **VcBoxFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Boxformats

Methoden

CopyFormatField

Methode von VcBoxFormat

Mit dieser Methode können Sie ein Boxformatfeld kopieren. Das neue VcBoxFormatField-Objekt wird zurückgegeben. Es erhält den nächsten, noch nicht vergebenen Index.

	Datentyp	Beschreibung
Parameter:		
⇨ position	FormatFieldInnerPositionEnum **Mögliche Werte:** vcInnerAbove 1 vcInnerBelow 3 vcInnerLeftOf 0 vcInnerRightOf 4	Position des neuen Boxformatfeldes oberhalb unterhalb links von rechts von
⇨ refIndex	Integer	Index des Referenz-Boxformatfeldes
Rückgabewert	VcBoxFormatField	Boxformatfeld-Objekt

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim formatField As VcBoxFormatField

Set boxFormat = VcTree1.BoxFormatCollection.FormatByIndex(2)
Set formatField = boxFormat.CopyFormatField(vcInnerRightOf, 0)
```

RemoveFormatField

Methode von VcBoxFormat

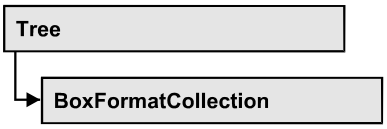
Mit dieser Methode können Sie ein Boxformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Boxformatfelder neu festgesetzt, so dass sie wieder fortlaufend numeriert sind.

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

Code-Beispiel

```
Dim boxFormat As VcBoxFormat
Dim i As Integer

boxFormat = VcTree1.BoxFormatCollection.FirstFormat

For i = 0 To boxFormat.FormatFieldCount - 1
   boxFormat.RemoveFormatField (i)
Next
```

7.9 VcBoxFormatCollection



Im VcBoxFormat-Auflistungsobjekt sind alle verfügbaren Boxformate zusammengefasst. Über **For Each boxFormat In BoxFormatCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Formate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **BoxFormatByName** und **BoxFormatByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen BoxFormate kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add…**, **Copy…** und **Remove…** ermöglichen das Hinzufügen, Kopieren und Löschen von Boxformaten.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstFormat
- FormatByIndex
- FormatByName
- NextFormat
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcBoxFormatCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Boxformatobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim format As VcBoxFormat

For Each format In VcTree1.BoxCollection
   Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcBoxFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Boxformatobjekte in der BoxFormat-Auflistung abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Boxformate

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim numberOfBoxformats As Long

Set boxFormatCltn = VcTree1.BoxFormatCollection
Dim numberOfBoxformats = boxFormatCltn.Count
```

Methoden

Add

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein neues Boxformat in der BoxFormatCollection anlegen. Wenn der Name noch nicht verwendet wird, wird das neue Boxformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ FormatName	String	Name des Boxformats

Rückgabewert	VcBoxFormat	Neues Boxformatobjekt

Code-Beispiel

```
Set newBoxFormat = VcTree1.BoxFormatCollection.Add("boxFormat1")
```

AddBySpecification

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein Boxformat über eine Boxformat-Spezifikation erzeugen. Dies dient der Persistenz von Boxformatobjekten. Die Spezifikation eines Boxformats kann erfragt (siehe VcBoxFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Boxformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ formatSpecification	String	Boxformatspezifikation
Rückgabewert	VcBoxFormat	Neues Boxformatobjekt

Copy

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein Boxformat kopieren. Wenn das Boxformat mit dem angegebenen Namen existiert und der Name des neuen Boxformats noch nicht verwendet wird, wird das neue Boxformatobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	**Beschreibung**
Parameter:		
⇨ FormatName	String	Name des zu kopierenden Boxformats
⇨ newFormatName	String	Name des neuen Boxformats
Rückgabewert	VcBoxFormat	Boxformatobjekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormat = boxFormatCltn.Copy("CurrentBoxFormat", "NewBoxFormat")
```

FirstFormat

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie auf das erste Boxformat der BoxFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Boxformate zu iterieren. Existiert kein Boxformat in der BoxFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBoxFormat	erstes Boxformat

Code-Beispiel

```
Dim format As VcBoxFormat

Set format = VcTree1.BoxFormatCollection.FirstFormat
```

FormatByIndex

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie auf ein einzelnes Boxformat über seinen Index zugreifen. Existiert kein Boxformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Boxformats
Rückgabewert	VcBoxFormat	Ermitteltes Boxformat-Objekt

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set format = boxFormatCltn.FormatByIndex(2)
```

FormatByName

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Boxformat zugreifen. Existiert kein Boxformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ formatName	String	Name des Boxformats
Rückgabewert	VcBoxFormat	Boxformat

Code-Beispiel

```
Dim formatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set formatCltn = VcTree1.BoxFormatCollection
Set format = formatCltn.FormatByName("Standard")
```

NextFormat

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Boxformate der BoxFormat-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFormat** den Initialwert erfasst haben. Sind alle Formate durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcBoxFormat	Nachfolgendes Boxformat

Code-Beispiel

```
Dim formatCltn As VcBoxFormatCollection
Dim format As VcBoxFormat

Set formatCltn = VcTree1.BoxFormatCollection
Set format = formatCltn.FirstFormat

While Not format Is Nothing
     List1.AddItem format.Name
     Set format = formatCltn.NextFormat
Wend
```

Remove

Methode von VcBoxFormatCollection

Mit dieser Methode können Sie ein Boxformat löschen. Wenn das Boxformat noch in irgendeinem anderen Objekt benutzt wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

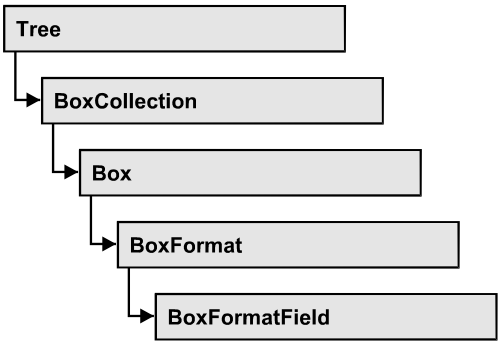
	Datentyp	Beschreibung
Parameter:		
⇨ FormatName	String	Name des Boxformats

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

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormat As VcBoxFormat

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormat = boxFormatCltn.FormatByIndex(1)
boxFormatCltn.Remove (boxFormat.Name)
```

7.10 VcBoxFormatField



Ein Objekt vom Typ **VcBoxFormatField** stellt ein Boxformatfeld, also ein Feld eines VcBoxFormat-Objekts dar. Ein Boxformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Boxformat untergebracht ist.

Eigenschaften

- Alignment
- FormatName
- GraphicsHeight
- Index
- MaximumTextLineCount
- MinimumTextLineCount
- MinimumWidth
- PatternBackgroundColorAsARGB
- PatternColorAsARGB
- PatternEx
- TextFont
- TextFontColor
- Type

Eigenschaften

Alignment

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Boxformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FormatFieldAlignmentEnum **Mögliche Werte:** vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	Ausrichtung des Feldinhalts unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.Alignment = vcFFACenter
```

FormatName

Nur-Lese-Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie den Namen des Boxformats erfragen, zu dem dieses Boxformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Boxformats

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
MsgBox boxFormatField.FormatName
```

GraphicsHeight

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** die Höhe der Grafik in dem Boxformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe der Grafik in mm 0 ... 200

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTGraphics
boxFormatField.GraphicsHeight = 150
```

Index

Nur-Lese-Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie den Index des Boxformatfelds im zugehörigen Boxformat erfragen.

	Datentyp	Beschreibung

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
MsgBox boxFormatField.Index
```

MaximumTextLineCount

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die maximale Anzahl der Zeilen in dem Boxformatfeld setzen oder erfragen, falls das Boxformatfeld vom Typ **vcFFTText** ist. Bitte sehen Sie auch die Eigenschaft **MinimumTextLine-Count**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Maximale Zeilenzahl 0 ... 20

Code-Beispiel

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

```
Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTText
boxFormatField.MaximumTextLineCount = 5
```

MinimumTextLineCount

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die minimale Anzahl der Zeilen in dem Boxformatfeld setzen oder erfragen, falls der Typ des Boxformatfeldes auf **vcFFTText** gesetzt wurde. Ist in einem Knoten mehr Text vorhanden, als in die minimale Anzahl der Zeilen hineinpasst, wird dieses Feld für diesen Knoten dynamisch bis zur maximalen angegebenen Anzahl der Zeilen ausgedehnt. Wenn Sie dieser Eigenschaft einen Wert zuweisen, sollten Sie anschließend auch erneut der Eigenschaft **MaximumTextLineCount** den gewünschten Wert setzen, sonst könnte es vorkommen, dass das Maximum durch das Minimum überschrieben wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Zeilenzahl 0 ... 20

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTText
boxFormatField.MinimumTextLineCount = 3
```

MinimumWidth

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Boxformatfeldes in mm festlegen oder erfragen. Die Breite des Feldes kann sich vergrößern, wenn unter oder über dem Feld andere Felder größere minimale Breiten besitzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Minimale Breite des Boxformatfeldes 0 ... 200

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
```

```
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.MinimumWidth = 100
```

PatternBackgroundColorAsARGB

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Boxformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Boxformats besitzen soll.

	Datentyp	**Beschreibung**
Eigenschaftswert	Long	Hintergrundfarbe des Boxformatfelds **Standardwert:** -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.BackColor = RGB(0, 255, 0)
```

PatternColorAsARGB

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Boxformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Boxformats besitzen soll.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Musterfarbe des Boxformatfeldes

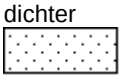
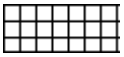
Code-Beispiel

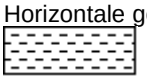
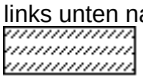
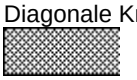
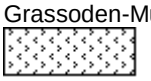
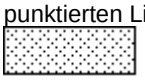
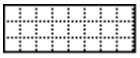
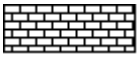
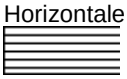
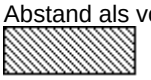
```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.formatField(0)
boxFormatField.PatternColor = RGB(0, 255, 0)
```

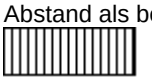
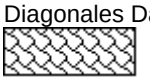
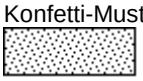
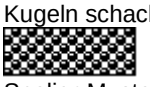
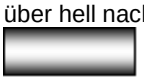
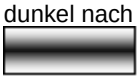
PatternEx

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Boxformatfeldes ein Muster setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp **Standardwert:** Wie im Dialog definiert
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppeltelter Liniendicke 

vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmall-CheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 

vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vc-HorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vc-HorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 
vcVerticalGradientPattern 62	Vertikaler Farbverlauf 

vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke 
vcZigZagPattern 2030	Horizontale Zickzack-Linien 

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Pattern = vcSingleColoredNoPattern
```

TextFont

Eigenschaft von VcBoxFormatField

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

	Datentyp	**Beschreibung**
Eigenschaftswert	StdFont	Schriftart des Boxformatfelds

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.TextFont.Bold = True
```

TextFontColor

Eigenschaft von VcBoxFormatField

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

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Boxformatfelds **Standardwert:** -1

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.TextFontColor = RGB(0, 255, 0)
```

Type

Eigenschaft von VcBoxFormatField

Mit dieser Eigenschaft können Sie den Typ des Boxformatfelds erfragen.

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

Code-Beispiel

```
Dim boxFormatCltn As VcBoxFormatCollection
Dim boxFormatField As VcBoxFormatField

Set boxFormatCltn = VcTree1.BoxFormatCollection
Set boxFormatField = boxFormatCltn.FirstFormat.FormatField(0)
boxFormatField.Type = vcFFTGraphics
boxFormatField.GraphicsHeight = 200
```

7.11 VcDataDefinition

Die Definition der Daten von Knoten und Verbindungen wird über den Dialog **Datentabellen verwalten**, erreichbar über die Eigenschaftenseite **Objekte: Datentabellen...**, eingestellt. Das Objekt ermöglicht den Zugriff auf die Namen und Typen der verfügbaren Felder. Die Datendefinition eines VcTree-Objektes enthält zwei Datendefinitionstabellen: **vcMaindata** und **vcRelations**.

Eigenschaften

- DefinitionTable

Eigenschaften

DefinitionTable

Nur-Lese-Eigenschaft von VcDataDefinition

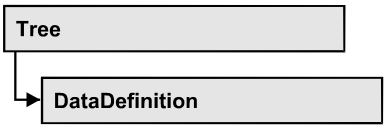
Diese Eigenschaft ermöglicht den Zugriff auf die Tabelle **vcMaindata** des Datendefinitionsobjekts. Diese enthält die Definitionen für die Knotendaten.

	Datentyp	**Beschreibung**
Parameter:		
⇨ tableType	DataTableEnum	Typ der Datendefinitionstabelle
	Mögliche Werte: vcMaindata 0	Tabellentyp **vcMaindata** (für Knoten)
Eigenschaftswert	VcDataDefinitionTable	Datendefinitionstabelle

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable

Set dataDefinition = VcTree1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
```

7.12 VcDataDefinition



Die Definition der Daten von Knoten wird über den Dialog **Datentabellen verwalten**, erreichbar über die Eigenschaftenseite **Objekte: Datentabellen...**, eingestellt. Das Objekt ermöglicht den Zugriff auf die Namen und Typen der verfügbaren Felder. Die Datendefinition eines VcTree-Objektes enthält die Datendefinitionstabelle **vcMaindata**.

Eigenschaften

- DefinitionTable

Eigenschaften

DefinitionTable

Nur-Lese-Eigenschaft von VcDataDefinition

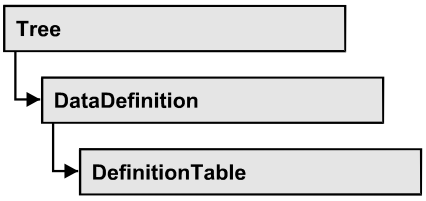
Diese Eigenschaft ermöglicht den Zugriff auf die Tabelle **vcMaindata** des Datendefinitionsobjekts. Diese enthält die Definitionen für die Knotendaten.

	Datentyp	**Beschreibung**
Parameter:		
⇨ tableType	DataTableEnum	Typ der Datendefinitionstabelle
	Mögliche Werte: vcMaindata 0	Tabellentyp **vcMaindata** (für Knoten)
Eigenschaftswert	VcDataDefinitionTable	Datendefinitionstabelle

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable

Set dataDefinition = VcTree1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)
```

7.13 VcDataDefinitionTable



Ein Objekt vom Typ VcDataDefinitionTable ist ein Element der Datendefinition und stellt eine Tabelle aus Datendefinitionsfeldern dar. Auf diese Felder können Sie einzeln über die Methoden **FieldByIndex** oder **FieldByName** zugreifen oder über eine Schleife mit **FirstField** und **NextField** alle Felder abfragen. Über die Eigenschaft **Count** erhalten Sie die Anzahl der Felder. Die Definitionstabelle wird auf der Eigenschaftenseite **Datentabellen verwalten** voreingestellt.

Eigenschaften

- _NewEnum
- Count

Methoden

- CreateDataField
- FieldByIndex
- FieldByName
- FirstField
- NextField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcDataDefinitionTable

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datendefinitionsfelder iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim datdeftable As VcDataDefinitionTable

For Each datdeftable In VcTree1.VcDataDefinition
   Debug.Print datdeftable.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcDataDefinitionTable

Mit dieser Eigenschaft kann die Anzahl der Felder in der Datendefinitionstabelle abgefragt werden. Anlegen können Sie Datenfelder im Dialog **Datentabellen verwalten** oder zur Laufzeit mit Hilfe der Methode **CreateDataField**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Felder

Code-Beispiel

```
Dim dataDefinition As VcDataDefinition
Dim dataDefinitionTable As VcDataDefinitionTable
Dim numberOfFields As Long

Set dataDefinition = VcTree1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)

numberOfFields = dataDefinitionTable.Count
```

Methoden

CreateDataField

Methode von VcDataDefinitionTable

Mit dieser Methode kann zur Laufzeit ein neues Datenfeld an das Ende der Datendefinitionstabelle angefügt werden. Das neue Datenfeld hat standardmäßig den Datentyp Integer; er kann aber mit Hilfe der Eigenschaft **Type** von **VcDefinitionField** geändert werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ newfieldName	String	Name des neuen Feldes
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable =  _ VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.CreateDataField("Description")
dataDefinitionField.Type = vcDefFieldAlphanumericType
VcTree1.DataTableCollection.Update
```

FieldByIndex

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie über den Index auf ein beliebiges Feld der Datentabelle zugreifen. Jedes Feld kann über seinen Namen oder über seinen Index angesprochen werden. Der Index für das erste Feld ist 0. Im Dialog **Datentabellen verwalten** können Sie die Datendefinitionen bearbeiten.

	Datentyp	**Beschreibung**
Parameter:		
⇨ fieldIndex	Integer	Index des Feldes
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField
Set dataDefinitionTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)

Set dataDefinitionField = dataDefinitionTable.FirstField
For I = 1 To dataDefinitionTable.Count
      List1.AddItem dataDefinitionField.Name
      Set dataDefinitionField = dataDefinitionTable.FieldByIndex(I)
Next
```

FieldByName

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie unter Verwendung des Feldnamens auf ein bestimmtes Feld der Datentabelle zugreifen. Existiert kein Feld unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**). Jedes Feld kann über seinen Namen oder über seinen Index

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

	Datentyp	Beschreibung
Parameter:		
⇨ fieldName	String	Feldname
Rückgabewert	VcDefinitionField	Datendefinitionsfeld

Code-Beispiel

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

Set dataDefinition = VcTree1.DataDefinition
Set dataDefinitionTable = dataDefinition.DefinitionTable(vcMaindata)

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

FirstField

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie auf das erste Feld der Datentabelle zugreifen, um anschließend in einer Schleife mit der Methode **NextField** über die nachfolgenden Felder zu iterieren. Existiert kein Feld in der Datentabelle, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Erstes Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefinitionField = dataDefinitionTable.FirstField
```

NextField

Methode von VcDataDefinitionTable

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Felder der Datentabelle zugreifen, nachdem Sie die Methode **FirstField** aufgerufen haben.

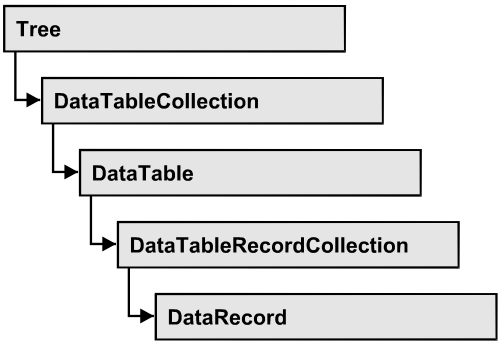
Wenn kein weiteres Feld mehr existiert, wird ein Leerobjekt zurückgegeben (in Visual Basic: Nothing).

	Datentyp	Beschreibung
Rückgabewert	VcDefinitionField	Nachfolgendes Datendefinitionsfeld

Code-Beispiel

```
Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)

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

or

Dim dataDefinitionTable As VcDataDefinitionTable
Dim dataDefinitionField As VcDefinitionField

Set dataDefinitionTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)

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

7.14 VcDataRecord



Ein Datensatz ist das logische Grundelement eines Objektes in einem Diagramm, z.B. eines Knotens o.ä. Die Objekte besitzen spezifische Eigenschaften, die in den Feldern des Datensatzes beschrieben werden. Zu den Datenfeldern des Datensatzes existieren entsprechende Beschreibungen, die Datentabellenfelder. Datensätze und Datentabellenfelder werden jeweils zu Collection-Objekten zusamengefasst und bilden eine Datentabelle.

Eigenschaften

- AllData
- DataField
- DataTableName
- ID

Methoden

- DeleteDataRecord
- IdentifyObject
- RelatedDataRecord
- UpdateDataRecord

Eigenschaften

AllData

Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können alle Daten eines Datensatzes gesetzt oder erfragt werden. Beim Setzen ist ein CSV-String (Semikolon als Trennzeichen) oder der Datentyp "Variant" erlaubt, der in einem Array alle

Datenfelder des Datensatzes erhält. Beim Erfragen wird eine Zeichenkette (String) zurückgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Variant	Alle Daten des Datensatzes

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRecValue() As Variant
Dim dataRecord As VcDataRecord

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata1")
Set dataRecCltn = dataTable.DataRecordCollection
ReDim dataRecValue(dataTable.DataTableFieldCollection.Count)
dataRecValue(0) = 1
dataRecValue(1) = "Node One"

'Variant
Set dataRecord = dataRecCltn.Add(dataRecValue)
'CSV
dataRecord.AllData = "1;Node One;"

dataRecord.UpdateDataRecord
```

DataField

Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie einem Datenfeld des Datensatzes einen Wert zuweisen oder einen gesetzten Wert erfragen. Wenn ein Datensatz durch diese Methode einen neuen Wert erhalten hat, muss anschließend die grafische Darstellung mit der Methode **UpdateDataRecord** aktualisiert werden.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Datenfeldes
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

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

DataTableName

Nur-Lese-Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie den Namen der Datentabelle erfragen, zu der dieser Datensatz gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der zugehörigen Tabelle

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

MsgBox dataRecord.DataTableName
```

ID

Nur-Lese-Eigenschaft von VcDataRecord

Mit dieser Eigenschaft können Sie die ID eines Datensatzes erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datensatz-ID

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection
Dim dataRecord As VcDataRecord
Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)
MsgBox dataRecord.ID
```

Methoden

DeleteDataRecord

Methode von VcDataRecord

Mit dieser Methode können Sie einen Datensatz löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Datensatz erfolgreich (true) / nicht erfolgreich (false) gelöscht

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecCltn.DataRecordByID(1)

dataRecord.DeleteDataRecord
```

IdentifyObject

Methode von VcDataRecord

Mit dieser Methode kann man erfragen, ob und welches datenbasierte Objekt aus dem Datensatz erzeugt wurde.

Die Methode liefert als Rückgabewert **true**, wenn ein datenbasiertes Objekt ermittelt werden konnte, d.h. wenn aus dem Datensatz für die Grafik ein datenbasiertes Objekt hergestellt wurde.

	Datentyp	Beschreibung
Parameter:		
⇨ establishedObject Param	Object	Erkanntes Objekt
establishedObjectTypeParam	VcObjectTypeEnum	Typ des erkannten Objekts
	Mögliche Werte:	
	vcObjTypeBox 15	Objekttyp **Box**
	vcObjTypeNode 2	Objekttyp **Knoten**
	vcObjTypeNodeInLegend 17	Objekttyp **Knoten im Legendenbereich**
	vcObjTypeNone 0	kein Objekt
Rückgabewert	Boolean	datenbasiertes Objekt wurde/wurde nicht erzeugt

RelatedDataRecord

Methode von VcDataRecord

Mit dieser Eigenschaft können Sie einem Datensatz einen weiteren zuordnen oder einen zugeordneten Datensatz erfragen. Bei der Verwendung von erweiterten Tabellen (extended data tables) können Datensätze einer Tabelle

über einen Primärschlüssel den Datensätzen einer anderen Tabelle zugeordnet werden.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Datenfeldes
Rückgabewert	VcDataRecord	Zugeordneter Datensatz

Code-Beispiel

```
Private Sub VcTree1_OnNodeLClick(ByVal node As VcTreeLib.VcNode, ByVal location
As VcTreeLib.LocationEnum, ByVal x As Long, ByVal y As Long, returnStatus As
Variant)

    Dim dataTable As VcDataTable
    Dim dataRecordCltn As VcDataRecordCollection
    Dim firstDataRecord As VcDataRecord
    Dim secondDataRecord As VcDataRecord

    Set dataTable = VcTree1.DataTableCollection.DataTableByIndex(0)
    Set dataRecordCltn = dataTable.DataRecordCollection

    Set firstDataRecord = dataRecordCltn.DataRecordByID(node.DataField(0))
    Set secondDataRecord = firstDataRecord.RelatedDataRecord(2)

    MsgBox secondDataRecord.AllData

End Sub
```

UpdateDataRecord

Methode von VcDataRecord

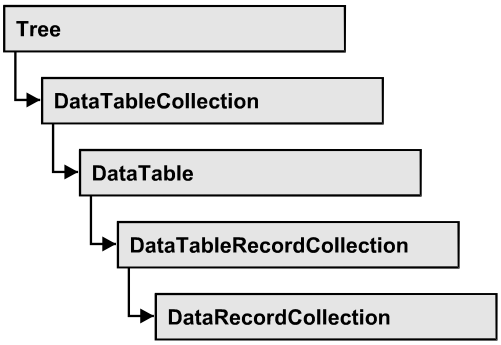
Nachdem Sie ein oder mehrere Datenfelder eines Datensatzes mit der Eigenschaft **DataField** verändert haben, aktualisieren Sie die grafische Darstellung im Diagramm mit **UpdateDataRecord**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Datensatz erfolgreich (true) / nicht erfolgreich (false) aktualisiert

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(1)

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

7.15 VcDataRecordCollection



In einem Objekt vom Typ **VcDataRecordCollection** sind die Datensätze einer Datentabelle zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Datensätze im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataRecord** und **NextDataRecord** können Sie iterativ auf die Datensätze zugreifen sowie mit **DataRecordByID** auf einzelne Datensätze; die Methoden **Add** und **Remove** ermöglichen das Hinzufügen und Entfernen von Datensätzen und mit **Update** können Sie die grafische Darstellung der Datensätze mit neu eingegebenen Daten aktualisieren.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- DataRecordByID
- FirstDataRecord
- GetNewUniqueID
- NextDataRecord
- Remove
- Update

Eigenschaften

_NewEnum

Eigenschaft von VcDataRecordCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datensätze iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

For Each dataRecord In dataRecordCltn
   Debug.Print dataRecord.AllData
Next dataRecord
```

Count

Nur-Lese-Eigenschaft von VcDataRecordCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datensätze in der Data-Record-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datensätze im Collection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecordCltn As VcDataRecordCollection

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
MsgBox "Number of DataRecords: " & dataRecordCltn.Count
```

Methoden

Add

Methode von VcDataRecordCollection

Mit dieser Methode können Sie einen neuen Datensatz in der DataRecord-Collection anlegen. Wenn die Datensatzbeschreibung die Anlage eines neuen Datensatzes erlaubte, wird der neue Datensatz zurückgegeben, andernfalls wird eine **VcPrimaryKeyNotUniqueException** erzeugt.

Nach dem Hinzufügen des Datensatzes muss die Methode **VcTree.EndLoading** aufgerufen werden, damit die Änderung wirksam wird.

	Datentyp	Beschreibung
Parameter:		
⇨ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)
Rückgabewert	VcDataRecord	Neue angelegter Datensatz

Code-Beispiel

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

'...

Dim dataTable As VcDataTable
Dim dataRecCltn As VcDataRecordCollection
Dim dataRec1 As VcDataRecord
Dim dataRecVal() As Variant

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecCltn = dataTable.DataRecordCollection

ReDim dataRecVal(dataTable.DataTableFieldCollection.Count)

dataRecVal(Main_ID) = 1
dataRecVal(Main_Name) = "Node 1"
dataRecVal(Main_Start) = DateSerial(2014, 1, 8)
dataRecVal(Main_Duration) = 8
Set dataRec1 = dataRecCltn.Add(dataRecVal)
VcTree1.EndLoading()

' equivalent
' dataRec1 = dataRecCltn.Add("1;Node 1;01.08.14;;8")
```

DataRecordByID

Methode von VcDataRecordCollection

Mit dieser Methode können Sie auf einen einzelnen Tabellendatensatz über seine Identifikation zugreifen. Existiert kein Datensatz unter der angegebenen ID, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter:		
⇨ dataRecordID	String	ID des Datensatzes
Rückgabewert	VcDataRecord	Datensatzobjekt

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
Set dataRecord = dataRecordCltn.DataRecordByID(0)
```

FirstDataRecord

Methode von VcDataRecordCollection

Mit dieser Methode können Sie auf den ersten Datensatz der Datenstz-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataRecord** über die nachfolgenden Datensätze zu iterieren. Existiert kein Datensatz in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Erster Datensatz

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
```

```
Set dataRecord = dataRecordCltn.FirstDataRecord
```

GetNewUniqueID

Methode von VcDataRecordCollection

Mit dieser Methode können Sie eine eindeutige ID für einen Datensatz generieren lassen. Die Methode ist nützlich, wenn Sie z.B. mit **Add** einen neuen Datensatz generieren und die ID nicht manuell vergeben.

	Datentyp	Beschreibung
Rückgabewert	Long	Neue Datensatz-ID

NextDataRecord

Methode von VcDataRecordCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datensätze der Datensatzauflistung zugreifen, nachdem Sie mit der Methode **FirstDataRecord** den Initialwert erfasst haben. Sind alle Datensätze durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Nachfolgender Datensatz

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcTree1.SuspendUpdate True

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

VcTree1.SuspendUpdate False
```

Remove

Methode von VcDataRecordCollection

Mit dieser Methode können Sie einen Datensatz löschen. Die Methode liefert **true** wenn gelöscht wurde und **false**, wenn nicht gelöscht wurde. Der Datensatzinhalt im Übergabeparameter wird dazu verwendet, um anhand der Identifizierung das Objekt zu finden.

	Datentyp	Beschreibung
Parameter:		
⇨ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)
Rückgabewert	Boolean	True

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection

VcTree1.SuspendUpdate True

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

VcTree1.SuspendUpdate False
VcTree1.EndLoading()
```

Update

Methode von VcDataRecordCollection

Mit dieser Methode können Sie einen Datensatz in der Datensatzliste aktualisieren, nachdem mittels der Methode **Add()** der Datensatz vorher hinzugefügt wurde. Falls der zu aktualisierende Datensatz nicht existiert, d.h. also nicht vorher angelegt wurde, wird er mittels Update() nun neu angelegt. Siehe auch **VcDataRecordCollection.Add()**.

Nach dem Aktualisieren des Datensatzes muss die Methode **VcTree.End-Loading** aufgerufen werden, damit die Änderung wirksam wird.

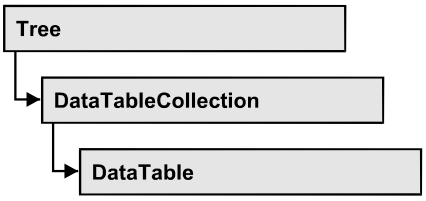
	Datentyp	Beschreibung
Parameter:		
⇨ dataRecordContent	Object	Inhalt des Datensatzes (als Array oder String)

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

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
Set dataRecordCltn = dataTable.DataRecordCollection
dataRecordCltn.Update("1;1.8.2017;;8")
VcTree1.EndLoading()
```

7.16 VcDataTable



Eine Datentabelle umfasst **Datensätze** (data records) mit ihren Datenfeldern und ihren Inhalten sowie die Beschreibungen der Datenfelder, die **Tabellen-datenfelder** (data table fields) genannt werden. Datensätze und Datentabellenfelder können in der Form von eigenen Auflistungen (collection) verwaltet werden.

Datentabellen ihrerseits können ebenfalls in eigenen Auflistungen verwaltet werden.

Eigenschaften

- DataRecordCollection
- DataTableFieldCollection
- Description
- MultiplePrimaryKeysAllowed
- Name

Eigenschaften

DataRecordCollection

Nur-Lese-Eigenschaft von VcDataTable

Diese Eigenschaft gibt die in der Datentabelle enthaltene Datensatz-Auflistung zurück. Die Datensatz-Auflistung enthält alle existierenden Datensätze einer Tabelle. Zu Programmbeginn ist sie leer.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataRecordCollection	DataRecordCollection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.FirstDataTable()
MsgBox dataTable.DataRecordCollection.Count
```

DataTableFieldCollection

Nur-Lese-Eigenschaft von VcDataTable

Diese Eigenschaft gibt die in der Datentabelle enthaltene Datentabellenfeld-Auflistung zurück. Die Datentabellenfeld-Auflistung enthält die Definition der Datenfelder eines Datensatzes der Tabelle. Zu Programmbeginn enthält sie die bereits zur Designzeit vereinbarten Datenfelder. Weitere Datenfelder können zur Laufzeit über die Methode **Add** des Objekts **DataTableField-Collection** hinzugefügt werden. Die Definition der Datentabellenfelder muss abgeschlossen sein, bevor die Tabelle mit Datensätzen gefüllt wird.

	Datentyp	Beschreibung
Eigenschaftswert	VcTableFieldCollection	DataTableFieldCollection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.DataTableByIndex(0)
MsgBox dataTable.DataTableFieldCollection.Count
```

Description

Eigenschaft von VcDataTable

Mit dieser Eigenschaft können sie eine Beschreibung der Datentabelle setzen oder erfragen. Sprechende Namen, z.B. der Name der Tabelle, sind häufig sehr lang und werden daher bei Previews nicht vollständig angezeigt, so dass ihr Nutzen nicht zum Tragen kommen kann. Damit Sie für die vollständige Anzeige kurze Namen verwenden können und trotzdem nicht auf die gewünschte Information verzichten müssen, können Sie in diesem Feld zusätzliche Informationen zum Tabellennamen speichern. Sein Inhalt wird im Dialog zur Datentabelle angezeigt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Beschreibung der Datentabelle **Standardwert:** Leere Zeichenkette

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Maindata")
dataTable.Description = "This table contains data for nodes"
```

MultiplePrimaryKeysAllowed

Eigenschaft von VcDataTable

Mit dieser Eigenschaft können Sie angeben oder erfragen, ob die Verwendung zusammengesetzter Primärschlüssel möglich ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Verwendung von zusammengesetzten Primärschlüsseln erlaubt (true)/nicht erlaubt (false) **Standardwert:** False

Name

Eigenschaft von VcDataTable

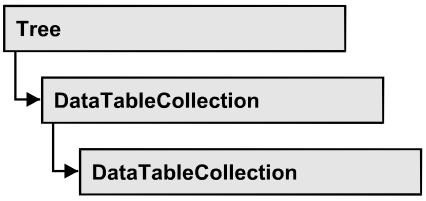
Mit dieser Eigenschaft können sie den Namen der Datentabelle setzen oder erfragen. Ein Tabellenname ist obligat und muss eindeutig sein; zudem ist eine leere Zeichenkette nicht erlaubt. Unterscheidungen in der Groß- und Kleinschreibung führen zu unterschiedlichen Namen. Mit der Methode **DataTableByName** des Objekts **DataTableCollection** können Sie über den Tabellennamen eine Referenz auf das Datentabellenobjekt erhalten.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle **Standardwert:** Leere Zeichenkette

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.DataTableByIndex(0)
MsgBox dataTable.Name
```

7.17 VcDataTableCollection



In einem Objekt vom Typ VcDataTableCollection sind die vorhandenen Datentabellen zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Tabellen im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataTable** und **NextDataTable** können Sie iterativ auf die Tabellen zugreifen sowie mit **DataTableByName** und **DataTableByIndex** auf einzelne Tabellen; die Methoden **Add** und **Copy** ermöglichen das Hinzufügen und Kopieren von Tabellen und mit **Update** können Sie dem XTree-Objekt die neuen Änderungen der Datenstrukturen bekanntgeben, das heißt aktualisieren.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- Copy
- DataTableByIndex
- DataTableByName
- FirstDataTable
- NextDataTable
- Update

Eigenschaften

_NewEnum

Eigenschaft von VcDataTableCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts können Sie über alle enthaltenen Datentabellen iterieren. In Visual Basic wird diese Eigenschaft

nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
For Each dataTable In dataTableCltn
   List1.AddItem (dataTable.Name)
Next
```

Count

Eigenschaft von VcDataTableCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datentabellen in der DataTable-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datentabellen im Collection-Objekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection

Set dataTableCltn = VcTree1.DataTableCollection
MsgBox (dataTableCltn.Count)
```

Methoden

Add

Methode von VcDataTableCollection

Mit dieser Methode können Sie eine neue Datentabelle in der DataTable-Auflistung anlegen. Wenn der Tabellenname noch nicht verwendet wurde, wird ein Objekt vom Typ **VcDataTable** zurückgegeben, andernfalls **Nothing** (in Visual Basic) oder **0** (in anderen Sprachen). Nur wenn die Eigenschaft **ExtendedDataTables** bei dem Objekt VcTree auf **True** gesetzt ist, können

Tabellen angelegt werden. Maximal 90 Datentabellen sind insgesamt in der Auflistung möglich.

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableName	String	Name der Datentabelle
Rückgabewert	VcDataTable	Neu angelegte Datentabelle

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.Add("Resources")
dataTableCltn.Update
```

Copy

Methode von VcDataTableCollection

Mit dieser Methode können Sie eine Datentabelle kopieren. Es wird nur die Tabellendefinition kopiert, jedoch nicht die eventuell bereits existierende Datensätze. Nur wenn die Eigenschaft **ExtendedDataTables** bei dem Objekt VcNet auf **True** gesetzt ist, können Tabellen angelegt werden. Wenn die Tabelle kopiert werden konnte, wird ein neues Objekt vom Typ **VcDataTable** zurückgegeben, andernfalls **Nothing** (in Visual Basic) oder **0** (in anderen Sprachen). Es wird bei den Tabellennamen generell zwischen Groß- und Kleinschreibung unterschieden.

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableName	String	Name der zu kopierenden Datentabelle (Quelltabelle)
⇨ newDataTableName	String	Name der neu zu erstellenden Datentabelle (Zieltabelle)
Rückgabewert	VcDataTable	Neu erstelltes Datentabellen-Objekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.Copy("Resources", "NewResources")
dataTableCltn.Update
```

DataTableByIndex

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf eine einzelne Datentabelle über ihren Index zugreifen. Der Index der ersten Tabelle ist 0. Existiert keine Tabelle unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (**Nothing** in Visual Basic oder **0** in anderen Sprachen).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index der Datentabelle
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByIndex(2)
MsgBox (dataTable.Name)
```

DataTableByName

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf eine einzelne Datentabelle über ihren Namen zugreifen. Existiert keine Tabelle unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (**Nothing** in Visual Basic oder **0** in anderen Sprachen).

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableName	String	Name der Tabelle
Rückgabewert	VcDataTable	Ermitteltes Datentabellenobjekt

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.DataTableByName("Resources")
MsgBox (dataTable.Description)
```

FirstDataTable

Methode von VcDataTableCollection

Mit dieser Methode können Sie auf die erste Datentabelle des DataTable-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataTable** über die nachfolgenden Tabellen zu iterieren. Existiert keine Datentabelle in der DataTable-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataTable	Erste Datentabelle

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.FirstDataTable
```

NextDataTable

Methode von VcDataTableCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datentabellen der DataTable-Auflistung zugreifen, nachdem Sie mit der Methode **FirstDataTable** den Initialwert erfasst haben. Sind alle Tabellen durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcDataTable	Nachfolgende Datentabelle

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable
Dim i As Integer

Set dataTableCltn = VcTree1.DataTableCollection
Set dataTable = dataTableCltn.FirstDataTable
For i = 0 To dataTableCltn.Count
   List1.AddItem (dataTable.Name)
   Set dataTable = dataTableCltn.NextDataTable
Next i
```

Update

Methode von VcDataTableCollection

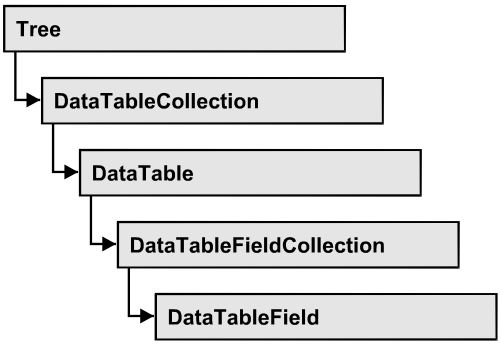
Mit dieser Methode können Sie neue Änderungen an Datenstrukturen aktualisieren. Der Aufruf ist notwendig, damit durchgeführte Änderungen an der Datentabellendefinition und an den Datentabellenfeldern in der VARCHART Komponente wirksam werden. Auf diese Weise werden bei mehreren Änderungen unnötige Aktualisierungen vermeiden.

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

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

dataTableCltn = VcTree1.DataTableCollection
dataTable = dataTableCltn.Add("Resources")
dataTable.DataTableFieldCollection.Add ("Id")
dataTableCltn.Update
```

7.18 VcDataTableField



Ein Objekt vom Typ **VcDataTableField** legt die Eigenschaften eines Feldes der Datentabelle fest. Zur Definition eines Datentabellenfeldes gehört der Name, der Datentyp und die Festlegung, ob das Datenfeld als Primärschlüssel dient, der zur eindeutigen Identifizierung eines Datensatzes herangezogen werden kann. Unter Verwendung des Primärschlüssels z.B. kann von anderen Datentabellen auf diese Datentabelle Bezug genommen werden. Um eine Beziehung zu einer Datentabelle mit Primärschlüssel aufzubauen, muss der Primärschlüssel im Feld **RelationshipFieldIndex** benannt werden.

Die DataTableField-Objekte einer Datentabelle werden über die Auflistung **DataTableFieldCollection** verwaltet.

Eigenschaften

- DataTableName
- DateFormat
- Editable
- Hidden
- Index
- Name
- PrimaryKey
- RelationshipFieldIndex
- Type

Eigenschaften

DataTableName

Nur-Lese-Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Namen der zugehörigen Datentabelle erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
MsgBox dataTable.DataTableFieldCollection.FirstDataTableField.DataTableName
```

DateFormat

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie das Datumsformat des Datentabellenfeldes festlegen oder erfragen. Das Datumsformat wird beim Lesen und Schreiben von CSV-Dateien verwendet und wenn der Formattyp **String** beim Hinzufügen eines Datensatzes über die Methode **Add** zur Datensatzauflistung verwendet wird. Diese Eigenschaft ist nur wirksam, wenn der Datentyp des Feldes auf **vcDataTableFieldDateTime** eingestellt ist.

Hinweis: Es sollte zuerst die Eigenschaft **Type** gesetzt werden, bevor die Eigenschaft **DateFormat** vereinbart wird.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat {DMYhms:;./} **Standardwert:** DD.MM.YYYY hh:mm:ss

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Type = vcDataTableFieldDateTimeType
'DateFormat = "01.12.2014"
```

```
dataTableField.DateFormat = "DD.MM.YYYY"
```

Editable

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das spezifizierte Datenfeld zur Laufzeit in der Tabelle (des Diagramms) und des Dialogs **Knoten bearbeiten** editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Feld editierbar (True) / nicht editierbar (False) **Standardwert:** True

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Editable = False
VcTree1.DataTableCollection.Update
```

Hidden

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Datenfeld zur Laufzeit in den Dialogen **EditNode** oder **EditLink** angezeigt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Feld wird nicht angezeigt (True) / angezeigt (False) **Standardwert:** False

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Hidden = True
VcTree1.DataTableCollection.Update
```

Index

Nur-Lese-Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Index des Datentabellenfelds in der zugehörigen Datentabelle erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datentabellenfeldes.

Name

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie den Namen des Datenfelds setzen oder erfragen. Der Name des Datenfelds erscheint innerhalb von Laufzeitdialogen wie beispielsweise dem **EditNode**-Dialog. In der API erfolgt der Zugriff auf die Datenfeldinhalte eines Datensatzes jedoch immer über den Index, den dieses Feld im **DataTableFieldCollection**-Objekt besitzt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Feldes **Standardwert:** Empty string

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField = dataTable.DataTableFieldCollection.Add("Start")
VcTree1.DataTableCollection.Update
```

PrimaryKey

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob dieses Datenfeld den Primärschlüssel enthält, der zur eindeutigen Identifikation eines Datensatz herangezogen werden kann. In einer Datentabelle kann immer nur ein Datenfeld die Kennung Primärschlüssel tragen. Die aktuelle Setzung hebt eine eventuell vorhandene Setzung bei einem anderen Datenfeld der gleichen Tabelle auf. Die Festlegung eines Primärschlüssels ist unerlässlich, wenn eine Beziehung von einer anderen Tabelle zu dieser Tabelle hergestellt werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Das Feld dient (True) / dient nicht (False) als Primärschlüssel **Standardwert:** False

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Id")
dataTableField.PrimaryKey = True
VcTree1.DataTableCollection.Update
```

RelationshipFieldIndex

Eigenschaft von VcDataTableField

Mit dieser Eigenschaft verbinden Sie ein Datenfeld und seine Beschreibung. Dazu setzen Sie hier den Index des Datensatzfeldes, auf welches sich die gesetzten Eigenschaften des Datentabellenfeldes beziehen sollen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datensatzfeldes, auf das sich die Datendefinition des Datentabellenfeldes bezieht. **Standardwert:** -1

Code-Beispiel

```
Dim dataTableTask As VcDataTable
Dim dataTaskFieldId As VcDataTableField
Dim dataTaskFieldName As VcDataTableField
Dim dataTableOperation As VcDataTable
Dim dataOperationFieldId As VcDataTableField
Dim dataOperationFieldName As VcDataTableField
Dim dataOperationFieldTaskId As VcDataTableField

'Create table Task
dataTableTask = VcTree1.DataTableCollection.Add("Task")
dataTaskFieldId = dataTableTask.DataTableFieldCollection.Add("Id")
dataTaskFieldId.PrimaryKey = True
dataTaskFieldName = dataTableTask.DataTableFieldCollection.Add("Name")
dataTaskFieldName.Type = vcDataTableFieldAlphanumericType

'Create table Operation
dataTableOperation = VcTree1.DataTableCollection.Add("Operation")
dataOperationFieldId = dataTableOperation.DataTableFieldCollection.Add("Id")
dataOperationFieldId.PrimaryKey = True
dataOperationFieldName = dataTableOperation.DataTableFieldCollection.Add("Name")
dataOperationFieldName.Type = vcDataTableFieldAlphanumericType
dataOperationFieldTaskId =
dataTableOperation.DataTableFieldCollection.Add("TaskId")
dataOperationFieldTaskId.Type = vcDataTableFieldIntegerType

'Link tables Task and Operations
```

```
dataOperationFieldTaskId.RelationshipFieldIndex =
VcTree1.DetectFieldIndex("Task", "Id")
```

Type

Eigenschaft von VcDataTableField

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

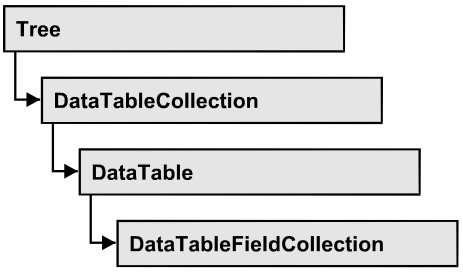
Hinweis: Durch Setzen der Eigenschaft **Type** kann sich die Eigenschaft **DateFormat** ändern. Durch Setzen des Eigenschaftswertes auf **vcDataTableAlphanumeric** oder **vcDataTableFieldInteger** wird ein eventuell eingestelltes Datumsformat auf "" gesetzt.

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

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.DataTableByName("Operation")
Set dataTableField =
dataTable.DataTableFieldCollection.DataTableFieldByName("Start")
dataTableField.Type = vcDataTableFieldDateTimeType
VcTree1.DataTableCollection.Update
```

7.19 VcDataTableFieldCollection



In einem Objekt vom Typ VcDataTableFieldCollection sind die Datentabellenfelder einer Datentabelle zusammengefasst. Mit der Eigenschaft **Count** kann die Anzahl der Felder im Auflistungsobjekt erfragt werden; mit dem Enumerator-Objekt und den Methoden **FirstDataField** und **NextDataField** können Sie iterativ auf die Felder zugreifen sowie mit **DataFieldByName** und **DataFieldByIndex** auf einzelne Felder; die Methoden **Add** und **Copy** ermöglichen das Hinzufügen und Kopieren von Feldern.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- Copy
- DataTableFieldByIndex
- DataTableFieldByName
- FirstDataTableField
- NextDataTableField

Eigenschaften

_NewEnum

Eigenschaft von VcDataTableFieldCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Datentabellenfelder iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator**

angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
For Each dataTableField In dataTable.DataTableFieldCollection
   List1.AddItem (dataTableField.Name)
Next
```

Count

Nur-Lese-Eigenschaft von VcDataTableFieldCollection

Mit dieser Eigenschaft können Sie die Anzahl der Datentabellenfelder in der DataTableField-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Datentabellenfelder im Collection-Objekt

Code-Beispiel

```
Dim dataTable As VcDataTable

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
MsgBox ("Number of data fields: " &  dataTable.DataTableFieldCollection.Count)
```

Methoden

Add

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie ein neues Datentabellenfeld in der DataTableFieldCollection anlegen. Wenn der Name noch nicht verwendet wurde, wird das neue Datenfeld zurückgegeben, andernfalls "Nothing" (Visual Basic) oder "0" (andere Sprachen). Maximal 9.999 Datenfelder können pro Tabelle angelegt werden.

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableFieldName	String	Name des neuen Datentabellenfeldes
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Add("Priority")
VcTree1.DataTableCollection.Update
```

Copy

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie ein Datentabellenfeld kopieren. Die Identifizierung des Feldes erfolgt über den Namen.

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableFieldName	String	Name des zu kopierenden Datentabellenfeldes (Quellfeld)
⇨ newDataTableFieldName	String	Name des neu zu erstellenden Datentabellenfeldes (Zielfeld)
Rückgabewert	VcDataTableField	Neu angelegtes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.Copy("Name", "NewName")
VcTree1.DataTableCollection.Update
```

DataTableFieldByIndex

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf ein einzelnes Datentabellenfeld über seinen Index zugreifen. Existiert kein Feld an dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ Index	Integer	Index des Datentabellenfeldes

Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.DataTableFieldByIndex(1)
MsgBox (dataTableField.Name)
```

DataTableFieldByName

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf ein einzelnes Datentabellenfeld über seinen Namen zugreifen. Existiert kein Feld unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	**Beschreibung**
Parameter:		
⇨ dataTableFieldName	String	Name des Datentabellenfeldes
Rückgabewert	VcDataTableField	Ermitteltes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.DataTableFieldBy("Name")
dataTableField.Editable = False
VcTree1.DataTableCollection.Update
```

FirstDataTableField

Methode von VcDataTableFieldCollection

Mit dieser Methode können Sie auf das erste Datenfeld der DataTableField-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextDataTableField** über die nachfolgenden Datenfelder zu iterieren. Existiert kein Datenfeld in der DataTableField-Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**).

	Datentyp	**Beschreibung**
Rückgabewert	VcDataTableField	Erstes Datentabellenfeld

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataTableField As VcDataTableField
```

```
Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableField = dataTable.DataTableFieldCollection.FirstDataTableField
```

NextDataTableField

Methode von VcDataTableFieldCollection

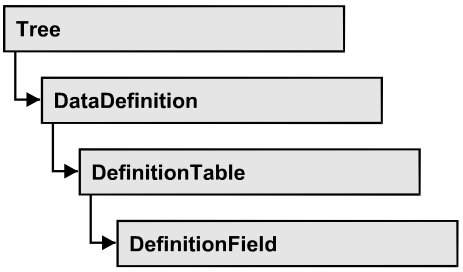
Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Datentabellenfelder der DataTableField-Auflistung zugreifen, nachdem Sie mit der Methode **FirstDataTableField** den Initialwert erfasst haben. Sind alle Felder durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	**Beschreibung**
Rückgabewert	VcDataTableField	Nachfolgendes Datentabellenfeld

Code-Beispiel

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

Set dataTable = VcTree1.DataTableCollection.FirstDataTable
Set dataTableFieldCltn = dataTable.DataTableFieldCollection
Set dataTableField = dataTableFieldCltn.FirstDataTableField
For i = 0 To dataTableFieldCltn.Count
   List1.AddItem (dataTableField.Name)
   Set dataTableField = dataTableFieldCltn.NextDataTableField
Next i
```

7.20 VcDefinitionField



Ein Objekt vom Typ VcDefinitionField definiert ein Feld der Datendefinitionstabelle. Die Definition besteht im Kern aus einem Namen und einer Festlegung des Datentyps.

Eigenschaften

- DateFormat
- Editable
- Hidden
- ID
- Name
- Type

Eigenschaften

DateFormat

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie das Datumsformat des Feldes in einer Datendefinitionstabelle festlegen oder erfragen. Diese Eigenschaft ist nur wirksam, wenn der Datentyp des Feldes auf **vcDefFieldDateTimeType** eingestellt ist.

Die DateFormat-Einstellung wird beim Lesen und Schreiben von CSV-Dateien verwendet und wenn der Formattyp **String** beim Hinzufügen eines Datensatzes über die Methoden **InsertNodeRecord** oder **InsertLinkRecord** des VcTree-Objektes verwendet wird.

Das Format der Datumsausgabe in der Grafik wird über die Eigenschaft **DateOutputFormat** bei VcTree gesteuert.

Hinweis: Es sollte zuerst die Eigenschaft Type gesetzt werden, bevor die Eigenschaft DateFormat vereinbart wird.

	Datentyp	Beschreibung
Eigenschaftswert	String	Datumsformat {DMYhms:;./} **Standardwert:** bei vcDefFieldDateTime DD.MM.YYYY hh:mm:ss

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Type = vcDefFieldDateTimeType
'DateFormat = "DD.MM.YYYY"
dataDefField.DateFormat = "01.12.2014"
```

Editable

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das spezifizierte Datenfeld zur Laufzeit in der Tabelle (des Diagramms) und des Dialogs **Knoten bearbeiten** editierbar sein soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Definitionsfeld editierbar/nicht editierbar **Standardwert:** True

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Editable = False
```

Hidden

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob ein Datenfeld zur Laufzeit versteckt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Definitionsfeld versteckt/nicht versteckt **Standardwert:** False

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
dataDefField.Hidden = True
```

ID

Nur-Lese-Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie den Index des Feldes einer Datendefinitionstabelle abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Definitionsfeldes

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.FieldByName("Start")
MsgBox dataDefField.ID
```

Name

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie den Namen des Feldes einer Datendefinitionstabelle setzen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Definitionsfeldes

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.CreateDataField("Start")
```

Type

Eigenschaft von VcDefinitionField

Mit dieser Eigenschaft können Sie den Typ des Feldes einer Datendefinitionstabelle erfragen oder setzen.

Hinweis: Durch Setzen der Eigenschaft **Type** wird die Eigenschaft **DateFormat** geändert!

vcDefFieldAlphanumericType: DateFormat = ""

vcDefFieldDateTimeType: DateFormat = "DD.MM.YYYY hh:mm:ss"

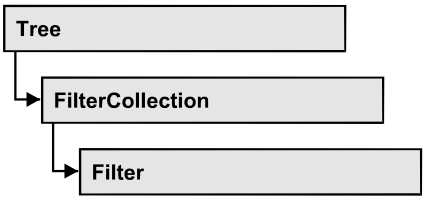
vcDefFieldIntegerType: DateFormat = ""

	Datentyp	**Beschreibung**
Eigenschaftswert	DefinitionFieldTypeEnum	Typ des Definitionsfeldes **Standardwert:** vcDefFieldIntegerType
	Mögliche Werte: vcDefFieldAlphanumericType 1 vcDefFieldDateTimeType 4 vcDefFieldIntegerType 2	 Datentyp **Alphanumerisch**: "" Datentyp **Datum**: DD.MM.YYYY Datentyp **Integer** (32 Bit): ""

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataDefField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataDefField = dataDefTable.CreateDataField("Start")
dataDefField.Type = vcDefFieldDateTimeType
```

7.21 VcFilter



Ein Objekt vom Typ VcFilter enthält Filterbedingungen (VcFilterSubCondition), z.B. zulässige Werte für die Datenfelder eines Knotens. Abhängig von den Daten trifft die Filterbedingung für einen Vorgang zu oder nicht. Filter werden verwendet, um z.B. einem Knoten ein bestimmtes Format zuzuweisen.

Nur wenn der Filter nach Änderungen der Filterbedingungen gültig ist, werden diese wirksam. Andernfalls bleiben die bisherigen Filterbedingungen wirksam (prüfbar über die Methoden VcFilter.IsValid und VcFilterSubCondition.IsValid).

Eigenschaften

- _NewEnum
- DatesWithHourAndMinute
- Name
- Specification
- StringsCaseSensitive
- SubCondition
- SubConditionCount

Methoden

- AddSubCondition
- CopySubCondition
- Evaluate
- IsValid
- RemoveSubCondition

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcFilter

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Filterbedingungsobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim fiSuCo As VcFilterSubCondition

For Each fiSuCo In filter
    Debug.Print fiSuCo.Index
Next
```

DatesWithHourAndMinute

Eigenschaft von VcFilter

Diese Eigenschaft entscheidet, ob beim Vergleich von Datums-Filterbedingungen Stunden und Minuten berücksichtigt werden. Die Einstellung kann nur geändert werden, wenn mindestens eine Unterbedingung mit einem Datumsvergleich vorhanden ist. Ansonsten ist der Eigenschaftswert immer False.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Stunden und Minuten werden berücksichtigt (True)/ nicht berücksichtigt (False)

Name

Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie den Namen des Filters erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Filters

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcTree1.FilterCollection

For Each filter In filterCltn
    ListBox.AddItem filter.name
Next filter
```

Specification

Nur-Lese-Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie die Spezifikation dieses Filters auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung eines Filters mit der Methode **VcFilterCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Filterspezifikation

StringsCaseSensitive

Eigenschaft von VcFilter

Diese Eigenschaft entscheidet, ob bei String-Filterbedingungen der Vergleich mit Unterscheidung von Groß- und Kleinschreibung stattfindet.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Vergleich mit Unterscheidung von Groß- und Kleinschreibung findet statt (True)/findet nicht statt (False)

SubCondition

Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie auf ein VcFilterSubCondition-Objekt per Index zugreifen.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index der Filterbedingung {0 ... VcFilter.SubConditionCount-1}
Eigenschaftswert	VcFilterSubCondition	Filterbedingungsobjekt

SubConditionCount

Nur-Lese-Eigenschaft von VcFilter

Mit dieser Eigenschaft können Sie die Anzahl der Filterbedingungen erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Filterbedingungen

Methoden

AddSubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine neue Filterbedingung an der angegebenen Stelle in der Collection der bestehenden Filterbedingungen erzeugen.

Das entsprechende VcFilterSubCondition-Objekt wird zurückgegeben. Die Eigenschaften dieses Objekt sind standardmäßig folgendermaßen gesetzt:

- DataFieldIndex: -1
- Operator: vcInvalidOp

- ComparisonValueAsString: "<INVALID>"
- ConnectionOperator: vcInvalidConnOp.

	Datentyp	Beschreibung
Parameter:		
⇨ atIndex	Integer	Index der neuen Filterbedingung {0 to VcFilter.SubConditionCount and –1 for "at the end of the Collection" (identical with the value VcFilter.SubConditionCount)}
Rückgabewert	VcFilterSubCondition	Filterbedingungsobjekt

CopySubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine Filterbedingung mit Hilfe der Indexangabe kopieren. Die neue Filterbedingung wird an der angegebenen Stelle in der Collection der bestehenden Filterbedingungen eingefügt und als VcFilterSubCondition-Objekt zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ fromIndex	Integer	Index der zu kopierenden Filterbedingung
⇨ atIndex	Integer	Index der neuen Filterbedingung {0 to VcFilter.SubConditionCount and –1 for "at the end of the Collection" (identical with the value VcFilter.SubConditionCount)}
Rückgabewert	VcFilterSubCondition	Filterbedingungsobjekt

Evaluate

Methode von VcFilter

Mit dieser Methode kann für einen bestimmten Datensatz geprüft werden, ob der gesetzte Filter zutrifft oder nicht. Es sollten sinnvollerweise nur Objekte übergeben werden, die intern mit Datensätzen der Datentabellen verbunden sind. Dies sind: **VcNode, VcLink, VcGroup, VcDataRecord**. Wird ein hier nicht aufgeführter Objekttyp übergeben, wird eine Exception ausgelöst.

	Datentyp	Beschreibung
Parameter:		
⇨ dataObjectParam	Variant	Datensatzobjekt
Rückgabewert	Boolean	Filter trifft für Datensatz zu (True)/nicht zu (False)

IsValid

Methode von VcFilter

Diese Methode prüft, ob alle Filterbedingungen korrekt formuliert sind. Nur wenn das der Fall ist, werden geänderte Filterbedingungen überhaupt wirksam. Andernfalls bleiben die bisherigen Filterbedingungen wirksam.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Filterbedingungen korrekt (True)/ nicht korrekt (False)

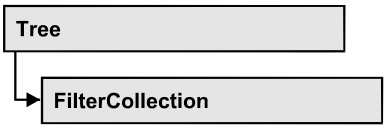
RemoveSubCondition

Methode von VcFilter

Mit dieser Methode können Sie eine Filterbedingung mit Hilfe der Indexangabe löschen.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index der zu löschenden Filterbedingung

7.22 VcFilterCollection



In einem Objekt vom Typ VcFilterCollection sind automatisch alle verfügbaren Filter zusammengefasst. Über **For Each filter In FilterCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Filter zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **FilterByName** und **FilterByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Filter kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Filtern.

Eigenschaften

- _NewEnum
- Count
- MarkedNodesFilter

Methoden

- Add
- AddBySpecification
- Copy
- FilterByIndex
- FilterByName
- FirstFilter
- NextFilter
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcFilterCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Filterobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* ange-

sprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim filter As VcFilter

For Each filter In VcTree1.FilterCollection
   Debug.Print filter.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcFilterCollection

Mit dieser Eigenschaft können Sie die Anzahl der Filterobjekte in der Filter-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim numberOfFilters As Long

Set filterCltn = VcTree1.FilterCollection
numberOfFilters = filterCltn.Count
```

MarkedNodesFilter

Nur-Lese-Eigenschaft von VcFilterCollection

Mit dieser Eigenschaft können Sie einen konstanten Pseudo-Filter holen, der nur bei **ActiveNodeFilter** eingesetzt werden kann und dort das Filtern auf die gerade markierten Knoten auslöst (Teildiagramm).

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Pseudo-Filter

Code-Beispiel

```
Set VcTree1.ActiveNodeFilter = VcTree1.FilterCollection.MarkedNodesFilter
```

Methoden

Add

Methode von VcFilterCollection

Mit dieser Methode können Sie einen neuen Filter in der FilterCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Filterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ newName	String	Name des Filters
Rückgabewert	VcFilter	Neues Filterobjekt

Code-Beispiel

```
Set newFilter = VcTree1.FilterCollection.Add("foo")
```

AddBySpecification

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter über eine Filter-Spezifikation erzeugen. Dies dient der Persistenz von Filterobjekten. Die Spezifikation eines Filters kann erfragt (siehe VcFilter-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann der gleiche Filter mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter:		
⇨ filterSpecification	String	Filterspezifikation
Rückgabewert	VcFilter	Neues Filterobjekt

Copy

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter kopieren. Wenn der Filter mit dem angegebenen Namen existiert und der Name des neuen Filters noch

nicht verwendet wird, wird das neue Filterobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ fromName	String	Name des zu kopierenden Filters
⇨ newName	String	Name des neuen Filters
Rückgabewert	VcFilter	Filterobjekt

FilterByIndex

Methode von VcFilterCollection

Mit dieser Methode können Sie auf einen einzelnen Filter über seinen Index zugreifen. Existiert kein Filter unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Filters
Rückgabewert	VcFilter	Ermitteltes Filterobjekt

FilterByName

Methode von VcFilterCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf einen bestimmten Filter zugreifen. Existiert kein Filter unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ filterName	String	Name des Filters
Rückgabewert	VcFilter	Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcTree1.FilterCollection
Set filter = filterCltn.FilterByName("Department A")
```

FirstFilter

Methode von VcFilterCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Filter der Filter-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFilter** über die nachfolgenden Filter zu iterieren. Existiert kein Filter in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcFilter	Erster Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcTree1.FilterCollection
Set filter = filterCltn.FirstFilter
```

NextFilter

Methode von VcFilterCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Filter der Filter-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFilter** den Initialwert erfasst haben. Sind alle Filter durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcFilter	Nachfolgender Filter

Code-Beispiel

```
Dim filterCltn As VcFilterCollection
Dim filter As VcFilter

Set filterCltn = VcTree1.FilterCollection
Set filter = filterCltn.FirstFilter

While Not filter Is Nothing
    Listbox.AddItem filter.Name
    Set filter = filterCltn.NextFilter
Wend
```

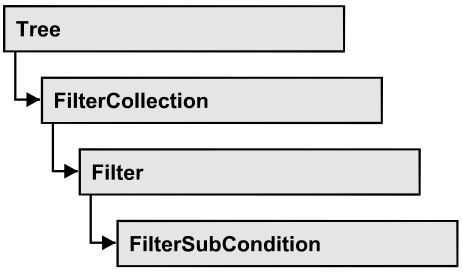
Remove

Methode von VcFilterCollection

Mit dieser Methode können Sie einen Filter löschen. Wenn der Filter noch in irgendeinem anderen Objekt benutzt wird, kann er nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter:		
⇨ name	String	Name des Filters
Rückgabewert	Boolean	Filter gelöscht (True)/nicht gelöscht (False)

7.23 VcFilterSubCondition



Ein Objekt vom Typ VcFilterSubCondition enthält eine einzelne Filterbedingung. Eine Filterbedingung hat im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, der ihre Position im Filter bestimmt.

Im Dialog **Filter bearbeiten** gibt es für jede Filterbedingung eine eigene Zeile. Die dort zur Designzeit dargestellten Eigenschaften sind mit der API hier zur Laufzeit nachträglich veränderbar.

Eigenschaften

- ComparisonValueAsString
- ConnectionOperator
- DataFieldIndex
- FilterName
- Index
- Operator

Methoden

- IsValid

Eigenschaften

ComparisonValueAsString

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Vergleichswert erfragen oder setzen. Dieser String muss einem bestimmten Format entsprechen:

- String: wird in doppelte Anführungszeichen eingeschlossen. Beispiel in VB: """Aachen"""; Beispiel in C/C++: "\"Aachen\""

- Datum: wird in #-Zeichen eingeschlossen. Beispiel: "#18/06/2015;12:34;56;#" (das Datumsformat ist immer "DD.MM.YYYY;hh:mm:ss;", da es sich hierbei um das interne Standardformat, unabhängig vom Betriebssystem und dessen lokalen Einstellungen, handelt). Ein spezieller Datums-Vergleichswert ist "<TODAY>".

- Datenfeld: wird in eckige Klammern eingeschlossen. Beispiel: "[ID]"

- Zahl: wird direkt angegeben. Beispiel: "52076"

- Liste: bei einem der vc...In-Operatoren: wird in geschweifte Klammern eingeschlossen. Die enthaltenen Werte müssen dann alle vom gleichen Typ (String, Datum oder Zahl) sein und können alle obigen Formate besitzen. Beispiel: "{"NETRONIC", [Name]}"

- Ungültig (z. B. nach Neuerzeugen einer Unterbedingung): "<INVALID>"

Der Typ des Vergleichswerts muss dem Datenfeldtyp und dem Typ des Operators entsprechen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Vergleichswert

ConnectionOperator

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Operator für die Verknüpfung mit der folgenden Unterbedingung erfragen oder setzen. Dabei bindet **vcAnd** stärker als **vcOr**.

	Datentyp	Beschreibung
Eigenschaftswert	ConnectionOperatorEnum	Operator für die Verknüpfung mit der folgenden Filterbedingung
	Mögliche Werte:	
	vcAnd 1	Und-Operator
	vcInvalidConnOp 0	ungültiger Operator
	vcOr 2	Oder-Operator

DataFieldIndex

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Index des Datenfeldes, dessen Inhalt verglichen werden soll, erfragen oder setzen. Der Datenfeldtyp muss dem Typ des Vergleichswerts und des Operators entsprechen.

Sonderwert: -1: kein Datenfeld (ungültig)

	Datentyp	Beschreibung
Eigenschaftswert	Long	Index des Datenfeldes, dessen Inhalt verglichen werden soll

FilterName

Nur-Lese-Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Namen des Filters erfragen, zu dem diese Filterbedingung gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Filters

Index

Nur-Lese-Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Index dieser Filterbedingung im zugehörigen Filter erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index der Filterbedingung im zugehörigen Filter

Operator

Eigenschaft von VcFilterSubCondition

Mit dieser Eigenschaft können Sie den Operator für den Vergleich erfragen oder setzen. Die über die API verfügbaren Operatoren entsprechen den Operatoren im Dialog **Filter bearbeiten**. Der Typ des Operators muss dem Datenfeldtyp und des Vergleichswerts entsprechen.

	Datentyp	Beschreibung
Eigenschaftswert	OperatorEnum **Mögliche Werte:** vcDateEarlier 27 vcDateEarlierOrEqual 28 vcDateEqual 25 vcDateIn 31 vcDateLater 29 vcDateLaterOrEqual 30 vcDateNotEqual 26 vcDateNotIn 32 vcIntEqual 9 vcIntGreater 13 vcIntGreaterOrEqual 14 vcIntIn 15 vcIntLess 11 vcIntLessOrEqual 12 vcIntNotEqual 10 vcIntNotIn 16 vcInvalidOp 0 vcStringBeginsWith 3 vcStringContains 5 vcStringEqual 1 vcStringIn 7 vcStringNotBeginsWith 4 vcStringNotContains 6 vcStringNotEqual 2 vcStringNotIn 8	Vergleichsoperator Datum früher als Datum früher als oder gleich Datum gleich Datum in Datum später als Datum später als oder gleich Datum ungleich Datum nicht in Integer gleich Integer größer Integer größer oder gleich Integer in Integer kleiner als Integer kleiner als oder größer Integer ungleich Integer nicht in ungültiger Operator String beginnt mit String enthält String gleich String enthält String beginnt nicht mit String enthält nicht String nicht gleich String nicht enthalten in

Methoden

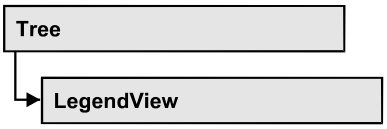
IsValid

Methode von VcFilterSubCondition

Diese Methode prüft, ob die Filterbedingung korrekt formuliert ist.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Filterbedingung korrekt (True)/ nicht korrekt (False)

7.24 VcLegendView



Ein Objekt vom Typ **VcLegendView** bezeichnet das Legendenansicht-Fenster.

Eigenschaften

- Border
- Height
- HeightActualValue
- Left
- LeftActualValue
- ParentHWnd
- ScrollBarMode
- Top
- TopActualValue
- Visible
- Width
- WidthActualValue
- WindowMode

Methoden

- Update

Eigenschaften

Border

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann gesetzt oder erfragt werden, ob die Legendenansicht einen Rahmen besitzt (nicht im Modus **vcPopupWindow**). Die Rahmenfarbe ist **Color.Black**. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rahmen um die Legendenansicht (True)/kein Rahmen um die Legendenansicht (False) **Standardwert:** True

Code-Beispiel

```
VcTree1.LegendView.Mode = vcNotFixed
VcTree1.LegendView.Border = True
```

Height

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die vertikale Ausdehnung der Legendenansicht erfragt werden. In den Modi **vcFixedAtTop**, **vcFixedAtBottom**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe der Legendenansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.Height = 100
```

HeightActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellte vertikale Ausdehnung der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Höhe der Legendenansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.HeightActualValue = 300
```

Left

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die linke Position der Legendenansicht erfragt werden. In den Modi **vcLVNotFixed** und **vcLVPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linke Position der Legendenansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.Left = 200
```

LeftActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellte linke Position der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche linke Position der Legendenansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.LeftActualValue = 150
```

ParentHWnd

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann im Modus **vcLVNotFixed** das HWnd-Handle des Vaterfensters festgelegt werden, wenn die Legendenansicht beispielsweise in einem selbst implementierten Rahmenfenster erscheinen soll. Standardmäßig steht dies auf dem HWnd-Handle des Vaterfensters des VARCHART-ActiveX-Hauptfensters. Diese Eigenschaft kann nur zur Laufzeit verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_HANDLE	Zugriffsnummer

Code-Beispiel

```
MsgBox (VcTree1.legendview.ParentHWnd)
```

ScrollBarMode

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann der Scrollbarmodus der Legendenansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LegendViewScrollBarModeEnum	Scrollbarmodus **Standardwert:** NoScrollBar
	Mögliche Werte:	
	vcAutomaticScrollBar 3	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig.
	vcHorizontalScrollBar 1	Anzeige einer horizontalen Bildlaufleiste, wenn nötig.
	vcNoScrollBar 0	Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt.

vcVerticalScrollBar 2	Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcTree1.LegendView.ScrollBarMode = vcAutomaticScrollBar
```

Top

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die obere Position der Legendenansicht erfragt werden. In den Modi **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	Long	Obere Position der Legendenansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.Top = 20
```

TopActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellte obere Position der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche obere Position der Legendenansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.TopActualValue = 40
```

Visible

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann festgelegt oder erfragt werden, ob die Legendenansicht sichtbar ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Legendenansicht sichtbar (True)/unsichtbar (False) **Standardwert:** False

Code-Beispiel

```
VcTree1.LegendView.Visible = True
```

Width

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die horizontale Ausdehnung der Legendenansicht erfragt werden. In den Modi **vcFixedAtLeft**, **vcFixedAtRight**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann diese Eigenschaft außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Horizontale Ausdehnung der Legendenansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.Width = 200
```

WidthActualValue

Nur-Lese-Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann die tatsächlich dargestellt horizontale Ausdehnung der Legendenansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche horizontale Ausdehnung der Legendenansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.WidthActualValue = 600
```

WindowMode

Eigenschaft von VcLegendView

Mit dieser Eigenschaft kann der Modus der Legendenansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	LegendViewWindowModeEnum	Modus der Legendenansicht **Standardwert:** vcPopupWindow
	Mögliche Werte: vcFixedAtBottom 4	Die Legendenansicht wird unten im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.

vcFixedAtLeft 1	Die Legendenansicht wird links im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
vcFixedAtRight 2	Die Legendenansicht wird rechts im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
vcFixedAtTop 3	Die Legendenansicht wird oben im Fenster des VARCHART ActiveX angezeigt. Dann kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
vcNotFixed 5	Die Legendenansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des VARCHART ActiveX und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft **VcWorldView.ParentHWnd** geändert werden.
vcPopupWindow 6	Die Legendenansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Es kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die **Schließen**-Schaltfläche in der Titelleiste ausgeschaltet werden.

Code-Beispiel

```
VcTree1.LegendView.WindowMode = vcNotFixed
```

Methoden

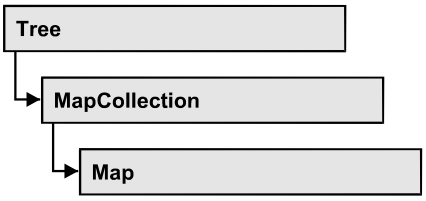
Update

Methode von VcLegendView

Mit dieser Methode wird die Legende aktualisiert.

	Datentyp	Beschreibung

7.25 VcMap



Eine Zuordnungstabelle (Map) legt über Datenfeldeinträge bestimmte Eigenschaften von Knoten, beispielsweise die Hintergrundfarbe, datenfeldabhängig fest.

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Über **For Each mapentry In Map** können Sie in einer Schleife auf alle Einträge der Tabelle zugreifen.

Eigenschaften

- _NewEnum
- ConsiderFilterEntries
- Count
- Name
- Specification
- Type

Methoden

- CreateEntry
- DeleteEntry
- FirstMapEntry
- GetMapEntry
- NextMapEntry

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcMap

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In**

collection angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim map As VcMap

For Each map in VcTree1.Map
   Debug.Print.map.Name
Next
```

ConsiderFilterEntries

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob bei der Zuordnung von Datenfeldeinträgen zu einer Zuordnungstabelle Filter berücksichtigt werden, um so auch Wertebereiche als Schlüsselwerte angeben zu können.

	Datentyp	Beschreibung

Count

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft kann die Anzahl der Einträge in der Zuordnungstabelle (Map) abgefragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Mapeinträge

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim numberOfEntries As Long

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
numberOfEntries = map.count
```

Name

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie den Namen der Zuordnungstabelle (Map) abfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapName As String

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap
mapName = map.Name
```

Specification

Nur-Lese-Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie die Spezifikation dieser Zuordnungstabelle auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Erzeugung einer Zuordnungstabelle mit der Methode **VcMapCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation der Zuordnungstabelle

Type

Eigenschaft von VcMap

Mit dieser Eigenschaft können Sie den Typ der Zuordnungstabelle (Map) abfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	MapTypeEnum	Typ der Zuordnungstabelle
	Mögliche Werte:	
	vcAnyMap 0	**Beliebig** (nur zum Selektieren verwendet)
	vcColorMap 1	**Farben**
	vcFontMap 8	**Schriften**

vcGraphicsFileMap 7	**Grafikdatei**
vcMillimeterMap 9	**Millimeter**
vcNumberMap 10	**Zahlen**
vcPatternMap 3	**Schraffuren**
vcTextMap 6	**Text**

Code-Beispiel

```
Dim mapCollection As VcMapCollection
Dim map As VcMap

Set mapCollection = VcTree1.MapCollection
mapCollection.SelectMaps (vcAnyMap)
Set map = mapCollection.MapByName("Map1")
map.Type = vcPatternMap
```

Methoden

CreateEntry

Methode von VcMap

Mit dieser Methode kann ein neuer Eintrag (= eine neue Zeile) für die Zuordnungstabelle (Map) erzeugt werden. Damit der neue Eintrag in der Zuordnungstabelle wirksam wird, sollte anschließend die Methode **MapCollection.Update()** aufgerufen werden.

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Map-Eintrag

Code-Beispiel

```
Set mapCltn = VcTree1.MapCollection
Set map = mapCltn.Add("MapColor")

map.Type = vcColorMap
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Green"
mapEntry.Color = RGB(0, 255, 0)
Set mapEntry = map.CreateEntry
mapEntry.DataFieldValue = "Red"
mapEntry.Color = RGB(255, 0, 0)
mapCltn.Update
```

DeleteEntry

Methode von VcMap

Mit dieser Methode kann ein Eintrag (= eine Zeile) der Zuordnungstabelle (Map) gelöscht werden. Damit die Löschung in der Zuordnungstabelle

wirksam wird, sollte anschließend die Methode **MapCollection.Update()** aufgerufen werden.

	Datentyp	Beschreibung
Parameter:		
⇨ mapEntry	VcMapEntry	Eintrag der Map
Rückgabewert	Boolean	Map-Eintrag erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

map.DeleteEntry mapEntry
mapCltn.Update
```

FirstMapEntry

Methode von VcMap

Mit dieser Methode können Sie auf den ersten Eintrag der Zuordnungstabelle (Map) zugreifen, um anschließend in einer Schleife mit der Methode **NextMapEntry** über die nachfolgenden Einträge zu iterieren. Existiert kein Eintrag im Map-Objekt, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Erster Map-Eintrag

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)

Set map = mapCltn.FirstMap
Set mapEntry = map.FirstMapEntry
```

GetMapEntry

Methode von VcMap

Diese Methode liefert den entsprechenden Eintrag der Zuordnungstabelle (Map) zu dem im Datenfeld angegebenen Wert.

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Eintrag der Zuordnungstabelle entsprechend Feldinhalt

NextMapEntry

Methode von VcMap

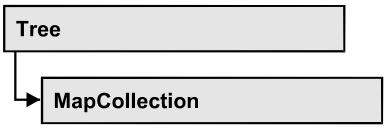
Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Einträge (Zeilen) des Map-Objekts zugreifen, nachdem Sie mit der Methode **FirstMapEntry** den Initialwert erfasst haben. Sind alle Einträge durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMapEntry	Nachfolgender Eintrag der Zuordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)

Set map = mapCltn.FirstMap
Set mapEntry = map.FirstMapEntry

While Not mapEntry Is Nothing
   List1.AddItem (mapEntry.Legend)
   Set mapEntry = map.NextMapEntry
Wend
```

7.26 VcMapCollection



In einem Objekt des Typs VcMapCollection sind die Zuordnungstabellen (Maps) zusammengefasst, die dem Auflistungsobjekt über die Methode **SelectMaps** zugewiesen wurden. Über **For Each map InMapCollection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Zuordnungstabellen zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaften **MapByName** und **MapByIndex**. Die Anzahl der im Auflistungsobjekt vorhandenen Zuordnungstabellen kann über die Eigenschaft **Count** erfragt werden. Die Methoden **Add**, **Copy** und **Remove** ermöglichen das Hinzufügen, Kopieren und Löschen von Zuordnungstabellen.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstMap
- MapByIndex
- MapByName
- NextMap
- Remove
- SelectMaps
- Update

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcMapCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Zuordnungstabellenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim map As VcMap

For Each map In VcTree1.MapCollection
   Debug.Print map.Count
Next
```

Count

Nur-Lese-Eigenschaft von VcMapCollection

Mit dieser Eigenschaft kann die Anzahl der Zuordnungstabellen (Maps) in der Map-Auflistung erfragt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Maps

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim numberOfMaps As Long

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps vcAnyMap
numberOfMaps = mapCltn.Count
```

Methoden

Add

Methode von VcMapCollection

Mit dieser Methode können Sie eine neue Zuordnungstabelle in der MapCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue Zuordnungstabellenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ mapName	String	Name der Zuordnungstabelle
Rückgabewert	VcMap	Neues Zuordnungstabellenobjekt

Code-Beispiel

```
Set newMap = VcTree1.MapCollection.Add("map1")
```

AddBySpecification

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle über eine Zuordnungstabellen-Spezifikation erzeugen. Dies dient der Persistenz von Zuordnungstabellen-Objekten. Die Spezifikation einer Zuordnungstabelle kann erfragt werden (siehe VcMap-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann die gleiche Zuordnungstabelle mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter:		
⇨ Specification	String	Zuordnungstabellenspezifikation
Rückgabewert	VcMap	Neues Zuordnungstabellenobjekt

Copy

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle kopieren. Wenn die Zuordnungstabelle mit dem angegebenen Namen existiert und der Name der neuen Zuordnungstabelle noch nicht verwendet wird, wird das neue Zuordnungstabellenobjekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ mapName	String	Name der zu kopierenden Zuordnungstabelle
⇨ newMapName	String	Name der neuen Zuordnungstabelle
Rückgabewert	VcMap	Zuordnungstabellenobjekt

FirstMap

Methode von VcMapCollection

Mit dieser Methode können Sie auf die erste Zuordnungstabelle (Map) der Map-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextMap** über die nachfolgenden Zuordnungstabellen zu iterieren. Existiert keine Zuordnungstabelle in der Map-Auflistung, wird ein Leerobjekt übergeben (in Visual Basic: **Nothing**). Zuvor muss mit der Methode **SelectMaps** die gewünschte Map-Auswahl getroffen worden sein.

	Datentyp	Beschreibung
Rückgabewert	VcMap	Erste Map

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap
```

MapByIndex

Methode von VcMapCollection

Mit dieser Methode können Sie auf eine einzelne Zuordnungstabelle über ihren Index zugreifen. Existiert keine Zuordnungstabelle unter dem

angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index der Zuordnungstabelle
Rückgabewert	VcMap	Ermitteltes Zuordnungstabellenobjekt

MapByName

Methode von VcMapCollection

Mit dieser Methode können Sie unter Verwendung des Namens der Zuordnungstabelle (Map) auf eine bestimmte Zuordnungstabelle zugreifen. Zuvor muss mit der Methode **SelectMaps** die gewünschte Auswahl der Zuordnungstabelle getroffen worden sein. Existiert kein Map-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ mapName	String	Name der Map
Rückgabewert	VcMap	Map

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.MapByName("Map_1")
```

NextMap

Methode von VcMapCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Zuordnungstabellen (Maps) der Map-Auflistung zugreifen, nachdem Sie mit der Methode **FirstMap** den Initialwert erfasst haben. Sind alle Maps durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcMap	Nachfolgende Zuordnungstabelle

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.FirstMap

While Not map Is Nothing
   List1.AddItem map.Name
   Set map = mapCltn.NextMap
Wend
```

Remove

Methode von VcMapCollection

Mit dieser Methode können Sie eine Zuordnungstabelle löschen. Wenn die Zuordnungstabelle noch in irgendeinem anderen Objekt benutzt wird, kann sie nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter:		
⇨ mapName	String	Name der Zuordnungstabelle
Rückgabewert	Boolean	Zuordnungstabelle gelöscht (True)/nicht gelöscht (False)

SelectMaps

Methode von VcMapCollection

Mit dieser Methode können Sie steuern, welche Typen von Zuordnungstabellen (Maps) in Ihre Map-Auflistung aufgenommen werden.

	Datentyp	Beschreibung
Parameter:		
⇨ selectionType	MapTypeEnum	Auszuwählender Map-Typ
	Mögliche Werte:	
	vcAnyMap 0	**Beliebig** (nur zum Selektieren verwendet)
	vcColorMap 1	**Farben**
	vcFontMap 8	**Schriften**
	vcGraphicsFileMap 7	**Grafikdatei**
	vcMillimeterMap 9	**Millimeter**
	vcNumberMap 10	**Zahlen**

	vcPatternMap 3 vcTextMap 6	**Schraffuren** **Text**
Rückgabewert	Long	Anzahl der Ausgewählten Maps

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps vcAnyMap
```

Update

Methode von VcMapCollection

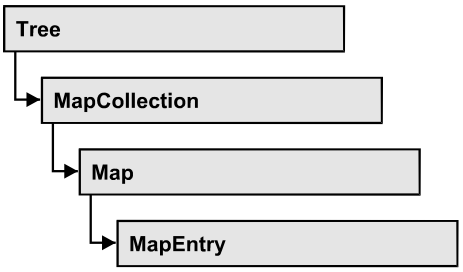
Mit dieser Methode können Sie die Darstellung aller Objekte, die durch die verwendeten Zuordnungstabellen (Maps) bestimmt werden, aktualisieren. Wenn Sie diese Methode nicht aufrufen, werden die Änderungen der Zuordnungstabellen (Maps) zur Laufzeit nicht ausgeführt. Sie sollten diese Methode erst am Ende des Codes zur Festlegung der Zuordnungstabellen und der Map-Auflistung verwenden, damit die Aktualisierung nicht schon ausgeführt wird, bevor alle Festlegungen der Zuordnungstabellen ausgeführt worden sind.

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

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps vcAnyMap
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

While Not mapEntry.DataFieldValue = "A"
   Set mapEntry = map.NextMapEntry
Wend

mapEntry.Color = RGB(0, 0, 0)

mapCltn.Update
```

7.27 VcMapEntry



Ein Objekt vom Typ VcMapEntry ist ein Eintrag einer Zuordnungstabelle (Map) und damit das Element einer Zuordnungstabelle. Ein Zuordnungstabelleneintrag enthält die Kombination aus einem Datenfeldinhalt des Vorgangsdatensatzes sowie bestimmten Attributen (z. B. Farbe, Grafikdatei, Schriftattribute).

Sie können in einer Zuordnungstabelle maximal 150 Zuordnungen festlegen. Falls Sie noch mehr Zuordnungen benötigen, erstellen Sie einfach eine neue Zuordnungstabelle, z. B. als Kopie der bereits vorhandenen.

Eigenschaften

- ColorAsARGB
- DataFieldValue
- FontBody
- FontName
- FontSize
- GraphicsFileName
- Pattern

Eigenschaften

ColorAsARGB

Eigenschaft von VcMapEntry

für Farben-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Farbe für den Zuordnungstabelleneintrag erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll

deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim colorOfMapEntry As OLE_COLOR

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcColorMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

colorOfMapEntry = mapEntry.Color
```

DataFieldValue

Eigenschaft von VcMapEntry

Mit dieser Eigenschaft können Sie den Inhalt des Datenfeldes des Zuordnungstabelleneintrags erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Inhalt des Datenfelds

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim dataFieldValue As String

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcAnyMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

dataFieldValue = mapEntry.DataFieldValue
```

FontBody

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie den Schriftgrad für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	FontBodyEnum	Schriftgrad

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontBodyOfMapEntry As FontBodyEnum

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontBodyOfMapEntry = vcBold
```

FontName

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Schriftart für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Schriftart

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontNameOfMapEntry As String

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontNameOfMapEntry = "Arial"
```

FontSize

Eigenschaft von VcMapEntry

für Schriften-Zuordnungstabellen: Mit dieser Eigenschaft können Sie die Schriftgröße für den Zuordnungstabelleneintrag erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Schriftgröße

Code-Beispiel

```
Dim mapCltn As VcMapCollection
```

```
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim FontSizeOfMapEntry As Long

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcFontMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

FontSizeOfMapEntry = 12
```

GraphicsFileName

Eigenschaft von VcMapEntry

Für Grafikdateien-Zuordnungstabellen: Mit dieser Eigenschaft können Sie den Namen der Grafikdatei des Zuordnungstabelleneintrags erfragen oder festlegen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. WMF mit eingebautem EMF)

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei

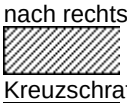
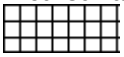
Code-Beispiel

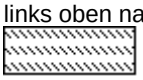
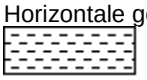
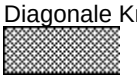
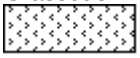
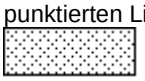
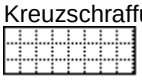
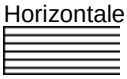
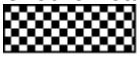
```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry

Set mapCltn = VcTree1.MapCollection
mapCltn.SelectMaps (vcGraphicsFileMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

mapEntry.GraphicsFileName = AppPath & "\picture1.bmp"
```

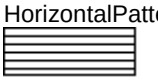
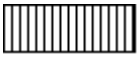
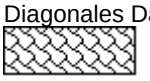
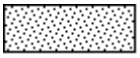
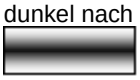
Pattern

Eigenschaft von VcMapEntry

für Schraffuren-Zuordnungstabellen (vcPatternMap): Mit dieser Eigenschaft können Sie den Schraffurtyp des Zuordnungstabelleneintrags erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	FillPatternEnum	Mustertyp
	Mögliche Werte:	
	vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 

vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppeltelter Liniendicke 
vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmall-CheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 

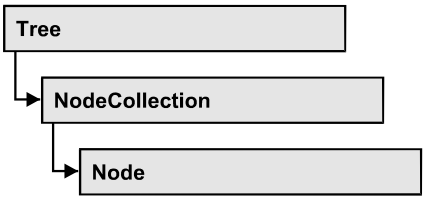
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 
vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vc-HorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vc-HorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 

vcVerticalGradientPattern 62	Vertikaler Farbverlauf 
vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke 
vcZigZagPattern 2030	Horizontale Zickzack-Linien 

Code-Beispiel

```
Dim mapCltn As VcMapCollection
Dim map As VcMap
Dim mapEntry As VcMapEntry
Dim pattern As FillPatternEnum

Set mapCltn = VcTree1.mapCollection
mapCltn.SelectMaps (vcPatternMap)
Set map = mapCltn.MapByName("Map1")
Set mapEntry = map.FirstMapEntry

pattern = vcBDiagonalPattern
```

7.28 VcNode



Knoten sind die Grundelemente eines Baumdiagramms. Das Aussehen eines Knotens wird über diejenigen NodeAppearance-Objekte bestimmt, deren Filter auf den Knoten zutreffen. Erzeugt werden Knoten über die Methode **VcTree.InsertNodeRecord** oder interaktiv.

Eigenschaften

- AllData
- Arrangement
- ChildNodeCollection
- Collapsed
- DataField
- ID
- InCollapsedSubtree
- LeftBrotherNode
- MarkNode
- ParentNode
- RightBrotherNode
- SubtreeNodeCollection

Methoden

- ArrangeSubtree
- Collapse
- DataRecord
- DeleteNode
- Expand
- RelatedDataRecord
- UpdateNode

Eigenschaften

AllData

Eigenschaft von VcNode

Mit dieser Eigenschaft können alle Daten auf einmal für den Knoten gesetzt oder erfragt werden. Beim Setzen ist ein CSV-String (Semikolon als Trennzeichen) oder ein Variant erlaubt, der in einem Feld (Array) alle Datenfelder des Knotens erhält. Beim Erfragen wird ein String zurückgegeben. (Siehe auch **InsertNodeRecord**.)

	Datentyp	Beschreibung
Eigenschaftswert	String/data field	Alle Daten des Datensatzes

Code-Beispiel

```
Private Sub VcTree1_OnNodeModify(ByVal node As VcTreeLib.VcNode,  _
                                 ByVal modificationType As _
                                 VcTreeLib.ModificationTypeEnum, _
                                 returnStatus As Variant)

    Dim allDataOfNode As String

    returnStatus = vcRetStatFalse

    allDataOfNode = node.AllData
    MsgBox allDataOfNode

End Sub
```

Arrangement

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob der dem Knoten anhängende Teilbaum horizontal oder vertikal angeordnet werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	ArrangementEnum	Ausrichtung **Standardwert:** vcHorizontal
	Mögliche Werte: vcHorizontal 0 vcVertical 1	 horizontale Anordnung vertikale Anordnung

Code-Beispiel

```
VcNode.Arrangement = vcHorizontal
```

ChildNodeCollection

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie die direkten Sohnknoten eines Knotens erfragen. Vgl. auch die Eigenschaft **SubtreeNodeCollection**.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	NodeCollection-Objekt, enthält Sohnknoten

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    Dim noOfChildren As Integer

    noOfChildren = node.ChildNodeCollection.Count
    MsgBox (noOfChildren)
    returnStatus = vcRetStatFalse

End Sub
```

Collapsed

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie erfragen, ob ein Knoten kollabiert ist (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knoten kollabiert/expandiert

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    Dim collapseState As Boolean

    collapseState = node.collapsed
    MsgBox (collapseState)
    returnStatus = vcRetStatFalse

End Sub
```

DataField

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie einem Datenfeld des Knotens einen Wert zuweisen oder einen gesetzten Wert erfragen. Wenn ein Knoten durch diese Methode einen neuen Wert erhalten hat, muss anschließend die Methode **UpdateNode** aufgerufen werden.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Datenfeldes
Eigenschaftswert	Variant	Inhalt des Datenfeldes

Code-Beispiel

```
Private Sub VcTree1_OnNodeRClick(ByVal node As VcTreeLib.VcNode, _
                                 ByVal location As VcTreeLib.LocationEnum, _
                                 ByVal x As Long, ByVal y As Long, _
                                 returnStatus As Variant)

   If MsgBox("Delete Node: " & node.dataField(0), vbYesNo, "Delete Node") = _
                                 vbYes Then node.DeleteNode

   returnStatus = vcRetStatNoPopup

End Sub
```

ID

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie die ID eines Knotens erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Knoten-ID

InCollapsedSubtree

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie erfragen, ob sich ein Knoten in einem kollabierten Teilbaum befindet (True) und damit unsichtbar ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knoten liegt/liegt nicht innerhalb eines kollabierten Teilbaumes

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    Dim inCollapsedSubtree As Boolean

    inCollapsedSubtree = node.inCollapsedSubtree
    MsgBox (inCollapsedSubtree)
    returnStatus = vcRetStatFalse

End Sub
```

LeftBrotherNode

Nur-Lese-Eigenschaft von VcNode

Der linke Bruder des Knotens wird zurückgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	VcNode	linker Bruderknoten

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)
    If node.LeftBrotherNode Is Nothing Then
        MsgBox "This node doesn´t have a left brother."
    Else
        MsgBox (node.LeftBrotherNode.AllData)
    End If
    returnStatus = vcRetStatFalse
End Sub
```

MarkNode

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob ein Knoten markiert ist. Die gesetzte Markierung ist nur dann sichtbar, wenn auf der Eigenschaftenseite **Knoten** unter **Knotenmarkierung** nicht **Ohne** ausgewählt ist.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Knoten markiert/nicht markiert

Code-Beispiel

```
Private Sub VcNet1_OnNodeRClick(ByVal node As VcTreeLib.VcNode, _
                                ByVal location As VcTreeLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
```

```
                                returnStatus As Variant)

    Dim nodeMarked As Boolean

    nodeMarked = node.MarkNode
    MsgBox (nodeMarked)
    returnStatus = vcRetStatNoPopup

End Sub
```

ParentNode

Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie einen Knoten als Sohnknoten unter dem angegebenen Vaterknoten einfügen oder den aktuellen Vaterknoten erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	VcNode	Vaterknoten

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                ByVal location As VcTreeLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)

    Dim parentNodeID As Integer

    parentNodeID = node.parentNode.DataField(0)
    MsgBox (parentNodeID)
    returnStatus = vcRetStatFalse

End Sub
```

RightBrotherNode

Nur-Lese-Eigenschaft von VcNode

Der rechte Bruder des Knotens wird zurückgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	VcNode	rechter Bruderknoten

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                            ByVal location As VcTreeLib.LocationEnum, _
                            ByVal x As Long, ByVal y As Long, _
                            returnStatus As Variant)
    If node.RightBrotherNode Is Nothing Then
        MsgBox "This node doesn´t have a right brother."
    Else
        MsgBox (node.RightBrotherNode.AllData)
    End If
    returnStatus = vcRetStatFalse
```

```
End Sub
```

SubtreeNodeCollection

Nur-Lese-Eigenschaft von VcNode

Mit dieser Eigenschaft können Sie den Teilbaum des Referenzknotens (der Knoten selbst sowie alle direkten und indirekten Sohnknoten des Referenzknotens) erfragen. Siehe auch **ChildNodeCollection**.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	Menge der Knoten, die den Teilbaum bilden

Code-Beispiel

```
Private Sub VcTree1_OnNodeRClick(ByVal node As VcTreeLib.VcNode,_
                                 ByVal location As VcTreeLib.LocationEnum, _
                                 ByVal x As Long, ByVal y As Long, _
                                 returnStatus As Variant)

    Dim noOfNodes As Integer

    noOfNodes = node.SubtreeNodeCollection.Count
    MsgBox (noOfNodes)
    returnStatus = vcRetStatNoPopup

End Sub
```

Methoden

ArrangeSubtree

Methode von VcNode

Mit dieser Methode können Sie einen Teilbaum horizontal oder vertikal anordnen. Im Unterschied zur Eigenschaft **Arrangement** wird mit dieser Methode zusätzlich die gleichnamige Eigenschaft aller Knoten im Teilbaum gesetzt.

	Datentyp	Beschreibung
Parameter:		
⇨ arrangement	ArrangementEnum	Ausrichtung
	Mögliche Werte: vcHorizontal 0 vcVertical 1	 horizontale Anordnung vertikale Anordnung
Rückgabewert	Void	

Code-Beispiel

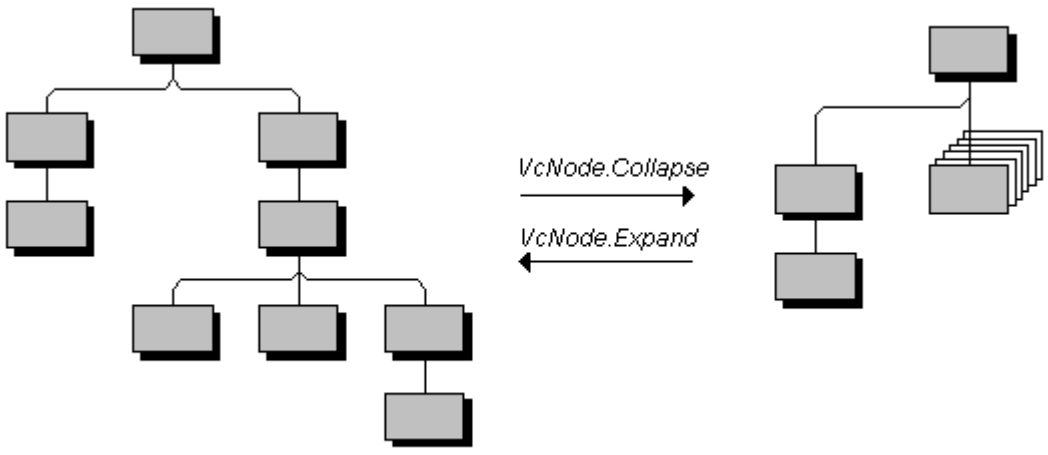
```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcTree1.NodeCollection
Set node = VcTree1.GetNodeByID("8")
node.ArrangeSubtree vcVertical
```

Collapse

Methode von VcNode

Mit dieser Methode können Sie den Knoten einschließlich des darunter hängenden Teilbaums kollabieren. Beim Kollabieren des Teilbaums mit **vcSelf** verschwinden alle Knoten optisch. Beim Kollabieren des Teilbaums mit **vcComplete** verschwinden die Knoten ebenfalls optisch, zusätzlich werden alle die Knoten, die keine Blattknoten sind, in sich kollabiert. Diese müssen dann, falls sie später wieder expandiert werden, einzeln expandiert werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ action	CollapseExpandEnum	Art der Kollabierung
	Mögliche Werte:	
	vcComplete 1	einschließlich der Knoten im untergeordneten Teilbaum
	vcSelf 0	ausschließlich der Knoten im untergeordneten Teilbaum
Rückgabewert	Void	



Kollabieren und Expandieren eines Unterbaumes

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    If MsgBox("Collapse Node No." & node.DataField(0) & "? ", vbYesNo, _
                                    "Collapse node") =  vbYes Then
    node.Collapse vcComplete
    returnStatus = vcRetStatFalse

End Sub
```

DataRecord

Methode von VcNode

Mit dieser Eigenschaft können Sie den Knoten als Datensatzobjekt erfragen. Über die Eigenschaften des Datensatzobjektes haben Sie auch Zugriff auf die entsprechende Datentabelle und TabellenAuflistung.

	Datentyp	Beschreibung
Rückgabewert	VcDataRecord	Zurückgegebener Datensatz

DeleteNode

Methode von VcNode

Mit dieser Methode können Sie einen Knoten löschen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich (true) / nicht erfolgreich (false) gelöscht

Code-Beispiel

```
Private Sub VcTree1_OnNodeRClick(ByVal node As VcTreeLib.VcNode, _
                                  ByVal location As _
                                  VcTreeLib.LocationEnum, ByVal x As Long, _
                                  ByVal y As Long, returnStatus As Variant)

    If MsgBox("Delete Node: " & node.DataField(0), vbYesNo, "Delete Node") = _
              vbYes Then node.DeleteNode
    returnStatus = vcRetStatNoPopup

End Sub
```

Expand

Methode von VcNode

Mit dieser Methode können Sie den Knoten expandieren, ggf. einschließlich seines darunter hängenden Teilbaumes. Bei der Verwendung von **vcSelf** wird nur der Knoten selbst expandiert, aber keine darunter hängenden kollabierten Teilbäume. Bei der Verwendung von **vcComplete** werden alle Knoten des Teilbaums expandiert.

	Datentyp	Beschreibung
Parameter:		
⇨ action	CollapseExpandEnum	Art der Expandierung
	Mögliche Werte:	
	vcComplete 1	einschließlich der Knoten im untergeordneten Teilbaum
	vcSelf 0	ausschließlich der Knoten im untergeordneten Teilbaum
Rückgabewert	Void	

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                            ByVal location As VcTreeLib.LocationEnum, _
                            ByVal x As Long, ByVal y As Long, _
                            returnStatus As Variant)

    If MsgBox("Expand Node No." & node.DataField(0) & "? ", vbYesNo, _
                            "Expand node") =  vbYes Then
    node.Expand vcComplete
    returnStatus = vcRetStatFalse

End Sub
```

RelatedDataRecord

Methode von VcNode

Mit dieser Eigenschaft können Sie einen Datensatz aus einer verknüpften Tabelle erfragen, der dem Datensatz der Knotendatentabelle zugeordnet ist. Der im Parameter übergebene Index bezeichnet das Feld im Datensatz, in dem der Schlüssel des zugeordneten Datensatzes steht.

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Datenfeldes, das den Schlüssel enthält
Rückgabewert	VcDataRecord	Zurückgegebener zugeordneter Datensatz

UpdateNode

Methode von VcNode

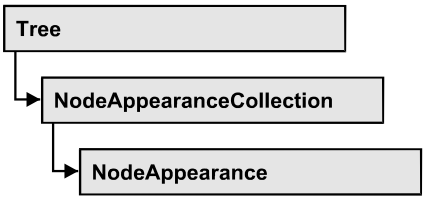
Nachdem Sie ein oder mehrere Datenfelder eines Knotens mit der Eigenschaft **DataField** verändert haben, aktualisieren Sie die Grafik mit **UpdateNode**.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich (true) / nicht erfolgreich (false) aktualisiert

Code-Beispiel

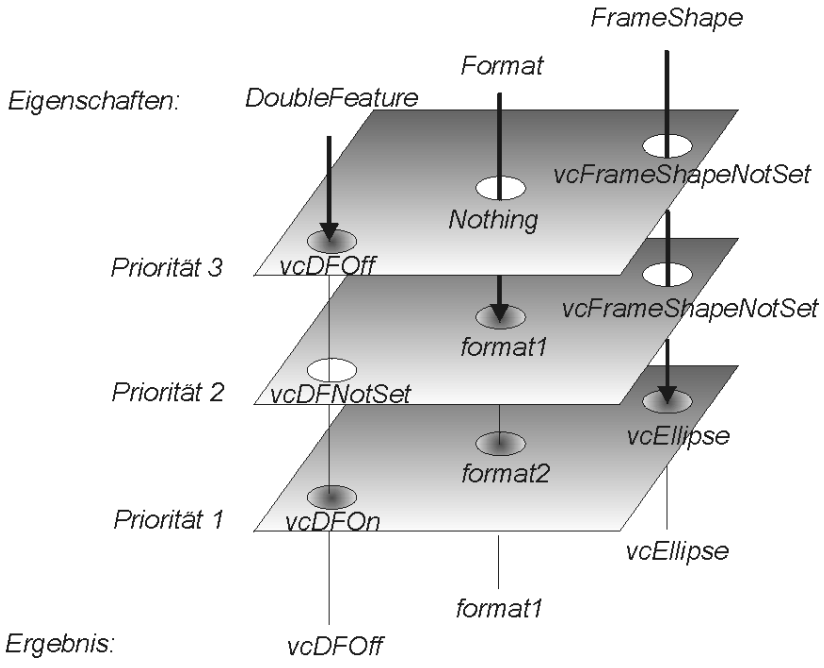
```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcTree1.NodeCollection
Set node = nodeCltn.FirstNode

node.DataField(12) = "Group A"
node.UpdateNode
```

7.29 VcNodeAppearance



Ein NodeAppearance-Objekt bestimmt das Aussehen aller Knoten, deren Daten die dem NodeAppearance-Objekt zugeordneten Filterbedingungen erfüllen. Verschiedene NodeAppearance-Objekte können im Dialogfeld **Knotenaussehen verwalten**, das Sie über die Eigenschaftenseite **Knoten** erreichen, voreingestellt werden.

Die Skizze zeigt, wie sich die Eigenschaften von NodeAppearance-Objekten auf das Aussehen eines Knotens auswirken. Die auf den Knoten zutreffenden NodeAppearances sind nach Priorität absteigend dargestellt. Nicht gesetzte Eigenschaften bei NodeAppearance-Objekten führen dazu, daß die Eigenschaft des nächsttieferen NodeAppearance-Objektes übernommen wird.



Eigenschaften

- BackColorAsARGB
- BackColorDataFieldIndex
- BackColorMapName
- DoubleFeature
- FilterName
- FormatName

- FrameAroundFieldsVisible
- FrameShape
- LegendText
- LineColor
- LineColorDataFieldIndex
- LineColorMapName
- LineThickness
- LineType
- Name
- Pattern
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternDataFieldIndex
- PatternMapName
- Piles
- Shadow
- ShadowColorAsARGB
- Specification
- StrikeThrough
- StrikeThroughColor
- ThreeDEffect
- VisibleInLegend

Methoden

- PutInOrderAfter

Eigenschaften

BackColorAsARGB

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Hintergrundfarbe eines Knotens einstellen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei

der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},0...255},{0...255},{0...255})

Code-Beispiel

```
Dim nodeAppearanceCltn As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCltn = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCltn.FirstNodeAppearance

nodeAppearance.BackColor = RGB(100, 100, 100)
```

BackColorDataFieldIndex

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der Verbindung mit der Eigenschaft **BackColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

BackColorMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Farbzuordnungstabelle für die Hintergrundfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **BackColorDataFieldIndex -1** angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuordnungstabelle

DoubleFeature

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie eine doppelte Umrahmung eines Knotens einstellen oder erfragen. Bei **vcDFNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcDFNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	AppearanceDoubleFeatureEnum	Verschiedene Formen von Doppellinien
	Mögliche Werte:	
	vcDFNotSet -1	Schalter für **DoubleFeature** nicht gesetzt
	vcDFOff 0	Schalter für **DoubleFeature** aus
	vcDFOn 1	Schalter für **DoubleFeature** an

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.DoubleFeature = vcDFOn
```

FilterName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Filter des NodeAppearance-Objekts setzen oder erfragen. Es gibt Sonderfilter, die unveränderlich sind:

- <ALWAYS>: gilt immer (beim Standard-Aussehen immer gesetzt)
- <NEVER>: gilt niemals

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Filtername

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim filtername As String

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance
```

```
filtername = nodeAppearance.filtername
```

FormatName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Format des NodeAppearance-Objekts setzen oder erfragen. Ist keine Eigenschaft gesetzt, wird die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes übernommen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht leer ist (s. Grafik bei VcNode-Appearance).

	Datentyp	Beschreibung
Eigenschaftswert	String	Name eines Knotenformat-Objekts oder leere Zeichenkette

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim format1 As VcNodeFormat

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

Set format1 = nodeAppearance.format
MsgBox (format1.name)
```

FrameAroundFieldsVisible

Eigenschaft von VcNodeAppearance

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

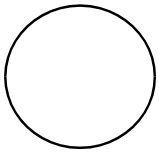
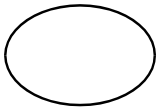
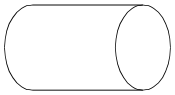
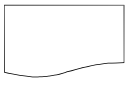
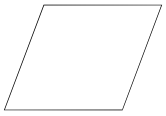
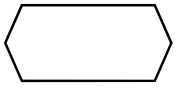
Diese Eigenschaft kann auch im Dialog **Knotenaussehen bearbeiten** gesetzt werden.

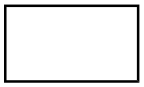
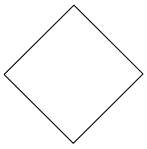
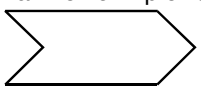
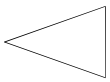
	Datentyp	Beschreibung
Eigenschaftswert	AppearanceFrameAroundFieldsVisibleEnum	Rahmen um Feld
	Mögliche Werte:	
	vcFFVNotSet -1	Feldumrandung nicht gesetzt
	vcFFVOff 0	Schalter für Feldumrandung aus
	vcFFVOn 1	Schalter für **Feldumrandung** an

FrameShape

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt die Rahmenform zuweisen oder erfragen. Bei **vcFrameShapeNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcFrameShapeNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	AppearanceFrameShapeEnum	Rahmenform
	Mögliche Werte:	
	vcCircle 11	Rahmenform kreisförmig 
	vcEllipse 12	Rahmenform elliptisch 
	vcFile 19	Rahmenform liegender Zylinder 
	vcFrameShapeNotSet -1	Rahmenform nicht gesetzt
	vcLeftArrow 17	Rahmenform pfeilförmig, nach links zeigend 
	vcListing 20	Rahmenform Dokument 
	vcNoFrameShape 1	keine Rahmenform
	vcOval 4	Rahmenform oval 
	vcParallelogram 9	Rahmenform Parallelogram 
	vcPointed 7	Rahmenform an den senkrechten Kanten spitz ausgezogen 

vcRectangle 2	Rahmenform rechteckig 
vcRhombus 21	Rahmenform Raute 
vcRightArrow 18	Rahmenform pfeilförmig, nach rechts zeigend 
vcRounded 3	Rahmenform rechteckig gerundet 
vcTriangleLeft 10	Rahmenform dreieckig, Spitze links 
vcTriangleUp 13	Rahmenform dreieckig, Spitze oben 

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.FrameShape = vcEllipse
```

LegendText

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie einem Knotenaussehen einen Text zuweisen oder erfragen, der in der Legende für das jeweilige Knotenaussehen angezeigt wird. Steht hier "", dann wird der Inhalt der Eigenschaft **Name** angezeigt.

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Legendentext des Knotenaussehens **Standardwert:** " " (content of the property **Name**)

LineColor

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt eine Linienfarbe zuweisen oder erfragen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	Color	RGB-Farbwerte oder **-1** ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.LineColor = RGB(256, 0, 100)
```

LineColorDataFieldIndex

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Farbzuordnungstabelle in der Eigenschaft **LineColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	**Beschreibung**
Eigenschaftswert	Integer	Datenfeldindex

LineColorMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Farbzuordnungstabelle für die Linienfarbe setzen oder erfragen. Wird hier "" oder bei der Eigenschaft **LineColorDataFieldIndex -1** angegeben, dann wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuordnungstabelle

LineThickness

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Linienstärke eines NodeAppearance-Objekts erfragen oder festlegen.

Wenn Sie diese Eigenschaft auf Werte zwischen 1 und 4 setzen, wird damit eine absolute Liniendicke in Pixel definiert, d.h. die Linien haben unabhängig vom Zoomfaktor immer die gleiche feste Linienstärke in Pixeln. Dies wird jedoch aufgrund der besseren Lesbarkeit beim Drucken in eine vom Zoomfaktor abhängige Liniendicke umgewandelt:

Wert	Punkte	mm
1	1/2 Punkt	0,09 mm
2	1 Punkt	0,18 mm
3	3/2 Punkt	0,26 mm
4	2 Punkt	0,35 mm

Ein Punkt ist 1/72 Zoll groß und stellt die Maßeinheit für Schriftgrößen dar.

Wenn Sie diese Eigenschaft auf Werte zwischen 5 und 1.000 setzen, wird damit eine Linienstärke in 1/100 mm definiert, d.h. die Linien bekommen eine tatsächliche Dicke in Pixeln, die abhängig vom Zoomfaktor ist.

Wenn Sie diese Eigenschaft auf **-1** setzen, kommt die gleichnamige Eigenschaft des in der Priorität nächst niedrigeren NodeAppearance-Objekts zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linienstärke LineType {1...4}: Werte in Pixeln LineType {5...1000}: Werte in 1/100 mm **Standardwert:** Wie auf der Eigenschaftenseite definiert

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")

nodeAppearance.LineThickness = 3
```

LineType

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt einen Linientyp zuweisen oder erfragen. Bei **vcNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	LineTypeEnum	Linientyp
	Mögliche Werte:	
	vcDashed 4	Linientyp **gestrichelt**
	vcDashedDotted 5	Linientyp **gestrichelt-gepunktet**
	vcDotted 3	Linientyp **gepunktet**
	vcLineType0 100	Linientyp 0
	vcLineType1 101	Linientyp 1
	vcLineType10 110	Linientyp 10
	vcLineType11 111	Linientyp 11
	vcLineType12 112	Linientyp 12
	vcLineType13 113	Linientyp 13
	vcLineType14 114	Linientyp 14
	vcLineType15 115	Linientyp 15
	vcLineType16 116	Linientyp 16

vcLineType17 117	Linientyp 17
vcLineType18 118	Linientyp 18
vcLineType2 102	Linientyp 2
vcLineType3 103	Linientyp 3
vcLineType4 104	Linientyp 4
vcLineType5 105	Linientyp 5
vcLineType6 106	Linientyp 6
vcLineType7 107	Linientyp 7
vcLineType8 108	Linientyp 8
vcLineType9 109	Linientyp 9
vcNone 1	Kein Linientyp
vcNotSet -1	Kein Linientyp zugewiesen
vcSolid 2	Linientyp **durchgezogen**

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.LineType = vcDotted
```

Name

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen eines NodeAppearance-Objekts erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name

Code-Beispiel

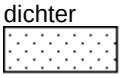
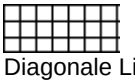
```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim nodeAppName As String

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppName = nodeAppearance.name
```

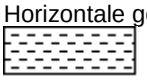
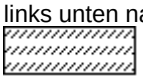
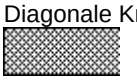
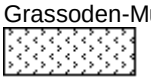
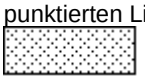
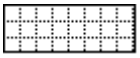
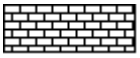
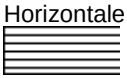
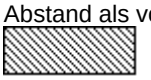
Pattern

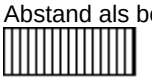
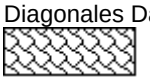
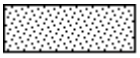
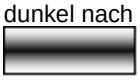
Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Muster des Knotens festlegen oder erfragen. Wenn in der Eigenschaft **PatternMapName** eine Zuordnungstabelle angegeben ist, steuert diese das Muster in Abhängigkeit von den Daten. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

Werte von vc05PercentPattern bis vc90PercentPattern lauten richtigerweise 2001 bis 2011.

	Datentyp	**Beschreibung**
Eigenschaftswert	FillPatternEnum	Mustertyp **Standardwert:** Wie im Dialog definiert
	Mögliche Werte: vc05PercentPattern... vc90PercentPattern 01 - 11	Punkte in Vordergrundfarbe auf Hintergrundfarbe; mit steigender Prozentzahl Vordergrundfarbe immer dichter 
	vcAeroGlassPattern 40	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcBDiagonalPattern 5	Diagonale Linien von links unten nach rechts oben 
	vcCrossPattern 6	Kreuzschraffur 
	vcDarkDownwardDiagonalPattern 2014	Diagonale Linien von links oben nach rechts unten, 50 % näher zusammen als vcFDiagonalPattern und mit doppelter Liniendicke 
	vcDarkHorizontalPattern 2023	Horizontale Linien mit 50% geringerem Abstand als vcHorizontalPattern und doppeltelter Liniendicke 

vcDarkUpwardDiagonalPattern 2015	Diagonale Linien von links unten nach rechts oben mit 50% geringerem Abstand als vcBDiagonalPattern und zweifacher Liniendicke 
vcDarkVerticalPattern 2022	Vertikale Linien mit 50% geringerem Abstand als vcVerticalPattern und doppelter Liniendicke 
vcDashedDownwardDiagonalPattern 2024	Gestrichelte diagonale Linien von links oben nach rechts unten 
vcDashedHorizontalPattern 2026	Horizontale gestrichelte Linien 
vcDashedUpwardDiagonalPattern 2025	Gestrichelte diagonale Linien von links unten nach rechts oben 
vcDashedVerticalPattern 2027	Vertikale gestrichelte Linien 
vcDiagCrossPattern 7	Diagonale Kreuzschraffur, klein 
vcDiagonalBrickPattern 2032	Diagonales Backstein-Muster 
vcDivotPattern 2036	Grassoden-Muster 
vcDottedDiamondPattern 2038	Diagonale Kreuzschraffur aus punktierten Linien 
vcDottedGridPattern 2037	Kreuzschraffur aus punktierten Linien 
vcFDiagonalPattern 4	Diagonale Linien von links oben nach rechts unten 
vcHorizontalBrickPattern 2033	Horizontales Backsteinmuster 
vcHorizontalGradientPattern 52	Horizontaler Farbverlauf 
vcHorizontalPattern 3	Horizontale Linien 
vcLargeCheckerboardPattern 2044	Schachbrettmuster mit doppelt so großen Quadraten wie vcSmall-CheckerBoardPattern 
vcLargeConfettiPattern 2029	Konfetti-Muster, groß 
vcLightDownwardDiagonalPattern 2012	Diagonale Linien von links oben nach rechts unten; mit 50% geringerem Abstand als vcBDiagonalPattern 

vcLightHorizontalPattern 2019	Horizontale Linien mit 50% geringerem Abstand als vc-HorizontalPattern 
vcLightUpwardDiagonalPattern 2013	Diagonale Linien von links unten nach rechts oben, mit 50% geringerem Abstand als vcB-DiagonalPattern 
vcLightVerticalPattern 2018	Vertikale Linien mit 50% geringerem Abstand als bei vcVerticalPattern 
vcNarrowHorizontalPattern 2021	Horizontale Linien mit 75% geringerem Abstand als vc-HorizontalPattern 
vcNarrowVerticalPattern 2020	Vertikale Linien mit 75% geringerem Abstand als bei vcVerticalPattern
vcNoPattern 1276	Kein Füllmuster
vcOutlinedDiamondPattern 2045	Diagonale Kreuzschraffur, groß 
vcPlaidPattern 2035	Schottenstoff-Muster 
vcShinglePattern 2039	Diagonales Dachschindel-Muster 
vcSmallCheckerBoardPattern 2043	Schachbrettmuster 
vcSmallConfettiPattern 2028	Konfetti-Muster 
vcSmallGridPattern 2042	Kreuzschraffur mit 50% geringerem Abstand als vcCrossPattern 
vcSolidDiamondPattern 2046	Schachbrettmuster mit diagonalen Quadraten 
vcSpherePattern 2041	Kugeln schachbrettartig angeordnet 
vcTrellisPattern 2040	Spalier-Muster 
vcVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
vcVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
vcVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 
vcVerticalGradientPattern 62	Vertikaler Farbverlauf 

vcVerticalPattern 2	Vertikale Linien 
vcVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 
vcWavePattern 2031	Horizontales Wellenmuster 
vcWeavePattern 2034	Muster mit verwobenen Streifen 
vcWideDownwardDiagonalPattern 2016	Diagonale Linien von links oben nach rechts unten, mit demselben Abstand wie vcFDiagonalPattern, aber mit dreifacher Liniendicke 
vcWideUpwardDiagonalPattern 2017	Diagonale Linien von links unten nach rechts oben, mit demselben Abstand vcBDiagonalPattern, aber dreifacher Liniendicke 
vcZigZagPattern 2030	Horizontale Zickzack-Linien 

PatternColorAsARGB

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Musterfarbe des Knotens festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

Wenn in der Eigenschaft **PatternColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Musterfarbe datenabhängig.

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},0...255},{0...255},{0...255})

PatternColorDataFieldIndex

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

PatternColorMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Farbzuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColor** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Farbzuordnungstabelle

PatternDataFieldIndex

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

PatternMapName

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den Namen einer Musterzuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn der Name einer Farbzuordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternDataFieldIndex** angegeben sind, wird das Muster des Layers aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **Pattern** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Musterzuordnungstabelle

Piles

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Anzahl von Knotenstapeln im Diagramm erfragen oder festlegen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl gestapelter Knoten oder **-1**

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Piles = 2
```

Shadow

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie dem NodeAppearance-Objekt einen Schatten zuweisen oder erfragen. Bei **vcShNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcShNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	AppearanceShadowEnum	Art der Schattensetzung
	Mögliche Werte:	
	vcShNotSet -1	Schalter für **Schatten** nicht gesetzt
	vcShOff 0	Schalter für **Schatten** aus
	vcShOn 1	Schalter für **Schatten an**

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Shadow = vcShOn
```

ShadowColorAsARGB

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Farbe des Schattens am Knoten einstellen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	Color	ARGB-Farbwerte ({0...255},0...255},{0...255},{0...255}) **Standardwert:** &hFFD8D8D8 (grau)

Code-Beispiel

```
Dim nodeAppearanceCltn As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCltn = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCltn.FirstNodeAppearance

nodeAppearance.ShadowColor = MakeARGB(100, 100, 100, 100)
```

Specification

Nur-Lese-Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Spezifikation dieses Knotenaussehens auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Knotenaussehens mit der Methode **VcNodeAppearanceCollection.AddBySpecification** benutzt werden.

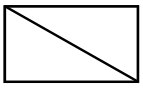
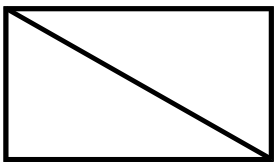
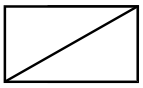
	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Knotenaussehens

StrikeThrough

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie das Durchstreichmuster eines NodeAppearance-Objekts setzen oder erfragen. Bei **vcStrikeThroughNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vcStrikeThroughNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	Beschreibung
Eigenschaftswert	AppearanceStrikeThroughEnum **Mögliche Werte:**	Durchstreichmuster

vcBackslashed 3	Knoten von links oben nach rechts unten durchgestrichen 
vcCrossed 4	Knoten gekreuzt durchgestrichen 
vcDoubleBackslashed 8	Knoten doppelt von links oben nach rechts unten durchgestrichen 
vcDoubleSlashed 7	Knoten doppelt von links unten nach rechts oben durchgestrichen
vcNoStrikeThrough 0	Knoten nicht durchgestrichen
vcSlashed 2	Knoten von links unten nach rechts oben durchgestrichen 
vcStrikeThroughNotSet -1	kein Durchstreichmuster gesetzt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.Strikethrough = vcBackslashed
```

StrikeThroughColor

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie die Farbe des Durchstreichmusters eines NodeAppearance-Objekts setzen oder erfragen. Bei **-1** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **-1** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	Color	RGB-Farbwerte oder **-1** ({0...255},{0...255},{0...255})

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
```

```
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.StrikeThroughColor = RGB(255, 0, 0)
```

ThreeDEffect

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie den 3D-Effekt für das NodeAppearance-Objekt erfragen oder festlegen. Bei **vc3DNotSet** kommt die gleichnamige Eigenschaft des in der Priorität nächstniedrigeren NodeAppearance-Objektes zum Tragen, dessen Filterbedingungen für den Knoten ebenfalls zutreffen und dessen Eigenschaftswert nicht mit **vc3DNotSet** besetzt ist (s. Grafik bei VcNodeAppearance).

	Datentyp	**Beschreibung**
Eigenschaftswert	AppearanceThreeDEffectEnum	Art der 3D-Effekt-Setzung
	Mögliche Werte:	
	vc3DNotSet -1	Schalter für **3D-Effekt nicht gesetzt**
	vc3DOff 0	Schalter für **3D-Effekt aus**
	vc3DOn 1	Schalter für **3D-Effekt an**

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

nodeAppearance.ThreeDEffect = vc3DOn
```

VisibleInLegend

Eigenschaft von VcNodeAppearance

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das NodeAppearance-Objekt in der Legende sichtbar ist. Diese Eigenschaft kann auch im Dialog **Knotenaussehen verwalten** festgelegt werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	Knotenaussehen in Legende sichtbar (True)/nicht sichtbar (False) **Standardwert:** True

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
```

```
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")

nodeAppearance.VisibleInLegend = False
```

Methoden

PutInOrderAfter

Methode von VcNodeAppearance

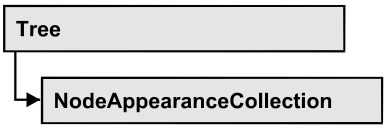
Mit dieser Methode können Sie dieses Knotenaussehen in der Auflistung aller Knotenaussehen hinter das durch den Namen angegebene setzen. Wenn als Name "" angegeben wird, wird das Knotenaussehen an die erste Stelle gesetzt. Die Reihenfolge der Knotenaussehen in der Auflistung entscheidet darüber, in welcher Reihenfolge sie auf die Knoten angewendet werden.

	Datentyp	**Beschreibung**
Parameter: refNodeAppearanceName	String	Name des Knotenaussehens, hinter das das aktuelle Knotenaussehen gesetzt werden soll.
Rückgabewert	Void	

Code-Beispiel

```
Dim nodeAppCltn As VcNodeAppearanceCollection
Dim nodeApp1 As VcNodeAppearance
Dim nodeApp2 As VcNodeAppearance

nodeAppCltn = VcGantt1.NodeAppearanceCollection()
nodeApp1 = nodeAppCltn.Add("nodeApp1")
nodeApp2 = nodeAppCltn.Add("nodeApp2")
nodeApp1.PutInOrderAfter("nodeApp2")
nodeAppCltn.Update()
```

7.30 VcNodeAppearanceCollection



In einem Objekt vom Typ VcNodeAppearanceCollection sind automatisch alle definierten NodeAppearance-Objekte zusammengefasst. Sie haben Zugriff auf ein Knotenaussehen über die Methode **NodeAppearanceBy-Name**. Mit der Eigenschaft **Count** können Sie die Anzahl der Node-Appearance-Objekte ermitteln.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstNodeAppearance
- NextNodeAppearance
- NodeAppearanceByIndex
- NodeAppearanceByName
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNodeAppearanceCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenaussehen-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim nodeApp As VcNodeAppearance

For Each nodeApp In VcTree1.NodeAppearanceCollection
   Debug.Print nodeApp.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcNodeAppearanceCollection

Mit dieser Eigenschaft können Sie die Anzahl der NodeAppearance-Objekte erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der NodeAppearance-Objekte

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance
Dim numberNodeAppColl As Integer

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection

numberNodeAppColl = nodeAppearanceCollectiont.Count
```

Methoden

Add

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein neues Knotenaussehen in der NodeAppearanceCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcNodeAppearance-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB). Bei dem neuen Knotenaussehen sind standardmäßig alle Eigenschaften auf transparent gesetzt.

	Datentyp	Beschreibung
Parameter:		
⇨ newName	String	Name des NodeAppearance-Objekts

Rückgabewert	VcNodeAppearance	Neues NodeAppearance-Objekt

Code-Beispiel

```
Set newNodeAppearance = VcTree1.NodeAppearanceCollection.Add("nodeapp1")
```

AddBySpecification

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein Knotenaussehen über eine Knotenaussehen-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Knotenaussehen-Objekten. Die Spezifikation eines Knotenaussehens kann erfragt (siehe VcNodeAppearance-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Knotenaussehen mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ nodeAppearanceSpecification	String	Knotenaussehen-Spezifikation
Rückgabewert	VcNodeAppearance	Neues Knotenaussehen-Objekt

Copy

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein Knotenaussehen kopieren. Wenn das Knotenaussehen mit dem angegebenen Namen existiert und der Name des neuen Knotenaussehens noch nicht verwendet wird, wird das neue Knotenaussehen-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	**Beschreibung**
Parameter:		
⇨ fromName	String	Name des zu kopierenden Knotenaussehens
⇨ newName	String	Name des neuen Knotenaussehens
Rückgabewert	VcNodeAppearance	NodeAppearance-Objekt

FirstNodeAppearance

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie auf das erste NodeAppearance-Objekt der Auflistung zugreifen, um anschließend mit der Methode **NextNode-Appearance** über die nachfolgenden Objekte zu iterieren. Existiert kein NodeAppearance-Objekt in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeAppearance	Erstes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance
```

NextNodeAppearance

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden NodeAppearance-Objekte der Auflistung zugreifen, nachdem Sie mit der Methode **FirstNodeAppearance** den Initialwert erfasst haben. Sind alle Objekte durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeAppearance	Nachfolgendes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance

While Not nodeAppearance Is Nothing
     Listbox.AddItem nodeAppearance.Name
     Set nodeAppearance = nodeAppearanceCollection.NextNodeAppearance
Wend
```

NodeAppearanceByIndex

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie auf ein einzelnes NodeAppearance-Objekt über seinen Index zugreifen. Existiert kein NodeAppearance-Objekt unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ index	Integer	Index des Knotenaussehens
Rückgabewert	VcNodeAppearance	Ermitteltes NodeAppearance-Objekt

Code-Beispiel

```
Dim NodeAppearanceCltn As VcNodeAppearanceCollection

Set nodeAppearanceCltn = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCltn.NodeAppearanceByIndex(2)
nodeApparance.LineThickness = 2
```

NodeAppearanceByName

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein NodeAppearance-Objekt über den Namen erfragen. Existiert kein NodeAppearance-Objekt unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ nodeAppearanceName	String	Name des NodeAppearance-Objekts
Rückgabewert	VcNodeAppearance	Zurückgegebenes NodeAppearance-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.NodeAppearanceByName("Standard")
```

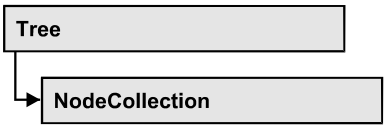
Remove

Methode von VcNodeAppearanceCollection

Mit dieser Methode können Sie ein NodeAppearance-Objekt löschen. Wenn das Objekt noch irgendwo verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird **False** zurückgegeben, sonst **True**.

	Datentyp	**Beschreibung**
Parameter:		
⇨ name	String	Name des Knotenaussehens
Rückgabewert	Boolean	Knotenaussehen gelöscht (True)/nicht gelöscht (False)

7.31 VcNodeCollection



Ein Objekt vom Typ VcNodeCollection beinhaltet alle im Diagramm vorhandenen Knoten. Mit der Methode **SelectNodes** können Sie eine Untermenge dieser Knoten selektieren. Über **For Each node InNode-Collection** oder die Methoden **First...** und **Next...** können Sie in einer Schleife auf alle Knoten zugreifen. Die Anzahl der im Auflistungsobjekt vorhandenen Knoten kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- _NewEnum
- Count

Methoden

- FirstNode
- NextNode
- SelectNodes

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNodeCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenobjekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim node As VcNode

For Each node In VcTree1.NodeCollection
   Debug.Print node.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcNodeCollection

Mit dieser Eigenschaft können Sie die Anzahl der Knoten in der KnotenAuflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl Knoten im NodeCollection-Objekt

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection

Set nodeCltn = VcTree1.NodeCollection
MsgBox "Number of nodes: " & nodeCltn.Count
```

Methoden

FirstNode

Methode von VcNodeCollection

Mit dieser Methode können Sie auf den Initialwert, d. h. den ersten Knoten der Knoten-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextNode** über die nachfolgenden Knoten zu iterieren. Existiert kein Knoten in der Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNode	Erster Knoten

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcTree1.NodeCollection
Set node = nodeCltn.FirstNode
```

NextNode

Methode von VcNodeCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Knoten der Node-Auflistung zugreifen, nachdem Sie mit der Methode **FirstNode** den Initialwert erfasst haben. Sind alle Knoten durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	**Beschreibung**
Rückgabewert	VcNode	Nachfolgender Knoten

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcTree1.NodeCollection
Set node = nodeCltn.FirstNode

While Not node Is Nothing
     node.MarkNode = False
     Set node = nodeCltn.NextNode
Wend
```

SelectNodes

Methode von VcNodeCollection

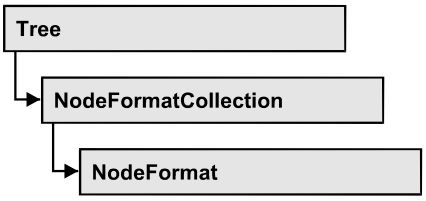
Mit dieser Methode können Sie steuern, welche Knoten in die Knotenauflistung aufgenommen werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ selType	SelectionTypeEnum	auszuwählende Knoten
	Mögliche Werte: vcAll 0 vcAllVisible 1 vcMarked 2	 Alle Objekte im Diagramm werden ausgewählt. Alle sichtbaren Objekte werden ausgewählt. Alle markierten Objekte werden ausgewählt.
Rückgabewert	Long	Anzahl ausgewählter Knoten

Code-Beispiel

```
Dim nodeCltn As VcNodeCollection
Dim node As VcNode

Set nodeCltn = VcTree1.NodeCollection
nodeCltn.SelectNodes vcSelected
```

7.32 VcNodeFormat



Ein Objekt vom Typ VcNodeFormat legt Inhalt und Erscheinungsbild eines Knotens fest. Knotenformate werden zur Designzeit im Dialogfeld **Knotenformate verwalten**, das Sie über die Eigenschaftenseite **Knoten** erreichen, verwaltet und bearbeitet.

Eigenschaften

- _NewEnum
- FieldsSeparatedByLines
- FormatField
- FormatFieldCount
- Name
- Specification
- WidthOfExteriorSurrounding

Methoden

- CopyFormatField
- RemoveFormatField

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNodeFormat

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenformatfeld-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode **GetEnumerator** angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim formatField As VcNodeFormatField

For Each formatField In format
    Debug.Print formatField.Index
Next
```

FieldsSeparatedByLines

Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie festlegen, ob innenliegende Felder durch sichtbare Linien getrennt werden (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	innenliegende Felder durch sichtbare Linien getrennt (True)/ nicht getrennt (False)

Code-Beispiel

```
Dim format As VcNodeFormat

Set format = VcTree1.NodeFormatCollection.FormatByName("format1")
format.FieldsSeparatedByLines = True
```

FormatField

Nur-Lese-Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie ein VcNodeFormatField-Objekt per Index holen. Der Index muss im Bereich von 0 bis .FormatFieldCount-1 liegen.

Hinweis für Benutzer einer Version vor 3.0: Der Index zählt bei dieser Methode nicht wie in den bisherigen Feldeigenschaften von 1 bis .FormatFieldCount!

	Datentyp	Beschreibung
Parameter:		
index	Integer	Index des Knotenformatfeldes 0 ... .FormatFieldCount-1
Eigenschaftswert	VcNodeFormatField	Knotenformatfeld

FormatFieldCount

Nur-Lese-Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Anzahl der Felder eines Knotenformats ermitteln.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Anzahl der Felder im Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat
Dim nameofFormat As String

Set formatCollection = VcTree1.NodeFormatCollection
Set format = formatCollection.FormatByName("Standard")

numberofFormatField = format.FormatFieldCount
```

Name

Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie den Namen des Knotenformats erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Knotenformatname

Code-Beispiel

```
Dim format As VcNodeFormat
Dim formatName As String

Set format = VcTree1.NodeFormatCollection.FirstFormat
formatName = format.Name
```

Specification

Nur-Lese-Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Spezifikation dieses Knotenformats auslesen. Die Spezifikation ist ein String, der nur lesbare ASCII-Zeichen im Bereich 32 bis 127 enthält und somit problemlos in Textdateien oder Datenbanken gespeichert werden kann. Dies ermöglicht Persistenz. Eine solche Spezifikation kann später zur Wiederherstellung eines Knotenformats mit der Methode **VcNodeFormatCollection.AddBySpecification** benutzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Spezifikation des Knotenformats

WidthOfExteriorSurrounding

Eigenschaft von VcNodeFormat

Mit dieser Eigenschaft können Sie die Breite (in mm) des Außenbereichs des Knotenfeldes setzen, d. h. den Abstand in Millimetern, den Knoten mit diesem Knotenformat zu benachbarten Knoten und zum Rand der Darstellung halten sollen. Standardmäßig beträgt die Breite des Außenbereichs 3 mm. Bei kleineren Werten kann es gelegentlich zu Überlagerungen von grafischen Elementen kommen. Daher sollten Sie den Standardwert nur in begründeten Fällen unterschreiten.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des Außenbereichs des Knotenfeldes (in mm) 0 ... 9

Methoden

CopyFormatField

Methode von VcNodeFormat

Mit dieser Methode können Sie ein Knotenformatfeld kopieren. Das neue VcNodeFormatField-Objekt wird zurückgegeben. Es erhält den nächsten, noch nicht vergebenen Index.

	Datentyp	Beschreibung
Parameter:		
⇨ position	FormatFieldPositionEnum	Position des neuen Knotenformatfeldes
	Mögliche Werte:	
	vcAbove 1	oberhalb
	vcBelow 3	unterhalb
	vcLeftOf 0	links von
	vcOutsideAbove 9	außerhalb, oberhalb
	vcOutsideBelow 11	außerhalb, unterhalb
	vcOutsideLeftOf 8	außerhalb, links von
	vcOutsideRightOf 12	außerhalb, rechts von
	vcRightOf 4	rechts von

⇨ refIndex	Integer	Index des Referenz-Knotenformatfeldes
Rückgabewert	VcNodeFormatField	Knotenformatfeld-Objekt

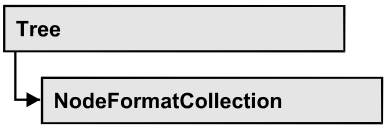
RemoveFormatField

Methode von VcNodeFormat

Mit dieser Methode können Sie ein Knotenformatfeld über den angegebenen Index löschen. Anschließend wird ggf. der Index aller Knotenformatfelder neu festgesetzt, so dass sie wieder fortlaufend numeriert sind.

	Datentyp	**Beschreibung**
Parameter:		
⇨ index	Integer	Index des zu löschenden Knotenformatfeldes

7.33 VcNodeFormatCollection



In einem Objekt vom Typ VcNodeFormatCollection sind alle verfügbaren Knotenformate zusammengefasst. Über **For Each nodeFormat InNode-FormatCollection** oder die Methoden **First…** und **Next…** können Sie in einer Schleife auf alle Knotenformate zugreifen. Sie haben Zugriff auf bestimmte Objekte über die Eigenschaft **FormatByName**. Die Anzahl der im Auflistungsobjekt vorhandenen Knotenformate kann über die Eigenschaft **Count** erfragt werden.

Eigenschaften

- _NewEnum
- Count

Methoden

- Add
- AddBySpecification
- Copy
- FirstFormat
- FormatByIndex
- FormatByName
- NextFormat
- Remove

Eigenschaften

_NewEnum

Nur-Lese-Eigenschaft von VcNodeFormatCollection

Diese Eigenschaft gibt ein Enumerator-Objekt zurück, das das OLE-Interface IEnumVariant implementiert. Mittels dieses Objekts kann man über alle enthaltenen Knotenformat-Objekte iterieren. In Visual Basic wird diese Eigenschaft nie angezeigt, sondern über den Befehl **For Each** *element* **In** *collection* angesprochen. In .NET-Sprachen wird stattdessen die Methode

GetEnumerator angeboten. Einige Entwicklungsumgebungen ersetzen diese Eigenschaft durch eigene Sprachkonstrukte.

	Datentyp	Beschreibung
Eigenschaftswert	Object	Referenzobjekt

Code-Beispiel

```
Dim format As VcNodeFormat

For Each format In VcTree1.NodeFormatCollection
   Debug.Print format.Name
Next
```

Count

Nur-Lese-Eigenschaft von VcNodeFormatCollection

Mit dieser Eigenschaft können Sie die Anzahl der Knotenformatobjekte in der NodeFormat-Auflistung erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Anzahl der Knotenformate

Code-Beispiel

```
Dim formatCltn As VcNodeFormatCollection
Dim numberOfFormats As Long

Set formatCltn = VcTree1.NodeFormatCollection
numberOfFormats = formatCltn.Count
```

Methoden

Add

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein neues Knotenformat in der NodeFormatCollection anlegen. Wenn der Name noch nicht verwendet wird, dann wird das neue VcNodeFormat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

Das Knotenformat besitzt standardmäßig folgende Eigenschaften:

- ein einziges Feld

- WidthOfExteriorSurrounding: 3 mm

Das Feld hat folgende Eigenschaften:

- Type: vcFFTText
- TextDataFieldIndex: in der Eigenschaftenseite **Allgemeines** festgelegte IDMinimumWidth: 3000
- Alignment: vcFFACenter
- BackColor: -1 (transparent)
- TextFontColor: RGB(0,0,0) (schwarz)
- TextFont: Arial, 10, normal
- LeftMargin, RightMargin, TopMargin, BottomMargin: 0,3 mm
- MinimumTextLineCount, MaximumTextLineCount: 1

	Datentyp	**Beschreibung**
Parameter:		
⇨ newName	String	Name des Knotenformats
Rückgabewert	VcNodeFormat	Knotenformat-Objekt

Code-Beispiel

```
Set newNodeFormat = VcTree1.NodeFormatCollection.Add("nodeformat1")
```

AddBySpecification

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat über eine Knotenformat-Spezifikation erzeugen. Dies ermöglicht die Persistenz von Knotenformat-Objekten. Die Spezifikation eines Knotenformats kann erfragt (siehe VcNodeFormat-Eigenschaft **Specification**) und gespeichert werden. Bei einer neuen Sitzung kann das gleiche Knotenformat mit der wieder eingelesenen Spezifikation samt des gespeicherten Namens wieder erzeugt werden.

	Datentyp	Beschreibung
Parameter:		
⇨ formatSpecification	String	Knotenformat-Spezifikation
Rückgabewert	VcNodeFormat	Neues Knotenformat-Objekt

Copy

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat kopieren. Wenn das Knotenformat mit dem angegebenen Namen existiert und der Name des neuen Knotenformats noch nicht verwendet wird, wird das neue Knotenformat-Objekt zurückgegeben, sonst "Nothing" (Visual Basic) oder "0" (andere Sprachen als VB).

	Datentyp	Beschreibung
Parameter:		
⇨ fromName	String	Name des zu kopierenden Knotenformats
⇨ newName	String	Name des neuen Knotenformats
Rückgabewert	VcNodeFormat	NodeFormat-Objekt

FirstFormat

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie auf das erste Knotenformat der NodeFormat-Auflistung zugreifen, um anschließend in einer Schleife mit der Methode **NextFormat** über die nachfolgenden Knotenformate zu iterieren. Existiert kein Knotenformat in der NodeFormat-Auflistung, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeFormat	Erstes Knotenformat

Code-Beispiel

```
Dim format As VcNodeFormat

Set format = VcTree1.NodeFormatCollection.FirstFormat
```

FormatByIndex

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie auf ein einzelnes Knotenformat über ihren Index zugreifen. Existiert kein Knotenformat unter dem angegebenen Index, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	Integer	Index des Knotenformats

Code-Beispiel

```
Dim nodeFormatCltn As VcNodeFormatCollection

Set nodeFormatCltn = VcTree1.NodeFormatCollection
Set nodeFormat = nodeFormatCltn.NodeFormatByIndex(2)
nodeFormat.WidthOfExteriorSurrounding = 2
```

FormatByName

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie unter Verwendung des Namens auf ein bestimmtes Knotenformat zugreifen. Existiert kein Knotenformat unter dem angegebenen Namen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Parameter:		
⇨ formatName	String	Name des Knotenformats
Rückgabewert	VcNodeFormat	Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat

Set formatCollection = VcTree1.NodeFormatCollection
Set format = formatCollection.FormatByName("Standard")
```

NextFormat

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie in einer Schleife auf die nachfolgenden Knotenformate der NodeFormat-Auflistung zugreifen, nachdem Sie mit der Methode **FirstFormat** den Initialwert erfasst haben. Sind alle Formate durchlaufen, wird ein Leerobjekt zurückgegeben (in Visual Basic: **Nothing**).

	Datentyp	Beschreibung
Rückgabewert	VcNodeFormat	Nachfolgendes Knotenformat

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection
Dim format As VcNodeFormat

Set formatCollection = VcTree1.NodeFormatCollection
Set format = formatCollection.FirstFormat

While Not format Is Nothing
     List1.AddItem format.Name
     Set format = formatCollection.NextFormat
Wend
```

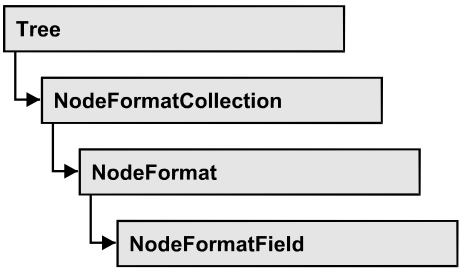
Remove

Methode von VcNodeFormatCollection

Mit dieser Methode können Sie ein Knotenformat löschen. Wenn das Knotenformat noch irgendwo verwendet wird, kann es nicht gelöscht werden. In diesem Fall wird False zurückgegeben, sonst True.

	Datentyp	Beschreibung
Parameter:		
⇨ name	String	Name des Knotenformats
Rückgabewert	Boolean	Knotenformat gelöscht (True)/nicht gelöscht (False)

7.34 VcNodeFormatField



Ein Objekt vom Typ VcNodeFormatField stellt ein Knotenformatfeld, also ein Feld eines VcNodeFormat-Objekts dar. Ein Knotenformatfeld besitzt im Gegensatz zu vielen anderen Objekten keinen Namen, sondern nur einen Index, unter dem es im Knotenformat untergebracht ist.

Eigenschaften

- Alignment
- BottomMargin
- CombiField
- ConstantText
- FormatName
- GraphicsFileName
- GraphicsFileNameDataFieldIndex
- GraphicsFileNameMapName
- GraphicsHeight
- Index
- LeftMargin
- MaximumTextLineCount
- MinimumTextLineCount
- MinimumWidth
- PatternBackgroundColorAsARGB
- PatternBackgroundColorDataFieldIndex
- PatternBackgroundColorMapName
- PatternColorAsARGB
- PatternColorDataFieldIndex
- PatternColorMapName
- PatternEx
- PatternExDataFieldIndex
- PatternExMapName
- RightMargin
- TextDataFieldIndex

- TextFont
- TextFontColor
- TextFontDataFieldIndex
- TextFontMapName
- TopMargin
- Type

Eigenschaften

Alignment

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Ausrichtung des Inhalts im Knotenformatfeld festlegen oder erfragen.

	Datentyp	**Beschreibung**
Eigenschaftswert	FormatFieldAlignmentEnum **Mögliche Werte:** vcFFABottom 28 vcFFABottomLeft 27 vcFFABottomRight 29 vcFFACenter 25 vcFFALeft 24 vcFFARight 26 vcFFATop 22 vcFFATopLeft 21 vcFFATopRight 23	Ausrichtung des Feldinhalts unten unten links unten rechts unten mittig links rechts oben oben links oben rechts

BottomMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des unteren Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	**Beschreibung**
Eigenschaftswert	Integer	Breite des unteren Randes des Knotenformatfeldes in mm 0 ... 9

CombiField

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Knotenfeld ein Kombifeld ist. (Vgl. Dialog **Knotenformat bearbeiten**).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Kombifeld (True)/ kein Kombifeld (False)

ConstantText

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie einen konstanten Text in dem Knotenformatfeld ausgeben, falls der Typ des Knotenformatfeldes auf **vcFFTText** und falls die Eigenschaft **TextDataFieldIndex** auf **-1** gesetzt wurde.

	Datentyp	Beschreibung
Eigenschaftswert	String	konstanter Text

FormatName

Nur-Lese-Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen des Knotenformats erfragen, zu dem dieses Knotenformatfeld gehört.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Knotenformats

GraphicsFileName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Namen einer Grafikdatei setzen oder erfragen, deren Inhalt in dem Knotenformatfeld ausgegeben wird. Der Name muss eine gültige Grafikdatei bezeichnen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafikdatei

GraphicsFileNameDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Datenfeldindex festlegen oder erfragen, der in der Eigenschaft **GraphicsFileNameMapName** benötigt wird. Beim Wert **-1** wird in dem Knotenformatfeld die Grafikdatei ausgeben, die im entsprechenden Knotenformat angegeben ist. Ist ein gültiger Datenfeldindex angegeben und keine Zuordnungstabelle, dann wird der Grafikdateiname direkt aus dem Inhalt des angegebenen Datenfelds entnommen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes

GraphicsFileNameMapName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** den Namen einer Zuordnungstabelle vom Typ **vcGraphicsFileMap** oder "" setzen oder erfragen. Wenn ein Name und zusätzlich ein Datenfeldindex in der Eigenschaft **GraphicsFileNameDataFieldIndex** angegeben ist, wird eine Grafik aus der Zuordnungstabelle angezeigt. Trifft kein Datenfeldeintrag zu, wird die Grafik aus der Eigenschaft **GraphicsFileName** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Grafik-Zuordnungstabelle

GraphicsHeight

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie beim Typ **vcFFTGraphics** die Höhe der Grafik in dem Knotenformatfeld festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Höhe der Grafik in mm 0 ... 99

Index

Nur-Lese-Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Index des Knotenformatfelds im zugehörigen Knotenformat erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Knotenformatfeldes

LeftMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des linken Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des linken Randes des Knotenformatfeldes in mm 0 ... 9

MaximumTextLineCount

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die maximale Anzahl der Zeilen in dem Knotenformatfeld setzen oder erfragen, falls das Knotenformatfeld vom Typ **vcFFTText** ist. Bitte sehen Sie auch die Eigenschaft **MinimumTextLine-Count**.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	maximale Zeilenzahl 0 ... 9

MinimumTextLineCount

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die minimale Anzahl der Zeilen in dem Knotenformatfeld setzen oder erfragen, falls der Typ des Knotenformatfeldes auf **vcFFTText** gesetzt wurde. Ist in einem Knoten mehr Text vorhanden, als in die minimale Anzahl der Zeilen hineinpasst, wird dieses Feld für diesen Knoten dynamisch bis zur maximalen angegebenen Anzahl der Zeilen ausgedehnt. Bitte sehen Sie auch Eigenschaft **MaximumTextLineCount**. Wenn Sie dieser Eigenschaft einen Wert zuweisen, sollten Sie anschließend auch erneut der Eigenschaft **MaximumTextLineCount** den gewünschten Wert setzen, sonst könnte es vorkommen, dass das Maximum durch das Minimum überschrieben wird.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Zeilenzahl 0 ... 9

MinimumWidth

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die minimale Breite des Knotenformatfeldes in mm festlegen oder erfragen. Die Breite des Feldes kann sich vergrößern, wenn unter oder über dem Feld andere Felder größere minimale Breiten besitzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	minimale Breite des Knotenformatfeldes in mm 0 ... 99

PatternBackgroundColorAsARGB

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Hintergrundfarbe des Knotenformatfeldes festlegen oder erfragen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe

erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

Wählen Sie den Wert **-1**, wenn das Feld die Hintergrundfarbe des Knotenformats besitzen soll.

Wenn in der Eigenschaft **PatternBackgroundColorMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Hintergrundfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Parameter:		
⇨ Rückgabewert	OLE_COLOR	Hintergrundfarbe des Knotenformatfelds
Eigenschaftswert	Long	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

PatternBackgroundColorDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternBackgroundColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Parameter:		
⇨ Rückgabewert	Integer	Datenfeldindex
Eigenschaftswert	Long	Datenfeldindex

PatternBackgroundColorMapName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen einer Farbzuordnungstabelle (Typ vcColorMap) für die Hintergrundfarbe setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn der Name einer Farbzuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **PatternBackgroundColorDataFieldIndex** angegeben ist, wird die Hintergrundfarbe aus der Zuordnungstabelle ausgewählt. Trifft kein

Datenfeldeintrag zu, wird die Hintergrundfarbe aus der Eigenschaft **PatternBackgroundColor** benutzt.

	Datentyp	Beschreibung
Parameter:		
⇨ Rückgabewert	String	Name der Farbzuordnungstabelle
Eigenschaftswert	String	Name der Farbzuordnungstabelle

PatternColorAsARGB

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Musterfarbe des Knotenformatfeldes erfragen oder festlegen. Farbwerte haben einen Transparenz- oder Alphawert, einen Rot-, einen Blau- und einen Grünanteil im Zahlenbereich von 0..255 (ARGB-Wert). Ein Alpha-Wert von 0 bedeutet vollständige Transparenz, während der Wert 255 eine voll deckende Farbe erzeugt. Bei der Umwandlung eines RGB-Wertes in einen ARGB-Wert muss ein Alpha-Wert von 255 hinzugegeben werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	ARGB-Farbwerte ({0...255},{0...255},{0...255},{0...255})

PatternColorDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternColorMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

PatternColorMapName

Eigenschaft von VcNodeFormatField

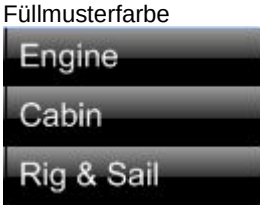
Mit dieser Eigenschaft können Sie den Namen einer Farbzuordnungstabelle (Typ vcColorMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Nur wenn ein Name einer Farbzuordnungstabelle und ein Datenfeldindex in der Eigenschaft **PatternColorDataFieldIndex** angegeben sind, wird die Musterfarbe des Kalendergitters aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Musterfarbe aus der Eigenschaft **PatternColor** ausgegeben.

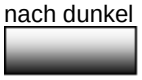
	Datentyp	**Beschreibung**
Eigenschaftswert	String	Name der Farbzuordnungstabelle

PatternEx

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie für den Hintergrund des Knotenformatfeldes ein Muster setzen oder erfragen.

	Datentyp	**Beschreibung**
Eigenschaftswert	FieldFillPatternEnum	Mustertyp **Standardwert:** Wie im Dialog definiert
	Mögliche Werte:	
	vcFieldNoPattern 1276	Kein Füllmuster
	vcAeroGlassPattern 44	Vertikaler Farbverlauf in der Füllmusterfarbe 
	vcFieldVerticalBottomLightedConvexPattern 43	Vertikaler Farbverlauf von dunkel nach hell 
	vcFieldVerticalConcavePattern 40	Vertikaler Farbverlauf von dunkel über hell nach dunkel 
	vcFieldVerticalConvexPattern 41	Vertikaler Farbverlauf von hell über dunkel nach hell 

vcFieldVerticalTopLightedConvexPattern 42	Vertikaler Farbverlauf von hell nach dunkel 	

PatternExDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der in Verbindung mit der Eigenschaft **PatternExMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeldindex

PatternExMapName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen einer Muster-Zuordnungstabelle (Typ vcPatternMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Muster-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **PatternExDataFieldIndex** angegeben ist, wird das Muster aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird das Muster aus der Eigenschaft **PatternEx** ausgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ Rückgabewert	String	Name der Musterzuordnungstabelle
Eigenschaftswert	String	Name der Musterzuordnungstabelle

RightMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des rechten Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des rechten Randes des Knotenformatfeldes in mm 0 ... 9

TextDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Index des Datenfelds, dessen Inhalt in dem Tabellenformatfeld dargestellt werden soll, erfragen oder setzen, sofern es sich um ein Feld des Datentyps **vcFFTText** handelt. Falls der Index **-1** ist, wird stattdessen der Inhalt der Eigenschaft **ConstantText** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Index des Datenfeldes

TextFont

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Schriftart des Knotenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn in der Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben ist, steuert diese die Schriftart in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	StdFont	Schriftart des Knotenformatfelds

TextFontColor

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Schriftfarbe des Knotenformatfeldes festlegen oder erfragen, falls der Typ des Feldes auf **vcFFTText** gesetzt wurde. Wenn über die Eigenschaft **TextFontMapName** eine Zuordnungstabelle angegeben wurde, steuert diese die Schriftfarbe in Abhängigkeit von den Daten.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_COLOR	Schriftfarbe des Knotenformatfelds **Standardwert:** -1

TextFontDataFieldIndex

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Datenfeldindex festlegen oder erfragen, der bei einer Schrift-Zuordnungstabelle in der Eigenschaft **TextFontMapName** benötigt wird. Wenn Sie hier **-1** angegeben, wird keine Zuordnungstabelle verwendet.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Datenfeldindex

TextFontMapName

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Namen einer Schrift-Zuordnungstabelle (Typ vcFontMap) setzen oder erfragen. Wird hier "" angegeben, dann wird keine Zuordnungstabelle verwendet. Wenn ein Name einer Schrift-Zuordnungstabelle und zusätzlich ein Datenfeldindex in der Eigenschaft **TextFontDataFieldIndex** angegeben ist, wird die Schriftart aus der Zuordnungstabelle ausgewählt. Trifft kein Datenfeldeintrag zu, wird die Schriftart aus der Eigenschaft **TextFont** ausgegeben.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Schrift-Zuordnungstabelle

TopMargin

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie die Breite des oberen Randes des Knotenformatfeldes in mm festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Breite des oberen Randes des Knotenformatfeldes in mm 0 ... 9

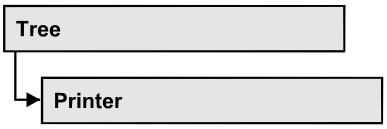
Type

Eigenschaft von VcNodeFormatField

Mit dieser Eigenschaft können Sie den Typ des Knotenformatfelds erfragen.

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

7.35 VcPrinter



Das VcPrinter-Objekt stellt Ihnen Eigenschaften zur Verfügung, die die Seitengestaltung und den Druckvorgang betreffen. Sie können die Randbreiten der Seiten einstellen sowie Seitenrahmen, Seitenzahlen, Seitenbeschriftung, Schnittmarkierungen und Druckdatum setzen. Weiterhin können Sie die Anzahl der Seiten, auf die das Diagram verteilt werden soll, sowie Vergrößerungsfaktor, Druckausrichtung, Hoch- bzw. Querformat, Papierformat und Farbmodus festlegen.

Eigenschaften

- AbsoluteBottomMarginInCM
- AbsoluteBottomMarginInInches
- AbsoluteLeftMarginInCM
- AbsoluteLeftMarginInInches
- AbsoluteRightMarginInCM
- AbsoluteRightMarginInInches
- AbsoluteTopMarginInCM
- AbsoluteTopMarginInInches
- Alignment
- CurrentHorizontalPagesCount
- CurrentVerticalPagesCount
- CurrentZoomFactor
- CuttingMarks
- DefaultPrinterName
- DocumentName
- FitToPage
- FoldingMarksType
- MarginsShownInInches
- MaxHorizontalPagesCount
- MaxVerticalPagesCount
- Orientation
- PageDescription
- PageDescriptionString
- PageFrame
- PageNumberMode

- PageNumbers
- PagePaddingEnabled
- PaperSize
- PrintDate
- PrinterName
- RepeatTitleAndLegend
- StartUpSinglePage
- ZoomFactorAsDouble

Eigenschaften

AbsoluteBottomMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des unteren Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des unteren Seitenrandes in cm

Code-Beispiel

```
VcTree1.Printer.AbsoluteBottomMarginInCM = 1.5
```

AbsoluteBottomMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des unteren Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des unteren Seitenrandes in Zoll

Code-Beispiel

```
VcTree1.Printer.AbsoluteBottomMarginInches = 0.5
```

AbsoluteLeftMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des linken Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des linken Seitenrandes in cm

Code-Beispiel

```
VcTree1.Printer.AbsoluteLeftMarginInCM = 1.5
```

AbsoluteLeftMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des linken Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des linken Seitenrandes in Zoll

Code-Beispiel

```
VcTree1.Printer.AbsoluteLeftMarginInInches = 0.5
```

AbsoluteRightMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des rechten Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des rechten Seitenrandes in cm

Code-Beispiel

```
VcTree1.Printer.AbsoluteRightMarginInCM = 1.5
```

AbsoluteRightMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Breite des rechten Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Breite des rechten Seitenrandes in Zoll

Code-Beispiel

```
VcTree1.Printer.AbsoluteRightMarginInInches = 0.5
```

AbsoluteTopMarginInCM

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des oberen Seitenrandes setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des oberen Seitenrandes in cm

Code-Beispiel

```
VcTree1.Printer.AbsoluteTopMarginInCM = 1.5
```

AbsoluteTopMarginInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die absolute Höhe des oberen Seitenrandes in der Maßeinheit Zoll setzen oder erfragen. Die tatsächliche Breite kann größer sein, wenn der verwendete Drucker nicht randlos drucken kann.

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Höhe des oberen Seitenrandes in Zoll

Code-Beispiel

```
VcTree1.Printer.AbsoluteTopMarginInInches = 0.5
```

Alignment

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die Ausrichtung des Ausdrucks auf einer Seite setzen oder erfragen. Sie hat nur dann eine Auswirkung, wenn die Gesamtgrafik auf einer einzigen Seite dargestellt wird, oder wenn die Eigenschaft **RepeatTitleAndLegend** eingeschaltet ist. In allen anderen Fällen wird die Gesamtgrafik zentriert ausgerichtet.

	Datentyp	Beschreibung
Eigenschaftswert	PrinterAlignmentEnum	Ausrichtung des Ausdrucks auf der Seite **Standardwert:** vcPCenterCenter
	Mögliche Werte:	
	vcPBottomCenter 28	vertikale Ausrichtung: unten, horizontale Ausrichtung: mittig
	vcPBottomLeft 27	vertikale Ausrichtung: unten, horizontale Ausrichtung: links
	vcPBottomRight 29	vertikale Ausrichtung: unten, horizontale Ausrichtung: rechts
	vcPCenterCenter 25	vertikale Ausrichtung: mittig, horizontale Ausrichtung: mittig
	vcPCenterLeft 24	vertikale Ausrichtung: mittig, horizontale Ausrichtung: links
	vcPCenterRight 26	vertikale Ausrichtung: mittig, horizontale Ausrichtung: rechts
	vcPTopCenter 22	vertikale Ausrichtung: oben, horizontale Ausrichtung: mittig
	vcPTopLeft 21	vertikale Ausrichtung: oben, horizontale Ausrichtung: links
	vcPTopRight 23	vertikale Ausrichtung: oben, horizontale Ausrichtung: rechts

Code-Beispiel

```
VcTree1.Printer.Alignment = vcPTopLeft
```

CurrentHorizontalPagesCount

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die tatsächliche Anzahl der Seiten des Ausdrucks in der Breite ermitteln. Siehe auch die Eigenschaften **CurrentVerticalPagesCount** und **MaxHorizontalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Anzahl Seiten in horizontaler Richtung

CurrentVerticalPagesCount

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die tatsächliche Anzahl der Seiten des Ausdrucks in der Höhe ermitteln. Siehe auch die Eigenschaften **CurrentHorizontalPagesCount** und **MaxVerticalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Anzahl Seiten in vertikaler Richtung

CurrentZoomFactor

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den tatsächlichen Zoomfaktor in Prozent für die Einstellung **FitToPage = False** erfragen (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Tatsächlicher Zoomfaktor

CuttingMarks

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Schnittmarkierungen auf eine Seite gedruckt werden sollen (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Schnittmarken werden (True) / werden nicht (False) gedruckt **Standardwert:** False

Code-Beispiel

```
VcTree1.Printer.CuttingMarks = True
```

DefaultPrinterName

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Namen des aktuellen Standard-Systemdruckers erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des aktuellen Standard-Systemdruckers

DocumentName

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Namen des Dokumentes bestimmen oder auslesen. Der Dokumentenname wird beim Drucker in der Liste der zu druckenden Dokumente angezeigt und hat bei speziellen Druckertreibern wie z. B. einigen, die PDF-Dateien erzeugen, besondere Funktionen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Dokumentenname **Standardwert:** " "

FitToPage

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob die über die Eigenschaften **MaxHorizontalPagesCount** und **MaxVerticalPagesCount** definierte Anzahl von Seiten gedruckt werden soll (True) oder ob das Diagramm in der mit der Eigenschaft **ZoomFactor** eingestellten Größe ausgegeben werden soll (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Diagramm wird auf eine definierte Anzahl von Seiten verteilt/wird in der voreingestellten Größe ausgegeben.

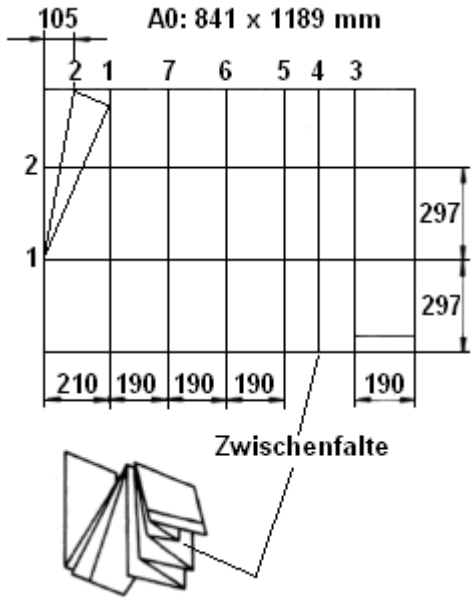
Code-Beispiel

```
VcTree1.Printer.FitToPage = True
```

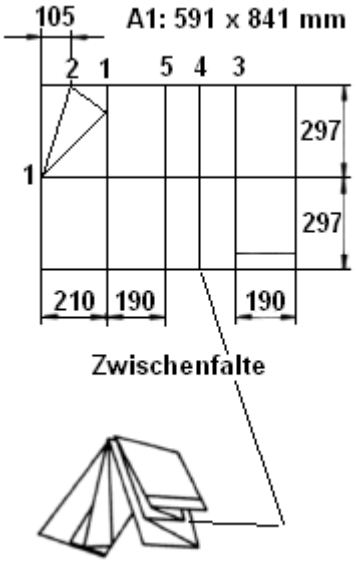
FoldingMarksType

Nur-Lese-Eigenschaft von VcPrinter

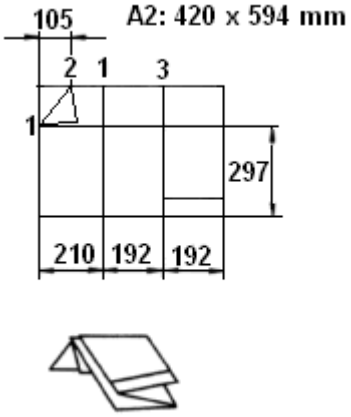
Mit dieser Eigenschaft können Sie folgende Faltmarkierungen nach DIN 824 für den Ausdruck festlegen oder erfragen. Diese ermöglichen das standardisierte Falten für DIN-A-Blattgrößen:



Faltung des DIN-A-0 Formats

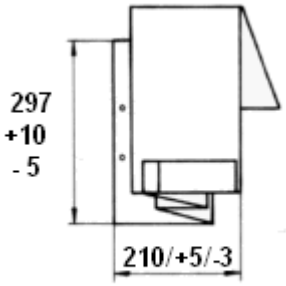
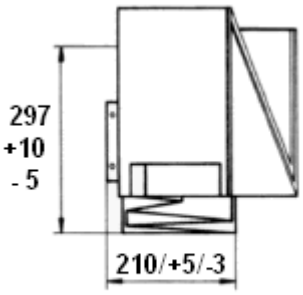
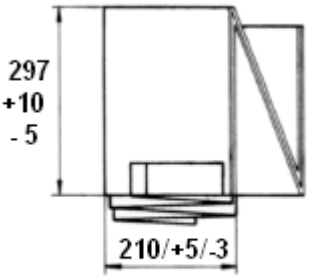


Faltung des DIN-A-1 Formats



Faltung des DIN-A-2 Formats

	Datentyp	**Beschreibung**
Eigenschaftswert	FoldingMarksTypeEnum **Mögliche Werte:**	Faltmarkierungen **Standardwert:** vcFMTNone

vcFMTDIN824FormA 65	Ausgabe von Faltmarkierungen nach DIN824-A: Die gefaltete Zeichnung kann gelocht und ohne Heftstreifen abgeheftet werden.  **Faltung nach DIN 824-A**
vcFMTDIN824FormB 66	Ausgabe von Faltmarkierungen nach DIN824-B: Die gefaltete Zeichnung kann gelocht und mit Heftstreifen abgeheftet werden.  **Faltung nach DIN 824-B**
vcFMTDIN824FormC 67	Ausgabe von Faltmarkierungen nach DIN824-C: Die gefaltete Zeichnung wird nicht gelocht, sondern in eine Sichthülle gelegt.  **Faltung nach DIN 824-C**
vcFMTNone 0	Keine Ausgabe von Faltmarkierungen

MarginsShownInInches

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder abfragen, ob im Dialog **Seite einrichten** die Maßeinheit für Seitenränder in Zoll ein- und ausgegeben wird (gegenwärtig nur zur Laufzeit möglich).

Hinweis: Damit im Dialog **Seite einrichten** glattere Werte erzielt werden, beträgt der interne Umrechnungsfaktor 2,5 cm/Zoll statt der eigentlich korrekten 2,54 cm/Zoll (1,5 cm entsprechen dann 0,6 Zoll, 1 cm 0,4 Zoll).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Maßeinheit der Seitenränder im Dialog **Seite einrichten** in Zoll (True)/ in cm (False)

MaxHorizontalPagesCount

Eigenschaft von VcPrinter

Diese Eigenschaft dient der Festlegung oder Erfragung der horizontalen Seitenzahl beim Drucken und für die Druckvorschau. Die Festlegung ist nur wirksam, wenn Sie in der Eigenschaft **ScalingMode** den Wert **vcFitToPageCount** oder **vcZoomWithHorizontalFit** festgelegt haben. S. auch Eigenschaft **MaxVerticalPagesCount** und **CurrentHorizontalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Maximale Anzahl Seiten in horizontaler Richtung **Standardwert:** 1

Code-Beispiel

```
VcTree1.Printer.MaxHorizontalPagesCount = 4
```

MaxVerticalPagesCount

Eigenschaft von VcPrinter

Diese Eigenschaft dient der Festlegung oder Erfragung der vertikalen Seitenzahl beim Drucken und für die Druckvorschau. Diese Festlegung ist nur wirksam, wenn Sie in der Eigenschaft **ScalingMode** den Wert **vcFitToPageCount** festgelegt haben. S. auch Eigenschaft **MaxHorizontalPagesCount** und **CurrentVerticalPagesCount**.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Maximale Anzahl Seiten in vertikaler Richtung **Standardwert:** 1

Code-Beispiel

```
VcTree1.Printer.MaxVerticalPagesCount = 4
```

Orientation

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen, ob die einzelnen Seiten des Ausdrucks im Hoch- oder Querformat verwendet werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	OrientationEnum	Ausrichtung **Standardwert:** VcPortrait
	Mögliche Werte: vcLandscape 42 vcPortrait 41	 Querformat Hochformat

Code-Beispiel

```
VcTree1.Printer.Orientation = vcLandScape
```

PageDescription

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Seitenbeschriftung für die linke untere Ecke jeder Seite erscheinen soll (True) oder nicht (False). Den Inhalt der Seitenbeschriftung legen Sie über die Eigenschaft **PageDescriptionString** fest.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	Seitenbeschriftung wird (True) / wird nicht (False) gedruckt **Standardwert:** False

Code-Beispiel

```
VcTree1.Printer.PageDescription = True
```

PageDescriptionString

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie eine Seitenbeschriftung für die linke untere Ecke jeder Seite festlegen oder erfragen. Die Ausgabe der Seitenbeschriftung können Sie mit der Eigenschaft **PageDescription** steuern. Für die Seitennummerierung können Sie folgende Platzhalter angeben, die dann beim Ausdruck durch die entsprechenden Inhalte ersetzt werden:

{PAGE} = fortlaufende Seitennummer

{NUMPAGES} = Gesamtanzahl der Seiten

{ROW} = Zeilenposition des Ausschnitts im Gesamtdiagramm

{COLUMN} = Spaltenposition des Ausschnitts im Gesamtdiagramm

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Seitenbeschriftung **Standardwert:** Leere Zeichenkette ""

Code-Beispiel

```
VcTree1.Printer.PageDescriptionString = "VARCHART chart"
```

PageFrame

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob um den Ausdruck ein Rahmen gezogen werden soll (True) oder nicht (False). Wenn die Eigenschaft **RepeatTableTimeScale** eingeschaltet ist, so wird der Rahmen um jede Einzelseite gezogen, anderenfalls wird er um die gesamte Grafik gezogen.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	Seitenrahmen wird dargestellt (True) / wirde nicht (False) dargestellt **Standardwert:** True

Code-Beispiel

```
VcTree1.Printer.PageFrame = True
```

PageNumberMode

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, wie die Seitennummerierung ausgegeben werden soll: "Seite N von M Seiten" oder "x.y" (Zeilennummer/Spaltennummer).

	Datentyp	**Beschreibung**
Eigenschaftswert	pageNumberModeEnum	Art der Seitennumerierung **Standardwert:** vcPRowColumn
	Mögliche Werte: vcPageNOfM 1597	"Seite N von M Seiten"

vcPRowColumn 1596	"x.y" (Zeilennummer.Spaltennummer)

Code-Beispiel

```
Dim printer As VcPrinter

Set printer = VcTree1.printer

With printer
    .Orientation = vcLandscape
    .PageNumberMode = vcPageNOfM
    .PageNumbers = True
    .FitToPage = False
End With

VcTree1.PrintPreview
```

PageNumbers

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob die Seitenzahl in der linken unteren Ecke einer Seite erscheinen soll (True) oder nicht (False). Die Art der Nummerierung können Sie mit Hilfe der Eigenschaft **PageNumberMode** festlegen.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	Seitenzahlen werden (True) / werden nicht (False) ausgegeben **Standardwert:** False

Code-Beispiel

```
VcTree1.Printer.PageNumbers = True
```

PagePaddingEnabled

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob zwischen dem Diagramm und den Boxen für Titel und Legende so viel Platz gelassen wird, dass die Boxen auf jeder Druckseite immer in voller Breite gedruckt werden können und fest am Blattrand positioniert sind. Ist die Eigenschaft auf **False** gesetzt, werden die Boxen ohne Zwischenraum am Diagramm gedruckt und können dann je nach Diagramm auf den verschiedenen Druckseiten in der Breite variieren.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zwischenraum zwischen Diagramm und Boxen für Legende/Titel wird (True) / wird nicht (False) ausgegeben **Standardwert:** True

Code-Beispiel

```
VcTree1.Printer.PagePaddingEnabled = True
```

PaperSize

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie die zu verwendende Papiergröße festlegen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	PaperSizeEnum **Mögliche Werte:** vcDIN_A2 66 vcDIN_A3 8 vcDIN_A4 9 vcISO_C 24 vcISO_D 25 vcISO_E 26 vcUS_LEGAL 5 vcUS_LETTER 1	Papiergröße DIN A2 DIN A3 DIN A4 ISO C ISO D ISO E US LEGAL US LETTER

Code-Beispiel

```
VcTree1.Printer.PaperSize = vcDIN_A3
```

PrintDate

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob das Druckdatum in der linken unteren Ecke jeder Seite erscheinen soll (True) oder nicht (False).

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Druckdatum wird/wird nicht ausgegeben

Code-Beispiel

```
VcTree1.Printer.PrintDate = True
```

PrinterName

Nur-Lese-Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Namen des aktuell ausgewählten Druckers auslesen oder setzen. Dies kann zum Speichern und Wiederherstellen des Zustands des Printer-Objekts verwendet werden.

Wenn man beim Setzen der Eigenschaft einen leeren String übergibt, wird der im System eingestellte Standarddrucker benutzt.

Hinweis: Bitte beachten Sie, dass bei Netzwerkdruckern der Druckername in UNC-Notation angegeben werden muss, bspw. "\\server01\printer5".

	Datentyp	Beschreibung
Eigenschaftswert	String	Druckername

RepeatTitleAndLegend

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Titel und Legende auf jeder Seite erscheinen sollen (True) oder nicht (False). Außerdem wird hiermit festgelegt, ob die Seitenaufteilung automatisch berechnet wird, so dass die Knoten nicht durchgeschnitten werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Titel und Legende werden auf jeder Seite wiederholt (True)/ Titel und Legende werden nur einmal ausgegeben und ggf. beim Seitenumbruch durchtrennt (False) **Standardwert:** False

Code-Beispiel

```
VcTree1.Printer.RepeatTitleAndLegend = True
```

StartUpSinglePage

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie festlegen oder erfragen, wie die Seitenansicht beim Aufruf aussehen soll: als Gesamtansicht über alle Blätter des Diagramms oder als Darstellung der ersten Seite.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	beim Aufruf der Seitenansicht: Darstellung der ersten Seite (True)/ Gesamtansicht über alle Blätter des Diagramms (False)

Code-Beispiel

```
Dim printer As VcPrinter

Set printer = VcTree1.printer

With printer
    .Orientation = vcLandscape
    .StartUpSinglePage = True
    .FitToPage = False
End With

VcTree1.PrintPreview
```

ZoomFactorAsDouble

Eigenschaft von VcPrinter

Mit dieser Eigenschaft können Sie den Zoomfaktor in Prozent für die Einstellung **FitToPage = False** setzen oder erfragen (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerungg).

	Datentyp	Beschreibung
Eigenschaftswert	Double	Zoomfaktor für das Diagramm **Standardwert:** 100

Code-Beispiel

```
VcTree1.Printer.ZoomFactorAsDouble = 150
```

7.36 VcRect

Ein Objekt vom Typ **VcRect** bezeichnet ein Rechteck-Objekt und wird nur im Ereignis VcTree.OnShowInPlaceEditor übergeben.

Eigenschaften

- Bottom
- Height
- Left
- Right
- Top
- Width

Eigenschaften

Bottom

Eigenschaft von VcRect

Diese Eigenschaft gibt die untere Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des unteren Rands des Rechtecks

Height

Nur-Lese-Eigenschaft von VcRect

Diese Eigenschaft gibt die Höhe des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe des Rechtecks

Left

Eigenschaft von VcRect

Diese Eigenschaft gibt die linke Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des linken Rands des Rechtecks

Code-Beispiel

```
Private Sub VcTree1_OnShowInPlaceEditor(ByVal editObject As Object, _
                           ByVal editObjectType As _
                           VcTreeLib.VcObjectTypeEnum, _
                           ByVal fieldIndex As Long, ByVal objRectComplete As _
                           VcTreeLib.VcRect, ByVal objRectVisible As _
                           VcTreeLib.VcRect, ByVal fldRectComplete As _
                           VcTreeLib.VcRect, ByVal fldRectVisible As _
                           VcTreeLib.VcRect, returnStatus As Variant)

   Dim oldScaleMode As Long

   If editObjectType = vcObjTypeNodeInTable Then
      returnStatus = vcRetStatFalse

      Set myEditObject = editObject
      myEditObjectType = editObjectType
      myEditObjectFieldIndex = fieldIndex

      oldScaleMode = Me.ScaleMode
      Me.ScaleMode = vbPixels

      Select Case fieldIndex
         Case 1   'Name
            Text1.Left = fldRectVisible.Left + VcTree1.Left
            Text1.Top = fldRectVisible.Top + VcTree1.Top
            Text1.Width = fldRectVisible.Width
            Text1.Height = fldRectVisible.Height

            Text1.Text = editObject.DataField(fieldIndex)
            Text1.Visible = True
            Text1.SetFocus

         Case 2, 3     'Start or End
            MonthView1.Left = fldRectVisible.Left + VcTree1.Left
            MonthView1.Top = fldRectVisible.Top + VcTree1.Top

            MonthView1.Value = editObject.DataField(fieldIndex)
            MonthView1.Visible = True
            MonthView1.SetFocus

         Case 13   'Employee
            Combo1.Left = fldRectVisible.Left + VcTree1.Left
            Combo1.Top = fldRectVisible.Top + VcTree1.Top
            Combo1.Width = fldRectVisible.Width

            Combo1.Text = editObject.DataField(fieldIndex)
            Combo1.Visible = True
            Combo1.SetFocus

      End Select

      Me.ScaleMode = oldScaleMode
```

```
   End If
End Sub
```

Right

Eigenschaft von VcRect

Diese Eigenschaft gibt die rechte Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des rechten Rands des Rechtecks

Top

Eigenschaft von VcRect

Diese Eigenschaft gibt die obere Position des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Position des oberen Rands des Rechtecks

Code-Beispiel

```
MonthView1.Top = fldRectVisible.Top + VcTree1.Top
```

Width

Nur-Lese-Eigenschaft von VcRect

Diese Eigenschaft gibt die Breite des Rechteckobjekts an.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Breite des Rechtecks

Code-Beispiel

```
Text1.Width = fldRectVisible.Width
```

7.37 VcTree

Ein Objekt vom Typ **VcTree** bezeichnet das VARCHART-XTree-Steuerelement selbst. Über Ereignisse können Sie dessen Interaktionen kontrollieren. Es kann durch eine Vielzahl von Eigenschaften und Methoden den Anforderungen entsprechend konfiguriert werden.

Eigenschaften

- ActiveNodeFilter
- AllowMultipleBoxMarking
- AllowNewNodes
- ArrangementField
- BorderArea
- BoxCollection
- BoxFormatCollection
- CollapseField
- ConfigurationName
- CtrlCXVProcessing
- CurrentVersion
- DataDefinition
- DataTableCollection
- DateOutputFormat
- DiagramBackColor
- DialogFont
- DoubleOutputFormat
- EditNewNode
- Enabled
- EnableSupplyTextEntryEvent
- EventReturnStatus
- EventText
- ExtendedDataTables
- FilePath
- FilterCollection
- FirstVerticalLevel
- FontAntiAliasingEnabled
- HorizontalNodeDistance
- HorizontalNodeIndent
- hWnd

- InPlaceEditingAllowed
- InteractionMode
- LegendView
- LevelField
- MapCollection
- MouseProcessingEnabled
- NodeAppearanceCollection
- NodeCollection
- NodeFormatCollection
- NodesDataTableName
- NodeTooltipTextField
- OLEDragMode
- OLEDragWithOwnMouseCursor
- OLEDragWithPhantom
- OLEDropMode
- ParentNodeIDDataFieldIndex
- Printer
- RoundedLinkSlantsEnabled
- RowLimit
- ScrollOffsetX
- ScrollOffsetY
- ShowToolTip
- StructureCodeDataFieldIndex
- StructureType
- ToolTipChangeDuration
- ToolTipDuration
- ToolTipPointerDuration
- ToolTipShowAfterClick
- TreeViewStyle
- VerticalLevelDistance
- VerticalNodeDistance
- WaitCursorEnabled
- WorldView
- ZoomFactor
- ZoomingPerMouseWheelAllowed

Methoden

- AboutBox
- Arrange
- Clear

- CopyNodesIntoClipboard
- CutNodesIntoClipboard
- DeleteNodeRecord
- DetectDataTableFieldName
- DetectDataTableName
- DetectFieldIndex
- DumpConfiguration
- EditNode
- EndLoading
- ExportGraphicsToFile
- GetAValueFromARGB
- GetBValueFromARGB
- GetGValueFromARGB
- GetNodeByID
- GetRValueFromARGB
- IdentifyFormatField
- IdentifyFormatFieldAsVariant
- IdentifyObjectAt
- IdentifyObjectAtAsVariant
- InsertNodeRecord
- InsertNodeRecordEx
- MakeARGB
- Open
- PageLayout
- PasteNodesFromClipboard
- PrintDirectEx
- PrinterSetup
- PrintIt
- PrintPreview
- PrintToFile
- Reset
- SaveAsEx
- ScrollToNodePosition
- ShowAlwaysCompleteView
- ShowExportGraphicsDialog
- SuspendUpdate
- UpdateNodeRecord
- Zoom
- ZoomOnMarkedNodes

Ereignisse

- Error
- ErrorAsVariant
- KeyDown
- KeyPress
- KeyUp
- OLECompleteDrag
- OLEDragDrop
- OLEDragOver
- OLEGiveFeedback
- OLESetData
- OLEStartDrag
- OnBoxLClick
- OnBoxLDblClick
- OnBoxModifyComplete
- OnBoxModifyCompleteEx
- OnBoxRClick
- OnDataRecordCreate
- OnDataRecordCreateComplete
- OnDataRecordDelete
- OnDataRecordDeleteComplete
- OnDataRecordModify
- OnDataRecordModifyComplete
- OnDataRecordNotFound
- OnDiagramLClick
- OnDiagramLDblClick
- OnDiagramRClick
- OnHelpRequested
- OnLegendViewClosed
- OnModifyComplete
- OnMouseDblClk
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnNodeCollapse
- OnNodeCreate
- OnNodeCreateCompleteEx
- OnNodeDelete
- OnNodeDeleteCompleteEx

- OnNodeExpand
- OnNodeLClick
- OnNodeLDblClick
- OnNodeModifyCompleteEx
- OnNodeModifyEx
- OnNodeRClick
- OnNodesMarkComplete
- OnNodesMarkEx
- OnSelectField
- OnShowInPlaceEditor
- OnStatusLineText
- OnSupplyTextEntry
- OnSupplyTextEntryAsVariant
- OnToolTipText
- OnToolTipTextAsVariant
- OnWorldViewClosed
- OnZoomFactorModifyComplete

Eigenschaften

ActiveNodeFilter

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Filter festlegen oder erfragen, der die darzustellenden Knoten selektiert. Es werden dabei immer die ausgefilterten Knoten mitsamt ihrer Teilbäume angezeigt

	Datentyp	Beschreibung
Eigenschaftswert	VcFilter	Filterobjekt **Standardwert:** Nothing

Code-Beispiel

```
Dim filter As VcFilter
Dim filterName As String

Set filter = VcTree1.ActiveNodeFilter

If Not Filter Is Nothing Then
    filterName = filter.Name
End If

Set VcTree1.ActiveNodeFilter = VcTree1.FilterCollection. _
                    FilterByName("Filter_1")
```

AllowMultipleBoxMarking

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Markieren von mehreren Boxen gleichzeitig zur Laufzeit möglich ist. Ist die Eigenschaft nicht gesetzt, ist zum Markieren mehrerer Boxen das zusätzliche Drücken der STRG-Taste erforderlich. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mehrfachmarkierung von Boxen möglich / nicht möglich **Standardwert:** True

Code-Beispiel

```
VcTree1.AllowMultipleBoxMarking = True
```

AllowNewNodes

Eigenschaft von VcTree

Mit dieser Eigenschaft wird dem Anwender das Anlegen neuer Knoten erlaubt (True) oder gesperrt (False). Wird diese Eigenschaft auf False gesetzt, kann der **Modus: Knoten erzeugen** nicht eingeschaltet werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Erzeugung neuer Knoten zugelassen/nicht zugelassen

Code-Beispiel

```
VcTree1.AllowNewNodes = False
```

ArrangementField

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Art der Anordnung eines Teilbaums synchron in einem Datenfeld halten. Der Inhalt des Datenfelds kann **0** (Teilbaum horizontal angeordnet) oder **1** (Teilbaum vertikal angeordnet) sein. Die horizontale Anordnung ist nur sichtbar, wenn der direkte und alle indirekten Vaterknoten horizontal angeordnet sind. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Knoten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Definitionsfeldindex oder "-1". Bei "-1" wird die Art der Anordnung nicht in einem Feld abgebildet. **Standardwert:** -1

Code-Beispiel

```
Dim subTreeArrangement As Integer

subTreeArrangement = VcTree1.ArrangementField
```

BorderArea

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf das BorderArea-Objekt, also auf den Titel- und Legendenbereich.

	Datentyp	Beschreibung
Eigenschaftswert	VcBorderArea	Titel- und Legendenbereich

Code-Beispiel

```
Dim borderArea As VcBorderArea

Set borderArea = VcTree1.BorderArea
```

BoxCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die Box-Auflistung und damit auf die verwendeten Boxen.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxCollection	BoxCollection-Objekt

Code-Beispiel

```
Dim boxCltn As VcBoxCollection

Set boxCltn = VcTree1.BoxCollection
```

BoxFormatCollection

Nur-Lese-Eigenschaft von VcTree

Mit dieser Eigenschaft haben Sie Zugriff auf die BoxFormat-Auflistung, in der alle zur Verfügung stehenden Box-Formate enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcBoxFormatCollection	BoxFormatCollection-Objekt

CollapseField

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Kollabierstatus eines Knotens synchron in einem Datenfeld halten. Der Inhalt des Datenfelds kann **0** (Knoten expandiert) oder **1** (Knoten kollabiert) sein. Der Knoten ist nur sichtbar, wenn der direkte Vaterknoten und jeder indirekte Vaterknoten expandiert sind. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Knoten** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Definitionsfeldindex oder" -1". Bei "-1" wird der Kollabierstatus nicht in einem Feld abgebildet. **Standardwert:** -1

Code-Beispiel

```
Dim nodeCollapsed As Integer

nodeCollapsed = VcTree1.CollapseField
```

ConfigurationName

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie eine Konfigurationsdatei (*.*ini*) laden, aus der alle relevanten Einstellungen übernommen werden. Das schließt eine zugehörige (ggf. andere) Datenschnittstelle (*.*ifd*) ein. Als Konfigurations-datei können Sie eine lokale Datei mit Pfad oder eine URL angeben.

- *lokale Datei:* Die Standard-Konfigurationsdatei **vctree.ini** sollte sich im gleichen Verzeichnis befinden wie **vctree.ocx**. Wenn Sie keinen Pfad angeben, wird die Konfigurationsdatei im Installationsverzeichnis erwartet. Falls die angegebene Datei nicht existiert, wird versucht, die Standard-Konfiguration zu laden, die aber nicht notwendigerweise beim Endanwender existiert.

- *URL:* Die Angabe einer URL als Konfigurationsdatei ist nur sinnvoll, wenn die Konfiguration über die API zur Laufzeit festgelegt wird, da dann die *ini*- und *ifd*-Dateien von der angegebenen URL heruntergeladen

werden. (Gibt man schon zur Designzeit eine URL als Konfigurationsdatei an, werden die *ini*- und *ifd*-Dateien zwar ebenfalls heruntergeladen, aber im Structured Storage (bei VB: *frx*-Datei) gespeichert. Dieser Speicher wird dann zur Laufzeit verwendet, statt die Dateien direkt herunterzuladen.) Sie können also, wenn Sie das VARCHART-ActiveX-Steuerelement in eine HTML-Seite einbetten, die URL für die *ini*- und *ifd*-Datei direkt setzen und müssen keine anderen Mechanismen verwenden, um temporär eine lokale Datei zu erzeugen, was von Browsern als unsicher eingestuft wird.

Hinweis: Beim Einlesen einer neuen Konfigurationsdatei gehen die bestehenden Daten verloren und müssen ggf. wieder eingelesen werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Dateiname **Standardwert:** vctree.ini

Code-Beispiel

```
VcTree1.ConfigurationName = "c:\VARCHART\XTree\sample.ini"
' or:
VcTree1.ConfigurationName = "http://members.tripod.de/netronic_te/ _
                                                    xtree_sample.ini"
```

CtrlCXVProcessing

Eigenschaft von VcTree

Mit dieser EigensMit dieser Eigenschaft werden die Tastenkombinationen Strg+C, Strg+X und Strg+V automatisch in die Zwischenablage-Operationen **CopyNodesToClipboard**, **CutNodesToClipboard** bzw. **PasteNodesFrom-Clipboard** übersetzt. Dieses Verhalten kann abgeschaltet werden, damit beispielsweise in Visual Basic-Applikationen keine Konflikte mit Tastenkombinationen (Shortcuts) für Menüpunkte entstehen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	Boolean	Tastenkombinationen werden/werden nicht in Zwischenablage-Operationen übersetzt **Standardwert:** True

Code-Beispiel

```
VcTree1.CtrlCXVProcessing = True
```

CurrentVersion

Nur-Lese-Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die aktuelle Versionsnummer eines VARCHART XTree-Objekts erfragen. Damit wird es auf einfache Art möglich, die zur Laufzeit im Kundensystem registrierte Version zu erkennen und ggf. eine Reparatur der Installation anzufordern, falls eine zu alte Version erkannt wird. Die Versionsnummer kann ebenfalls über die Eigenschaftenseite der Datei **vctree.ocx** im Tab **Version** eingesehen sowie über die FILEVERSION-Ressource dieser Datei gelesen werden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Versionsnummer

Code-Beispiel

```
MsgBox VcTree1.CurrentVersion
```

DataDefinition

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf das aktuelle VcDataDefinition-Objekt, um darin Namen oder Feldtypen abzufragen.

Die Datendefiniton des VcTree enthät die Datendefinitionstabelle **vcMaindata**.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataDefinition	DataDefinition-Objekt

Code-Beispiel

```
Dim dataDefTable As VcDataDefinitionTable
Dim dataField As VcDefinitionField

Set dataDefTable = VcTree1.DataDefinition.DefinitionTable(vcMaindata)
Set dataField = dataDefTable.FieldByName("Start")
index = dataField.ID
```

DataTableCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die DataTable-Auflistung und auf die darin verwendeten Datentabellen.

	Datentyp	Beschreibung
Eigenschaftswert	VcDataTableCollection	Ermitteltes Auflistungsobjekt der Datentabellen

Code-Beispiel

```
Dim dataTableCltn As VcDataTableCollection
Dim dataTable As VcDataTable

Set dataTableCltn = VcTree1.DataTableCollection
For Each dataTable In dataTableCltn
    List1.AddItem (dataTable.Name)
Next
```

DateOutputFormat

Eigenschaft von VcTree

Mit dieser Eigenschaft stellen Sie das Ausgabeformat von Terminen ein. Für das Datumsformat stehen folgende Kürzel zur Verfügung:

D: Wochentagsname erster Buchstabe

TD: Wochentagsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

DD. Tagesnummer zweistellig: 01-31 (nicht anpassbar)

DDD. die ersten drei Buchstaben des Wochentagsnamens (nicht anpassbar)

M: erster Buchstabe des Monatsnamens (nicht anpassbar)

TM. Monatsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

MM: zweistellige Monatsnummer: 01-12

MMM: erste drei Buchstaben des Monatsnamens (nicht anpassbar)

YY: zweistellige Jahreszahl

YYYY. vierstellige Jahreszahl

WW: zweistellige Nummer der Kalenderwoche: 01-53

TW: Text für "Kalenderwoche" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

Q: einstellige Quartalsnummer: 1-4

TQ: Quartalsname (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

hh: Stunde zweistellig im 24-Stunden-Format: 00-23

HH: Stunde zweistellig im 12-Stunden-Format: 01-12

Th: Text für "Uhr" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

TH: "am" oder "pm" (kann über das Ereignis **OnSupplyTextEntry** angepasst werden)

mm: Minute zweistellig: 00-59

ss: Sekunde zweistellig: 00-59

TS: kurzes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

TL: langes Datumsformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

TT: Zeitformat, wie in der Windows-Systemsteuerung über die Regions- und Sprachoptionen definiert

Hinweis Bitte setzen Sie Zeichen, die nicht als Datumsbestandteile interpretiert werden sollen, einen Backslash '\' voran. '\\' z.B. ergibt '\' in der Ausgabe. Die Sonderzeichen: **:, /, -** und **Leerzeichen** brauchen nicht durch '\' gekennzeichnet zu werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ dateFormat	String	Datumsformat MYhms:;/}
Eigenschaftswert	Date/Time	Datum {DMYhms:;/}

Code-Beispiel

```
VcTree1.DateOutputFormat = "DD.MM.YY"
```

DiagramBackColor

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Hintergrundfarbe Ihres Baumdiagramms setzen oder erfragen. Die Standard-Farbe ist weiß (=RGB (255,255,255)). Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte ({0...255},{0...255},{0...255}) **Standardwert:** (255,255,255)

Code-Beispiel

```
VcTree1.BackColor = RGB(200, 100, 150)
```

DialogFont

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Schriftgröße und den Schrifttyp der zur Laufzeit erscheinenden Dialogfelder im VARCHART XTree erfragen oder festlegen. Als Objekt wird ein Fontobjekt Ihrer Programmierumgebung erwartet, z. B. bei Visual Basic ein Objekt der Klasse **StdFont**.

	Datentyp	Beschreibung
Eigenschaftswert	String	Schriftname

Code-Beispiel

```
Dim newFont As New StdFont

newFont.Size = 14
newFont.Name = "Arial"
VcTree1.DialogFont = newFont
```

DoubleOutputFormat

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie das Ausgabeformat von Zahlen als double-Wert im Baum-Diagramm einstellen oder erfragen. Das Format kann über folgende Zeichen dargestellt werden:

- Text
- I
- D

sowie die Trennzeichen Komma und Punkt. Dabei steht **Text** für einen beliebigen Text, **I** für die Ziffern vor dem Dezimaltrenner und D für je eine Ziffer hinter dem Dezimaltrenner. Die erlaubte, generelle Reihenfolge ist

Text I D Text, wobei Komma und Punkt an beliebiger Stelle eingefügt werden können. Als Beispiel sei die Zahl -284901,3458 gegeben. Über das Format **I,DDDD ppm** wird sie als **-284901,3458 ppm** dargestellt. Über das Format **$I,III.DD** wird sie als **$-284,901.35** dargestellt.

	Datentyp	Beschreibung
Eigenschaftswert	String	Zeichenfolge, die das Double-Format beschreibt, z.B. "I,DDDD ppm"

Code-Beispiel

```
VcTree1.DoubleOutputFormat = "I,DDDD ppm"
```

EditNewNode

Eigenschaft von VcTree

Mit dieser Eigenschaft wird festgelegt, ob bei der Erzeugung eines neuen Knotens das Dialogfeld **Vorgänge bearbeiten** erscheinen soll oder nicht. Die Eigenschaft **AllowNewNodes** muss auf **True** gesetzt sein, damit überhaupt ein neuer Knoten erzeugt werden kann. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	**Vorgänge bearbeiten**-Dialogfeld erscheint /erscheint nicht

Code-Beispiel

```
VcTree1.EditNewNode = False
```

Enabled

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie das VARCHART-XTree-Steuerelement so abschalten, dass es auf Maus- und Tastenbefehle nicht reagiert.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	VARCHART-ActiveX-Steuerelement angeschaltet/abgeschaltet

Code-Beispiel

```
VcTree1.Enabled = False
```

EnableSupplyTextEntryEvent

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie das Ereignis **OnSupplyTextEntry** aktivieren oder deaktivieren. Mit Hilfe dieses Ereignisses können Sie die Texte aller Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen und Monats- und Tagesnamen, die zur Laufzeit erscheinen, verändern, beispielsweise um sie in unterschiedliche Sprachen zu übersetzen. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft

Code-Beispiel

```
VcTree1.EnableSupplyTextEntryEvent = True
```

EventReturnStatus

Eigenschaft von VcTree

Diese Eigenschaft benötigten Sie nur dann, wenn Sie mit einer Entwicklungsumgebung arbeiten, die keinen Rückgabewert in einer Ereignisprozedur erlaubt. Dies ist beispielsweise bei JavaScript der Fall. Mit dieser Eigenschaft überschreiben Sie innerhalb der Ereignismethode den defaultmäßigen returnStatus mit dem gewünschten Wert. Jede Setzung wirkt nur innerhalb des Ereignisses, in dem sie erfolgt ist.

	Datentyp	Beschreibung
Eigenschaftswert	ReturnStatusEnum	Rückgabewert des Ereignisses **Standardwert:** vcRetStatOK
	Mögliche Werte:	
	vcRetStatDefault 2	Das Default-Verhalten wird nicht verändert.
	vcRetStatFalse 0	Das Default-Verhalten wird nicht durchgeführt.
	vcRetStatNoPopup 4	Das Aufspringen des Rechte-Maustasten-Menüs wird unterdrückt.
	vcRetStatOK 1	Das Default-Verhalten wird durchgeführt.

Code-Beispiel

```
Private Sub VcTree1_OnDiagramRClick(ByVal x As Long, ByVal y As Long,
returnStatus As Variant)

    VcTree1.EventReturnStatus = vcRetStatNoPopup

End Sub
```

EventText

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft benötigten Sie nur dann, wenn Sie mit einer Entwicklungsumgebung arbeiten, die keine Setzung des Übergabeparameters in einer Ereignisprozedur erlaubt. Dies ist beispielsweise bei JavaScript der Fall. Mit dieser Eigenschaft setzen Sie den ToolTipText. Jede Setzung wirkt nur innerhalb des Ereignisses, in dem sie erfolgt ist.

	Datentyp	Beschreibung
Eigenschaftswert	String	Tool Tip

Code-Beispiel

```
Private Sub VcTree1_OnSupplyTextEntry(ByVal controlIndex As
VcTreeLib.TextEntryIndexEnum, TextEntry As String, returnStatus As Variant)

        VcTree1.EventText = "Order189"
End Sub
```

ExtendedDataTables

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie zwischen der Beschränkung auf zwei Datentabellen (Maindata und Relations) und der weiterentwickelten Verwendung von bis zu 90 Datentabellen wechseln. Die Verwendung der zweiten Option wird empfohlen. Diese Eigenschaft muss zu Beginn des Programms gesetzt werden, bevor Datentabellen und Datensätze angelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	**False**: nur zwei Datentabellen (Maindata und Relations) **True**: bis zu 99 Datentabellen **Standardwert:** False

Code-Beispiel

```
VcTree1.ExtendedDataTables = True
```

FilePath

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie einen Verzeichnispfad setzen, damit auch Grafikdateien, bei denen nur ein relativer Dateiname angegeben ist, in dem

angegebenen Verzeichnis gefunden werden. Andernfalls wird die Datei in dem gerade aktiven Arbeitsverzeichnis der Applikation und im Installationsverzeichnis des VARCHART-ActiveX-Steuerelements gesucht.

Üblicherweise wird diese Eigenschaft beim Start der Applikation während des Initialisierens des VARCHART-ActiveX-Steuerelements gesetzt, der Einfachheit halber wohl zumeist mit dem Verzeichnispfad der Applikation oder einem Unterverzeichnis desselben. Dies hat den Vorteil, dass die Applikation in einem beliebigen Verzeichnis installiert sein kann.

	Datentyp	**Beschreibung**
Eigenschaftswert	String	Verzeichnispfad **Standardwert:** " "

Code-Beispiel

```
Dim graphicsPath As String

graphicsPath = App.Path & "\bitmaps"
VcTree1.FilePath = graphicsPath
```

FilterCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die Filter-Auflistung und damit auf die verwendeten Filter.

	Datentyp	**Beschreibung**
Eigenschaftswert	VcFilterCollection	FilterCollection-Objekt

Code-Beispiel

```
Dim filterCollection As VcFilterCollection

Set filterCollection = VcTree1.FilterCollection
```

FirstVerticalLevel

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob ab einer bestimmten Ebene die Knoten vertikal angeordnet sind. Bei **-1** ist diese Eigenschaft abgeschaltet. Die Anordnung wird über die Methode **Arrange** durchgeführt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Nummer der Ebene, ab welcher der Teilbaum vertikal angeordnet ist **Standardwert:** -1

Code-Beispiel

```
VcTree1.FirstVerticalLevel = 3
VcTree1.Arrange
```

FontAntiAliasingEnabled

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie festlegen oder erfragen, ob Schriftzeichen optisch per GDI+ geglättet werden sollen. Wenn dies bei bestimmten Schriftarten, insbesondere bei nichtlateinischen Zeichen, zu verminderter Lesefähigkeit führt, sollte die Eigenschaft auf **False** gesetzt werden.

Die Glättung per GDI+ bewirkt zusätzlich, dass die Texte bei jeder Zoomstufe die gleiche relative Ausdehnung haben, so dass immer dieselbe Anzahl Zeichen z.B. in ein Knotenfeld passt. Wenn diese Eigenschaft aber auf False steht, dann wird stattdessen die Einstellung des Betriebssystems übernommen (einstellbar in der **Systemsteuerung**, Dialogfeld **Anzeige**, Reiter **Darstellung: Effekte**). Wenn dort eine Kantenglättung eingeschaltet ist, dann werden also Texte weiterhin geglättet. Es kann dann aber sein, dass bei manchen Zoomstufen mehr Text sichtbar ist als bei anderen, weil die systemeigene Kantenglättung dies nicht garantiert.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Textzeichen werden optisch geglättet / nicht geglättet.

HorizontalNodeDistance

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den horizontalen Abstand zwischen zwei horizontal angeordneten Knoten erfragen oder festlegen. Die Einheit beträgt mm. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Entfernung (mm) **Standardwert:** 0

Code-Beispiel

```
VcTree1.HorizontalNodeDistance = 10
```

HorizontalNodeIndent

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die horizontale Einrückung vertikal angeordneter Knoten erfragen oder festlegen. Die Einheit beträgt mm. Diese Eigenschaft kann auch über die Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Entfernung (mm) **Standardwert:** 0

Code-Beispiel

```
VcTree1.HorizontalNodeIndent = 30
```

hWnd

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft gibt eine Zugriffsnummer zurück. Die Arbeitsumgebung Microsoft Windows kennzeichnet jedes Formular und jedes Steuerelement in einer Anwendung durch Zuweisen einer Zugriffsnummer mit der Bezeichnung **hWnd**. Die Eigenschaft **hWnd** wird im Zusammenhang mit dem Aufruf der Windows-API verwendet. Viele Funktionen der Windows-Arbeitsumgebung benötigen die hWnd-Zugriffsnummer des aktiven Fensters als Argument.

Hinweis: Der Wert dieser Eigenschaft kann sich während der Programmausführung ändern. Sie sollten den Wert von **hWnd** daher nicht in einer Variablen speichern.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Zugriffsnummer

Code-Beispiel

```
MsgBox (Me.hWnd)
```

InPlaceEditingAllowed

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob in Knotenfeldern und Boxen das direkte Editieren möglich ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** eingestellt werden.

Hinweis: Falls einzelne Datenfelder nicht editierbar sein sollen, darf im Dialog **Datentabellen verwalten** das Attribut **editierbar** nicht ausgewählt sein.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Direktes Editieren in Knotenfeldern möglich (True) / nicht möglich (False) **Standardwert:** True

Code-Beispiel

```
VcTree1.InPlaceEditingAllowed = True
```

InteractionMode

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie einen der verfügbaren Interaktionsmodi einstellen oder erfragen.

	Datentyp	Beschreibung
Eigenschaftswert	InteractionModeEnum	Interaktionsmodus **Standardwert:** vcPointer
	Mögliche Werte: vcCreateNode 2 vcPointer 0	 Erzeugemodus für Knoten Selektiermodus

Code-Beispiel

```
VcTree1.InteractionMode = vcCreateNode
```

LegendView

Nur-Lese-Eigenschaft von VcTree

Über diese Eigenschaft erhalten Sie Zugriff auf das VcLegendView-Objekt, das die Legendenansicht des Diagramms definiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcLegendView	LegendView-Objekt

Code-Beispiel

```
Dim legendview As VcLegendView

Set legendview = VcTree1.LegendView
legendview.Visible = True
```

LevelField

Eigenschaft von VcTree

Mit dieser Eigenschaft kann man festlegen, in welchem Datenfeld die Ebenennummer der Knoten abgelegt wird. Die Ebenennummern zählen von 1 an aufwärts. Diese Eigenschaft kann auch über die Eigenschaftenseite **Knoten** festgelegt werden.

Hinweis: Zur Laufzeit können Sie nicht die Ebene des Knotens verändern, indem Sie den Wert des Ebenennummer-Datenfeldes ändern.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Ebenennummer **Standardwert:** -1

Code-Beispiel

```
VcTree1.LevelField = 4
```

MapCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die Map-Auflistung, in dem eine bestimmte Menge von Zuordnungstabellen enthalten ist. Die enthaltenen Zuordnungstabellen werden durch die Methode **VcMapCollection.SelectMaps** definiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcMapCollection	MapCollection-Objekt

Code-Beispiel

```
Dim mapCollection As VcMapCollection

Set mapCollection = VcTree1.MapCollection
mapCollection.SelectMaps vcAnyMap
```

MouseProcessingEnabled

Eigenschaft von VcTree

Diese Eigenschaft kann dazu genutzt werden, Ihre eigene Verarbeitung von Mausereignissen zu ermöglichen. Wenn Sie eine eigene Verarbeitung vom Ereignis **OnMouseDown** bis zum **OnMouseUp** durchführen möchten, setzen Sie die Eigenschaft **MouseProcessingEnabled** für diese Zeit auf False. Dann ignoriert VARCHART XTree bis zum Zurücksetzen auf True alle Mausbewegungen und Klicks.

Diese Eigenschaft kann auch in den OnMouse*-Ereignissen gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft (True)/ nicht in Kraft (False) **Standardwert:** True

NodeAppearanceCollection

Nur-Lese-Eigenschaft von VcTree

Mit dieser Eigenschaft haben Sie Zugriff auf die NodeAppearance-Auflistung, in der alle zur Verfügung stehenden NodeAppearance-Objekte enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeAppearanceCollection	NodeAppearanceCollection-Objekt

Code-Beispiel

```
Dim nodeAppearanceCollection As VcNodeAppearanceCollection
Dim nodeAppearance As VcNodeAppearance

Set nodeAppearanceCollection = VcTree1.NodeAppearanceCollection
Set nodeAppearance = nodeAppearanceCollection.FirstNodeAppearance
nodeAppearance.BackColor = RGB(200, 200, 200)
```

NodeCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die Knotenauflistung, in der abhängig von **SelectNodes** alle Knoten (vcAll), nur die markierten (vcMarked) oder nur die sichtbaren (vcAllVisible) Knoten enthalten sind.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeCollection	NodeCollection-Objekt

Code-Beispiel

```
Dim numberOfNodes As Long

numberOfNodes = VcTree1.NodeCollection.Count
```

NodeFormatCollection

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft ermöglicht den Zugriff auf die Format-Auflistung und damit auf die verwendeten Formate.

	Datentyp	Beschreibung
Eigenschaftswert	VcNodeFormatCollection	NodeFormatCollection-Objekt

Code-Beispiel

```
Dim formatCollection As VcNodeFormatCollection

Set formatCollection = VcTree1.NodeFormatCollection
```

NodesDataTableName

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Namen der Tabelle setzen oder erfragen, die die Felder für die Darstellung der Knoten zur Verfügung stellt. Dies ist nur möglich, solange noch keine Daten geladen wurden.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name der Datentabelle aus der die Knotendaten kommen

Code-Beispiel

```
Dim dataTable As VcDataTable
Dim dataRecord As VcDataRecord

'create Node DataTable
```

```
 Set dataTable = VcTree1.DataTableCollection.Add("NodeDataTable")
VcTree1.NodesDataTableName = dataTable.Name
dataTable.DataTableFieldCollection.Add("Id").PrimaryKey = True
'Load Data
Set dataTable = VcTree1.DataTableCollection.DataTableByName("NodeDataTable")
Set dataRecord = dataTable.DataRecordCollection.Add("1;Node One;")
Set dataRecord = dataTable.DataRecordCollection.Add("2;Node Two;")
VcTree1.EndLoading
```

NodeTooltipTextField

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, welches Datenfeld eines Knotens für Tooltips in VMF-Dateien genutzt werden soll. Drückt man im WebViewer die rechte Maustaste, erscheint der zugeordnete Text. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Knoten** eingestellt werden.

	Datentyp	**Beschreibung**
Eigenschaftswert	Integer	Index des Knoten-Datenfelds für Tooltiptexte **Standardwert:** 4

Code-Beispiel

```
VcTree1.NodeTooltipTextField = 1
```

OLEDragMode

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob das Ziehen eines Knotens über die Grenze der VARCHART-XTree-Komponente hinaus erlaubt sein soll. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** einstellen.

Wenn **OLEDragMode** auf **vcOLEDragManual** gesetzt wird, müssen Sie die Methode **OLEDrag** aufrufen, um das Ziehen eines Knotens zu starten. Bei **vcOLEDragAutomatic** hingegen wird das Ziehen eines Knotens über die Grenzen der VARCHART-XTree-Komponente automatisch gestartet.

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

VARCHART XTree verpackt die Daten in das Standard-Zwischenablage-Format CF_TEXT (für Visual-Basic-Benutzer: das vbCFText-Format), das mühelos gelesen werden kann.

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

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

	Datentyp	**Beschreibung**
Eigenschaftswert	OLEDragModeEnum	Verschiebemodus für Bewegungen aus der VARCHART-XTree-Komponente heraus **Standardwert:** vcOLEDragManual
	Mögliche Werte: vcOLEDragAutomatic 1 vcOLEDragManual 0	 Methode OLEDrag wird automatisch aufgerufen Methode OLEDrag muß eigens aufgerufen werden.

Code-Beispiel

```
VcTree1.OLEDragMode = vcOLEDragAutomatic
```

OLEDragWithOwnMouseCursor

Nur-Lese-Eigenschaft von VcTree

Diese Eigenschaft bestimmt, ob während eines OLE-Drag-Vorgangs der Cursor in der Zielkomponente gesetzt werden soll. Bei OLE Drag & Drop ist es möglich, den Cursor in der Quellkomponente über das Ereignis **OLEGiveFeedback** zu setzen. Daher würde ein Setzen durch die Zielkomponente zu einem Flimmern der konkurrierenden Cursor führen, was man über diese Eigenschaft beeinflussen kann.

Außerdem können bei eingeschaltetem Cursor und der auf **vcOLEDropManual** gesetzten Eigenschaft Objekte außerhalb der Anlagerungsstellen eines Knotens nicht fallengelassen werden, während dies bei ausgeschaltetem Cursor möglich ist.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Mauszeiger wird/wird nicht in der Zielkomponente gesetzt **Standardwert:** True

Code-Beispiel

```
VcTree1.OLEDragWithOwnMouseCursor = False
```

OLEDragWithPhantom

Eigenschaft von VcTree

Diese Eigenschaft bestimmt, ob während eines OLE-Drag-Vorgangs ein Phantom erscheinen soll oder nicht. Das Abschalten des Phantoms ist für Anwendungen gedacht, die beim Hineindraggen eines Objekts kein neues Objekt erzeugen, sondern z. B. den Knoten, auf dem dann ein Objekt fallengelassen wird, nur neu attributieren.

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Phantom erscheint/erscheint nicht **Standardwert:** True

Code-Beispiel

```
VcTree1.OLEDragWithPhantom = False
```

OLEDropMode

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie erfragen oder festlegen, ob ein Knoten aus einer anderen VARCHART-XTree-Komponente in die aktuelle Komponente herein gezogen werden darf.

Bei **OLEDropNone** ist dies nicht erlaubt. Bei **vcOLEDropManual** erhalten Sie beim Dropping das Ereignis **OLEDragDrop**, so dass Sie die übertragenen Daten selbst weiterverarbeiten können, um z. B. einen Knoten zu erzeugen oder eine Datei einzulesen. Wenn Quell- und Zielkomponente identisch sind, erhalten Sie wie bei abgeschaltetem OLE Drag&Drop eins der Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**. Bei **vcOLEDropAutomatic** wird der Dropping-Vorgang von der Komponente

selbst verarbeitet, das heißt, es wird, falls möglich, ein Knoten an entsprechender Mausposition erzeugt.

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

Sie können diese Eigenschaft auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	**Beschreibung**
Eigenschaftswert	OLEDropModeEnum	Verschiebemodus für Bewegungen in die VARCHART-ActiveX-Komponente herein **Standardwert:** vcOLEDropNone
	Mögliche Werte:	
	vcOLEDropAutomatic 2	Die Daten des übertragenen Objektes werden automatisch verarbeitet, d. h. es wird ein den Daten entsprechender Knoten an der Stelle des Droppings erzeugt.
	vcOLEDropManual 1	Das Ereignis **OLEDragDrop** wird aufgerufen, und die Daten des übertragenen Objektes müssen vom Programmierer verarbeitet werden.
	vcOLEDropNone 0	Dropping von nicht eigenen Objekten ist in der VARCHART-ActiveX-Komponente nicht zugelassen.

Code-Beispiel

```
VcTree1.OLEDropMode = vcOLEDropAutomatic
```

ParentNodeIDDataFieldIndex

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes setzen oder erfragen, das den Strukturcode des Baumdiagramms mit den IDs des Vaterknoten enthält. Dazu sollte der Strukturtyp auf **vcParentChild** gesetzt worden sein (s. Eigenschaft **StructureType**) .

	Datentyp	**Beschreibung**
Eigenschaftswert	Long	Datenfeld-Index

Printer

Eigenschaft von VcTree

Mit dieser Methode haben Sie Zugriff auf das Printer-Objekt. Mit diesem Objekt können Sie die Eigenschaften des aktuell verwendeten Druckers erfragen oder setzen.

	Datentyp	Beschreibung
Eigenschaftswert	VcPrinter	Druckerobjekt

Code-Beispiel

```
Dim printerZoomfactor As Integer
Dim printerCuttingMarks As String

printerZoomfactor = VcTree1.Printer.ZoomFactor
printerCuttingMarks = VcTree1.Printer.CuttingMarks
```

RoundedLinkSlantsEnabled

Nur-Lese-Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie einstellen oder erfragen, ob die Schrägen bei Verbindungen als Viertelkreise statt als gerade Linien dargestellt werden sollen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Allgemeines** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	System.Boolean Slants of links are to be displayed/not displayed as quarter circles	in

Code-Beispiel

```
VcTree1.RoundedLinkSlantsEnabled = True
```

RowLimit

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Höhe einer Baumstruktur erfragen oder festlegen. Diese Eigenschaft können Sie auch auf der Eigenschaftenseite **Layout** setzen.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Maximale Anzahl der Zeilen der Baumstruktur: {0...255}. Bei 0 ist keine Beschränkung wirksam. **Standardwert:** 0

Code-Beispiel

```
Dim rowLimit As Integer

rowLimit = VcTree1.RowLimit
```

ScrollOffsetX

Eigenschaft von VcTree

Mit dieser Eigenschaft kann der aktuelle Scroll-Offset des angezeigten Ausschnitts in x-Richtung gespeichert werden und bei einem neuen Start der gleichen Anwendung wieder gesetzt werden. Letzteres setzt voraus, dass zuvor auch der Zoomfaktor auf dieselbe Weise gesetzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Scroll-Offset in x-Richtung

ScrollOffsetY

Eigenschaft von VcTree

Mit dieser Eigenschaft kann der aktuelle Scroll-Offset des angezeigten Ausschnitts in y-Richtung gespeichert werden und bei einem neuen Start der gleichen Anwendung wieder gesetzt werden. Letzteres setzt voraus, dass zuvor auch der Zoomfaktor auf dieselbe Weise gesetzt wird.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Scroll-Offset in y-Richtung

ShowToolTip

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie das Ereignis **OnTooltipText** aktivieren oder deaktivieren. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Allgemeines** festgelegt werden. Mit Hilfe dieses Ereignisses können Sie die Texte der Tooltips bestimmen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft **Standardwert:** False

Code-Beispiel

```
VcTree1.ShowToolTip = True
```

StructureCodeDataFieldIndex

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Index eines Datenfeldes setzen oder erfragen, das den nummerierten Strukturcode des Baumdiagramms enthält. Dazu sollte der Strukturtyp auf **vcHierarchy** gesetzt worden sein (s. Eigenschaft **StructureType**) .

	Datentyp	Beschreibung
Eigenschaftswert	Long	Datenfeld-Index

StructureType

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den Strukturtyp des Baumdiagramms setzen oder erfragen. Die Struktur kann entweder einer Hierachie aus eigenen Ziffern folgen oder über die ID-Nummern der Vaterknoten definiert werden.

	Datentyp	Beschreibung
Eigenschaftswert	StructureTypeEnum	Strukturtyp des Baums-Diagramms **Standardwert:** vcHierarchy
	Mögliche Werte:	
	vcHierarchy 3	Der Struktur-Code folgt einer hierarchischen Ordnung nach dem Muster 1, 1.1, 1.1.1 etc.
	vcParentChild 2	Der Struktur-Code besteht aus den IDs der Vaterknoten

Code-Beispiel

```
VcTree1.StructureType = vcParentChild
```

ToolTipChangeDuration

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, die vergeht, bevor das nächste ToolTip-Fenster auf dem Bildschirm erscheint, wenn der Mauszeiger auf das nächste Objekt gesetzt wird. Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Standardwert von 98 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden. Maximalwert = 32767 ms **Standardwert:** -1

Code-Beispiel

```
VcTree1.ToolTipText = "Object"
VcTree1.ToolTipChangeDuration = 1000
```

ToolTipDuration

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, während der das ToolTip-Fenster sichtbar bleiben soll (sofern der Mauszeiger innerhalb des umgebenden Rechtecks eines Objektes unbewegt bleibt). Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Standardwert von 5.000 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden. Maximalwert = 32767 ms **Standardwert:** -1

Code-Beispiel

```
VcTree1.ToolTipText = "Object"
VcTree1.ToolTipDuration = 1000
```

ToolTipPointerDuration

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie die Zeitdauer setzen, während der der Mauszeiger innerhalb des umgebenden Rechtecks eines Objektes unbewegt bleiben muss, damit das ToolTip-Fenster erscheint. Einheit: Millisekunden. Mit der Ziffer -1 stellen Sie den Maximalwert von 480 Millisekunden ein.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Zeitdauer in Millisekunden **Standardwert:** -1

Code-Beispiel

```
VcTree1.ToolTipText = "Object"
VcTree1.ToolTipPointerDuration = 1000
```

ToolTipShowAfterClick

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie einstellen, ob das angezeigte ToolTip-Fenster beim Anklicken des Objektes verschwinden soll (Standard-Verhalten) oder entsprechend seiner eingestellten Zeiten weiterhin angezeigt werden soll.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	ToolTip-Fenster verschwindet (false) oder bleibt (true) **Standardwert:** False

Code-Beispiel

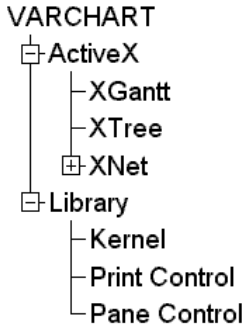
```
VcTree1.ToolTipShowAfterClick = True
```

TreeViewStyle

Eigenschaft von VcTree

Diese Eigenschaft bestimmt, ob vertikale Ebenen wie in TreeView-Steuerelementen mit **Plus**- oder **Minus**-Zeichen dargestellt werden sollen. Dabei bedeutet ein Plus-Zeichen, dass der darunterliegende Teilbaum kollabiert ist, ein Minus-Zeichen, dass er expandiert ist. Die Zeichen werden nur bei Knoten angezeigt, die Sohnknoten besitzen. Ein Mausklick auf eines der beiden Zeichen überführt den Knoten in den jeweils anderen Kollabierzustand.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Eigenschaft in Kraft/nicht in Kraft



TreeView-Stil: Plussymbol für kollabierten Teilbaum, Minussymbol für expandierten Teilbaum

Code-Beispiel

```
VcTree1.TreeViewStyle = True
```

VerticalLevelDistance

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den vertikalen Abstand zwischen zwei horizontal angeordneten Knotenebenen erfragen oder festlegen. Die Einheit beträgt mm. Diese Eigenschaft kann auch über die Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Entfernung (mm) **Standardwert:** 0

Code-Beispiel

```
VcTree1.VerticalLevelDistance = 10
```

VerticalNodeDistance

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie den vertikalen Abstand zwischen zwei vertikal angeordneten Knoten erfragen oder festlegen. Die Einheit beträgt mm. Diese Eigenschaft kann auch über die Eigenschaftenseite **Layout** eingestellt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Integer	Entfernung (mm) **Standardwert:** 0

Code-Beispiel

```
VcTree1.VerticalNodeDistance = 10
```

WaitCursorEnabled

Nur-Lese-Eigenschaft von VcTree

Mit dieser Eigenschaft kann man steuern, ob bei zeitkritischen Aufrufen (wie ScheduleProject) ein Wartecursor gesetzt werden soll oder nicht.

Die Eigenschaft kann auch auf der Eigenschaftseite **Allgemeines** gesetzt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Wartecursor wird/wird nicht gesetzt

WorldView

Nur-Lese-Eigenschaft von VcTree

Über diese Eigenschaft erhalten Sie Zugriff auf das VcWorldView-Objekt, das die Komplettansicht des Diagramms definiert.

	Datentyp	Beschreibung
Eigenschaftswert	VcWorldView	Komplettansicht-Objekt

Code-Beispiel

```
Dim worldview As VcWorldView

Set worldview = VcTree1.WorldView
worldview.Visible = True
```

ZoomFactor

Eigenschaft von VcTree

Mit dieser Eigenschaft kann der absolute Zoomfaktor der Bildschirmdarstellung in % angegeben oder erfragt werden (Zoomfaktor = 100: Originalgröße, Zoomfaktor > 100: Vergrößerung, Zoomfaktor < 100: Verkleinerung).

	Datentyp	Beschreibung
Eigenschaftswert	Integer {1...10000}	Zoomfaktor

Code-Beispiel

```
VcTree1.ZoomFactor = 200
```

ZoomingPerMouseWheelAllowed

Eigenschaft von VcTree

Mit dieser Eigenschaft können Sie setzen oder erfragen, ob das Zoomen per Mausrad zugelassen ist. Um zu zoomen, muss der Anwender dann die Strg-Taste festhalten und das Mausrad drehen.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Zoomen erlaubt (true)/nicht erlaubt (False)

Code-Beispiel

```
VcTree1.ZoomingPerMouseWheelAllowed = False
```

Methoden

AboutBox

Methode von VcTree

Mit dieser Methode können Sie die **About**-Box aufrufen. Sie enthält eine Übersicht über die jeweils verwendeten Programm- und Bibliotheksdateien mit absolutem Pfad und Versionsnummer. Dies dient der vereinfachten Hotline-Unterstützung. Die Übersicht kann mit der Maus selektiert und mit Strg+C herauskopiert werden, um sie z. B. mit Strg+V in eine Mail einzufügen.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.AboutBox
```

Arrange

Methode von VcTree

Mit dieser Methode können Sie die Knotenanordnung vornehmen, wie über die Eigenschaft **FirstVerticalLevel** festgelegt.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.FirstVerticalLevel = 2
VcTree1.Arrange
```

Clear

Methode von VcTree

Diese Methode sollte nur aufgerufen werden, wenn Knoten im Diagramm vorhanden sind. Mit dieser Methode werden alle grafisch darstellbaren Objekte aus dem Diagramm gelöscht und der initiale Zustand der ini-Datei wird wieder hergestellt.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Knoten erfolgreich gelöscht {True}

Code-Beispiel

```
VcTree1.Clear
```

CopyNodesIntoClipboard

Methode von VcTree

Mit dieser Methode werden die markierten Knoten in den Zwischenspeicher kopiert. Siehe auch die Methoden **CutNodesIntoClipboard** und **PasteNodesFromClipboard**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.CopyNodesIntoClipboard
```

CutNodesIntoClipboard

Methode von VcTree

Mit dieser Methode werden die markierten Knoten ausgeschnitten und in den Zwischenspeicher verschoben. Siehe auch **CopyNodesIntoClipboard** und **PasteNodesFromClipboard**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.CutNodesIntoClipboard
```

DeleteNodeRecord

Methode von VcTree

Mit dieser Methode können Sie einen Knoten löschen. Der Knoten wird durch die ID im Datensatz festgelegt. Welches Datenfeld als Identifizierung verwendet wird, wird auf der Eigenschaftenseite **Datendefinition** festgelegt.

	Datentyp	Beschreibung
Parameter:		
⇨ nodeRecord	Variant	Datensatz des Knotens
Rückgabewert	Boolean	Datensatz des Knotens erfolgreich (True) / nicht erfolgreich (False) gelöscht

Code-Beispiel

```
VcTree1.DeleteNodeRecord "A100;;;;;;"
```

DetectDataTableFieldName

Methode von VcTree

Mit dieser Eigenschaft können Sie über den Index eines Tabellendatenfeldes seinen Namen erfragen.

	Datentyp	Beschreibung
Parameter:		
⇨ fieldIndex	Long	Index des Datentabellenfeldes, dessen Name ermittelt werden soll
Rückgabewert	String	Zurückgegebener Name des Datentabellenfeldes

Code-Beispiel

```
'Find the name of a DataTableField
Dim fieldName As String

fieldName = VcTree1.DetectDataTableFieldName(0)
```

DetectDataTableName

Methode von VcTree

Mit dieser Eigenschaft können Sie über den Index einer Datentabelle ihren Namen erfragen.

	Datentyp	Beschreibung
Parameter:		
⇨ fieldIndex	Long	Index der Datentabelle, deren Name ermittelt werden soll
Rückgabewert	String	Name der Datentabelle

Code-Beispiel

```
'Find the name of a DataTable
Dim tableName As String

tableName = VcTree1.DetectDataTableName(0)
```

DetectFieldIndex

Methode von VcTree

Mit dieser Eigenschaft können Sie über den Namen einer Datentabelle und den Feldnamen den Index eines Tabellendatenfeldes erfragen.

	Datentyp	Beschreibung
Parameter:		
⇨ dataTableName	String	Name der Datentabelle, in der sich das Feld befindet, dessen Index ermittelt werden soll
⇨ dataTableFieldName	String	Name des Datentabellenfeldes, dessen Index ermittelt werden soll
Rückgabewert	String	Zurückgegebener Index des Datentabellenfeldes

Code-Beispiel

```
'Find the index of a DataTableField
Dim fieldIndex As Integer

fieldIndex = VcTree1.DetectFieldIndex("Maindata", "Name")
```

DumpConfiguration

Methode von VcTree

Mit dieser Methode können Sie die Konfiguration, bestehend aus .INI und .IFD-Datei, speichern.

Die Methode sollte lediglich zu Diagnosezwecken genutzt werden.

	Datentyp	Beschreibung
Parameter:		
⇨ FileName	String	Dateiname, ggf. mit Pfad.
⇨ encoding	EncodingEnum	Art der Kodierung
	Mögliche Werte:	
	vcANSIEncoding 1	Wird eine Datei in der ANSI-Kodierung gespeichert, so geschieht dies in Abhängigkeit von den lokalen Einstellungen des Windows-Betriebssystems, d.h. die Datei enthält Zeichen, die nur in der aktuell eingestellten Sprachversion auch wieder korrekt eingelesen werden können.
	vcUnicodeEncoding 2	Wird eine Datei in Unicode-Kodierung gespeichert, ist sie unabhängig von irgendwelchen Einstellungen. Dies Verfahren sollte, wenn möglich, bevorzugt werden. Eine in Unicode-Kodierung gespeicherte Datei erfordert jedoch in Visual Basic 6 eine spezielle Behandlung, wenn sie dort unabhängig von der VARCHART Komponente eingelesen werden soll.
Rückgabewert	Boolean	Datei erfolgreich (True)/nicht erfolgreich (False) abgespeichert.

EditNode

Methode von VcTree

Mit dieser Methode wird der Dialog **Vorgänge bearbeiten** für den angegebenen Knoten aufgerufen.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Knoten, dessen Daten editiert werden sollen
Rückgabewert	Boolean	Knotendaten wurden editiert/Editierung abgebrochen

Code-Beispiel

```
VcTree1.EditNode node
```

EndLoading

Methode von VcTree

Mit dieser Methode wird das Ende des Ladevorgangs bei **InsertNodeRecord** angezeigt. Dadurch wird eine Aktualisierung der Grafik ausgelöst.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ladevorgang beendet {True}

Code-Beispiel

```
VcTree1.EndLoading
```

ExportGraphicsToFile

Methode von VcTree

Mit dieser Methode können Sie ein Baumdiagramm in einer Datei abspeichern, ohne einen **Speichern unter**-Dialog zu erzeugen. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. mit eingebautem EMF)

Nur EMF, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

Beim Exportieren in Bitmapgrafikformate kann durch Angabe einer 0 bei der gewünschten Pixelzahl in X- oder Y-Richtung eine verzerrungsfreie Grafik

exportiert werden. Sind beide Pixelanzahlen 0, dann wird die Größe der exportierten Grafik in Pixeln von VARCHART XTree wie folgt berechnet:

- PNG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Die DPI-Zahl wird auch in der ausgegebenen PNG-Datei abgelegt, so dass Anzeigeprogramme die richtige Anzeigegröße bei gegebenem Zoomfaktor ermitteln können.

- GIF, TIFF, BMP, JPEG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.

Bei den Vektorgrafikformaten kann keine Pixelanzahl vorgegeben werden, sondern es werden folgende Koordinatenräume benutzt:

- WMF: Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.

- EMF/EMF+: Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand in beiden Richtungen X und Y verwendet.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten finden Sie im Kapitel: "Wichtige Konzepte: Grafikformate".

	Datentyp	**Beschreibung**
Parameter:		
⇨ FileName	String	Dateiname, ggf. mit Pfad.
⇨ PrintOutputFormat	PrintOutputFormat	Format der abzuspeichernden Datei
	Mögliche Werte:	
	vcBMP 2	Datei wird im Format BMP geschrieben.
	vcEMF 9	Datei wird im Format EMFgeschrieben.
	vcEMFPlus 12	Datei wird als *.EMF-Datei geschrieben, beinhaltet aber nur EMF+-Format.
	vcEMFWithEMFPlusIncluded 11	Datei wird als *.EMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF+-Format
	vcGIF 4	Datei wird im Format GIF geschrieben.
	vcJPG 5	Datei wird im Format JPG geschrieben.

	vcPNG 7 vcTIF 8 vcVMF 0 vcWMF 1 vcWMFWithEMFIncluded 10	Datei wird im Format PNG geschrieben. Datei wird im Format TIF geschrieben. Datei wird im Format VMF geschrieben. Datei wird im Format WMF geschrieben. Datei wird als *.WMF-Datei geschrieben, beinhaltet aber zusätzlich das EMF-Format.
⇨ SizeX	Integer	Breite des exportierten Diagramms in Pixeln. Nur bei Pixelformaten möglich. Bei Angabe von 0 wird der Wert unter Beachtung des Seitenverhältnisses berechnet.
⇨ SizeY	Integer	Höhe des exportierten Diagramms in Pixeln. Nur bei Pixelformaten möglich. Bei Angabe von 0 wird der Wert unter Beachtung des Seitenverhältnisses berechnet.
Rückgabewert	Boolean	Datei erfolgreich (True) / nicht erfolgreich (False) abgespeichert.

Code-Beispiel

```
VcTree1.ExportGraphicsToFile"C:\temp\export", vcVMF, 0, 0
```

GetAValueFromARGB

Methode von VcTree

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Alpha-Wert eines ARGB-Wertes.

	Datentyp	**Beschreibung**
Parameter:		
⇨ argb	Long	ARGB- Wert, aus dem der Alpha-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Alpha-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcTree1.MakeARGB(alpha,red,green,blue)
alpha = VcTree1.GetAValueFromARGB(argb)
```

GetBValueFromARGB

Methode von VcTree

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Blauwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter:		
⇨ argb	Long	ARGB- Wert, aus dem der Blau-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Blau-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcTree1.MakeARGB(alpha,red,green,blue)
blue = VcTree1.GetBValueFromARGB(argb)
```

GetGValueFromARGB

Methode von VcTree

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Grünwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter:		
⇨ argb	Long	ARGB- Wert, aus dem der Grün-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Grün-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
```

```
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcTree1.MakeARGB(alpha,red,green,blue)
green = VcTree1.GetGValueFromARGB(argb)
```

GetNodeByID

Methode von VcTree

Mit dieser Methode können Sie auf einen einzelnen Knoten über seine Identifikation zugreifen, die im Dialog **Datentabellen verwalten** festgelegt wurde. Wenn die Identifikation aus mehreren Feldern besteht (zusammengesetzter Primärschlüssel), muss diese mehrteilige ID folgendermaßen angegeben werden:

ID=ID1|ID2|ID3

	Datentyp	Beschreibung
Parameter:		
⇨ nodeID	Variant	Identifikation des Knotens
Rückgabewert	VcNode	Knoten

Code-Beispiel

```
Dim node As VcNode

Set node = VcTree1.GetNodeByID("10")
```

GetRValueFromARGB

Methode von VcTree

Ein Farbwert setzt sich aus vier Teilen zusammen: A (Alpha), R (Rot), G (Grün) und B (Blau). Der Alpha-Wert 0 bedeutet Volltransparenz und 255 ist ohne Transparenz. Die Farbwerte für R, G und B werden mit höherer Zahl heller, d.h. R,G,B von 0, 0, 0 ist schwarz und 255,255,255 ist weiß. Diese Methode erfragt den Rotwert eines ARGB-Wertes.

	Datentyp	Beschreibung
Parameter:		
⇨ argb	Long	ARGB- Wert, aus dem der Rot-Wert ermittelt werden soll
Rückgabewert	Integer	Zurückgegebener Rot-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = alpha + 11
red = red + 11
green = green + 11
blue = blue + 11
argb = VcTree1.MakeARGB(alpha,red,green,blue)
red = VcTree1.GetRValueFromARGB(argb)
```

IdentifyFormatField

Methode von VcTree

Mit dieser Methode können Sie das aktuell verwendete Format des angegebenen Knotens sowie den Index des an der bezeichneten Position befindlichen Formatfeldes erfragen. Falls sich an der bezeichneten Position ein Feld befindet, wird **True** zurückgegeben, andernfalls **False**.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **IdentifyFormatFieldAsVariant** benutzen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ x	Long	X-Koordinate der Position
⇨ y	Long	Y-Koordinate der Position
⇨ node	VcNode	Referenzknoten
⇦ format	VcNodeFormat	Identifiziertes Format
⇦ formatFieldIndex	Integer	Format-Feldindex
Rückgabewert	Boolean	Ein Formatfeld befindet sich/befindet sich nicht an der angegebenen Position

Code-Beispiel

```
Private Sub Vctree1_OnNodeLClick(ByVal node As VcTreeLib.VcNode, _
                                 ByVal location As VcTreeLib.LocationEnum, _
                                 ByVal x As Long, ByVal y As Long, _
                                 returnStatus As Variant)
    Dim foundFlag As Boolean
    Dim format As VcNodeFormat
    Dim formatFieldIndex As Integer

    foundFlag = VcTree1.IdentifyFormatField(x, y, node, format, _
                                                        formatFieldIndex)
    If foundFlag Then
        MsgBox "You hit the field with the index " + CStr(formatFieldIndex)
    End If
End Sub
```

IdentifyFormatFieldAsVariant

Methode von VcTree

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyFormatField**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

IdentifyObjectAt

Methode von VcTree

Mit dieser Methode können Sie das Objekt, das sich an einer bestimmten Position des Diagramms befindet, identifizieren. Der Typ des Objekts wird zurückgegeben. Derzeit können nur Knoten identifiziert werden.

Hinweis: Falls Sie VBScript verwenden, können Sie wegen der Parameter by-Reference nur die analoge Methode **IdentifyObjectAtAsVariant** benutzen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇦ identifiedObject	Object	Erkanntes Objekt
⇦ identifiedObjectType	VcObjectTypeEnum	Typ des erkannten Objekts
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeNode 2 vcObjTypeNodeInLegend 17 vcObjTypeNone 0	 Objekttyp **Box** Objekttyp **Knoten** Objekttyp **Knoten im Legendenbereich** kein Objekt
Rückgabewert	Boolean	Objekt identifiziert/Objekt nicht identifiziert

Code-Beispiel

```
Private Sub VcTree1_OnMouseMove(ByVal button As Integer, ByVal Shift As Integer,
ByVal x As Long, ByVal y As Long)

    Dim identifiedObject As Object
    Dim identifiedObjectType As VcObjectTypeEnum
    Dim node As VcNode

    Call VcTree1.IdentifyObjectAt(x, y, identifiedObject, identifiedObjectType)

    Select Case identifiedObjectType
```

```
        Case VcObjectTypeEnum.vcObjTypeNodeInDiagram,
VcObjectTypeEnum.vcObjTypeNodeInTable
            Set node = identifiedObject
            Label1.Caption = node.DataField(1)
        Case Else
            Label1.Caption = ""
    End Select

End Sub
```

IdentifyObjectAtAsVariant

Methode von VcTree

Diese Methode ist bis auf die Parameter identisch mit der Methode **IdentifyObjectAt**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

InsertNodeRecord

Methode von VcTree

Mit dieser Methode werden die Daten eines Knotens geladen. Die Daten werden als CSV-String (mit Semikolon als Trennzeichen) gemäß der auf der Eigenschaftenseite **Datendefinition** festgelegten Struktur übergeben. **EndLoading** sollte am Ende des kompletten Ladevorgangs einmal aufgerufen werden.

	Datentyp	**Beschreibung**
Parameter: ⇨ nodeRecord	data field/string	Datensatz des Knotens
Rückgabewert	VcNode	Knoten

Code-Beispiel

```
' data format: "Number;Name;Start date;Finish date;Group code;Group name"
VcTree1.InsertNodeRecord "A100;Node 1;12.09.14;17.09.14;5;Planning"
VcTree1.InsertNodeRecord "A105;Node 5;13.09.14;18.09.14;7;Testing"

VcTree1.EndLoading
```

InsertNodeRecordEx

Methode von VcTree

Mit dieser Methode können Sie Knoten unter Angabe einer Anlagerungsposition und eines Referenzknotens, an den der neue Knoten angelagert werden soll, einfügen. Die Daten werden dabei als CSV-String (mit Trennzeichen Semikolon) gemäß der auf der Eigenschaftenseite **Datendefinition** festgelegten Struktur übergeben. Die Methode **EndLoading** sollte am Ende des kompletten Ladevorgangs einmal aufgerufen werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ nodeRecord	Variant	Datensatz des Knotens
⇨ position	InsertionPositionEnum	Mögliche Anlagerungspositionen
	Mögliche Werte:	
	vcIPFirstChild 3	als ersten Sohnknoten des Referenzknotens einfügen
	vcIPLastChild 4	als letzten Sohnknoten des Referenzknotens einfügen
	vcIPLeftBrother 31	als linken Bruderknoten des Referenzknotens einfügen
	vcIPNone 0	nicht zulässige Einfügeposition (gilt nur für DataObject.DropInsertionPosition)
	vcIPNormal 1	ohne Referenzknoten einfügen
	vcIPParent 6	als Vaterknoten des Referenzknotens einfügen
	vcIPRightBrother 32	als rechten Bruderknoten des Referenzknotens einfügen
⇨ referenceNode	VcNode	Referenzknoten
Rückgabewert	VcNode	Knoten

Code-Beispiel

```
Dim node1 As VcNode

Set node1 = VcTree1.InsertNodeRecordEx("A2;Node 1;12.09.14;17.09.14;5;Planning",
vcIPFirstChild, VcTree1.GetNodeByID("A1"))
VcTree1.EndLoading
```

MakeARGB

Methode von VcTree

Mit dieser Methode können Sie aus den vier Einzelwerten einer Farbe einen ARGB-Wert bilden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ alpha	Integer	Alpha-Wert

⇨ red	Integer	Rot-Wert
⇨ green	Integer	Grün-Wert
⇨ blue	Integer	Blau-Wert
Rückgabewert	Long	Zurückgegebener ARGB-Wert

Code-Beispiel

```
Dim alpha As Integer
Dim red As Integer
Dim green As Integer
Dim blue As Integer
Dim argb As Long
alpha = FF
red = A0
green = 34
blue = ABargb = VcTree1.MakeARGB(alpha,red,green,blue)
```

Open

Methode von VcTree

Mit dieser Methode werden die Daten aus der angegebenen Datei geladen. In der Datei müssen die Daten, wie auf der Eigenschaftenseite **Datendefinition** festgelegt, im CSV-Format (mit Semikolon als Trennzeichen) gespeichert sein.

	Datentyp	**Beschreibung**
Parameter:		
⇨ fileName	String	Dateiname
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcTree1.Open "C:\Data\project1.wbs"
```

PageLayout

Methode von VcTree

Mit dieser Methode wird der Dialog **Seite einrichten** aufgerufen.

	Datentyp	**Beschreibung**
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcTree1.PageLayout
```

PasteNodesFromClipboard

Methode von VcTree

Mit dieser Methode fügen Sie Knoten aus der Zwischenablage an einer bestimmten Position in das Chart ein.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Referenzknoten
⇨ pastePosition	PastePositionEnum	Position, an der die Knoten aus der Zwischenablage eingefügt werden.
	Mögliche Werte:	
	vcPasteAfter 1	Objekte werden in einer horizontalen Anordnung rechts, in einer vertikalen Anordnung unterhalb des Referenzknotens eingefügt.
	vcPasteAsFirstChild 2	Objekte werden als erste Sohnknoten des Referenzknotens eingefügt.
	vcPasteAsLastChild 3	Objekte werden als letzte Sohnknoten des Referenzknotens eingefügt.
	vcPasteBefore 0	Objekte werden in einer horizontalen Anordnung links, in einer vertikalen Anordnung oberhalb des Referenzknotens eingefügt.
Rückgabewert	Void	

Code-Beispiel

```
Dim nodecollection As VcNodeCollection

Set nodecollection = VcTree1.nodecollection
nodecollection.SelectNodes (vcMarked)

If nodecollection.Count = 1 Then
    VcTree1.PasteNodesFromClipboard nodecollection.FirstNode, vcPasteAsLastChild
End If
```

PrintDirectEx

Methode von VcTree

Mit dieser Methode können Sie das Diagramm direkt ausdrucken, ohne dass zuvor ein Dialogfeld erscheint. Der Rückgabewert gibt bei nicht erfolgreichem Druck den Grund für die Ursache an. Dies kann z.B. auch ein Eintrag in einer Logdatei sein.

	Datentyp	Beschreibung	
Rückgabewert	PrintResultStatusEnum	**Mögliche Werte:**	
		vcPrintingSucceeded 0:	Druck verlief erfolgreich
		vcNoPrinterInstalled 1	Es wurde weder der über VcPrinter.PrinterName angegebene noch ein im Windows-Betriebssystem als Standarddrucker gekennzeichneter Drucker gefunden.
		vcPrintingAbortedByUser 2:	Der Druck wurde durch den Anwender abgebrochen.
		vcPrintingAbortedByDriver 3:	Der Druck wurde durch den Windows-Druckertreiber abgebrochen.
		vcUnprintablePageLayout 4:	Es konnte nicht gedruckt werden, weil das Seitenlayout zusammen mit den Druckereigenschaften wie Papiergröße und Ränder zu einem nicht druckbaren Layout führten.

Code-Beispiel

```
PrintStatusResultEnum status = VcTree1.PrintDirectEx()
If status <> vcPrintingSucceeded Then
   Debug.Print "Printing failed: " & status & vbCrLf
End If
```

PrinterSetup

Methode von VcTree

Mit dieser Methode wird das Windows-Dialogfeld **Drucker einrichten** aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcTree1.PrinterSetup
```

PrintIt

Methode von VcTree

Mit dieser Methode wird der Ausdruck der Grafik ausgelöst. Es werden die unter **PageLayout** aktuell eingestellten Parameter verwendet.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcTree1.PrintIt
```

PrintPreview

Methode von VcTree

Mit dieser Methode wird die Druckvorschau aufgerufen.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Ohne Bedeutung {True}

Code-Beispiel

```
VcTree1.PrintPreview
```

PrintToFile

Methode von VcTree

Mit dieser Methode können Sie das Diagramm direkt in eine Datei drucken. Ob dies gelingt, hängt vom Druckertreiber ab, da viele PDF-Druckertreiber keine Dateinamen akzeptieren.

	Datentyp	Beschreibung
Parameter: ⇨ fileName	String	Dateiname
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.PrintToFile
```

Reset

Methode von VcTree

Mit dieser Methode wird je nach eingestelltem Wert von resetAction der komplette Inhalt sämtlicher Datentabellen gelöscht bzw. der zur Entwurfszeit auf den Eigenschaftenseiten eingestellte Zustand wiederhergestellt

	Datentyp	Beschreibung
Parameter:		
⇨ resetAction	ResetActionEnum	Objekte, die gelöscht oder neu initialisiert werden
	Mögliche Werte:	
	vcEmptyAllDataTables 4	Der komplette Inhalt sämtlicher Datentabellen wird gelöscht, die Datentabellen selbst bleiben bestehen.
	vcReloadConfiguration 2	Komplette Neuinitialisierung. Alle Einstellungen und erzeugten Objekte verfallen.
Rückgabewert	Boolean	Objekte im Diagramm erfolgreich gelöscht (True)

Code-Beispiel

```
VcTree1. Reset(vcRemoveNodes) = True
```

SaveAsEx

Methode von VcTree

Mit dieser Methode werden die aktuellen Daten in die angegebene Datei im CSV-Format (Trennzeichen: Semikolon) gespeichert. Dabei wird die auf der Eigenschaftenseite **Objekte** unter **Datentabellen** festgelegte Struktur verwendet. Datentabellen, die keine Datensätze enthalten, werden nicht gespeichert. Ist kein Name angegeben, wird die zuletzt bei **Open** angegebene Datei überschrieben (entspricht der üblichen **Save**-Funktion).

	Datentyp	Beschreibung
Parameter:		
⇨ fileName	String	Dateiname
⇨ encoding	EncodingEnum	Art der Kodierung
	Mögliche Werte:	
	vcANSIEncoding 1	Wird eine Datei in der ANSI-Kodierung gespeichert, so geschieht dies in Abhängigkeit von den lokalen Einstellungen des Windows-Betriebssystems, d.h. die Datei enthält Zeichen, die nur in der aktuell eingestellten Sprachversion auch wieder korrekt eingelesen werden können.

	vcUnicodeEncoding 2	Wird eine Datei in Unicode-Kodierung gespeichert, ist sie unabhängig von irgendwelchen Einstellungen. Dies Verfahren sollte, wenn möglich, bevorzugt werden. Eine in Unicode-Kodierung gespeicherte Datei erfordert jedoch in Visual Basic 6 eine spezielle Behandlung, wenn sie dort unabhängig von der VARCHART Komponente eingelesen werden soll.
Rückgabewert	Boolean	Speicherung erfolgreich (True)/nicht erfolgreich (False) erfolgt

Code-Beispiel

```
VcTree1.SaveAs "C:\Data\project1.wbs"

' or

VcTree1.SaveAs ""
```

ScrollToNodePosition

Methode von VcTree

Mit dieser Methode können Sie zu der Position eines bestimmten Knotens scrollen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ Node	VcNode	Knoten, zu dem gescrollt werden soll
Rückgabewert	Boolean	Scrollen erfolgreich (True) / nicht erfolgreich (False) durchgeführt

Code-Beispiel

```
Private Sub Vctree1_OnNodeLClick(ByVal node As VcTreeLib.VcNode, _
                            ByVal location As VcTreeLib.LocationEnum, _
                            ByVal x As Long, ByVal y As Long, _
                            returnStatus As Variant)
    'scroll the diagram so that the node is completely on screen
    VcTree1.ScrollToNodePosition node

End Sub
```

ShowAlwaysCompleteView

Methode von VcTree

Mit dieser Methode können Sie die Darstellung so einstellen, dass stets das komplette Diagramm angezeigt wird. Der Zoomfaktor paßt sich bei jeder Änderung des Charts automatisch an. Der maximale Zoomfaktor von 100% wird nicht überschritten, die Knoten werden also höchstens in Originalgröße dargestellt. Siehe auch Eigenschaft **ZoomFactor** und Methode **Zoom**.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.ShowAlwaysCompleteView
```

ShowExportGraphicsDialog

Methode von VcTree

Mit dieser Methode können Sie ein **Speichern unter**-Dialogfeld aufrufen, um die Darstellung abzuspeichern. Mögliche Formate:

- *.BMP (Microsoft Windows Bitmap)
- *.EMF (Enhanced Metafile oder Enhanced Metafile Plus)
- *.GIF (Graphics Interchange Format)
- *.JPG (Joint Photographic Experts Group)
- *.PNG (Portable Network Graphics)
- *.TIF (Tagged Image File Format)
- *.VMF (Viewer Metafile)
- *.WMF (Microsoft Windows Metafile, ggf. mit eingebautem EMF)e)

*.WMF mit eingebautem EMF

Nur EMF, EMF+, VMF und WMF sind Vektorformate, in denen das Diagramm auflösungsunabhängig gespeichert werden kann. Die übrigen Formate sind pixelorientiert und bieten damit nicht beliebige Auflösungen.

Das VMF-Format wird in der Zukunft nicht mehr weiterentwickelt, aus Kompatibilitätsgründen für bestehende Anwendungen aber zunächst noch weiter unterstützt.

Detaillierte Erläuterungen zu den einzelnen Grafik-Formaten lesen Sie bitte im Kapitel: **Wichtige Konzepte: Grafikformate**.

Beim Exportieren wird die Größe des exportierten Diagramms in Pixeln wie folgt berechnet:

- PNG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen.

- GIF, TIFF, BMP, JPEG: Es wird eine Auflösung von 100 dpi bei einem Zoomfaktor von 100% angenommen. Wird alternativ im Parameter SizeX ein Wert <= -50 angegeben, so wird die absolute Zahl als DPI-Vorgabe genommen. Es gibt aber zusätzlich eine interne Begrenzung auf 50 MB Größe für die im Speicher für das Exportieren benötigte, nicht komprimierte Ausgangsbitmap, so dass größere Grafiken eine kleinere Auflösung bekommen als gewünscht.

- WMF: Es wird eine feste Auflösung angenommen, bei der die größere Ausdehnung die Koordinaten von 0 bis 10.000 benutzt. Die kleinere Ausdehnung benutzt entsprechend einen kleineren Maximalwert für die Koordinaten für eine verzerrungsfreie Darstellung.

- EMF/EMF+: Es wird die volle Auflösung mit Koordinaten in 1/100 mm Abstand verwendet.

	Datentyp	Beschreibung
Rückgabewert	Boolean	Diagramm erfolgreich (True) / nicht erfolgreich (False) exportiert

Code-Beispiel

```
VcTree1.ShowExportGraphicsDialog
```

SuspendUpdate

Methode von VcTree

Bei größeren Datenmengen kann es unter Umständen zu lange dauern, wenn man bei einer großen Anzahl von Knoten dieselbe Aktion durchführt. Dies kann man mit Hilfe der Methode **SuspendUpdate** beschleunigen. Klammern Sie den Code für die wiederholte Aktion wie im Code-Beispiel durch **SuspendUpdate (True)** und **SuspendUpdate (False)** ein. Dann wird das Update nicht für jeden Knoten einzeln, sondern für alle gemeinsam durchgeführt, wodurch die Performance erhöht wird.

	Datentyp	Beschreibung
Rückgabewert	Boolean	SuspendUpdate(True): Beginn der SuspendUpdate Methode/ SuspendUpdate(False): Ende der SuspendUpdate Methode

Code-Beispiel

```
VcTree1.SuspendUpdate (True)

   If updateFlag Then
      For Each node In nodeCltn
         If node.DataField(2) < "07.09.14" Then
            node.DataField(13) = "X"
            node.UpdateNode
            counter = counter + 1
         End If
      Next node
   Else
      For Each node In nodeCltn
         If node.DataField(2) < "07.09.14" Then
            node.DataField(13) = ""
            node.UpdateNode
            counter = counter + 1
         End If
      Next node
   End If

VcTree1.SuspendUpdate (False)
```

UpdateNodeRecord

Methode von VcTree

Mit dieser Methode können die Daten eines bestehenden Datensatzes eines Knotens verändert werden. Der Datensatz wird durch die auf der Eigenschaftenseite **Datendefinition** festgelegte ID identifiziert. Diese Methode kommt zur Anwendung, wenn externe Änderungen in der Grafik auf dem Bildschirm nachvollzogen werden sollen.

	Datentyp	Beschreibung
Parameter:		
⇨ nodeRecord	Variant	Datensatz des Knotens
Rückgabewert	VcNode	Aktualisierter Knoten

Code-Beispiel

```
VcTree1.UpdateNodeRecord "A100;Activity 1;12.09.14;18.09.14;6;Planning"
```

Zoom

Methode von VcTree

Mit dieser Methode kann die Bildschirmdarstellung um den angegebenen Prozent-Faktor vergrößert (Zoomfaktor > 100) oder verkleinert (Zoomfaktor < 100) werden.

	Datentyp	Beschreibung
Parameter:		
⇨ zoomFactor	Integer	Zoomfaktor {11...999}, andere Werte bleiben unberücksichtigt
Rückgabewert	Boolean	Zoomen erfolgreich durchgeführt {True}

Code-Beispiel

```
VcTree1.Zoom 120
```

ZoomOnMarkedNodes

Methode von VcTree

Mit dieser Methode können Sie auf die markierten Knoten zoomen.

	Datentyp	Beschreibung
Rückgabewert	Void	

Code-Beispiel

```
VcTree1.ZoomOnMarkedNodes
```

Ereignisse

Error

Ereignis von VcTree

Dieses Ereignis tritt nur dann auf, wenn ein unvorhergesehener Fehler im Code des VARCHART XTree entdeckt wird. NETRONIC ist bemüht, jeden dieser Fehler zu beseitigen. Um sie auf einfache Art beim Kunden, z. B. in einer Datei, protokollieren zu können, werden sie nun über dieses Ereignis nach außen bekanntgegeben. Das Parameterprofil ist durch den ActiveX-Standard vorgegeben. Dadurch sind die übergebenen Parameter teilweise

konstant. Die Nummer sollte im Ereignis immer gegengeprüft werden, um bei zukünftigen Erweiterungen nicht alle Fehlerarten pauschal abzublocken.

	Datentyp	**Beschreibung**
Parameter:		
⇨ Description	String	Fehlerbeschreibungstext
⇨ Scode	Long	&h800a402f (konstant)
⇨ Source	String	Name des Controls (konstant)
⇨ HelpFile	String	Hilfedatei: "" (konstant)
⇨ HelpContext	Long	Hilfekontext: 0 (konstant)
⇦ CancelDisplay	Boolean	Beim Wert **True** wird kein normaler Fehler mit der Nummer 71 mehr ausgegeben, der über das 'On Error GoTo'-Konstrukt abfangbar wäre.

Code-Beispiel

```
Private Sub VcTree1_Error(Number As Integer, Description As String, _
          Scode As Long, Source As String, HelpFile As String, HelpContext _
          As Long, CancelDisplay As Boolean)

   Debug.Print CStr(Number) + " " + Description

End Sub
```

ErrorAsVariant

Ereignis von VcTree

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **Error**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

KeyDown

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender eine Taste drückt, während VARCHART XTree den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen. (Zum Interpretieren von ANSI-Zeichen verwenden Sie das KeyPress-Ereignis.)

	Datentyp	Beschreibung
Parameter:		
⇨ KeyCode	Integer	Tasten-Code wie vbKeyF1 (F1-Taste) oder vbKeyHome (POS1-Taste)
⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat SHIFT den Wert 6.

Code-Beispiel

```
Private Sub VcTree1_KeyDown(KeyCode As Integer, Shift As Integer)
    MsgBox "key pressed"
End Sub
```

KeyPress

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender eine ANSI-Taste drückt und wieder loslässt, während VARCHART XTree den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen.

	Datentyp	Beschreibung
Parameter:		
⇨ KeyAscii	Integer	Eine Ganzzahl, die den numerischen Tasten-Code einer Standard-ANSI-Taste zurückgibt. KeyAscii wird als Referenz übergeben. Wird das Argument geändert, wird ein anderes Zeichen an das Objekt gesendet. Das Ändern von KeyAscii auf 0 hebt den Tastenanschlag auf, d.h. das Objekt erhält kein Zeichen.

Code-Beispiel

```
Private Sub VcTree1_KeyPress(KeyAscii As Integer)
    MsgBox "Key pressed and released."
End Sub
```

KeyUp

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender eine Taste loslässt, während VARCHART XTree den Fokus hat. Mit Hilfe der Key-Ereignisse können Sie mit Hilfe der Tastatur Funktionen des VARCHART ActiveX auslösen. (Zum Interpretieren von ANSI-Zeichen verwenden Sie das KeyPress-Ereignis.)

	Datentyp	Beschreibung
Parameter:		
⇨ KeyCode	Integer	Tasten-Code wie vbKeyF1 (F1-Taste) oder vbKeyHome (POS1-Taste)
⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.

Code-Beispiel

```
Private Sub VcTree1_KeyUp(KeyCode As Integer, Shift As Integer)
    MsgBox "key released"
End Sub
```

OLECompleteDrag

Ereignis von VcTree

Dieses Ereignis tritt auf, wenn ein OLE Drag & Drop-Vorgang beendet wurde und die Quellkomponente darüber informiert wird, dass eine Drag-Aktion entweder erfolgreich durchgeführt oder abgebrochen wurde.

	Datentyp	Beschreibung
Parameter:		
⇨ effect	Long	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet. Effect ist zunächst ein Wert, der die von der Drag-Quelle unterstützten OLE-Drag & Drop-Operationen bezeichnet. Die Zielkomponenente sollte diesen Wert prüfen und beim Ablegevorgang festlegen.
	Mögliche Werte:	
	vcDropEffectCopy 1	Eine Drop-Operation führt zum Kopieren der Daten vom Entnahme- zum Ablageort. Die ursprünglichen Daten werden durch die Drag-Operation nicht geändert.

vcDropEffectLink 4	Es wurde eine Verknüpfung für Daten von der Drag-Quelle zum Drop-Ziel erstellt.
vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten vom Entnahme- zum Ablageort verschoben werden. Am Entnahmeort sollten die Daten nach erfolgtem Verschieben entfernt werden.
vcDropEffectNone 0	Ablageziel kann die Daten nicht akzeptieren

OLEDragDrop

Ereignis von VcTree

Dieses Ereignis tritt auf, wenn bei einem OLE Drag & Drop-Vorgang Daten auf dem Drop-Ziel abgelegt werden und die **OLEDropMode**-Eigenschaft des Drop-Ziels auf **vcOLEDropManual** gesetzt ist und Ziel- und Quellkomponente nicht identisch sind. Bei Gleichheit der beiden Komponenten erhält man stattdessen eines der beiden Ereignisse **OnNodeModifyEx** oder **OnNodeCreate**.

	Datentyp	**Beschreibung**
Parameter:		
⇨ data	DataObject	Das OLE Drag & Drop-DataObject, in welchem die Daten importiert werden
⇨ effect	Long	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet.
	Mögliche Werte:	
	vcDropEffectCopy 1	Eine Drop-Operation führt zum Kopieren der Daten vom Entnahme- zum Ablageort. Die ursprünglichen Daten werden durch die Drag-Operation nicht geändert.
	vcDropEffectLink 4	Es wurde eine Verknüpfung für Daten von der Drag-Quelle zum Drop-Ziel erstellt.
	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten vom Entnahme- zum Ablageort verschoben werden. Am Entnahmeort sollten die Daten nach erfolgtem Verschieben entfernt werden.
	vcDropEffectNone 0	Ablageziel kann die Daten nicht akzeptieren
⇨ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: **1** (links), **2** (rechts) oder **4** (Mitte).
⇨ shift	Integer	Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Drop-Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen **1**, **2** und **4** entsprechen. Der Parameter **shift** zeigt den Status dieser Tasten an; es können einige, alle oder keines der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise die <STRG>- und die <ALT>-Tasten gedrückt werden, ist der Wert von **shift** "6".
⇨ y	Long	Y-Koordinate des Mauszeigers

⇨ x	Long	X-Koordinate des Mauszeigers

OLEDragOver

Ereignis von VcTree

Dieses Ereignis tritt auf, wenn Daten über ein Drop-Ziel gezogen und die **OLEDropMode**-Eigenschaft des Drop-Ziels auf **vcOLEDropManual** gesetzt ist.

	Datentyp	**Beschreibung**
Parameter:		
⇨ data	DataObject	Das OLE-Drag & Drop-DataObject, in das die Daten importiert werden
⇔ effect	Long	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet.
	Mögliche Werte:	
	vcDropEffectCopy 1	Die Drop-Operation führt zum Kopieren der Daten von der Quelle zum Ziel. Die ursprünglichen Daten wurden durch die Drag-Operation nicht geändert.
	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten von der Quelle zum Ziel verschoben werden. Die Quelle sollte die Daten nach erfolgtem Verschieben aus sich selbst entfernen.
	vcDropEffectNone 0	Ziel kann die Daten nicht akzeptieren.
⇨ button	Integer	Enthält eine Zahl, die angibt, welche Maustaste gedrückt ist: 1 (links), 2 (rechts) oder 4 (Mitte).
⇨ shift	Integer	Enthält eine Zahl, die den Zustand der Modifizierungstasten. Die gültigen Modifizierungstasten sind die UMSCHALT-, STRG- und ALT-Tasten, die den Zahlen 1, 2 und 4 entsprechen. Der Parameter Shift zeigt den Status dieser Tasten an; es können einige, alle oder keines der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise sowohl die STRG- als auch die ALT-Tasten gedrückt würden, wäre der Wert von Shift 6.
⇨ x	Long	Enthält die horizontale Position des Mauszeigers
⇨ y	Long	Enthält die vertikale Position des Mauszeigers
⇨ state	OLEDragStateEnum	Enthält eine Konstante, die die Richtung angibt, in die die Daten gezogen werden.

OLEGiveFeedback

Ereignis von VcTree

Dieses Ereignis tritt nach jedem OLEDragOver-Ereignis auf der Drop-Quellkomponente auf. OLEGiveFeedback gibt der Quellkomponente die Möglichkeit, dem Benutzer ein "visuelles Feedback" zu geben. Dies kann z. B. die Änderung des Cursors sein, um anzuzeigen, was passiert, wenn der Benutzer das Objekt ablegt; in anderen Fällen wird ein visuelles Feedback für die ausgewählten Objekte in der Quellkomponente gegeben, um anzuzeigen, was geschehen wird.

	Datentyp	Beschreibung
Parameter:		
⇨ effect	Long	Ein Wert, der die beim Ablegen der Daten auf das Drop-Ziel ausgeführte Aktion bezeichnet. Effect ist anfangs ein Wert, der die von der Drag-Quelle unterstützten OLE-Drag & Drop-Operationen bezeichnet. Die Zielkomponenente sollte diesen Wert prüfen und beim Ablegevorgang festlegen.
	Mögliche Werte:	
	vcDropEffectCopy 1	Eine Drop-Operation führt zum Kopieren der Daten vom Entnahme- zum Ablageort. Die ursprünglichen Daten werden durch die Drag-Operation nicht geändert.
	vcDropEffectLink 4	Es wurde eine Verknüpfung für Daten von der Drag-Quelle zum Drop-Ziel erstellt.
	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten vom Entnahme- zum Ablageort verschoben werden. Am Entnahmeort sollten die Daten nach erfolgtem Verschieben entfernt werden.
	vcDropEffectNone 0	Ablageziel kann die Daten nicht akzeptieren
⇦ defaultCursors	Boolean	Verwendung des Standard-Cursors (True) oder eines benutzerdefinierten Cursors (False)

Code-Beispiel

```
Private Sub VcTree1_OLEGiveFeedback(ByVal Effect As Long, _
                                    DefaultCursors As Boolean)
    If Effect <> vcOLEDropEffectNone Then
        'activate own mouse cursor
        MousePointer = vbCustom
        MouseIcon = LoadPicture("h_point.cur")
        DefaultCursors = False
    End If
End Sub
```

OLESetData

Ereignis von VcTree

Dieses Ereignis tritt bei einer Drag-Quelle auf, wenn ein Drop-Ziel die **GetData**-Methode aufruft und es in dem OLE Drag & Drop-DataObject keine Daten in einem festgelegten Format gibt.

	Datentyp	**Beschreibung**
Parameter:		
⇨ data	DataObject	DataObject, das bei der SetData-Methode verwendet wird, um die Daten dort abzulegen. Die Komponente ruft die SetData-Methode auf, um das entsprechende Format zu laden.
⇨ dataFormat	Integer	Ein numerischer oder Zeichenwert, der das Format der Daten bezeichnet, die die GetData-Methode verlangt. Die Drag-Quelle verwendet diesen Wert zur Bestimmung des Formats der im DataObject unterzubringenden Daten. Eine Tabelle, die die numerischen oder Zeichenwerte für das jeweilige Datenformat zusammen mit einer Beschreibung des Datenformats anzeigt, finden Sie unter **GetData**-Methode.

OLEStartDrag

Ereignis von VcTree

Dieses Ereignis tritt auf, wenn die Methode **OLEDrag** ausgeführt wird oder wenn eine VARCHART-XTree-Komponente eine OLE Drag & Drop-Operation auslöst und die **OLEDragMode**-Eigenschaft auf **vcOLEDrag-Automatic** gesetzt ist.

Dieses Ereignis legt die Datenformate und Ablege-Effekte fest, die die Quellkomponente unterstützt. Es kann auch zum Einfügen von Daten in das DataObject-Objekt verwendet werden.

Die Quellkomponente sollte den logischen Operator **Or** auf die unterstützten Werte anwenden und das Ergebnis in den Parameter **allowedEffect** übernehmen. Die Zielkomponente kann diesen Wert zum Bestimmen der passenden Aktion (und wie das geeignete Benutzer-Feedback aussehen soll) verwenden.

Das Hinzufügen von Daten in das **DataObject** sollten Sie eventuell verschieben, bis die entsprechende Anforderung von der Zielkomponente kommt. Dadurch kann die Zielkomponente Zeit sparen, weil nicht mehrere

Formate geladen werden müssen. Wenn das Ziel die **GetData**-Methode für das **DataObject** ausführt, tritt das **OLESetData**-Ereignis der Quelle auf, wenn die angeforderten Daten nicht im **DataObject** enthalten sind. Zu diesem Zeitpunkt können die Daten in das **DataObject** geladen werden, wodurch wiederum die Daten für das Ziel bereitgestellt werden.

Falls der Benutzer keine Formate in das **DataObject** lädt, wird die Drag&Drop-Operation abgebrochen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ data	DataObject	Objekt vom Typ **DataObject**, das Formate, die von der Quelle bereitgestellt werden, und möglicherweise die Daten für diese Formate enthält. Wenn das Objekt keine Daten enthält, werden sie aus der Quellkomponente geladen, wenn das Steuerelement die GetData-Methode aufruft.
⇔ allowedEffect	Long	Eine lange Integerzahl, die die von der Quellkomponente unterstützten Effekte enthält. Die Werte für diesen Parameter in dem Ereignis sollten vom Programmierer bereitgestellt werden.
	Mögliche Werte:	
	vcDropEffectCopy 1	Eine Drop-Operation führt zum Kopieren der Daten vom Entnahme- zum Ablageort. Die ursprünglichen Daten werden durch die Drag-Operation nicht geändert.
	vcDropEffectLink 4	Es wurde eine Verknüpfung für Daten von der Drag-Quelle zum Drop-Ziel erstellt.
	vcDropEffectMove 2	Die Drop-Operation führt dazu, dass Daten vom Entnahme- zum Ablageort verschoben werden. Am Entnahmeort sollten die Daten nach erfolgtem Verschieben entfernt werden.
	vcDropEffectNone 0	Ablageziel kann die Daten nicht akzeptieren

Code-Beispiel

```
Private Sub VcTree1_OLEStartDrag(ByVal data As VcTreeLib.DataObject, _
                                 allowedEffects As Long)
    allowedEffects = vbDropEffectCopy

    ' make sure that dragging is allowed only from one XTree control
    ' into another one
    data.SetData Empty, myOLEDragFormat
End Sub
```

OnBoxLClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Box klickt. Das getroffene VcBox-Objekt wird zusammen mit der Position des Mauszeigers (x,y-Koordinaten) als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ box	VcBox	Getroffene Box
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnBoxLClick(ByVal box As VcTreeLib.VcBox, _
                  ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    Text1.Text = box.FieldText(1)

End Sub
```

OnBoxLDblClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste auf eine Box doppelklickt. Das getroffene VcBox-Objekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ box	VcBox	Getroffene Box
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnBoxLDblClick(ByVal box As VcTreeLib.VcBox, _
                  ByVal x As Long, ByVal y As Long, returnStatus As Variant)
    box.FieldText(0) = Text1.Text
End Sub
```

OnBoxModifyComplete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn die Modifizierung der Box abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇨ box	VcBox	Veränderte Box

Code-Beispiel

```
Private Sub VcTree1_OnBoxModifyComplete(ByVal box As _
                  VcTreeLib.VcBox)

    MsgBox "The box has been modified."

End Sub
```

OnBoxModifyCompleteEx

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn die Modifizierung der Box abgeschlossen ist. Das veränderte VcBox-Objekt und der Modifikationstyp werden als Parameter mitgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ modificationType	BoxModificationTypeEnum	Art der Veränderung
	Mögliche Werte:	
	vcBMTAnything 1	beliebige Veränderung
	vcBMTNothing 0	keine Veränderung
	vcBMTTextModified 4	Text der Box verändert
	vcBMTXYOffsetModified 2	Offset verändert

Code-Beispiel

```
Private Sub VcTree1_OnBoxModifyCompleteEx(ByVal box As _
                  VcTreeLib.VcBox)

    MsgBox "The box has been modified."

End Sub
```

OnBoxRClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste auf eine Box klickt. Das getroffene Boxobjekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter zurückgegeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇨ box	VcBox	getroffene Box

⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnBoxRClick(ByVal box As VcTreeLib.VcBox, _
                  ByVal x As Long, ByVal y As Long, returnStatus As Variant)

    ' Start own popup menu at the current mouse cursor position
    PopupMenu mnuBoxPopup

End Sub
```

OnDataRecordCreate

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender interaktiv ein Objekt erzeugt hat, das einen Datensatz generiert. Das neu erzeugte Datensatz-Objekt wird als Parameter zurückgegeben, so dass eine Validierung vorgenommen werden kann.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnDataRecordCreateComplete**.

Durch Setzen des Rückgabestatus kann die Anlage verhindert werden.

Wenn eine Verbindung oder ein Knoten angelegt wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNodeCreate** und **OnLinkCreate**).

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Angelegter Datensatz
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0 vcRetStatOK 1	 Der Datensatz wird nicht angelegt. Der Datensatz wird angelegt.

Code-Beispiel

```
Private Sub VcTree1_OnDataRecordCreate(ByVal node As VcTreeLib.VcDataRecord, _
                                       returnStatus As Variant)
```

```
    'Show own "Edit" dialog for the new data record
    '(EditNewDataRecord attribute must be set to off!)
    On Error GoTo CancelError
    frmEditDialog.Show Modal, Me

    addDataRecord dataRecord.AllData

    Exit Sub

CancelError:
    returnStatus = vcRetStatFalse

End Sub
```

OnDataRecordCreateComplete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn das interaktive Anlegen eines Objektes beendet ist, das einen Datensatz erzeugt. Das DataRecord-Objekt, die Form des Anlegens (hier nur **vcDataRecordCreated** und **vcDataRecordCreated-ByResourceScheduling**) und die Information, ob der angelegte Datensatz der einzige Datensatz oder der letzte einer Menge ist (derzeit immer **True**), werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

Wenn eine Verbindung oder ein Knoten angelegt wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNode-CreateComplete** und **OnLinkCreateComplete**).

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Angelegter Datensatz
⇨ creationType	CreationTypeEnum	Typ des Anlegens
	Mögliche Werte:	
	vcDataRecordCreated 6	Datensatz wurde durch Interaktion angelegt
	vcDataRecordCreatedByResourceScheduling 5	Datensatz wurde automatisch durch Ressourcenplanung angelegt
	vcNodeCreated 1	Knoten durch "Stempeln" angelegt
	vcNodesCloned 4	selektierte Knoten wurden durch Ziehen mit der Maus bei gleichzeitigem Drücken der Strg-Taste kopiert

⇨ isLastNodeInSeries	Boolean	**True**: Angelegter Datensatz ist der einzige oder der letzte Datensatz einer Menge **False**: Angelegter Datensatz ist nicht der einzige oder der letzte Datensatz einer Menge

Code-Beispiel

```
Private Sub VcTree1_OnDataRecordCreateComplete(ByVal dataRecord As _
                                VcTreeLib.VcDataRecord, ByVal creationType As _
                                VcTreeLib.CreationTypeEnum, _
                                ByVal isLastDataRecordInSeries As Boolean)
   addDataRecord dataRecord.AllData
End Sub
```

OnDataRecordDelete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs ein Objekt löscht, das auf einem Datensatz-Objekt basiert. Der betroffene Datensatz wird als Parameter zurückgegeben, so dass Sie z. B. noch eine Überprüfung vornehmen und bei negativem Ergebnis dieser Prüfung die Löschung ggf. durch Setzen des Rückgabestatus verhindern können.

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Gelöschter Datensatz
⇔ returnStatus	Variant	Rückgabestatus
	Mögliche Werte:	
	vcRetStatFalse 0	Der Datensatz wird nicht gelöscht.
	vcRetStatOK 1	Der Datensatz wird gelöscht.

Code-Beispiel

```
Private Sub VcTree1_OnDataRecordDelete(ByVal node As VcTreeLib.VcNode, _
                                returnStatus As Variant)
  'deny the deletion of the last data record in the chart
    If VcTree1.DataRecordCollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last data record cannot be deleted.")
    End If
End Sub
```

OnDataRecordDeleteComplete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn das Löschen eines Objektes, das auf einem Datensatz-Objekt basiert, beendet ist. Der Datensatz und die Information, ob

der betroffene Datensatz der einzige oder der letzte Datensatz einer Menge ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

Wenn eine Verbindung oder ein Knoten gelöscht wurde, können Sie zudem auf das analoge Verbindungs- oder Knoten-Ereignis reagieren und hier vor der Weiterverarbeitung zusätzlich grafische Daten prüfen (s. **OnNodeDelete-Complete** und **OnLinkDeleteComplete**).

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Gelöschter Datensatz
⇨ isLastNodeInSeries	Boolean	**True**: Gelöschter Datensatz ist der einzige oder der letzte Datensatz einer Menge **False**: Gelöschter Datensatz ist nicht der einzige oder der letzte Datensatz einer Menge

OnDataRecordModify

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn ein Objekt interaktiv verändert wurde und sich der zu Grunde liegende Datensatz verändert. Das veränderte VcDataRecord-Objekt und der Modifikationstyp werden als Parameter zurückgegeben.

Die mit diesem Ereignis übermittelten Daten dürfen nur gelesen, aber nicht verändert werden. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnDataRecordModifyComplete**.

Durch Setzen des Rückgabestatus kann die Änderung verhindert werden.

	Datentyp	**Beschreibung**
Parameter:		
⇨ dataRecord	VcBox	Veränderte Box
⇨ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcAnything 1 vcMoved 8 vcNothing 0	 Änderungstyp nicht näher bestimmt Objekt wurde verschoben keine Änderung
⇦ returnStatus	Variant	Rückgabestatus
	Mögliche Werte: vcRetStatFalse 0	 Die Veränderung wird rückgängig gemacht.

vcRetStatOK 1	Die Veränderung wird durchgeführt.

OnDataRecordModifyComplete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn die Modifizierung des Datensatzes abgeschlossen ist.

	Datentyp	Beschreibung
Parameter:		
⇨ dataRecord	VcDataRecord	Veränderter Datensatz

Code-Beispiel

```
Private Sub VcTree1_OnDataRecordModifyComplete(ByVal box As _
                  VcTreeLib.VcBox)

    MsgBox "The data record has been modified."

End Sub
```

OnDataRecordNotFound

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn ein abhängiger Datensatz nicht gefunden wurde. Der Index des Feldes im aktuellen Datensatz, in dem der Schlüsselwert des abhängigen Datensatzes steht, wird zurückgegeben und bietet so Informationen über den nicht gefundenen Datensatz.

	Datentyp	Beschreibung
Parameter:		
⇦ index	Long	Index des Feldes, das den Schlüssel des abhängigen Datensatzes enthält

OnDiagramLClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste im Diagrammbereich klickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten des Cursors) wird als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnDiagramLClick(ByVal x As Long, _
                                    ByVal y As Long, returnStatus As Variant)
    Dim zoomfactor As Integer

    zoomfactor = VcTree1.Zoomfactor + 10
    VcTree1.Zoomfactor = zoomfactor
End Sub
```

OnDiagramLDblClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der linken Maustaste im Diagrammbereich doppelklickt und dabei kein Objekt trifft. Die Position (x,y-Koordinaten des Mauszeigers) wird als Parameter zurückgegeben.

	Datentyp	Beschreibung
Parameter:		
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnDiagramLDblClick(ByVal x As Long, _
                                    ByVal y As Long, returnStatus As Variant)

    Dim zoomfactor As Integer

    zoomfactor = VcTree1.Zoomfactor - 10
    VcTree1.Zoomfactor = zoomfactor

End Sub
```

OnDiagramRClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit der rechten Maustaste in den Diagrammbereich klickt und dabei weder Stichwertlinie noch Knoten trifft. Die Position (x,y-Koordinaten) wird als Parameter übergeben. Das

integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	Beschreibung
Parameter:		
⇨ x	Long	X-Koordinate
⇨ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnDiagramRClick(ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    If MsgBox("Vertical from level 1?", vbYesNo, "First vertical level?") _
                               = vbYes Then
        VcTree1.FirstVerticalLevel = 1
        VcTree1.Arrange
    End If

    returnStatus = vcRetStatNoPopup

End Sub
```

OnHelpRequested

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender in einem zur Laufzeit angezeigten Dialog die **F1**-Taste drückt. Die Applikation erhält damit die Möglichkeit, ihr eigenes Hilfesystem aufzurufen, um dialog- und anwendungsbezogene Hilfe anbieten zu können.

	Datentyp	Beschreibung
Parameter:		
⇦ dialogType	DialogTypeEnum	Dialog, für den Hilfe angefordert wird
	Mögliche Werte:	
	vcEditDataRecordDialog 5400	Hilfe wurde für den **Datensatz bearbeiten** Dialog angefordert
	vcPageSetupDialog 4097	Hilfe wurde für den **Seite einrichten** Dialog angefordert
	vcPrintPreviewDialog 4096	Hilfe wurde für den **Druckvorschau** Dialog angefordert

OnLegendViewClosed

Ereignis von VcTree

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Legendenansicht geschlossen wird.

	Datentyp	Beschreibung
Parameter: ⇨ (no parameter)		

Code-Beispiel

```
Private Sub VcTree1_OnLegendViewClosed()
    MsgBox "Do you want to close the legend view window?", vbOKCancel
End Sub
```

OnModifyComplete

Ereignis von VcTree

Dieses Ereignis tritt immer dann auf, wenn Daten interaktiv im Chart verändert werden, also explizit nach den folgenden Ereignissen:

- OnBoxModifyComplete
- OnNodeCreateCompleteEx
- OnNodeDelete
- OnNodeModifyComplete

Mit diesem Ereignis ist es möglich, sich im Anwenderprogramm eine Marke zu setzen, die daran erinnert, dass beim Beenden des Programms die Daten noch gespeichert werden müssen.

	Datentyp	Beschreibung
Parameter: ⇦ (no parameter)		Kein Parameter

OnMouseDblClk

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender eine Maustaste doppelklickt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇨ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: **1** (links), **2** (rechts) oder **4** (Mitte).
⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers

OnMouseDown

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender auf eine Maustaste klickt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇨ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: **1** (links), **2** (rechts) oder **4** (Mitte).
⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers

OnMouseMove

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender die Maus bewegt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇨ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: **1** (links), **2** (rechts) oder **4** (Mitte).
⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers

OnMouseUp

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender eine gedrückte Maustaste wieder loslässt.

Bitte beachten Sie auch die Eigenschaft **MouseProcessingEnabled**.

	Datentyp	Beschreibung
Parameter:		
⇨ button	Integer	Zahl, die angibt, welche Maustasten gedrückt sind: **1** (links), **2** (rechts) oder **4** (Mitte).

⇨ Shift	Integer	Eine Ganzzahl, die dem Zustand der Tasten UMSCHALT, STRG und ALT zu dem Zeitpunkt entspricht, an dem das Ereignis aufgetreten ist. Das Argument Shift ist ein Bitfeld, bei dem die niederstwertigen Bits der UMSCHALT-Taste (Bit 0), der STRG-Taste (Bit 1) und der ALT-Taste (Bit 2) entsprechen. Diese Bits entsprechen jeweils dem Wert 1, 2 und 4. Einige, alle oder keine dieser Bits können gesetzt werden, um anzuzeigen, dass einige, alle oder keine der Tasten gedrückt sind. Wenn zum Beispiel sowohl STRG als auch ALT gedrückt werden, hat shift den Wert 6.
⇨ x	Long	X-Koordinate des Mauszeigers
⇨ y	Long	Y-Koordinate des Mauszeigers

OnNodeCollapse

Ereignis von VcTree

Dieses Ereignis wird aufgerufen, wenn ein Knoten durch eine Benutzeraktion kollabiert wird. Steht im Parameter **action** der Wert **vcSelf**, so ist der angegebene, nun kollabierte Knoten auch der selektierte. Mit ihm wird die Operation per Kontextmenü durchgeführt. Steht im Parameter **action** der Wert **vcComplete**, befindet sich der selektierte Knoten in dem Teilbaum unter dem Knoten.

	Datentyp	**Beschreibung**
Parameter:		
⇨ action	CollapseExpandEnum	Typ des Kollabierens
	Mögliche Werte:	
	vcComplete 1	einschließlich der Knoten im untergeordneten Teilbaum
	vcSelf 0	ausschließlich der Knoten im untergeordneten Teilbaum
⇨ node	VcNode	Knotenobjekt
⇔ returnStatus	Variant	Rückgabestatus

OnNodeCreate

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten erzeugt. Das Knotenobjekt wird als Parameter übergeben, so dass eine Datenvalidierung vorgenommen werden kann. (Der Dialog **Daten bearbeiten** wurde ggf.

vorher über die Eigenschaft **VcTree.EditNewNode** aktiviert.) Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten nicht erzeugt.

Dieses Ereignis sollte nur verwendet werden, um Daten des aktuellen Knotens auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeCreateCompleteEx**.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	anzulegender Knoten
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeCreate(ByVal node As VcTreeLib.VcNode, _
                                returnStatus As Variant)
    'show own edit dialog for the new node
    '  (EditNewNodes attribute must be set to off!)
    On Error GoTo CancelError
    frmEditDialog.Show Modal, Me

    'create a record in the underlying database of the application
    addDataRecord node.AllData

    Exit Sub

CancelError:
    returnStatus = vcRetStatFalse
End Sub
```

OnNodeCreateCompleteEx

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn das interaktive Anlegen eines Knotens abgeschlossen ist. Das Knotenobjekt, der Typ des Knotenanlegens und die Information, ob der angelegte Knoten der einzige Knoten oder der letzte Knoten einer Menge ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Angelegter Knoten
⇨ creationType	CreationTypeEnum	Typ des Knotenanlegens
	Mögliche Werte:	
	vcDataRecordCreated 6	Datensatz wurde durch Interaktion angelegt
	vcDataRecordCreatedByResourceScheduling 5	Datensatz wurde automatisch durch Ressourcenplanung angelegt

		vcNodeCreated 1	Knoten durch "Stempeln" angelegt
		vcNodesCloned 4	selektierte Knoten wurden durch Ziehen mit der Maus bei gleichzeitigem Drücken der Strg-Taste kopiert
⇨ isLastNodeInSeries	Boolean		Angelegter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge

Code-Beispiel

```
Private Sub VcTree1_OnNodeCreateCompleteEx(ByVal node As _
                                VcTreeLib.VcNode, ByVal creationType As _
                                VcTreeLib.CreationTypeEnum, _
                                ByVal isLastNodeInSeries As Boolean)
   'create a record in the underlying database of the application
   addDataRecord node.AllData
End Sub
```

OnNodeDelete

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender mit Hilfe des Kontextmenüs einen Knoten löscht. Der betroffene Knoten wird als Parameter zurückgegeben, so dass Sie z. B. noch eine Überprüfung vornehmen und bei negativem Ergebnis dieser Prüfung die Löschung ggf. verhindern können. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Löschvorgang verhindert; durch Setzen auf **vcRetStatOK** wird die Löschung durchgeführt; bei **vcRetStatDefault** bleibt das vordefinierte Default-Verhalten unverändert und die Löschung wird ebenfalls durchgeführt, und bei **vcRetStatPopup** wird das Kontext-Menü aufgerufen, über das Sie dem Benutzer weitere Interaktionen anbieten können.

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Knotenobjekt
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnNodeDelete(ByVal node As VcTreeLib.VcNode, _
                                 returnStatus As Variant)
   'deny the deletion of the last node in the chart
    If VcTree1.nodecollection.Count = 1 Then
        returnStatus = vcRetStatFalse
        MsgBox ("The last node in the chart cannot be deleted.")
    End If
End Sub
```

OnNodeDeleteCompleteEx

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn das interaktive Löschen eines Knotens abgeschlossen ist. Das Knotenobjekt und die Information, ob der gelöschte Knoten der zuletzt gelöschte einer Menge ist, werden als Parameter zurückgegeben, so dass eine Datenvalidierung vorgenommen werden kann.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Gelöschter Knoten
⇨ isLastNodeInSeries	Boolean	Gelöschter Knoten ist (True) / ist nicht (False) letzte Knoten einer Menge

OnNodeExpand

Ereignis von VcTree

Dieses Ereignis wird aufgerufen, wenn ein Knoten durch eine Benutzeraktion expandiert wird. Steht im Parameter **action** der Wert **vcSelf**, so ist der angegebene, nun kollabierte Knoten auch der selektierte. Mit ihm wird die Operation per Kontextmenü durchgeführt. Steht hier **vcComplete**, gehört der angegebene Knoten zu der Menge der Sohnknoten des selektierten Knotens. Wird der Rückgabestatus auf **vcRetStatFalse** gesetzt, wird die Aktion abgebrochen und die angegebenen Knoten werden nicht expandiert.

	Datentyp	Beschreibung
Parameter:		
⇨ action	CollapseExpandEnum	Typ des Kollabierens
	Mögliche Werte:	
	vcComplete 1	einschließlich der Knoten im untergeordneten Teilbaum
	vcSelf 0	ausschließlich der Knoten im untergeordneten Teilbaum
⇨ node	VcNode	Knotenobjekt
⇔ returnStatus	Variant	Rückgabestatus

OnNodeLClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der linken Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Knotenobjekt
⇨ location	LocationEnum	Bereich im Chart
	Mögliche Werte: vcInDiagram 1	im Knotenbereich
⇨ x	Long	X-Koordinate
⇨ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnNodeLClick(ByVal node As VcTreeLib.VcNode, _
                                 ByVal location As VcTreeLib.LocationEnum, _
                                 ByVal x As Long, ByVal y As Long, _
                                 returnStatus As Variant)
    'change data field of the node
    node.DataField(10) = 1 - CInt(node.DataField(10))
End Sub
```

OnNodeLDblClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der linken Maustaste doppelt klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) des Mauszeigers als Parameter übergeben. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der integrierte Dialog **Vorgänge bearbeiten** unterdrückt.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	Knotenobjekt
⇨ x	Long	X-Koordinate
⇨ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnNodeLDblClick(ByVal node As VcTreeLib.VcNode, _
                                    ByVal location As VcTreeLib.LocationEnum, _
                                    ByVal x As Long, ByVal y As Long, _
                                    returnStatus As Variant)

    If node.RightBrotherNode Is Nothing Then
        MsgBox "No right brother"
    Else
        MsgBox (node.RightBrotherNode.AllData)
    End If

    returnStatus = vcRetStatFalse

End Sub
```

OnNodeModifyCompleteEx

Ereignis von VcTree

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer die Knotenhierarchie geändert hat.

	Datentyp	Beschreibung
Parameter:		
⇨ node	VcNode	veränderter Knoten
⇨ isLastNodeInSeries	Boolean	veränderter Knoten ist/ist nicht der einzige Knoten bzw. der letzte Knoten einer Menge
⇨ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte:	
	vcAnything 1	Änderungstyp nicht näher bestimmt
	vcMoved 8	Objekt wurde verschoben
	vcNothing 0	keine Änderung

OnNodeModifyEx

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender interaktiv einen Knoten verändert. Dabei kann der Knoten verschoben oder ein Wert im Dialogfeld **Vorgänge bearbeiten** verändert worden sein. Die Daten des Knotens vor und nach der Veränderung werden als Parameter zurückgegeben. Über den Parameter **modificationType** erhalten Sie nähere Informationen über die Art der Veränderung. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** wird der Knoten nicht verändert.

Dieses Ereignis sollte nur verwendet werden, um Daten des aktuellen Knotens auszulesen. Um sie zu verändern, verwenden Sie bitte das Ereignis **OnNodeModifyComplete**.

	Datentyp	**Beschreibung**
Parameter:		
⇨ oldNode	VcNode	Knoten vor der Veränderung
⇨ node	VcNode	Zu verändernder Knoten
⇨ modificationType	ModificationTypeEnum	Art der Veränderung
	Mögliche Werte: vcAnything 1 vcMoved 8 vcNothing 0	 Änderungstyp nicht näher bestimmt Objekt wurde verschoben keine Änderung
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnNodeModifyEx(ByVal oldNode As _
                                   VcTreeLib.VcNode, ByVal node As _
                                   VcTreeLib.VcNode, ByVal modificationType As _
                                   VcTreeLib.ModificationTypeEnum, returnStatus _
                                   As Variant)

    ' Revoke the modification if the node would change the group
    If modificationType And vcChangedGroup Then
        MsgBox "The node cannot be moved into another group."
        returnStatus = vcRetStatFalse
    End If

End Sub
```

OnNodeRClick

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der Anwender auf einen Knoten mit der rechten Maustaste klickt. Das getroffene Knotenobjekt wird zusammen mit der Position (x,y-Koordinaten) als Parameter übergeben. Das integrierte Kontextmenü kann durch Setzen des Rückgabestatus unterdrückt werden. Sie können so an der entsprechenden Position Ihr gewünschtes Kontextmenü anzeigen.

	Datentyp	**Beschreibung**
Parameter:		
⇨ node	VcNode	Knotenobjekt
⇨ location	LocationEnum	Bereich im Chart
	Mögliche Werte: vcInDiagram 1	 im Knotenbereich

⇨ x	Long	Y-Koordinate
⇨ y	Long	Y-Koordinate
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcNet1_OnNodeRClick(ByVal node As VcTreeLib.VcNode, _
                                ByVal location As VcTreeLib.LocationEnum, _
                                ByVal x As Long, ByVal y As Long, _
                                returnStatus As Variant)
    ' start a popup menu at the current mouse cursor position
    PopupMenu mnuNodePopup

    returnStatus = vcRetStatNoPopup
End Sub
```

OnNodesMarkComplete

Ereignis von VcTree

Mit diesem Ereignis wird das Ende einer Markier- oder Demarkieroperation von Knoten angezeigt.

	Datentyp	Beschreibung
Parameter:		
⇦ (no parameter)		Kein Parameter

Code-Beispiel

```
Private Sub VcTree1_OnNodesMarkComplete()
    MsgBox "Nodes have been successfully marked."
End Sub
```

OnNodesMarkEx

Ereignis von VcTree

Mit diesem Ereignis wird bekanntgegeben, dass der Benutzer einen oder mehrere Knoten zum Markieren oder zur Aufhebung der Markierung ausgewählt hat. In der nodeCollection sind diese Knoten verzeichnet. Die Parameter **button** und **shift** geben an, welche Steuer- und Maustasten gedrückt wurden. Wenn man den Rückgabestatus auf **vcRetStatFalse** setzt, wird das Markieren oder die Aufhebung der Markierung verhindert.

	Datentyp	Beschreibung
Parameter:		
⇨ nodeCollection	VcNodeCollection	NodeCollection, die die vom Anwender selektierten Knoten enthält. Wenn in das Diagramm geklickt wurde, ist die Collection leer.

⇨ button	Integer	Zahl, die angibt, auf welche Weise markiert wurde: **0**: über die Tastatur, **1**: linke Maustaste, **2**: rechte Maustaste, **4**: mittlere Maustaste
⇨ shift	Integer	Zahl, die den Zustand der Modifizierungstasten zu dem Zeitpunkt angibt, zu dem Daten über das Drop-Ziel gezogen werden. Die gültigen Modifizierungstasten sind die <UMSCHALT>-, <STRG>- und <ALT>-Tasten, die den Zahlen **1**, **2** und **4** entsprechen. Der Parameter **shift** zeigt den Status dieser Tasten an; es können einige, alle oder keines der drei Zahlen gesetzt werden, was jeweils anzeigt, dass einige, alle oder keine der Tasten gedrückt wird. Wenn beispielsweise die <STRG>- und die <ALT>-Tasten gedrückt werden, ist der Wert von **shift** "6".
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnNodesMarkEx(ByVal NodeCollection As _
                                  VcTreeLib.VcNodeCollection, _
                                  ByVal button As Integer, _
                                  ByVal shift As Integer, _
                                  returnStatus As Variant)
    If MsgBox("Mark this node?", vbYesNo, "Marking nodes") = _
    vbNo Then returnStatus = vcRetStatFalse
End Sub
```

OnSelectField

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn ein Feld einer Box selektiert wird. Die Selektion kann durch Setzen des Rückgabestatus verhindert werden.

	Datentyp	**Beschreibung**
Parameter:		
editObject	Object	
editObjectType	VcObjectTypeEnum	
	Mögliche Werte:	
	vcObjTypeBox 15	Objekttyp **Box**
	vcObjTypeNode 2	Objekttyp **Knoten**
	vcObjTypeNodeInLegend 17	Objekttyp **Knoten im Legendenbereich**
	vcObjTypeNone 0	kein Objekt
fieldIndex	Long	
objRectComplete	VcRect	
objRectVisible	VcRect	
fldRectComplete	VcRect	
fldRectVisible	VcRect	
returnStatus	Variant	

OnShowInPlaceEditor

Ereignis von VcTree

Dieses Ereignis tritt ein, wenn der im Programm implementierte Editor gestartet wird.

Das Ereignis wird erst aktiviert, wenn diie entsprechenden Eigenschaften **InPlaceEditingOnGroupsInDiagramEnabled, InPlaceEditingOnGroupsInTableEnabled, InPlaceEditingOnNodesInDiagramEnabled, InPlaceEditingOnNodesInTableEnabled** auf True gesetzt sind.

Durch setzen des Rückgabestatus auf **False** kann dieses Ereignis verhindert werden, so dass an den übergebenen Koordinaten ein eigener Editor aufgerufen werden kann.

	Datentyp	**Beschreibung**
Parameter:		
⇨ editObject	Object	editiertes Objekt
⇨ editObjectType	VcObjectTypeEnum	Objekttyp
	Mögliche Werte: vcObjTypeBox 15 vcObjTypeNode 2 vcObjTypeNodeInLegend 17 vcObjTypeNone 0	 Objekttyp **Box** Objekttyp **Knoten** Objekttyp **Knoten im Legendenbereich** kein Objekt
⇨ fieldIndex	Long	Feldindex
⇨ objRectComplete	VcRect	Komplettes Rechteck des getroffenen Objekts
⇨ objRectVisible	VcRect	Sichtbares Rechteck des getroffenen Objekts
⇨ fldRectComplete	VcRect	Komplettes Rechteck des getroffenen Feldes
⇨ fldRectVisible	VcRect	sichtbares Rechteck des getroffenen Feldes
returnStatus	Variant	

Code-Beispiel

```
Private Sub VcTree1_OnShowInPlaceEditor(ByVal editObject As Object, _
                 ByVal editObjectType As VcTreeLib.VcObjectTypeEnum, _
                 ByVal fieldIndex As Long, ByVal objRectComplete As _
                 VcTreeLib.VcRect, ByVal objRectVisible As _
                 VcTreeLib.VcRect, ByVal fldRectComplete As _
                 VcTreeLib.VcRect, ByVal fldRectVisible As _
                 VcTreeLib.VcRect, returnStatus As Variant)

   Dim oldScaleMode As Long

   If editObjectType = vcObjTypeNode Then
      returnStatus = vcRetStatFalse

      Set myEditObject = editObject
      myEditObjectType = editObjectType
      myEditObjectFieldIndex = fieldIndex
      oldScaleMode = Me.ScaleMode
      Me.ScaleMode = vbPixels
```

```
      Select Case fieldIndex
         Case 1   'Name
            Text1.Left = fldRectVisible.Left + VcTree1.Left
            Text1.Top = fldRectVisible.Top + VcTree1.Top
            Text1.Width = fldRectVisible.Width
            Text1.Height = fldRectVisible.Height
            Text1.Text = editObject.DataField(fieldIndex)
            Text1.Visible = True
            Text1.SetFocus

         Case 2, 3     'Start or End
            MonthView1.Left = fldRectVisible.Left + VcTree1.Left
            MonthView1.Top = fldRectVisible.Top + VcTree1.Top
            MonthView1.Value = editObject.DataField(fieldIndex)
            MonthView1.Visible = True
            MonthView1.SetFocus

         Case 13   'Employee
            Combo1.Left = fldRectVisible.Left + VcTree1.Left
            Combo1.Top = fldRectVisible.Top + VcTree1.Top
            Combo1.Width = fldRectVisible.Width
            Combo1.Text = editObject.DataField(fieldIndex)
            Combo1.Visible = True
            Combo1.SetFocus

      End Select

      Me.ScaleMode = oldScaleMode

   End If

End Sub
```

OnStatusLineText

Ereignis von VcTree

Dieses Ereignis tritt immer dann ein, wenn eine Information von allgemeiner Bedeutung bereitgestellt wird. Das kann ein funktionaler Hinweis sein (z. B. beim Laden), aber auch die ID des Knotens, auf dem der Cursor gerade steht.

	Datentyp	**Beschreibung**
Parameter:		
⇨ text	String	Text

Code-Beispiel

```
Private Sub VcNet1_OnStatusLineText(ByVal Text As String)
    'show text on status bar
    txtStatusBar.Text = Text
End Sub
```

OnSupplyTextEntry

Ereignis von VcTree

Dieses Ereignis tritt nur auf, wenn Sie die VcTree-Eigenschaft **EnableSupplyTextEntryEvent** auf **True** gesetzt haben. Das Ereignis tritt auf, wenn ein Text ausgegeben werden soll. Sie können hier alle vorgegebenen Texte durch eigene Texte ersetzen, z. B. um sie in unterschiedliche Sprachen zu übersetzen. Das betrifft die Kontextmenüs, Dialogfelder, Infoboxen, Fehlermeldungen und Monats- und Tagesnamen.

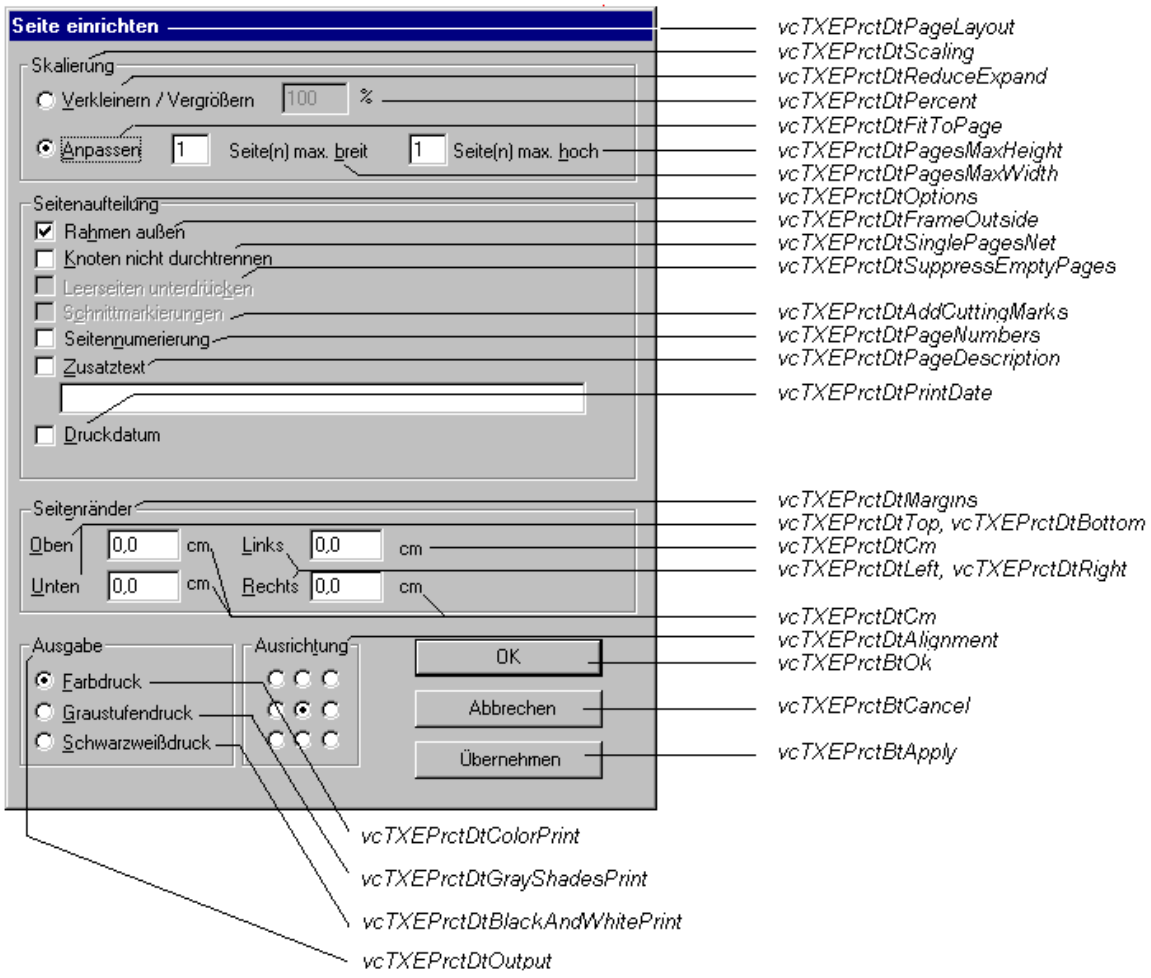
	Datentyp	**Beschreibung**
Parameter:		
⇨ controlIndex	TextEntryIndexEnum	Zu ersetzender Text
	Mögliche Werte:	
	vcTXECtxmenArrowMode 2116	Text im Kontextmenü: **Selektier-Modus**
	vcTXECtxmenCollapse 2184	Text im Kontextmenü: **Kollabieren**
	vcTXECtxmenCopyNodes 2152	Text im Kontextmenü: **Knoten kopieren**
	vcTXECtxmenCreateNodeMode 2117	Text im Kontextmenü: **Knoten erzeugen**
	vcTXECtxmenCutNodes 2151	Text im Kontextmenü: **Knoten ausschneiden**
	vcTXECtxmenDeleteNode 2101	Text im Kontextmenü: **Knoten löschen**
	vcTXECtxmenEditNode 2100	Text im Kontextmenü: **Daten bearbeiten**
	vcTXECtxmenExpand 2185	Text im Kontextmenü: **Expandieren**
	vcTXECtxmenExpandComplete 2186	Text im Kontextmenü: **Vollständig expandieren**
	vcTXECtxmenFilePrint 2122	Text im Kontextmenü: **Drucken**
	vcTXECtxmenFilePrintPreview 2121	Text im Kontextmenü: **Druckvorschau**
	vcTXECtxmenFilePrintSetup 2120	Text im Kontextmenü: **Drucker einrichten**
	vcTXECtxmenFirstChild 2182	Text im Kontextmenü: **Knoten einfügen als ersten Sohnknoten**
	vcTXECtxmenFullDiagram 2156	Text im Kontextmenü **Gesamtbaum wiederherstellen**
	vcTXECtxmenGraphicExport 2123	Text im Kontextmenü: **Grafik exportieren**
	vcTXECtxmenHorizontal 2187	Text im Kontextmenü: **Horizontal**
	vcTXECtxmenHorizontalComplete 2188	Text im Kontextmenü: **Horizontal vollständig**
	vcTXECtxmenLastChild 2182	Text im Kontextmenü: **Knoten einfügen als letzten Sohnknoten**
	vcTXECtxmenPageLayout 2119	Text im Kontextmenü: **Seite einrichten**
	vcTXECtxmenPasteNodesAfter 2180	Text im Kontextmenü: **Knoten einfügen hinter**
	vcTXECtxmenPasteNodesBefore 2181	Text im Kontextmenü: **Knoten einfügen vor**
	vcTXECtxmenPasteNodesFirstChild 2182	Text im Kontextmenü: **Knoten als ersten Sohnknoten einfügen**
	vcTXECtxmenPasteNodesLastChild 2182	Text im Kontextmenü: **Knoten als letzten Sohnknoten einfügen**
	vcTXECtxmenShowLegendView 2158	Text im Kontextmenü **Legendenansicht anzeigen**
	vcTXECtxmenShowWorldView 2157	Text im Kontextmenü **Komplettansicht anzeigen**

vcTXECtxmenSubDiagram 2155	Text im Kontextmenü **Teilbaum erstellen**
vcTXECtxmenVertical 2189	Text im Kontextmenü: **Vertikal**
vcTXEDateAM 2225	Ausgabetext für **vormittags**
vcTXEDateCW 2223	Ausgabetext für **Kalenderwoche**
vcTXEDateDay0 2212	Ausgabetext für **Montag**
vcTXEDateDay1 2213	Ausgabetext für **Dienstag**
vcTXEDateDay2 2214	Ausgabetext für **Mittwoch**
vcTXEDateDay3 2215	Ausgabetext für **Donnerstag**
vcTXEDateDay4 2216	Ausgabetext für **Freitag**
vcTXEDateDay5 2217	Ausgabetext für **Samstag**
vcTXEDateDay6 2218	Ausgabetext für **Sonntag**
vcTXEDateMonth0 2200	Ausgabetext für **Januar**
vcTXEDateMonth1 2201	Ausgabetext für **Februar**
vcTXEDateMonth10 2210	Ausgabetext für **November**
vcTXEDateMonth11 2211	Ausgabetext für **Dezember**
vcTXEDateMonth2 2202	Ausgabetext für **März**
vcTXEDateMonth3 2203	Ausgabetext für **April**
vcTXEDateMonth4 2204	Ausgabetext für **Mai**
vcTXEDateMonth5 2205	Ausgabetext für **Juni**
vcTXEDateMonth6 2206	Ausgabetext für **Juli**
vcTXEDateMonth7 2207	Ausgabetext für **August**
vcTXEDateMonth8 2208	Ausgabetext für **September**
vcTXEDateMonth9 2209	Ausgabetext für **Oktober**
vcTXEDateOClock 2224	Ausgabetext für **Uhr**
vcTXEDatePM 2226	Ausgabetext für **nachmittags**
vcTXEDateQuarter0 2219	Ausgabetext für **1. Quartal**
vcTXEDateQuarter1 2220	Ausgabetext für **2. Quartal**
vcTXEDateQuarter2 2221	Ausgabetext für **3. Quartal**
vcTXEDateQuarter3 2222	Ausgabetext für **4. Quartal**
vcTXEDlgLegArrangement 2046	Text im Dialog **Legendenattribute**: **Anordnung**
vcTXEDlgLegBottomMargin 2052	Text im Dialog **Legendenattribute**: **Unterer Rand:**
vcTXEDlgLegFixedToColumns 2048	Text im Dialog **Legendenattribute**: **nach Spaltenanzahl**
vcTXEDlgLegFixedToRows 2047	Text im Dialog **Legendenattribute**: **nach Zeilenanzahl**
vcTXEDlgLegFixedToRowsAndColumns 2049	Text im Dialog **Legendenattribute**: **nach Zeilen- und Spaltenanzahl**
vcTXEDlgLegIdcancel 2042	Schaltfläche im Dialog **Legendenattribute**: **Abbrechen**
vcTXEDlgLegIdd 2040	Dialog Legendenattribute Beschriftung der Titelzeile
vcTXEDlgLegIdok 2041	Schaltflächentext im Dialog **Legendenattribute**: **OK**
vcTXEDlgLegLegendElements 2045	Text im Dialog **Legendenattribute**: **Legendenelemente**
vcTXEDlgLegLegendFont 2053	Schaltfläche im Dialog **Legendenattribute**: **Schriftart...** für Legende
vcTXEDlgLegLegendTitleFont 2044	Schaltfläche im Dialog **Legendenattribute**: **Schriftart...** für Legendentitel
vcTXEDlgLegLegendTitleVisible 2043	Text im Dialog **Legendenattribute**: **Legendentitel sichtbar**
vcTXEDlgLegMargins 2050	Text im Dialog **Legendenattribute**: **Ränder**
vcTXEDlgLegTopMargin 2051	Text im Dialog **Legendenattribute**: **Oberer Rand:**
vcTXEDlgNedCaptionPrefix 2024	Dialog **Vorgänge bearbeiten**,: Text für Beschriftungszeile: “Knoten”
vcTXEDlgNedIdapply 2027	Dialog **Vorgänge bearbeiten**, "Übernehmen"-Schaltfläche
vcTXEDlgNedIdcancel 2016	Text im Dialog **Vorgänge bearbeiten**: **Abbrechen**

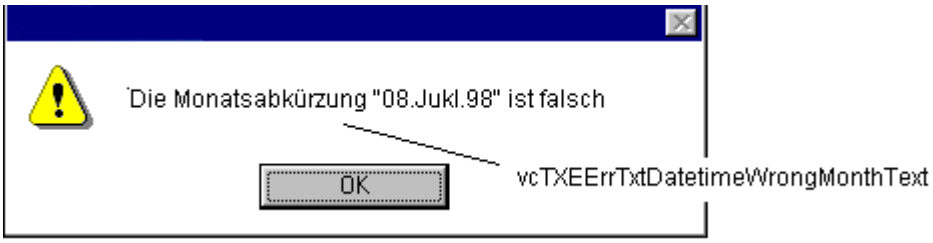
vcTXEDlgNedIdclose 2029	Dialog **Vorgänge bearbeiten**: **Schließen**-Schaltfläche
vcTXEDlgNedIdd 2014	Überschrift des Dialogs **Vorgänge bearbeiten**
vcTXEDlgNedIdhelp 2028	Dialog **Vorgänge bearbeiten**: **Hilfe**-Schaltfläche
vcTXEDlgNedIdok 2015	Text im Dialog **Vorgänge bearbeiten**: **OK**
vcTXEDlgNedNamesColStr 2018	Text im Dialog **Vorgänge bearbeiten**: **Datenfelder**
vcTXEDlgNedTTGotoFirst 2032	Dialog **Vorgänge bearbeiten**: Tooltipptext **Ersten ausgewählten Vorgang anzeigen**
vcTXEDlgNedTTGotoLast 2035	Dialog **Vorgänge bearbeiten**, Tooltipp “Letzten ausgewählten Vorgang anzeigen“
vcTXEDlgNedTTGotoNext 2034	Dialog **Vorgänge bearbeiten**, Tooltipptext **Nächsten ausgewählten Vorgang anzeigen**
vcTXEDlgNedTTGotoPrev 2033	Dialog **Vorgänge bearbeiten**: Tooltipptext **Vorherigen ausgewählten Vorgang anzeigen**
vcTXEDlgNedValuesColStr 2019	Text im Dialog **Vorgänge bearbeiten**: **Werte**
vcTXEErrTxtEntryTooLong 2730	Meldungstext: "Eintrag ist zu lang, %s Zeichen sind möglich."
vcTXEErrTxtWrongLongInteger 2729	Meldungstext: "Eintrag ist kein Integer oder zu lang."
vcTXEPrctBtAll 2306	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Übersicht**
vcTXEPrctBtApply 2318	Schaltflächen-Text im **Seite einrichten**-Dialog: **Anwenden**
vcTXEPrctBtCancel 2302	Schaltflächen-Text im **Druck-Info**-Fenster: **Abbrechen**
vcTXEPrctBtClose 2303	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Schließen**
vcTXEPrctBtFitToPage 2308	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Einpassen**
vcTXEPrctBtNext 2305	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Weiter**
vcTXEPrctBtOk 2301	Schaltflächen-Text des **Seite einrichten**-Dialogs: **OK**
vcTXEPrctBtPageLayout 2311	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Seite einrichten**
vcTXEPrctBtPreviewZoomFactorItems 2321	Einträge für die Kombobox **Zoomfaktor** des **Druckvorschau**-Dialogs: **Auto|75%|100%|150%|200%**
vcTXEPrctBtPrevious 2304	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Vorher**
vcTXEPrctBtPrint 2313	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Drucken**
vcTXEPrctBtPrinterSetup 2312	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Drucker einrichten**
vcTXEPrctBtSingle 2307	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Einzelseite**
vcTXEPrctBtZoomPrint 2319	Schaltflächen-Text des **Druckvorschau**-Dialogs: **Ausschnitt drucken…**
vcTXEPrctDtAddCuttingMarks 2514	Text des **Seite einrichten**-Dialogs: **Zuschnittmarken**
vcTXEPrctDtAlignment 2526	Text des **Seite einrichten**-Dialogs: **Ausrichtung**

vcTXEPrctDtAlignmentItems 2583	Text des **Seite einrichten**-Dialogs: **Oben links|Oben|Oben rechts|Links|Mittig|Rechts|Unten links|Unten|Unten rechts**
vcTXEPrctDtBottom 2521	Text des **Seite einrichten**-Dialogs: **Unten**
vcTXEPrctDtCm 2530	Text des **Seite einrichten**-Dialogs: **cm**
vcTXEPrctDtCurrentValues 2581	Text des **Seite einrichten**-Dialogs: **Aktuell**
vcTXEPrctDtEnableDiagram 2559	Text des **Seite einrichten**-Dialogs: **Diagramm anzeigen**
vcTXEPrctDtExportPage 2568	
vcTXEPrctDtFitToPage 2508	Text des **Seite einrichten**-Dialogs: **Anpassen an Seitenzahl**
vcTXEPrctDtFoldingMarksItems 2577	Text des **Seite einrichten**-Dialogs: **Form A|Form B|Form C**
vcTXEPrctDtFoldingMarksText 2576	Text des **Seite einrichten**-Dialogs:**"&Faltmarkierungen (DIN 824)**
vcTXEPrctDtFooterGroup 2584	Text des **Seite einrichten**-Dialogs: **Fußzeile**
vcTXEPrctDtFrameOutside 2515	Text des **Seite einrichten**-Dialogs: **Rahmen außen**
vcTXEPrctDtInch 2588	Text des **Seite einrichten**-Dialogs: **Zoll**
vcTXEPrctDtLeft 2520	Text des **Seite einrichten**-Dialogs: **Links**
vcTXEPrctDtMargins 2529	Text des **Seite einrichten**-Dialogs: **Mindestgrößen für die Seitenränder**
vcTXEPrctDtMaxPages 2580	Text des **Seite einrichten**-Dialogs: **Seiten**
vcTXEPrctDtOff 2557	Text **Aus** Dialog
vcTXEPrctDtOptions 2528	Text des **Seite einrichten**-Dialogs: **Seitenaufteilung**
vcTXEPrctDtPageDescription 2562	Text des **Seite einrichten**-Dialogs: **Text**
vcTXEPrctDtPageLayout 2532	Fenstertitel des **Seite einrichten**-Dialogs
vcTXEPrctDtPageNumberingItems 2582	Text des **Seite einrichten**-Dialogs: **Zeile.Spalte|Spalte.Zeile|Seite/Anzahl**
vcTXEPrctDtPageNumbers 2518	Text des **Seite einrichten**-Dialogs: **Seitennummerierung**
vcTXEPrctDtPagePadding 2585	Text des **Seite einrichten**-Dialogs: **Seiten mit Leerraum auff&üllen**
vcTXEPrctDtPagePreview 2533	Fenstertitel des Dialogs **Druckvorschau**
vcTXEPrctDtPagesMaxHeight 2511	Text des **Seite einrichten**-Dialogs: **Maximale Höhe**
vcTXEPrctDtPagesMaxWidth 2510	Text des **Seite einrichten**-Dialogs: **Maximale Breite**
vcTXEPrctDtPercent 2509	Text des **Seite einrichten**-Dialogs: **%**
vcTXEPrctDtPrint 2506	Text im Druck-Info-Fenster: **Drucke**
vcTXEPrctDtPrintDate 2564	Text des **Seite einrichten**-Dialogs: **&Druckdatum**
vcTXEPrctDtPrintingPage 2556	Text im Druck-Info-Fenster: **Seite %1 von %2 wird gedruckt auf**
vcTXEPrctDtProjectName 2502	Text im Druck-Info-Fenster: **Projektname**
vcTXEPrctDtReduceExpand 2507	Text des **Seite einrichten**-Dialogs: **Zoomfaktor**
vcTXEPrctDtRight 2522	Text des **Seite einrichten**-Dialogs: **Rechts**
vcTXEPrctDtScaling 2527	Text des **Seite einrichten**-Dialogs: **Skalierung**
vcTXEPrctDtScalingMode 2578	Text des **Seite einrichten**-Dialogs: **&Modus:**

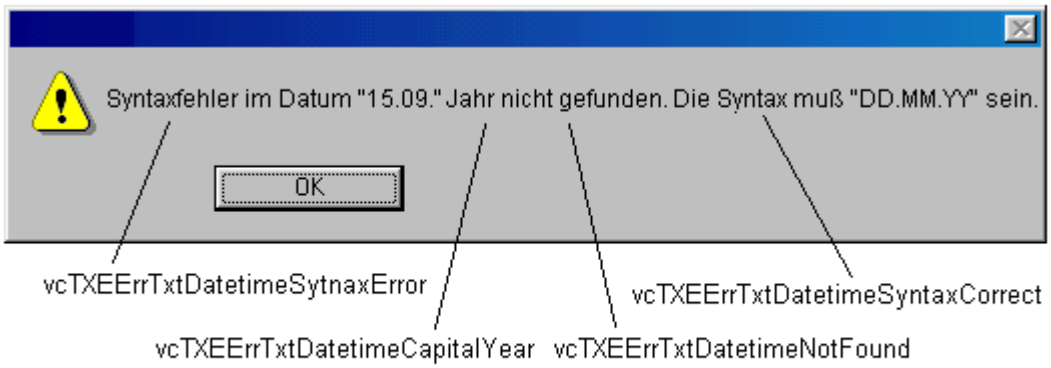
	vcTXEPrctDtStatusBarCurrentValues 2586	**Statuszeilentext** des **Druckvorschau**-Dialogs: **%1 Seiten in %2 Zeilen und %3 Spalten**
	vcTXEPrctDtStatusBarSelectedPage 2587	**Statuszeilentext** des **Druckvorschau**-Dialogs: **Seite %1 selektiert (in Zeile %2, Spalte %3)**
	vcTXEPrctDtTableColumnRange 2575	Text des **Seite einrichten**-Dialogs: **Tabellenspalten (1-5;7)**
	vcTXEPrctDtTop 2519	Text des **Seite einrichten**-Dialogs: **Oben**
	vcTXEPrctDtZoomFactor 2579	Text des **Seite einrichten**-Dialogs:**&Zoomfaktor:**
	vcTXEPrctMtAdjustBottomAndTopMargin 2437	Meldungstext: **Der untere Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert.\r\nAußerdem wird der obere Rand auf %2 cm reduziert.**
	vcTXEPrctMtAdjustLeftAndRightMargin 2434	Meldungstext: **Der linke Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert.\r\nAußerdem wird der rechte Rand auf %2 cm reduziert.**
	vcTXEPrctMtAdjustRightAndLeftMargin 2435	Meldungstext: **Der rechte Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert.\r\nAußerdem wird der linke Rand auf %2 cm reduziert.**
	vcTXEPrctMtAdjustTopAndBottomMargin 2436	Meldungstext: **Der obere Rand ist außerhalb des Wertebereichs und wird deshalb auf %1 cm reduziert.\r\nAußerdem wird der untere Rand auf %2 cm reduziert.**
	vcTXEPrctMtBottomMargin 2409	Meldungstext:**Der untere Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.**
	vcTXEPrctMtIncompatibleVcVersion 2414	Meldungstext: **VcVersion inkompatibel**
	vcTXEPrctMtLeftMargin 2406	Meldungstext: **Der linke Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.**
	vcTXEPrctMtPrinterNotInstalled 2411	Meldungstext: **Kein Drucker installiert**
	vcTXEPrctMtPrintingNotPossible 2402	Meldungstext: **Drucken z. Zt nicht möglich**
	vcTXEPrctMtRightMargin 2408	Meldungstext:**Der rechte Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.**
	vcTXEPrctMtSelectPaperSize 2413	Meldungstext: **Gewählte Blattgröße zu klein**
	vcTXEPrctMtTopMargin 2407	Meldungstext:**Der obere Rand ist außerhalb des Wertebereichs und wird deshalb auf %s cm reduziert.**
	vcTXEPrctMtValueOutOfRange 2404	Meldungstext: **Außerhalb des Wertebereichs %1 bis %2**
	vcTXEPrctMtWillBeAdjustedTo 2410	Meldungstext: **Wird korrigiert auf**
⇨ textEntry	String	Text, der den Standardtext ersetzen soll
⇔ returnStatus	Variant	Rückgabestatus



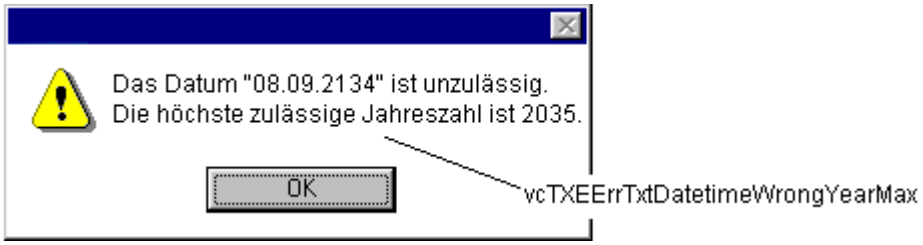
Konstanten des Dialogs ***Seite einrichten***



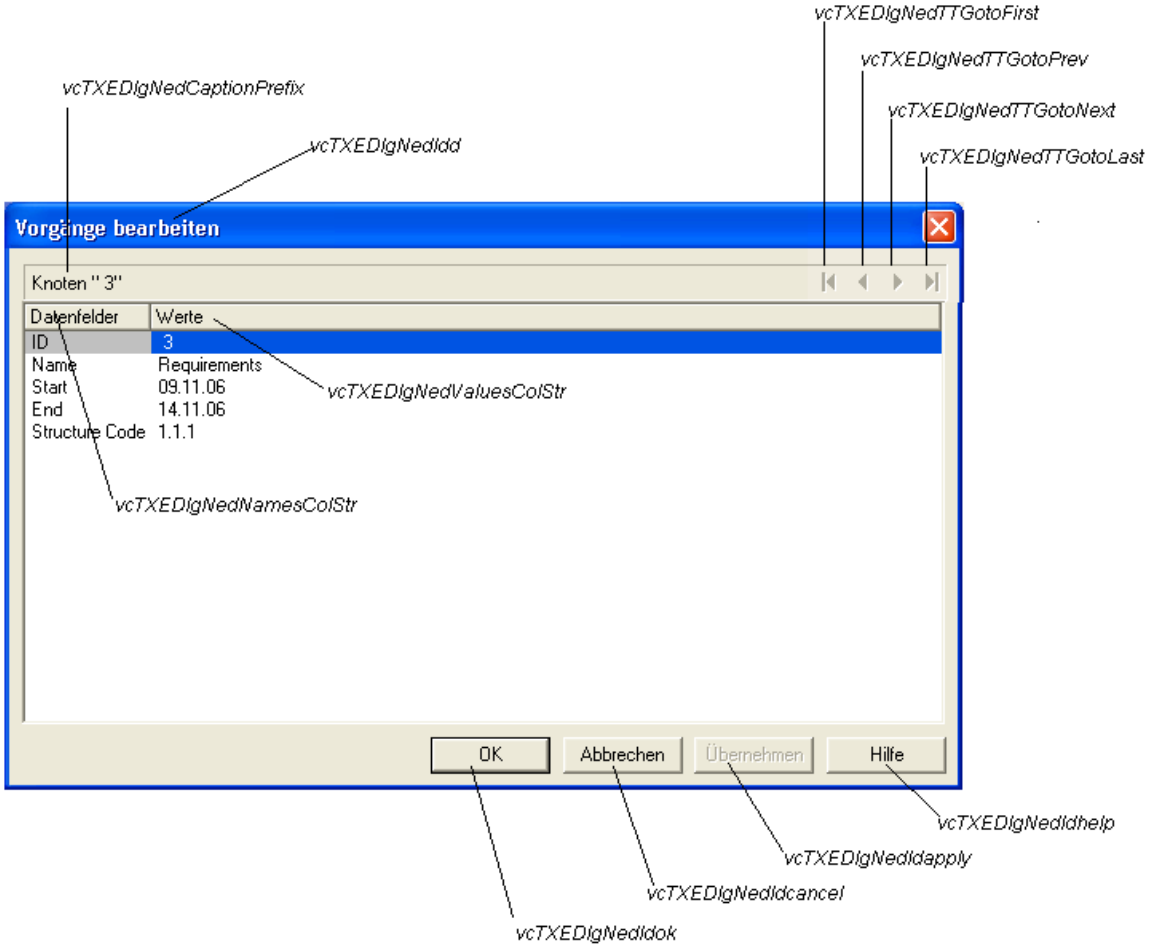
Konstanten der Fehlermeldung ***Fehler im Datum, Monat falsch***



*Konstanten der Fehlermeldung **Syntaxfehler im Datum***

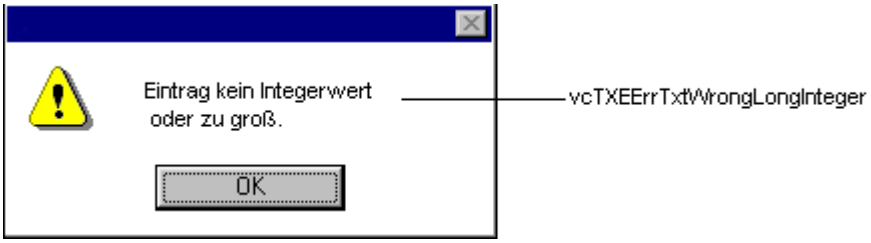


*Konstante der Fehlermeldung **Fehler im Datum, Jahr zu groß***

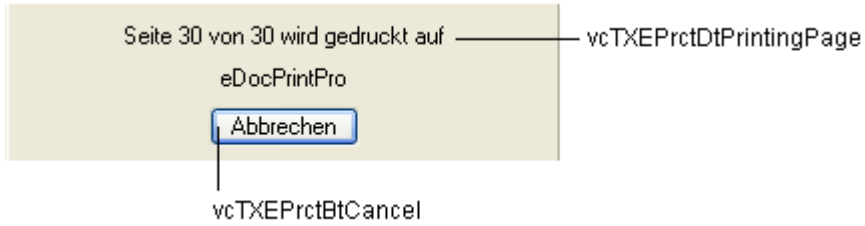


*Konstanten des Dialogs **Vorgänge bearbeiten***

*Konstanten der Fehlermeldung **Wert zu groß***

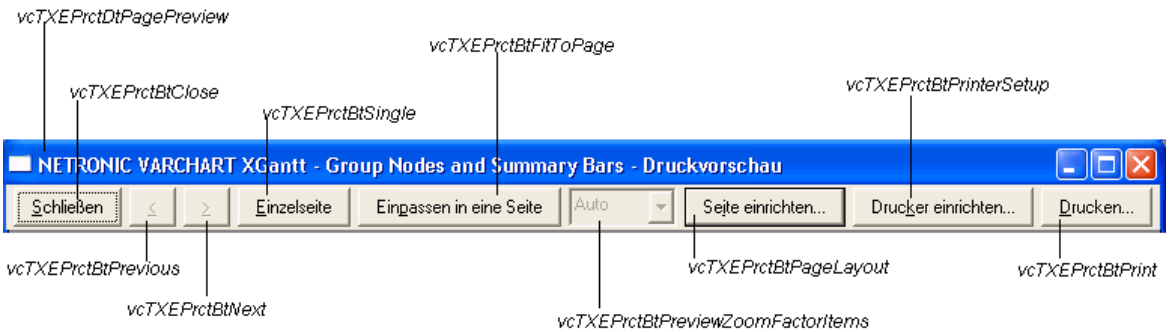


*Konstanten der Fehlermeldung **Wert kein Integer***



*Konstanten der Infobox **Drucken***

*Konstanten der Tastenbeschriftungen in der **Druckvorschau im Einzelansichtsmodus***



*Konstanten der Tastenbeschriftungen in der **Druckvorschau im Übersichtsmodus***

*Konstanten der Statuszeile im Dialog **Druckvorschau***

Code-Beispiel

```
Private Sub VcTree1_OnSupplyTextEntry(ByVal controlIndex As _
                                VcTreeLib.TextEntryIndexEnum, _
                                TextEntry As String, _
                                returnStatus As Variant)
    Select Case controlIndex
        Case vcTXECtxmenCollapse
            TextEntry = "Collapse nodes"
        Case vcTXECtxmenExpand
            TextEntry = "Expand nodes"
        End Select
End Sub
```

OnSupplyTextEntryAsVariant

Ereignis von VcTree

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **OnSupplyTextEntry**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

OnToolTipText

Ereignis von VcTree

Dieses Ereignis tritt nur auf, wenn Sie die VcTree-Eigenschaft **ShowToolTip** auf **True** gesetzt haben. Das Ereignis tritt auf, sobald der Cursor auf ein XTree-Objekt bewegt wird. Es liefert Informationen über das Objekt, den Objekttyp und die Koordinaten des Cursors. Sie können mit Hilfe dieses Ereignisses die vorgegebenen Texte durch eigene Texte ersetzen, z. B. um sie in unterschiedliche Sprachen zu übersetzen. Durch Setzen des Rückgabestatus auf **vcRetStatFalse** oder Leerlassen des Textstrings "" können Sie den Tooltip an dieser Stelle unterdrücken.

	Datentyp	**Beschreibung**
Parameter:		
⇨ hitObject	Object	Objekt
⇨ hitObjectType	VcObjectTypeEnum	Objekttyp
	Mögliche Werte:	
	vcObjTypeBox 15	Objekttyp **Box**
	vcObjTypeNode 2	Objekttyp **Knoten**

	vcObjTypeNodeInLegend 17 vcObjTypeNone 0	Objekttyp **Knoten im Legendenbereich** kein Objekt
⇨ x	Long	X-Koordinate
⇨ y	Long	Y-Koordinate
⇨ ToolTipText	String	Anzuzeigender Text, kann maximal 1024 Zeichen lang sein
⇔ returnStatus	Variant	Rückgabestatus

Code-Beispiel

```
Private Sub VcTree1_OnToolTipText(ByVal hitObject As Object, _
                                  ByVal hitObjectType As _
                                  VcTreeLib.VcObjectTypeEnum, _
                                  ByVal x As Long, ByVal y As Long, _
                                  toolTipText As String, _
                                  returnStatus As Variant)
    If hitObjectType = vcObjTypeNode Then
        toolTipText = "The cursor has been moved over a node!"
    End If
End Sub
```

OnToolTipTextAsVariant

Ereignis von VcTree

Dieses Ereignis ist bis auf die Parameter identisch mit dem Ereignis **OnToolTipText**. Die gesonderte Implementierung wurde notwendig, weil beispielsweise die Sprache VBScript Parameter by-Reference (gekennzeichnet durch ⇦) nur verwenden kann, wenn diese Parameter vom Typ VARIANT sind.

OnWorldViewClosed

Ereignis von VcTree

Dieses Ereignis wird aufgerufen, wenn das Popup-Fenster der Komplettansicht geschlossen wird.

	Datentyp	**Beschreibung**
Parameter:		
⇨ (no parameter)		

Code-Beispiel

```
Private Sub VcTree1_OnWorldViewClosed()
    MsgBox "Do you want to close the worldview window?", vbOKCancel
End Sub
```

OnZoomFactorModifyComplete

Ereignis von VcTree

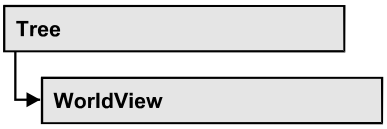
Dieses Ereignis tritt ein, wenn der Anwender in der Komplettansicht (WorldView) die Größe des Rechtecks verändert hat oder markierte Objekte gezoomt hat. Sie können stufenlos zoomen, indem Sie bei gedrückter Strg-Taste das Mausrad drehen. In bestimmten Schritten können Sie zoomen, indem Sie bei gedrückter Strg-Taste die Plus- bzw. Minus-Tasten des Ziffernblocks der Tastatur drücken.

	Datentyp	Beschreibung
Parameter:		
⇨ (no parameter)		

Code-Beispiel

```
Private Sub VcTree1_OnZoomFactorModifyComplete()
     MsgBox "Zoomfactor: " & VcTree1.ZoomFactor
End Sub
```

7.38 VcWorldView



Ein Objekt vom Typ **VcWorldView** bezeichnet das Komplettansicht-Fenster.

Eigenschaften

- Border
- Height
- HeightActualValue
- Left
- LeftActualValue
- MarkingColor
- Mode
- ParentHWnd
- ScrollBarMode
- Top
- TopActualValue
- UpdateBehaviorName
- Visible
- Width
- WidthActualValue

Eigenschaften

Border

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann gesetzt oder erfragt werden, ob die Komplettansicht einen Rahmen besitzt (nicht im Modus **vcPopupWindow**). Die Rahmenfarbe ist **Color.Black**. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Rahmen um die Komplettansicht (True)/kein Rahmen um die Komplettansicht (False) **Standardwert:** True

Code-Beispiel

```
VcTree1.WorldView.Mode = vcNotFixed
VcTree1.WorldView.Border = True
```

Height

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die vertikale Ausdehnung der Komplettansicht erfragt werden. In den Modi **vcFixedAtBottom**, **vcFixedAtTop**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Höhe der Komplettansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.WorldView.Height = 100
```

HeightActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte vertikale Ausdehnung der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche Höhe der Komplettansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.Height = 300
```

Left

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die linke Position der Zusätzliche Ansichten erfragt werden. In den Modi **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Linke Position der Komplettansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.WorldView.Left = 200
```

LeftActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte linke Position der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche linke Position der Komplettansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.LeftActualValue = 150
```

MarkingColor

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die Farbe der Linie des Rechtecks erfragt oder gesetzt werden, das in der Zusätzliche Ansichten den aktuell gewählten Ausschnitt anzeigt. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Color	RGB-Farbwerte **Standardwert:** RGB(0, 0, 255)

Code-Beispiel

```
VcTree1.WorldView.MarkingColor = RGB(255, 0, 0)
```

Mode

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann der Modus der Gesamtansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	WorldViewModeEnum	Modus der Gesamtansicht **Standardwert:** vcPopupWindow
	Mögliche Werte: vcFixedAtBottom 4	Die Komplettansicht wird unten im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Höhe verändert werden, während Position und Breite vorgegeben sind.

vcFixedAtLeft 1	Die Komplettansicht wird links im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
vcFixedAtRight 2	Die Komplettansicht wird rechts im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Breite festgelegt werden, während Position und Höhe vorgegeben sind.
vcFixedAtTop 3	Die Komplettansicht wird oben im Fenster des Steuerelements angezeigt. Das Bezugssystem der Koordinaten ist das Steuerelement. Bei dieser Einstellung kann nur die Höhe festgelegt werden, während Position und Breite vorgegeben sind.
vcNotFixed 5	Die Komplettansicht ist ein untergeordnetes Kindfenster des aktuellen Vaterfensters des Steuerelements und kann an beliebiger Position mit beliebiger Ausdehnung angeordnet werden. Das Bezugssystem der Koordinaten ist das Vaterfenster. Das Kindfenster ist ohne eigenen Fensterrahmen und kann vom Benutzer nicht interaktiv verschoben werden. Das Vaterfenster kann bei Bedarf über die Eigenschaft **VcWorldView.ParentHWnd** geändert werden.
vcPopupWindow 6	Die Komplettansicht ist ein Popup-Fenster, das einen eigenen Rahmen besitzt und vom Benutzer in Position und Größe verändert werden kann. Das Bezugssystem der Koordinaten ist der Bildschirm. Das Fenster kann über das Standard-Kontextmenü ein- bzw. ausgeschaltet oder über die **Schließen**-Schaltfläche in der Titelleiste ausgeschaltet werden.

Code-Beispiel

```
VcTree1.WorldView.Mode = vcFixedAtBottom
```

ParentHWnd

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann im Modus **vcNotFixed** das HWnd-Handle des Vaterfensters festgelegt werden, wenn die Komplettansicht beispielsweise in einem selbst implementierten Rahmenfenster erscheinen soll. Standardmäßig steht dies auf dem HWnd-Handle des Vaterfensters des VARCHART-ActiveX-Hauptfensters. Diese Eigenschaft kann nur zur Laufzeit verwendet werden.

	Datentyp	Beschreibung
Eigenschaftswert	OLE_HANDLE	Zugriffsnummer

Code-Beispiel

```
MsgBox (VcTree1.worldview.ParentHWnd)
```

ScrollBarMode

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann der Scrollbarmodus der Komplettansicht erfragt oder gesetzt werden. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	WorldViewScrollBarModeEnum	Scrollbarmodus **Standardwert:** NoScrollBar
	Mögliche Werte:	
	vcAutomaticScrollBar 3	Anzeige einer horizontalen oder vertikalen Bildlaufleiste, wenn nötig.
	vcHorizontalScrollBar 1	Anzeige einer horizontalen Bildlaufleiste, wenn nötig.
	vcNoScrollBar 0	Es wird immer das vollständige Diagramm ohne Bildlaufleisten angezeigt.
	vcVerticalScrollBar 2	Anzeige einer vertikalen Bildlaufleiste, wenn nötig.

Code-Beispiel

```
VcTree1.WorldView.ScrollBarMode = vcAutomaticScrollBar
```

Top

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die obere Position der Komplettansicht erfragt werden. In den Postionen **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann sie außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Obere Position der Komplettansicht

Code-Beispiel

```
VcTree1.WorldView.Top = 20
```

TopActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte obere Position der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche obere Position der Komplettansicht **Standardwert:** 0

Code-Beispiel

```
VcTree1.LegendView.TopActualValue = 40
```

UpdateBehaviorName

Eigenschaft von VcWorldView

Mit dieser Eigenschaft können Sie den Namen des Aktualisierungsverhaltens erfragen oder festlegen.

	Datentyp	Beschreibung
Eigenschaftswert	String	Name des Aktualisierungsverhaltens

Visible

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann festgelegt oder erfragt werden, ob die Komplettansicht sichtbar ist. Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Boolean	Komplettansicht sichtbar (True)/unsichtbar (False) **Standardwert:** False

Code-Beispiel

```
VcTree1.WorldView.Visible = True
```

Width

Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die horizontale Ausdehnung der Komplettansicht erfragt werden. In den Positionen **vcFixedAtLeft**, **vcFixedAtRight**, **vcNotFixed** und **vcPopupWindow** der Eigenschaft **Mode** kann diese Eigenschaft außerdem gesetzt werden.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

Diese Eigenschaft kann auch auf der Eigenschaftenseite **Zusätzliche Ansichten** festgelegt werden.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Horizontale Ausdehnung der Komplettansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.WorldView.Width = 200
```

WidthActualValue

Nur-Lese-Eigenschaft von VcWorldView

Mit dieser Eigenschaft kann die tatsächlich dargestellte horizontale Ausdehnung der Komplettansicht erfragt werden. Dieser tatsächliche Wert kann in den Modi **vcLVFixedAtBottom**, **vcLVFixedAtLeft**, **vcLVFixedAtRight**, **vcLVFixedAtTop** von dem eingestellten Wert abweichen, da in diesen Modi je nach Einstellung die Höhe oder Breite vorgegeben ist.

Bei den Koordinaten handelt es sich um Gerätekoordinaten. In Visual Basic ist also ggf. eine Umrechnung von/in Twips über die Benutzung der Eigenschaften **App.TwipsPerPixelX** und **App.TwipsPerPixelY** notwendig.

	Datentyp	Beschreibung
Eigenschaftswert	Long	Tatsächliche horizontale Ausdehnung der Komplettansicht {0, ...} **Standardwert:** 100

Code-Beispiel

```
VcTree1.LegendView.WidthActualValue = 600
```

8 Index

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

W

Z