



Version 5.9.01

www.vectorsoft.de

support@vectorsoft.de

orga@vectorsoft.de

© Copyright 1987-2021 by **vectorsoft AG**

Update-Beschreibung CONZEPT 16-Version 5.9

Die Update-Beschreibung informiert über Änderungen in den jeweiligen Release-Ständen

Versionen

- Release 5.9.01

CONZEPT 16-Release 5.9.01 - 11. Mai 2021

 Die Clients der Version 5.9.01 benötigen den Server der Version 5.9.01

 Beim ersten Öffnen einer Datenbank mit dem Standard-Client 5.9 wird die Datenbank konvertiert. Eine Rückkonvertierung auf die vorhergehende Version ist nicht möglich.

An folgenden Komponenten Änderungen durchgeführt:

- Alle Komponenten
- Installationsroutine
- Server
- Client
- Programmierschnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.9.01

• Neue Systemvoraussetzungen

 Für den Betrieb von CONZEPT 16 wird mindestens Windows 7 / Windows Server 2008 R2 oder Linux mit Kernel 3.x vorausgesetzt. Auf den folgenden Betriebssystemen ist CONZEPT 16 in der Version 5.9 nicht mehr lauffähig:

- ◆ Windows XP
- ◆ Windows Server 2003
- ◆ Windows Vista
- ◆ Windows Server 2008

 Für den Betrieb des ODBC-Treibers wird bei Windows 7 und Windows Server 2008 R2 zusätzlich das Service Pack 1 vorausgesetzt. Nach Installation des bis zur Version 5.8.12 verfügbaren ODBC-Legacy-Treibers führt eine Installation des Service Pack 1 **nicht** dazu, dass die erneute Durchführung der Installationsroutine (Aktion Aktualisieren) den neuen ODBC-Treiber installiert. Um den aktualisierten ODBC-Treiber zu installieren, muss dieser zunächst deinstalliert und anschliessend wieder installiert werden (Aktion Ändern).

• Bibliotheken

◆ Neue Version der PDF-Bibliothek

Die PDF-Bibliothek wurde auf Version 4.0.36.99 aktualisiert.

Diese Version enthält wichtige Fehlerbehebungen.

◆ Aktualisierung der XML-Bibliothek

Kontakt

Die verwendete libxml2-Bibliothek wurde auf die Version 2.9.10 aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

♦ Aktualisierung der OpenSSL-Bibliothek

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.1.1g aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

CONZEPT 16-Installationsroutine - Version 5.9.01

• Chromium-Package

Das in dieser Version neu hinzugefügte Chromium-Objekt benötigt zur Ausführung weitere Komponenten. Diese werden in Form einer Datei (Chromium-Package) zur Verfügung gestellt. Das Chromium-Package ist separat über die Installationsroutine installierbar.

CONZEPT 16-Server - Version 5.9.01

• Lizenz

 Unabhängig vom eingesetzten Betriebssystem wird zum Betrieb der 5.9.01 eine Lizenz vom Stand 5.9.01 (siehe Control-Center Eintrag Lizenz -> Stand) benötigt. Dies erfordert einen aktiven Softwarepflegevertrag. Ist die automatische Aktualisierung der Lizenzdatei (siehe Konfiguration) aktiviert, ist kein manueller Eingriff erforderlich.

• Erweiterung des Datenbank-Cache Limit

Die Größe des Datenbank-Cache kann nun bis zu 500 GiB (512GB) betragen.

• Lange Schlüsselwerte

Schlüssel können nun bis zu 950 (zuvor 240) Zeichen umfassen. Die neue Option muss im Datenstruktur-Editor bzw. in der Konfiguration des Standard-Client über die Option "Lange Schlüsselwerte" beim jeweiligen Schlüssel gesetzt werden, damit diese aktiv ist.

Einzelne Schlüsselfelder haben nach wie vor eine Begrenzung von maximal 255 Zeichen.

Nach der Änderung der Option ist eine Schlüsselreorganisation notwendig. Diese wird im Falle des Datenstruktur-Editors automatisch durchgeführt.

Über die Option KeyInfo() mit der Option KeyMaxLen kann abgefragt werden, wie groß die maximale Schlüssellänge ist. Die Funktion liefert 240, wenn die Option nicht gesetzt ist und bei gesetzter Option 950.

• Datensätze mit Änderungszeitpunkt

Im Datenstruktur-Editor wie auch in der Konfiguration des Standard-Client gibt es nun die Möglichkeit Datensätze mit Änderungszeitpunkt zu speichern. Wird die Option aktiviert, dann wird der Änderungszeitpunkt beim hinzufügen und ändern von Datensätzen auf die aktuelle Zeit des Datenbank-Servers gesetzt. Es wird die UTC-Zeit gespeichert.

Kontakt

Der Zeitstempel wird im Falle eines RecReplace() auch dann gesetzt, wenn keine Änderung der Feldinhalte des Datensatzes vorliegt.

Nach dem Erneuten Lesen des Datensatzes mit RecRead() kann der Zeitstempel mit der Option RecModified beim Befehl RecInfo() als 64-Bit-Wert ausgelesen und bei Bedarf über CnvAB() in eine Zeichenkette konvertiert werden.

Über die Option FileRecModified beim Befehl FileInfo() kann ermittelt werden, ob die Option bei einer Tabelle gesetzt ist (Resultat = 1) oder nicht (Resultat = 0).

Die Datensatzverwaltung zeigt im Fenster "Datensatz bearbeiten" den Zeitstempel des Datensatzes an, sofern gesetzt.

CONZEPT 16-Client - Version 5.9.01

• **64-Bit Advanced-Client und Debugger**

Der Advanced-Client und der Debugger stehen nun in einer 64-Bit Version bereit. Dies erlaubt die Allokation von mehr Arbeitsspeicher während des Betriebs von CONZEPT 16. Zur Umsetzung wurden die bestehenden DLLs umbenannt um die Zugehörigkeit zum 32- bzw. 64-Bit-Client wiederzuspiegeln. Weiterhin wurden die Komponenten vom Watcom-Compiler auf den Visual Studio 2015-Compiler umgestellt, was die Fehleranalyse im Absturzfall in den meisten Fällen vereinfacht.

• **Log-Viewer als 64-Bit-Variante**

Der Log-Viewer steht nun als 64-Bit-Variante zur Verfügung. Während die 32-Bit-Version maximal 500.000 Einträge auf einmal darstellt ist die 64-Bit-Variante in der Lage bis zu 2.000.000 Einträge darzustellen. Die Installationsroutine installiert je nach verwendeter Betriebssystem-Plattform die entsprechende Variante. Welche Variante gerade ausgeführt wird, ist im Titel des Log-Viewers sichtbar.



Das Extrahieren der 64-Bit-Variante aus der Installationsroutine (Option cpn_action=extract) ist gegenwärtig nicht möglich. Die Option extrahiert immer die 32-Bit-Variante.

• **alpha-Felder und alpha-Variablen mit 8192 Zeichen**

Alpha-Felder und Alpha-Variablen können eine Maximallänge von bis zu 8192 Zeichen besitzen. Datenbanken, die Memo-Felder beinhalten werden beim Öffnen durch den Client weiterhin in Alpha-Felder mit einer Länge von 4096 umgewandelt.

• **Chromium-Objekt**

Das Chromium-Objekt ermöglicht die Anzeige von Web-basierten Inhalten. Basierend auf der Chrome-Engine stellt es eine moderne Alternative zum WebNavigator-Objekt dar. Über die Eigenschaft InstallPathCommon des App-Objektes können die Pfade definiert werden, die für die Installation des Chromium-Objektes berücksichtigt werden sollen.

- **Canvas-Objekt**

Das Canvas-Objekt ist ein neues Oberflächen-Objekt und dient zur Anzeige und Interaktion vektorbasierter Grafikobjekte. Das Canvas-Objekt ist hierbei ein Container, der die Ausgabefläche für hinzugefügte CanvasGraphic-Objekte unterschiedlicher Formen bereit stellt.

- **CodeEdit**

- ♦ **Neue Eigenschaften des CodeEdit-Objektes**

Das CodeEdit-Objekt wurde um die Eigenschaften SBarStyle, sowie die neuen Eigenschaften EditorAllowControlChars und EditorShowFlags erweitert.

Die Eigenschaft EditorAllowControlChars definiert, ob Steuerzeichen (ASCII-Code < 32) eingegeben werden können. Es ist zu beachten, dass nicht alle Steuerzeichen eingegeben und angezeigt werden können.

Mit der Eigenschaft EditorShowFlags können anzuzeigende Elemente des CodeEdit-Objektes definiert werden.

- ♦ **Behandlung von EmulateKeys**

Ist bei einem übergeordneten Objekt des CodeEdit-Objektes die Eigenschaft EmulateKeys auf true gesetzt, führen die Tasten(-kombinationen)     +  und  zum Verlassen des Objektes.

- ♦ **Neues Ereignis EvtEditorHelp**

Mit der Tastenkombination Strg+F1 wird beim CodeEdit die kontextbezogene Hilfe ausgelöst. Mit dem neuen Ereignis EvtEditorHelp kann eine eigene Behandlung implementiert werden.

- **Befehle und Eigenschaften**

- ♦ **RmtData / Erweiterung der Schlüssellänge**

Die Länge des Schlüssels für RmtData-Befehle wurde von 250 auf 2000 erweitert. Schlüssel und Daten bei RmtDataWrite() dürfen maximal 4601 Byte lang werden, ansonsten wird der Laufzeitfehler ErrStringOverflow ausgelöst.

Die reine Datenlänge (ohne Schlüssel) darf 4096 Byte nicht übersteigen, sonst wird ebenfalls der Laufzeitfehler ErrStringOverflow ausgelöst.

- ♦ **Automatische Größenanpassung des Scrollbereiches einer Scrollbox**

Die Eigenschaften ScrollWidth und ScrollHeight der Scrollbox können nun den Wert -1 annehmen. In diesem Fall wird die Breite bzw. Höhe des Scrollbereiches auf die Breite bzw. Höhe der Scrollbox angepasst, wenn sich die Größe der Scrollbox ändert. In der entsprechenden Richtung kann dann nicht mehr gescrollt werden.

- ♦ **Dialogpositionierung mit WinDialogCenter**

Kontakt

Die Option WinDialogCenter zentrierte den Dialog auf dem Primärbildschirm, wenn dieser nicht zur Gänze zum Eltern-Dialog zentriert dargestellt werden konnte. Nun erfolgt die Positionierung zum nächsten Monitor, so dass ein besserer Bezug zum Eltern-Dialog hergestellt werden kann. Die Optionen WinDialogCenterX und WinDialogCenterY wurden nicht verändert.

♦ Nachselektion auf dem Server ausführen

Bei dem Befehl SelRun() können nun die Argumente SelBase und SelServer kombiniert werden, um eine Nachselektion auf dem Server auszuführen. Dies ist zum Beispiel beim Sortieren einer größeren Selektionsmenge sinnvoll.

♦ Prozess-Architektur ermitteln

Mit der neuen Eigenschaft ProcessArchitecture des Sys-Objektes kann ermittelt werden, ob der Prozess in der 32- oder der 64-Bit-Variante läuft.

♦ Erweitern des Barcode-Objekts um die Eigenschaft Design

Das Barcode-Objekt wurde um die Eigenschaften Design und DesignFlags erweitert.

• ZUGFeRD

♦ Neue Optionen bei PdfClose() für ZUGFeRD 2.x-konforme PDF-Dateien

Bei PdfClose() kann jetzt neben dem Profil (PdfModePdfZUGFeRDBasic, PdfModePdfZUGFeRDComfort, PdfModePdfZUGFeRDExtended) auch die Version der ZUGFeRD-Spezifikation angeben werden:

PdfModePdfZUGFeRD10 ZUGFeRD Version 1.0
PdfModePdfZUGFeRD20 ZUGFeRD Version 2.0
PdfModePdfZUGFeRD21 ZUGFeRD Version 2.1

Wird ein ZUGFeRD-Profil aber keine Version angegeben, wird Version 1.0 verwendet. Das Beispiel ZUGFeRD_Basic in der Code-Library wurde entsprechend angepasst, um ein ZÜGFeRD 2.1-konformes PDF zu generieren.

♦ PDF mit XRechnung-Profil

Die neue Konstante PdfModePdfZUGFeRDXRechnung kann in Kombination mit der Konstante PdfModePdfZUGFeRD21 bei der Funktion PdfClose() angegeben werden, um eine elektronische Rechnung gemäß dem Profil "XRechnung" von ZUGFeRD zu speichern.

Dieses Profil verlangt, dass die XML-Datei der Rechnung mit der Beziehung "Source" und dem Namen "xrechnung.xml" mit der PDF-Datei assoziiert wird. Mit der neuen Konstante PdfAttachAssociateSource bei den Funktionen PdfAttachFile und PdfAttachMem kann die XML-Datei an die PDF-Datei entsprechend angehängt werden. Zusätzlich gibt es die neue Konstante PdfAttachAssociateAlternative, die die XML-Datei als Alternative angehängt werden. Die bisherige Konstante

Kontakt

PdfAttachAssociateZUGFeRD verhält sich wie die Konstante PdfAttachAssociateAlternative und wird zur Abwärtskompatibilität vorerst bestehen bleiben.

• **Unicode-Unterstützung**

◆ **Unicode-Unterstützung für dynamische Objekte**

Objekte, die mit den Befehlen WinCreate() (mit Angabe des Elternobjektes) oder WinAdd() zu einem Frame mit gesetzter Unicode-Eigenschaft hinzugefügt werden, werden auch mit Unicode-Unterstützung erstellt.

◆ **Fsi- und Bin-Befehle mit UTF-8-Dateinamen**

Die Befehle FsiDelete(), FsiPath(), FsiPathChange(), FsiPathCreate(), FsiPathDelete(), BinImport() und BinExport() unterstützen die Angabe der Option FsiNameUtf8, um angegebene Dateinamen im UTF8-Zeichensatz zu verwenden.

• **Designer**

◆ **Datensatzverwaltung Import mit veränderten Feldtypen**

Beim Import von Datensätzen schlägt die Datensatzverwaltung eine automatische Feldzuweisung vor. Diese wurde dahingehend erweitert, dass auch Datentyperweiterungen berücksichtigt werden. Bisher musste Name und Datentyp übereinstimmen. Jetzt findet auch eine automatische Zuordnung zu einem Feld mit einem erweiterten Datentyp statt. Eine Übersicht der Konvertierungsmöglichkeiten befindet sich unter Datensatzverwaltung - Typkonvertierung.

• **Debugger**

◆ **Ausgewählte Zeilen im Protokoll des Debuggers kopieren bzw. löschen**

In der Protokoll-Ansicht des Debuggers besteht nun die Möglichkeit, mehrere Zeilen per Multiselect auszuwählen. Die selektierten Zeilen können dann über das Kontextmenü bzw. über die

Tasten(-kombinationen) + in die Zwischenablage kopiert bzw. über gelöscht werden.

◆ **Detailansicht der Variablen ausblendbar**

Die Anzeige der Variabldetails in der Variablenliste kann in den Einstellungen unter Anzeige ein- bzw. ausgeblendet werden.

◆ **Zeilennummern in Prozeduranzeige aus-/einblenden**

Die Zeilennummernanzeige in der Prozeduranzeige des Debuggers kann über das Kontextmenü der Prozeduranzeige oder über die

Tastenkombination + aus- bzw. eingeblendet werden.

• **Debug-Optionen**

◆ **Protokollierung für Speichermanagement-Defekte**

Kontakt

Wird beim Starten des Clients die neue Debug-Option 0x8000 angegeben, werden beim Auftreten von irregulären Zuständen der CONZEPT 16-Speicherverwaltung weitere Informationen und protokolliert.

• Sonstige Erweiterungen

◆ Ermittlung, ob der Client unter Wine läuft

Wird der Client unter Wine betrieben, wird dies in den Systeminformationen (Menü ? -> Info) des Designers und des Debuggers angezeigt.



Bei der Verwendung von Wine treten folgende bekannte Einschränkungen auf:

- ◊ Es können keine Ressourcen-Prüfungen beim Öffnen von Dialogen durchgeführt werden.
- ◊ Es können keine Mengen-Begrenzung des Cache für das DocView ermittelt werden.



Jeglicher Support für Wine ist ausgeschlossen.

CONZEPT 16-Programmierschnittstelle - Version 5.9.01

• Erweiterung um Datentypen

C16_ProcArgument und C16_ProcResult funktioniert jetzt mit allen in CONZEPT 16 vorhandenen Datentypen. Der Befehl C16_ArgInfo() der DLL-Schnittstelle liefert nun ebenfalls alle verfügbaren Datentypen zurück. Die Strukturen für die Datentypen decimal, caltime, point und rect sind in der Header-Datei c16.h hinzugekommen.

• Entfernung der Bibliotheken für Borland C++ / Watcom C++ / Visual Studio 6



Die statischen Link-Bibliotheken (c16_pgxw.lib / c16_pgxe.lib / c16_pgxw_vc6.mvc) für Borland C++, Watcom C++ und Visual Studio 6 werden nicht mehr mitgeliefert.

Diese werden ersetzt durch die Link-Bibliotheken für Visual Studio (verwendbar ab Version 2005) unter demselben Namen (c16_pgxw.lib / c16_pgxe.lib). Bisher wurden die Varianten für Visual Studio mit der Dateiendung .mvc mitgeliefert.

• Anpassung einiger bestehender Funktionen an 64-Bit

Folgende Funktionen wurden verändert:

- ◆ C16_KeyReorgWatch - Die Struktur vC16_KeyReorgInfo liefert die Datensatz-ID als vXLONG (64-Bit-Wert).
- ◆ C16_RecLink - Das Argument aLinkNo wurde auf vXLONG (64-Bit) erweitert.
- ◆ C16_RecRead64 - Das Argument aKeyNo wurde auf vXLONG (64-Bit) erweitert.



Durch die Anpassungen ist existierender Programmcode in der Programmierschnittstelle oder DLL-Schnittstelle (für die geänderten Funktionen) u. U. nicht mehr kompatibel und muss geändert und neu

übersetzt werden!

- **Suche nach regulären Ausdrücken auch ohne graphische Erweiterung**

Die Befehle StrFindRegEx() und TextSearchRegEx() stehen nun auch in Prozeduren der Programmierschnittstelle (ohne graphische Erweiterung) zur Verfügung.

 Hierfür werden die Dateien c16_icu_w32.dll bzw. c16_icu_w64.dll für die 32- bzw. 64-Bit-Variante der Programmierschnittstelle benötigt.

- **C16_SelInfo**

Die Programmierschnittstelle hat nun die neue Funktion C16_SelInfo in Analogie zum SelInfo()-Befehl. Hierdurch kann u.a. die Anzahl der Datensätze der Selektion ermittelt werden.

Update-Beschreibung CONZEPT 16-Version 5.8

Die Update-Beschreibung informiert über Änderungen in den jeweiligen Release-Ständen

CONZEPT 16-Release 5.8.12 - 29. November 2019



Die Clients der Version 5.8.12 benötigen den Server der Version 5.8.12

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Server
- Client
- Programmierschnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.8.12

• Bibliotheken

◆ Aktualisierung der XML-Bibliothek

Die verwendete libxml2-Bibliothek wurde auf die Version 2.9.7 aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

◆ Aktualisierung der Zip-Bibliothek

Die verwendete zlib-Bibliothek wurde auf die Version 1.2.11 aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

◆ Aktualisierung der OpenSSL-Bibliothek

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.1.1d aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

• Einschränkung der erlaubten Komponenten für Benutzerrecht Externer Zugriff (UrmOldPropExtAccess)

Besitzt die Hauptbenutzergruppe eines Benutzer das Recht Externer Zugriff (UrmOldPropExtAccess), kann sich der Benutzer mit den Komponenten Programmierschnittstelle, ODBC-Treiber, PHP-Schnittstelle, Druckprozessor, SOA-Service, Web-Schnittstelle, sowie DbaConnect() aus Standard- und Advanced-Client zugegriffen werden. Über fest definierte Eigenschaften bei den Benutzergruppen können die Zugriffsrechte pro Komponente verboten werden.

• Sammlung von Stammzertifikaten aktualisiert

Die Sammlung von Stammzertifikaten (siehe Speicherorte von Konfigurationsdateien) wurde auf die Version 70 von Mozilla Firefox aktualisiert. Diese Datei wird jedoch nur aktualisiert, wenn sie seit der Installation nicht um eigene Stammzertifikate erweitert wurde.

CONZEPT 16-Server - Version 5.8.12

• Lizenz



Unabhängig vom eingesetzten Betriebssystem wird zum Betrieb der 5.8.12 eine Lizenz vom Stand 5.8.12 (siehe Control-Center Eintrag Lizenz ->

Stand) benötigt. Dies erfordert einen aktiven Softwarepflegevertrag. Ist die automatische Aktualisierung der Lizenzdatei (siehe [Konfiguration](#)) aktiviert, ist kein manueller Eingriff erforderlich.

- **Linux Kernel 5.x**

Mit dieser Version wird der Linux Kernel 5.x unterstützt.

CONZEPT 16-Client - Version 5.8.12

- **CodeEdit**

- ◆ **Markierung für maximale Zeilenlänge im CodeEdit**

Im CodeEdit-Objekt kann mit der Eigenschaft EditorShowLineLimit eine Markierung für die maximale Zeilenlänge aktiviert werden.

- ◆ **CodeEdit unterstützt MenuAction**

Wird die MenuAction eines MenuItem bzw. eines Toolbar-Button auf eine der _WinMnuActEdit...-Konstanten gesetzt, reagiert nun auch ein CodeEdit-Objekt darauf, wenn es den Eingabefokus besitzt.

- ◆ **Hervorhebung eigener Schlüsselwörter im CodeEdit**

Mit den neuen Befehlen WinEditorKeywordsAdd() und WinEditorKeywordsRemove() können eigene Schlüsselwörter zu den CodeEdit-Objekten hinzugefügt bzw. wieder entfernt werden. Die Hervorhebung von CONZEPT 16-eigenen Schlüsselwörtern kann mit WinEditorKeywordsRemove() ebenfalls entfernt werden. Diese werden dann abhängig vom Typ hervorgehoben. Da die Liste der Schlüsselwörter global für alle CodeEdit-Objekte gilt, muss die Darstellung eines einzelnen Objektes mit der Funktion WinEditorKeywordsUpdate() aktualisiert werden, sofern nicht nach dem Hinzufügen der Schlüsselwörter ein neuer Text geladen wird.

- ◆ **Auto vervollständigungsliste im CodeEdit-Objekt unterstützt Modern Theme Style**

Die Darstellung der Auto vervollständigungsliste im CodeEdit-Objekt kann mittels der neuen Theme-Elemente CodeEditList (_WinTypeCodeEditList) und CodeEditListColumn (_WinTypeCodeEditListColumn) im Modern Theme Style angepasst werden.

- ◆ **Neue Eigenschaft EditorAclFlags zum Ausblenden von Elementen in der Auto vervollständigungsliste**

Mit der neuen Eigenschaft EditorAclFlags des CodeEdit-Objektes kann definiert werden, welche Elemente in der Auto vervollständigungsliste angezeigt werden.

- ◆ **Neuer Befehl WinEditorGoTo()**

Mit dem neuen Befehl WinEditorGoTo() kann zu einer Position gescrollt und der Cursor auf die angegebene Position gesetzt werden. Mit dem bisherigen Befehl WinEditorSetSelection() wurde nur die Markierung

gesetzt, ein Scrollen zu der Position war nicht möglich.

- **GanttGraph**

- ◆ **Größenänderung von Box-Objekten im GanttGraph per Drag & Drop**

Wird in der Eigenschaft GanttFlags das Flag WinGanttIvlBoxSizingAllowed aktiviert, kann die Größe von Box-Objekten per Drag & Drop geändert werden. Hierfür wird das Ereignis EvtIvlDropItem ausgelöst. Durch den Rückgabewert false kann die Größenänderung unterbunden werden.

- ◆ **Schriftart und -größe für Interval- und Box-Objekte definierbar**

Die Interval- und Box-Objekte verfügen nun über die Eigenschaften Font und FontParent. Über diese beiden Eigenschaften kann die Schriftart für jedes Objekt individuell definiert werden. Ist FontParent auf true gesetzt, werden die Schriftart und -größe vom GanttGraph-Objekt verwendet.

- ◆ **Transparente Hintergrundfarbe für Interval- und Box-Objekte**

Die Farbe ColBkg von Interval- und Box-Objekten kann auf teiltransparente Farben gesetzt werden. Transparente Farben können mit dem neuen Befehl WinColorOpacitySet() erzeugt werden.

- **Modern Theme Style**

- ◆ **Optimierung von ThemeSets**

Die Verwendung von Theme-Objekten mit vielen ThemeSets wurde optimiert, so dass diese bei wiederholter Verwendung desselben Theme und unterschiedlicher ThemeSets deutlich schneller geladen werden.

- ◆ **Beschreibung für ThemeSets**

Im Theme-Editor ist es nun mögliche eine Kurzbeschreibung für ThemeSets zu setzen. Diese wird im Spalten-Header angezeigt. Die Kurzbeschreibung kann beim Erstellen eines neuen ThemeSets im Theme-Editor oder per Doppelklick auf den Spaltenkopf editiert werden. Prozedural kann die Beschreibung über die Eigenschaft ThemeDescription ermittelt oder gesetzt werden.

- **Sonstige Befehle und Eigenschaften**

- ◆ **SckListen() validiert angegebene Optionen**

Der Befehl SckListen() hat die Optionen nicht validiert. Wurde der Deskriptor eines Frames in diesem statt im nächsten Argument angegeben, wurde dieser als Optionen interpretiert. Die angegebenen Optionen werden nun validiert und führen bei ungültigen Kombinationen zu dem Laufzeitfehler ErrValueInvalid.



Diese Änderung kann zu Inkompatibilitäten führen.

- ◆ **Neuer Befehl WinCopy()**

Kontakt

Mit dem neuen Befehl WinCopy() können Oberflächenobjekte kopiert werden, sofern es sich bei dem Objekt selbst und allen untergeordneten Objekten um mit WinCreate() erstellbare Objekte handelt.

♦ Neuer Befehl WinShutdownBlock()

Das Ereignis EvtEndSession wird aufgerufen, wenn die Windows-Sitzung beendet wird. In dem Ereignis können offene Vorgänge in der Anwendung abgeschlossen werden. Mit dem neuen Befehl WinShutdownBlock() kann zusätzlich verhindert werden, dass Windows die Sitzung beendet. Er sollte nur dann verwendet werden, wenn eine Operation in Verarbeitung ist, die abgeschlossen werden muss.

♦ Neue Befehle für (teil-)transparente Farben

Mit den neuen Befehlen WinColorOpacitySet() und WinColorOpacityGet() kann die Deckkraft einer Farbe gesetzt bzw. ermittelt werden. Teiltransparente Farben können derzeit für Box- und Interval-Objekte im GanttGraph verwendet werden.

♦ XmlLoad() unterstützt Zeichensatz ISO-8859-15

Bisher wurde von XmlLoad() beim Laden von XML-Dateien mit Encoding ISO-8859-15 der Fehlerwert ErrXmlFatal zuückgegeben. Als Fehlermeldung wurde Unsupported encoding ISO-8859-15 ermittelt. Dieser Zeichensatz wird nun unterstützt.

♦ Neue Konstante FsiFileCRC32 für die Befehle FsiFileInfo() und FsiFileProcess()

Mit der Konstante FsiFileCRC32 kann bei FsiFileInfo() die CRC32-Prüfsumme einer Datei im Hexadezimalen Format ermittelt werden. Bei FsiFileProcess() wird bei Angabe der Konstanten die CRC32-Prüfsumme über die Datei, sowie über den internen Dateiheader gebildet, oder verifiziert.

♦ Erhöhung der maximalen Anzeigedauer für WinLayer()

Die Zeitspanne für WinLayer() wurde vergrößert. Die maximal mögliche Zeitspanne wurde von 30 Sekunden auf 2.147.483.647 (MaxInt) Millisekunden erhöht, was etwas über 596 Stunden entspricht.

♦ Neue Eigenschaft MobileMode

Die Eigenschaft MobileMode wurde beim App-Objekt eingeführt, um die Unterstützung von Oberflächenobjekten hinsichtlich der Verwendung auf mobilen Geräten (Smartphone, Tablet) zu verbessern.

♦ Neue Ausprägung WinModeColorAll bei Eigenschaft ModeColor

Durch die neue Ausprägung WinModeColorAll ist es möglich, in der Farbauswahl zusätzlich zu den bei der Ausprägung WinModeColorFull vorhandenen Farben, auch die Werte WinColUndefined und WinColTransparent auszuwählen.

♦ Auslösen eines Laufzeitfehlers bei Deadlocks

Wird in der Eigenschaft Options des Sys-Objektes die Option DeadLockRTE gesetzt, wird bei Datensatzoperationen statt der

Kontakt

Rückgabe von rDeadlock der Laufzeitfehler ErrDeadLock ausgelöst.
Der Laufzeitfehler kann mittels try-Block abgefangen werden.

♦ Neupositionierung von Listen beim Ändern von Spalteneigenschaften verhindern

Mit der Option WinAppExtLstPosSelected der Eigenschaft FlagsExt des App-Objektes kann festgelegt werden, ob das Setzen von Spalteneigenschaften bei Listen-Objekten (RecList, DataList, StoList, etc.) zu einer Neupositionierung des selektierten Eintrages führt. Standardmäßig wird der selektierte Eintrag mittig positioniert. Durch Setzen von WinAppExtLstPosSelected wird eine Neupositionierung verhindert.

 Bei Änderung der Spalteneigenschaften in EvtLstDataInit erfolgt unabhängig von der Option keine Neupositionierung.

♦ Synchronisation von Druckjobs

Mit der neuen Ausprägung WinAppExtSyncPrint der Eigenschaft FlagsExt des App-Objektes kann die Synchronisation von parallel laufenden Druckjobs aktiviert werden. Standardmäßig ist diese ausgeschaltet, da dadurch die Performance der Druckausgabe beeinträchtigt werden kann und nur bei parallel laufenden Druckjobs benötigt wird.

♦ Verschlucken von Tastatureingaben während laufender Ereignisse verhindern

Mit der Option WinAppExtWaitKeyInput kann verhindert werden, dass Tastatureingaben verloren gehen, während Ereignisse durchgeführt werden. Die neue Option bewirkt, dass bei einem Tastendruck (sofern gerade eine Ereignisfunktion durchgeführt wird), die Nachrichtenverarbeitung in der Oberfläche angehalten wird. Diese wird fortgesetzt nachdem die Ereignisfunktion durchgeführt wurde. Dadurch gehen keine Benutzereingaben verloren, jedoch kann der Anwender während der Zeit auch keine Interaktion mit der Anwendung durchführen.

 WinSleep() verarbeitet trotz gesetzter WinAppExtWaitKeyInput-Option die Nachrichtenschleife. Der Aufruf von WinSleep() ohne WinMsgNoKeyboardInput oder WinMsgNoInput führt dazu, dass Tastendrücke wieder verloren gehen, sofern gerade Ereignisse durchgeführt werden.

♦ EvtMenuContext bei Rechtsklick auf Listen und TreeViews mit FocusByMouse

Ist bei einer Liste (RecList, DataList, etc.) oder einem TreeView die Eigenschaft FocusByMouse gesetzt, wird beim Rechtsklick auf einen neuen Knoten oder Listeneintrag das Ereignis EvtLstSelect bzw. EvtNodeSelect ausgelöst. Findet hier eine längere Verarbeitung statt, wird das Ereignis EvtMenuContext nicht ausgelöst. Dies kann zur Anzeige eines leeren oder falschen Kontextmenüs führen. Durch Setzen

Kontakt

der Ausprägung WinAppExtContextMenuSelect in der Eigenschaft FlagsExt des App-Objektes, wird in diesem Fall der Aufruf von EvtMenuContext verzögert.

- Wird diese Ausprägung gesetzt, kann es zu einem veränderten Verhalten beim Einsatz von WinMenuContext() (z. B. doppelt aufgerufene Kontextmenüs) kommen.

♦ Neue Eigenschaften beim Picture-Objekt zur Erstellung von Zeichnungen und Unterschriften

Das Picture-Objekt wurde um die neuen Eigenschaften SignMode, SignWidth und SignColor erweitert. Wird die Eigenschaft SignMode aktiviert, kann mit gedrückter linker Maustaste eine Zeichnung im Picture-Objekt erstellt werden. Dies kann auch zum Aufzeichnen von Unterschriften verwendet werden. Die erzeugte Zeichnung kann inklusive Bild mit dem ebenfalls neuen Befehl WinPicSaveImage() gespeichert werden.

♦ Gedrückte Maustaste im TreeView-Objekt ermitteln

Über die neue Konstante WinNodeMouseSelect kann mit dem Befehl WinInfo() im Ereignis EvtNodeSelect ermittelt werden, ob und wenn ja, mit welcher Maustaste ein Knoten selektiert wurde. Bei aktiverter Mehrfachauswahl bezieht sich die Information auch auf das Selektieren mehrerer Knoten bzw. deselektieren von Knoten.

• Designer

♦ Datensatzverwaltung / Vergleiche mit Feldern

In der Datensatzverwaltung kann nun auf der rechten Seite der Abfrage ebenfalls ein Feld der gleichen Tabelle angegeben werden.

♦ Unterstützung der Zertifizierung von Datenbankelementen

Der Client bietet nun eine Zertifizierungsunterstützung für Datenbankelemente. Aktuell können Dialoge, Menüs, Prozeduren und Tabellen validiert werden. Zum einen gibt es hierfür den Validierungs-Editor im Designer. Für anwendungsseitige Erweiterungen bezüglich der Validierung stehen darüber hinaus die Validierungsbefehle zur Verfügung. Die verwendeten Symbole stehen für die TileTheme-Modi WinTileThemeFlatColor und WinTileThemeFlatMono über die ImageTile mit den WinImgVal...-Konstanten zur Verfügung.

♦ Neue Ansichten des Editors

Der Editor wurde um eine Willkommensseite und ein MergeTool erweitert. Die Willkommensseite bietet einen Schnellzugriff auf wichtige Funktionen des Editors und wird immer geöffnet, wenn kein anderes Dokument geöffnet ist. Im MergeTool können Prozeduren und Texte verglichen werden. Dies ist zum Beispiel nützlich um unterschiedliche Versionen einer Prozedur miteinander zu vergleichen.

• Debugger

♦ **Detailansicht und Detailbearbeitung für Datentypen**

Im Debugger werden die Datentypen (außer handle), die entweder strukturiert sind, oder mehrere Interpretationsmöglichkeiten besitzen in einer Detailseite unter der Liste der überwachten Variablen dargestellt. Es wird jeweils die Detailseite für die ausgewählte Variable angezeigt. Die Interpretationsmöglichkeiten werden ebenfalls in der Prozeduransicht als Tooltip von Variablen angezeigt.

♦ **Anzeige der Zeilennummer in der Prozedurauftruf-Liste**

In der Liste der Prozederaufrufe des Debuggers wird die Zeilennummer angezeigt, aus der der Funktionsaufruf erfolgte.



Hierzu wird ebenfalls die Version 5.8.12 des jeweiligen Clients vorausgesetzt.

♦ **Suchen in der Prozeduransicht**

In der Prozeduransicht des Debuggers können nun mit den Tastenkombinationen + , + , + und + die entsprechenden Suchleisten für Text, Funktionen und Zeilennummern aufgerufen werden.

♦ **Debugger-Protokoll leeren**

Mit der neuen Konstante DbgTraceClear bei dem Befehl DbgControl() kann das Protokoll des Debuggers geleert werden.

• **Debug-Optionen**

♦ **Neue Debug-Option für lange dauernde Ladeoperationen von Frame-Objekten**

Wird beim Starten des Clients die neue Debug-Option 0x2000 angegeben, werden Ladeoperationen von Frame-Objekten protokolliert, die länger als eine definierte Zeitspanne dauern.

♦ **Neue Debug-Option für Protokollierung der GDI-Plus-Bibliothek**

Wird beim Starten des Clients die neue Debug-Option 0x4000 angegeben, werden die Initialisierung und Terminierung der GDI-Plus-Bibliothek protokolliert.

• **Performance-Optimierung für Dialoge mit vielen Unterobjekten**

Die Erstellung von Dialogen mit vielen Objekten und Unterobjekten, die in NotebookPage-Objekten enthalten sind, wurde optimiert. Bei großen Dialogen kann dies zu einem deutlichen Geschwindigkeitszuwachs bei der Dialogerstellung und -anzeige führen.

CONZEPT 16-Programmierschnittstelle - Version 5.8.12

• **Fehlende Zeichensatzwandlung für Zeichenketten bei der Eingabe**

Kontakt

Für die folgenden Funktionen wurde die fehlende Zeichensatzwandlung des Namens hinzugefügt:

Funktion	Interne API	Externe API
<u>C16_FileInfoByName</u>		✗
<u>C16_SbrInfoByName</u>		✗
<u>C16_KeyInfoByName</u>		✗
<u>C16_LinkInfoByName</u>		✗
<u>C16_FldInfoByName</u>		
<u>C16_FldInfoExtByName</u>	✗	



Diese Änderung kann zu Inkompatibilitäten mit bestehenden DLLs führen.

• Unterstützung weiterer Befehle

In der Programmierschnittstelle ohne graphische Erweiterung werden nun zusätzlich folgende Befehle unterstützt:

- ◆ DllCall()
- ◆ DllLoad()
- ◆ DllUnload()
- ◆ FsiFileInfo() (mit einer Hash-Operation)
- ◆ FsiFileProcess()
- ◆ JsonLoad()
- ◆ JsonSave()
- ◆ MailOpen() (mit einer _SckTls...-Konstanten)
- ◆ MemDecrypt()
- ◆ MemEncrypt()
- ◆ MemGenKeyPair()
- ◆ MemHash()
- ◆ MemHMAC()
- ◆ MemSign()
- ◆ MemVerify()
- ◆ SckConnect() (mit einer _SckTls...-Konstanten)
- ◆ SckStartTls()



Zur Benutzung der Programmierschnittstelle werden zusätzlich die Dateien c16_ssl_w32.dll und c16_ssl_w64.dll benötigt.

CONZEPT 16-Release 5.8.11 - 11. März 2019



Die Clients der Version 5.8.11 benötigen den Server der Version 5.8.11

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Server
- Client
- SOA-Service
- Druckprozessor
- Programmierschnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.8.11

• CONZEPT 16 unter Linux

 Für den Betrieb des CONZEPT 16-Servers, des SOA-Services und der PHP-Schnittstelle unter Linux wird mindestens der Kernel 3.x und die glibc 2.17 benötigt. Der Kernel 2.x wird **nicht** mehr unterstützt.

• Änderung an vpYear, vpMonth und vpDay

 Die Eigenschaften vpYear, vpMonth und vpDay von den Datentypen date und caltime wurden so verändert, dass beim Setzen einer Komponente die jeweils anderen Komponenten angepasst werden, sofern diese einen ungültigen Inhalt besitzen. Damit wird erreicht, dass immer ein gültiges Datum entsteht. Diese Änderung fand bereits in der Version 5.8.10 statt.

• Neue Version der PDF-Bibliothek

Die PDF-Bibliothek wurde auf Version 4.0.25.69 aktualisiert.

Diese Version enthält wichtige Fehlerbehebungen.

 Die c16_pdfw.dll kann nicht mit einem Client betrieben werden, dessen Version älter ist, als die der c16_pdfw.dll, gleichgültig ob Standard-, Advanced- oder API-Client.

• Aktualisierung der OpenSSL-Bibliothek

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.1.1a aktualisiert. Diese Version unterstützt TLS 1.3 und enthält zudem Fehler- und Sicherheitskorrekturen.



Inkompatibilitäten zu vorherigen Versionen:

- ◆ Cipher Suites mit RC4 / 3DES werden in keinem der Modi mehr unterstützt.
- ◆ Der RSA-Schlüssel des Server-Zertifikats muss nun mindestens 1024 Bit lang sein.
- ◆ Der Schlüsselaustausch findet bei SckTlsHigh nicht mehr mittels RSA statt.

Die Liste der aktuell unterstützten Ciphersuites kann bei den Konstanten SckTlsLow, SckTlsMed, SckTlsHigh und SckTlsMax eingesehen werden.

• Sammlung von Stammzertifikaten aktualisiert

Die Sammlung von Stammzertifikaten (siehe Speicherorte von Konfigurationsdateien) wurde auf die Version 64 von Mozilla Firefox aktualisiert. Diese Datei wird jedoch nur aktualisiert, wenn sie seit der Installation nicht um eigene Stammzertifikate erweitert wurde.

CONZEPT 16-Server - Version 5.8.11

• Lizenz

 Unabhängig vom eingesetzten Betriebssystem wird zum Betrieb der 5.8.11 eine Lizenz vom Stand 5.8.11 (siehe Control-Center Eintrag Lizenz ->

Kontakt

Stand) benötigt. Dies erfordert einen aktiven Softwarepflegevertrag. Ist die automatische Aktualisierung der Lizenzdatei (siehe [Konfiguration](#)) aktiviert, ist kein manueller Eingriff erforderlich.

• Reservierte Benutzer für benötigte Hintergrund-Dienste

In der [Konfiguration des Servers](#) kann mit der neuen Einstellung [UserResvCount](#) die Anzahl der reservierten Benutzer festgelegt werden. Diese Anzahl an Benutzern kann sich nicht mit [Standard](#)- oder [Advanced-Client](#) an der Datenbank anmelden. Damit wird sichergestellt, dass eine ausreichende Anzahl von lizenzierten Benutzern für andere Clients ([SOA-Service](#), [DLL-Schnittstelle](#), [ODBC-Schnittstelle](#) etc.) reserviert ist.

-  Ist die Anzahl der reservierten Benutzer gleich oder größer als die Anzahl der lizenzierten Benutzer, steht für [Standard](#)- bzw. [Advanced-Client](#) dennoch ein Benutzer zur Verfügung.

• Log-Eintrag bei zwangsweiser Benutzerabmeldung

Beim Abmelden eines Benutzers im [Standard-Client](#) (Menübefehl Info/Benutzer) oder per Prozedurbefehl [UserClear\(\)](#) wird nun der folgende (informative) [Log-Eintrag](#) [Forced user logoff <Benutzername> \(<Rechnername>\)](#) im Protokoll des Datenbankprozesses angelegt.

• Log-Eintrag Invalid Tree Root für temporäre Bäume

Der Log-Eintrag [Invalid Tree Root](#) wird für temporäre Bäume nicht mehr protokolliert. Statt dessen wird ab Client-Version 5.8.11 der Fehler [_rFailed](#) und andernfalls [_rNoRec](#) bei dem jeweiligen Befehl zurückgegeben.

CONZEPT 16-Client - Version 5.8.11

• Laufzeitfehler bei [TextWrite\(\)](#) bei Erreichen des Limits

Wird bei Verwendung von [TextWrite\(\)](#) das Limit für interne Texte und Prozeduren von ca 119 MB überschritten, wurde das Dokument bisher überschrieben. Nun wird der Laufzeitfehler [_ErrLimitExceeded](#) ausgelöst. Das Dokument ist bis zu der Stelle vor Erreichen des Limits gespeichert.

• Neuer Barcode-Type [WinBarcode2dGS1_DataMatrix](#)

Das [Barcode](#)- und das [PrtBarcode](#)-Objekt unterstützen nun den Typ GS1 DataMatrix. Dieser wird mit der Konstante [WinBarcode2dGS1_DataMatrix](#) in der Eigenschaft [TypeBarcode](#) definiert. Das benötigte FNC1-Steuerzeichen zu Beginn wird automatisch gesetzt. Als Trennzeichen zwischen Zeichenketten variabler Länge kann das "FNC1 Codewort" oder der "Group Separator" verwendet werden. Diese werden in CONZEPT 16 wie folgt erzeugt: FNC1: StrCnv(StrChar(129), _StrFromANSI), Group Separator: StrCnv(StrChar(29), _StrFromANSI).

• Rechteckige Darstellung von DataMatrix-Barcodes

Wird ein Barcode vom Typ DataMatrix ([WinBarcode2dDataMatrix](#) / [WinBarcode2dGS1_DataMatrix](#)) erzeugt und die [AreaHeight](#) beträgt weniger als zwei Drittel der [AreaWidth](#), wird der Barcode in der rechteckigen Variante

mit zwei Teilen nebeneinander dargestellt.

- **Breakpoints im Editor setzen**

Das Kontextmenü des Editors zum Setzen und Entfernen von Breakpoints erscheint nun zusätzlich zur Lesezeichen- und Breakpointspalte auch in der Zeilenummernspalte und in der Foldingspalte.

-  bei automatischer Klammerung im Editor

Wird im Editor eine öffnende runde Klammer hinzugefügt, merkt sich der Editor die aktuelle Zeile. Wird nun  gedrückt, bevor diese Zeile verlassen wurde und das letzte Zeichen der Zeile ist eine runde schließende Klammer, wird der Cursor hinter diese Klammer gesetzt. Wird die Tastenkombination  +  gedrückt, wird stattdessen an der aktuellen Position ein Zeilenumbruch eingefügt und sich die neue Zeile für das  gemerkt.

- **Anzeige der Dokumentation**

Durch ein Windows-Update kann der Inhalt von CHM-Hilfdateien nicht mehr angezeigt werden, wenn diese auf einem Netzlaufwerk gestartet wird. Wird der Client nicht lokal ausgeführt, wird die Dokumentation automatisch nach %AppData%\CONZEPT 16\ kopiert und von dort angezeigt.

- **Tastenkürzel für Bearbeitungsmodus**

Für den Bearbeitungsmodus von Dokumenten im Editor und Fenstern und Druckobjekten im Designer wurde das Tastenkürzel  +  +  hinzugefügt.

- **Prozeduren im Editor mit zusammengeklappten Funktionsblöcken öffnen**

In den Einstellungen kann definiert werden, ob beim Öffnen eines Dokumentes die Folding-Punkte geschlossen sein sollen. Standardmäßig sind alle Punkte offen. Befand sich der Cursor vor dem Schließen des Designers innerhalb einer Funktion, wird der Funktionsblock, in dem der Cursor steht, immer geöffnet.

- **Neue Eigenschaft JobProcExtended auch bei Sys-Objekt**

Beim Sys-Objekt kann nun zusätzlich die Eigenschaft JobProcExtended abgefragt werden, um zu ermitteln, ob die aktuell ausgeführte Prozedur die Ausführung von Oberflächen-Befehlen unterstützt oder nicht.

- **Server Name Indication bei SckConnect(), SckStartTls() und MailOpen()**

Bei den Befehlen SckConnect(), SckStartTls() und MailOpen() kann mit der Option SckTlsSNI die verschlüsselte Verbindung mittels Server Name Indication (SNI) aufgebaut werden.

- **Fehlerwerte bei RmtCall()**

RmtCall() wurde um einen Rückgabewert erweitert. Ist der RmtCall()-Aufruf erfolgreich, wird die ID des neuen Benutzers zurückgeliefert, sonst ein Fehlercode.



RmtCall() wartete in der Version 5.8.10 solange, bis ein Benutzer frei wurde, falls das Prozedurlimit erreicht war. Nun wird der Fehlercode

ErrLimitExceeded zurückgegeben.

- **Prozedurlimit per DbaInfo() ermitteln**

DbaInfo wurde um die Konstanten DbaRmtProcLimit und DbaRmtProcCount erweitert um das Prozedurlimit und die Anzahl der in Ausführung befindlichen RmtCall()-Aufrufe zu ermitteln.

-  Wird vor dem Aufruf von RmtCall() die Anzahl der bereits benutzten RmtCall()-Aufrufe mit DbaInfo(_DbaRmtProcCount) mit dem Prozedurlimit DbaInfo(_DbaRmtProcLimit) verglichen, kann trotzdem der Fehlercode ErrLimitExceeded zurückgegeben werden, auch wenn das Limit zuvor nicht erreicht war. In diesem Fall wurde in der Zwischenzeit durch einen anderen Client ein RmtCall() gestartet.

- **Angabe eines Themes beim Starten von Designtools**

Beim Starten eines Designtools kann nun mit dem Argument c16theme zusätzlich eines der Standard-Themes ("OfficeBlue" / "OfficeDark") angegeben werden. Sofern Windows unter hohem Kontrast läuft, wird das Argument ignoriert. Beim Starten der Designtools aus dem Designer wird automatisch das aktuell ausgewählte Theme verwendet.

- **Erweiterung der Theme-Objekte um ThemeSets**

Innerhalb von Theme-Objekten können mit WinThemeSetOpen() ThemeSets sowie im Theme-Editor angelegt werden. Diese ermöglichen Abweichungen vom definierten Theme für einzelne Eigenschaften. Ein ThemeSet verfügt über die gleichen Eigenschaften, wie ein Theme. Alle nicht gesetzten Eigenschaften werden je nach gesetztem Wert in den Eigenschaften ThemeSetId, StyleTheme und ThemeName vom ThemeSet des Elternobjektes oder vom normalen Theme übernommen.



Pro Theme können beliebig viele ThemeSets angelegt werden. Im Theme-Editor können jedoch immer nur die ersten 10 ThemeSets bearbeitet werden.

- **Neue Theme-Eigenschaft ThemeSizeFixed**

Mit der neuen Theme-Eigenschaft ThemeSizeFixed kann für die Titelzeile von AppFrame, Frame und MdiFrame-Objekte, sowie für die Registerreiter von Notebook-Objekten eine feste Höhe definiert werden. Die Schriftgröße der jeweiligen Objekte passt sich automatisch an den verfügbaren Platz an, sofern die Eigenschaft einen Wert größer 0 besitzt.

- **Neuer Befehl WinFlash()**

Mit dem neuen Befehl WinFlash() kann der Anwender darauf hingewiesen werden, dass die Anwendung seine Aufmerksamkeit erfordert. Im Regelfall wird der Befehl verwendet, wenn die Anwendung nicht im Vordergrund oder minimiert ist.

- **Unterstützung für hohen Kontrast im Modern Theme Style**

Sofern eine Anwendung unter Modern Theme Style läuft, wird empfohlen, beim Starten der Anwendung zu prüfen, ob Windows im Modus "Hoher Kontrast"

Kontakt

läuft. Sofern dies der Fall ist, sollte das Theme 'WindowsColor' für die Anwendung aktiviert werden. Für die Überprüfung wurde WinInfo() um die neue Option WinHighContrast erweitert.

Im Designer, Debugger und den Designer-Tools wird bei aktiviertem hohem Kontrast automatisch das Theme 'WindowsColor' verwendet. Die Theme-Auswahl im Designer und Debugger wird in diesem Fall in den Einstellungen deaktiviert.

-  Bei einem Wechsel der Windows-Einstellung in oder vom Modus "Hoher Kontrast" muss der Designer bzw. der Debugger neu gestartet werden, damit sich der Modus auswirkt.

Wird die Installationsroutine bei aktiviertem hohem Kontrast ausgeführt, verwendet diese ebenfalls die definierten Farben von Windows.

- **Neue Farb-Konstante WinColHotLight**

Die neue Farb-Konstante WinColHotLight definiert die von Windows vorgegebene Farbe für Links und hervorgehobene Elemente.

- **Feldnamen mit nicht alphanumerischen Zeichen im Debugger anklickbar**

Im Debugger können nun auch Felder mit nicht alphanumerischen Zeichen bzw. in der Feldnummernnotation ("<Tabellennummer>,<Teildatensatznummer>,<Feldnummer>") mit Doppelklick gesucht und überwacht werden. Im Designer wird mit der Tastenkombination  +  auf das entsprechende Feld in der Datenbankinfo positioniert.

- **Unicode-Sequenzen bei JsonSave() und XmlSave()**

JsonSave() und XmlSave() wandeln nun das Zeichen \ bzw. & nicht mehr um, wenn ihnen eine Unicode-Zeichen-Sequenz folgt.

- **Tausendertrennung im Chart-Objekt**

Mit der Eigenschaft ChartNumFmtThousandSep des Chart-Objektes kann ein Tausendertrennzeichen definiert werden.

- **Neues Objekt Slider**

Bei dem neuen Objekt Slider handelt es sich um einen Schieberegler zur Auswahl eines ganzzahligen Wertes aus einem vorgegebenen Bereich.

- **Zuletzt geöffnete Menüs und Themes**

Im Designer werden im Menüpunkt **Datei / Öffnen über Liste** nun auch zuletzt geöffnete Menüs und Themes aufgelistet.

- **Übernahme von Werten einer Popup-Liste bei Bearbeitung im RecView**

Beim Starten des Editievorgangs im RecView wird das Ereignis EvtLstEditActivate aufgerufen. Dem dort übergebenen Edit-Objekt kann im Ereignis EvtLstEditStartItem eine Populiste zugeordnet werden. Wird in der Populiste über die Eigenschaft LstFlags die Option WinLstEditClose gesetzt, wird beim Selektieren eines Eintrags mittels Maus, oder der definierten Taste

(EditKeyEnd), der Eintrag übernommen und die Editierung abgeschlossen.

- **Neue Option XmlLoadHugeTextNode bei XmlLoadO**

Bei XmlLoad kommt es beim Laden von Dateien mit vielen Daten in einem Knoten (mehr als 10 MB) zu einem Fehler. Der Befehl XmlError(_XmlErrorText) liefert in dem Fall die Meldung xmlSAX2Characters: huge text node. Nun kann bei XmlLoad() als Option XmlLoadHugeTextNode angegeben werden, um auch Knoten mit vielen Daten zu laden. Die Daten des Knotens können anschließend mit CteNodeValueAlpha(.... _CteNodeValueRead) ausgelesen werden.

CONZEPT 16-SOA-Service - Version 5.8.11

- **Minidump-Generierung im SOA-Service mit graphischer Erweiterung**

Im SOA-Service mit graphischer Erweiterung (c16_proc_extended = Y) kann die Generierung von Minidumps aktiviert werden. Dies geschieht über die Einträge AppLogOptions und AppLogPath in der Konfigurationsdatei. Im SOA-Service ohne graphische Erweiterung werden die Einträge ignoriert, da hier bereits Minidump-Dateien angelegt werden.

- **Angabe eines Ports für den Alert-Mail-Versand**

Mit dem neuen Konfigurationseintrag alert_mail_port kann der Port definiert werden, über den der Mailserver für den Versand der Alert-Mail kontaktiert wird.

CONZEPT 16-Druckprozessor - Version 5.8.11

- **Minidump-Generierung im Druckprozessor**

Im Druckprozessor kann die Generierung von Minidumps aktiviert werden. Dies geschieht über die Einträge AppLogOptions und AppLogPath in der Konfigurationsdatei.

CONZEPT 16-Programmierschnittstelle - Version 5.8.11

- **Erweiterte Programmierschnittstelle für Visual Studio 2005 und neuer**

Die Programmierschnittstelle fehlte in der erweiterten Version für Visual Studio. Die Import-Bibliothek für Visual Studio 2005 und neuer steht in der Datei \API\lib\c16_pgxe.mvc bereit.

CONZEPT 16-Release 5.8.10 - 31. Juli 2018



Die Clients der Version 5.8.10 benötigen den Server der Version 5.8.10

Für den Betrieb des Clients werden die Laufzeitkomponenten für Visual C++ 2015 nicht mehr benötigt.

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten

- Client
- Laufwerkstreiber

CONZEPT 16 - Alle Komponenten - Version 5.8.10

• Wegfall von RSA-Ciphersuites bei SckTlsHigh

Bei der Konstanten SckTlsHigh und dem tls_security_level=high in den Konfigurationsdateien des SOA-Services sind die Ciphersuites mit RSA-Schlüsselaustausch aufgrund fehlender Forward Secrecy weggefallen. Dies betrifft die folgenden Suites:

- ◆ CAMELLIA256-SHA
- ◆ AES256-GCM-SHA384
- ◆ AES256-SHA256
- ◆ AES256-SHA
- ◆ CAMELLIA128-SHA
- ◆ AES128-GCM-SHA256
- ◆ AES128-SHA256
- ◆ AES128-SHA

• Neue Version der PDF-Bibliothek

Die PDF-Bibliothek wurde auf Version 4.0.20.51 aktualisiert.

Die bisherige Bibliothek erlaubte keine Generierung von PDF/A-1b-Dateien mit transparenten Bildern. Die aktuelle Version wandelt Bilder mit transparentem Hintergrund um, so dass ein nicht transparentes Bild entsteht und erzeugt eine PDF/A-1b-Datei mit den umgewandelten, nicht transparenten Bildinhalten. Zusätzlich sind Fehlerkorrekturen enthalten.



Die c16_pdfw.dll kann **nicht** mit einem Client vor Version 5.8.09d betrieben werden, gleichgültig ob Standard-, Advanced- oder API-Client

• Aktualisierung der OpenSSL-Bibliothek

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.2o aktualisiert. Diese Version enthält Fehler- und Sicherheitskorrekturen.

• Sammlung von Stammzertifikaten aktualisiert

Die Sammlung von Stammzertifikaten (siehe Speicherorte von Konfigurationsdateien) wurde auf die aktuelle Version von Mozilla Firefox aktualisiert. Diese Datei wird jedoch nur aktualisiert, wenn sie seit der Installation nicht um eigene Stammzertifikate erweitert wurde.

• Änderung an vpYear, vpMonth und vpDay



Die Eigenschaften vpYear, vpMonth und vpDay von den Datentypen date und caltime wurden so verändert, dass beim Setzen einer Komponente die jeweils anderen Komponenten angepasst werden, sofern diese einen ungültigen Inhalt besitzen. Damit wird erreicht, dass immer ein gültiges Datum entsteht.

CONZEPT 16-Client - Version 5.8.10

- **Neuer Befehl WinIconPreload()**

Mit dem Befehl WinIconPreload() können Icons aus der Datenbank in den Cache geladen werden, um die Anzeige von vielen Icons bei Dialogstart zu beschleunigen.

- **Neue Option SelKeyUpdate bei SelRead()**

Mit der Option SelKeyUpdate bei SelRead() werden Datensätze mit geänderten Schlüsselwerten, die über RecRead() gelesen werden, in der Selektion aktualisiert.

- **Neue Option RecUpdateCounter bei RecInfo()**

Mit der Option RecUpdateCounter bei RecInfo() kann der Update-Zähler einer Datei ermittelt werden. Dieser wird beim Öffnen der Datenbank sowie nach der Datenbankreorganisation mit 0 initialisiert. Der Zähler wird bei Verwendung von RecInsert(), RecDelete(), RecDeleteAll() und RecReplace() erhöht. Dies kann dazu verwendet werden, um festzustellen, ob Änderungen an einer Datei durchgeführt wurden.

- **Erweiterte Auto vervollständigungsliste**

Die Auto vervollständigungsliste des Editors ( + ) wurde um die Elemente der Datenstruktur (Tabellen, Felder, etc.) erweitert.

- **Neue Optionen bei PdfClose() und PdfMode**

Mit den Optionen PdfModePdfA2b, PdfModePdfA2bNormal, PdfModePdfA3b und PdfModePdfA3bNormal bei dem Befehl PdfClose() und der Eigenschaft PdfMode können PDF/A-Dokumente mit Konformität zu PDF/A-2b bzw. PDF/A-3b erzeugt werden. Schlägt dies fehl, wird bei PdfModePdfA2bNormal und PdfModePdfA3bNormal ein normales PDF-Dokument erzeugt.

- **Automatische Klammer vervollständigung im Editor ausstellbar**

In den Einstellungen kann die automatische Klammer vervollständigung ausgeschaltet werden.

- **Variablen suche im Debugger**

In der Suche im Bereich Alle Variablen im Debugger ist nun Multiselect möglich. Zusätzlich kann die Suche auf bestimmte Datentypen (z. B. alpha) und Datenbereiche (z. B. lokale Variablen) eingeschränkt werden.

- **Name von bearbeiteten Variablen im Debugger**

Wird eine Variable im Debugger bearbeitet, wird der Name jetzt im Titel des Bereiches "Überwachte Variablen" angezeigt. Somit ist der Name auch sichtbar, wenn die Inhaltsspalte sehr breit gezogen ist.

- **AreaWidth und AreaHeight nicht mehr virtuelle Eigenschaften**

Die Eigenschaften AreaWidth und AreaHeight sind nun keine virtuellen Eigenschaften mehr. Sie stehen nun als richtige Eigenschaften bei allen Objekten bereit, die auch die Eigenschaft Area besitzen. Über die Eigenschaft LockAreaSize des Application-Objektes wird hierbei gesteuert, ob beim Ändern von AreaLeft / AreaTop die Eigenschaften AreaRight / AreaBottom (LockAreaSize = true) oder die Eigenschaften AreaWidth / AreaHeight

Kontakt

(LockAreaSize = false) verändert werden sollen. Über das Kontextmenü der Eigenschaftsliste kann mit dem Menüpunkt "Größe fixieren" das Verhalten von LockAreaSize ebenfalls im Designer beeinflusst werden.

- **Auf Beenden-Signal von Windows reagieren**

Neues Ereignis EvtEndSession bei Frame, AppFrame und TrayFrame, welches ausgelöst wird, wenn der Benutzer der Windows-Sitzung abgemeldet werden soll, oder der Rechner heruntergefahren bzw. neu gestartet wird. Mit dem Ereignis kann das Abmelden nicht abgebrochen werden. Nach standardmäßig 5 Sekunden erhält der Benutzer eine Liste der Programme, die noch nicht beendet sind und kann hier dennoch alle Programme ohne weitere Rückfrage im Programm beenden.

- **Filterung in der Palette**

In der Palette besteht nun die Möglichkeit der Filterung nach Objekttyp. Hierbei werden die Wildcard-Operatoren * und ? unterstützt. Die Filterergebnisse werden auf der neuen Registerseite "Filter" angezeigt.

- **Ausführung von Oberflächen-Befehlen**

Mit der neuen Eigenschaft spJobProcExtended kann ermittelt werden, ob ein SOA-Task oder eine durch JobStart() ausgeführte Ereignisfunktion die Ausführung von Oberflächen-Befehlen unterstützt. Die Eigenschaft ist sowohl beim Job- als auch beim JobControl-Objekt vorhanden.

CONZEPT 16-Laufwerkstreiber - Version 5.8.10

- **Aktualisierte Version**

Der Laufwerkstreiber wurde auf die Version 6.1.148 aktualisiert. Diese Version enthält Fehlerkorrekturen.

CONZEPT 16-Release 5.8.09 - 19. Februar 2018



Die Clients der Version 5.8.09 benötigen den Server der Version 5.8.09



Für den Betrieb des Clients wird die Laufzeitkomponente für Visual C++ 2015 benötigt.

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Installationsroutine
- Client

CONZEPT 16 - Alle Komponenten - Version 5.8.09

- **Neue Version der PDF-Bibliothek**

Die PDF-Bibliothek wurde auf Version 4.0.13.35 aktualisiert.

- **Aktualisierung der OpenSSL-Bibliothek**

Kontakt

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.2n aktualisiert.
Diese Version enthält Fehler- und Sicherheitskorrekturen.

- **Sammlung von Stammzertifikaten aktualisiert**

Die Sammlung von Stammzertifikaten (siehe [Speicherorte von Konfigurationsdateien](#)) wurde auf die aktuelle Version von Mozilla Firefox aktualisiert. Diese Datei wird jedoch nur aktualisiert, wenn sie seit der Installation nicht um eigene Stammzertifikate erweitert wurde.

CONZEPT 16-Installationsroutine - Version 5.8.09

- **Laufzeitkomponenten extrahieren**

Die mitgelieferten Laufzeitkomponenten können aus der Installationsroutine über die Kommandozeile mit dem neuen Schalter /cpn_red=1 bei der Aktion /cpn_action=extract extrahiert werden. Somit können die benötigten Laufzeitkomponenten auch auf anderen Rechnern installiert werden, wenn der Client nur über ein Netzlaufwerk bereit gestellt wird.

CONZEPT 16-Client - Version 5.8.09

- **Neuer Editor für Prozeduren und Texte**

Der bestehende Editor wurde durch ein neues Objekt ersetzt. Dieses unterstützt beispielsweise folgende neue Funktionen:

- ◆ Code-Folding
- ◆ Zooming
- ◆ Markierung von Textvorkommen
- ◆ Auto vervollständigungsliste mit Typ-Bildchen
- ◆ Highlighting für XML- und JSON-Dokumente

Zudem steht der neue Editor als Oberflächenobjekt (CodeEdit) zur Verfügung.

- **Neues Barcode-Objekt**

Mit dem neuen Objekt Barcode können, analog zu dem PrtBarcode-Objekt, Barcodes auch in der Oberfläche angezeigt werden. Barcodes können zudem mit dem Befehl WinBarcodeSaveImage() als Bild gespeichert werden.

- **Neue Option bei WinLayer()**

Mit der neuen Option WinLayerDCross wird im Layer-Modus das Fenster mit einer Kreuz-Schraffur überdeckt.

- **Designertools sperren Designer nicht mehr**

Die Designertools (Benutzerverwaltung, BLOB-Verwaltung, Datensatzverwaltung und Ressourcenverwaltung) sperren den Designer nicht mehr beim Aufruf. Diese Tools können zudem beim Start des Advanced-Client als Kommandozeilenargument /C16DesignTool angegeben werden.

- **Modern Theme Style im Designer und Debugger**

Kontakt

Der Designer und der Debugger werden nun im Modern Theme Style dargestellt. In den Einstellungen kann zwischen den Standard-Themes OfficeBlue und OfficeDark ausgewählt werden. Zusätzlich werden neue Symbole verwendet. Diese werden standardmäßig bunt gezeichnet. In den Einstellungen kann mit "Symbole passend zum Theme einfärben" definiert werden, dass alle Symbole einfarbig mit der Schriftfarbe des jeweiligen Objektes dargestellt werden.

• **Neue Ausprägungen für TileTheme**

Die Eigenschaft TileTheme des App-Objektes wurde um die Konstanten WinTileThemeFlatColor und WinTileThemeFlatMono erweitert. Diese Symbole sind an Windows 10 bzw. Office 2016 angelehnt. Mit der Option WinTileThemeFlatMono werden die Symbole einfarbig in der Schriftfarbe des jeweiligen Theme-Objektes dargestellt.

• **Update einspielen trotz abweichender Datenstruktur**

Das OEM-Kit erlaubt jetzt auch das Einspielen von Feldern trotz abweichender Datenstruktur. Ist ein Feld mit dem Feldnamen bereits in der Zieldatenbank vorhanden, wird das neue Feld ohne Name im Teildatensatz eingefügt.

• **Anwendungsprotokoll im Standard- und Advanced-Client**

Das Anwendungsprotokoll kann nun auch im Standard- und Advanced-Client (zusätzlich zum Designer) aktiviert werden. Hierzu muss der Konfigurationseintrag AppLogOptions oder das Kommandozeilenargument C16AppLogOptions den Wert 4 enthalten.

Ist das Anwendungsprotokoll aktiviert oder wird die Applikation aus dem Designer gestartet, werden bei Ausnahmen Minidump-Dateien generiert. Der Pfad zum Minidump wird im Eintrag im Anwendungsprotokoll protokolliert.

• **Ermittlung der Buildnummer mit SysOS()**

Zusätzlich zum Betriebssystem kann mit SysOS() optional unter Windows auch die Buildnummer ermittelt werden. Weiterhin wird die Buldnummer an folgenden Stellen angezeigt bzw. protokolliert:

- ◆ Log-Viewer - Eigenschaften der Log-Datei
- ◆ Log-Eintrag: "System information"
- ◆ Designer / Debugger im Info-Dialog
- ◆ Standard-Client (Info / System)
- ◆ Web-Administration (Status)

CONZEPT 16-Release 5.8.08 - 18. September 2017



Die Clients der Version 5.8.08 benötigen den Server der Version 5.8.08

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Client

CONZEPT 16 - Alle Komponenten - Version 5.8.08

- **Sammlung von Stammzertifikaten aktualisiert**

Die Sammlung von Stammzertifikaten (siehe [Speicherorte von Konfigurationsdateien](#)) wurde auf die aktuelle Version von Mozilla Firefox aktualisiert. Diese Datei wird jedoch nur aktualisiert, wenn sie seit der Installation nicht um eigene Stammzertifikate erweitert wurde.

- **Aktualisierung der OpenSSL-Bibliothek**

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.2l aktualisiert.

CONZEPT 16-Client - Version 5.8.08

- **Neue Eigenschaft Utf8 beim PrtBarcode-Objekt**

Mit der neuen Eigenschaft Utf8 beim PrtBarcode-Objekt kann angegeben werden, ob QR-Codes im UTF8-Zeichensatz erstellt werden (default) oder im Zeichensatz ISO-8859-1. Bei allen anderen Barcode-Typen wird die Eigenschaft ignoriert.

- **Datenstruktureditor und Datenbankinfo: Offene Knoten nicht merken**

Über die neue Einstellung im Designer Anzeige der Datenbankäume wiederherstellen kann die Speicherung der offenen Knoten im Datenstruktureditor und in der Datenbankinfo unterbunden werden. Standardmäßig ist die Option gesetzt.

- **Eigenes Symbol beim InputCtrl für DateEdit-, DecimalEdit- und FloatEdit-Objekte**

Für DateEdit, DecimalEdit und FloatEdit kann nun ein eigenes Symbol für das Input-Control (Eigenschaft InputCtrl) definiert werden. Hierfür gibt es die neuen Eigenschaften Icon, ImageTileUser, ImageTile und TileSize. Im Falle von ImageTileUser werden die Bilder aus den Eigenschaften TileName... ermittelt.

- **Erweiterung von Modern Theme Style**

Der Modern Theme Style wurde um das GanttGraph-Objekt und weitere Eigenschaften erweitert.

- **Neue Eigenschaften für WinDialogBox()**

Das App-Objekt wurde um die Eigenschaften DialogBoxFlags, DialogBoxFont, DialogBoxFontButton und DialogBoxTextButton erweitert. Mit diesen Eigenschaften können Schriftarten- und Größen, sowie Texte für Schaltflächen in WinDialogBox() verändert werden. Im Modern Theme Style werden die Eigenschaften DialogBoxFont und DialogBoxFontButton ignoriert.

- **Neue Ausprägung WinAppExtLayoutComplete der Eigenschaft FlagsExt**

Die Eigenschaft FlagsExt des App-Objektes wurde um die Ausprägung WinAppExtLayoutComplete erweitert, durch die das Ereignis EvtPosChanged bei Frame- bzw. AppFrame-Objekten erst auslöst wird, nachdem der FrameClient positioniert und in der Größe angepasst wurde. Standardmäßig hat der FrameClient bei Auslösung des Ereignisses noch die ursprüngliche Größe.

- **WinInfo() mit WinInteriorHeight / WinInteriorWidth bei**

Frame-Objekten

Mit WinInfo() und den Optionen _WinInteriorHeight und _WinInteriorWidth kann nun auch bei Frame-Objekten die Größe des FrameClient-Objektes ermittelt werden. Bisher war dies nur bei AppFrame-Objekten möglich.

- **CteNode-Objekte mit alphanumerischem Inhalt**

Die Länge eines CteNode-Objektes mit alphanumerischem Inhalt (spType = TypeAlpha) ist nicht längenlimitiert. Mit der Eigenschaft spValueAlpha lassen sich jedoch maximal 65520 Byte schreiben bzw. lesen. Um dies zu umgehen, wurde der Befehl CteNodeValueAlpha() hinzugefügt. Dieser schreibt und liest auch alphanumerische Inhalte mit mehr als 65520 Byte.

- **CSV-Import in der Datensatzverwaltung**

Über die Datensatzverwaltung können nun auch CSV-Dateien importiert werden. Das Format für CSV-Dateien entspricht dem Format beim CSV-Export.

- **Typkonvertierung beim Import in der Datensatzverwaltung**

Der Import von Dateien im Format C16R wurde erweitert, so dass Quell- und Zieldatentyp der zu importierenden Felder abweichen kann.

- **EvtKeyItem für Umlaute auslösen**

Über die Option WinAppExtUmlautKeySupport der Eigenschaft FlagsExt kann erreicht werden, dass das Ereignis EvtKeyItem auch für Umlaute generiert wird.

- **Standard-Icons für Modern Theme Style**

Die Standard-Icons von Dialogen und im Icon-Objekt (siehe TypeIcon) werden automatisch in einem flachen Stil gezeichnet, wenn der Modern Theme Style für das jeweilige Objekt aktiviert ist. Dies betrifft auch das Icon in WinDialogBox(), wenn diese im Modern Theme Style dargestellt wird.

- **Optimierter Zugriff auf Eigenschaften**

Das Setzen von Win-, Com- und Prt-Properties wurde optimiert. Bei Eigenschaften dieser Art ist ein Thread-Wechsel notwendig, der Synchronisierungen notwendig macht. Das Setzen der genannten Eigenschaften erfolgt jetzt ca. doppelt so schnell. Das macht sich hauptsächlich dann bemerkbar, wenn sehr viele Eigenschaften hintereinander gesetzt oder gelesen werden.

- **Eigenschaft spURI des HTTP-Objektes**

Die Eigenschaft spURI wurde auf 4095 Byte vergrößert.

CONZEPT 16-Release 5.8.07a - 12. Mai 2017

Die Version enthält Fehlerkorrekturen. Bitte hierzu die Datei c16_error.txt im Installationsverzeichnis lesen.

CONZEPT 16-Release 5.8.07 - 28. April 2017

 **Windows 2000 wird ab diesem Release nicht mehr unterstützt.**

 Die Clients der Version 5.8.07 benötigen den Server der Version 5.8.07

An folgenden Komponenten wurden Änderungen durchgeführt:

- Server
- Client

CONZEPT 16-Server - Version 5.8.07

- **Neues Script-Utility-Kommando down**

Mit dem Kommando down des Script-Utilities werden alle Benutzer abgemeldet. Diese Funktion ist gleichbedeutend mit Alle Benutzer abmelden in der Web-Administration. Der Server muss hierfür ein Kennwort haben, welches auch bei der Ausführung des Kommandos angegeben werden muss.

CONZEPT 16-Client - Version 5.8.07

- **Erweiterung der Funktionen FsiFileInfo() und FsiFileProcess() um SHA-Hashes**

Bei den Befehlen FsiFileInfo() und FsiFileProcess() können nun die neuen Konstanten FsiFileSHA1, FsiFileSHA256, FsiFileSHA384 und FsiFileSHA512 angegeben werden um den SHA-Hash der angegebenen Datei zu berechnen.

- **Protokollierung für Dialog-Befehle**

Mit der Option DbgFrameOpenOn können Aufrufe von Dialog-Befehlen im Debugger protokolliert werden. Durch Aktivierung der Funktion lässt sich schnell erkennen welche Dialoge prozedural geöffnet werden und wo sich die Aufrufe im Quelltext befinden. Mit DbgFrameOpenOff lässt sich die Protokollierung wieder abschalten.

- **Drag & Drop von Dateien unterbinden**

In das WebNavigator-Objekt können Dateien standardmäßig per Drag & Drop gezogen und deren Inhalt angezeigt werden. Mit der Eigenschaft DropFiles kann dies unterbunden werden.

- **Anzeige des Datenbank-Alias in der Titelleiste**

In der Titelleiste im Designer geöffneter Frame- und PrintForm-Objekte kann jetzt optional der Datenbank-Alias angezeigt werden. Dies ist hilfreich, wenn mehr als eine Designer-Instanz läuft und nicht der integrierte Modus aktiviert ist. Dadurch lassen sich die Design-Frames besser zur Datenbank zuordnen. Die Option wird über das Häkchen "Anzeige des Datenbankalias im Fenstertitel" in den Einstellungen aktiviert. Im integrierten Modus wird die Option nicht berücksichtigt.

- **Asymmetrische Verschlüsselung**

MemEncrypt() und MemDecrypt() wurden um Optionen erweitert, mit denen Daten asymmetrisch mit RSA ver- bzw. entschlüsselt werden können. Ein

asymmetrisches Schlüsselpaar kann mit MemGenKeyPair() erzeugt werden.

- **PrintDevice Eigenschaft Orientation**

Beim PrintDevice kann die Eigenschaft Orientation nun auch gesetzt werden. Diese wird beim Befehl WinDocPrint() berücksichtigt. Beim Drucken über PrtJobClose() wird die Seitenorientierung nach wie vor vom PrintDoc übernommen.

- **Standardisierte Komprimierung von Dateien und Memory-Objekten**

Mit den neuen Befehlen MemCompress(), MemUncompress(), FsiFileCompress() und FsiFileUncompress() kann der Inhalt von externen Dateien und Memory-Objekten komprimiert bzw. dekomprimiert werden. Von den Befehlen werden die Formate DEFLATE, GZIP und ZLIB unterstützt.

- **Anhangdateien aus PDF-Dokumenten exportieren**

Mit dem neuen Befehl PdfAttachInfoGet() können Informationen von Anhängen eines PDF-Dokumentes ausgelesen und mit den Befehlen PdfAttachExportFile() und PdfAttachExportMem() die Anhänge exportiert werden. Mit der neuen Eigenschaft PdfAttachCount kann zudem die Anzahl der Anhänge des PDFs ermittelt werden.

- **Erweiterung der Proxy-Authentifizierung**

Neben der Basic-Authentifizierung wird zusätzlich das Verfahren NTLM (NT LAN Manager) unterstützt. Über die neue Eigenschaft spProxyAuthType wird das zu verwendende Authentifizierungsverfahren (HttpProxyAuthTypeBasic / HttpProxyAuthTypeNTLM) festgelegt.

- **Job-Verarbeitung**

Der Befehl JobControl() wurde erweitert. Mit der neuen Option JobEventReceiver kann ein Frame-Deskriptor angegeben werden, der für die Verarbeitung des Ereignisses EvtJob verwendet wird.

- **Erweiterungen der Datensatzverwaltung**

Der Import von Datensätzen wurde erweitert, so dass nun festgelegt werden kann, wie mit importierten Datensätzen verfahren werden soll. Beim Export wird nun der aktuell eingestellte Filter berücksichtigt, so dass auch nur die angezeigten Datensätze exportiert werden. Weiterhin wurde die Datensatzverwaltung für die Verwendung von Tabellen mit 64-Bit Datensatz-IDs angepasst.

- **Erweiterung von Modern Theme Style**

Der Modern Theme Style wurde um weitere Objekte und Eigenschaften erweitert. Mit Hilfe der Eigenschaft ThemeOptFillShade können Farbverläufe dargestellt werden.

CONZEPT 16-Release 5.8.06 - 14. Dezember 2016



Die Clients der Version 5.8.06 benötigen den Server der Version 5.8.06

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten

- Server
- Client
- Programmierschnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.8.06

- **Windows Server 2016**

Ab dieser Version wird das Betriebssystem Microsoft Windows Server 2016 unterstützt.

- **Aktualisierung der OpenSSL-Bibliothek**

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.2j aktualisiert.

CONZEPT 16-Server - Version 5.8.06

- **Lizenzen**



Unabhängig vom eingesetzten Betriebssystem muss zum Betrieb der 5.8.06 die Lizenzdatei aktualisiert werden. Dies erfordert einen aktiven Softwarepflegevertrag.

CONZEPT 16-Client - Version 5.8.06

- **Neue Eigenschaften zum Ermitteln der logischen Prozessoren und des verfügbaren Arbeitsspeichers**

Das Sys-Objekt wurde um die Eigenschaften LogicalProcessors und PhysicalMemoryMB erweitert, mit denen die Anzahl der logischen Prozessoren und die Größe des zur Verfügung stehenden physikalischen Arbeitsspeichers des Computers ermittelt werden können.

- **Neue Konstanten bei DbaInfo() für Datenträgerinformationen**

Mit den neuen Konstanten DbaAreaVolumeSizeMB und DbaAreaVolumeFreeMB bei DbaInfo() können die Größe und der freie Platz auf dem Datenträger in MB ermittelt werden, auf dem die Datenbank liegt.

- **Erweiterung der Rechtschreibprüfung im CtxDocEdit**

Für die Rechtschreibprüfung im CtxDocEdit-Objekt wurden die Eigenschaften SpellSplitWords und SpellIgnoreAllCapsWords hinzugefügt, mit denen zusammengesetzte Wörter in ihre Wortbestandteile aufgetrennt und Wörter, die nur aus Großbuchstaben bestehen, ignoriert werden können.

- **Dynamische Erstellung von Toolbar-Buttons und Statusbar-Buttons**

Mittels WinCreate() können Toolbar-Button- und Statusbar-Button-Objekte dynamisch erstellt und zu einer existierenden Toolbar bzw. Statusbar hinzugefügt werden. Die Objekte können mit WinRemove() / WinDestroy() wieder entfernt / gelöscht werden. Der Befehl WinDestroy(Obj, true) kann verwendet werden, um alle dynamisch erstellten Toolbar-Buttons bzw. Statusbar-Buttons aus der Toolbar bzw. der Statusbar zu löschen.

- **TLS-Verschlüsselung nachträglich für eine Socket-Verbindung initiieren**

Mit dem Befehl SckStartTls() kann für eine bestehende Socket-Verbindung eine TLS-Verschlüsselung initiiert werden.

- **Verwendung von Dateien mit Unicode-Zeichen im Namen**

Bei den Befehlen FsiOpen(), FsiDirOpen() und FsiAttributes() kann jeweils eine der Optionen FsiNameC16 und FsiNameUtf8 angegeben werden. Diese Option spezifiziert den Zeichensatz, in dem der Dateiname angegeben ist. Standardmäßig wird die Option FsiNameC16. Enthält der Dateiname UTF-8-Zeichen, muss die Option FsiNameUtf8 angegeben werden, um die Datei zu verarbeiten. Wurde ein Verzeichnis mit FsiDirOpen(<Dateiname>, FsiNameUtf8) geöffnet, werden die Dateinamen durch FsiDirRead() auch im UTF-8-Zeichensatz ermittelt.



Die Optionen beziehen sich ausschließlich auf den Zeichensatz, der für den Verzeichnis- bzw. Dateinamen verwendet wird und nicht auf den Zeichensatz des Dateiinhaltes. Für die Verarbeitung von Daten eines bestimmten Zeichensatzes existieren entsprechende Befehle (FsiReadMem(), FsiWriteMem(), FsiOpen() mit FsiANSI bzw. FsiPure).

- **Erweiterung der PDF-Schnittstelle**

Die PDF-Schnittstelle wurde um die Möglichkeit erweitert, Dateianhänge in PDF-Dokumente einzufügen. Hierzu existieren zwei neue Befehle PdfAttachFile() und PdfAttachMem(). Des Weiteren besteht nun die Möglichkeit ZUGFeRD-konforme PDF-Dokumente zu generieren. Hierfür wurde der Befehl PdfClose() um neue Optionen erweitert. Ein ausführliches Beispiel hierzu befindet sich in der CodeLibrary.

- **Einführung des Modern Theme Style**

Der Modern Theme Style ermöglicht die Anpassung der graphischen Benutzeroberfläche an das Erscheinungsbild von Modern UI. Zu diesem Zweck wurde das Theme-Objekt mit zahlreichen Eigenschaften hinzugefügt, über das Themes generiert werden können. Dies ist sowohl prozedural (siehe WinThemeOpen()) als auch mit dem Theme-Editor möglich. Mit der Eigenschaft ThemeName kann ein definierte Theme bei den Oberflächenobjekten ausgewählt werden. Zudem wurde das Objekt ToolbarMenu hinzugefügt, um auch Menüs im Modern UI-Stil darzustellen.

- **Protokollierungsoptionen des Anwendungsprotokolls**

Über den Eintrag AppLogOptions in der Konfigurationsdatei des Standard- bzw. Advanced-Clients kann eingestellt werden, ob zusätzliche Debug-Informationen im Anwendungsprotokoll protokolliert werden sollen. Folgende Werte sind möglich:

- ◆ 0 - Keine zusätzliche Protokollierung.
- ◆ 2 - Protokolliert den Ablauf, wenn der Designer terminiert wird.

Die Protokollierung kann auch über die Kommandozeilenoption C16AppLogOptions beim Starten des Standard- bzw. Advanced-Clients oder die Umgebungsvariable C16AppLogOptions gesetzt werden.

- **ReadOnly bei RecListPopup- und StoListPopup-Objekten**

Kontakt

Die Objekte RecListPopup und StoListPopup verfügen über die Eigenschaft ReadOnly.

- **Neue Eigenschaft TabSelect**

Ist die neue Eigenschaft TabSelect gesetzt, wird der aktuell im DataListPopup-, RecListPopup- oder StoListPopup-Objekt ausgewählte Eintrag in das Eingabefeld übernommen, wenn das Objekt mit bzw. + verlassen wird. Standardmäßig ist die Eigenschaft auf true gesetzt.

- **Hervorhebung von Container-Objekten im Designer**

Der neue Menüpunkt Objektumrandung hervorheben im Menü Anzeige des Designers (bzw. die Tastenkombination + +) hebt die Umrandung von Container-Objekten (GroupBox, NotebookPage, Scrollbox) mit einem gestrichelten Rahmen hervor. Hierdurch können die Objektbegrenzungen beim Positionieren, Vergrößern, etc. besser erkannt werden, insbesondere bei kontrastschwachen Windows-Themes. Die Funktion wirkt sich auf alle im Designer geöffneten Frames und PrintForms aus.

Die Farbe der Umrandung kann in den Einstellungen im Bereich Designer über die Auswahl "Hervorhebungsfarbe" angepasst werden.

- **Objekte nach Typ iterieren**

Mit WinInfo() bzw. PrtInfo() lassen sich Oberflächenobjekte unter Angabe des Objekttyps iterieren. Der übergebene Objekttyp stellt dabei nicht immer den exakten Objekttyp dar, sondern eine Objektgruppe.

Die Konstante WinTypeExact kann bei WinInfo() zusammen mit dem Objekttyp angegeben werden, um Objekte anhand des exakten Objekttyps statt der Objektgruppe zu ermitteln.

- **Layer-Modus für Fensterobjekte**

Werden graphische Objekte positioniert oder Eigenschaften gesetzt, die die Darstellung (z.B. Farbe und Schriftart) verändern, kann dies zu ungewünschten Flackereffekten in der Anwendung führen. Durch die Eigenschaft AutoUpdate lassen sich die Effekte in einigen Fällen zwar reduzieren, bei objektübergreifenden Anpassungen vieler Objekte jedoch meist nicht. Für die Optimierung der Ausgabe muss zudem der Programmcode geändert werden.

Um solchen Fällen entgegenzuwirken, wurde der Befehl WinLayer() implementiert. Dieser "friert" die Ausgabe eines vorgegebenen Frame-Objektes für eine bestimmte Zeitspanne ein. Änderungen an untergeordneten graphischen Objekten, während dieser Zeitspanne werden zwar durchgeführt, jedoch erst sichtbar nachdem die Zeitspanne abgelaufen ist, bzw. der Layer-Modus abgeschaltet wurde.

- **Erweiterung von DbgDump()**

Der Befehl DbgDump() wurde erweitert, so dass nun zusätzlich die Zeilennummer der jeweiligen Aufrufe protokolliert wird.

- **Erweiterung von FldDef() und FldDefByName()**

Kontakt

Bei den Befehlen FldDef() und FldDefByName() kann NULL kann die Konstante NULL als Feldwert angegeben werden.

- **EvtKeyItem für Steuerungstasten**

Für Steuerungstasten wurde das Ereignis EvtKeyItem bisher nicht ausgelöst. Durch Setzen der Option WinAppExtFlowKeySupport bei der neuen Eigenschaft FlagsExt des Application-Objektes ist dies nun möglich.

- **Neue Eigenschaften zur Darstellung von Spaltenüberschriften**

Die Darstellung der Objekte DataList, RecList, RecView und StoList sowie deren Popup-Varianten kann mit den neuen Eigenschaften HdrVerticalText, HdrHeight und HdrWordBreak modifiziert werden. Durch sie kann der Text der Spaltenüberschriften vertikal oder mehrzeilig dargestellt und die Höhe der Spaltenüberschriften definiert werden.

- **Benutzerdefinierte Kacheln in gesperrter Darstellung**

Über die Eigenschaft GrayingModeUser kann die Hervorhebung zwischen normaler und gesperrter Darstellung verbessert werden.

- **Überdeckung untergeordneter Objekte im Groupbox-Objekt**

Über die Option WinAppExtTransparentBkg der Eigenschaft wpFlagsExt des App-Objektes kann die Überdeckung des Rahmens von GroupBox-Objekten definiert werden.

CONZEPT 16-Programmierschnittstelle - Version 5.8.06

- **Erweiterungen für Prozeduren**

Die Programmierschnittstelle wurde um die Möglichkeit erweitert, Prozeduren zu verarbeiten. Analog zur Funktion C16_TextOpen steht nun auch die Funktion C16_ProcOpen zur Verfügung. Diese liefert ein Handle auf die angegebene Prozedur. Die Prozedurverarbeitung erfolgt anschließend über die bereits vorhandenen Text-Funktionen.

Zusätzlich können Prozeduren mit der Funktion C16_ProcCompile übersetzt werden.

CONZEPT 16-Release 5.8.05 - 9. September 2016



Die Clients der Version 5.8.05 benötigen den Server der Version 5.8.05

Dies ist eine interne Version.

CONZEPT 16-Release 5.8.04 - 10. Mai 2016



Die Clients der Version 5.8.04 benötigen den Server der Version 5.8.04

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Installationsroutine
- Client

- SOA-Service
- Laufwerkstreiber
- Programmierschnittstelle
- ODBC-Schnittstelle
- Designer Plugin-Schnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.8.04

- **Performance beim Export von binären Objekten optimiert**

Der Export von binären Objekten durch den Datenbankserver wurde optimiert, so dass ein Export von großen oder vielen binären Objekten schneller durchgeführt wird.

- **Aktualisierung der OpenSSL-Bibliothek**

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.2h aktualisiert.

 Bei der Verbindungsoption SckTlsLow wurden durch das Update alle SSLv2-Chiffren sowie alle Chiffren, deren Verschlüsselung unter 128 Bit liegt, entfernt. Dies betrifft auch die Export-Chiffren.

CONZEPT 16-Installationsroutine - Version 5.8.04

- **Minimal vorausgesetzte Version für Upgradeinstallation angepasst**

Der Upgrade des Programmstandes auf Version 5.8.04 wird von der Installationsroutine nur noch unterstützt, wenn die installierte Version nicht kleiner als Version 5.6.01 ist. Bei einem älteren Programmstand wird die Installationsroutine mit einer entsprechenden Meldung abgebrochen.

- **Fehlerbehandlung der Installationsroutine geändert**

 Durch eine Aktualisierung der Installationsroutine, werden nur noch die Fehlerwerte 0 bis 5 als Errorlevel zurückgegeben. Als Kommandozeilenargument resultfile kann eine Datei angegeben werden, in der auch die Fehlerwerte 244 bis 255 abgelegt werden, falls ein Fehler auftritt. Die Resultatdatei wird prioritär behandelt.

CONZEPT 16-Client - Version 5.8.04

- **Ausgabe mehrerer Formate bei PrtJobClose()**

Bei PrtJobClose() können die Konstanten PrtJobPreview, PrtJobPreviewValidate, PrtJobPrint, PrtJobPdf, PrtJobEmf, PrtJobTif und PrtJobXml kombiniert werden, um mehrere Formate nacheinander auszugeben.

- **Focus-Ereignisse bei Calendar-Objekt**

Das Calendar-Objekt wurde um die Ereignisse EvtFocusInit, EvtFocusTerm und EvtFocusCancel erweitert.

- **Alle Debugger-Verbindungen beenden**

Über den Menüpunkt Alle Sitzungen beenden werden alle offenen Debugger-Sitzungsfenster geschlossen. Wenn die Abfragen Abfrage beim Beenden der Sitzung bzw. Abfrage beim Speichern in den Einstellungen aktiviert sind, findet eine Abfrage des Benutzers statt. Durch deaktivieren der Einstellung kann dies unterbunden werden.

• **Startleiste für Prozeduren**

Im Editor wurde eine Toolbar hinzugefügt, über die beliebige Prozeduren ausgeführt werden können. Die Leiste kann über die Menüpunkte Prozedur/Ausführen andere... bzw. Startleiste aktiviert werden. Zusätzlich kann die Tastenkombination  +  +  verwendet werden um die Startleiste anzuzeigen.



Bisher hatte der Taschenrechner diese Tastenkombination. Dieser kann jedoch nicht mehr über eine Tastenkombination erreicht werden. Der Menüpunkt existiert aber natürlich noch.

• **Neue Gruppierungseigenschaften**

Mit den neuen Eigenschaften AlignGroupingBottomOrder, AlignGroupingLeftOrder, AlignGroupingRightOrder und AlignGroupingTopOrder kann die Darstellungsreihenfolge der automatisch angeordneten Objekte verändert werden.

• **Neue Ausprägungen bei Grouping**

Mit den Ausprägungen WinGroupingCenterTileVert und WinGroupingCenterTileHorz können Objekte zentriert in ihrem Elternobjekt positioniert werden. Die Größe der Unterobjekte wird durch die Eigenschaften AlignHeight und AlignWidth bestimmt.

• **Lokales Debugging**

Bei Verwendung des aktuellen Clients zusammen mit dem aktuellen Debugger ist nun auch ein Debugging aller Ereignisse möglich, selbst wenn Client und Debugger auf derselben Maschine laufen. Bisher wurden in diesem Fall Breakpoints und DbgStop()-Anweisungen für bestimmte Ereignisse ignoriert. Das Status-Icon in der Titelleiste der Session ist blau, wenn das lokale Debugging aktiv ist. Betroffen sind folgende Ereignisse:

- ◆ EvtChangedActive
- ◆ EvtDragInit
- ◆ EvtDropEnter
- ◆ EvtDropLeave
- ◆ EvtFocusInit
- ◆ EvtFocusTerm
- ◆ EvtHelpTip
- ◆ EvtKeyItem
- ◆ EvtLstDataInit
- ◆ EvtLstEditActivate
- ◆ EvtLstEditCommit
- ◆ EvtLstEditEndGroup
- ◆ EvtLstEditEndItem

- ◆ EvtLstEditFinished
- ◆ EvtLstEditStartGroup
- ◆ EvtLstEditStartItem
- ◆ EvtLstGroupArrange
- ◆ EvtLstGroupInit
- ◆ EvtLstRecControl
- ◆ EvtLstViewInit
- ◆ EvtMdiActivate
- ◆ EvtMouseMove
- ◆ EvtPosChanged
- ◆ EvtTimer

- **Optimierte Darstellung von Objekten mit Icon**

Wird bei WinDialog() bzw. WinDialogRun() die Option WinDialogCreateHidden angegeben, werden alle Symbole die in der Eigenschaft Icon angegeben wurden mit Sichtbarwerden des Dialoges ohne Verzögerung sichtbar.

- **Protokollierung von Abstürzen im Anwendungsprotokoll**

Unter Windows XP bzw. der Server-Variante Windows 2003 werden keine Anwendungsfehler (Abstürze) mehr geloggt.

- **WinInfo(WinFocusKey) mit **

Über WinInfo(...., WinFocusKey) kann nun auch ermittelt werden, ob die -Taste zum Verlassen des Objektes führte.

- **Neue Konstanten für Laufzeitfehler**

Die Konstanten ErrFrameDiffers und ErrFrameNotFound wurden hinzugefügt. Sie werden ausgelöst, wenn ein Frame, der durch ein with-Konstrukt referenziert wird, nach dem Übersetzen der Prozedur verändert wurde, oder nicht vorhanden ist.

- **Neue Befehle zur sequenziellen XML-Verarbeitung**

Mit den neuen Befehlen XmlOpenReader() und XmlOpenWriter() kann eine XML-Datei zum sequenziellen Lesen bzw. Schreiben geöffnet werden. Mit XmlRead() wird ein Knoten gelesen. Anschließend können mit XmlGetValueInt() und XmlGetValueAlpha() die Eigenschaften und Werte des Knotens ermittelt werden. Eine Datei, die zum Schreiben geöffnet ist, kann mit XmlWrite() beschrieben werden. Mit XmlClose() kann die Datei geschlossen werden.

- **PDF-Generierung bei ChartSave()**

Mit der neuen Option ChartFormatPdf bei ChartSave() können nun Chart-Objekte als PDF gespeichert werden.

- **Balkenbeschriftung in Koordinaten-Diagrammen**

Wird bei ChartDataOpen() und ChartDataAdd() die neue Option ChartDataExtra angegeben, kann zusätzlich eine Balkenbeschriftung gesetzt werden. Diese wird angezeigt, wenn die Eigenschaft ChartXYStyleLabel auf ChartXYStyleLabelDataExtra gesetzt wird. Zusätzlich kann das Aussehen der Beschriftungen mit den Eigenschaften ChartXYLabelColData, ChartXYLabelColSum und ChartXYLabelRotData angepasst werden.

- **Y-Achse pro Datenreihe in Koordinaten-Diagrammen**

Mit der neuen Eigenschaft ChartXYAxisY kann eine zusätzliche Y-Achse pro Datenreihe generiert werden. Die Position und das Aussehen der Y-Achsen können über die Eigenschaften ChartXYAxisOffsetY, ChartXYAxisColY, ChartXYAxisTitleY und ChartXYAxisTitleAlignY definiert werden. Zusätzlich kann mit der Eigenschaft ChartXYTitleAlignY auch die Position der Beschriftung für die vertikale Hauptachse gesetzt werden.

- **Erweiterung der Eigenschaft Name**

Die Eigenschaft Name für Dialoge, Menüs und PrintForms wurde auf 60 Zeichen erweitert.



Dialoge, Menüs und PrintForms mit mehr als 40 Zeichen können in älteren Clients nicht geladen werden!

- **Das CtxDocEdit wurde aktualisiert**

Das CtxDocEdit wurde auf Version 17 Service Pack 3 aktualisiert.

- **Neue Konstanten für FileInfoQ und FileInfoByNameQ**

Mit den neuen Konstanten FileEncrypted, FileDialog64, FileSeqInsert und FileFldBuf können die Einstellungen chiffrierte Speicherung, 64-Bit Datensatz-IDs, sequentielles Einfügen und eigene Feldpuffer abgefragt werden.

- **Neue Optionen in ImageTile und HdrImageTile**

Es wurden die folgenden Optionen zu ImageTile und HdrImageTile hinzugefügt:

- ◆ WinImgArrowLeft
- ◆ WinImgArrowRight
- ◆ WinImgArrowUp
- ◆ WinImgArrowDown
- ◆ WinImgSearch
- ◆ WinImgSearchNext
- ◆ WinImgSearchPrev
- ◆ WinImgReplace
- ◆ WinImgReplaceAll
- ◆ WinImgKeyRepair

- **Neue Option WinRtfEditAddPictAsMetafile der Eigenschaft RtfEditFlags**

Ab Windows 8.1 werden Bilder, die über die Zwischenablage in ein RtfEdit-Objekt eingefügt werden, als Bilder mit dem entsprechenden nativen Format eingefügt (z. B. png oder jpeg). Diese Dateien können unter älteren Betriebssystemen (z. B. Windows 7) nicht dargestellt werden. Wird die Option WinRtfEditAddPictAsMetafile in der Eigenschaft RtfEditFlags des RtfEdit-Objektes angegeben, werden die Bilder aus der Zwischenablage in einem kompatiblen Format eingefügt. Um Bilder per Drag & Drop im kompatiblen Format einzufügen existiert weiterhin die Option WinRtfEditAddFileAsPicture.

- **Anzeige des Aliasnamen der Datenbank in der Titelzeile des Designers**

Im Display "Datenbank" des Designers wird nun der Datenbank-Alias angezeigt.
Im Titel des Designers steht weiterhin der (Datei-)Name der Datenbank.

- **Benutzerdefinierte Tiles in verschiedenen Größen**

Oberflächenobjekte, die über die Eigenschaft ImageTileUser verfügen, mit Ausnahme von MenuItem, berücksichtigen nun auch die Eigenschaft TileSize, die entweder direkt beim Objekt oder beim App-Objekt definiert ist. In Abhängigkeit vom Wert der Eigenschaft TileSize wird folgende Kachelgrafik referenziert:

- ♦ **TrueColor-Modus (24, 32 BPP)**

- ◊ BtnTile.tcm.<TileSize>
- ◊ BtnTile.tcm
- ◊ BtnTile.hcm.<TileSize>
- ◊ BtnTile.hcm
- ◊ BtnTile.<TileSize>
- ◊ BtnTile

- ♦ **HiColor-Modus (16 BPP)**

- ◊ BtnTile.hcm.<TileSize>
- ◊ BtnTile.hcm
- ◊ BtnTile.<TileSize>
- ◊ BtnTile

- ♦ **Sonstige Modi**

- ◊ BtnTile.<TileSize>
- ◊ BtnTile



Die Grafiken mit der Erweiterung .<TileSize> werden nur verwendet, wenn die Eigenschaft TileTheme auf WinTileThemeEnhanced gesetzt ist.

- **Erweiterung der Eigenschaft TileTheme**

Die Eigenschaft wurde um die neue Konstante WinTileThemeModern erweitert. Über die neue Konstante werden die CONZEPT 16-Standard-Icons in einer an Windows 8/8.1/10 angelehnten (flachen) Darstellung gezeichnet.

- **Eigenschaft StyleTheme bei Systemdialogen**

Die Systemdialoge WinComFileOpen und WinComFileSave wurden um die Eigenschaft StyleTheme erweitert. Ist die Eigenschaft auf WinStyleThemeSystem gesetzt, wird die Baumsicht auf der linken Seite des Dialogs angezeigt. Der Wert der Eigenschaft wird bei WinOpen() vom App-Objekt übernommen.

- **HttpGetData() ignoriert fehlende Zeilenumbrüche**

Werden Daten in Blöcken (Transfer-Encoding: chunked) geschickt, werden fehlende Zeilenumbrüche zwischen den Daten und der nächsten Blocklänge ignoriert.

- **Neuer Befehl HdEnum()**

Mit HdEnum() können alle Deskriptoren enumeriert werden. Optional kann ein Deskriptor-Typ angegeben werden um nur die Deskriptoren des Typs

durchzugehen.

- **Verschlüsselte Verbindungen über einen HTTPS-Proxy bei HttpClose()**

Mit der neuen Konstante HttpUseWebProxyTLS bei HttpClose() kann mit dem HTTP-Objekt eine verschlüsselte Verbindung über einen Proxy-Server aufgebaut werden. Erfordert der Proxy-Server eine Basic-Authentifizierung muss zusätzlich die Eigenschaft ProxyAuthorization gesetzt werden.

- **Eigenschaft Version beim PrintJob**

Beim Öffnen eines Druckjobs über PrtJobOpen() mit der Option PrtJobOpenWrite wird nun auch das Release der CONZEPT 16-Version (c16_objw.dll) gespeichert. Beim Lesen des Druckjobs mit der Option PrtJobOpenRead kann die Version über die Eigenschaft Version des Druckjobs ermittelt werden. Sie enthält den Wert 0, wenn der Druckjob mit einem Client vor Release 5.8.04 erstellt wurde. Die Eigenschaft kann nur gelesen werden.



Wird ein Druckjob mit dem aktuellen Client erstellt, können Clients älter als Version 5.5.04 den Druckjob nicht öffnen. Ältere Druckjobs können weiterhin mit der neuen Version geöffnet werden.

- **Zeilenummern im Editor**

Im Menü Bearbeiten des Editors können über den Menüpunkt Zeilenummern anzeigen oder die Tastenkombination + Zeilenummern am linken Rand der geöffneten Prozeduren bzw. Texte angezeigt werden. Die Option ist auch in den Einstellungen konfigurierbar.

- **E-Mails speichern und gespeicherte E-Mails versenden**

Mit der neuen Option MailCreateEML bei MailOpen() kann eine E-Mail statt zu versenden für den späteren Versand oder die Archivierung gespeichert werden. Bei MailData() kann eine bestehende EML-Datei mit der Option MailFileEML als Inhalt definiert werden.

CONZEPT 16-SOA-Service - Version 5.8.04

- **Fehlerbehebung in der Laufwerkstreiber-Referenzimplementation**



Es wurde ein Fehler bei der Zeichensatzwandlung im Laufwerkstreiber behoben. Bei Verwendung des neuen SOA-Drive muss das Makro mC16toUTF8 in der Referenzimplementation angepasst werden. Sofern also die Referenzimplementation verwendet wird, bitte die Prozedur DrvVar aus der CodeLibrary neu importieren.

CONZEPT 16-Laufwerkstreiber - Version 5.8.04

- **Aktualisierte Version**

Der Laufwerkstreiber wurde auf die Version 6.0.173 aktualisiert. Die neue Version enthält Fehlerkorrekturen, die Windows 10 betreffen, sowie Optimierungen des treiberinternen Datei-Caches.

- **Leeren des Cache-Verzeichnisses beim Start**

Kontakt

Der Inhalt des Drive-Cache-Verzeichnisses (Unterordner DRIVE_CACHE, siehe [DrvInit](#)) wird beim Start vollständig geleert.

-  Sollten dort Dateien (z. B. anderer Programme) enthalten sein, sind diese nach dem Start des Drive-SOA-Task nicht mehr vorhanden. Es ist daher wichtig, dass dieses Verzeichnis exklusiv dem Laufwerkstreiber zur Verfügung steht.

CONZEPT 16-Programmierschnittstelle - Version 5.8.04

• Neue Befehle für binäre Objekte

Die folgenden Befehle wurden der [Programmierschnittstelle](#) zur Verwaltung von binären Verzeichnissen und Objekten hinzugefügt:

- ◆ [C16_BinClose](#)
- ◆ [C16_BinCopy](#)
- ◆ [C16_BinDelete](#)
- ◆ [C16_BinDirRead](#)
- ◆ [C16_BinInfo](#)
- ◆ [C16_BinMove](#)
- ◆ [C16_BinOpen](#)
- ◆ [C16_BinReadToFile](#)
- ◆ [C16_BinReadToMem](#)
- ◆ [C16_BinRename](#)
- ◆ [C16_BinUpdate](#)
- ◆ [C16_BinWriteFromFile](#)
- ◆ [C16_BinWriteFromMem](#)

• Ältere Datenbanken mit [C16_OpenArea](#) öffnen

-  Um Datenbanken von einem Stand kleiner als 5.8 zu öffnen, muss bei [C16_OpenArea](#) im Argument aProcCacheKB die neue Konstante _OpenMode5x angegeben werden.

CONZEPT 16-ODBC-Schnittstelle - Version 5.8.04

• Neue Version des ODBC-Treibers

Der ODBC-Treiber basiert nun auf Basis der SimbaEngina 9.5 und enthält Fehlerkorrekturen und Erweiterungen (z.B. CASE-Statement). Der neue Treiber ist ab Windows 7 SP1 bzw. Windows Server 2008 R2 lauffähig. Für ältere Betriebssysteme kann der ODBC-Treiber in einer [Legacy](#)-Version installiert werden.

-  Nach Installation des ODBC-Legacy-Treibers führt eine Aktualisierung des Betriebssystems (z. B. Installation des Service Pack 1 unter Windows 7) **nicht** dazu, dass die erneute Durchführung der [Installationsroutine](#) (Aktion [Aktualisieren](#)) den neuen ODBC-Treiber installiert. Um den aktualisierten ODBC-Treiber zu installieren, muss dieser zunächst deinstalliert und anschliessend wieder installiert werden (Aktion [Ändern](#)).



Die Verwendung des ODBC-Legacy-Treibers wird nicht empfohlen, da für diese Komponente keine Fehlerkorrekturen mehr durchgeführt werden.

CONZEPT 16 - Designer Plugin-Schnittstelle - Version 5.8.04

• Neue Kommandos der Plugin-Schnittstelle

Die Plugin-Schnittstelle wurde um die folgenden Kommandos erweitert:

- ◆ Designer.Storage.LastLockingInfo
- ◆ Designer.Storage.<ObjectName>.ExportFile
- ◆ Designer.Storage.<ObjectName>.ExportStream
- ◆ Designer.Storage.ImportFile
- ◆ Designer.Storage.ImportStream
- ◆ Designer.LastXmlError

Zusätzlich wurde das Kommando Designer.Forms.Selection.GetPreview um die optionalen Argumente Width und Height erweitert.



Neue Kommandos setzen den Designer der Version 5.8.04 voraus.

CONZEPT 16-Release 5.8.03 - 3. Dezember 2015



Die Clients der Version 5.8.03 benötigen den Server der Version 5.8.03

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Client
- Laufwerkstreiber

CONZEPT 16 - Alle Komponenten - Version 5.8.03

• Neue Fehlerkonstante ErrSelValueSet

Die neue Fehlerkonstante ErrSelValueSet wird von SelRun() zurückgegeben, wenn zwei Selektionen über SelUnion, SelInter oder SelMinus kombiniert werden und eine der Selektionen eine Wertmenge enthält.

• Neues optionales Argument bei MailClose()

Bei MailClose() kann optional eine alpha-Variable angegeben werden, die bei Fehlern die letzte Antwort des Mail-Servers enthält. Der zurückgegebene Text ist leer, wenn der Fehler nicht auf eine Antwort des Mail-Servers zurückgeht (z. B. Socket-Fehler).

• Neue Option StrLower1252 bei StrCnv()

Der Befehl StrCnv() konvertiert Zeichenketten. Mit dem Parameter StrLower wurden nur die Zeichen A-Z, sowie die Umlaute Ä, Ö und Ü in Kleinbuchstaben gewandelt. Mit dem neuen Parameter StrLower1252 werden alle Großbuchstaben der Windows-Codepage 1252 in Kleinbuchstaben gewandelt.

CONZEPT 16-Client - Version 5.8.03

- **Neue Eigenschaft CalculatorSize**

Über die neue Eigenschaft CalculatorSize des FloatEdit bzw. DecimalEdit kann die Größe der Taschenrechner-Eingabehilfe beeinflusst werden, um eine bessere Bedienung über Touch-Eingabe zu erreichen.

- **Neues Ereignis EvtDropOver**

Das Ereignis EvtDropOver wird während einer Drag & Drop-Operation ausgelöst, wenn der Mauszeiger innerhalb eines Objektes bewegt oder eine der Hilfstasten (, oder) betätigt wird.

- **Fehler beim Öffnen des Anwendungsprotokolls**

Tritt beim Öffnen des Anwendungsprotokolls ein Fehler auf, wird in der Fehlermeldung auch der Fehlerwert angezeigt.

CONZEPT 16-Laufwerkstreiber - Version 5.8.03

- **Aktualisierte Version**

Der Laufwerkstreiber wurde auf die neuste Version aktualisiert. Die neue Version enthält Fehlerkorrekturen.

CONZEPT 16-Release 5.8.02 - 14. Oktober 2015



Die Clients der Version 5.8.02 benötigen den Server der Version 5.8.02

Diese Version enthält nur Fehlerkorrekturen.

CONZEPT 16-Release 5.8.01 - 7. Oktober 2015



Die Clients der Version 5.8.01 benötigen den Server der Version 5.8.01

An folgenden Komponenten wurden Änderungen durchgeführt:

- Alle Komponenten
- Client
- SOA-Service
- Laufwerkstreiber
- Programmierschnittstelle
- Designer Plugin-Schnittstelle

CONZEPT 16 - Alle Komponenten - Version 5.8.01

- **Windows 10**

Ab dieser Version wird das Betriebssystem Microsoft Windows 10 unterstützt.

- **Umstellung der Datensatz-ID**



Die Reihenfolge der Datensatz-IDs der Datensatzpuffer wurde verändert. Ab der Version 5.8 werden die Datensatz-IDs

Kontakt

fortlaufend, beginnend ab 1 generiert. Wird die Datensatz-ID in Datensätzen als Referenz gespeichert, sind Anpassungen notwendig. Die Thematik wird detaillierter in den Blog-Artikeln Version 5.8 / Teil 1: Rec-ID unlimited und Version 5.8 / Teil 2: Implizite Typkonvertierung behandelt.

Beim sequentiellen Einfügen wurden die IDs wie folgt vergeben: 0x01000000 (16.777.216), 0x02000000 (33.554.432), ..., 0xFF000000 (-16.777.216), 0x00010000 (65.536), 0x01010000 (16.842.752), 0x02010000 (33.619.968), ..., 0xFFFFFFF (-16.777.217), 0xFFFFFFFF (-1).

Beim gestreuten Einfügen (sequentiellen Einfügen ist deaktiviert) erfolgte die Vergabe wie folgt: 0xFFFFFFFF (-1), 0xFFFFFFF (-2), 0xFFFFFFF (-3), ..., 0x80000001 (MinInt), 0x80000000, 0x7FFFFFFF (MaxInt), ..., 0x00000002 (2), 0x00000001 (1).

• Implizite Typkonvertierung zwischen int und bigint

Bei Zuweisungen, Argumenten und in Ausdrücken werden Variablen, Konstanten oder Felder vom Typ int und bigint nun implizit konvertiert, wenn der Zieltyp (int bzw. bigint) nicht dem Quelltyp entspricht.

• 64-Bit Datensatz-IDs

Die Befehle RecRead() und RecInfo() wurden angepasst, damit dort auch bigint-Werte übergeben bzw. zurückgegeben werden können. Durch die implizite Typkonvertierung ist keine Änderung an den Aufrufen von RecRead() nötig. Die Verwendung von 64-Bit Datensatz-IDs kann im Datenstruktureditor gesetzt werden.



Wird bei RecInfo() die Option RecSetID angegeben, kommt es bei negativen Datensatz-IDs zum Laufzeitfehler ErrValueInvalid.

• ID vom Typ bigint

Die Eigenschaft ID wurde von int auf bigint erweitert. Entsprechend wurden auch die folgenden Befehle angepasst:

- ◆ CteInsertItem
- ◆ CteInsertNode



Bestehende Prozeduren, die mit der Version 5.7 übersetzt wurden und einen der Befehle verwenden, erwarten nach wie vor einen int-Wert, um die Kompatibilität mit der Vorgängerversion zu gewährleisten.

• Aktualisierung der OpenSSL-Bibliothek

Die verwendete OpenSSL-Bibliothek wurde auf die Version 1.0.1p aktualisiert.

• Reguläre Ausdrücke

Die Befehle StrFindRegEx() und TextSearchRegEx() können jetzt neben dem Standard- und Advanced-Client auch im SOA-Task und im Datenbank-Server ausgeführt werden. Unter Windows wurde die Datei c16_icuw.dll durch folgende DLLs ersetzt:

c16_icu_w32.dll

Kontakt

benötigt vom Standard-, Advanced-Client, der Programmierschnittstelle (erweiterte Variante), Druckprozessor sowie vom 32-Bit SOA-Task und dem 32-Bit-Datenbank-Prozess.

c16_icu_w64.dll benötigt vom 64-Bit SOA-Task und dem 64-Bit-Datenbank-Prozess.

- **Neue Fehlerkonstante ErrDbaServerRelease**

Der Fehler ErrDbaServerRelease wird zurückgegeben, wenn prozedural eine Verbindung zu einer Datenbank aufgebaut wird (z. B. DbaConnect()) und die Server-Version kleiner als die Version des zu verbindenden Clients ist.

CONZEPT 16-Client - Version 5.8.01

- **Anzeige des Prime-Counters im Datenstruktureditor**

Im Datenstruktureditor wird zusätzlich zu Datei- und Datensatzanzahl der Prime-Counter der ausgewählten Tabelle angezeigt.

- **Ereignisse mit Übergabe der Datensatz-ID**

Bei Ereignissen, die die Datensatz-ID übergeben bekommen, ist es möglich das Argument aID / aRecID als bigint zu definieren.

Folgende Ereignisse wurden angepasst:

- ◆ EvtHelpTip
- ◆ EvtKeyItem
- ◆ EvtLstDataInit
- ◆ EvtLstEditFinished
- ◆ EvtLstRecControl
- ◆ EvtLstSelect
- ◆ EvtLstSelectRange
- ◆ EvtMenuContext
- ◆ EvtMenuItem



Wird das Argument aID / aRecID als int definiert, wird der Laufzeitfehler ErrValueOverflow ausgelöst, wenn die zu übergebende Datensatz-ID nicht in den Wertebereich von int passt.

- **Neue Eigenschaft DbRecID64 bei RecList- und RecListPopup-Objekten**

Die RecList- und RecListPopup-Objekte wurden um die Eigenschaft DbRecID64 erweitert. Mit Hilfe der Eigenschaft kann eine Datensatz-ID als bigint ermittelt werden. Die Eigenschaft DbRecID existiert weiterhin wie bisher.

- **Befehle mit bigint-Argument für Datensatz-ID statt int-Argument**

Die folgenden Befehle wurden angepasst, damit eine 64-Bit Datensatz-ID angegeben werden kann:

- ◆ WinMsdInsert
- ◆ WinMsdDelete
- ◆ WinMsdRead

Kontakt

Das Argument, in dem die Datensatz-ID angegeben wird, ist nun vom Typ bigint.

-  Bestehende Prozeduren, die mit der Version 5.7 übersetzt wurden und einen der Befehle verwenden, erwarten nach wie vor einen int-Wert, um die Kompatibilität mit der Vorgängerversion zu gewährleisten.

- **Nachrichtenverarbeitung bei WinSleep() einschränken**

Der Befehl WinSleep() wurde um ein optionales Argument erweitert. Mit dem neuen Argument kann angegeben werden, welche Nachrichten verarbeitet werden soll. So ist es z. B. möglich, Tastatur- oder Mausnachrichten (Eingabe-Nachrichten) nicht zu verarbeiten und in der Nachrichten-Warteschlange zu belassen, um sie zu einem späteren Zeitpunkt verarbeiten zu können.

- **Eigenschaften CurrentInt und CryptKey bei Spalten eines RecView**

Bei den Objekten Column und Item des RecView-Objektes können nun die Eigenschaften CurrentInt und CryptKey angegeben werden. Mit CurrentInt kann die Nummer des Bildes im ausgewählten TIFF-Bild oder Icon festgelegt werden. Um verschlüsselte binäre Objekte im RecView anzuzeigen, kann das Passwort in CryptKey definiert werden.

- **Rahmeneigenschaften beim RecView-Objekt**

Das RecView-Objekt wurde um die Eigenschaften BorderType, BorderWidth sowie BorderColFg erweitert. Der Rahmen wird entsprechend der Eigenschaften für jede Spalte und Gruppe gezeichnet.

Durch Setzen der Eigenschaft BorderType auf WinBorderTypeSolidRightBottom bzw. WinBorderTypeDotRightBottom lässt sich ein Raster (vergleichbar zur RecList) zeichnen.

- **Neue Konstanten bei WinInfo()**

Mit den Konstanten WinInteriorHeight und WinInteriorWidth kann bei WinInfo() der Bereich eines AppFrame-Objektes ermittelt werden, in dem MdiFrame-Objekte dargestellt und angeordnet werden können. Dies ist der Client-Bereich des AppFrame-Objektes ohne etwaige Toolbar- und Statusbar-Objekte.

CONZEPT 16-SOA-Service - Version 5.8.01

- **Automatische Neustart nach anormaler Terminierung**

Mit dem neuen Eintrag autorevive in der SOA-Service - Konfigurationsdatei des SOA-Service kann eingestellt werden, ob der entsprechende SOA-Task nach einer anormalen Terminierung automatisch wieder gestartet werden soll (autorevive=Y). Default-Einstellung ist autorevive=N. Das Startintervall beträgt 10 Sekunden. Die Einstellung kann auch im Control-Center definiert werden.

CONZEPT 16-Laufwerkstreiber - Version 5.8.01

- **Aktualisierte Version**

Der Laufwerkstreiber wurde auf die neuste Version aktualisiert. Die neue Version enthält Fehlerkorrekturen.

CONZEPT 16-Programmierschnittstelle - Version 5.8.01

- **Neue Funktionen für Verarbeitung von 64-Bit Datensatz-IDs**

Die Programmierschnittstelle wurde für die Verarbeitung von 64-Bit Datensatz-IDs angepasst. Hierzu wurden die Funktionen C16_RecInfo64 und C16_RecRead64 hinzugefügt.



Die Funktionen C16_RecInfo und C16_RecRead sind noch aus Kompatibilitätsgründen vorhanden und sollten in Zukunft nicht mehr verwendet werden.

CONZEPT 16 - Designer Plugin-Schnittstelle - Version 5.8.01

- **Plugin-Schnittstelle Version 1.1**

Die Plugin-Schnittstelle liegt nun in der Version 1.1 vor. Die API ist abwärtskompatibel mit der Vorgängerversion 1.0. Neue Kommandos setzen allerdings den Designer des CONZEPT 16 Release 5.8 voraus.

API - Neue Funktionen

- ◆ Plugin.Core:NextSerial
- ◆ Plugin.Converter:AddArgMem
- ◆ Plugin.Converter:AddExecResult
- ◆ Plugin.Converter:GetArgStrDecB64
- ◆ Plugin.Converter:GetArgStrLen
- ◆ Plugin.Converter:GetArgMem

API - Erweiterte Funktionen

- ◆ Plugin.Core:ReceiveLine
- ◆ Plugin.Core:ApiInfo
- ◆ Plugin.Converter:GetArgStr
- ◆ Plugin.Converter:GetArgInt
- ◆ Plugin.Converter:GetArgLogic
- ◆ Plugin.Converter:ReplyAuth

Neue Kommandos

- ◆ Designer.GetInfo
- ◆ Designer.Forms.Export
- ◆ Designer.Forms.Import
- ◆ Designer.Forms.GetCount
- ◆ Designer.Forms.GetObject
- ◆ Designer.Forms.Selection.Get
- ◆ Designer.Forms.Selection.New

Kontakt

- ◆ [Designer.Forms.Selection.GetCount](#)
- ◆ [Designer.Forms.Selection.GetObject](#)
- ◆ [Designer.Forms.Selection.GetPreview](#)
- ◆ [Designer.Forms.Selection.Menu.AddItem](#)
- ◆ [Designer.Forms.Selection.Menu.<Name>.Change](#)
- ◆ [Designer.Forms.Selection.Menu.<Name>.Remove](#)
- ◆ [Designer.Editor.Document.Menu.AddItem](#)
- ◆ [Designer.Editor.Document.Menu.<Name>.Change](#)
- ◆ [Designer.Editor.Document.Menu.<Name>.Remove](#)
- ◆ [Designer.Editor.Document.Menu.AddItem](#)
- ◆ [Designer.Menu.Plugins.AddItem](#)
- ◆ [Designer.Menu.Plugins.<Name>.Change](#)
- ◆ [Designer.Menu.Plugins.<Name>.Remove](#)
- ◆ [Designer.Storage.*](#)
- ◆ [Designer.PluginsShutdown](#)

Neue Ereignisse

- ◆ [Designer.Forms.WriteLockChangeDone](#)
- ◆ [Designer.Forms.Selection.Menu.MenuSelect](#)
- ◆ [Designer.Editor.Document.Menu.MenuSelect](#)
- ◆ [Designer.Menu.Plugins.MenuSelect](#)
- ◆ [Designer.Exit](#)

Weitere Update-Beschreibungen

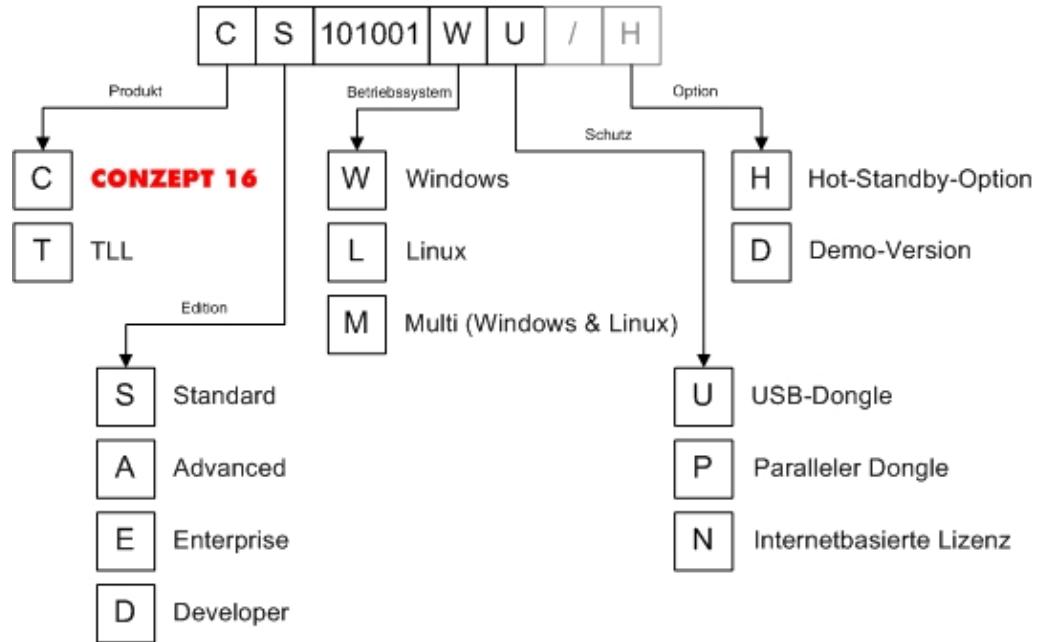
- [Update-Beschreibung 5.7](#)
- [Update-Beschreibung 5.6](#)
- [Update-Beschreibung 5.5](#)
- [Update-Beschreibung 5.4](#)
- [Update-Beschreibung 5.3](#)
- [Update-Beschreibung 5.2](#)
- [Update-Beschreibung 5.1](#)
- [Update-Beschreibung 5.0](#)

- [Update-Beschreibung des Clients 4.7](#)
- [Update-Beschreibung des Clients 4.6](#)
- [Update-Beschreibung des Clients 4.5](#)
- [Update-Beschreibung des Clients 4.4](#)

Lizenztypen

Beschreibung der Lizenztypen des CONZEPT 16-Servers

Alle Lizenzen sind mit einer eindeutigen Lizenznummer versehen, aus der sich der Lizenztyp und die Art des Lizenzschutzes ersehen lässt. Die Lizenznummern sind wie folgt aufgebaut:



Das erste Zeichen - Das Lizenzprodukt

Es werden zwei Lizenztypen ausgeliefert. "C" bezeichnet eine unbefristete CONZEPT 16-Lizenz. Diese Lizenz wird nur mit einem Dongle (USB oder Parallelport) ausgeliefert. Mit "T" ist eine TLL-Lizenz (Time-limited License) gekennzeichnet. Die Lizenz besitzt keinen Dongle, der Lizenzschutz erfolgt über einen Lizenzserver.

Das zweite Zeichen - Die Edition

Es gibt vier verschiedene Lizenz-Editionen. Die Editionen werden durch das zweite Zeichen unterschieden. Je nach Lizenz-Edition stehen dem Server gewisse Funktionen zur Verfügung:

Übersicht über die Leistungsmerkmale der verschiedenen Lizenztypen

	Developer-Edition (D)	Standard-Edition (S)	Advanced-Edition (A)	Enterprise (E)
Server für Windows				
Server für Linux				
Server Multilizenz			X	
Server für 32-Bit				
Server für 64-Bit	1			1
1 bis 8 logische Prozessoren				

Kontakt

9 bis 16 logische Prozessoren	✗	
mehr als 16 logische Prozessoren	✗	✗
Replikationssätze erzeugen	✗	
Windows-Standard-Client		
Windows-Advanced-Client	✗	
PDF-Verarbeitung		
Windows-Druckertreiber	✗	
DLL-Schnittstelle	✗	
ODBC-Schnittstelle	✗	✗
Web-Schnittstelle	✗	
SOA-Service	✗	
PHP-Schnittstelle	✗	✗
Laufwerkstreiber	✗	✗

¹ Die Standard- und Advanced-Edition können auch auf 64-Bit-Plattformen eingesetzt werden. Der maximal zuweisbare Datenbankcache liegt dann allerdings bei 1 GB.

Das dritte bis achte Zeichen - Die eindeutige Lizenznummer

Hierbei handelt es sich um eine eindeutige, sechsstellige Lizenznummer der CONZEPT 16-Serverlizenzen.

Das neunte Zeichen - Das Betriebssystem

In diesem Zeichen wird das Betriebssystem kodiert, unter dem die Lizenz eingesetzt werden kann. CONZEPT 16-Serverlizenzen können entweder unter Windows, unter Linux oder unter beiden Betriebssystemen eingesetzt werden.

W Die Lizenz kann nur unter Windows-Betriebssystemen eingesetzt werden.

L Die Lizenz kann nur unter Linux eingesetzt werden.

M Die Lizenz kann unter Windows und Linux eingesetzt werden.

Befindet sich eine Evaluierungslizenz im Einsatz, wird das durch die Buchstaben EV im neunten und zehnten Zeichen angezeigt. Die vorangegangenen Zeichen entsprechen der Lizenznummer der Lizenz, aus der die Evaluierungsversion erzeugt wurde. Informationen zum Erstellen einer Evaluierungsversion befinden sich im Abschnitt Evaluierungslizenz.

Das zehnte Zeichen - Der Lizenzschutz

CONZEPT 16-Lizenzen können mit einem Lizenz mit Hardwareschutz (Dongle) (USB oder Parallel-Port) oder einer Lizenz mit Softwareschutz (internetbasierte Lizenz) betrieben werden. Dieses Zeichen steht für den Typ des Lizenzschutzes.

U USB-Dongle

P Parallel-Dongle

N Internetbasierte Lizenzierung

Das Suffix - Optionen

Es werden folgende Lizenzoptionen angeboten:

H Hot-Standby

D Demo-Version

Die Hot-Standby-Option ist im Abschnitt CONZEPT 16-Server - Hot-Standby beschrieben. Die Demo-Version ist eine zeitlich limitierte Lizenz, die bei einem potentiellen Neukunden installiert werden kann, um die Funktionsweise des Programms im Umfeld des Neukunden zu demonstrieren. Bis auf die zeitliche Limitierung entspricht sie der gleichen Lizenz ohne die Option "/D".

Evaluierungsversion

Die Lizenznummer einer Evaluierungsversion entspricht der Lizenznummer der Developer-Edition, mit der sie generiert wurde (siehe Evaluierungslizenz). Sie unterscheidet sich aber im neunten und zehnten Zeichen. Bei einer Evaluierungslizenz steht an diesen Stellen "EV". Die Lizenz "CD101432EV" ist also eine Evaluierungslizenz zum Beispiel von der Lizenz "CD101432MU/H". Da bei der Evaluierungslizenz kein Dongle benötigt wird und sie auch nur unter Windows installiert werden kann, ist auch eine Unterscheidung der Dongletypen und Betriebssysteme nicht notwendig.

Kontakt

Installation

Beschreibung der Installation von CONZEPT 16

In diesem Kapitel befinden sich die CONZEPT 16-Systemvoraussetzungen und Hardware-Empfehlungen. Zudem werden im folgenden die CONZEPT 16-Installationsroutine und Besonderheiten bei der Installation beschrieben.

- Systemvoraussetzungen
- Hardware-Empfehlung für den Datenbank-Server
- Installationsroutine
- Speicherorte von Konfigurationsdateien

CONZEPT 16 - Systemvoraussetzungen

Beschreibung der CONZEPT 16-Systemvoraussetzungen

Alle CONZEPT 16-Komponenten können ausschließlich auf den Systemarchitekturen IA32 (Intel 32-Bit Architektur) und x64 (Intel/AMD 64-Bit Architektur) eingesetzt werden. Einige CONZEPT 16-Komponenten unterstützen zwar 64-Bit Betriebssystem-Versionen, diese werden dort aber in der 32-Bit Variante betrieben. In diesem Fall ist die Plattform mit einem * gekennzeichnet.

Im folgenden werden die Systemvoraussetzungen der einzelnen CONZEPT 16 Komponenten erläutert:

- Server
- Client und CONZEPT 16-DLL-Schnittstelle
- SOA-Service
- Druckertreiber
- ODBC-Treiber
- Web-Schnittstelle
- PHP-Schnittstelle
- Laufwerkstreiber

CONZEPT 16-Server

Der CONZEPT 16-Server kann auf den Betriebssystemen Windows und Linux in einer 32-Bit und in einer 64-Bit Variante betrieben werden. Abhängig von der Anzahl der logischen Prozessoren werden bestimmte Lizenzen vorausgesetzt. Andere Systemvoraussetzungen sind seitens CONZEPT 16 nicht gegeben.

Unter Linux wird mindestens die Kernel-Version 3.x vorausgesetzt.

Windows 7 (32-Bit)	Abhängig von der Anzahl der logischen Prozessoren können
Windows 8 (32-Bit)	die Standard- (maximal 8 logische Prozessoren), die Advanced-
Windows 8.1 (32-Bit)	(maximal 16 logische Prozessoren) oder die Enterprise-Edition
Windows 10 (32-Bit)	(mehr als 16 logische Prozessoren) installiert werden.
Linux Kernel 3.x (32-Bit)	
Linux Kernel 4.x (32-Bit)	
Kernel 5.x (32-Bit)	
Windows 7 (64-Bit)	Abhängig von der Anzahl der logischen Prozessoren können
Windows Server 2008 R2	die Standard- (maximal 8 logische Prozessoren), die Advanced- (maximal 16 logische Prozessoren) oder die Enterprise-Edition
Windows 8 (64-Bit)	(mehr als 16 logische Prozessoren) installiert werden.
Windows Server 2012	
Windows 8.1 (64-Bit)	
Windows Server 2012 R2	
Windows 10 (64-Bit)	
Windows Server 2016	
Windows Server 2019	
Linux Kernel 3.x (64-Bit)	

4.x (64-Bit) Linux

Kernel 5.x (64-Bit)

Für Windows-Betriebssysteme wird als Lizenzschutz ein USB- oder ein Parallelport-Dongle angeboten. Für Linux-Systeme stehen nur USB-Dongle zur Verfügung. Neben dem Dongle als Lizenzierung wird noch eine internetbasierte Lizenzierung angeboten. Nähere Informationen dazu befinden sich im Abschnitt Lizenz mit Softwareschutz beschrieben.

-  Wird ein Dongle verwendet, muss bei der Installation unter Windows der Benutzer über die entsprechenden Rechte zum Installieren des Dongletreibers und des Serverdienstes verfügen. Unter Linux-Systemen werden zur Installation des CONZEPT 16-Servers vom Benutzer Lese-, Schreib-, und Ausführungsrechte im Verzeichnis "opt" und im Zielverzeichnis des CONZEPT 16-Servers benötigt.
-  Bei der Installation unter Linux muss darauf geachtet werden, dass die Version der installierten glibc mindestens 2.17 sein muss.

CONZEPT 16-Client und CONZEPT 16-DLL-Schnittstelle

Der CONZEPT 16-Standard- und Advanced-Client sowie die CONZEPT 16 Programmierschnittstelle können auf den folgenden Betriebssystemen eingesetzt werden:

- Windows 7 (32/64*-Bit)
- Windows Server 2008 R2*
- Windows 8 (32/64*-Bit)
- Windows Server 2012*
- Windows 8.1 (32/64*-Bit)
- Windows Server 2012 R2*
- Windows 10 (32/64*-Bit)
- Windows Server 2016*
- Windows Server 2019*

CONZEPT 16-SOA-Service

Der CONZEPT 16-SOA-Service ist ein Dienst, welcher eine prozedurale Verarbeitung in einer CONZEPT 16-Datenbank durchführen kann. Betriebssystem ohne Dienst-Unterstützung können für den CONZEPT 16-SOA-Service daher nicht verwendet werden. Folgende Plattformen eignen sich für den SOA-Service:

- Windows 7 (32/64*-Bit)
- Windows Server 2008 R2*
- Windows 8 (32/64*-Bit)
- Windows Server 2012*
- Windows 8.1 (32/64*-Bit)
- Windows Server 2012 R2*
- Windows 10 (32/64*-Bit)
- Windows Server 2016*
- Windows Server 2019*
- Linux Kernel 3.x (32/64-Bit)
- Linux Kernel 4.x (32/64-Bit)
- Linux Kernel 5.x (32/64-Bit)

Kontakt

Die gestarteten Tasks des SOA-Service können auf 64-Bit Betriebssystemen auch in einer 64-Bit Variante betrieben werden. Dazu muss in der Konfigurationsdatei der Eintrag 64bit gesetzt sein.

CONZEPT 16-Druckertreiber

Der CONZEPT 16-Druckertreiber teilt sich in zwei Komponenten auf, den eigentlichen Gerätetreibern und einem Druckprozessor, also dem Druckserver. Die Gerätetreiber können auf den folgenden Plattformen eingerichtet werden:

- Windows 7 (32/64-Bit)
- Windows Server 2008 R2
- Windows 8 (32/64-Bit)
- Windows Server 2012
- Windows 8.1 (32/64-Bit)
- Windows Server 2012 R2
- Windows 10 (32/64-Bit)
- Windows Server 2016
- Windows Server 2019

Der Druckprozessor des CONZEPT 16-Druckertreibers kann auf den folgenden Systemen eingesetzt werden:

- Windows 7 (32/64*-Bit)
- Windows Server 2008 R2*
- Windows 8 (32/64*-Bit)
- Windows Server 2012*
- Windows 8.1 (32/64*-Bit)
- Windows Server 2012 R2*
- Windows 10 (32/64*-Bit)
- Windows Server 2016*
- Windows Server 2019*

ODBC-Treiber

Der ODBC-Treiber kann auf den folgenden Betriebssystemen installiert werden:

- Windows 7 (32/64-Bit) (mit SP1)
- Windows Server 2008 R2 (mit SP1)
- Windows 8 (32/64-Bit)
- Windows Server 2012
- Windows 8.1 (32/64-Bit)
- Windows Server 2012 R2
- Windows 10 (32/64-Bit)
- Windows Server 2016
- Windows Server 2019

Web-Schnittstelle

Für den Einsatz der CONZEPT 16-Web-Schnittstelle sind die Internet Information Services (IIS) ab der Version 3.0 notwendig. Für die Einstellungen des IIS siehe

Kontakt

Einstellungen des Internet Information Services. Die Web-Schnittstelle kann auf den folgenden Plattformen eingesetzt werden:

- Windows 7 (32/64-Bit)
- Windows Server 2008 R2
- Windows 8 (32/64-Bit)
- Windows Server 2012
- Windows 8.1 (32/64-Bit)
- Windows Server 2012 R2
- Windows 10 (32/64-Bit)
- Windows Server 2016
- Windows Server 2019

PHP-Schnittstelle

Voraussetzung für den Einsatz der CONZEPT 16-PHP-Schnittstelle ist unabhängig von der Betriebssystem-Umgebung ein installierter PHP-Interpreter. Dieser muss in der Version 5.2.x, 5.3.x oder 5.4.x vorhanden sein.

Die PHP-Schnittstelle kann auf 32-Bit sowie 64-Bit Betriebssystemen eingesetzt werden. Da derzeit für Windows-Plattformen keine 64-Bit PHP-Interpreter angeboten werden, können dort 32-Bit Interpreter mit der 32-Bit PHP-Schnittstelle von CONZEPT 16 verwendet werden.

Laufwerkstreiber

Der Laufwerkstreiber ist unter Windows Vista, 7, 8, 8.1, 10 sowie Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019 lauffähig. Der Laufwerkstreiber kann auf 32-Bit sowie 64-Bit Betriebssystemen eingesetzt werden.

Kontakt

Hardware-Empfehlung für den Datenbank-Server

Aufstellung der empfohlenen Hardwarekonfiguration beim Einsatz des CONZEPT 16-Servers

Siehe [Blog](#)

Die Hardware des Datenbank-Servers beeinflusst wesentlich die Leistungsmerkmale des gesamten Datenbanksystems. Daher ist beim Einsatz einer Datenbank ein starkes Augenmerk auf die zu verwendende Hardware zu legen. Vor der Investition in die einzelnen Server-Bausteine muss jedoch der Einsatzzweck des Systems genau bestimmt werden. Dabei spielt die Datenbankapplikation die größte Rolle. Denn diese bestimmt die spätere Arbeitsweise mit der gesamten Software-Umgebung und bedingt damit die Hardware-Konfiguration des Datenbank-Servers.

Die Server-Hardware muss je nach Applikation bestimmte Voraussetzungen für einen performanten Einsatz erfüllen. Bei jeder Applikation und Installation müssen diese Voraussetzungen überprüft und gegebenenfalls variiert werden. Daher verstehen sich alle Angaben auch nur als ungefähre Richtwerte. Nachfolgend die Faktoren, die die Performanz eines Datenbanksystems beeinflussen:

- Die Anzahl der Anwender, die mit dem System arbeiten **und**
- die Häufigkeit der Datenbankanfragen
- Das zu erwartende Transaktionsvolumen, das die Anwendung erzeugt
- Die voraussichtliche Größe der Datenbank, in Bezug auf die Laufzeit des Systems

Anwenderzahl und Anfragehäufigkeit

Die Anzahl der Benutzer und die Häufigkeit der Anfragen wirkt sich in erster Linie auf die Prozessorbelastung aus. Zu empfehlen sind dabei unabhängig von der Benutzeranzahl mindestens zwei Kerne, sodass zwei logische Einheiten die Anfragen parallel bearbeiten können. Ab einer Benutzeranzahl von 50 Anwendern ist zu einem System mit 4 Prozessorkernen zu raten. Die Anzahl der Prozessorkerne lässt sich entsprechend der folgenden Tabelle weiterführen:

bis 50 Benutzer 2 - 4 Kerne

bis 100 Benutzer 4 - 8 Kerne

bis 200 Benutzer 8 oder mehr Kerne

Zu dem Aspekt der Benutzeranzahl muss natürlich auch die Häufigkeit der Anfragen entsprechend berücksichtigt werden. Bei einer hohen Anfragefrequenz sollte auch ein höherer Wert auf die Prozessorkonfiguration gelegt werden. Eine umfassende Selektionsverarbeitung ist beispielsweise ein Merkmal, das sich auch auf die Wahl des Prozessors auswirken muss.



Höhere Taktraten des Prozessors sind wichtiger als eine große Anzahl von Cores.

Die vermutete Anfragehäufigkeit muss sich auch in der Hauptspeicherausstattung des Systems wiederspiegeln. Denn auch wenn der Server eine große Menge an Anfragen durch die Prozessoren verarbeiten kann, müssen dennoch die Daten aus der Datenbank gelesen werden. Eine gute Hauptspeicherausstattung führt ebenfalls zu einer schnelleren Verarbeitungszeit der Anfragen, da die Daten bereits im Cache

Kontakt

vorgehalten werden können (siehe hierzu auch Transaktionsvolumen).

Transaktionsvolumen

Der Einsatzzweck der Datenbankapplikation, die Applikationslogik und die Häufigkeit sowie das Volumen der Datenbankanfragen bestimmen das zu erwartende Transaktionsvolumen (siehe auch Transaktionsmanagement des CONZEPT 16 Servers). Um Transaktionen verarbeiten zu können, muss der Datenbank-Cache entsprechend eingerichtet werden. Bezuglich der Hardware-Konfiguration des Datenbank-Servers muss sich das Volumen der Transaktionen also direkt auf den zur Verfügung stehenden Hauptspeicher auswirken.

Der einer Datenbank zur Verfügung stehende Hauptspeicher muss in der Regel 10% der Datenbankgröße betragen. Mehr als diese 10% sind insbesondere bei vermehrtem Datenaufkommen ratsam, insofern es die Systemumgebung zulässt. Daher wird empfohlen den Datenbank-Server auf einer 64-Bit-Platform einzusetzen. Dazu eignen sich in erster Linie "Windows Server 2012" und der "Linux Kernel 3.x". Dort besteht dann die Möglichkeit den notwendigen Hauptspeicher für die Datenbanken einzurichten (siehe auch Architektur des Servers). Bei Datenbanken, die nur als Container für binäre Daten dienen, ist es nicht notwendig einen großen Cache anzulegen. Ein großer Datenbank-Cache ist nur nötig, wenn Daten überwiegend in der Baumstruktur einer CONZEPT 16-Datenbank organisiert werden (siehe Architektur der Datenbank).



Der Hauptspeicher eines Datenbank-Servers stellt nicht nur eine Verbindung zwischen Datenverarbeitung und Datenspeicherung her. Durch das Zwischenspeichern von gelesenen oder noch zu schreibenden Daten steuert er auch einen maßgeblichen Teil zur Performance des gesamten Systems bei. Daher muss dieser Komponente eine erhöhte Aufmerksamkeit geschenkt werden. In der Regel ist der Hauptspeicher die kritische Komponente eines Systems und führt gegebenenfalls zu einer höheren oder niedrigeren Performance.

Neben der Größe des Hauptspeichers werden auch die benötigten Input / Output-Kapazitäten des Storage-Systems vom Transaktionsvolumen beeinflusst. Selbst ein großer Hauptspeicher kann in der Regel nicht alle Informationen vorhalten. Die notwendigen Schreib- und Leseoperationen auf der Speichereinheit dürfen dann nicht zu einer inakzeptablen Verzögerung führen (siehe hierzu auch Datenbankgröße).

Datenbankgröße

Die Größe der Datenbank beeinflusst nicht nur wie erwähnt die Größe des notwendigen Hauptspeichers, sondern ebenfalls wie das Transaktionsvolumen auch das notwendige Storage-System. Für das Arbeiten mit einem Datenbanksystem werden generell SSDs mit SATA- (Serial Advanced Technology Attachment) oder SAS- (Serial Attached SCSI) Anschluss mit einer möglichst niedrigen Latenz empfohlen.

Die Speichergrundlage eines Datenbanksystems sollte ein RAID-Verbund sein. Für eine optimierte Schreib- und Leseleistung empfiehlt sich der Einsatz eines Stripings (RAID 0). Dabei werden die Daten auf mehrere SSDs verteilt geschrieben, wodurch sich die einzelnen Zugriffsgeschwindigkeiten addieren. Um einem Fehlerfall im Storagebereich vorzubeugen, wird dies noch mit einem Mirroring (RAID 1)

Kontakt

kombiniert. Der Einsatz eines RAID 10 Systems eignet sich also am besten für den Einsatz eines Datenbanksystems.

CONZEPT 16-Installationsroutine

Beschreibung der CONZEPT 16-Installationsroutine

Installation

Siehe [automatisieren](#)

[\(Blog\)](#)

Für die Installation von CONZEPT 16 gibt es eine Windows-Installationsroutine, die den Datenbank-Server und alle Client-Komponenten enthält. Die Installationsroutine befindet sich auf der CONZEPT 16-CD und kann aus dem Kundenbereich des Internet-Portals der vectorsoft AG (www.vectorsoft.de) heruntergeladen werden. Über die Installationsroutine ergeben sich folgende Möglichkeiten:

- [Installation](#)
- [Ändern der Installation](#)
- [Aktualisieren der Komponenten](#)
- [Entfernen](#)
- [Kommandozeilen-Parameter und Rückgabewerte der Installationsroutine](#)

Installation

Die Installationsroutine führt den Administrator durch alle notwendigen Schritte, um den Datenbank-Server und alle zur Verfügung stehenden Datenbank-Clients auf einem Zielsystem einzurichten. Während der Installation können die einzurichtenden Komponenten ausgewählt werden (Für eine detaillierte Beschreibung der Installation der Komponenten siehe Verknüpfungen im unten abgebildeten Komponentenbaum). Folgende Module enthält die CONZEPT 16-Installationsroutine:

- [Server](#) (siehe auch [Hot-Standby Installation](#)
 - ◆ Server (Windows)
 - ◆ [Server \(Linux\)](#)
 - ◆ Control-Center
- [Dokumentation](#)
- Client
 - ◆ [Standard- & Advanced-Client](#) (ebenfalls [Debugger](#))
 - ◆ SOA-Service
 - ◊ [SOA-Service \(Windows\)](#)
 - ◊ [SOA-Service \(Linux\)](#)
 - ◆ [API-Client](#)
 - ◆ [Web-Client](#)
 - ◆ [PHP-Client](#)
 - ◆ [Drucker](#)
 - ◊ Druckertreiber
 - ◊ Druckerprozessor
 - ◆ [ODBC \(32-bit\)](#)
 - ◊ ODBC-Treiber (32-bit)
 - ◆ ODBC (64-bit)
 - ◊ ODBC-Treiber (64-bit)
 - ◆ [Laufwerkstreiber](#)
- [Gemeinsame Komponenten](#)

Kontakt

Nach dem Starten der Installationsroutine wird der Startbildschirm angezeigt. Hier erfolgt eine Information, welche CONZEPT 16-Version im weiteren Verlauf auf dem System installiert wird. Die Installation wird über [Weiter >] fortgeführt. Über [< Zurück] kann auf den folgenden Seiten zum jeweils vorherigen Schritt zurückgewechselt werden. Mit [Abbrechen] kann der Administrator die Installation vorzeitig abbrechen.

Der Großteil der CONZEPT 16-Programmdateien wird bei der Installation einfach aus der Installationsdatei entpackt. Einige Module, wie beispielsweise Dienste, werden allerdings auch auf dem Zielsystem eingerichtet und mit der Software-Umgebung verknüpft. Solche Module älterer Versionen (Version 5.5 oder älter) müssen vor der Installation des CONZEPT 16-Pakets durch den Administrator vom Zielsystem entfernt werden. Installierte Programmteile der Version 5.6.01 erkennt die Installationsroutine automatisch und deinstalliert diese. Alle Konfigurations- und Log-Dateien werden dabei selbstständig gesichert und für die neue Installation weiterverwendet. Dies gilt auch für die Konfigurationsdateien des CONZEPT 16 Service-Clients, dessen Programmdateien aber nicht von der Installationsroutine entfernt werden können. Erkennt die Installationsroutine installierte und zu entfernende CONZEPT 16-Komponenten, werden diese nach der Anzeige der Fehlerkorrekturen angezeigt.

Für die Installation des CONZEPT 16-Pakets wird in diesem Fenster der Zielpfad für die Programmdateien festgelegt. Dabei werden automatisch Unterverzeichnisse für die jeweiligen Komponenten verwendet.

Im nächsten Schritt werden die CONZEPT 16-Komponenten definiert, die auf dem Zielsystem eingerichtet werden sollen. Auf einem zentralen System können beispielsweise alle Module erstellt und anschließend für das Netzwerk freigegeben oder verteilt werden. Durch die Auswahlmöglichkeit der Module können aber auch nur spezielle Programmteile auf dem aktuellen System eingerichtet beziehungsweise erstellt werden.

In diesem Dialog wird festgelegt, in welchem Ordner die Startmenüverknüpfungen angelegt werden sollen. Dabei kann ein freiählbarer Name vergeben oder einen bereits vorhandener Ordner aus dem aus dem Programmregister im Startmenü ausgewählt werden. Über die Option im unteren Bereich kann das Erstellen von Startmenüverknüpfungen unterbunden werden.

Anschließend folgen spezielle Konfigurationsmöglichkeiten für bestimmte Komponenten. Diese können Sie den entsprechenden Installationsbeschreibungen entnehmen (siehe Verknüpfungen im Komponentenbaum).

Für einige Module werden Laufzeitkomponenten von Microsoft Visual Studio benötigt. Sind die Module ausgewählt und die entsprechenden Laufzeitkomponenten nicht auf dem System installiert, werden die benötigten Komponenten im folgenden Dialog angezeigt.

[Liste der benötigten Laufzeitkomponenten](#)

Kontakt

Anschließend erscheint eine Zusammenfassung der Installationseinstellungen. In der darauffolgenden Prozessleiste kann der Installationsstatus abgelesen werden. Mit [Fertigstellen] wird die Installation nach dem Einrichten der Programmdateien abgeschlossen.

Gemeinsame Komponenten

Mit gemeinsamen Komponenten werden beispielsweise Programmbibliotheken und ActiveX-Steuerelemente für Ctx-Objekte wie das CtxDocEdit exportiert. Diese müssen auf jedem Arbeitsplatz, der die betreffenden Objekte verwenden soll, installiert werden.

Chromium Package

Das Chromium-Objekt benötigt zur Ausführung weitere Komponenten. Diese Komponenten sind im Chromium-Package enthalten, dessen Basis das Chromium-Embedded-Framework (CEF) ist Chromium Embedded Framework in CONZEPT 16 integriert.

vectorsoft stellt das Chromium-Package in einer Datei zur Verfügung, deren Name folgender Nomenklatur folgt:

c16_com_cro_<CEF-Chromium-Version>.cpn

<CEF-Chromium-Version> bildet hierbei die CEF- und Chromium-Version der im Package enthaltenen Komponenten ab.

Beispiel:

c16_com_cro_85_3_7+g7de1243+chromium-85_0_4183_102.cpn CEF-Version: 85.3.7
Chromium-Version 85.0.4183.102

Diese Nomenklatur folgt der offiziellen Benennung auf der CEF-Seite: Automatisierte Builds Das Chromium-Package wird von vectorsoft gepflegt, wenn Sicherheits-Updates oder wichtige Neuerungen des CEF erfolgen bzw. der Chromium-Komponenten.

Installation des Chromium-Package

Da das Chromium-Package nur von Anwendern benötigt wird, die auch das neue Chromium-Objekt verwenden und um die Installation des Chromium-Packages von der Distribution von CONZEPT 16-Releases abzukoppeln, ist das Chromium-Package nicht wie alle anderen Komponenten in der Setup-Routine enthalten.

Die Setup-Routine berücksichtigt beim Starten die Existenz eines Chromium-Packages. Liegt das Chromium-Package im selben Verzeichnis, wie die Setup-Routine, wird in der Liste der Komponenten unter "Common Components" der Eintrag "Chromium" angezeigt.

Die Setup-Routine prüft beim Start die Integrität des Chromium-Package und dessen Kompatibilität zum zu installierenden bzw. bereits installierten CONZEPT 16-Client.

Kontakt

Bei einem Fehler wird eine entsprechende Meldung angezeigt und die Setup-Routine beendet sich.

Der Inhalt des Chromium-Packages kann auch über die Schalteroptionen

```
c16_setup_5.8.12z.exe /cpn_action=extract /cpn_com=2 /dir=<Verzeichnis>
```

in das angegebene Verzeichnis extrahiert werden. Bei der Verwendung des Packages ist die Kompatibilität mit dem CONZEPT 16-Client zu berücksichtigen.

Startmenüeinträge

Abhängig von den installierten Modulen werden folgende Verknüpfungen im Programmverzeichnis des Startmenüs angelegt. Dies ist nicht der Fall, wenn das Anlegen der Startmenüeinträge während der Installation unterbunden wurde.

\Client\Designer-Tools (optional)

Benutzerverwaltung

BLOB-Verwaltung

Datensatzverwaltung

Ressourcenverwaltung

Validierungs-Editor

\Client

Advanced-Client

Standard-Client

Debugger

\Server

Control-Center

Lizenzinformationen

Log-Viewer

CONZEPT 16-Installation anpassen

CONZEPT 16-Dokumentation

CONZEPT 16-Webseite

CONZEPT 16-Webseite

Ändern der Installation

Die Installationsroutine kann auch auf einem System gestartet werden, auf dem die Installation bereits durchgeführt wurde. In diesem Fall wird eine Seite angezeigt, auf der der Administrator definiert, ob die Installation geändert, aktualisiert oder entfernt werden soll.

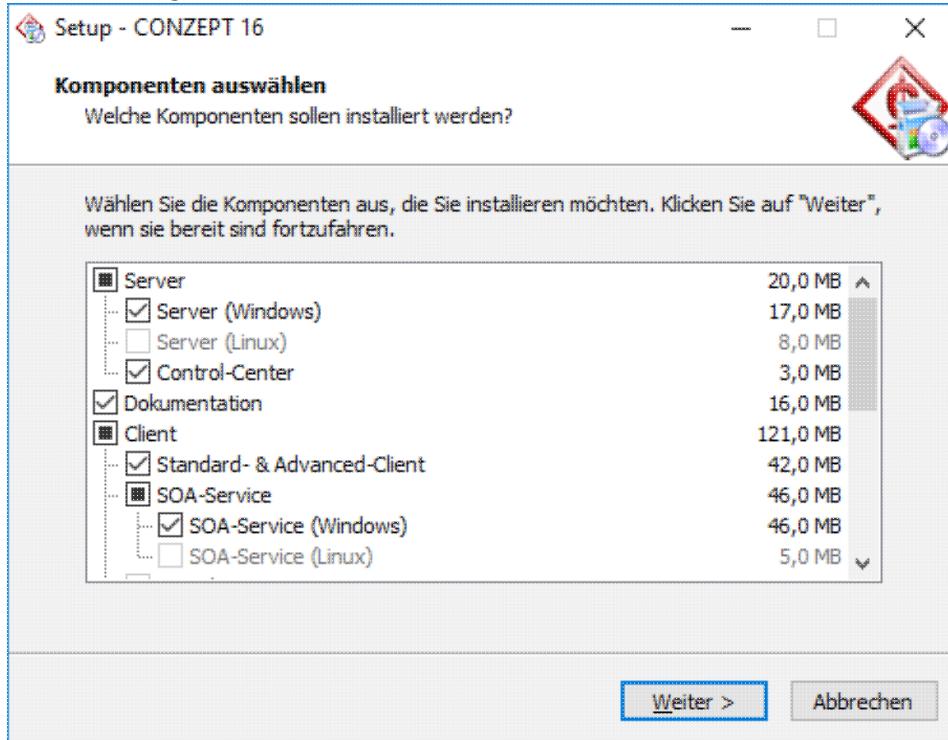
Kontakt

Wird die Funktion [Ändern] ausgewählt erscheint die Komponentenauswahl, ähnlich der Auswahl bei der Erstinstallation. Neue Komponenten können durch hinzufügen eines Häkchens installiert werden. Komponenten bei denen das Häkchen entfernt wird, werden deinstalliert.

Aktualisieren der Komponenten

Wird nach dem Starten der Installationsroutine die Funktion [Aktualisieren] ausgewählt, bekommt der Administrator den Komponentenbaum mit allen bereits installierten Komponenten angezeigt. Alle hier markierten Einträge werden im nächsten Schritt von der Installationsroutine aktualisiert. Das Auswählen von noch nicht installierten Modulen ist hier nicht möglich.

- i** Beim Aktualisieren ist zu beachten, dass die zu aktualisierenden Programmteile während des Vorgangs nicht zur Verfügung stehen und laufende Dienste zur Aktualisierung kurz gestoppt werden müssen. Daher ist beispielsweise bei einer Aktualisierung des Datenbank-Servers zu beachten, dass sich keine Benutzer mehr in den Datenbanken befinden. Ansonsten kann der Vorgang nicht korrekt durchgeführt werden.



Vor der Installation wird überprüft, ob die Komponenten ersetzt werden können. Dazu werden die Dienste angehalten. Programme, die noch geöffnet sind und Dienste, die nicht angehalten werden konnten, werden vor der Installation in einem Dialog angezeigt.

Über die Schaltfläche [Aktualisieren] kann die Anzeige der offenen Applikationen erneuert werden.

Entfernen

Bei der Aktionsauswahl wird die Funktion [Entfernen] ausgewählt, um alle auf dem System installierten CONZEPT 16-Komponenten zu entfernen. Die Deinstallation kann ebenfalls über die Software-Steuerung des Betriebssystems aufgerufen werden.

Bei der Deinstallation werden keine Konfigurationsdateien entfernt. Die Dateien verbleiben in den entsprechenden Verzeichnissen (siehe Speicherorte von Konfigurationsdateien). Bei einer erneuten Installation können diese Konfigurationsdateien dann auch von der Installationsroutine verwendet werden.

Befinden sich im Startverzeichnis der Installationsroutine Konfigurationsdateien, werden diese nur dann zur Installation verwendet, wenn auf dem System noch keine Konfigurationsdateien vorhanden sind. Um die vorhandenen Konfigurationsdateien zu ignorieren, muss das Setup mit dem Parameter overwrite aufgerufen werden.

Kontakt

CONZEPT 16-Server - Installation des Servers unter Windows

Beschreibung der Installation des CONZEPT 16-Servers unter Windows

Der CONZEPT 16-Server wird über die CONZEPT 16 Installationsroutine eingerichtet.

Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Server. Über die Installationsroutine können die folgenden Komponenten für den Server unter Windows installiert werden:

- **Server (Windows)**

Diese Option beinhaltet den Datenbank-Server, das Lizenzinformationsprogramm und das Script-Utility.

- **Control-Center**

Mit dieser Option wird das Control-Center zur Administration des Datenbank-Servers installiert. Eine separate Installation des Control-Centers ist dann sinnvoll, wenn der Datenbank-Server vom aktuellen System aus nur administriert werden soll. Gerade wenn der Server auf einem Linux-System eingerichtet wurde, bietet diese Variante eine sehr komfortable Möglichkeit zur Server-Verwaltung. Das Control-Center beinhaltet ebenfalls den Log-Viewer.

Installationsverlauf

Während der Installation des Datenbank-Servers über die Installationsroutine werden zusätzliche Dialoge zur Einrichtung des Servers angezeigt. Über das erste Konfigurationsfenster kann entschieden werden, ob der CONZEPT 16-Datenbank-Server als Dienst eingerichtet werden soll und somit beim Systemstart automatisch mit initialisiert wird (siehe auch Starten und Stoppen unter Windows). Ein Anmelden eines Benutzers an das Betriebssystem ist dann für den Serverbetrieb nicht notwendig. Wird der Server als Dienst eingerichtet, kann der Administrator über die zweite Option festlegen, dass der Server-Dienst nach der Installation automatisch gestartet werden soll.

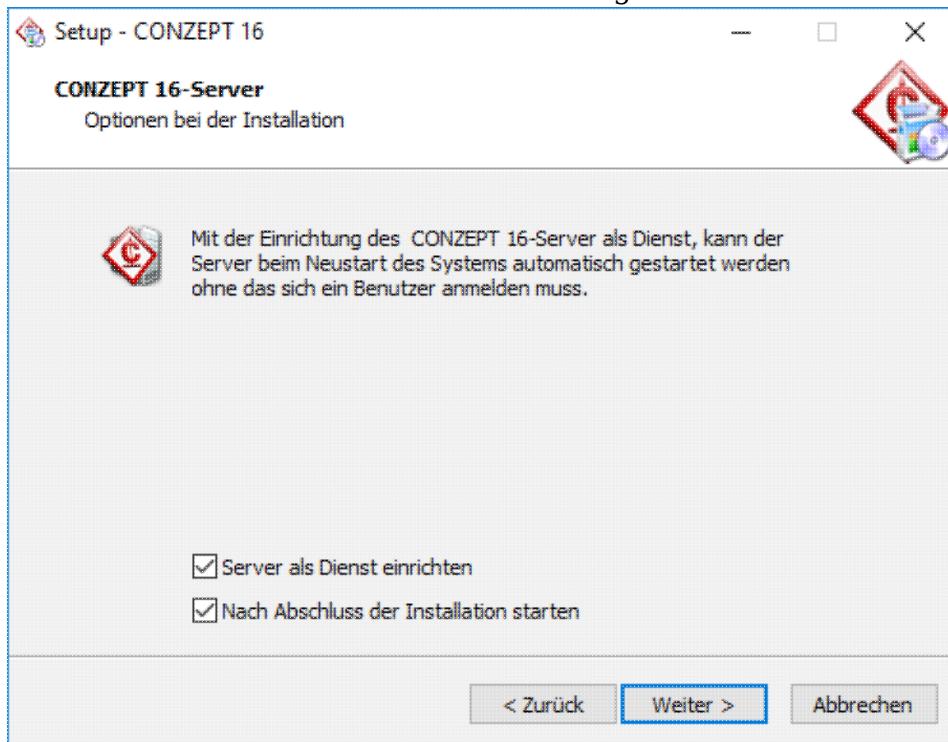
Falls der Server nicht als Dienst eingerichtet wird, ist die zweite Option nur aktiv, wenn das Control-Center ebenfalls zur Installation ausgewählt wurde. Der Datenbank-Server wird in diesem Fall nach der Installation durch das Control-Center im Detached-Mode gestartet. Dies ist dann auch bei jedem nächsten Start des Control-Centers der Fall. Diese Einstellung kann später im Control-Center noch angepasst werden.



Windows Server-Betriebssysteme räumen dem Dateicache für Dateifreigaben im Netzwerk in der Regel eine sehr hohe Priorität ein. Dies kann unter Umständen dazu führen, dass der Datenbank-Cache des CONZEPT 16-Servers benachteiligt wird und es zu Performance-Verlusten kommt. Um diese Priorisierung zu umgehen, kann in der Netzwerkkonfiguration des Betriebssystems die Option "Lastenausgleich durchführen" aktiviert werden. Ist diese Option zum Zeitpunkt der Server-Installation nicht aktiv, bietet die Installationsroutine dem Administrator über das zusätzliche Kontrollkästchen "Systemcache-Einstellung anpassen" die Möglichkeit, den Lastenausgleich automatisch aktivieren zu

Kontakt

lassen. Nach der Installation wird bei Aktivierung der Option durch die Installationsroutine ein Neustart angeboten.



Für den Betrieb des CONZEPT 16-Servers stehen zwei Lizenzmodelle zur Verfügung. Die verwendete Lizenzierung wird in dem folgenden Dialog ausgewählt:

Es kann zwischen "Dongle-Lizenz installieren" ([Lizenz mit Hardwareschutz \(Dongle\)](#), es wird ein Dongle benötigt) und "Internetbasierte Lizenz installieren" ([Lizenz mit Softwareschutz \(Internetbasierte Lizenz\)](#), es wird eine Lizenzidentitätsdatei (c16_?????.idn) benötigt) gewählt werden. Über die Schaltfläche [>>] können weitere Informationen zu den jeweiligen Lizenzmodellen angezeigt werden.

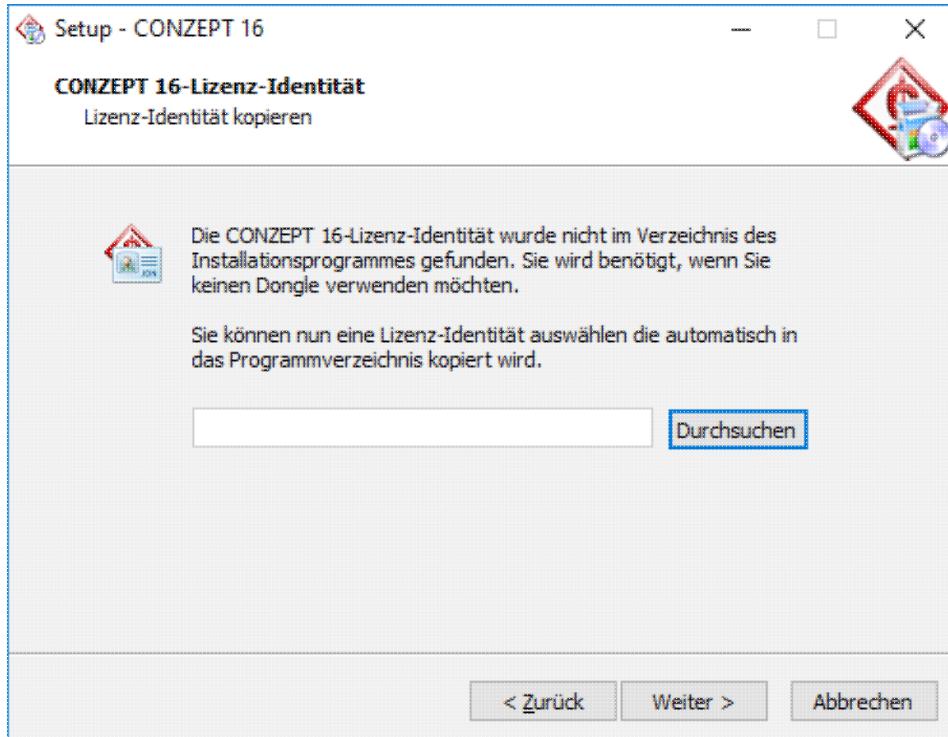
Anschließend wird der Speicherort der [Lizenzdatei](#) ausgewählt, welche die Lizenzausprägungen zu den Lizenznummern enthält. Die Datei wird dann im Verzeichnis für [Konfigurationsdateien](#) abgespeichert. Falls die Lizenzdatei noch nicht vorliegt, kann über den Link auch direkt die vectorsoft-Hompepage angezeigt werden, um die aktuelle Lizenzdatei herunterzuladen.

- Dieses Fenster wird nicht angezeigt, falls eine Lizenzdatei bereits im Verzeichnis der Installationsroutine abgelegt wurde und somit automatisch übernommen werden kann. Falls eine Lizenzdatei aus dem Programmstand des Servers 5.1 übernommen wird, taucht dieses Fenster wiederum auf, damit eine Lizenzdatei der Version 5.2 angegeben werden kann, insofern diese nicht schon im alten Programmverzeichnis vorhanden war.

Wurde bei den Optionen der Lizenzinstallation "Internetbasierte Lizenz installieren" ausgewählt, kann in einem weiteren Dialog der Speicherort der Lizenz-Identitätsdatei angegeben werden. Die Datei wird dann ebenfalls im Verzeichnis für

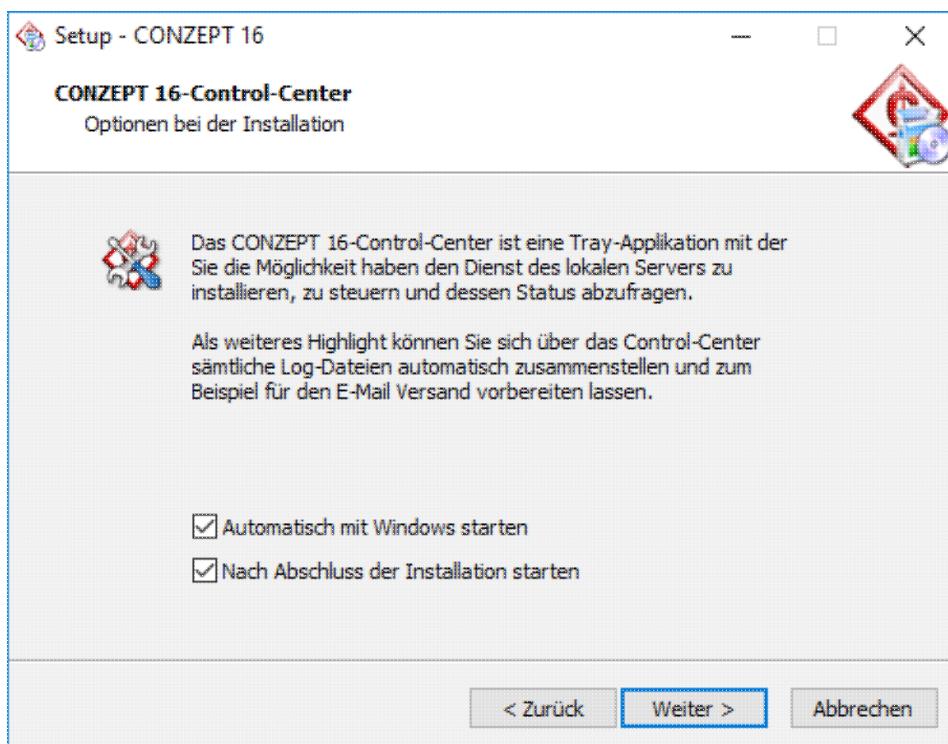
Kontakt

Konfigurationsdateien abgespeichert. Befindet sich diese Datei bereits im Verzeichnis der Installationsroutine, erfolgt keine Abfrage und die Datei wird in das korrekte Verzeichnis kopiert.

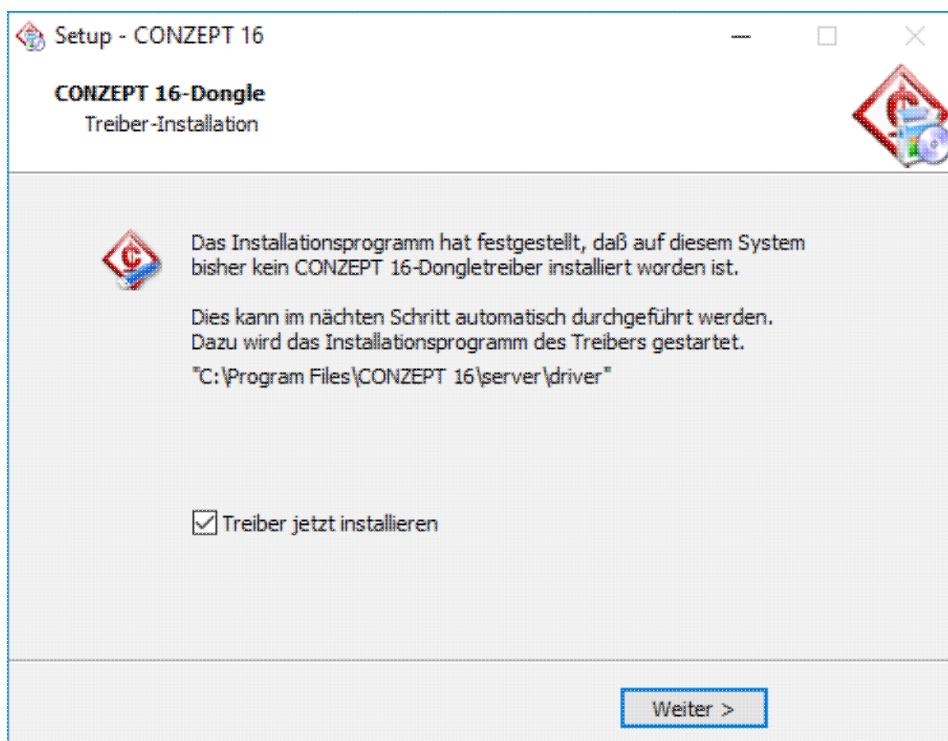


Wurde bei der Installation auch das Control-Center ausgewählt, kann hier entschieden werden, ob das Control-Center so eingerichtet werden soll, dass es bei jeder Benutzeranmeldung mitgestartet wird. Diese Einstellung kann nach der Installation in den Einstellungen des Control-Centers noch angepasst werden. Mit der zweiten Option wird definiert, ob das Control-Center nach der Installation automatisch gestartet wird. Die zweite Option ist deaktiv, falls der Server im Anschluss durch das Control-Center im Detached-Mode gestartet werden soll.

Kontakt



Falls nach der Installation auf dem System kein Treiber für den CONZEPT 16 Lizenzdongle gefunden und keine internetbasierte Lizenz verwendet wird, kann die Installationsroutine die Treiberinstallation im Anschluss automatisch starten. Dazu wird die in diesem Fenster angebotene Option aktiviert. Die Installation des Dongles kann aber auch später mit der Datei c16_dongle.exe aus dem Verzeichnis \driver gestartet werden. Erst nach der Installation des Dongle-Treibers darf der CONZEPT 16-Lizenzdongle auf das System aufgesteckt werden.



Installation des Dongle-Treibers

Zur Installation des Dongle-Treibers müssen zunächst die Lizenzvereinbarungen bestätigt werden.

In diesem Dialog muss die komplette Installation ausgewählt werden. Nur dann werden alle notwendigen Treiber für den Betrieb eines CONZEPT 16-Lizenzdongles installiert.

Programmdateien

Dateien des Servers

c16_control.exe	<u>Control-Center</u>
c16_control_admin.exe	Administrator-Modul des Control-Centers
c16_icu_w32.dll	Unicode-Bibliothek für 32-Bit-Systeme
c16_icu_w64.dll	Unicode-Bibliothek für 64-Bit-Systeme
c16_lgbview.exe	<u>Log-Viewer</u>
c16_licinfo.exe	Programm zum Auslesen der Lizenzdatei
c16_serv_cmd_win.exe	<u>Script-Utility</u>
c16_serv_dba_w32.dll	Datei des Datenbank-Prozesses für 32-Bit-Systeme
c16_serv_dba_w64.dll	Datei des Datenbank-Prozesses für 64-Bit-Systeme
c16_serv_mgr_w32.dll	Datei des Manager-Prozesses für 32-Bit-Systeme
c16_serv_mgr_w64.dll	Datei des Manager-Prozesses für 64-Bit-Systeme
c16_serv_rpu_w32.dll	<u>Replikations Utility</u> für 32-Bit-Systeme
c16_serv_rpu_w64.dll	<u>Replikations Utility</u> für 64-Bit-Systeme
c16_serv_svc_win.dll	Datei des Service-Prozesses
c16_serv_vss_a32.dll	Datei des <u>VSS-Writers</u> für 32-Bit-Systeme (Windows Server 2003 / Windows Server 2008 / Windows Vista)
c16_serv_vss_a64.dll	Datei des <u>VSS-Writers</u> für 64-Bit-Systeme (Windows Server 2003 / Windows Server 2008 / Windows Vista)
c16_serv_vss_w32.dll	Datei des <u>VSS-Writers</u> für 32-Bit-Systeme ab Windows Server 2008 R2 / Windows 7
c16_serv_vss_w64.dll	Datei des <u>VSS-Writers</u> für 64-Bit-Systeme ab Windows Server 2008 R2 / Windows 7
c16_ssl_w32.dll	SSL-Bibliothek für 32-Bit-Systeme
c16_ssl_w64.dll	SSL-Bibliothek für 64-Bit-Systeme
c16.vra	Ressourcendatei
\driver	Treiber für den CONZEPT 16-Lizenz-Dongle
c16_dongle.exe	Installationsroutine für den Dongle-Treiber
\example	Beispieldateien

Kontakt

area_backup.cmd	Skript für das CONZEPT 16-Script-Utility (Kopieren einer Datenbank)
backup_onoff.cmd	Skript für das CONZEPT 16-Script-Utility (Ein-/Ausschalten von <u>Sicherungssereignissen</u>)
 i	Soll eine Internet-basierte Lizenz installiert werden, kann die notwendige Lizenz-Identitätsdatei in das Verzeichnis der Installationsroutine kopiert werden. Sie wird dann automatisch in das richtige Verzeichnis kopiert.
 i	Die Lizenzinformationen werden aus der <u>Lizenzdatei</u> ausgelesen. Diese kann vor dem Start der Installation im Verzeichnis der Installationsroutine abgelegt werden. Von dort aus wird sie automatisch bei der Installation in das entsprechende Verzeichnis für <u>Konfigurationsdateien</u> kopiert.
 i	Der Datenbank-Server entnimmt seine Konfiguration aus der <u>Datenraumtabelle</u> . Diese kann bei der Installation vordefiniert werden. Hierzu wird die Datei c16_serv.ars vor dem Start der Installation in das Verzeichnis der Installationsroutine kopiert.

Kontakt

CONZEPT 16-Server - Installation des Servers unter Linux

Beschreibung der Installation des CONZEPT 16-Servers unter Linux

Siehe [Blog](#)

Die Programmdateien des [CONZEPT 16-Servers](#) unter Linux werden über die CONZEPT 16 [Installationsroutine](#) erstellt. Die Programmdateien und die Installationsroutine für den [Lizenz-Dongle](#) befinden sich anschließend im Installationsverzeichnis unter \Server\linux im komprimierten TAR-Archiv c16_server_lnx.tar.gz. Dieses wird auf ein Linux-System transferiert und dort mit dem folgenden Kommando dekomprimiert und entpackt:

```
tar -zxvf c16_server_lnx.tar
```

Dies geht auch in einzelnen Schritten mit den Kommandos:

```
gzip -d c16_server_lnx.tar.gztar -xvf c16_server_lnx.tar
```

Inhalt des Archivs

c16_serv_svc_l32	Serviceprozess (Daemon) 32-Bit
c16_serv_svc_l64	Serviceprozess (Daemon) 64-Bit
c16_serv_mgr_l32	Managerprozess für 32-Bit
c16_serv_mgr_l64	Managerprozess für 64-Bit
c16_serv_dba_l32	Datenbankprozess für 32-Bit
c16_serv_dba_l64	Datenbankprozess für 64-Bit
c16_serv_cmd_l32	Script-Utility für 32-Bit
c16_serv_cmd_l64	Script-Utility für 64-Bit
c16_serv_rpu_l32	Replication Utility für 32-Bit
c16_serv_rpu_l64	Replication Utility für 64-Bit
c16.vra	Ressourcendatei
c16_serv	Init-Skript (SUSE)
c16_error	Liste der behobenen Fehler
\driver	Treiber für den CONZEPT 16-Lizenz-Dongle
sntl-sud-7.5.2.tar.gz	Dongle-Treiber für den Linux-Kernel 2.6
debian_7.5.1.tar.gz	Dongle-Treiber im Debian Binärpaketformat

Installation des Dongle-Treibers

Wird eine internetbasierte Lizenz verwendet, muss der Dongle-Treiber nicht installiert werden.

Zu Beginn der Installation muss der Dongle-Treiber eingerichtet werden. Die Dongle-Treiber für den CONZEPT 16-Lizenz-Dongle werden ebenfalls als komprimierte TAR-Archive zur Verfügung gestellt und können wie oben beschrieben dekomprimiert und entpackt werden. Dabei ist zu beachten, dass der Treiber für den Kernel 2.6 mindestens SUSE Linux 10 SP 2 voraussetzt. Zur Installation sind folgende Schritte notwendig:

- Im Verzeichnis der entpackten Dateien das Installationsskript mit der Anweisung "sh sud_install.sh" ausführen.

Kontakt

- Lizenzvereinbarung bestätigen.
- Nach erfolgreicher Installation wird die Meldung "Sentinel USB Daemon installation was successful" ausgegeben. Anschließend kann der CONZEPT 16-USB-Dongle auf einen USB-Port aufgesteckt werden.

Bei korrekter Installation wird distributionsabhängig entweder bei jedem Neustart des Rechners der USB-Daemon automatisch mitgestartet oder er muss manuell geladen werden. Dies geschieht über die Anweisung "sh load_daemon.sh start" aus dem Verzeichnis opt/safenet_sentinel/common_files/sentinel_usb_daemon.

Die Deinstallation des Dongle-Treibers kann über die Anweisung sh sud_uninstall.sh vorgenommen werden. Dieses Skript befindet sich im mitgelieferten Archiv.

Installation des CONZEPT 16-Servers

Zu den Dateien des CONZEPT 16-Servers wird die Lizenzdatei und eine eventuell schon vorhandene Datenraumtabelle kopiert. Anschließend kann der Server gestartet werden (siehe Starten und Stoppen des Servers unter Linux).



Der Server kann so eingerichtet werden, dass er beim Start des Betriebssystems automatisch mitgestartet wird. Details zum Boot-Konzept der verschiedenen Distributionen befinden sich in der entsprechenden Dokumentation der Distribution.

Update des CONZEPT 16-Servers

Zum Update des CONZEPT 16-Servers wird dieser zunächst beendet. Dann können die neuen Programmdateien in das Verzeichnis des Servers entpackt und der Server in der neuen Version gestartet werden.

Kontakt

CONZEPT 16-Server - Installation der Hot-Standby Option

Beschreibung zur Installation der Hot-Standby Option des CONZEPT 16-Servers

Serversysteme und Netzanbindung

Für die CONZEPT 16 Hot-Standby Option werden zwei CONZEPT 16-Datenbank-Server benötigt. Diese werden über eine direkte Gigabit-Ethernet Verbindung oder vergleichbarem miteinander verbunden. Zusätzlich müssen diese Rechner ebenfalls vom Netzwerk aus erreicht werden können. In jedem Rechner müssen also mindestens zwei Netzwerkanschlüsse installiert sein, einen zur Kommunikation mit dem Netzwerk und einen zur Kommunikation mit dem anderen Serversystem. Weitere Hardware wird für den Hot-Standby Betrieb nicht benötigt.

Arbeitet eines der Systeme zusätzlich als Fileserver, ist zu beachten, dass beim Ausfall des Systems die Dateien des CONZEPT 16-Clients nicht mehr zur Verfügung stehen. Das kann vermieden werden, wenn der CONZEPT 16-Client auf den lokalen Festplatten der Clientsysteme abgelegt wird. Ein Versionsabgleich könnte dann zum Beispiel durch eine Verteilungssoftware oder Skript bei der Anmeldung vorgenommen werden (siehe auch FAQ).

Die Systeme können sich von der Leistungsfähigkeit und sogar vom verwendeten Betriebssystem unterscheiden. Im Normalbetrieb wird das Sekundärsystem zwar wesentlich geringer belastet als das Primärsystem, allerdings muss es bei einem Ausfall des Primärsystems in der Lage sein, die anfallenden Benutzeranfragen und Transaktionen ausreichend schnell bearbeiten zu können. Auf dem Sekundärsystem können neben dem CONZEPT 16-Server natürlich auch weitere Dienste (z. B. Druckdienste oder Job-Server) betrieben werden.

Die Hot-Standby-Option funktioniert nur, wenn das Sekundärsystem die gleichen Voraussetzungen für den Einsatz der verwendeten Lizenz erfüllt, wie das Primärsystem. Die Voraussetzungen beziehen sich auf die Anzahl der logischen Prozessoren, das Betriebssystem und ob es sich um ein 64-Bit System handelt. Hot-Standby kann zum Beispiel nicht eingesetzt werden, wenn das Primärsystem unter einem Windows-Betriebssystem läuft und das Sekundärsystem unter Linux, wenn eine Standard-Edition verwendet wird (Diese läuft nur unter Windows oder Linux). Beim Einsatz einer Advanced-Edition gibt es keine Probleme, da diese Lizenz unter Windows und Linux betrieben werden kann.

Datenbankserver

Auf beiden Rechnern muss der CONZEPT 16-Server in der gleichen Version inklusive Lizenzdatei installiert werden. Bei einer Lizenz mit Hardwareschutz muss der Dongle-Treiber mindestens auf dem Primärsystem installiert sein. Auf dem sekundären Server wird der Dongle-Treiber nur benötigt, um bei einem längeren Ausfall des Primärsystems gegebenenfalls ein Umstecken des Dongles zu ermöglichen. Bei einer Lizenz mit Softwareschutz darf sich die Lizenzidentitätsdatei (*.idn) nur auf dem Primärsystem befinden. Ohne ein Umstecken des Dongles bzw. Kopieren der Lizenzidentitätsdatei können alle Hot-Standby Datenbanken auf dem Sekundärsystem noch weitere 60 Tage geöffnet werden. Die Zeitspanne wird ab dem letzten Kontakt, durch Aktualisierung der Datenbank, mit dem Primärsystem gemessen und ist unabhängig davon, ob der Server in der Zwischenzeit neu gestartet wurde. Danach ist der Login für neue Clients gesperrt.

Kontakt

 Während dieser Zeit dürfen die Einstellungen der Datenbank (Name, Datenbank und die Adresse des Primärsystems (Serveradresse)) nicht verändert werden. Zum Starten des Systems muss der Dongle (bei Verwendung einer Lizenz mit Hardwareschutz) beim Primärsystem aufgesteckt werden. Das Sekundärsystem wird ohne eine Lizenz gestartet.

 Wird der CONZEPT 16-Server mit Clients der Versionen 4.7 oder 5.0 betrieben, können Datenbanken nur solange geöffnet werden, wie der CONZEPT 16-Server nicht neu gestartet wurde und weniger als 60 Tage seit der letzten Aktualisierung durch das Primärsystem vergangen sind. Bei einem Neustart des Systems muss der Dongle auf das Sekundärsystem gesteckt oder die Lizenzidentitätsdatei auf das Sekundärsystem kopiert werden, damit sich Clients der Versionen 4.7 oder 5.0 mit einer Datenbank verbinden können.

Auf dem Sekundärserver kann eine zusätzliche Serverlizenz eingerichtet werden. So können auch Datenbanken geöffnet werden, die nicht für den Hot-Standby Betrieb eingerichtet wurden. Falls die 60-Tage Frist einer Hot-Standby Datenbank nach einem Primärserverausfall verstrichen ist, wird dann auch der Login nicht gesperrt, sondern die Lizenz des Sekundärservers verwendet. Dabei ist darauf zu achten, dass es nach Ablauf der primären Lizenz zu Einschränkungen in der Anzahl der Benutzer beziehungsweise des Funktionsumfangs kommen kann, wenn unterschiedliche Serverlizenzen verwendet werden. Die Lizenz des Sekundärservers benötigt dazu keine Hot-Standby-Option.

Besitzt der Sekundärserver **keine** eigene Lizenz, sollte in der Konfiguration des Servers die Option Hot-Standby-Betrieb ohne Dongle aktiviert werden.

 Um aufgetretene Probleme rechtzeitig zu erfahren, sollte unbedingt die automatische Mailbenachrichtigung (siehe Konfiguration des Servers) beim Primär- und beim Sekundärserver eingerichtet werden. Da die Benutzer nach einer kurzen Unterbrechung ihre Arbeit mit dem System fortsetzen können, ist in der Regel nicht damit zu rechnen, dass der Ausfall an den Systemadministrator gemeldet wird. Das Benachrichtigungssystem des CONZEPT 16-Servers informiert dann in jedem Fall umgehend von dem Wechsel zum Sekundärsystem.

Einstellungen der Datenbanken

Die Einstellungen der Datenbanken erfolgt über die Web-Administration des Servers. Einstellungen können auch über ältere Clients (ab Version 5.1) vorgenommen werden. Es stehen dann aber nicht alle Einstellungen zur Verfügung. Ältere Versionen bekommen von dem Server keine Lizenz und können nicht gestartet werden. Im laufenden Betrieb wird eine Lizenz für die älteren Versionen vom Primärserver auf den Sekundärserver übertragen. Die Lizenz befindet sich im Hauptspeicher des Sekundärservers.

Die Datenbank muss auf beiden Serversystemen eingetragen werden. Der symbolische Name der Datenbank muss auf beiden Systemen gleich sein. Alle anderen Eintragungen können abweichen. Die physikalische Datenbankdatei darf nur auf dem Primärsystem abgelegt werden. Die Datenbank wird beim ersten Öffnen automatisch auf den Sekundärserver übertragen.

Kontakt

i Im laufenden Hot-Standby-Betrieb empfängt der Sekundärserver Datenbankupdates vom Primärserver und puffert diese im Datenbankcache. Von dort werden sie in die Datenbank übertragen. Ist das sekundäre System zu stark ausgelastet und kann neue Transaktionen nicht aus dem Cache in die Datenbank schreiben, kann es zu einer Trennung der Verbindung kommen. Diese wird automatisch durchgeführt, wenn der Cache des sekundären Systems mehr als eine Minute zu über 90 Prozent belegt ist (Siehe auch Database HSB Update Failure in der Log-Datei des Datenbankservers). Da ein zu klein konfigurierter Datenbankcache auf dem sekundären System praktisch sofort zu diesem Zustand führen kann, ist unbedingt darauf zu achten, dessen Größe korrekt einzustellen.

Die Einstellungen für den Betrieb mit der Hot-Standby Option werden über die Administration des Servers vorgenommen. Die Einstellungen werden nach der Auswahl der Datenbank über den Link Hot-Standby aufgerufen. Optional kann auch die Datenraumtabellen des Datenbankservers direkt editiert werden.

Die folgenden Einstellungen können für den Hot-Standby-Betrieb vorgenommen werden:

The screenshot shows the CONZEPT 16 software interface for managing databases. The title bar reads "CONZEPT 16 - Datenbank & mehr". The address bar shows the URL "127.0.0.1:4745/area/hotstandby?ca1=CodeLibrary". The main menu has tabs for "Status", "Datenbank" (selected), and "Konfiguration". Below these are sub-tabs: "Manager", "Einstellungen" (selected), "Backup", "Hot-Standby" (selected), "Service", and "Statistik". A sidebar icon shows a server with a 'C' inside a red diamond. The main content area is titled "Hot-StandBy Einstellungen für 'CodeLibrary'". It contains several configuration items with checkboxes:

Serveradresse	192.168.0.2;10.1.3.91
Sekundärserver	<input type="checkbox"/>
Automatische Aktivierung	<input type="checkbox"/>
Automatische Synchronisation	<input checked="" type="checkbox"/>
Timeout Verbindungsaufbau [Sekunden]	3
Verbindungs-Intervall [Minuten]	5
Automatische Abschaltung bei Master-Master Betrieb	<input checked="" type="checkbox"/>
Login sperren wenn Standby-Server nicht mehr verfügbar	<input type="checkbox"/>

A green checkmark icon with the text "Übernehmen" is located at the bottom right of the configuration area.

• Serveradresse

Kontakt

Hier wird die IP-Adresse des anderen Servers angegeben. Erfolgt die Eintragung beim Primärserver, wird hier die IP-Adresse des Sekundärservers angegeben. In der Eintragung des Sekundärservers wird die IP-Adresse des Primärserver eingetragen. Bei diesen IP-Adressen handelt es sich um die Adressen der Netzwerkkarten der Direktverbindung. Es sollten keine Namen anstelle der numerischen IP-Adressen verwendet werden, da sonst im Falle von Problemen mit der Namensauflösung keine Verbindung zwischen den Servern hergestellt werden kann.

Die IP-Adresse der "normalen" Netzwerkverbindung muss nach einem Semikolon ebenfalls angegeben werden. Ist diese Adresse nicht vorhanden, können keine Statusinformationen über die normale Netzverbindung ausgetauscht werden. Bei einem Ausfall der Direktverbindung kann es dann vorkommen, dass ein gleichzeitiges Anmelden an Primär- und Sekundär-Server möglich ist. Durch die Angabe der "normalen" Netzwerkverbindung können Statusinformationen zwischen den Servern ausgetauscht werden und ein gleichzeitiges Anmelden an beide Datenbanken ist ausgeschlossen. Zudem wird die Erreichbarkeit des Sekundärsystems für die Clients durch das Primärsystem überprüft. Ist das Sekundärsystem nicht für die Clients erreichbar, zum Beispiel wegen Einstellungen der Firewall, wird eine entsprechende Warnung in die Protokolldatei geschrieben.



Die IP-Adressen der Direktverbindung und die IP-Adressen der "normalen" Netzwerkverbindung müssen in unterschiedlichen Netzen liegen.

• **Sekundärsystem**

Mit dieser Einstellung wird die Unterscheidung der beiden Server in "Primärsystem" und "Sekundärsystem" vorgenommen. Die Einstellung muss daher auf beiden Servern unterschiedlich sein. Die Einstellung muss nach der Installation nicht mehr verändert werden, da die Rolle des jeweiligen Servers ("Aktiv" oder "Standby") davon unabhängig ist.

• **Automatische Aktivierung**

Ist diese Option nicht gesetzt, wird durch eine Sperre verhindert, dass sich Benutzer direkt an die Datenbank anmelden können, wenn der aktive Server ausgefallen ist (siehe [Einrichten von Sperren](#)).

• **Automatische Synchronisation**

Ist diese Option nicht gesetzt, wird durch eine Sperre verhindert, dass die Datenbank nach einem Serverausfall automatisch wieder mit der aktiven Datenbank synchronisiert wird (siehe [Einrichten von Sperren](#)). Scheitert eine Synchronisation an dieser Sperre, wird zusätzlich die Login-Sperre gesetzt und ein entsprechender Eintrag in die Protokolldatei der Datenbank vorgenommen. Die Sperre muss durch den Administrator über die Funktion [Freigabe für Login](#) entfernt werden.

• **Timeout Verbindungsaufbau**

Beim Öffnen der Datenbank versucht der aktive Server eine Verbindung zum Standby-Server aufzubauen. Gelingt dies nicht in der hier angegebenen Zeitspanne, wird die Datenbank auf dem Standby-Server nicht aktualisiert. Der

Kontakt

nächste Verbindungsauftakt wird nach der Zeit, die in dem Feld "Verbindungs-Intervall" angegeben ist, initiiert. Der Timeout kann zwischen 3 Sekunden und 300 Sekunden (5 Minuten) betragen.

Ist der Standby-Server nicht mehr erreichbar, verzögert sich die Anmeldung des ersten Benutzers um den hier eingetragenen Wert. Die Ausführung von Serviceoperationen verzögert sich ebenfalls um diesen Wert.

Empfohlener Wert: 3 Sekunden.

• **Verbindungs-Intervall**

Kommt es beim Öffnen einer Datenbank nicht zu einer Verbindung zum Standby-Server, wird in regelmäßigen Abständen versucht diese Verbindung erneut herzustellen. Es können hier Wartezeiten zwischen 1 Minute und 15 Minuten eingestellt werden.

Falls die Verbindung zum Standby-Server abbricht, werden erneute Verbindungsversuche auch in diesen Zeitabständen durchgeführt.

Empfohlener Wert: 5 Minuten.

• **Automatische Abschaltung bei Master-Master Betrieb**

Bei eingerichtetem Hot-Standby-Betrieb dürfen sich nie beide Datenbanken im gleichzeitig Master-Modus befinden, da sich in diesem Zustand Clients auf beiden Datenbanken arbeiten können und dadurch später Datenverluste entstehen.

Erkennt einer der beiden beteiligten Datenbankserver einen solchen Zustand und ist diese Option gesetzt, werden beide Datenbank geschlossen, die Benutzer ausgeloggt und die Datenbanken mit einer Login- und einer Synchronisations-Sperre versehen. In den Protokollen der Datenbanken wird der Fehler Database shutdown requested due to HSB master-master conflict eingetragen.

Um wieder mit der Datenbank arbeiten zu können, muss der Administrator anhand der Protokolle entscheiden, welches System sich fälschlicherweise im Master-Modus befand. Diese Datenbank muss umbenannt oder verschoben werden. Bei der richtigen Datenbank muss danach eine "Freigabe für Login" erfolgen.

Ein Master-Master-Betrieb kann beispielsweise auftreten, wenn die aktive Datenbank (Master) auf einer virtuellen Maschine läuft, die eingefroren wird. Übernimmt in diesem Fall das Standby-System die Masterrolle, sind nach dem Aufwecken des virtuellen Systems beide Datenbanken im Master-Modus.

• **Login sperren wenn Standby-Server nicht mehr verfügbar**

Wenn diese Option eingeschaltet ist, wird bei aktiven Hot-Standby-Betrieb eine bedingte Sperre in der Master-Datenbank eingetragen.

Diese Sperre verhindert das Öffnen der Datenbank im Master-Betrieb, wenn das Standby-System nicht erreichbar ist. Die Sperre kann über die Weboberfläche

Kontakt

des Servers angezeigt und auch entfernt werden.

Im normalen Betrieb hat die Sperre keine Auswirkungen, sie wird auch nicht auf die Standby-Datenbank übertragen. Die Sperre ist nur in folgendem Szenario wirksam:

Nach erfolgreichem Hot-Standby-Betrieb wird die Master-Datenbank geschlossen. Ist beim erneuten Öffnen der Datenbank der Standby-Server nicht verfügbar, kann sich kein Benutzer an der Datenbank anmelden. Sobald der Standby-Server wieder funktioniert, können sich wieder Benutzer an der Master-Datenbank anmelden. Fallen nacheinander zunächst der Primär- und anschließend der Sekundärserver aus, können sich erst dann wieder Benutzer an der Datenbank anmelden, wenn beide Server wieder gestartet sind und so die tatsächliche Master-Datenbank ermittelt werden kann.

Bei längerer Nichtverfügbarkeit des Standby-Systems muss die Sperre über die Weboberfläche des Masters entfernt werden. Die Sperre wird automatisch unwirksam, wenn die HSB-Adresse bei der Masterdatenbank entfernt wird und somit der Hot-Standby-Betrieb deaktiviert wird.



Für die Verbindungen des Clients zum Server kann eine Maximalzeit eingestellt werden, die der Client beim Verbindungsaufbau zum Server warten darf. Danach wird automatisch der Sekundärserver angesprochen. Weiteres dazu unter [Konfigurationsdatei / TcpConnectTimeout](#).

Beispiel zur Konfiguration der Hot-Standby Systeme

Die Netzwerkkarten der Direktverbindung zwischen den beiden Serversystemen verwenden in dieser Konfiguration die Adressen 192.168.0.1 und 192.168.0.2. Die Adressen zur Kommunikation mit dem Netzwerk über die beiden Zweitnetzwerkkarten könnten dann beispielsweise 10.0.0.1 und 10.0.0.2 lauten.

Konfiguration des Primärservers

Datenbankname	Beispieldatenbank
Speicherort	c:\c16\ca1\Examples-1
Adresse des anderen Servers	192.168.0.2;10.0.0.2
Sekundärsystem	deaktiviert
Automatische Aktivierung	deaktiviert
Automatische Synchronisation	aktiviert
Timeout Verbindungsaufbau	3 Sekunden
Verbindungs-Intervall	5 Minuten
Automatische Abschaltung bei Master-Master Betrieb	aktiviert
Login sperren wenn Standby-Server nicht mehr verfügbar	deaktiviert
Konfiguration des Sekundärservers	

Datenbankname	Beispieldatenbank
Speicherort	c:\c16\hotstandby\Examples-2
Adresse des anderen Servers	192.168.0.1;10.0.0.1

Kontakt

Sekundärsystem	aktiviert
Automatische Aktivierung	aktiviert
Automatische Synchronisation	aktiviert
Timeout Verbindungsaufbau	3 Sekunden
Verbindungs-Intervall	5 Minuten
Automatische Abschaltung bei Master-Master Betrieb	aktiviert
Login sperren wenn Standby-Server nicht mehr verfügbar	deaktiviert

Aufruf der Datenbank über eine Verknüpfung

Wenn eine Datenbank über eine Verknüpfung geöffnet werden soll, kann generell die Executable des Clients, gefolgt von den Parametern Datenbankserver, Datenbankaliasname, Datenbankbenutzer und Passwort, in der Verknüpfung angegeben werden. So kann der Anwender direkt in die für ihn vorgesehene Umgebung einsteigen. Beim Hot-Standby Betrieb würde dies aber bedeuten, dass der Client nur versuchen würde den Primärserver zu erreichen. Daher können als Datenbankserver zwei verschiedene Systeme, getrennt von einem + angegeben werden. Zwischen Primärserver, der Konjunktion und dem Sekundärserver dürfen dabei keine Leerzeichen vorhanden sein. Der Client versucht sich dann zunächst beim ersten angegebenen Server anzumelden. Ist dieser nicht erreichbar oder ist die Datenbank älter als die des anderen Servers, verbindet sich der Client automatisch mit dem zweiten Server.

Beispiel:

```
c:\c16\client\c16_winc.exe dbserver+dbbackup codelibrary
```

oder

```
c:\c16\client\c16_winc.exe 10.0.0.1+10.0.0.2 codelibrary
```

Kontakt

Installation der Dokumentation

Beschreibung der Installation der CONZEPT 16-Dokumentation
Dokumentation

Siehe aus

Netzwerkpfaden

Die Dateien der CONZEPT 16-Dokumentation werden über die CONZEPT 16
Installationsroutine eingerichtet. Die Dokumente befinden sich anschließend im
Installationsverzeichnis unter \Doc.

Dokumentationsunterlagen

c16.chm Dokumentation

c16.rid Seitenzuordnungs-Datei

c16_error.txt Eventuell vorhandenes Fehlerdokument des Versionsstandes

Die Online-Dokumentation zu CONZEPT 16 enthält die Beschreibung aller CONZEPT
16-Programmmodulen. Die Dokumentation kann ohne weiteres unter einem
Windows-Betriebssystem direkt geöffnet werden. Um die kontextsensitive Hilfe beim
CONZEPT 16-Client zu verwenden, werden die Online-Dokumentation c16.chm und
die Datei c16.rid benötigt. Ist eine der beiden Dateien nicht vorhanden, kann die
Dokumentation aus CONZEPT 16 heraus nicht aufgerufen werden.

Der Client sucht diese Dateien in der folgenden Reihenfolge:

1. relativ zum Client-Pfad unter ..\Doc (Client-Pfad = Pfad aus dem der Client
gestartet wurde)
2. Client-Pfad
3. Installationspfad\Doc

Kontakt

Installation des Clients

Beschreibung der Installation des CONZEPT 16-Clients

Der CONZEPT 16-Client (in Advanced- und Standard-Ausführung) wird über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Client.

Dateien des Clients

c16.ca1	Leere Datenbank
c16.hlp	Hilfe-Datei / Nur Standard-Client
c16.tla	Token-Übersetzungstabelle
c16.tlp	Token-Übersetzungstabelle / Nur Standard-Client
c16.xls	Token-Übersetzungstabelle / Nur Standard-Client
c16.utl	OEM-Kit-Unterstützung
c16.vra	Ressourcendatei
c16.vrs	Zeichenketten deutsch / Nur Standard-Client
c16_1250.cpw	Codepage Window 1252 (ANSI)
c16_1250.fon	Font für Codepage Windows 1252 (ANSI)
c16_1254.cpw	Codepage Window 1254
c16_1254.fon	Font für Codepage Windows 1254
c16_apgi.exe	Advanced-Client (32-Bit)
c16_apgi_w64.exe	Advanced-Client (64-Bit)
c16_bar2d_w32.dll	Barcode-Bibliothek (32-Bit)
c16_bar2d_w64.dll	Barcode-Bibliothek (64-Bit)
c16_chart_w32.dll	Chart-Bibliothek (32-Bit)
c16_chart_w64.dll	Chart-Bibliothek (64-Bit)
c16_coded_w32.dll	CodeEdit-Bibliothek (32-Bit)
c16_coded_w64.dll	CodeEdit-Bibliothek (64-Bit)
c16_complay_w32.dll	Compatibility Layer (32-Bit) / Nur Standard-Client
c16_ctxoffice.ocx	CtxOffice-Steuerelement (32-Bit)
c16_ctxoffice_w64.ocx	CtxOffice-Steuerelement (64-Bit)
c16_ctxreg.exe	CTX-Registration (32-Bit)
c16_ctxreg_w64.exe	CTX-Registration (64-Bit)
c16_debg.dll	Debug Message Bibliothek
c16_debg.exe	Debugger (32-Bit)
c16_debg_w64.exe	Debugger (64-Bit)
c16_diff_w32.dll	Difference-Bibliothek (32-Bit)
c16_diff_w64.dll	Difference-Bibliothek (64-Bit)
c16_dlg_w32.dll	Dialog Designer (32-Bit)
c16_dlg_w64.dll	Dialog Designer (64-Bit)
c16_dlx_w32.dll	Dialog Designer Extension (32-Bit)
c16_dlx_w64.dll	Dialog Designer Extension (64-Bit)
c16_en.vrs	Zeichenketten englisch / Nur Standard-Client

Kontakt

c16_graph_w32.dll	Graphics-Bibliothek (32-Bit)
c16_graph_w64.dll	Graphics-Bibliothek (64-Bit)
c16_icu_w32.dll	Unicode-Bibliothek (32-Bit)
c16_icu_w64.dll	Unicode-Bibliothek (64-Bit)
c16_img_w32.dll	Bildanzeige-Bibliothek (32-Bit) / Nur Standard-Client
c16_obj_w32.dll	GUI-Objekte (32-Bit)
c16_obj_w64.dll	GUI-Objekte (64-Bit)
c16_pdf_w32.dll	PDF-Bibliothek (32-Bit)
c16_pdf_w64.dll	PDF-Bibliothek (64-Bit)
c16_pfdx_w32.dll	PDF-Anzeige-Bibliothek (32-Bit)
c16_pfdx_w64.dll	PDF-Anzeige-Bibliothek (64-Bit)
c16_prt_w32.dll	Graphischer Druckertreiber (32-Bit) / Nur Standard-Client
c16_res_w32.dll	GUI-Ressourcen (32-Bit)
c16_res_w64.dll	GUI-Ressourcen (64-Bit)
c16_ssl_w32.dll	SSL-Bibliothek (32-Bit)
c16_ssl_w64.dll	SSL-Bibliothek (64-Bit)
c16_sys_w32.dll	System-Objekte (32-Bit)
c16_sys_w64.dll	System-Objekte (64-Bit)
c16_tool.utl	Lizenzdatei für Toolbars / Nur Standard-Client
c16_tool_w32.dll	Toolbars (32-Bit) / Nur Standard-Client
c16_tpi_w32.dll	TAPI-Bibliothek (32-Bit) / Nur Standard-Client
c16_tr.vrs	Zeichenketten türkisch / Nur Standard-Client
c16_win.fon	Schriftart / Nur Standard-Client
c16_winc.exe	Standard-Client
c16_xml_w32.dll	XML-Bibliothek (32-Bit)
c16_xml_w64.dll	XML-Bibliothek (64-Bit)
c16_zip_w32.dll	ZIP-Bibliothek (32-Bit)
c16_zip_w64.dll	ZIP-Bibliothek (64-Bit)
\example	Beispielkonfigurationsdateien
c16.cfg	Konfigurationsdatei für den Standard-Client
c16_apgi.cfg	Konfigurationsdatei für den Advanced-Client

Splash-Screen beim Starten des Clients

Beim Start von CONZEPT 16 wird ein Splash-Screen angezeigt. Dieser Splash-Screen kann durch ein eigenes Bild ersetzt werden. Das Bild muss den Namen "C16_TOOL.*" besitzen und sich im Verzeichnis des Clients befinden. Es muss 450 Pixel breit und 250 Pixel hoch sein und kann in den Formaten BMP, JPG und TIF vorliegen. Unterhalb des Bildes bleibt ein Streifen für Statusmeldungen beim Start des Clients frei.

Beispiel:

My Splash-Screen

Das Bild kann mit der rechten Maustaste angeklickt und mit "Speichern unter" unter dem Namen c16_tool.bmp im Client-Verzeichnis gespeichert werden. Es wird dann beim Starten des Clients angezeigt.

Soll beim Start des Clients kein Splash-Screen angezeigt werden, muss in der Konfigurationsdatei des Clients der Eintrag SplashOn = N eingetragen werden.

Kontakt

Installation des Debuggers

Beschreibung der Installation des CONZEPT 16-Debuggers

Die Programmdateien des externen Debuggers befinden sich im Programmstand des CONZEPT 16-Clients. Es müssen keine weiteren Anpassungen vorgenommen werden. Folgende Dateien werden dabei für den Debugger benötigt:

Dateien des Debuggers

c16_chart_w32.dll	Chart-Bibliothek (32-Bit)
c16_chart_w64.dll	Chart-Bibliothek (64-Bit)
c16_coded_w32.dll	CodeEdit-Bibliothek (32-Bit)
c16_coded_w64.dll	CodeEdit-Bibliothek (64-Bit)
c16_debg.dll	Debug Message Bibliothek
c16_debg.exe	Debugger (32-Bit)
c16_debg_w64.exe	Debugger (64-Bit)
c16_diff_w32.dll	Difference-Bibliothek (32-Bit)
c16_diff_w64.dll	Difference-Bibliothek (64-Bit)
c16_graph_w32.dll	Graphics-Bibliothek (32-Bit)
c16_graph_w64.dll	Graphics-Bibliothek (64-Bit)
c16_icu_w32.dll	Unicode-Bibliothek (32-Bit)
c16_icu_w64.dll	Unicode-Bibliothek (64-Bit)
c16_obj_w32.dll	GUI-Objekte (32-Bit)
c16_obj_w64.dll	GUI-Objekte (64-Bit)
c16_pdf_w32.dll	PDF-Bibliothek (32-Bit)
c16_pdf_w64.dll	PDF-Bibliothek (64-Bit)
c16_res_w32.dll	GUI-Ressourcen (32-Bit)
c16_res_w64.dll	GUI-Ressourcen (64-Bit)
c16_ssl_w32.dll	SSL-Bibliothek (32-Bit)
c16_ssl_w64.dll	SSL-Bibliothek (64-Bit)
c16_sys_w32.dll	System-Objekte (32-Bit)
c16_sys_w64.dll	System-Objekte (64-Bit)
c16_xml_w32.dll	XML-Bibliothek (32-Bit)
c16_xml_w64.dll	XML-Bibliothek (64-Bit)
c16_zip_w32.dll	ZIP-Bibliothek (32-Bit)
c16_zip_w64.dll	ZIP-Bibliothek (64-Bit)



Falls sich der Debugger nicht auf dem gleichen Rechner wie der verwendete CONZEPT 16-Client befindet, muss der TCP/IP-Port 4721 des Debugger-Systems für eingehende Verbindungen geöffnet sein.

Kontakt

Installation des Druckertreibers

Beschreibung der Installation des CONZEPT 16-Druckertreibers

Der Komponenten des CONZEPT 16-Druckertreibers werden über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Printer.

-  Bei der Installation des Druckprozessors wird die Druckwarteschlange des Betriebssystems gestoppt und anschließend wieder gestartet. Bestehen Abhängigkeiten zu anderen Diensten, kann die Druckwarteschlange unter Umständen nicht korrekt beendet oder neugestartet werden. In diesem Fall wird eine entsprechende Dialogbox ausgegeben. Die Abhängigkeiten der Druckerwarteschlange können dann manuell überprüft und die entsprechenden Dienste manuell angehalten und später wieder gestartet werden.

Druckertreiber

Die Druckertreiber umfassen die zu installierenden Druckertreiber zum Generieren von PDF- und TIFF-Dokumenten und den sogenannten Port-Monitor, der für die Weiterleitung der Druckjobs an den Druckprozessor verantwortlich ist.

-  Die zu installierenden Druckertreiber können über die Konfigurationsdatei c16_setup_printer.cfg definiert werden, die vor dem Start der Installation in das Verzeichnis der Installationsroutine kopiert wird. Um bereits installierte Konfigurationsdateien zu überschreiben, muss die Installationsroutine mit dem Parameter /OVERWRITE angegeben werden. Für die Installation von Druckertreibern für den Einsatz als Netzwerkdrucker siehe Netzwerkdrucker.

Dateien der Druckertreiber

\driver	Dateien der Druckertreiber
c16_infsetup64.dll	
c16_prtdrv.inf	
c16_prtmoncfg.dll	
c16_prtmoncom32.dll	
c16_prtmoncom64.dll	
c16_prtmondrv32.dll	
c16_prtmondrv64.dll	
c16_prtpdf32.dll	
c16_prtpdf32.ini	
c16_prtpdf64.dll	
c16_prtpdf64.ini	
c16_prtpdf.ppd	
c16_prttif32.dll	
c16_prttif32.ini	
c16_prttif64.dll	
c16_prttif64.ini	
c16_prttif.gpd	

Kontakt

\example	Beispielkonfigurationsdatei für die einzurichtenden Druckertreiber
c16_setup_printer.cfg	

Druckprozessor

Der Druckprozessor führt die Verarbeitung der Druckjobs durch. Die Installationsroutine richtet den Druckprozessor als Dienst ein.

-  Der Druckprozessor entnimmt seine Konfiguration beim Start des Dienstes aus der Konfigurationsdatei des Druckprozessors. Diese kann vordefiniert werden. Sie wird hierzu vor dem Start der Installation in das Verzeichnis der Installationsroutine kopiert.
-  Damit der Druckprozessor Druckaufträge von anderen Rechnern empfangen kann, muss der TCP/IP-Port 4729 auf dem Druckserver für eingehende Verbindungen geöffnet sein.

Dateien des Druckprozessors

\service	Dateien des Druckprozessors
c16.tla	Token-Übersetzungstabelle
c16.vra	Ressourcendatei
c16_bar2d_w32.dll	Barcode-Bibliothek (32-Bit)
c16_chart_w32.dll	Chart-Bibliothek (32-Bit)
c16_coded_w32.dll	CodeEdit-Bibliothek (32-Bit)
c16_diff_w32.dll	Difference-Bibliothek (32-Bit)
c16_graph_w32.dll	Graphics-Bibliothek (32-Bit)
c16_icu_w32.dll	Unicode-Bibliothek (32-Bit)
c16_obj_w32.dll	GUI-Objekte (32-Bit)
c16_pdf_w32.dll	PDF-Bibliothek (32-Bit)
c16_pfdx_w32.dll	PDF-Anzeige-Bibliothek (32-Bit)
c16_ppcsvc.exe	Druckprozessor
c16_res_w32.dll	GUI-Ressourcen (32-Bit)
c16_ssl_w32.dll	SSL-Bibliothek (32-Bit)
c16_sys_w32.dll	System-Objekte (32-Bit)
c16_xml_w32.dll	XML-Bibliothek (32-Bit)
c16_zip_w32.dll	ZIP-Bibliothek (32-Bit)

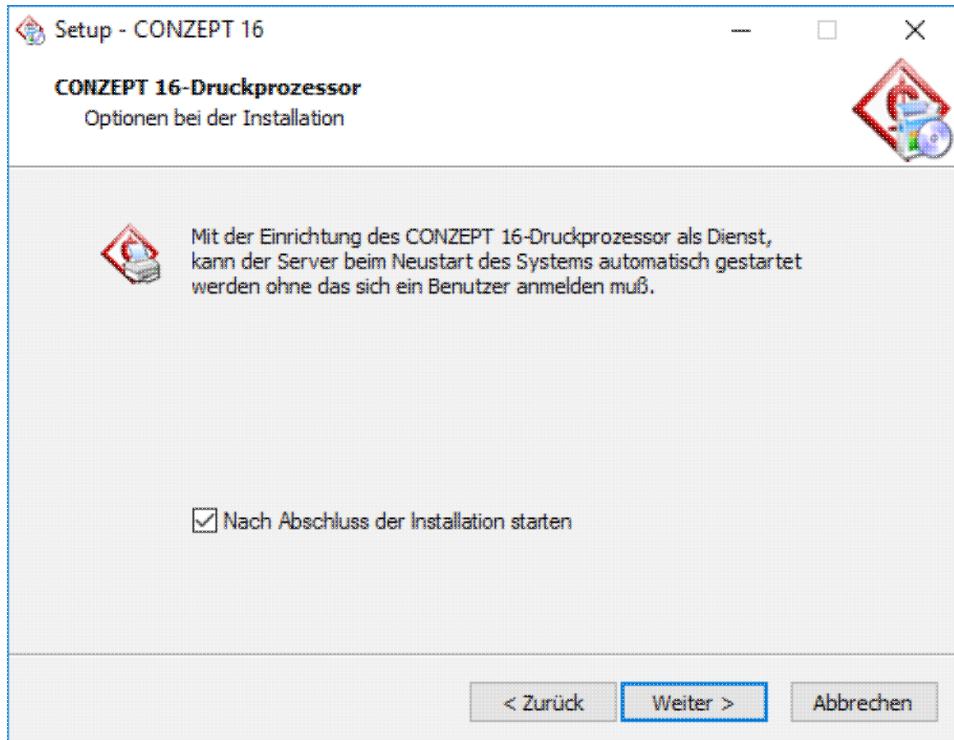
\example

c16_ppcsvc.cfg	Beispielkonfigurationsdatei des Druckerprozessors
----------------	---

Installationsverlauf

Der Druckprozessor kann nach dem Installationsvorgang automatisch gestartet werden. Über die Installationsroutine wird dies durch das Aktivieren der folgenden Option erreicht. Diese wird allerdings nur dann berücksichtigt, wenn eine Konfigurationsdatei für den Dienst vorhanden ist.

Kontakt



Der Druckprozessor ist in der Lage PDF-Dateien zu erzeugen. Dazu benötigt er entweder den Acrobat Distiller oder das frei verfügbare Ghostscript. Ist Ghostscript auf dem Zielsystem nicht eingerichtet, wird vor dem Start der Installation noch ein weiterer Dialog angezeigt. Hier kann der Download von Ghostscript gestartet werden.

Befindet sich die Ghostscript-Installationsdatei gs*w***.exe im gleichen Verzeichnis wie die Installationsroutine, kann mit dem Aktivieren der Option Ghostscript auch gleich mitinstalliert werden. In diesem Fall wird hinter der Option der Text "Ghostscript installieren" angezeigt. Die Installation von Ghostscript erfolgt unabhängig von der Installation des CONZEPT 16-Druckertreibers. Bei der Installation müssen alle Komponenten von Ghostscript installiert werden (entspricht der Standardeinstellung der Installationsroutine von Ghostscript).

Kontakt

Installation der externen Windows Programmierschnittstelle

Beschreibung der Installation der externen Windows Programmierschnittstelle von CONZEPT 16

Die externe Windows Programmierschnittstelle wird über die CONZEPT 16

Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Api.

Installationsverlauf

Bei der Installation werden lediglich die entsprechenden Dateien in das Installationsverzeichnis kopiert.

Programmdateien

Dateien der Programmierschnittstelle

c16.tla	Token-Übersetzungstabelle
c16.vra	Ressourcendatei
c16_bar2d_w32.dll	Barcode-Bibliothek (32-Bit)
c16_chart_w32.dll	Chart-Bibliothek (32-Bit)
c16_coded_w32.dll	CodeEdit-Bibliothek (32-Bit)
c16_diff_w32.dll	Difference-Bibliothek (32-Bit)
c16_graph_w32.dll	Graphics-Bibliothek (32-Bit)
c16_icu_w32.dll	Unicode-Bibliothek (32-Bit)
c16_icu_w64.dll	Unicode-Bibliothek (64-Bit)
c16_obj_w32.dll	GUI-Objekte (32-Bit)
c16_pdf_w32.dll	PDF-Bibliothek (32-Bit)
c16_pfdx_w32.dll	PDF-Anzeige-Bibliothek (32-Bit)
c16_pgx_w64.dll	Programmierschnittstelle (64-Bit)
c16_pgxe.dll	Erweiterte Programmierschnittstelle (32-Bit)
c16_pgxw.dll	Programmierschnittstelle (32-Bit)
c16_res_w32.dll	GUI-Ressourcen (32-Bit)
c16_ssl_w32.dll	SSL-Bibliothek (32-Bit)
c16_ssl_w64.dll	SSL-Bibliothek (64-Bit)
c16_sys_w32.dll	System-Objekte (32-Bit)
c16_xml_w32.dll	XML-Bibliothek (32-Bit)
c16_zip_w32.dll	ZIP-Bibliothek (32-Bit)
\example	Beispiel in Delphi
\include	Header- und Unit-Dateien für C/C++ und Delphi
c16.h	Include-Datei für C/C++-Compiler
c16_dll.h	Include-Datei für C/C++-Compiler (eigene DLL)
c16_pgx.h	Include-Datei für C/C++-Compiler (c16_pgxw.dll / c16_pgx_w64.dll)

Kontakt

\lib	Bibliotheken zum Linken der Programmierschnittstelle
c16_pgx_w64.lib	Import-Bibliothek zum Linken der Schnittstelle für 64-Bit in Microsoft Visual Studio
c16_pgxe.lib	Import-Bibliothek zum Linken der erweiterten Schnittstelle in Microsoft Visual Studio 2005 oder neuer
c16_pgxw.lib	Import-Bibliothek zum Linken der Schnittstelle in Microsoft Visual Studio 2005 oder neuer

Die erweiterte Programmierschnittstelle ermöglicht zusätzlich das Aufrufen von Oberflächenobjekten aus einer CONZEPT 16-Datenbank.

Installation der ODBC-Schnittstelle

Beschreibung der Installation der ODBC-Schnittstelle von CONZEPT 16

Die ODBC-Schnittstelle wird über die CONZEPT 16 Installationsroutine installiert. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Odbc8. Über die Installationsroutine kann der ODBC-Treiber installiert werden. ODBC-Clients können lokal über diesen auf eine CONZEPT 16-Datenbank zugreifen. Diese Installation ist für den normalen Betrieb ausreichend.

Eine Beschreibung des ODBC-Treibers befindet sich im Abschnitt ODBC-Treiber.

Programmdateien

Dateien der ODBC-Schnittstelle

```
\w32\Driver  
c16_odbc_driver.dll  
c16_odbc_driver_messages_en-US.xml  
ODBCMessages_en-US.xml  
simbaicudt53_32.dll  
simbaicuin53_32.dll  
simbaicuuc53_32.dll  
SQLEngineMessages_en-US.xml  
\w64\Driver  
c16_odbc_driver.dll  
c16_odbc_driver_messages_en-US.xml  
ODBCMessages_en-US.xml  
simbaicudt53_64.dll  
simbaicuin53_64.dll  
simbaicuuc53_64.dll  
SQLEngineMessages_en-US.xml
```



Nach der Installation des Programmstandes müssen die CONZEPT 16 Datenquellen, welche vom ODBC-Treiber zur Verfügung gestellt werden sollen, beim Windows ODBC-Treiber Manager bekanntgegeben werden. Siehe dazu ODBC-Datenquellen einrichten.

Kontakt

Installation der Web-Schnittstelle

Beschreibung der Installation der CONZEPT 16-Web-Schnittstelle

Die Dateien der Web-Schnittstelle werden über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Web.

Dateien der Web-Schnittstelle

c16_web_iis7_w32.dll Client-DLL für IIS7 32-Bit (\Inetpub\scripts)

c16_web_iis7_w64.dll Client-DLL für IIS7 64-Bit (\Inetpub\scripts)

\example

Beispielkonfigurationsdateien der Web-Schnittstelle

c16_web.ini

Initialisierungsdatei (\Inetpub\scripts oder \WinNT\System32)

c16_web.cfg

Konfigurationsdatei (\Inetpub\scripts oder beliebiges Verzeichnis)

Die Datei c16_web_iis*.dll ist die eigentliche Web-Schnittstelle. Damit der Server von einem Browser angesprochen werden kann, muss die Schnittstelle in ein Verzeichnis der Website kopiert werden, das über Script-Berechtigung verfügt. In der Standardinstallation ist dies das Verzeichnis \InetPub\scripts. Es muss nur eine der Dateien kopiert werden. Welche der Dateien hängt vom verwendeten Internet Informations Service ab:

Betriebssystem

zu kopierende Datei

Windows 7 (32-Bit) oder höher / IIS Version 7

c16_web_iis7_w32.dll

Windows 7 (64-Bit), Windows Server 2008 R2 oder höher / IIS Version 7

c16_web_iis7_w64.dll

Die Datei sollte nach c16_web.DLL umbenannt werden. Es kann aber auch ein beliebiger anderer Name verwendet werden. Dadurch ändern sich auch die Namen aller anderen Dateien, da der Name bis auf die Dateiendung gleich sein muss.



Die CONZEPT 16-Web-Schnittstelle kann nur zusammen mit den Internet Information Services (ab Version 7) eingesetzt werden.

Die Konfiguration der Web-Schnittstelle wird über Konfigurationsdateien vorgenommen. Nachdem der CONZEPT 16-Server gestartet und die entsprechenden Einstellungen der Web-Schnittstelle getroffen wurden, kann mit einem Browser über die Web-Schnittstelle eine Datenbank angesprochen werden.

Kontakt

Installation des SOA-Services unter Windows

Beschreibung der Installation des SOA-Services unter Windows

Die Dateien der SOA-Service werden über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \SOA.

Dateien des SOA-Service

c16_soa_cmd_win.exe Script-Utility
c16_soa_drv_win.dll Laufwerkstreiber
c16_soa_svc_win.dll System-Dienst
c16_soa_tsk_w32.dll Taskprogramm (32-Bit)
c16_soa_tsk_w64.dll Taskprogramm (64-Bit)
c16_soa_tsk_win.dll Taskprogramm (32-Bit mit grafischer Erweiterung)

Zusätzlich benötigte Dateien in diesem Verzeichnis:

c16.tla Token-Übersetzungstabellen
c16.vra Ressourcendatei
c16_bar2d_w32.dll Barcode-Bibliothek (32-Bit)
c16_chart_w32.dll Chart-Bibliothek (32-Bit)
c16_coded_w32.dll CodeEdit-Bibliothek (32-Bit)
c16_diff_w32.dll Difference-Bibliothek (32-Bit)
c16_graph_w32.dll Graphics-Bibliothek (32-Bit)
c16_icu_w32.dll Unicode-Bibliothek (32-Bit)
c16_obj_w32.dll GUI-Objekte (32-Bit)
c16_pdf_w32.dll PDF-Bibliothek (32-Bit)
c16_pfd_w32.dll PDF-Anzeige-Bibliothek (32-Bit)
c16_res_w32.dll GUI-Ressourcen (32-Bit)
c16_ssl_w32.dll SSL-Bibliothek (32-Bit)
c16_ssl_w64.dll SSL-Bibliothek (64-Bit)
c16_sys_w32.dll System-Objekte (32-Bit)
c16_xml_w32.dll XML-Bibliothek (32-Bit)
c16_xml_w64.dll XML-Bibliothek (64-Bit)
c16_zip_w32.dll ZIP-Bibliothek (32-Bit)
c16_zip_w64.dll ZIP-Bibliothek (64-Bit)

Die Konfiguration des SOA-Service wird über Konfigurationsdateien vorgenommen. Nach der Installation ist im Datenverzeichnis von CONZEPT das Unterverzeichnis SOA eingerichtet. In diesem Verzeichnis muss die Konfigurationsdatei des Dienstes als auch die Konfigurationsdateien der einzelnen Tasks stehen.

Nach der Installation kann der SOA-Service Systemdienst als auch die einzelnen Tasks über das Control-Center gestartet und gestoppt werden. Das Control-Center kann auch einzelne Tasks anlegen und entfernen (siehe SOA-Service - Konfigurationsdatei). Alternativ können Tasks über das Script-Utility abgefragt, gestartet und gestoppt werden.

Kontakt

Je nach Betriebssystem und Konfiguration wird eine der c16_soa_tsk_w???.dll-Dateien verwendet:

- c16_soa_tsk_w32.dll - 32-Bit Betriebssystem oder 64bit = N und c16_proc_extended = N
- c16_soa_tsk_w64.dll - 64-Bit Betriebssystem und 64bit = Y
- c16_soa_tsk_win.dll - 32-Bit Betriebssystem oder 64bit = N und c16_proc_extended = Y oder nicht gesetzt

Kommt es zum Absturz des SOA-Services, wird bei den Dateien c16_soa_tsk_w32.dll und c16_soa_tsk_w64.dll eine Minidump-Datei (c16_soa_tsk_w???.mdmp) erzeugt. Diese befindet sich in einem der Verzeichnisse in der folgenden Reihenfolge:

- Pfad, der durch die TMP-Umgebungsvariable definiert ist.
- Pfad, der durch die TEMP-Umgebungsvariable definiert ist.
- Pfad, der durch die USERPROFILE-Umgebungsvariable definiert ist.
- Windows-Verzeichnis.

Kontakt

PHP-Schnittstelle - Installation

Beschreibung der Installation der CONZEPT 16-PHP-Schnittstelle

Die Dateien der PHP Schnittstelle werden über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Php.

Dateien der PHP-Schnittstelle

php5.2_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.2.x)
php5.2ts_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.2.x, thread-sicher)
php5.2_c16_x32.so	PHP-Schnittstelle für Linux-Systeme (PHP 5.2.x)
php5.2_c16_x64.so	PHP-Schnittstelle für Linux 64 Bit-Systeme (PHP 5.2.x)
php5.3_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.3.x)
php5.3ts_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.3.x, thread-sicher)
php5.3_c16_x32.so	PHP-Schnittstelle für Linux-Systeme (PHP 5.3.x)
php5.3_c16_x64.so	PHP-Schnittstelle für Linux 64 Bit-Systeme (PHP 5.3.x)
php5.4_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.4.x)
php5.4ts_c16_x32.dll	PHP-Schnittstelle für Windows-Systeme (PHP 5.4.x, thread-sicher)
php5.4_c16_x32.so	PHP-Schnittstelle für Linux-Systeme (PHP 5.4.x)
php5.4_c16_x64.so	PHP-Schnittstelle für Linux 64 Bit-Systeme (PHP 5.4.x)

\scripts	Verzeichnis für Beispielskripts
c16.css	Beispiel für ein "Cascading Style-Sheet"
c16_call.php	CONZEPT 16-Funktion aufrufen
c16_catalog.php	Daenstruktur auslesen
c16_info.php	Allgemeine Informationen über den Status des PHP-Interpreters
c16_reclink.php	Verknüpfte Datensätze ausgeben
c16_recop.php	Datensatzverarbeitung
c16_selection.php	Selektionen durchführen
def_connect.php	Verbindungseinstellungen zum Datenbank-Server
def_table.php	Funktionen zum Erstellen von Tabellen
php_c16_def.php	Definitionsdatei für Konstanten

 Da derzeit für Windows-Plattformen keine 64-Bit PHP-Interpreter angeboten werden, können dort 32-Bit Interpreter mit der 32-Bit PHP-Schnittstelle von CONZEPT 16 verwendet werden.

Thread-Sicherheit

Neben der PHP-Version, dem Systemtyp und dem Betriebssystem sind PHP-Erweiterungen auch von der Thread-Sicherheit der PHP-Installation abhängig.

Bei der Windows-Version der Web-Umgebung "XAMPP" mit dem Webserver "Apache" wird in der Regel die thread-sichere Variante von PHP verwendet. Der "Internet

Kontakt

Information Services" (IIS) verwendet im Gegensatz dazu üblicherweise die nicht-thread-sichere Variante.

 Je nach verwendeter Umgebung ist unter Windows die thread-sichere oder die nicht-thread-sichere Variante der PHP-Schnittstelle einzusetzen. Die jeweils andere Version kann nicht geladen werden.

Bei der Installation von PHP unter Linux kommt hingegen typischerweise die nicht-thread-sichere Variante zum Einsatz.

Voraussetzungen

Damit der PHP-Interpreter die CONZEPT 16 PHP-Schnittstelle verwenden kann, muss diese in der Datei php.ini mit Name und Pfad eingetragen oder innerhalb eines PHP-Skripts geladen werden.

 Zum Betrieb der PHP-Schnittstelle muss eine funktionsfähige PHP-Skriptsprache der Version 5.2, 5.3 oder 5.4 installiert sein. Niedrigere oder höhere PHP-Releasestände werden nicht unterstützt.

Beispiel für den Eintrag in der php.ini

```
extension=php5.4_c16_x32.dll
```

Unter Linux-Systemen wird eine andere Datei eingebunden:

```
extension=php5.4_c16_x32.so
```

oder

```
extension=php5.4_c16_x64.so
```

Stehen die Extensions nicht im gleichen Verzeichnis, wie der Skript-Interpreter, muss der Eintrag "extension_dir" auf das entsprechende Verzeichnis gesetzt werden.

Beispiel zum Laden der CONZEPT 16-PHP-Schnittstelle innerhalb des PHP-Skripts

Anweisungen für Linux-Betriebssystem

```
<?php if (!extension_loaded('CONZEPT 16')) dl('php5.4_c16_x32.so'); ?> <?php if (!exten  
Temporäre Dateien, die von der PHP-Schnittstelle angelegt werden, werden unter  
Windows im Pfad für temporäre Dateien gespeichert (Systemvariablen TEMP oder  
TMP). Unter Linux werden temporäre Dateien in das Verzeichnis /tmp gespeichert.  
Soll ein anderes Verzeichnis verwendet werden, muss folgender Eintrag in die Datei  
php.ini eingetragen werden:
```

```
[CONZEPT 16]c16.path.temp = <Pfadname>
```

 Nach der Änderung der Datei muss die PHP-Erweiterung neu gestartet werden.

Zur Überprüfung der Installation kann das Programm "php.exe" oder "php-cgi.exe" verwendet werden. Das Programm ist in der Installation von PHP enthalten und stellt den Interpreter der PHP-Skripten dar. Die Installation der PHP-Schnittstelle kann mit

Kontakt

dem Parameter -i überprüft werden. In der ausgegebenen HTML-Seite werden im Abschnitt "CONZEPT 16 " Informationen zur PHP-Schnittstelle angezeigt.

Im normalen Betrieb erwartet der Interpreter eine Datei mit PHP-Code und generiert daraus zum Beispiel eine HTML-Seite. Die gleiche Ausgabe von php.exe -i erhält man, wenn eine Datei mit folgendem Inhalt übergeben wird:

```
<html><head></head><body><?php  phpinfo();?></body></html>
```

Dies ist eine Datei mit einem PHP-Skript. Wird diese Datei mit dem Namen info.php gespeichert, kann sie mit dem PHP-Interpreter aufgerufen und die Ausgabe in eine Datei mit Namen info.htm umgeleitet werden. Dies kann über folgende Anweisung erfolgen: php info.php > info.htm. Die entstandene HTML-Datei kann in einem Browser angezeigt werden.

Eine erzeugtes PHP-Skript kann über die gleiche Vorgehensweise einfach getestet werden.

Kontakt

Kommandozeilen-Parameter und Rückgabewerte der Installationsroutine
Auflistung und Erläuterung der möglichen Kommandozeilen-Parameter sowie der
Rückgabewerte der Installationsroutine

Siehe [Installationsroutine](#)

Die [Installation](#) kann teilweise oder auch ganz ohne einen Eingriff des Benutzers erfolgen. Der Installationsroutine können dazu verschiedene Parameter in der Kommandozeile übergeben werden. Die Parameter werden dabei mit "/" eingeleitet. Die Groß-/Kleinschreibung der Parameter wird nicht berücksichtigt.

```
c16_setup_*.exe [/? | /h | /help] [/lang=de|en] [/silent | /verysilent] [/nodriver]  
[/noicons] [/overwrite] [/nouninstall] [/nomsgbox] [/norestart] [/cpn_action=<action>]  
[/cpn_srv=<srv>] [/cpn_doc=<doc>] [/cpn_cln=<cln>] [/cpn_com=<com>]  
[/cpn_red=<red>] [/chk_off=<chk>] [/dir=<directory>] [/resultfile=<file>]
```

Die Schalter /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com definieren welche Komponenten (in Abhängigkeit von /cpn_action) berücksichtigt werden sollen. Die Komponenten werden als hexadezimal addierte Werte angegeben (siehe Beispiele). Mit /chk_off werden Checkboxen deselekziert und ausgegraut dargestellt. Die Werte für die einzelnen Checkboxen können addiert werden, um mehrere Checkboxen zu deaktivieren.

- **/?, /h oder /help**

Die Online-Hilfe wird entpackt und die Hilfe zu den Übertragungsparametern der Installation werden angezeigt. Es können auch alle anderen Seiten der Online-Hilfe angezeigt werden.

- **/lang=de|en**

Es wird eine Sprache angegeben, in der die Installationsroutine Informationen oder Abfragen anzeigt. Zur Zeit stehen folgende Sprachen zur Auswahl:

de deutsch

en englisch

- **/silent**

Es werden nur die Splash-Screens und die Fortschrittsbalken angezeigt.

- **/verysilent**

Es werden nur die Splash-Screens angezeigt.

- **/nodriver**

Die Installation des Dongle-Treibers wird übergangen.

- **/noicons**

Es werden keine Einträge in das Startmenü vorgenommen.

- **/overwrite**

Die Konfigurationsdateien aus dem Startverzeichnis werden verwendet, unabhängig davon, ob auf dem System bereits Konfigurationsdateien existieren.

- **/nomsgbox**

Kontakt

Zum Abschluss der Deinstallation wird ein Dialogbox angezeigt. Die Anzeige des Dialogs kann mit dieser Option unterdrückt werden.

- **/norestart**

Ist diese Option übergeben worden, wird nicht zu einem Neustart des Computers aufgefordert, falls ein Neustart erforderlich ist. Ein Neustart wird erforderlich, wenn der Laufwerkstreiber installiert, aktualisiert oder deinstalliert wird.

-  Ist die Option /verysilent angegeben, wird automatisch ein Neustart ausgeführt, wenn er erforderlich ist. Mit dieser Option wird auch dann der Neustart unterbunden.

- **/cpn_action=<action>**

Dieser Schalter legt die Art der durchzuführenden Installation fest:

- ◆ **install**

Es wird eine Neuinstallation durchgeführt. Dieser Wert ist nur erlaubt, wenn noch keine Installation vorhanden ist. Über die Schalter /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com müssen die zu installierenden Komponenten angegeben werden. Das Installationsverzeichnis kann in /dir angegeben werden. Ist kein Installationsverzeichnis angegeben, erfolgt die Installation in das Verzeichnis "CONZEPT 16" im Programm-Verzeichnis.

- ◆ **change**

Führt eine Änderung der installierten Komponenten durch. Der Wert ist nur gültig, wenn bereits eine Installation vorhanden ist. Welche Komponenten geändert werden sollen, wird durch die Schalter /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com angegeben.

Wird keiner der Schalter übergeben, wird lediglich die Auswahl-Seite (Ändern, Aktualisieren, Entfernen) übersprungen.

-  Nicht angegebene, jedoch bereits installierte Komponenten werden deinstalliert! Dies kann durch den optionalen Parameter /nouninstall unterbunden werden. Ist dieser angegeben, werden nicht angegebene Komponenten ignoriert.

- ◆ **update**

Führt eine Aktualisierung der installierten Komponenten durch. Der Wert ist nur gültig, wenn bereits eine Installation vorhanden ist. Welche Komponenten aktualisiert werden sollen, wird durch die Schalter /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com angegeben.

Hierbei können natürlich **nur** Komponenten angegeben werden, die auch bereits installiert sind.

Kontakt

Wird keiner der Schalter übergeben, wird die Auswahl-Seite (Ändern, Aktualisieren, Entfernen) übersprungen und alle installierten Komponenten werden aktualisiert.

◆ **remove**

Führt eine Deinstallation der installierten Komponenten durch. Der Wert ist nur gültig, wenn bereits eine Installation vorhanden ist.

Welche Komponenten deinstalliert werden sollen, wird durch die Schalter /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com angegeben.

Wird keiner der Schalter übergeben, wird die Auswahl-Seite (Ändern, Aktualisieren, Entfernen) übersprungen und alle installierten Komponenten entfernt.

◆ **extract**

Extrahiert die Komponenten in das durch /dir=<dir>angegebene Verzeichnis.

Die Komponenten werden durch /cpn_srv, /cpn_doc, /cpn_cln und /cpn_com angegeben. Existiert das angegebene Verzeichnis nicht, wird es erstellt.

• **/cpn_srv=<srv>**



Bei den übergebenen Zahlenwerten handelt es sich um Hexadezimalzahlen. Das wirkt sich besonders bei der Addition von mehreren Optionen aus. Soll zum Beispiel die Web-Schnittstelle und die PHP-Schnittstelle installiert werden, muss der Wert 0000000C angegeben werden.

00000001 Windows Server

00000002 Linux Server

00000004 Control-Center

• **/cpn_doc=<doc>**

00000001 Online-Hilfe

• **/cpn_cln=<cln>**

00000001 Standard- und Advanced-Client

00000002 API-Client

00000004 Web-Schnittstelle

00000008 PHP-Schnittstelle

00000010 Druckertreiber

00000020 Druckprozessor

00000200 SOA-Service (Windows)

00000400 ODBC-Treiber (32-bit)

00000800 ODBC-Treiber (64-bit)

00001000 SOA-Service (Linux)

00010000 Laufwerkstreiber

• **/cpn_com=<com>**

00000001 Gemeinsame Komponenten des CtxDocEdit

00000002 Extern vorliegendes Chromium-Package

• **/cpn_red=<red>**

00000001 Visual C++ Laufzeitkomponenten



Diese Option wird nur bei /cpn_action=extract ausgewertet.

• **/chk_off=<chk>**

00000001 Server als Dienst einrichten

00000002 Server nach Abschluss der Installation starten

00000004 Systemcache-Einstellung anpassen (siehe [Installation des Servers unter Windows](#))

00000100 Control-Center automatisch mit Windows starten

00000200 Control-Center nach Abschluss der Installation starten

Durch Setzen der jeweiligen Flags wird die entsprechende Checkbox deselektiert und ausgegraut.

• **/dir=<directory>**

In dieser Option wird das Verzeichnis angegeben, in das die Programmstände extrahiert werden sollen. Existiert das Verzeichnis nicht, wird es angelegt.

• **/resultfile=<file>**

Mit dieser Option kann der Pfad und der Name einer Resultat-Datei angeben werden. Wird keine Datei angegeben, werden nur die Fehlerwerte 0 bis 5 zurückgegeben. War die Installation erfolgreich, wird **keine** Resultatdatei generiert. Ansonsten wird die Datei <file> mit dem Inhalt ExitCode <Fehlerwert> erzeugt. Der Fehlerwert ist ein numerischer Wert im Bereich 244 bis 255:

244 Upgrade nicht möglich, da installierte Version kleiner 5.6.01.

245 Vollversion ist installiert und Setup der Evaluierungsversion wird ausgeführt.

246 Evaluierungsversion ist installiert und Setup der Vollversion wird ausgeführt.

248 Eine oder mehrere Datenbanken sind noch offen. Rückgabewert tritt nur auf, wenn /silent oder /verysilent übergeben wurde.

249 Fehler beim Upgraden der Version 5.1 auf eine höhere Version.

250 Fehler beim Verschieben der Anwendungsdateien.

251 Fehler beim extrahieren der Dateien.

252 Aufgrund der übergebenen Kommandozeilenargumente hat die Installationsroutine festgestellt, dass keine Änderungen zur vorhandenen Installation vorliegen und die Routine deshalb beendet wurde.

253 Die bereits vorhandene CONZEPT 16-Installation ist neuer als die zu installierende Version.

254 Betriebssystem wird nicht unterstützt.

255 Fehler bei der Übergabe der Kommandozeilenargumente oder ungültige Kombination der Argumente.

Beispiele:

Kontakt

Installation des Windows- und des Linux-Servers, des Control-Centers und der Dokumentation:

```
c16_setup_*.exe /cpn_action=install /cpn_srv=7 /cpn_doc=1
```

Installation des Windows Servers, des Control-Centers, der Dokumentation und des Standard- und Advanced-Clients in dem Verzeichnis "C:\CONZEPT 16":

```
c16_setup_*.exe /verysilent /noicons /cpn_action=install /dir="c:\CONZEPT 16" /cpn_srv=5 /cpn_doc=1 /cpn_cln=1
```

Update des installierten Windows-Servers. Alle anderen Komponenten werden entfernt:

```
c16_setup_*.exe /cpn_action=change /cpn_srv=1
```

Update des installierten Windows-Servers. Alle anderen Komponenten bleiben unverändert:

```
c16_setup_*.exe /cpn_action=change /cpn_srv=1 /nouninstall
```

Update des Druckertreibers (20 hexadezimal), des Druckprozessors (40 hexadezimal) und des ODBC-Treibers (80 hexadezimal, Summe = e0):

```
c16_setup_*.exe /cpn_action=change /cpn_cln=e0 /nouninstall
```

Alle Clients deinstallieren:

```
c16_setup_*.exe /cpn_action=remove /cpn_cln=ffffffff
```

Extrahieren aller Server-Komponenten in das Verzeichnis "c:\temp":

```
c16_setup_*.exe /cpn_action=extract /cpn_srv=ffffffff /dir="C:\temp"
```

Deselektierten und Sperren der Checkboxen "Server als Dienst einrichten" (001), "Server nach Abschluss der Installation starten" (002) und "Control-Center automatisch mit Windows starten" (100, Summe = 103):

```
c16_setup_*.exe /chk_off=103
```

Rückgabewerte der Installationsroutine

Wenn keine Resultatdatei vorliegt, gibt der Errorlevel der Installationsroutine Aufschluss über Erfolg oder Nicht-Erfolg der Installation. Dieser kann bei der Verarbeitung von Batches herangezogen werden.

Errorlevel Beschreibung

- 0 Kein Fehler - Routine erfolgreich durchgeführt.
- 1 Fehler bei der Initialisierung.
- 2

Kontakt

Die Installation wurde durch den Benutzer abgebrochen (bevor die Installation startete bzw. bei der Willkommen-Seite).

3, 4 Fataler Fehler aufgetreten.

5 Die Installation wurde vom Benutzer abgebrochen.

Bei der Deinstallation wird von der Installationsroutine immer der Errorlevel 0 zurückgegeben.

Kontakt

Benötigte Laufzeitkomponenten

Laufzeitkomponenten für den Betrieb der CONZEPT 16-Komponenten

Folgende Laufzeitkomponenten werden zum Betrieb von CONZEPT 16-Komponenten benötigt.

Redistributable	<u>VC 2012 (32-Bit)</u>	<u>VC 2012 (64-Bit)</u>	<u>VC 2013 (32-Bit)</u>	<u>VC 2013 (64-Bit)</u>
------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Komponente

ODBC-Treiber *

Gemeinsame

Komponenten *

 Bei den mit * markierten Komponenten hängt die zu installierende Art der Laufzeitkomponenten davon ab, ob die 32- oder die 64-Bit Version der Komponente verwendet werden soll.

Die Laufzeitkomponenten können über die Installationsroutine mit folgendem Kommando extrahiert, oder von den verlinkten Webseiten heruntergeladen werden:

```
c16_setup_*.exe /cpn_action=extract /cpn_red=1 /dir="C:\temp"
```

Speicherorte von Konfigurationsdateien

Ablagestruktur der CONZEPT 16-Konfigurationsdateien

Die Konfigurationsdateien von CONZEPT 16 werden unter Windows im allgemeinen Verzeichnis für Applikationsdaten, in einem Unterverzeichnis \CONZEPT 16, abgelegt und geladen. Dort befinden sich weitere Unterverzeichnisse für die verschiedenen CONZEPT 16-Komponenten. Wird eine Konfigurationsdatei nicht im Standardverzeichnis gefunden, wird sie aus dem Programmverzeichnis geladen und auch dort wieder geschrieben. Ist sie dort auch nicht vorhanden, wird sie im Standardverzeichnis angelegt. Unter Linux-Systemen befinden sich die Konfigurationsdateien immer direkt im Verzeichnis des jeweiligen Programmstandes.

-  Werden die CONZEPT 16-Komponenten über die Installationsroutine im Programmverzeichnis des Betriebssystems installiert, wirkt sich eine Änderung des Verzeichnisnamens - von \CONZEPT 16 auf eine beliebige andere Bezeichnung - auch auf den Speicherort der Konfigurationsdateien aus. Der Ordner für Konfigurationsdateien lautet dann nicht \CONZEPT 16, sondern entspricht dem neuen Verzeichnisnamen.

Bereits in den Programmverzeichnissen vorhandene Konfigurationsdateien können über das Control-Center aus den Programmpfaden in die entsprechenden Betriebssystempfade kopiert werden. Um Verwechslungen zu vermeiden, sollten die Dateien anschließend aus ihren ursprünglichen Verzeichnissen entfernt werden.

-  In der Regel hat der Nicht-Administrator keine Rechte im Verzeichnis für allgemeine Applikationsdaten. Daher müssen die entsprechenden Rechte den Benutzern eingeräumt werden. Üblicherweise sind diese Verzeichnisse auch versteckt. Das Kopieren der Dateien sollte deshalb über das Control-Center durchgeführt werden.

Standardverzeichnisse für Konfigurationsdateien unter Windows

<i>Betriebssystem</i>	<i>Verzeichnis</i>
Windows Server 2016	C:\ProgramData
Windows 10	C:\ProgramData
Windows Server 2012 R2	C:\ProgramData
Windows 8.1	C:\ProgramData
Windows Server 2012	C:\ProgramData
Windows 8	C:\ProgramData
Windows Server 2008 R2	C:\ProgramData
Windows 7	C:\ProgramData
Windows Server 2008	C:\ProgramData
Windows Vista	C:\ProgramData
Windows 2003	C:\Dokumente und Einstellungen\All Users\Anwendungsdaten
Windows XP	C:\Dokumente und Einstellungen\All Users\Anwendungsdaten

Konfigurationsdateien der CONZEPT 16-Komponenten

- **Common**

Kontakt

Folgende Dateien werden im Standardverzeichnis (siehe oben) unter CONZEPT 16\Common abgelegt:

ca-bundle.crt Sammlung von Stammzertifikaten

Die Datei kann um weitere Stammzertifikate erweitert werden. Sie wird dann nicht mehr von der CONZEPT 16-Installationsroutine überschrieben.

• Server

Folgende Dateien werden im Standardverzeichnis (siehe oben) unter CONZEPT 16\Server abgelegt:

c16_xxxxxx.idn	<u>CONZEPT 16-Lizenz-Identitätsdatei</u>
c16.lic	<u>CONZEPT 16-Lizenzdatei</u>
c16_serv.ars	<u>Datenraumtabelle des Servers</u>
c16_serv_backup.ars	Sicherungskopie der <u>Datenraumtabelle des Servers</u>
c16_serv_svc.lgb	<u>Log-Datei</u> des CONZEPT 16-Server <u>Service-Prozesses</u>
c16_serv_mgr.lgb	Log-Datei des CONZEPT 16-Server Manager-Prozesses
c16_control.ini	Control-Center Konfigurationsdatei
c16_lgbview.ini	Log-Viewer Konfigurationsdatei
\Profiles	Verzeichnis für vom <u>Control-Center</u> abgeholt Log-Dateien

Im Installationsverzeichnis der CONZEPT 16-Servers wird die Datei c16_serv_mgr.http angelegt, wenn eine Lizenz mit Softwareschutz verwendet oder die Lizenzdatei (c16.lic) automatisch aktualisiert wird.

• Standard-Client / Advanced-Client

Folgende Dateien werden im Standardverzeichnis (siehe oben) unter CONZEPT 16\Client abgelegt:

c16.hst	<u>Hostnamen-Tabelle</u>
c16.cfg	<u>Konfigurationsdatei des Standard-Clients</u>
c16_apgi.cfg	<u>Konfigurationsdatei des Advanced-Clients</u>
c16_debg.ini	Konfigurationsdatei des Debuggers

• Druckprozessor

Folgende Dateien werden im Standardverzeichnis (siehe oben) unter CONZEPT 16\Printer abgelegt:

c16_ppcsvc.cfg	<u>Konfigurationsdatei des Druckprozessors</u>
c16_ppcsvc.log	<u>Log-Datei des Druckprozessors</u>

• SOA-Service

Folgende Dateien werden im Standardverzeichnis (siehe oben) unter CONZEPT 16\SOA abgelegt:

c16_soa.cfg	<u>Konfigurationsdatei des SOA-Service</u>
c16_soa.lgb	<u>Log-Datei des SOA-Service</u>

Kontakt

Zusätzlich werden für jeden definierten Task innerhalb des SOA-Services eine weitere Konfigurationsdatei und eine Protokolldatei in diesem Verzeichnis angelegt.

CONZEPT 16-Client

Beschreibung des CONZEPT 16-Clients

 Eine Auflistung der Betriebssysteme, auf denen der Client betrieben werden kann, finden Sie in den [Systemvoraussetzungen](#).

Die Aufgaben des CONZEPT 16-Datenbanksystems werden von mindestens zwei Programmen wahrgenommen. Der CONZEPT 16-Client schickt Anfragen an den CONZEPT 16-Server. Diese Anfragen beinhalten unter anderem das Öffnen von Datenbanken, das Anmelden von Benutzer und das Anfordern von Datensätzen. Die Verarbeitung der Datensätze findet beim Client statt.

Der CONZEPT 16-Client benötigt dabei keine Zugriffsberechtigungen auf die Datenbank selbst (die CA?-Dateien), sondern lediglich die Kommunikation mit dem CONZEPT 16-Server.

Die Bedienung der Entwicklungsumgebung wird in dem Kapitel Designer beschrieben. Alle weiteren Kapitel enthalten Beschreibungen der Elemente, aus denen eine Applikation zusammengestellt werden kann. Als Einstieg in die Entwicklung von Datenbanksystemen mit CONZEPT 16, wird das Tutorial empfohlen, das mit der CONZEPT 16-Evaluierungslizenz installiert wird.

Kontakt

Installation des Clients

Beschreibung der Installation des CONZEPT 16-Clients

Der CONZEPT 16-Client (in Advanced- und Standard-Ausführung) wird über die CONZEPT 16 Installationsroutine eingerichtet. Die Programmdateien befinden sich anschließend im Installationsverzeichnis unter \Client.

Dateien des Clients

c16.ca1	Leere Datenbank
c16.hlp	Hilfe-Datei / Nur Standard-Client
c16.tla	Token-Übersetzungstabelle
c16.tlp	Token-Übersetzungstabelle / Nur Standard-Client
c16.xls	Token-Übersetzungstabelle / Nur Standard-Client
c16.utl	OEM-Kit-Unterstützung
c16.vra	Ressourcendatei
c16.vrs	Zeichenketten deutsch / Nur Standard-Client
c16_1250.cpw	Codepage Window 1252 (ANSI)
c16_1250.fon	Font für Codepage Windows 1252 (ANSI)
c16_1254.cpw	Codepage Window 1254
c16_1254.fon	Font für Codepage Windows 1254
c16_apgi.exe	Advanced-Client (32-Bit)
c16_apgi_w64.exe	Advanced-Client (64-Bit)
c16_bar2d_w32.dll	Barcode-Bibliothek (32-Bit)
c16_bar2d_w64.dll	Barcode-Bibliothek (64-Bit)
c16_chart_w32.dll	Chart-Bibliothek (32-Bit)
c16_chart_w64.dll	Chart-Bibliothek (64-Bit)
c16_coded_w32.dll	CodeEdit-Bibliothek (32-Bit)
c16_coded_w64.dll	CodeEdit-Bibliothek (64-Bit)
c16_complay_w32.dll	Compatibility Layer (32-Bit) / Nur Standard-Client
c16_ctxoffice.ocx	CtxOffice-Steuerelement (32-Bit)
c16_ctxoffice_w64.ocx	CtxOffice-Steuerelement (64-Bit)
c16_ctxreg.exe	CTX-Registration (32-Bit)
c16_ctxreg_w64.exe	CTX-Registration (64-Bit)
c16_debg.dll	Debug Message Bibliothek
c16_debg.exe	Debugger (32-Bit)
c16_debg_w64.exe	Debugger (64-Bit)
c16_diff_w32.dll	Difference-Bibliothek (32-Bit)
c16_diff_w64.dll	Difference-Bibliothek (64-Bit)
c16_dlg_w32.dll	Dialog Designer (32-Bit)
c16_dlg_w64.dll	Dialog Designer (64-Bit)
c16_dlx_w32.dll	Dialog Designer Extension (32-Bit)
c16_dlx_w64.dll	Dialog Designer Extension (64-Bit)
c16_en.vrs	Zeichenketten englisch / Nur Standard-Client

Kontakt

c16_graph_w32.dll	Graphics-Bibliothek (32-Bit)
c16_graph_w64.dll	Graphics-Bibliothek (64-Bit)
c16_icu_w32.dll	Unicode-Bibliothek (32-Bit)
c16_icu_w64.dll	Unicode-Bibliothek (64-Bit)
c16_img_w32.dll	Bildanzeige-Bibliothek (32-Bit) / Nur Standard-Client
c16_obj_w32.dll	GUI-Objekte (32-Bit)
c16_obj_w64.dll	GUI-Objekte (64-Bit)
c16_pdf_w32.dll	PDF-Bibliothek (32-Bit)
c16_pdf_w64.dll	PDF-Bibliothek (64-Bit)
c16_pfdx_w32.dll	PDF-Anzeige-Bibliothek (32-Bit)
c16_pfdx_w64.dll	PDF-Anzeige-Bibliothek (64-Bit)
c16_prt_w32.dll	Graphischer Druckertreiber (32-Bit) / Nur Standard-Client
c16_res_w32.dll	GUI-Ressourcen (32-Bit)
c16_res_w64.dll	GUI-Ressourcen (64-Bit)
c16_ssl_w32.dll	SSL-Bibliothek (32-Bit)
c16_ssl_w64.dll	SSL-Bibliothek (64-Bit)
c16_sys_w32.dll	System-Objekte (32-Bit)
c16_sys_w64.dll	System-Objekte (64-Bit)
c16_tool.utl	Lizenzdatei für Toolbars / Nur Standard-Client
c16_tool_w32.dll	Toolbars (32-Bit) / Nur Standard-Client
c16_tpi_w32.dll	TAPI-Bibliothek (32-Bit) / Nur Standard-Client
c16_tr.vrs	Zeichenketten türkisch / Nur Standard-Client
c16_win.fon	Schriftart / Nur Standard-Client
c16_winc.exe	Standard-Client
c16_xml_w32.dll	XML-Bibliothek (32-Bit)
c16_xml_w64.dll	XML-Bibliothek (64-Bit)
c16_zip_w32.dll	ZIP-Bibliothek (32-Bit)
c16_zip_w64.dll	ZIP-Bibliothek (64-Bit)
\example	Beispielkonfigurationsdateien
c16.cfg	Konfigurationsdatei für den Standard-Client
c16_apgi.cfg	Konfigurationsdatei für den Advanced-Client

Splash-Screen beim Starten des Clients

Beim Start von CONZEPT 16 wird ein Splash-Screen angezeigt. Dieser Splash-Screen kann durch ein eigenes Bild ersetzt werden. Das Bild muss den Namen "C16_TOOL.*" besitzen und sich im Verzeichnis des Clients befinden. Es muss 450 Pixel breit und 250 Pixel hoch sein und kann in den Formaten BMP, JPG und TIF vorliegen. Unterhalb des Bildes bleibt ein Streifen für Statusmeldungen beim Start des Clients frei.

Beispiel:

My Splash-Screen

Das Bild kann mit der rechten Maustaste angeklickt und mit "Speichern unter" unter dem Namen c16_tool.bmp im Client-Verzeichnis gespeichert werden. Es wird dann beim Starten des Clients angezeigt.

Soll beim Start des Clients kein Splash-Screen angezeigt werden, muss in der Konfigurationsdatei des Clients der Eintrag SplashOn = N eingetragen werden.

Starten des CONZEPT 16-Clients

Starten des CONZEPT 16-Standard- und Advanced-Clients

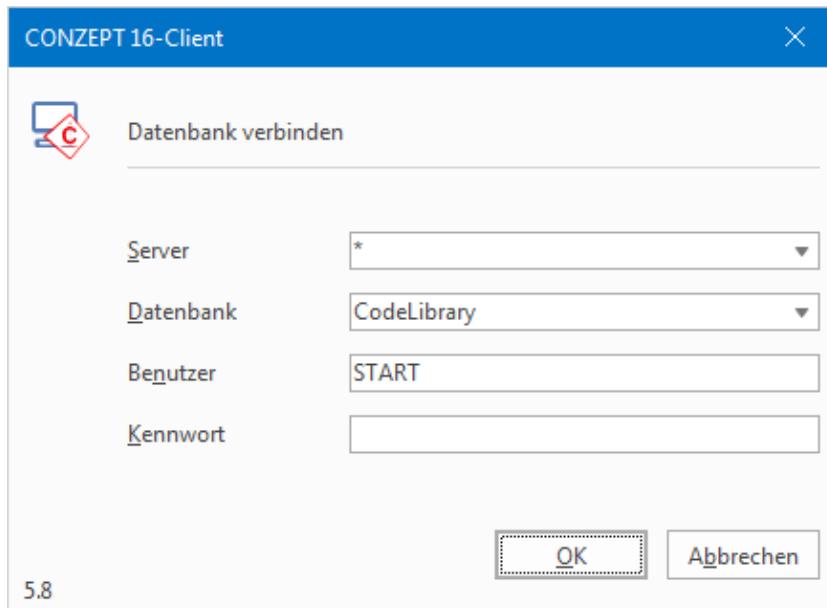


Eine Auflistung der Betriebssysteme, auf denen der CONZEPT 16-Client betrieben werden kann, finden Sie in den [Systemvoraussetzungen](#).

Wurde bei der Installation der Standard- und/oder der Advanced-Client installiert, befinden sich alle notwendigen Dateien zum Starten in dem Unterverzeichnis Client. Wurden bei der Installation Verknüpfungen im Startmenü angelegt, können beide Clients über diese Verknüpfungen gestartet werden. Der Start kann aber auch direkt aus dem Verzeichnis erfolgen.

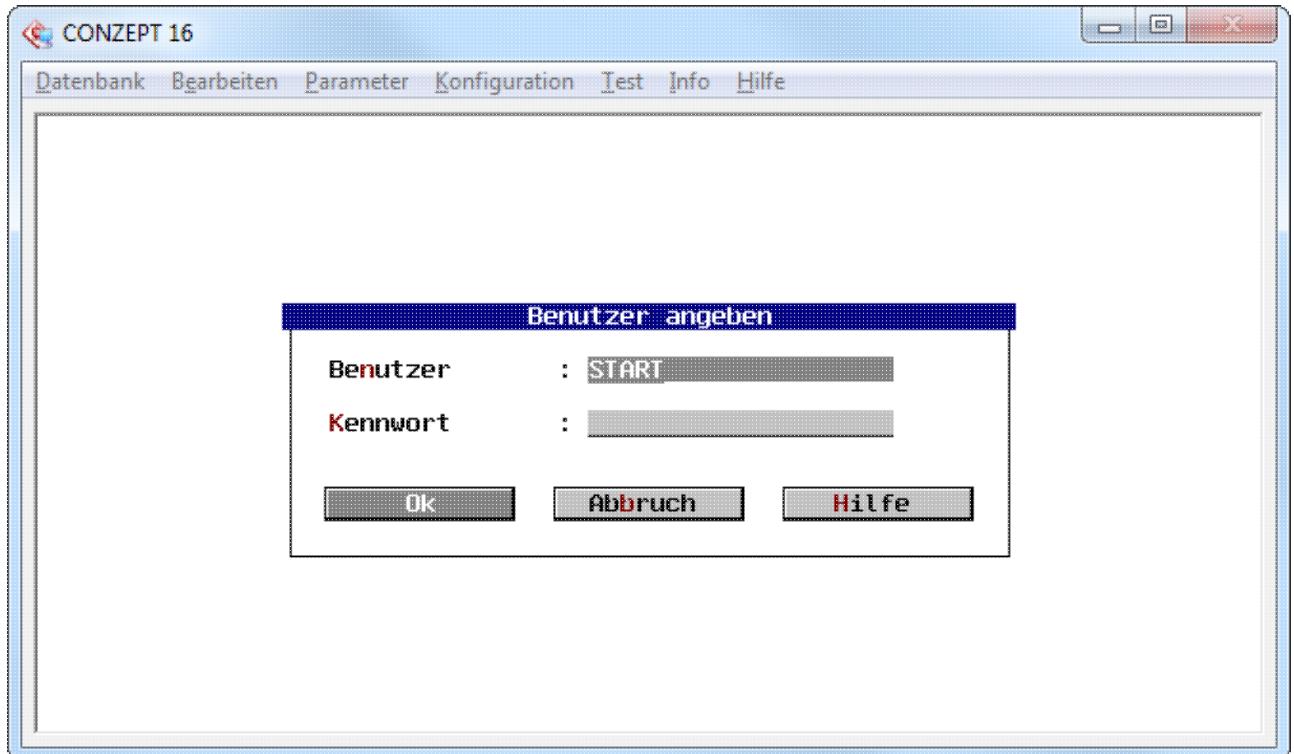
Starten des Advanced-Clients

Der Advanced-Client kann nur gestartet werden, wenn als Lizenz eine Advanced-, Enterprise- oder Developer-Edition verwendet wird (siehe [Lizenztypen](#)). Der Client wird mit dem Programm c16_apgi.exe (Advanced-Client) aufgerufen. Nach dem Start muss der Name oder die IP-Adresse des CONZEPT 16-Servers (ist der Server auf der gleichen Maschine gestartet, kann * angegeben werden), der Name der Datenbank sowie der Benutzer und dessen Kennwort angegeben werden. Sind alle Angaben korrekt, wird die Datenbank geöffnet und je nach Berechtigung des angemeldeten Benutzers die Applikation oder die Entwicklungsumgebung gestartet.



Starten des Standard-Clients

Jede Lizenz berechtigt zum Starten des Standard-Clients. Der Client wird mit dem Programm c16_winc.exe (Standard-Client) aufgerufen. Nach dem Start kann eine Datenbank über den Menüpunkt "Datenbank / Öffnen..." geöffnet werden. In den folgenden Dialogen wird der Name bzw. die IP-Adresse des CONZEPT 16-Servers angegeben, anschließend die Datenbank aus einer Liste ausgewählt und danach der Benutzer und sein Kennwort angegeben. Sind alle Angaben korrekt, wird die Datenbank geöffnet und je nach Berechtigung des angemeldeten Benutzers die Applikation oder die Entwicklungsumgebung gestartet.



Parameter beim Start des Clients zum Öffnen einer Datenbank

Beim Start des Clients (Standard- und Advanced-Client) können die notwendigen Informationen zum Öffnen einer Datenbank auch gleich als Parameter übergeben werden:

```
"c:\program files\CONZEPT 16\client\c16_apgi.exe" <server> <database> <user> <password>
```

- **<server>**

Hier kann der Name oder die IP-Adresse des Servers angegeben werden. Ist der Server auf dem gleichen Rechner gestartet, wie der Client, kann * angegeben werden. Bei Verwendung der Hot-Standby-Option werden hier der Primär- und der Sekundärserver getrennt durch ein +-Zeichen angegeben.

- **<database>**

Der symbolische Name der Datenbank. Der Name wird beim Eintragen der Datenbank beim Server angegeben (siehe Konfiguration des Servers).

- **<user>**

Der Name des Benutzers. Der Benutzer muss zuvor in der Datenbank über die Benutzerverwaltung angelegt worden sein. In einer leeren Datenbank sind die Benutzer SUPERUSER und USER vordefiniert. Der Benutzer SUPERUSER kann auch mit SU angegeben werden. Der Superuser kann nicht aus der Datenbank gelöscht werden.

- **<password>**

Kontakt

Das Kennwort des zuvor genannten Benutzers. Das Kennwort sollte nicht in einer Verknüpfung angegeben werden, da es dort unkodiert abgelegt ist. Die vordefinierten Benutzer haben kein Kennwort. Für diese Benutzer sollten Kennwörter vergeben werden.

Parameter können von hinten weggelassen werden. Die Anmeldung erfolgt dann soweit es mit den Angaben möglich ist. Hat der angegebene Benutzer ein Kennwort, wird der Benutzer zur Angabe aufgefordert. Ist nur ein Server angegeben, gelangt man in die Auswahl der Datenbanken usw.

Beispiel:

```
"c:\program files\CONZEPT 16\client\c16_apgi.exe" * Codelibrary Start
```

In diesem Beispiel wird die Datenbank "Codelibrary" auf dem lokalen Server mit dem Benutzer "Start" geöffnet.

Übergabe weiterer Parameter

Alle weiteren Parameter werden in /<name>=<value>-Paaren übergeben. Diese Übergabeparameter können innerhalb der Applikation mit der Anweisung SysGetArg() abgefragt werden.

Folgende Übergabeparameter werden direkt vom Client ausgewertet:

- **/c16=<path>**

Hier kann der Pfad zu den Programmdateien des Clients angegeben werden. Die DLLs werden standardmäßig aus dem gleichen Verzeichnis geladen, aus dem auch der Client gestartet wurde.

- **/c16cfg=<path\file>**

In der Konfigurationsdatei stehen Voreinstellungen, die beim Start des Clients berücksichtigt werden sollen. Standardmäßig wird die Datei c16.cfg (Standard-Client) bzw. c16_apgi.cfg (Advanced-Client) aus dem gleichen Verzeichnis geladen, aus dem der Client gestartet wird. Soll eine andere Datei oder eine Datei aus einem anderen Verzeichnis verwendet werden, kann diese Datei hier angegeben werden.

- **/c16tmp=<path>**

Für temporäre Dateien wird der Windows-Temp-Pfad verwendet. Soll ein anderer Pfad verwendet werden, kann er hier angegeben werden.

- **/c16lang=DE|EN|TR**

Die Anzeigesprache im Designer und der textbasierten Oberfläche (Standard-Client) entspricht der Sprache der Benutzeroberfläche des Betriebssystems. Wird die Sprache des Systems nicht von CONZEPT 16 unterstützt, wird Englisch als Sprache verwendet. Die Sprache kann aber auch mit diesem Parameter vorgegeben werden. Es kann zwischen den Sprachen Deutsch (DE), Englisch (EN) und Türkisch (TR) gewählt werden. Im Designer stehen nur die Sprachen Deutsch und Englisch zur Verfügung.

• **/c16splashon=y|n**

Mit diesem Kommandozeilenargument kann der Splash-Screen beim Starten des Clients unterbunden werden (/c16splashon=n). Standardmäßig wird der Splash-Screen angezeigt. Ist sowohl der Eintrag SplashOn in der Konfigurationsdatei als auch dieses Kommandozeilenargument angegeben, wird das Kommandozeilenargument bevorzugt.

• **/c16loggingui=y|n**

Mit diesem Kommandozeilenargument kann bei der Anmeldung der grafische Login-Dialog des Advanced-Clients verwendet werden (/c16loggingui=y). Standardmäßig wird eine Masken-basierte Anmeldung verwendet. Ist sowohl der Eintrag LoginGui in der Konfigurationsdatei als auch dieses Kommandozeilenargument angegeben, wird das Kommandozeilenargument bevorzugt.

• **/load_user_profile=y|n**

Mit diesem Kommandozeilenargument kann beim Advanced-Client verhindert (/load_user_profile=n) werden, dass das Benutzerprofil des ausgewählten Benutzers geladen wird, wenn der Designer gestartet wird. Beim Verlassen des Designers werden dann die aktuellen Einstellungen gespeichert. Standardmäßig wird das Benutzerprofil geladen.

• **/c16applogpath=<path>**

Mit diesem Kommandozeilenargument kann der Pfad des Anwendungsprotokolls definiert werden. Ist das Argument nicht gesetzt, wird in der Konfigurationsdatei der Eintrag AppLogPath und die Umgebungsvariable C16AppLogPath auf vorhandensein geprüft. Ist kein Eintrag gesetzt, wird der Clientpfad verwendet.

• **/c16applogoptions=<level>**

Mit diesem Kommandozeilenargument können zusätzliche Debug-Informationen für das Anwendungsprotokolls aktiviert werden. Ist das Argument nicht gesetzt, wird in der Konfigurationsdatei der Eintrag AppLogOptions und die Umgebungsvariable C16AppLogOptions auf vorhandensein geprüft. Ist kein Eintrag gesetzt, werden keine zusätzlichen Debug-Informationen protokolliert. Folgende Werte sind möglich:

- ◆ 0 - Keine zusätzliche Protokollierung.
- ◆ 2 - Protokolliert den Ablauf, wenn der Designer terminiert wird.
- ◆ 4 - Anwendungsprotokoll auch im Standard- und Advanced-Client (außerhalb des Designers) aktivieren.

• **/c16pluginport=<port>**

Mit diesem Kommandozeilenargument kann der Port der Designer Plugin-Schnittstelle definiert werden. Es muss ein freier Port mit einer gültigen Portnummer (<= 65535) sein. Ist das Argument nicht gesetzt, wird in der Konfigurationsdatei der Eintrag PluginPort und die Umgebungsvariable C16PluginPort auf vorhandensein geprüft. Ist kein Eintrag gesetzt, wird die Plugin-Schnittstelle nicht aktiviert.

• **/c16debugoptions=<wert>**

Kontakt

Mit diesem Kommandozeilenargument können Debug-Optionen des Clients aktiviert werden. Ist das Argument nicht gesetzt, wird in der Konfigurationsdatei der Eintrag DebugOptions und die Umgebungsvariable C16DebugOptions auf vorhandensein geprüft. Folgende Werte können miteinander kombiniert (verodert) werden:

- ◆ 0x0001 - Protokollierung der Aufrufe der Tapi-Schnittstelle.
- ◆ 0x0002 - Protokollierung der PDF-Schnittstelle. Protokolliert werden Aufrufe, die über die PDF-Befehle laufen und solche die beim Drucken über PrtJobClose() mit der Option PrtJobPdf durchgeführt werden.
- ◆ 0x0004 - Sicherung der Metafiles, die beim Druck bzw. bei dem Befehl PdfInsertMeta() angelegt werden. Die Dateien werden im Pfad der Protokolldateien abgelegt.
- ◆ 0x0020 - Druckjob-Verarbeitung. Die Verarbeitung von PrtJobClose() wird protokolliert.
- ◆ 0x1000 - Protokollierung der Ereignisaufraufe bei Oberflächenobjekten.
- ◆ 0x2000 - Performance-Protokollierung für das Laden und Erstellen von Frame-Objekten. Es werden nur Operationen protokolliert, bei denen die Erstellung länger als 3, die Initialisierung länger als 5, oder der Gesamtprozess länger als 30 Millisekunden benötigt hat.
- ◆ 0x4000 - Die Initialisierung und die Terminierung der GDI-Plus-Bibliothek werden protokolliert.
- ◆ 0x8000 - Weitere Informationen zu irregulären Zuständen der Speicherverwaltung werden protokolliert.

Zusätzlich muss mit dem Argument c16debuglogpath der Pfad für die Protokolldateien definiert werden.

• **/c16debuglogpath=<pfad>**

Dieses Kommandozeilenargument definiert den Pfad für die Protokolldateien für das Argument c16debugoptions. Ist das Argument nicht gesetzt, wird in der Konfigurationsdatei der Eintrag DebugLogPath und die Umgebungsvariable C16DebugLogPath auf vorhandensein geprüft.

• **/c16designtool=<tool>**

Beim Starten des Advanced-Client kann mit diesem Kommandozeilenargument angegeben werden, dass statt des Designers oder der Applikation ein Designtool mit den jeweiligen Benutzerrechten gestartet werden soll. Folgende Werte können angegeben werden:

UserManagement	<u>Benutzerverwaltung</u>
BlobManagement	<u>BLOB-Verwaltung</u>
RecordManagement	<u>Datensatzverwaltung</u>
ResourceManagement	<u>Ressourcenverwaltung</u>

• **/c16theme=<theme>**

Beim Starten eines Designtools (siehe Argument c16designtool) kann mit diesem Argument das zu verwendende Theme angegeben werden. Aktuell können hier die Standard-Themes "OfficeBlue" und "OfficeDark" angegeben werden. Sofern Windows unter hohem Kontrast läuft, wird das Argument ignoriert.

• **/c16profilepath=<pfad>**

Kontakt

Dieses Kommandozeilenargument definiert den Pfad für eine Profildatei. Diese Datei wird anstelle des in der Datenbank vorliegenden Benutzerprofils verwendet. Dabei wird wie folgt vorgegangen:

1. Profildatei nicht vorhanden

Ist die Datei nicht vorhanden wird der Designer mit dem Datenbankprofil des Benutzers gestartet. Beim Beenden des Designers wird die angegebene Datei generiert und das aktuelle Profil des Benutzers dort gespeichert. Eine Veränderung des Profils in der Datenbank findet nicht statt.

2. Profildatei nicht lesbar oder Version ungültig

Ist die Datei vorhanden, kann das Profil jedoch nicht geladen werden (z. B. keine gültige Profildatei oder ungültige Profilverision), dann wird eine entsprechende Fehlermeldung angezeigt und der Designer startet ebenfalls mit dem Datenbankprofil des Benutzers. Beim Beenden wird dann das Datenbankprofil des Benutzers aktualisiert. Es findet keine Veränderung des extern vorliegenden Datei statt.

3. Profildatei lesbar und gültig

Ist die Datei vorhanden und kann diese erfolgreich geladen werden, dann erfolgt die Wiederherstellung des Benutzereinstellungen aus dieser Datei. Beim Beenden des Designers wird der aktuelle Zustand wieder in die Datei zurückgespeichert. Eine Veränderung des Profils in der Datenbank findet nicht statt. Nach dem Ladevorgang wird die Datei geschlossen und wird vom Designer bis zur Beendigung nicht mehr verwendet.

Bei einem Versionswechsel (z. B. 5.8 auf 5.9) ist das Profil in der Regel für den älteren Client nicht mehr lesbar. Aus diesem Grund wird eine Profilverision in der externen Datei gespeichert. Diese Profilverision hindert den Designer eines älteren Clients das extern vorliegende Profil zu laden, da dies zu unvorhersehbaren Resultaten führen kann. Es tritt Fall (2) in Kraft.

Der Profildatei enthält Daten in binärer Form und wird verschlüsselt gespeichert. Die Profildatei erhält die Dateierweiterung .bds für binary designer settings. Die Erweiterung wird auch angehängt, wenn bereits eine andere Erweiterung vorhanden ist.

Fehlermeldungen beim Öffnen der Datenbank

Hier muss zwischen den einzelnen Schritten der Anmeldung unterschieden werden:

[Fehler bei der Verbindung zum CONZEPT 16-Server](#)

[Fehler beim Öffnen der Datenbank](#)

[Fehler bei der Anmeldung](#)

Fehler bei der Verbindung zum CONZEPT 16-Server

- **Server nicht vorhanden**

Kontakt

Der Client konnte keine Verbindung zum CONZEPT 16-Server aufnehmen. Mögliche Ursachen können sein, dass der CONZEPT 16-Server nicht gestartet ist oder die Verbindung auf Grund von Firewalleinstellungen nicht zugelassen wurde. Der Client versucht auf Port 4722 die Verbindung herzustellen.

Fehler beim Öffnen der Datenbank

- **Datenbank nicht vorhanden**

Der angegebene symbolische Name ist nicht in der Datenraumtabelle vorhanden.

- **Fehler bei Serverstart**

Der Datenbankprozess konnte nicht gestartet werden. Die Fehlerursache ist in der Protokolldatei der Datenbank (<Datenbankname>.lgb) oder des Managerprozesses (c16_serv_mgr.lgb) eingetragen (siehe CONZEPT 16-Server - Log-Einträge).

- **Datenbank ist durch den Administrator gesperrt.**

Der Administrator hat die Funktion Login sperren ausgeführt. Eine Anmeldung ist erst wieder nach Durchführung von Login freigeben möglich.

- **Datenbank ist durch das Herunterfahren gesperrt**

Die Datenbank wird gerade geschlossen und ist bis zum Beenden des Datenbankprozesses gesperrt. Zu einem späteren Zeitpunkt kann die Datenbank wieder geöffnet werden.

- **Datenbank ist durch eine Serviceoperation gesperrt**

Auf der Datenbank wird gerade eine Serviceoperation durchgeführt, die exklusiven Zugriff hat (beispielsweise eine Optimierung der Datenbank oder das Einspielen eines Updates).

- **Datenbank ist gesperrt (Standby-Modus)**

Die Datenbank kann nicht geöffnet werden, da der andere Server des Hot-Standby-Systems die Datenbank im aktiven Betrieb hat.

- **Datenbank ist gesperrt (Sperre für Rollback)**

Die Datenbank ist nicht abgeschlossen, ein Rollback kann aufgrund einer gesetzten Rollback-Sperre aber nicht gestartet werden. Die Sperre kann über die Web-Administration aufgehoben werden.

- **Datenbank ist gesperrt (Sperre für Login)**

Bei der Datenbank ist eine Login-Sperre eingetragen. Die Sperre kann über die Web-Administration aufgehoben werden.

- **Datenbank ist gesperrt (Standby-System nicht verfügbar)**

Bei der Datenbank ist eine Login-Sperre für den Fall der Nichtverfügbarkeit des Standby-Servers eingetragen. Die Sperre kann über die Web-Administration aufgehoben werden.

- **Datenbank ist gesperrt**

Kontakt

Die Sperrung ist durch keine der obigen Fälle verursacht. Der Status des Servers und der Datenbank muss per Web-Administration - und gegebenenfalls anhand der Protokoll-Dateien - überprüft werden.

Fehler bei der Anmeldung

• Anmeldung fehlgeschlagen

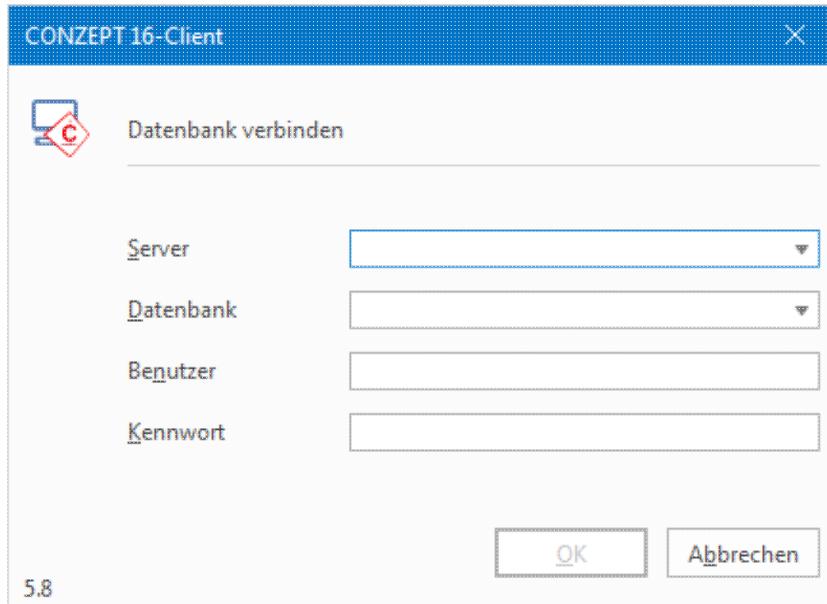
Der angegebene Benutzer ist nicht vorhanden, das angegebene Kennwort stimmt nicht oder die maximale Anzahl der Sitzungen (siehe UrmPropMaxSessions) ist erreicht. Diese Meldung wird nicht angezeigt, wenn in den Basisparametern der Datenbank die Einstellung "Automatisches Verlassen bei ungültigem Benutzer" gesetzt ist. In diesem Fall wird der Client ohne Fehlermeldung beendet.

Advanced-Client

Beschreibung des CONZEPT 16-Advanced-Clients

i Eine Auflistung der Betriebssysteme, auf denen der Advanced-Client betrieben werden kann, finden Sie in den [Systemvoraussetzungen](#).

Im Gegensatz zum Standard-Client ist der Advanced-Client ausschließlich zum einfachen übersichtlichen Aufrufen von Datenbanken gedacht. Die auf dem Server zur Verfügung stehenden Datenbanken werden übersichtlich in einer Liste dargestellt, in der ebenfalls weitere [Informationen](#) zur Datenbank ersichtlich werden.



• Protokolle

Falls ein Server noch nicht in der Datei C16.HST festgehalten ist muss hier explizit das Protokoll, welches der angegebene Server verwendet, ausgewählt werden.

• Server

In diesem Abschnitt werden die in der Datei C16.HST vermerkten und erreichbaren Server in einer Auswahlliste bereitgestellt. Zudem sind in dieser Liste zusätzlich Informationen zum Server, wie zum Beispiel den Versionsstand oder das verwendete Protokoll, ersichtlich.

Server	*			
Datenbank	Name	Version	Protokoll	Lizenz
Benutzer	vm1	5.8.08	TCP	CE152000MN

i Falls am Arbeitsplatz des Clients ebenfalls ein Server läuft, wird dieser immer als * in der Liste angezeigt.

• Datenbank

In dieser Liste werden alle eingetragenen Datenbanken des Servers inklusive Informationen aufgelistet. Falls keine Datenbank vorhanden ist oder der Server

Kontakt

nicht gefunden wurde wird diese Liste leer dargestellt.

• **User & Password**

Hier wird der zur Anmeldung benötigte User mit dem dazugehörigen Passwort angegeben.

In der Konfigurationsdatei (c16_apgi.cfg) können zudem Einstellungen des Advanced-Client verändert werden.

Der Advanced-Client verfügt nur über eine grafische Oberfläche und kann auch nur A+ Prozeduren verarbeiten. Durch den Verzicht auf die Textoberfläche und die A-Prozeduren, wird der Client in der Verarbeitung schneller.

Die Bedienung der Entwicklungsumgebung wird in dem Kapitel Designer beschrieben. Alle weiteren Kapitel enthalten Beschreibungen der Elemente, aus denen eine Applikation zusammengestellt werden kann. Als Einstieg in die Entwicklung von Datenbanksystemen mit CONZEPT 16, wird das Tutorial empfohlen, das mit der CONZEPT 16-Evaluierungslizenz installiert wird.

Limitationen des Anwendungsprozesses

Aufstellung der Limitationen des Anwendungsprozesses

Jede CONZEPT 16-Anwendung wird als Prozess vom Betriebssystem ausgeführt.

Prozesse unterliegen den vorgegebenen Einschränkungen.

Windows-Prozesse mit graphischer Oberfläche

Die graphische Benutzeroberfläche eines Prozesses unter Windows besteht aus Fensterobjekten (Frame, Label, Edit, Button, etc.), die Ressourcen von Windows in Anspruch nehmen. Bei Fensterobjekten sind dies vor Allem User- und GDI-Ressourcen.

• User-Ressourcen

Jedes von Windows dargestellte Fensterobjekt benötigt mindestens eine User-Ressource. Die Anzahl der User-Ressourcen ist Windows-seitig auf 10000 limitiert.

• GDI-Ressourcen

Für die graphische Ausgabe benötigen Fensterobjekte GDI-Ressourcen. Je nach Komplexität eines Objektes kann ein Fensterobjekt viele GDI-Ressourcen verbrauchen. Die Anzahl der GDI-Ressourcen ist Windows-seitig auf 10000 limitiert.

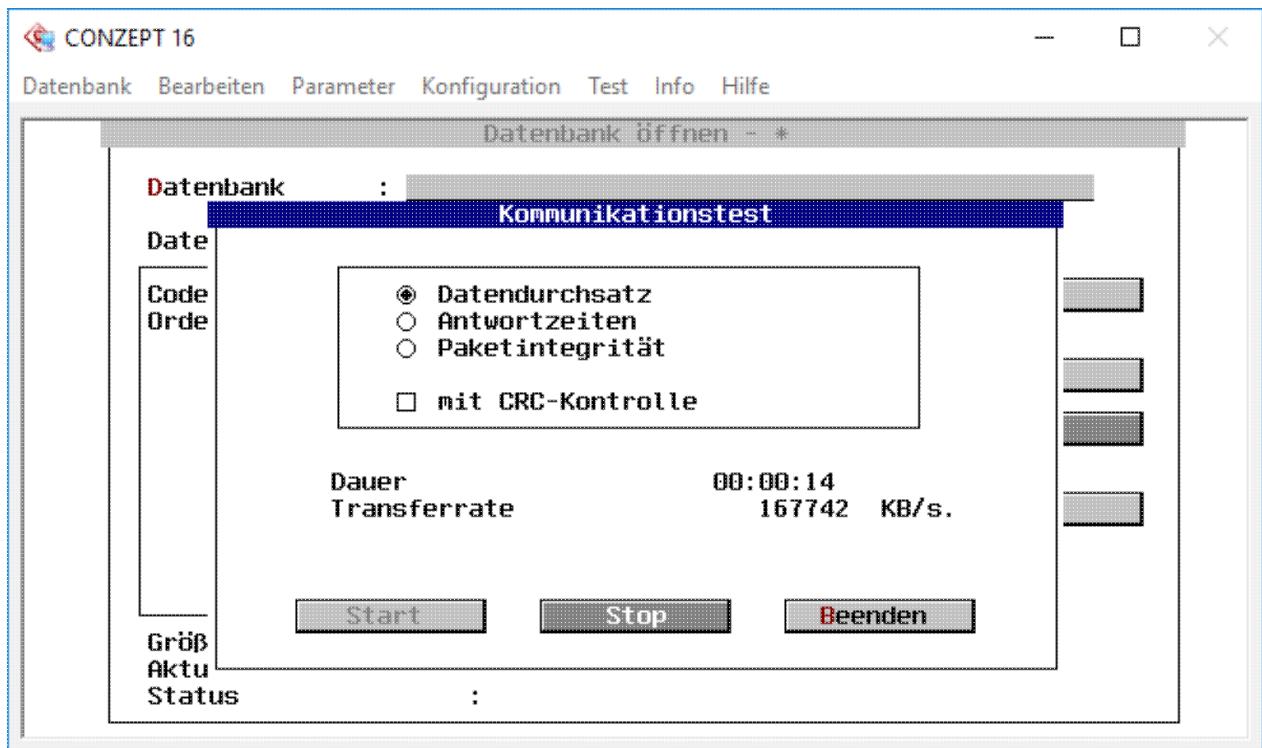
Kontakt

Datenbank eintragen - Test

Kommunikation zwischen Client und Server testen

Test...

Diese Funktion steht beim Standard-Client zur Verfügung, nachdem ein Server ausgewählt wurde. Sie ermittelt die Zuverlässigkeit und die ungefähre Netto-Transferrate zwischen Client und Server in KB pro Sekunde. Dabei wird sowohl die Sende- als auch die Empfangsrate überprüft. Im anschließenden Dialog kann die Art des Testes ausgewählt werden.



Mit [Start] wird der Test gestartet. Der Test wird solange wiederholt, bis [Stop] gedrückt wird. Der Dialog wird mit [Beenden] verlassen.

• Datendurchsatz

Hierbei wird die maximale Übertragungsrate zwischen Client und Server ermittelt, wobei eine für das jeweilige Protokoll günstige Paketgröße gewählt wird (zwischen 1400 und 2076 Bytes).

Die folgende Tabelle soll als Anhaltspunkt dienen (geringe Netzlast, eine Station im Test, Protokoll TCP/IP):

Medium	typischer Durchsatz	minimaler Durchsatz
Ethernet 10mb	450 - 900 KB/s	400 KB/s
Fast Ethernet 100mb	2000 - 4000 KB/s	1800 KB/s
Gigabit Ethernet 1000mb	5000 - 25000 KB/s	4000 KB/s
Geringere Durchsätze können auf Probleme in der Netzwerkumgebung hindeuten, evtl. gehen Datenpakete verloren usw.		

• Antwortzeiten

Neben der maximalen Transferrate ist die Antwortzeit des Servers das zweite, fast noch wichtigere Kriterium für die Netzwerkleistung. Die ist der Zeitraum, der zwischen dem Absenden einer Client-Anfrage und dem Eintreffen der Serverantwort vergeht. Bei diesem Test sendet der Server ein Anfragepaket direkt wieder an den Client, wobei die minimale Paketgröße verwendet wird. Daraus ergibt sich zum einen die Anzahl von Anfragen pro Sekunde, zum anderen die Verzögerung (Latenz), die bei einer einzelnen Anfrage entsteht.

Die folgende Tabelle soll als Anhaltspunkt dienen (geringe Netzlast, eine Station im Test, Protokoll TCP/IP):

Medium	typische Anzahl	minimale Anzahl
Ethernet 10mb	500	2000 A/s
Fast Ethernet 100mb	2500	5000 A/s
Gigabit Ethernet 1000mb	2500	15000 A/s

• Paketintegrität

Hierbei werden Pakete maximaler Größe (2076 Bytes) ausgetauscht, wobei der Austausch immer mit eingeschalteter CRC-Überprüfung stattfindet. Die Anzahl von CRC-Fehlern muss immer 0 sein, ansonsten läuft die Verbindung nicht stabil. Bei einem Sendefehler ist das Paket nicht korrekt beim Server angekommen, bei einem Empfangsfehler war das Antwortpaket vom Server nicht in Ordnung.

Die Tests "Datendurchsatz" und "Antwortzeiten" können auch mit eingeschalteter CRC-Prüfung (aktiviertes Kontrollkästchen "mit CRC-Kontrolle") durchgeführt werden. Standardmäßig verwenden diese beiden Tests keine CRC-Absicherung.



Bei Tests sollten mehrere Clients (gegebenenfalls alle) parallel testen, da bestimmte Fehler nur bei sehr hoher Netzwerklast auftreten.

Kontakt

Konfiguration des Clients

Eintragungen in den Konfigurationsdateien

Das Verhalten des Clients wird hauptsächlich durch Einstellungen in der Datenbank gesteuert. Verschiedene Einstellungsmöglichkeiten werden allerdings schon vor dem Öffnen einer Datenbank benötigt. Diese Einstellungen werden in den Dateien C16.CFG bzw. C16_APGL.CFG (Konfigurationsdatei) und C16.HST (Host-Tabelle) vorgenommen.

In der Konfigurationsdatei finden sich Einträge zur textbasierten Oberfläche, zur verwendeten Sprache und zur Umgebung des Clients. Die Host-Tabelle dient zur Identifizierung von CONZEPT 16-Servern im Netzwerk über das Protokoll TCP/IP.

Kontakt

Konfigurationsdatei - C16.CFG

Eintragungen in der Konfigurationsdatei

Die Konfigurationsdatei enthält Informationen im ASCII-Format, die das Verhalten von CONZEPT 16 beeinflussen, unabhängig von der benutzten Datenbank. Innerhalb der Datei beansprucht jeder Parameter eine Zeile. Eine Beispiel-Konfigurationsdatei befindet sich im Abschnitt [Beispiel - c16.cfg](#).

Die CONZEPT 16-Konfigurationsdatei kann einen beliebigen Namen haben, lediglich die Endung .CFG ist vorgegeben. So ist es möglich, Starteinstellungen für verschiedene Konfigurationen oder Benutzer vorzunehmen. Welche Konfigurationsdatei beim Start angesprochen wird, kann mit der Umgebungsvariable oder dem Startparameter C16CFG gesetzt werden. Ist nichts angegeben, sucht CONZEPT 16 im Standardpfad für [Konfigurationsdateien](#) nach der Datei c16.cfg. Ist dort keine entsprechende Datei vorhanden, wird in dem Verzeichnis, aus dem der Client gestartet wurde, die Datei gesucht.

Folgende Einträge können in der Konfigurationsdatei vorgenommen werden:

<u>: (Semikolon)</u>	Kommentar
<u>AppLogPath</u>	Pfad des Anwendungsprotokolls
<u>AppLogOptions</u>	Protokollierungsoptionen des Anwendungsprotokolls
<u>Break</u>	Programmabbruch möglich
<u>Codepage</u>	Zeichentabelle setzen
<u>DateFormat</u>	Datumsformat
<u>DateWindow</u>	Datumsfenster
<u>DebugOptions</u>	Debug-Optionen des Clients
<u>DebugLogPath</u>	Protokollpfad für Debug-Ausgaben des Clients
<u>Echo</u>	Echo bei Kennworteingabe
<u>FontSize</u>	Fontgröße
<u>FontType</u>	Fonttyp
<u>Lang_Display</u>	Sprache der Anzeige auswählen
<u>LocalDs</u>	Verzeichnis der lokalen Datenstruktur
<u>LoginGui</u>	Grafischen Login-Dialog verwenden
<u>LoginGuiPicture</u>	Hintergrundbild für Login-Dialog
<u>PasswordRetry</u>	Anzahl der Anmeldeversuche
<u>PluginPort</u>	Port der Plugin-Schnittstelle
<u>SaveWindowPos</u>	Fensterposition speichern
<u>ScreenSize</u>	Bildschirmauflösung
<u>ServerAutoStart</u>	Server starten
<u>SplashOn</u>	Splash-Screen beim Start anzeigen
<u>TcpConnectTimeout</u>	Verbindungs-Timeout des Clients
<u>TimeFormat</u>	Zeitformat
<u>Unique</u>	Eindeutiger Benutzername
<u>Upgrade</u>	Datenbankkonvertierung unterbinden
<u>Win3DMode</u>	3D-Modus einschalten

WindowPos_Win letzte Fensterposition

• ; (Semikolon) - Kommentar

Mit dem Zeichen ";" wird eine Kommentarzeile eingeleitet. Die gesamte restliche Zeile wird nicht ausgewertet.

;Kommentar

• **AppLogPath - Pfad des Anwendungsprotokolls**

Definiert den Pfad des Anwendungsprotokolls. Der Eintrag wird nur ausgewertet, wenn das Kommandozeilenargument c16AppLogPath beim Starten des Clients nicht angegeben wurde. Ist dieser Eintrag in der Konfigurationsdatei ebenfalls nicht vorhanden, wird die Umgebungsvariable C16AppLogPath geprüft. Standardmäßig wird der Clientpfad verwendet.

AppLogPath=C:\CONZEPT 16\Logs\

• **AppLogOptions - Protokollierungsoptionen des Anwendungsprotokolls**

Definiert, ob zusätzliche Debug-Informationen im Anwendungsprotokoll protokolliert werden sollen. Der Eintrag wird nur ausgewertet, wenn das Kommandozeilenargument c16AppLogOptions beim Starten des Clients nicht angegeben wurde. Ist dieser Eintrag in der Konfigurationsdatei ebenfalls nicht vorhanden, wird die Umgebungsvariable C16AppLogOptions geprüft. Standardmäßig werden keine zusätzlichen Debug-Informationen protokolliert. Folgende Werte sind möglich:

- ◆ 0 - Keine zusätzliche Protokollierung.
- ◆ 2 - Protokolliert den Ablauf, wenn der Designer terminiert wird.
- ◆ 4 - Anwendungsprotokoll auch im Standard- und Advanced-Client (außerhalb des Designers) aktivieren.

AppLogOptions=2

• **Break - Programmabbruch möglich**

Beim Beenden von CONZEPT 16 wird grundsätzlich eine Sicherheitsabfrage eingeblendet. Diese Sicherheitsabfrage kann durch eine eigene Routine ersetzt werden. Dazu muss die "Prozedur am Ende" in den Benutzerparametern auf die entsprechende Prozedur gesetzt werden. Die Sicherheitsabfrage von CONZEPT 16 wird durch BREAK=Y deaktiviert.

BREAK=Y/N

• **Codepage - Zeichentabelle setzen**

Hier kann unabhängig von dem im Betriebssystem eingestellten Gebietsschema eine Zeichentabelle eines bestimmten Landes angegeben werden. Die Datei C16_<nummer>.CPW wird verwendet, wenn das entsprechende Gebietsschema gesetzt ist. Der Eintrag in der Konfigurationsdatei ist nur notwendig, wenn trotz eines Gebietsschemas eine andere Codepage verwendet werden soll. Die Eintragungen wirken sich ebenfalls auf die Druckersteuersequenzen, die die Angabe einer Codepage zulassen, aus.

CODEPAGE=n

• **DateFormat - Datumsformat**

Mit der Einstellung DateFormat wird das Datumsformat in Masken und Dialogen vorgegeben. Das definierte Datumsformat des Betriebssystems wird ignoriert.

Folgende Formate sind möglich:

dd?MM?yydd?MM?yyyyMM?dd?yyMM?dd?yyyyyy?MM?ddyyyy?MM?dd

Es ist auf die Gross- und Kleinschreibung zu achten und das Fragezeichen steht für ein beliebiges Trennzeichen.

DATEFORMAT=dd/MM/yy; -> 12/12/06

• **DateWindow - Datumsfenster**

Mit diesem Eintrag kann ein Datumsfenster angelegt werden. Dieses Datumsfenster wirkt sich nur auf die Verarbeitung von Datumsfeldern und -konstanten ohne Jahrhundertangabe aus. Ist die angegebene (zweistellige) Jahreszahl kleiner als der eingetragene Wert, wird das Datum um das aktuelle Jahrhundert ergänzt. Ist die Jahreszahl größer oder gleich dem Wert, wird das Datum um das vorhergehende Jahrhundert ergänzt. Die Ergänzung betrifft die Eingabe in Datumsfelder, Transfer von Daten und die Datumskonstanten in der Programmierung.

Bei Angabe von 0 wird das Datum um das aktuelle Jahrhundert ergänzt.

DATEWINDOW=n

Zum Beispiel wird bei einem Eintrag DATEWINDOW=20 die Eingaben zu folgenden Daten ergänzt:

Eingabe Datum

01.01.12 01.01.2012

23.04.98 23.04.1998

15.11.20 15.11.1920

23.02.19 23.02.2019

• **DebugOptions - Debug-Optionen des Clients**

Definiert die Debug-Optionen des Clients. Der Eintrag wird nur ausgewertet, wenn das Kommandozeilenargument c16DebugOptions beim Starten des Clients nicht angegeben wurde. Ist dieser Eintrag in der Konfigurationsdatei ebenfalls nicht vorhanden, wird die Umgebungsvariable C16DebugOptions geprüft. Folgende Werte können miteinander kombiniert (verodert) werden:

- ◆ 0x0001 - Protokollierung der Aufrufe der Tapi-Schnittstelle.
- ◆ 0x0002 - Protokollierung der PDF-Schnittstelle. Protokolliert werden Aufrufe, die über die PDF-Befehle laufen und solche die beim Drucken über PrtJobClose() mit der Option _PrtJobPdf durchgeführt werden.

Kontakt

- ◆ 0x0004 - Sicherung der Metafiles, die beim Druck bzw. bei dem Befehl PdfInsertMeta() angelegt werden. Die Dateien werden im Pfad der Protokolldateien abgelegt.
- ◆ 0x0020 - Druckjob-Verarbeitung. Die Verarbeitung von PrtJobClose() wird protokolliert.
- ◆ 0x1000 - Protokollierung der Ereignisaufrufe bei Oberflächenobjekten. Zusätzlich muss mit dem Eintrag DebugLogPath der Pfad für die Protokolldateien definiert werden.

DebugOptions=1

• **DebugLogPath - Debug-Protokollpfad für Debug-Ausgaben des Clients**

Der Eintrag definiert den Pfad für die Protokolldateien für den Eintrag DebugOptions. Der Eintrag wird nur ausgewertet, wenn das Kommandozeilenargument c16DebugLogPath beim Starten des Clients nicht angegeben wurde. Ist dieser Eintrag in der Konfigurationsdatei ebenfalls nicht vorhanden, wird die Umgebungsvariable C16DebugLogPath geprüft.

DebugLogPath=C:\CONZEPT 16\Client\Logs\

• **Echo - Ausgabe bei Kennworteingabe**

Mit dieser Angabe wird das Bildschirmecho bei Kennworteingaben in CONZEPT 16 abgeschaltet. Statt der Anzeige von Sternchen wird überhaupt nichts angezeigt. Dies gilt sowohl für Kennwörter von Benutzern als auch für das Kennwort des OEM-Kits.

ECHO=N/Y

• **FontSize - Fontgröße**

Hiermit wird die Größe des benutzten Fonts (horizontal x vertikal) für die zeichenorientierte Oberfläche eingestellt. Ist ein Font der angegebenen Größe nicht verfügbar, wird der nächstkleinere Font ausgewählt. Standardmäßig wird ein Font mit 8 x 14 benutzt.

Nachfolgende Fontgrößen sind verfügbar:

6 x 10 8 x 16
8 x 13 9 x 16
6 x 14 10 x 18
7 x 14 12 x 20
8 x 14 12 x 22
FONTSIZE=9x16

In der grafischen Oberfläche wird der verwendete Font und seine Größe in der Eigenschaft Font bei den Objekten angegeben. Die wählbaren Fontgrößen hängen von dem verwendeten Font ab. Bei einem TrueType-Font können alle Schriftgrößen verwendet werden.

• **FontType - Fonttyp**

In der zeichenorientierten Oberfläche kann mit den Fonttypen TERMINAL

Kontakt

(serifenlos) oder COURIER (Serifen) gearbeitet werden. Dazu wird der Fontname bei FONTTYPE angegeben. Standardmäßig wird TERMINAL benutzt.

Nachfolgende Fonttypen sind verfügbar:

TERMINAL 6 x 10 TERMINAL 8 x 16
TERMINAL 6 x 14 TERMINAL 10 x 18
TERMINAL 7 x 12 TERMINAL 12 x 20
TERMINAL 7 x 13 TERMINAL 12 x 22
TERMINAL 8 x 13 COURIER 7 x 14
TERMINAL 8 x 14 COURIER 9 x 16
FONTTYPE=COURIER

In der grafischen Oberfläche wird der verwendete Font in der Eigenschaft Font bei den Objekten angegeben. Die wählbaren Fonts hängen von den auf dem System installierten Fonts ab.

- **Lang_Display - Sprache der Anzeige auswählen**

In dieser Einstellung wird die Sprache angegeben, in der die Anzeige des Designers und der textbasierten Oberfläche (Standard-Client) erfolgen soll. In der textbasierten Oberfläche stehen die Sprachen Deutsch (DE), Englisch (EN) und Türkisch (TR) zur Auswahl. Im Designer nur Deutsch und Englisch. Ist das Kommandozeilenargument oder die Umgebungsvariable /c16lang gesetzt, wird die Einstellung der Konfigurationsdatei ignoriert. Wird keine Sprache angegeben oder ist der Eintrag auf *U gesetzt, wird die Sprache der Benutzeroberfläche des Betriebssystems verwendet. Wird diese Sprache nicht unterstützt, wird Englisch verwendet.

Lang_Display=DE

- **LocalDs - Verzeichnis der lokalen Datenstruktur**

Wird in den Basisparametern des Standard Clients angegeben, dass eine lokale Datenstruktur angelegt werden soll, wird in dem Verzeichnis, das in diesem Eintrag angegeben wird, eine Datei mit dem Namen der Datenbank und der Dateierweiterung .dst angelegt. Wird die entsprechende Datenbank geöffnet, wird lediglich ein Vergleich der Zeitstempel der Datenstrukturen vorgenommen. Hat sich die Datenstruktur nicht verändert, kann diese aus der Datei geladen werden. Auf diese Weise entfällt eine unter Umständen längere Wartezeit beim Übertragen der Datenstruktur über zum Beispiel eine ISDN-Verbindung. Ist kein Verzeichnis angegeben, wird die lokale Datenstruktur im Verzeichnis des Clients abgelegt.

LocalDs=C:\C16\Datastructures

Eine lokale Datenstruktur kann nur mit einer Enterprise- oder Developer-Edition (siehe Lizenztypen) angegeben werden.

- **LoginGui - Grafischen Login-Dialog verwenden**

Wird dieser Eintrag auf Y gesetzt, wird im Standard-Client zur Anmeldung der grafische Login-Dialog des Advanced-Clients verwendet. Standardmäßig findet

Kontakt

eine Masken-basierte Anmeldung statt.

LoginGui=Y

• **LoginGuiPicture - Hintergrundbild für Login-Dialog**

Im Login-Dialog kann ein Hintergrundbild angezeigt werden. Das Bild kann im Format JPEG, TIFF, PNG, BMP oder GIF vorliegen und wird im Dialog zentriert dargestellt. Ein passgenaues Bild für den Login-Dialog besitzt die Maße 430 x 290 Pixel. Dieser Eintrag gibt Pfad und Dateiname der extern vorliegenden Datei an.

LoginGuiPicture=C:\C16\LoginBackground.png

Im Standard-Client wirkt sich diese Einstellung nur aus, wenn LoginGui aktiviert ist.

• **PasswordRetry - Anzahl der Anmeldeversuche**

Wird bei der Anmeldung an die Datenbank ein falsches Kennwort angegeben, wird der Client beendet. Mit dieser Einstellung kann die Anzahl der Kennwortwiederholungen angegeben werden. Ist PasswordRetry = 0 gesetzt, wird der Client bei einem falschen Kennwort sofort, bei PasswordRetry = 2 erst nach der dritten fehlerhaften Eingabe geschlossen. Es können Werte von 0 bis 9 angegeben werden.

PASSWORDRETRY=0..9

• **PluginPort - Port der Plugin-Schnittstelle**

Wird in diesem Eintrag eine gültige Portnummer (<= 65535) angegeben, wird die Plugin-Schnittstelle des Designers aktiviert. Der Eintrag wird nur ausgewertet, wenn das Kommandozeilenargument c16PluginPort beim Starten des Clients nicht angegeben wurde. Ist dieser Eintrag in der Konfigurationsdatei ebenfalls nicht vorhanden, wird die Umgebungsvariable C16PluginPort geprüft.

PLUGINPORT=1337

• **SaveWindowPos - Fensterposition speichern**

Ist dieser Parameter gesetzt, wird die letzte Fensterposition des CONZEPT 16-Fensters in der Konfigurationsdatei im Eintrag windowpos_win gesichert. Bei erneutem Aufruf wird das CONZEPT 16-Fenster an der letzten gesicherten Position geöffnet.

SAVEWINDOWPOS=Y/N

• **ScreenSize - Bildschirmauflösung**

Damit wird die Anzahl der Spalten und Zeilen der textbasierten Oberfläche eingestellt. Das Minimum beträgt 80 x 25, das Maximum 140 x 60. Zwischen beiden Grenzen kann die Auflösung beliebig gewählt werden.

SCREENSIZE=AxB

Beispiel: SCREENSIZE=90x28

- **ServerAutoStart - Server starten**

Mit diesem Parameter kann ein lokaler CONZEPT 16-Server automatisch beim Start des Clients aktiviert werden. Als Argument wird das Verzeichnis angegeben, in dem sich der Server (C16_serv_*.dll) befindet. Ist bereits ein CONZEPT 16-Server auf dem Rechner aktiv, der CONZEPT 16-Server als Dienst installiert oder wird der Client in einer Terminal-Session gestartet, wird der Parameter ignoriert.

SERVERAUTOSTART=<pfad>

Beispiele:

; Aktuelles Verzeichnis SERVERAUTOSTART=.; Verzeichnis C16SERVERAUTOSTART=C:\C16\SERVER

- **SplashOn - Splash-Screen beim Start anzeigen**

Beim Starten des Standard-Clients (c16_winc.exe) wird ein Splash-Screen angezeigt. Die Anzeige kann mit folgendem Eintrag unterbunden werden:

SplashOn = N

Ist kein Eintrag vorhanden oder SplashOn = Y wird der Splash-Screen angezeigt.

- **TcpConnectTimeout - Verbindungs-Timeout des Clients**

Für die TCP/IP-Verbindungen des Clients zum Server kann eine Maximalzeit eingestellt werden, die der Client beim Verbindungsaufbau zum Server warten darf. Dazu wird clientseitig in der Konfigurationsdatei C16.CFG der Eintrag TcpConnectTimeout=x vorgenommen, wobei x die maximale Wartezeit in Sekunden ist. Diese Einstellung ist primär für den Hot-Standby-Betrieb wichtig, da nach dieser Zeit versucht wird, die Datenbank auf dem Sekundärsystem zu öffnen. Die Einstellung kann aber auch im normalen Betrieb verwendet werden. In diesem Fall bekommt der Anwender eine entsprechende Timeout-Meldung. Ist kein Timeout angegeben, wird der Default-Wert von 10 Sekunden verwendet.

TCPCONNECTTIMEOUT=10

- **TimeFormat - Zeitformat**

Mit der Einstellung TimeFormat wird das Zeitformat in Masken und Dialogen vorgegeben. Das definierte Zeitformat des Betriebssystems wird ignoriert.

Folgendes Format ist möglich:

HH?mm

Es ist auf die Groß- und Kleinschreibung zu achten und das Fragezeichen steht für ein beliebiges Trennzeichen.

TIMEFORMAT=HH:mm

- **Unique - Eindeutiger Benutzername**

Kontakt

Durch diese Angabe wird bei einer Netzwerkversion erzwungen, dass jeder Benutzer einer Datenbank im Mehrplatzbetrieb mit einem anderen Benutzernamen die Datenbank aufruft. Somit wird vermieden, von zwei Arbeitsplätzen aus mit demselben Benutzernamen auf der Datenbank zu arbeiten. Wird versucht, die Datenbank mit einem Benutzernamen aufzurufen, der bereits registriert ist, so erfolgt die Meldung "Benutzer bereits vorhanden.".

UNIQUE=Y/N

• **Upgrade - Datenbankkonvertierung unterbinden**

Mit diesem Parameter kann die automatische Konvertierung von Datenbanken mit älteren Versionen (ab Version 4.7) in eine Datenbank der Version 5.3 verhindert werden. Durch die Eintragung von Upgrade=N erfolgt nicht die Abfrage, ob die Datenbank konvertiert werden soll, sondern lediglich die Meldung "Datenbankversion nicht korrekt.".

UPGRADE=Y/N

• **Win3DMode - 3D-Modus einschalten**

Mit diesem Parameter kann zwischen der 3D-Darstellung und der 2D-Darstellung in der textbasierten Oberfläche gewechselt werden. Der Parameter beeinflusst sowohl die Entwicklungsumgebung, als auch die Applikation. Eingabefelder, Schaltflächen und Listen werden mit einem Rahmen versehen, der ihnen ein drei-dimensionales Aussehen verleiht.

WIN3DMODE=Y/N

• **windowpos_win - letzte Fensterposition**

In dieser Einstellung wird die letzte Fensterposition des Clients abgelegt, wenn der Eintrag savewindowpos=y vorhanden ist. Die Position wird in Pixeln ausgehend von der oberen linken Bildschirmecke angegeben.

WINDOWPOS_WIN=100,100

Kontakt

Konfigurationsdatei - C16_APGL.CFG

Eintragungen in der Konfigurationsdatei

Die Konfigurationsdatei enthält Informationen im ASCII-Format, die das Verhalten von CONZEPT 16 unabhängig von der benutzten Datenbank beeinflussen. Innerhalb der Datei beansprucht jeder Parameter eine Zeile.

Die CONZEPT 16-Konfigurationsdatei kann einen beliebigen Namen haben, lediglich die Endung .cfg ist vorgegeben. So ist es möglich, Starteinstellungen für verschiedene Konfigurationen oder Benutzer vorzunehmen. Welche Konfigurationsdatei beim Start angesprochen wird, kann mit dem Startparameter C16CFG gesetzt werden. Ist nichts angegeben, sucht CONZEPT 16 im Standardpfad für Konfigurationsdateien nach der Datei c16_apgl.cfg. Ist dort keine entsprechende Datei vorhanden, wird in dem Verzeichnis, aus dem der Client gestartet wurde, die Datei gesucht.

Folgende Einträge können in der Konfigurationsdatei vorgenommen werden:

<u>: (Semikolon)</u>	Kommentar
<u>AppLogPath</u>	Pfad des <u>Anwendungsprotokolls</u>
<u>AppLogOptions</u>	Protokollierungsoptionen des <u>Anwendungsprotokolls</u>
<u>DebugOptions</u>	Debug-Optionen des Clients
<u>DebugLogPath</u>	Protokollpfad für Debug-Ausgaben des Clients
<u>Lang_Display</u>	Sprache der Anzeige auswählen
<u>LocalDs</u>	Verzeichnis der lokalen Datenstruktur
<u>LoginGuiPicture</u>	Hintergrundbild für Login-Dialog
<u>PluginPort</u>	Port der <u>Plugin-Schnittstelle</u>
<u>ServerAutoStart</u>	Server starten
<u>TcpConnectTimeout</u>	Verbindungs-Timeout des Clients

Die Eintragungen entsprechen den gleichnamigen Einstellungen in der c16.cfg.

Host-Tabelle - C16.HST

Eintragungen in der Host-Tabelle.

Wird der CONZEPT 16-Client gestartet, verbindet er sich nicht automatisch mit einem CONZEPT 16-Server und einer Datenbank. Der Server kann über eine Liste ausgewählt werden. Danach wird eine Liste der eingetragenen Datenbanken des Servers angezeigt.

In der Serverliste tauchen allerdings nur Server auf, die über das TCP/IP-Protokoll erreicht werden können und in der Host-Tabelle des CONZEPT 16-Clients eingetragen sind. Soll ein anderer Server angesprochen werden, muss die IP-Adresse oder der Rechnername des Datenbankservers manuell in das Eingabefeld eingetragen werden.

Die Host-Tabelle ist die Datei C16.HST im Standardverzeichnis für Konfigurationsdateien. Die Datei wird nicht automatisch bei der Installation des Clients angelegt, sondern kann bei Bedarf mit einem ASCII-Editor erzeugt werden.

Folgende Einträge sind in der Datei C16.HST möglich:

- ; (Semikolon) - Kommentar

Mit dem Zeichen ";" wird eine Kommentarzeile eingeleitet. Die gesamte restliche Zeile wird nicht mehr ausgewertet.

; Kommentar

- TCP/IP - IP-Adresse oder Servername

Mit diesem Eintrag wird ein CONZEPT 16-Server über TCP/IP erreichbar. Dazu wird die IP-Adresse oder (falls ein Domain Name Service vorhanden ist) der Name des Servers angegeben.

TCP:192.168.0.1TCP:vs1



Werden in der Host-Tabelle Server eingetragen, die nicht im Netzwerk erreichbar sind, kommt es zu größeren Verzögerungen beim Aufbau der Serverliste. Sollte ein Server nicht zur Verfügung stehen, sollte daher der Eintrag in der Host-Tabelle auskommentiert werden.

Kontakt

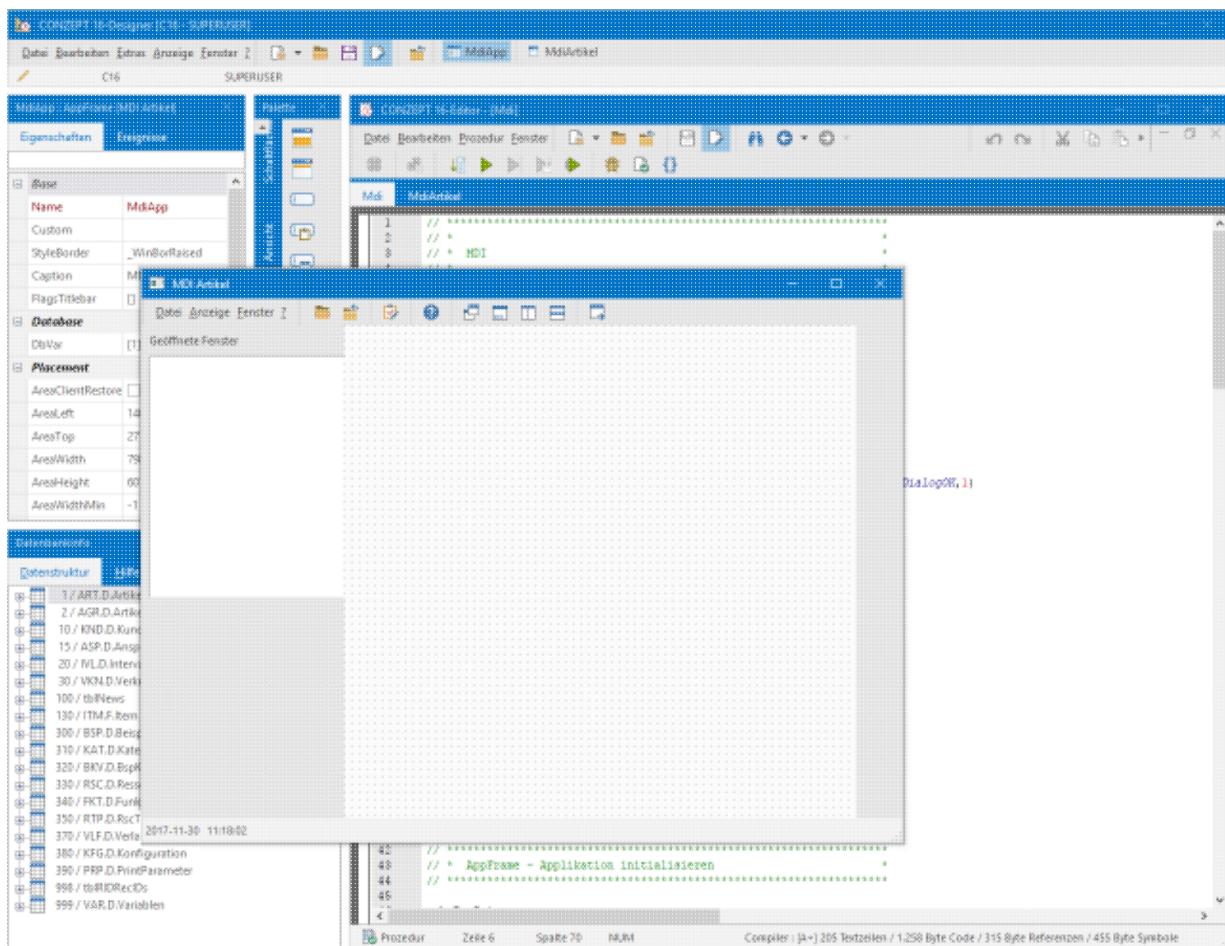
CONZEPT 16-Designer

Mit dem Designer können Anwendungen für eine CONZEPT 16-Datenbank geschrieben werden.

Siehe [Objektausrichtung](#)
[\(Blog\)](#)

Der Designer ist integraler Bestandteil der Entwicklungsumgebung. Mit ihm wird die Benutzerschnittstelle der Anwendung bestehend aus Standard-, Dialog- und MDI-Fenstern (Multiple Document Interface) definiert und mit entsprechenden Standardattributten wie Titel, Rahmentyp, Systemmenü usw. versehen.

Der Designer besteht aus einer Reihe von Werkzeugfenstern. In diesen Fenstern werden Informationen und Werkzeuge dargestellt. Alle Fenster können auch bei Bedarf ein- oder ausgeblendet werden. Die Entwicklungsumgebung kann variabel an die Bedürfnisse des jeweiligen Entwicklungsstadiums angepasst werden.



Der Designer gliedert sich in mehrere Bildschirmelemente. In der Menüzeile werden alle Elemente über den Menüeintrag Anzeige ein- oder ausgeblendet. Wird der Designer geschlossen, wird der Zustand der Oberfläche für jeden Benutzer in der Datenbank gespeichert. Bei einem erneuten Aufruf des Designers durch diesen Benutzer, werden alle zuletzt geöffneten Fenster wieder geöffnet.

Die Entwicklungsoberfläche besteht aus verschiedenen Fenstern:

- [Menüzeile](#)
- [Symbolleisten](#)
- [Eigenschaften](#)
- [Objekte](#)
- [Palette](#)
- [Anwendungsfenster](#)
- [Editor](#)
- [Assistent](#)
- [Datenbankinfo](#)
- [Taschenrechner](#)

Verschiedene Oberflächen-Objekte werden über eigene Objekt-Editoren bearbeitet:

- [Menü-Editor](#)
- [DataList-Editor](#)
- [DataListPopup-Editor](#)
- [RecList-Editor](#)
- [RecListPopup-Editor](#)
- [RecView-Editor](#)
- [GanttGraph-Editor](#)
- [TreeView-Editor](#)
- [Toolbar-Editor](#)
- [StatusBar-Editor](#)
- [DateEditCalendar-Editor](#)

Die verschiedenen Parameter der Entwicklungsumgebung werden über den Menüeintrag **Datei / Einstellungen** konfiguriert.

Die Anordnung der Werkzeugfenster, sowie die Position und Größe der Dialoge zum Öffnen und Speichern von Dialogen, Menüs usw., werden benutzerbezogen in der Datenbank gespeichert. Beim erneuten Aufrufen des Designers wird der vorherige Zustand des Designers wieder hergestellt.

In fast allen Bereichen kann mit der Windows-Zwischenablage gearbeitet werden, um zum Beispiel Text in den [Editor](#) einzufügen. Im Bereich der verschiedenen Editoren können untergeordnete Objekte (Menü-Einträge, Spalten von Listen, Buttons usw.) in die Zwischenablage kopiert und von dort aus in ein anderes gleichartiges Objekt eingefügt werden. Die Selektion der Objekte erfolgt mit den üblichen Tastaturkommandos (Selektieren mit der Maus, Hinzufügen/Entfernen von Objekt mit gedrückter -Taste, Markieren von Bereichen mit gedrückter -Taste).

Datenstruktureditor

Erstellen und ändern der Datenstruktur

Menü

Siehe Bearbeiten,

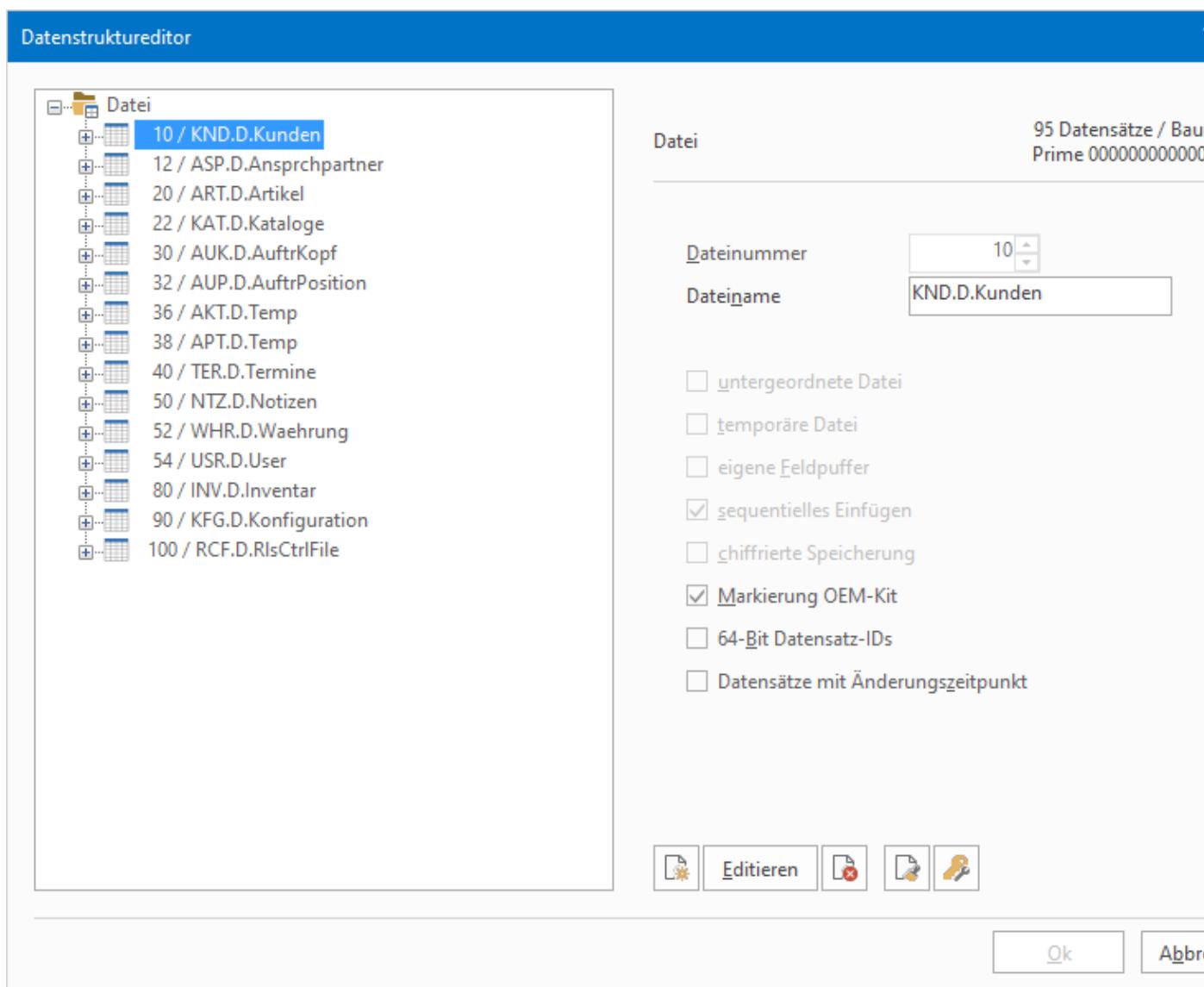
Konventionen

Die Datenstruktur kann über den Menüeintrag **Extras / Datenstruktureditor...** aufgerufen werden.

Bei Änderungen in der Datenstruktur ist zu beachten, dass die neuen Elemente erst nach einem Aufbau der Datenstruktur zur Verfügung stehen. Die Datenstruktur wird durch Schließen des Datenstruktureeditors neu aufgebaut. Die neue Datenstruktur wird dann an jeden Client übertragen, der sich ab diesem Zeitpunkt an der Datenbank anmeldet. Eine Ausnahme bildet der Client, der die Datenstruktur neu aufgebaut hat, er besitzt automatisch die neue Struktur. Damit im laufenden Betrieb die neue Datenstruktur auch von anderen Clients genutzt werden kann, müssen sich diese Clients neu anmelden.

Nach dem Aufruf des Menüeintrages wird folgender Dialog dargestellt:

Kontakt



In der linken Hälfte des Dialoges kann auf die Elemente der Datenstruktur zugegriffen werden. Wird ein Element selektiert, werden in der rechten Hälfte Informationen über das Element dargestellt. Bei Auswahl von "Datei" werden Informationen zur Datenstruktur (Anzahl der Dateien, der Schlüssel, Größe der Datenstruktur usw.) dargestellt. Der Eintrag kann durch Doppelklicken aufgeklappt werden und zeigt dann alle bereits existierenden Dateien an.

Wird in dem Baum die Taste **F3** oder die Tastenkombination **Strg**+**F3** gedrückt, kann in einem Eingabefeld ein Name aus der Datenstruktur angegeben werden. Nach Abschließen der Eingabe mit **Enter** wird auf das entsprechende Strukturelement (Datei, Teildatensatz, Feld, Schlüssel usw.) positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten **F3** und **Home**+**F3** wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Datenstruktur gesucht. Ist das Ende bzw. der Anfang der Datenstruktur erreicht, wird die Suche am jeweils anderen Ende

Kontakt

fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Alle Elemente in denen Änderungen vorgenommen wurden, die noch nicht in der Datenstruktur enthalten sind, werden in einem anderen Font und mit einem grünen Symbol gekennzeichnet. Erst nach Neuaufbau der Datenstruktur können diese Elemente von einem neu angemeldetem Client verwendet werden.

Wurden Änderungen an Schlüsseln in Dateien mit Datensätzen vorgenommen, müssen entweder alle Schlüssel oder die betroffenen Schlüssel über die Schaltfläche reorganisiert werden.

Das Erstellen oder Ändern der Datenstruktur unterteilt sich in folgende Bereiche:

- Datei bearbeiten
- Teildatensatz bearbeiten
- Feld bearbeiten
- Schlüssel bearbeiten
- Verknüpfung bearbeiten

Mit dem Schließen des Dialoges mit der Schaltfläche [Übernehmen] wird die Datenstruktur neu aufgebaut. Ab diesem Zeitpunkt wird die neue Datenstruktur an neu angemeldete Clients übertragen.



Um sicher zu gehen, dass alle mit der gleichen Datenstruktur arbeiten, müssen sich alle Clients neu anmelden.

Datei bearbeiten

Anlegen, ändern und löschen von Dateien

[Datenstruktureditor](#),

[Teildatensatz](#)

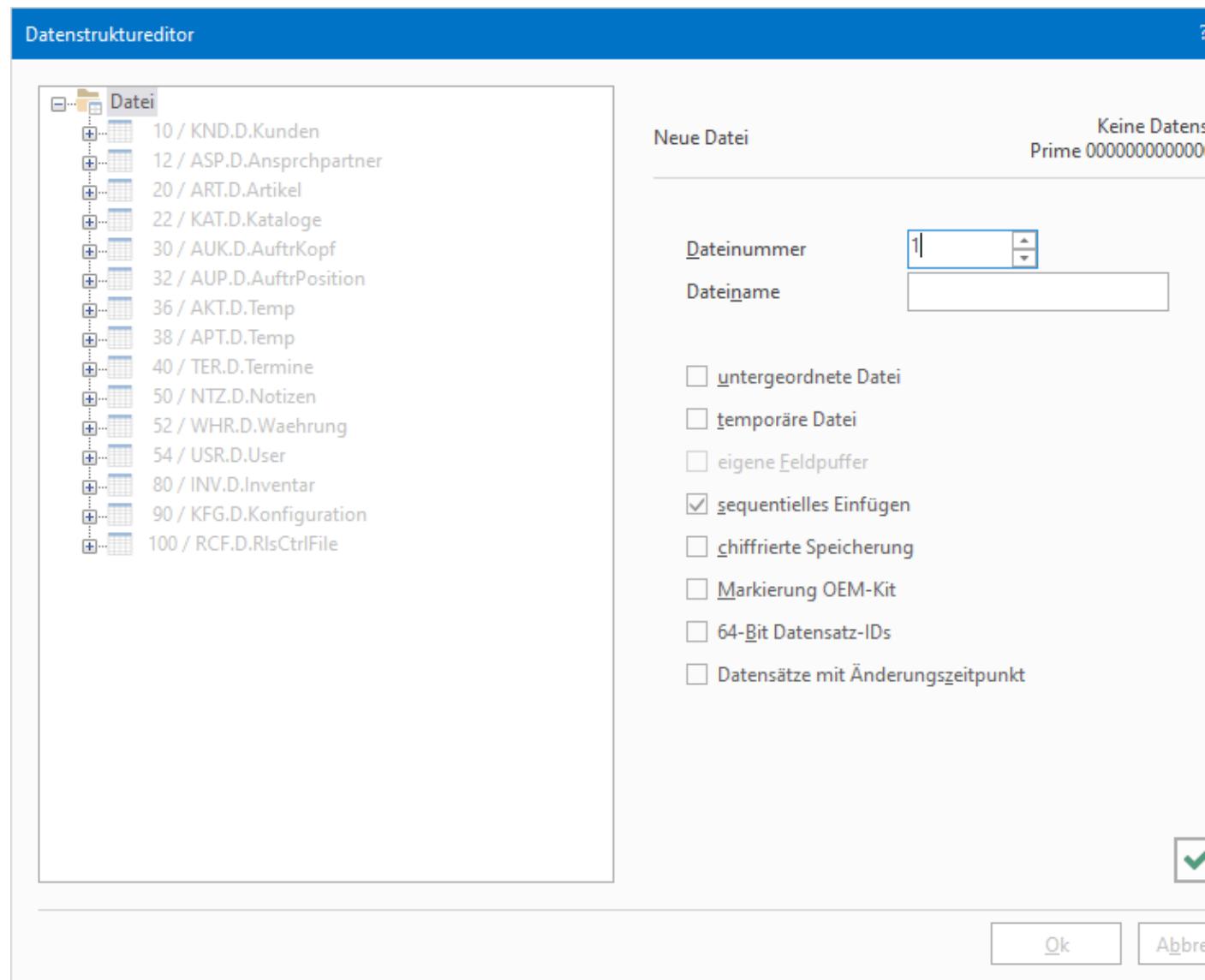
Siehe [bearbeiten](#),

[Temporäre Tabellen](#)

[\(Blog\)](#)

Neue Datei anlegen

Zum Anlegen einer neuen Datei muss auf der linken Seite der Eintrag "Datei" ausgewählt und die Schaltfläche  gedrückt werden.



Durch Drücken der Schaltfläche  kann der Vorgang abgebrochen werden. Die neue Datei wird erst nach Drücken der Schaltfläche  angelegt.

Kontakt

Die neue Datei steht nicht sofort zur Verfügung. Erst, wenn der Dialog mit der Schaltfläche [Übernehmen] geschlossen wird, erfolgt ein Neuaufbau der Datenstruktur. Sobald sich nach diesem Zeitpunkt ein Client an der Datenbank anmeldet, verfügt er ebenfalls über die neue Datenstruktur.

• Dateinummer

Im Feld "Dateinummer" wird die Nummer der Datei angegeben. Es kann eine beliebige Nummer zwischen 1 und 999 gewählt werden, die noch nicht für eine andere Datei verwendet wurde. Weitere Angaben müssen nicht erfolgen.

Die Dateinummer dient zur Identifizierung der Datei. Sie muss bei allen dateispezifischen Operationen (z. B. Datensatz lesen oder Datensatz speichern) angegeben werden.

• Dateiname

Einer Datei kann ein Name zugeordnet werden. Der Name kann dann anstelle der Dateinummer angegeben werden. Durch die Verwendung von Namen wird der Quellcode besser lesbar und die Pflege des Systems erheblich vereinfacht. Der Name der Datei muss wie die Dateinummer, einmalig in der Datenbank sein.

Der Name darf maximal 20 Zeichen lang sein.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden. Die Regeln zur Namensvergabe von Datenstrukturelementen werden im Abschnitt Konventionen erläutert.

• untergeordnete Datei

Für die Verwendung von untergeordneten Dateien besteht ab der Version 4.4 keine Notwendigkeit mehr. Der Parameter kann bei bestehenden Dateien nicht verändert werden. Die Datenstruktur von untergeordneten Dateien kann nicht verändert werden.

Untergeordnete Dateien wurden benötigt, um unterschiedliche Sichtweisen auf einen Datenbestand zu realisieren, da Masken eng mit der Datenstruktur verbunden sind. Durch die Verwendung von Dialogen können beliebig viele unterschiedliche Fenster, die den Inhalt einer Datei anzeigen, generiert werden.

• temporäre Datei

Wird eine Datei als temporäre Datei angelegt, werden keine Dateiinhalte und Schlüsseleinträge in die Datenbank gespeichert. Alle Inhalte verbleiben im Cache der Datenbank und werden beim Schließen der Datenbank automatisch gelöscht. Maximal 80% des Datenbankcaches kann zur Speicherung von temporären Dateien verwendet werden (siehe Konfiguration des Servers). Pro GB Datenvolumen in temporären Dateien werden zusätzlich zum Cache 14 MB Hauptspeicher benötigt.

Eine bestehende Datei kann in eine temporäre Datei umgewandelt werden, wenn keine Datensätze in der Datei gespeichert sind.

- Die Durchführung einer Optimierung der Datenbank führt zum Löschen des Datenbankcaches, dadurch werden ebenfalls die Inhalte in den temporären Tabellen geleert.
- Wird eine bestehende Datei in eine temporäre Datei umgewandelt, werden beim Einspielen einer Definitionsdatei mit dieser Datenstruktur in eine andere Datenbank alle Datensätze in der umgewandelten Datei gelöscht!

• **eigene Feldpuffer**

Wie bei den untergeordneten Dateien besteht auch für die Feldpuffer keine Notwendigkeit mehr. Über die Prozedurbefehle der Familie RecBuf...() können zur Laufzeit Feldpuffer erzeugt werden.

• **sequentielles Einfügen**

Mit diesem Dateiattribut wird die Speicherung von Datensätzen in der Datei beeinflusst. Im Normalfall werden neue Sätze "sequentiell" eingefügt. Neue Datensätze werden dabei hintereinander ans Ende geschrieben.

Ist dieses Attribut nicht gesetzt, werden die Datensätze gestreut eingefügt, d. h. die Datensätze werden zwischen die vorhandenen Datensätze geschrieben.

Dieses Dateiattribut kann nur bei leerer Datei verändert werden. Beim Anlegen einer neuen Datei wird dieses Attribut automatisch gesetzt.

• **chiffrierte Speicherung**

Dieser Parameter lässt sich nur bei leerer Datei verändern. Wird er gesetzt, werden die Datensätze in chiffrierter Form in der Datenbank gespeichert. Somit lassen sich Daten außerhalb von CONZEPT 16 nicht mehr erkennen, was die Sicherheit vonbrisanten Daten erhöht.

- ⚠ Die chiffrierte Speicherung betrifft nur die Tabelleninhalte. In Schlüsseln werden die Werte immer in Klartext gespeichert. Somit sollten Schlüssel nur über nicht schützenswerte Felder gebildet werden.

• **Markierung OEM-Kit**

Wird dieser Parameter gesetzt, werden die Datensätze für die Auslagerung mit dem OEM-Kit markiert. Auf diese Weise kann der Inhalt einer Datei mit einem Datenbank-Update in mehrere Datenbanken eingelesen werden.

• **64-Bit Datensatz-IDs**

Durch Aktivieren dieses Parameters werden Datensatz-IDs in der Datei mit bis zu 64-Bit gespeichert. Er sollte aktiviert werden, wenn in der Datei viele Einfüge- und Löschoperationen durchgeführt werden. Er kann unter folgenden Bedingungen aktiviert werden:

- ♦ sequentielles Einfügen ist **aktiviert**
- ♦ untergeordnete Datei ist **deaktiviert**

Kontakt

- ◆ Der Prime-Counter der Datei **muss kleiner** als 0xFB000000 sein. Dieser wird rechts oben im Dialog angezeigt.
- **Datensätze mit Änderungszeitpunkt**

Wird dieser Parameter gesetzt, dann wird der Änderungszeitpunkt beim Hinzufügen und Ändern von Datensätzen im Datensatz gespeichert. Maßgeblich ist hierbei die aktuelle Uhrzeit des Datenbank-Servers (UTC).

Mit dem Befehl RecInfo() kann der Zeitstempel eines Datensatzes gelesen werden.

In der Datensatzverwaltung kann der Zeitstempel eines Datensatzes eingesehen werden.

Datei ändern

Zum Ändern einer Datei muss auf der linken Seite die Datei ausgewählt und die Schaltfläche  gedrückt werden.

Durch Drücken der Schaltfläche  kann der Vorgang abgebrochen werden. Die Änderungen an der Datei werden erst nach Drücken der Schaltfläche  gespeichert.

Sind in der Datei bereits Datensätze enthalten, kann nur die "Markierung für das OEM-Kit" verändert werden.

Die Änderungen stehen nicht sofort zur Verfügung. Erst, wenn der Dialog mit der Schaltfläche [Ok] geschlossen wird, erfolgt ein Neuaufbau der Datenstruktur. Sobald sich nach diesem Zeitpunkt ein Client an der Datenbank anmeldet, verfügt er ebenfalls über die neue Datenstruktur.

Datei löschen

Vorhandene Dateien können über die Schaltfläche  gelöscht werden.

 Mit Löschen der Struktur werden ebenfalls alle Datensätze der Datei gelöscht.

Teildatensatz bearbeiten

Anlegen, ändern und löschen von Teildatensätzen

Datei

bearbeiten,

Siehe Feld

bearbeiten

Teildatensätze bieten die Möglichkeit die Felder eines Datensatzes zu gruppieren. Innerhalb eines Kunden-Datensatzes können so Felder der Adresse und Felder der Bankverbindung in unterschiedliche Teildatensätze gespeichert werden.

Darüber hinaus kann die Aktivität der Teildatensätze gesteuert werden. Dadurch entsteht eine variable Datenstruktur, in der Varianten gespeichert werden können. Zum Beispiel können in der Datei für Kundendaten ebenfalls Privatadressen gespeichert werden. In diesem Fall wird der Teildatensatz für die Bankverbindung deaktiviert und ein Teildatensatz mit Geburtstag und Hobby aktiviert.

Teildatensatz anlegen

Nachdem der Eintrag einer bereits erstellten Datei aufgeklappt wurde, kann der Eintrag "Teildatensatz" angeklickt werden. Ein neuer Teildatensatz wird über die Schaltfläche  angelegt.

Kontakt

Datenstruktureditor

Neuer Teildatensatz

Name

bedingter Teildatensatz

feldabhängig

Bezogener TDS

Bedingungsfeld

Ok Abbrechen

Durch Drücken der Schaltfläche wird der Vorgang abgebrochen. Der neue Teildatensatz wird erst nach Drücken der Schaltfläche angelegt.

Der neue Teildatensatz steht nicht sofort zur Verfügung. Erst, wenn der Dialog mit der Schaltfläche [Übernehmen] geschlossen wird, erfolgt ein Neuaufbau der Datenstruktur. Sobald sich nach diesem Zeitpunkt ein Client an der Datenbank anmeldet, verfügt er über die neue Datenstruktur.

• Name

In diesem Eingabefeld kann dem Teildatensatz ein Name gegeben werden. In der Programmierung erfolgt die Referenz auf den Teildatensatz über seine Nummer (wird automatisch vergeben) oder seinen Namen. Die Verwendung von Namen führt zu einem wesentlich einfacher zu pflegendem System. Der Name muss in der Datenbank eindeutig sein.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden. Weitere Informationen zur Vergabe von Namen befinden sich im Abschnitt Konventionen.

• **Bedingter Teildatensatz**

Alle Teildatensätze, die nach dem ersten Teildatensatz angelegt werden, können bedingte Teildatensätze sein. Ein bedingter Teildatensatz wird nur dann zusammen mit dem Datensatz gespeichert, wenn er aktiv ist. Der Teildatensatz kann über einen Prozedurbefehl (TdsState()) oder den Inhalt eines Feldes aktiviert werden. Die Felder "Feldabhängig", "Bezogener Teildatensatz" und "Bedingungsfeld" sind nur bei einem bedingten Teildatensatz aktiv.

• **Feldabhängig**

Ist dieser Parameter gesetzt, wird die Aktivität des Teildatensatzes über den Inhalt eines Feldes gesteuert. Das Feld wird in den Parametern "bezogener Teildatensatz" und "Bedingungsfeld" angegeben.

• **Bezogener Teildatensatz**

In diesem Parameter wird der Teildatensatz angegeben, in dem das Bedingungsfeld enthalten ist. Das Bedingungsfeld kann nicht im Teildatensatz enthalten sein, dessen Aktivität es steuert. Der entsprechende Teildatensatz kann aus einer Liste ausgewählt werden.

• **Bedingungsfeld**

Hier wird das Bedingungsfeld angegeben. Es kann aus allen Feldern des in "Bezogener Teildatensatz" angegebenen Teildatensatzes ausgewählt werden.

Teildatensätze ändern

Die Parameter eines Teildatensatzes können nach der Auswahl des Teildatensatzes über die Schaltfläche verändert werden.

Teildatensätze löschen

Vorhandene Teildatensätze, die keine Felder beinhalten, und in dessen Datei keine Datensätze vorhanden sind, können über die Schaltfläche gelöscht werden.

Kontakt

Feld bearbeiten

Anlegen, ändern und löschen von Feldern

Teildatensatz

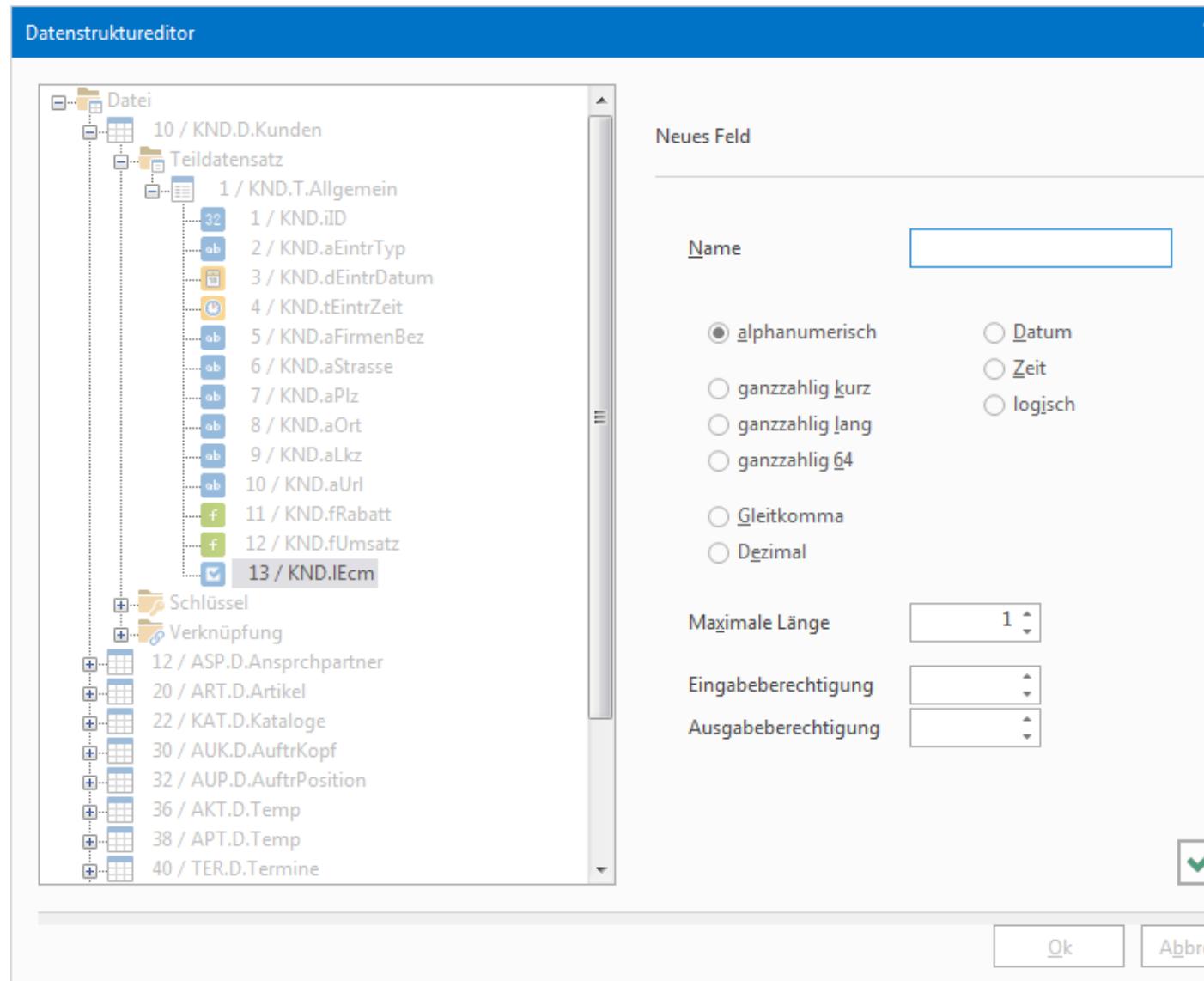
bearbeiten

Siehe [Schlüssel](#)

bearbeiten

Wird ein bereits bestehender Teildatensatz selektiert und die Schaltfläche  gedrückt, wird ein neues Feld angelegt. Das Feld wird erst beim Drücken der

Schaltfläche  angelegt. Der Vorgang kann durch Drücken der Schaltfläche  abgebrochen werden.



• Name

Im Feld "Name" ist der Name des neu anzulegenden Feldes anzugeben. Eine Eingabe ist an dieser Stelle nicht zwingend. In der Programmierung wird der Inhalt des Feldes über seinen Namen referenziert. Bei der Verwendung von

"sprechenden" Namen wird der Quelltext wesentlich lesbarer.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden. Weitere Informationen zur Vergabe von Namen befinden sich im Abschnitt Konventionen.



Der Name des Feldes muss in der Datenbank eindeutig sein.

• **Feldtyp**

Der Feldtyp entscheidet darüber, welche Daten in dem Feld gespeichert werden können. Jeder Feldtyp entspricht einem Datentyp. Alle Feldtypen mit Ausnahme des Typs alpha werden mit fester Länge gespeichert. Inhalte vom Typ alpha werden nur in ihrer tatsächlichen Länge in der Datenbank abgelegt.

• **Maximale Länge**

Ist das Feld vom alphanumerischen Typ, so wird im Textfeld "Maximale Länge" die Definitionslänge angegeben (zwischen 1 und 8192 Zeichen). In textorientierten Anwendungselementen (Masken, Listenformate, Zugriffslisten etc.) können nur die ersten 250 Zeichen des alpha-Feldes angezeigt oder eingegeben werden. Bei Transfers kann die volle Länge importiert oder exportiert werden.

• **Eingabeberechtigung**

Im Feld "Eingabeberechtigung" kann eine Berechtigungsstufe zwischen 0 und 255 eingegeben werden. Dadurch können nur Benutzer, die für diese Datei mindestens eine Berechtigung gleich oder größer der angegebenen besitzen, Eingaben in diesem Feld vornehmen.

• **Ausgabeberechtigung**

Mit dem Eintrag im Feld "Ausgabeberechtigung" wird festgelegt ab welcher Berechtigungsstufe der Benutzer das Feld angezeigt bekommt. Zulässig sind Werte zwischen 0 und 255.

Felder ändern

Die Parameter eines Feldes können nach der Auswahl des Feldes über die

Editieren

Schaltfläche verändert werden. Befinden sich Datensätze in der Datei, können nur Änderungen an der Ein- und Ausgabeberechtigung vorgenommen werden. Darüber hinaus kann nur die Eingabelänge von Felder vom Typ alpha verlängert werden.

Sollen weitere Änderungen vorgenommen werden, muss der Inhalt der Datei gesichert (zum Beispiel mit einem Transfer) und anschließend gelöscht werden. Dann können Änderungen in der Datenstruktur vorgenommen werden. Beim Zurückspielen der Datensätze müssen entsprechend der Änderungen in der Datenstruktur die Datensätze angepasst werden.

Felder löschen

Kontakt

Felder können nur gelöscht werden, wenn keine Datensätze in der Datei gespeichert sind. Es kann die gleiche Vorgehensweise, wie beim Ändern von Feldern verwendet werden. Nachdem das Feld ausgewählt wurde, kann es mit der Schaltfläche gelöscht werden.

Liste von Feldern importieren

Auf einem existierendem, leeren Teildatensatz ist es mit der Tastenkombination  +  möglich eine Liste von Feldern zu importieren. Diese Liste muss als CSV-Datei in Format <Feldname>,<Feldtyp>[,<Feldlänge> vorliegen. Die Feldlänge ist nur bei alpha-Feldern anzugeben. Pro hinzuzufgendem Feld muss eine Zeile in der CSV-Datei eingetragen sein. Folgende Feldtypen sind zulässig:

- alpha
- word / int16
- long / int32
- bigint / int64
- float
- decimal
- date
- time
- logic

Schlüssel bearbeiten

Anlegen, ändern und löschen von Schlüsseln

Feld

bearbeiten,

Verknüpfung

Siehe bearbeiten,

Schlüsselwerte

und -attribute

(Blog)

In der Entwicklungsumgebung kann die Datenstruktur über den Menüeintrag **Extras / Datenstruktureditor** bearbeitet werden.

Wird in der Datenstruktur auf der linken Seite ein Schlüssel selektiert, werden in der rechten Hälfte Informationen über den Schlüssel dargestellt. Wird der Eintrag "Schlüssel" einer Datei durch Doppelklicken aufgeklappt, werden alle bereits existierenden Schlüssel angezeigt. Existierende Schlüssel können ebenfalls durch Doppelklicken aufgeklappt werden und zeigen dann die Schlüsselfelder des Schlüssels an.

Alle Schlüssel oder Schlüsselfelder in denen Änderungen vorgenommen wurden, werden in einem anderen Font und mit einem grünen Symbol gekennzeichnet. Erst nach Neuaufbau der Datenstruktur und gegebenenfalls einer Reorganisation des Schlüssels können diese Schlüssel von einem neu angemeldetem Client verwendet werden.

Schlüssel anlegen

Um einen neuen Schlüssel zu erzeugen, muss der Eintrag "Schlüssel" unter der Datei, in welcher der Schlüssel angelegt werden soll, oder ein bereits bestehender Schlüssel angeklickt und die Schaltfläche  gedrückt werden.

Der Schlüssel wird erst angelegt, wenn die Schaltfläche  gedrückt wird. Der gesamte Vorgang kann durch Drücken der Schaltfläche  abgebrochen werden.

• Name

Hier kann der Name des Schlüssels angegeben werden. Der Name kann in der Programmierung als Referenz auf diesen Schlüssel verwendet werden. Wird kein Name angegeben, kann der Schlüssel nur über seine Nummer angesprochen werden. Die Nummer des Schlüssels wird automatisch vergeben.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden. Weitere Informationen zur Vergabe von Namen befinden sich im Abschnitt Konventionen.

• Eindeutig

Kontakt

Durch Aktivierung des Kontrollkästchens "eindeutig" kann die Eindeutigkeit eines Schlüssels erzwungen werden. Der erste Schlüssel jeder Datei ist immer eindeutig, bei ihm kann dieses Attribut nicht verändert werden. Alle weiteren definierten Schlüssel sind üblicherweise nicht eindeutig, d. h. es sind mehrere Datensätze mit den gleichen Schlüsselwerten zulässig. Wird das Attribut "eindeutig" aktiviert, kann später in der Datei immer nur ein Datensatz mit einem bestimmten Schlüsselwert existieren. Dies muss berücksichtigt werden, wenn beim Einfügen von neuen Daten die Meldung "Datensatz ist schon gespeichert" erscheint. Bei mehreren eindeutigen Schlüsseln kann jeder dieser eindeutigen Schlüssel mit einem bereits gespeicherten Satz kollidieren.

Sind bereits Daten in der Datei gespeichert, kann die nachträgliche Änderung von "nicht eindeutig" auf "eindeutig" dazu führen, dass der Index des Schlüssels doppelte Werte enthalten würde. Dieser Fall wird bei der erforderlichen Reorganisation des Schlüssels festgestellt. Der Schlüssel kann dann nur benutzt werden, wenn das Attribut wieder auf "nicht eindeutig" geändert wird und eine erneute Reorganisation stattfindet.

- **Listeneintrag**

Durch Aktivieren des Kontrollkästchens "Listeneintrag" wird sichergestellt, dass der Schlüssel in der Schlüsselliste der Verwaltung angezeigt wird.

- **Lange Schlüsselwerte**

Durch Aktivieren des Kontrollkästchens "Lange Schlüsselwerte" umfasst der Schlüssel bis zu 950 Zeichen. Standardmäßig sind bis zu 240 Zeichen möglich. Die einzelnen Schlüsselfelder haben unabhängig davon immer eine Begrenzung von maximal 255 Zeichen.

Nach einer Änderung der Option wird eine Schlüsselreorganisation durchgeführt.

Einfügen von Schlüsselfeldern

Um ein neues Schlüsselfeld zu erzeugen, muss der Schlüssel, dem ein Schlüsselfeld zugeordnet werden soll, oder ein bereits bestehendes Schlüsselfeld angeklickt und die Schaltfläche  gedrückt werden.

Das Schlüsselfeld wird erst angelegt, wenn die Schaltfläche  gedrückt wird. Der gesamte Vorgang kann durch Drücken der Schaltfläche  abgebrochen werden.

- **Teildatensatz**

In diesem Feld wird der Teildatensatz des Schlüsselfeldes ausgewählt. Die Felder des ersten Schlüssels müssen aus dem ersten Teildatensatz stammen. Eine Auswahl des Teildatensatzes ist daher für den ersten Schlüssel nicht möglich.

- **Feld**

Kontakt

Nach der Auswahl des Teildatensatzes kann in diesem Feld das Schlüsselfeld ausgewählt werden. Die Auswahl beschränkt sich auf den in "Teildatensatz" ausgewählten Teildatensatz.

- **Absteigende Sortierung**

Damit kann definiert werden, ob das Feld im Schlüssel aufsteigend oder absteigend sortiert wird.

- **Groß-/Kleinwandlung** (Nur bei alphanumerischen Feldern)

Wird dieser Parameter aktiviert, so werden alle Kleinbuchstaben für den Schlüsseleintrag in Großbuchstaben umgewandelt. Dies ermöglicht einen von der Schreibweise unabhängigen Zugriff.

- **Umlaute in alphabetischer Sortierung** (Nur bei alphanumerischen Feldern)

Ist "Groß-/Kleinschreibung" aktiviert, folgt dieser Parameter. Mit dieser Option werden die Umlaute "ä", "ö", "ü" und "ß" wie "ae", "oe", "ue" und "ss" behandelt. Dies kann für sortierte Listen wesentlich sein, da die normale Sortierung nach ASCII-Werten erfolgt (Umlaute folgen erst nach Z).

- **Ohne Sonderzeichen** (Nur bei alphanumerischen Feldern)

Mit dieser Option können alle Zeichen, die nicht 0 bis 9 oder A bis Z sind, bei der Schlüsselerzeugung ignoriert werden. Für den Zugriff wird somit die Stellung von Kommata, Leerzeichen oder ähnlichem unwichtig.

- **SOUNDEX 1**

Diese Funktion ermöglicht eine "phonetische" Suche. Doppelte Buchstaben werden durch einfache Buchstaben ersetzt. Ähnlich lautende Buchstabenfolgen werden gleichgesetzt, um auf einen Datensatz wie z. B. "Meier, Josef" zuzugreifen, unabhängig von seiner Schreibweise (auch "Mayer, Joseph" oder "Maier, Yossef" würden den richtigen Satz finden). Detaillierte Informationen zur Zeichenumwandlung mit Soundex befinden sich in der Beschreibung der Konstante StrSoundex1.

- **SOUNDEX 2**

In dieser Stufe werden zusätzlich zu Soundex 1 noch weitere Umsetzungen durchgeführt. Detaillierte Informationen befinden sich in der Bechreibung der Konstanten StrSoundex2.



Wird die Soundex-Funktion benutzt, ist die Sortierfolge des Schlüssels ungeeignet für Listen oder Selektionen. In diesem Fall kann man einen weiteren Schlüssel ohne SOUNDEX über das gleiche Feld definieren.

- **Maximale Länge**

In diesem Dialogfeld wird die Eingabe einer "Maximalen Länge" verlangt. Damit kann die Anzahl der für die Schlüsselerzeugung relevanten Zeichen reduziert werden (eventuell bei langen Schlüsseln notwendig). Im Normalfall wird hier 0 eingegeben. Bei 0 wird die definierte Feldlänge benutzt.



Kontakt

Bei eindeutigen Schlüsseln sollten keine Gleitkommafelder verwendet werden. Durch die Umwandlung vom dezimalen in das binäre Zahlensystem entstehen Ungenauigkeiten, die unter Umständen dazu führen können, dass ein Datensatz nicht gespeichert oder gefunden werden kann. Gerade bei Werten mit einer hohen Anzahl von Stellen kommt es zu Rundungsfehlern. Zu beachten ist ebenfalls, dass eine Berechnung des Schlüsselwertes immer mit einem Fehler behaftet ist.

Die maximale Länge des Schlüsselwertes ist auf 240 Bytes beschränkt. D. h. besteht ein Schlüssel aus einem Feld ganzzahlig lang und einem alphanumerischen-Feld, können nur die ersten 235 Zeichen des alpha-Feldes unterschieden werden. Von den 240 Bytes werden 4 Bytes für das ganzzahlige Feld benötigt und 235 Byte (plus ein längenbegrenzendes Byte) für das alpha-Feld.

Die Limitation besteht auch, wenn die Option "Lange Schlüsselwerte" aktiviert ist. Nur liegt hier die Gesamtlänge bei 950 Byte.



Befindet sich in dem Schlüssel ein alpha-Feld, dessen Wert die noch zur Verfügung stehenden Bytes ausfüllt, werden nachfolgende Schlüsselfelder nicht mehr berücksichtigt. Um dies zu verhindern, kann die maximale Länge in dem entsprechenden Dialog-Feld angegeben werden.

Ändern von Schlüsselfeldern

Die Eigenschaften eines Schlüsselfeldes können nach der Auswahl des Feldes über die Schaltfläche verändert werden. Es können alle Eigenschaften des Schlüsselfeldes geändert werden. Befinden sich in der betreffenden Datei Datensätze, kann der Schlüssel erst nach einer Reorganisation benutzt werden.

Löschen von Schlüsselfeldern

Das selektierte Schlüsselfeld kann durch Drücken der Schaltfläche gelöscht werden. Befinden sich in der betreffenden Datei Datensätze, kann der Schlüssel erst nach einer Reorganisation benutzt werden.

Schlüssel ändern

Eigenschaften des Schlüssels können nach der Auswahl des Schlüssels über die Schaltfläche verändert werden. Es können alle Eigenschaften des Schlüssels geändert werden. Lediglich der erste Schlüssel ist immer ein eindeutiger Schlüssel.

Schlüssel löschen

Der selektierte Schlüssel kann durch Drücken der Schaltfläche gelöscht werden.

Schlüssel reorganisieren

Die Schlüsselreorganisation wird über die Schaltfläche angestoßen.

Nach Drücken der Schaltfläche werden entweder alle Schlüssel der Datenbank (Eintrag "Datei" wurde selektiert), alle Schlüssel einer Datei (eine Datei wurde

Kontakt

selektiert) oder der betreffende Schlüssel angezeigt.

Alle Schlüssel, die reorganisiert werden müssen, besitzen in der Spalte "*" einen Haken. Sollen einzelne Schlüssel nicht reorganisiert werden, können sie von der Reorganisation ausgenommen werden, indem der Haken in der Spalte "*" durch einen Doppelklick entfernt wird. Der Fortschritt der Reorganisation wird in einem Dialog angezeigt.



Die Reorganisation von Schlüsseln benötigt in Abhängigkeit der Anzahl der Datensätze in der betreffenden Datei Zeit. Durch Vergrößerung des Datenbankpuffers kann die benötigte Zeit verringert werden.

Verknüpfung bearbeiten

Anlegen, ändern und löschen von Verknüpfungen

Schlüssel

Siehe [bearbeiten](#),

Datenstruktur

Verknüpfungen sind ein Bestandteil der Datenstruktur. Sie können über den Menüeintrag **Extras / Datenstruktureditor** bearbeitet werden.

Bei Änderungen in der Datenstruktur ist zu beachten, dass die neuen Elemente erst nach einem Aufbau der Datenstruktur zur Verfügung stehen.

Wird in der Datenstruktur auf der linken Seite eine Verknüpfung selektiert, werden in der rechten Hälfte Informationen über die Verknüpfung dargestellt. Wird der Eintrag "Verknüpfung" einer Datei durch Doppelklicken aufgeklappt, werden alle bereits existierenden Verknüpfungen angezeigt. Existierende Verknüpfungen können ebenfalls durch Doppelklicken aufgeklappt werden und zeigen dann die Verknüpfungsfelder der Verknüpfung an.

Alle Verknüpfungen oder Verknüpfungsfelder in denen Änderungen vorgenommen wurden, werden in einem anderen Font und mit einem grünen Symbol gekennzeichnet. Erst nach Neuaufbau der Datenstruktur können diese Verknüpfungen von einem neu angemeldetem Client verwendet werden.

Verknüpfung anlegen

Um eine neue Verknüpfung zu erzeugen, muss der Eintrag "Verknüpfung" unter der Datei, in welcher die Verknüpfung angelegt werden soll, oder eine bereits bestehende Verknüpfung angeklickt und die Schaltfläche  gedrückt werden.

Die Verknüpfung wird erst angelegt, wenn die Schaltfläche  gedrückt wird. Der gesamte Vorgang kann durch Drücken der Schaltfläche  abgebrochen werden.

• Name

Hier kann der Name der Verknüpfung angegeben werden. Der Name kann in der Programmierung als Referenz auf diese Verknüpfung verwendet werden. Wird kein Name angegeben, kann die Verknüpfung nur über seine Nummer angesprochen werden. Die Nummer der Verknüpfung wird automatisch vergeben.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden. Weitere Informationen zur Vergabe von Namen befinden sich im Abschnitt [Konventionen](#).

• Mit Datei

Hier wird der Name der Datei ausgewählt, die mit der aktuellen Datei verknüpft werden soll. Die Eingabe der gleichen Datei, also eine Verknüpfung in die

Kontakt

aktuelle Datei hinein, ist nicht möglich. Ist in der Datenbank nur eine Datei vorhanden, kann folglich auch keine Verknüpfung definiert werden.

- **Über Schlüssel**

In diesem Feld wird der Schlüssel ausgewählt, über den später der Zugriff auf die verknüpften Datensätze erfolgt.

- **Listeneintrag**

Mit dem Kontrollkästchen "Listeneintrag" wird entschieden ob die Verknüpfung in der Verknüpfungsliste der Verwaltung angezeigt werden soll.

Verknüpfungsfeld anlegen

Um ein neues Verknüpfungsfeld zu erzeugen, muss die Verknüpfung, dem ein Verknüpfungsfeld zugeordnet werden soll, oder ein bereits bestehendes

Verknüpfungsfeld angeklickt und die Schaltfläche  gedrückt werden.

Das Verknüpfungsfeld wird erst angelegt, wenn die Schaltfläche  gedrückt wird.

Der gesamte Vorgang kann durch Drücken der Schaltfläche  abgebrochen werden.

Da aus den Feldern der Verknüpfung der Schlüssel einer anderen Datei gebildet wird, können bei jedem Feld auch die Attribute eines Schlüsselfeldes gesetzt werden. Wird auf verknüpfte Daten zugegriffen, wird aufgrund der Verknüpfungsfelder ein Schlüssel in der verknüpften Datei gebildet. Ob ein Datensatz der anderen Datei verknüpft ist, entscheidet die Übereinstimmung des durch die Verknüpfung gebildeten Schlüssels und des im Datensatz tatsächlich vorhandenen Schlüssels. Dabei muss der Schlüssel des Datensatzes bis zur Länge des erzeugten Schlüssels übereinstimmen (ausgenommen die Felder mit "Nur Zugriffspositionierung"). Dazu stehen wie bei der Definition von Schlüsselfeldern entsprechende Attribute zur Verfügung.

Beispiel:

In der zu verknüpfenden Datei existiert ein Schlüssel, der aus drei ganzzahlig kurzen Feldern besteht. Wird die Verknüpfung aus einem ganzzahlig kurzen Feld gebildet, so sind zu einem Datensatz, der z. B. den Wert 551 im Verknüpfungsfeld hat, alle Datensätze der anderen Datei verknüpft, die im ersten Schlüsselfeld des benutzten Schlüssels auch 551 enthalten, unabhängig von den Werten der beiden anderen Schlüsselfelder.

Beim Zugriff über Verknüpfung wird der Schlüssel der verknüpften Datei also nicht aus den Feldern dieses Schlüssels, sondern aus den Feldern der Verknüpfung erzeugt.

Nur Zugriffspositionierung

Wird dieses Kontrollkästchen aktiviert, so entscheidet beim Zugriff auf verknüpfte Datensätze der aktuelle Wert des Feldes darüber, auf welchen von mehreren verknüpften Datensätzen der Zugriff erfolgen soll. Der Inhalt des Feldes ist also nur für den Zugriff auf die Menge der verknüpften Datensätze relevant und nicht für die

Verknüpfung selbst!

Damit ist es möglich, nicht nur auf den ersten bzw. letzten verknüpften Datensatz zuzugreifen, sondern z. B. direkt auf den fünften oder zehnten.

Beispiel:

Zu einem Kunden in der Kundendatei existieren Aufträge in der Auftragsdatei. Die Verknüpfung geschieht über einen Schlüssel, bestehend aus Kundennummer, Auftragsdatum und Auftragsnummer, der durch das Verknüpfungsfeld "Kundennummer" gebildet wird. In diesem Fall kann bei der Ausführung der Verknüpfung nur beim ersten bzw. letzten Auftrag des Kunden begonnen werden.

Wird die Verknüpfung nun um ein Datumsfeld mit "Nur Zugriffspositionierung" ergänzt, kann durch die Belegung des Datumsfeldes auch innerhalb der verknüpften Aufträge auf eine bestimmte Stelle verknüpft werden.

-  Die Feldtypen Gleitkomma, ganzzahlig kurz und ganzzahlig lang sind bei Verknüpfungen nicht kompatibel. Beispielsweise kann von einem Gleitkomma-Feld aus nicht auf einen ganzzahligen Schlüssel verknüpft werden.

Verknüpfungsfeld ändern

Die Eigenschaften eines Verknüpfungsfeldes können nach der Auswahl des Feldes über die Schaltfläche  verändert werden. Es können alle Eigenschaften des Verknüpfungsfeldes geändert werden.

Verknüpfungsfeld löschen

Das selektierte Verknüpfungsfeld kann durch Drücken der Schaltfläche gelöscht werden.

Verknüpfung ändern

Eigenschaften der Verknüpfung können nach der Auswahl der Verknüpfung über die Schaltfläche  verändert werden. Es können alle Eigenschaften der Verknüpfung geändert werden.

Verknüpfung löschen

Die selektierte Verknüpfung kann durch Drücken der Schaltfläche gelöscht werden.

Datenstruktur Konventionen

Konventionen der Datenstruktur

Die Elemente der Datenstruktur werden durch Nummern und teilweise durch Namen gekennzeichnet. Nummern werden immer vergeben, Namen können, müssen aber nicht, zugewiesen werden. Die Datenstruktur besitzt folgende Elemente:

Element	Nummer Name
Tabelle	
Teildatensatz	
Feld	
Schlüssel	
Schlüsselfeld	
Verknüpfung	
Verknüpfungsfeld	

Die Nummer einer Datei kann im Bereich zwischen 1 und 999 frei vergeben werden, alle anderen Nummern werden durch CONZEPT 16 zugewiesen. Der Name eines Elements kann bis zu 20 Stellen lang sein und beliebige Zeichen enthalten. Die erste Stelle des Namens muss ein Zeichen zwischen A und Z bzw. zwischen a und z sein. Das letzte Zeichen darf kein Leerzeichen sein. In einem Namen ist der Doppelpunkt (:) nicht zulässig, bei Feldnamen können außerdem der Schrägstrich (/) und das Komma (,) nicht verwendet werden.

Der Name eines Elements muss in der Datenbank eindeutig sein, d. h. ein Element mit diesem Namen kann nur einmal vorhanden sein. So besteht zwar die Möglichkeit, sowohl eine Datei mit dem Namen "TEST" und ein Feld mit dem Namen "TEST" zu definieren, nicht aber zwei Felder (auch nicht aus verschiedenen Dateien) mit dem Namen "TEST". Dies dient später einem einfachen, direkten Zugriff auf jedes Element in der Datenbank. Bei einem Namen können sowohl Groß- als auch Kleinbuchstaben benutzt werden, zur Identifikation ist diese Unterscheidung belanglos. Der Name "TEST" und der Name "test" (oder auch "TesT" usw.) wird von CONZEPT 16 als identisch erkannt. Wie bereits erwähnt kann der Name eines Elements auch weggelassen werden (leerer Name). In diesem Fall wird auf das Element natürlich nur über die Nummer zugegriffen.

Empfehlungen zur Namensvergabe

Je nach Strukturelement wird dem Namen des Elementes das Präfix **tbl** für Datei, **sbr** für Teildatensatz, **f** für Felder, **key** für Schlüssel und **Ink** für Verknüpfung vorangestellt. Die Benennung innerhalb der Datenstruktur orientiert sich immer an einem Dateikürzel und einem Feldkürzel. Bei beiden Kürzeln ist auf eine sinnvolle, namhafte und möglichst kurze Benennung zu achten.

Je nach Datentyp sollte bei Feldern sollte ein Buchstabe für den Datentyp zwischen dem **f** und dem Dateikürzel eingefügt werden. Unserer Empfehlung nach wären das **a** für alphanumerisch, **w** für ganzzahlig kurz, **i** für ganzzahlig lang, **b** für ganzzahlig 64, **f** für Gleitkomma, **m** für Dezimal, **d** für Datum, **t** für Zeit und **l** für logische Werte.

Beispiel:

Element	Name
Datei	tblCstCustomer
Teildatensatz	sbrCstCommon
Feld	faCstName
Schlüssel	keyCstID (Der Name des Schlüssels leitet sich aus dem Schlüsselfeld ab.)

Die Verwendung von Umlauten, Sonderzeichen und dem Leerzeichen ist prinzipiell erlaubt, doch kann es bei der Verwendung von externen Programmen (zum Beispiel beim Zugriff auf die CONZEPT 16-Datenbank über die ODBC-Schnittstelle) zu Problemen führen.



Grundsätzlich sollten für Namen in der Datenstruktur nur Buchstaben, Zahlen und die Zeichen "." und "_" verwendet werden.

Änderungen der Datenstruktur von mehrere Benutzern

Es besteht die Möglichkeit, dass mehrere Benutzer gleichzeitig die Datenstruktur verändern. Sie können sogar in der gleichen Datei Änderungen vornehmen. Es ist aber nicht möglich, das gleiche Feld, den gleichen Schlüssel oder die gleiche Verknüpfung zu bearbeiten. Wird dies trotzdem versucht, erscheint die Meldung "Exklusive Sperre nicht möglich".

Änderungen von anderen Benutzern werden in der Datenstruktur erst sichtbar, wenn der Datenstruktureditor erneut aufgerufen wird. Das Bearbeiten der Datenstruktur mit mehreren Benutzern ist somit nur sinnvoll, wenn sich die Benutzer entsprechend absprechen.

Aufruf

Der Datenstruktureditor kann über den Menüpunkt **Extras / Datenstruktureditor** erreicht werden.

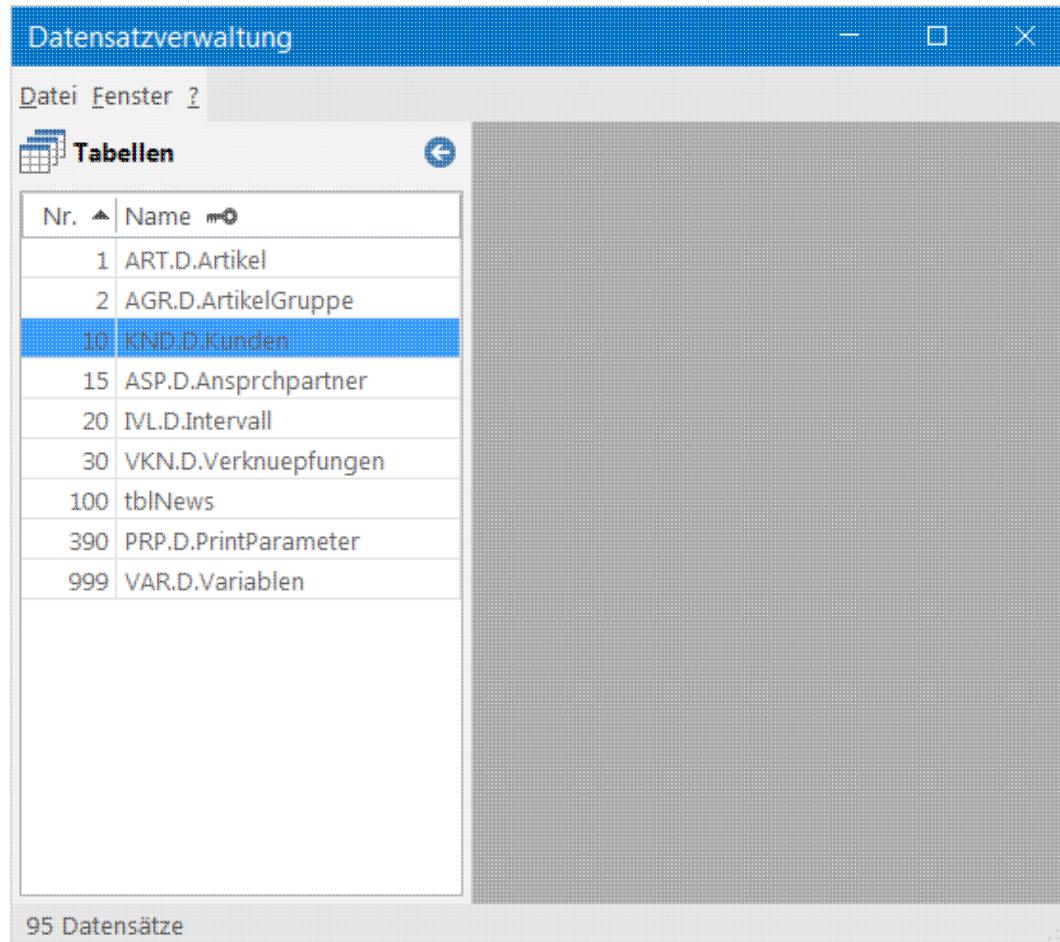
Datensatzverwaltung
Verwaltung von Datensätzen
Siehe [Blog](#)

Über den Menüeintrag **Extras / Datensatzverwaltung...** werden die Tabellen und die dazugehörigen Datensätze der Datenbank angezeigt.

i Zum Aufrufen benötigt der Benutzer die Rechte [UrmPermRead](#) und [UrmIdPermRead](#) auf der [Standard-Elementgruppe](#) _table oder einer Elementgruppe, die mindestens eine Tabelle enthält.

Es gibt die Möglichkeit Datensätze anzulegen, zu ändern und zu löschen, sowie Tabellen zu leeren. Weiterhin können direkt die verknüpften Datensätze in einer anderen Tabelle angezeigt werden.

Nach dem Aufruf des Menüpunktes "Extras" / "Datensatzverwaltung..." wird folgender Dialog angezeigt:



Auf der linken Seite werden die Tabellen angezeigt. Die Anzeige kann über die Schaltfläche ausgeblendet und mit der Schaltfläche wieder eingeblendet werden.

In der Liste kann mit **strg** + **F** nach Tabellennamen und Nummern gesucht werden. Mit **strg** + **F3** wird auf den nächsten und mit **strg** + **↑** + **F3** auf den vorherigen Suchtreffer positioniert. Die Groß- / Kleinschreibung wird bei der Suche nicht

Kontakt

beachtet. Die Verwendung von den Wildcards * und ? ist möglich. Standardmäßig wird nach <Suchbegriff>* gesucht.

Durch Doppelklick bzw.  wird der Inhalt der ausgewählten Tabelle angezeigt.

Wird in der Liste der Tabellen mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



Wird zusätzlich die -Taste gedrückt, erscheint das erweiterte Kontextmenü:



• Suchen

Öffnet die Suche nach Tabellen. Diese Aktion wird ebenfalls über die Tasten  +  ausgelöst.

• Vorwärts weitersuchen

Sucht die nächste Tabelle, die dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die -Taste ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

• Rückwärts weitersuchen

Kontakt

Sucht die vorherige Tabelle, die dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die Tasten + ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

- **Öffnen**

Der Inhalt der Tabelle wird auf der rechten Seite angezeigt. Ist die Tabelle bereits geöffnet, wird das Fenster in den Vordergrund geholt. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

- **Öffnen (neues Fenster)**

Der Inhalt der Tabelle wird auf der rechten Seite in einem neuen Fenster angezeigt. Diese Aktion wird ebenfalls über die Tasten + ausgelöst. Dieser Menüpunkt wird nur angezeigt, wenn das Menü mit gedrückter -Taste geöffnet wurde.

- **Leeren**

Der Inhalt der Datei wird nach Bestätigung einer Abfrage geleert.

- **Datensatz einfügen...**

Ein Fenster mit den Feldern der Tabelle wird angezeigt, in dem ein neuer Datensatz angelegt werden kann.

- **Datensätze exportieren...**

Export von Datensätzen.

- **Datensätze importieren...**

Import von Datensätzen.

Kontakt

Datensatzverwaltung Tabelle

Inhalt einer Tabelle in der Datensatzverwaltung

Siehe [Blog](#)

Die Tabelleninhalte werden in der [Datensatzverwaltung](#) auf der rechten Seite in einem extra Dialog pro Datei angezeigt. Der Dialog sieht wie folgt aus:

The screenshot shows a software interface for managing database tables. At the top, a title bar reads "Tabelle KND.D.Kunden [10]". Below it is a toolbar with various icons for file operations like saving, opening, and filtering. The main area is divided into two tabs: "Abfrage" (Query) and "Felder" (Fields). The "Abfrage" tab contains a large empty text input field with a search icon and a dropdown menu. The "Felder" tab displays a table of fields with columns for index (S.), number (Nr.), name, and type. The data is as follows:

S.	Nr.	Name	Typ
✓	1 / 1	KND.iID	int
✓	1 / 2	KND.aEintrTyp	alpha(3)
✓	1 / 3	KND.dEintrDatum	date
✓	1 / 4	KND.tEintrZeit	time
✓	1 / 5	KND.aFirmenBez	alpha(50)
✓	1 / 6	KND.aStrasse	alpha(30)
✓	1 / 7	KND.aPlz	alpha(5)
✓	1 / 8	KND.aOrt	alpha(30)
✓	1 / 9	KND.aLkz	alpha(3)
✓	1 / 10	KND.aUrl	alpha(60)
✓	1 / 11	KND.fRabatt	float
✓	1 / 12	KND.fUmsatz	float
✓	1 / 13	KND.lEcm	logic

Below the table, a preview window shows 95 rows of data from the table, with columns labeled KND.iID, KND.aEintrTyp, and KND.dEintrDatum. The data includes entries like (1.020, 2FX, 26.06.1998) and (1.021, 1TL, 07.04.1992).

Mit der Schaltfläche wird die Anzeige der Datensätze aktualisiert. Über das Feld links davon kann ein Schlüssel der Tabelle ausgewählt werden, um die Tabelle nach diesem Schlüssel zu sortieren. Die Schaltfläche dreht die Sortierreihenfolge um.

Die Schaltflächen „ „ „ und „ „ entsprechen den Aktionen im Kontextmenü im Bereich [Datensätze](#).

Mit einem Klick auf öffnet oder schließt sich ein Eingabebereich, in dem eine Abfrage zur Einschränkung der Ergebnisse angegeben werden kann. Nähere Informationen hierzu finden Sie im Bereich [Abfrage](#).

Mit einem Klick auf öffnet oder schließt sich ein Liste der Felder der Tabelle. Nähere Informationen hierzu finden Sie im Bereich [Felder](#).

Abfrage

Im Bereich Abfrage können Vergleiche von Feldern mit Werten oder anderen Feldern eingegeben werden. Mit der Schaltfläche wird die Abfrage gestartet. Fehler in der

Kontakt

Abfrage werden unter dem Eingabeobjekt angezeigt. Die Schaltfläche entfernt die Einschränkung der Datensätze.

In der Abfrage sind alle, im Bereich Vergleichsoperatoren erläuterten, Vergleichsoperatoren möglich. Zusätzlich kann ein Bereich mit between [...] [...] angegeben werden. Mehrere Vergleiche können mit den Verknüpfungsoperatoren and, or, xor, not und ! kombiniert werden. Mit runden Klammern kann die Abfrage verschachtelt werden.

Die Abfragen müssen wie folgt aufgebaut sein:

<Ausdruck> = <Feld> <Vergleichsoperator> (<Wert> | <Feld>) [<Verknüpfungsoperator> <Ausdruck>]

Felder können in folgenden Schreibweisen angegeben werden:

- <Feldname>
- "<Feldname>"
- "<Tabellennummer>,<Teildatensatznummer>,<Feldnummer>"
- "<Teildatensatznummer>,<Feldnummer>"

Zugriff auf Felder

Der Zugriff auf Felder kann wahlweise über das Kontextmenü des Eingabeobjektes, über die Tastenkürzel + , + , + und + oder über den Bereich Felder.

Feldwerte

Die Werte müssen anhand des Datentyps wie folgt angegeben werden:

Datentyp Format

<u>alpha</u>	'Zeichenkette'
<u>word</u>	5
<u>int</u>	-10, +20, 50
<u>bigint</u>	-2147483648, +3258594759, 76646
<u>float</u>	-5.4, 3.5
<u>decimal</u>	-8.3, 7.6
<u>date</u>	yyyy-MM-dd
<u>time</u>	HH:mm[:ss[.n[n]]]
<u>logic</u>	<u>true</u> , <u>false</u> , y, n

Die Abkürzungen bei date und time stehen dabei für folgende Angaben:

yyyy Jahr vierstellig (1900..2154)

MM Monat (01..12)

dd Tag des Monats (01..31)

HH Stunden zweistellig (00..23)

mm Minuten zweistellig (00..59)

ss Sekunden zweistellig (00..59)

n Zehntelsekunde (0..9)

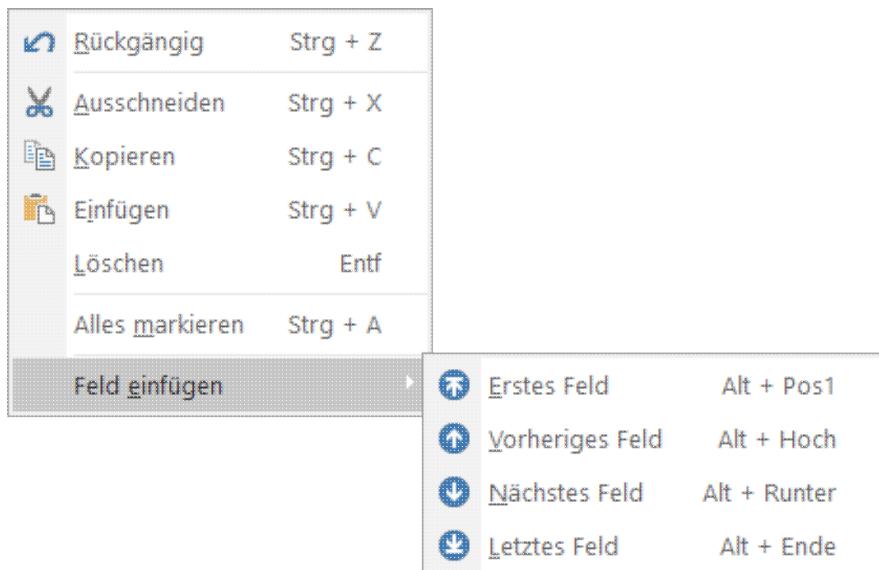
nn Hundertstelsekunden zweistellig (00..99)

Alternativ kann null als Wert angegeben werden, um das Feld mit dem **NULL**-Wert des Feldtyps zu vergleichen.

Beispiele:

```
// Alle Kunden mit einer ID > 1022, deren Eintragstyp nicht mit 1 beginnt
KND.iID > 1022 and !KND.iEntryType = 1
```

Wird in dem Eingabeobjekt der Abfrage mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



• Rückgängig

Letzte Eingabe rückgängig machen

• Ausschneiden

Markierten Bereich ausschneiden

• Kopieren

Markierten Bereich in die Zwischenablage kopieren

• Einfügen

Text der Zwischenablage einfügen

• Löschen

Markierten Bereich löschen

• Alles markieren

Gesamten Inhalt des Eingabeobjektes markieren

• Feld einfügen

Feld in der Abfrage an der aktuellen Position einfügen

♦ **Erstes Feld**

Das erste Feld der Tabelle wird eingefügt

♦ **Vorheriges Feld**

Das vorherige Feld der Tabelle wird eingefügt

♦ **Nächstes Feld**

Das nächste Feld der Tabelle wird eingefügt

♦ **Letztes Feld**

Das letzte Feld der Tabelle wird eingefügt

Abhängig von einem eventuell vorhandenen Wortteil vor dem Cursor wird nur der Rest des Feldes ergänzt. Steht bspw. in dem Eingabeobjekt vor der aktuellen Cursor-Position "KND.a", werden nur Felder ergänzt, die zu der aktuellen Tabelle gehören und mit "KND.a" beginnen. Der eingefügte Teil wird markiert, sodass er bei der nächsten Aktion über einen der vier Menüpunkte ersetzt wird.

Felder

Im Bereich Felder wird eine Liste der Felder der Tabelle dargestellt. Diese können per Drag & Drop und per Kontextmenü in das Eingabeobjekt im Bereich Abfrage eingefügt werden.

In der Liste kann mit nach Feldnamen gesucht werden. Mit wird auf den nächsten und mit auf den vorherigen Suchtreffer positioniert. Die Groß- / Kleinschreibung wird bei der Suche nicht beachtet. Die Verwendung von den Wildcards * und ? ist möglich. Standardmäßig wird nach <Suchbegriff>* gesucht.

Wird in der Liste der Felder mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



• **Suchen**

Öffnet die Suche nach Feldern. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

• **Vorwärts weitersuchen**

Kontakt

Sucht das nächste Feld, das dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die -Taste ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

- **Rückwärts weitersuchen**

Sucht das vorherige Feld, das dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die Tasten + ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

- **Ausblenden / Einblenden**

Versteckt / zeigt die ausgewählten Felder im Bereich Datensätze.

- **Feld in Abfrage einfügen**

Fügt das selektierte Feld in der Abfrage an der aktuellen Cursor-Position ein. Der Menüpunkt wird nur angezeigt, wenn der Bereich Abfrage geöffnet ist.

Datensätze

Im Bereich Datensätze wird eine Liste der enthaltenen Datensätze der Tabelle angezeigt. Die Liste kann durch Verknüpfungen oder Abfragen eingeschränkt sein. Der Spaltentitel enthält den Feldnamen.

In der Liste werden standardmäßig die ersten 50 Felder der Datei dargestellt. Maximal werden 100 Spalten angezeigt. Über den Bereich Felder können Felder einge- und ausgeblendet werden.

Über der Liste wird die aktuelle Anzahl der Datensätze angezeigt. Sie passt sich bei Einschränkungen der Ergebnismenge an. Werden Datensätze markiert, erscheint die Anzahl in Klammern daneben.

Wird in der Liste der Datensätze mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



Ist keine Verknüpfung enthalten, fällt der Menüpunkt "Verknüpfen" weg.

- **Aktualisieren**

Der Inhalt der Tabelle wird aktualisiert.

- **Datensatz einfügen...**

Ein Fenster mit den Feldern der Tabelle wird angezeigt, in dem ein neuer Datensatz angelegt werden kann.

- **Datensatzkopie einfügen...**

Ein Fenster mit den Feldern der Tabelle wird angezeigt, in dem ein neuer Datensatz angelegt werden kann. Die Feldinhalte werden vom aktuellen Datensatz kopiert.

- **Datensatz bearbeiten...**

Ein Fenster mit den Feldern der Tabelle wird angezeigt, in dem der aktuelle Datensatz verändert werden kann.

- **Datensatz löschen... / Datensätze löschen...**

Nach Bestätigung einer Abfrage werden die ausgewählten Datensätze gelöscht. Konnten nicht alle Datensätze gelöscht werden, wird in einer Meldung angezeigt, wie viele Datensätze nicht gelöscht wurden.

- **Verknüpfen**

Nach einer Auswahl der Verknüpfung werden die verknüpften Datensätze in einem weiteren Fenster angezeigt.

- **Kopieren**

Der Inhalt der aktiven Zelle wird in die Zwischenablage kopiert. Sind mehrere Datensätze ausgewählt, wird der Inhalt der aktiven Spalte von diesen kopiert.

Kontakt

Datensatzverwaltung Datensatz einfügen / bearbeiten

Datensatz in eine(r) Tabelle einfügen / bearbeiten

Zum Einfügen oder Bearbeiten eines Datensatzes in der Datensatzverwaltung öffnet sich der folgende Dialog:

In dem Fenster wird für jeden Teildatensatz eine Gruppe dargestellt, die die dazugehörigen Felder enthält. Die Dialoggröße skaliert mit der Anzahl der Felder. Bei alpha-Feldern kann über die Schaltfläche der Text in einem extra Fenster eingesehen und bearbeitet werden.

Mit einem Klick auf OK wird der Datensatz gespeichert und entsperrt. Mit Abbrechen werden die Änderungen verworfen. Ein Klick auf Übernehmen speichert den Datensatz, behält die Sperre aber aufrecht. In diesem Fall wird das Fenster weiterhin angezeigt.

Über den Teildatensätzen wird vom gespeicherten Datensätzen die Datensatz-ID und die Größe des Datensatzes in der Datenbank (siehe RecLenPacked) angezeigt. Die Datensatzgröße wird nur beim Speichern aktualisiert.

Wird ein neuer Datensatz, mit einem Klick auf Übernehmen, angelegt, wird der Dialog in den Bearbeitungsmodus versetzt.

Wurde der Dialog von einem Fenster mit dem Tabelleninhalt heraus aufgerufen, wird der Datensatz in diesem Fenster aktualisiert. In allen anderen Fenstern der Datei muss auf gedrückt werden, damit die Änderungen sichtbar werden.

Kontakt

Kann ein Datensatz aufgrund einer Sperre durch einen anderen Benutzer oder fehlender Berechtigungen nicht geändert werden, steht dies im Titel des Dialogs. Der genaue Grund wird angezeigt, wenn mit der Maus über das Symbol gefahren wird.

Datensatzverwaltung Export

Export der Datensätze einer Tabelle in der Datensatzverwaltung

Siehe [Blog](#)

Format auswählen

Sollen Datensätze in der [Datensatzverwaltung](#) exportiert werden, öffnet sich der folgende Assistent:

Im ersten Schritt wird das Zielformat ausgewählt. Der Aufbau der Formate ist bei den möglichen [Transferformaten der Datensatzverwaltung](#) beschrieben.

Beim CSV-Format können noch zusätzlich Zeichen für die Feld- und Dezimaltrennung definiert werden. Optional lassen sich auch die Spaltennamen in der ersten Zeile einfügen.

Ein Klick auf "Hilfe" zeigt die Dokumentation zur aktuellen Seite des Assistenten.

Felder zuordnen

Wird auf "Weiter" geklickt, können die zu exportierenden Felder gewählt werden.

Die ersten Spalte zeigt die Position an. In der zweiten Spalte kann ausgewählt werden, ob das Feld exportiert werden soll. Es können nur Felder ausgewählt werden, die zugewiesen sind. Die Auswahl kann mit der Taste oder per Doppelklick gesetzt oder aufgehoben werden.

In der Spalte "Name" wird das Quellfeld in der Datenbank angezeigt. Dies kann über einen Doppelklick oder die Taste neu zugewiesen werden.

Die Spalte "Typ" gibt den Quelldatentyp wieder. Bei [alpha](#)-Feldern steht in Klammern die Anzahl der Zeichen dahinter.

In der Spalte "Externer Name" wird zunächst der Feldname angezeigt. Dieser kann durch Doppelklick oder die Taste beliebig geändert werden.

Die Sortierung der Einträge kann per Drag & Drop, über das Kontextmenü und über die Tastenkombinationen + , + , + und + modifiziert werden.

Wird in der Liste der Felder mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



- **Auswählen / Abwählen**

Aktuelles Feld zum Export auswählen oder abwählen.

- **Feld zuordnen**

Feldzuordnung der aktuellen Zeile ändern.

- **Positionieren**

Zielposition der aktuellen Zeile in der Datei ändern.

- ◆ **An den Anfang**

Aktuelle Zeile an den Anfang verschieben.

- ◆ **Nach oben**

Aktuelle Zeile nach oben verschieben.

- ◆ **Nach unten**

Aktuelle Zeile nach unten verschieben.

- ◆ **Ans Ende**

Aktuelle Zeile an das Ende verschieben.

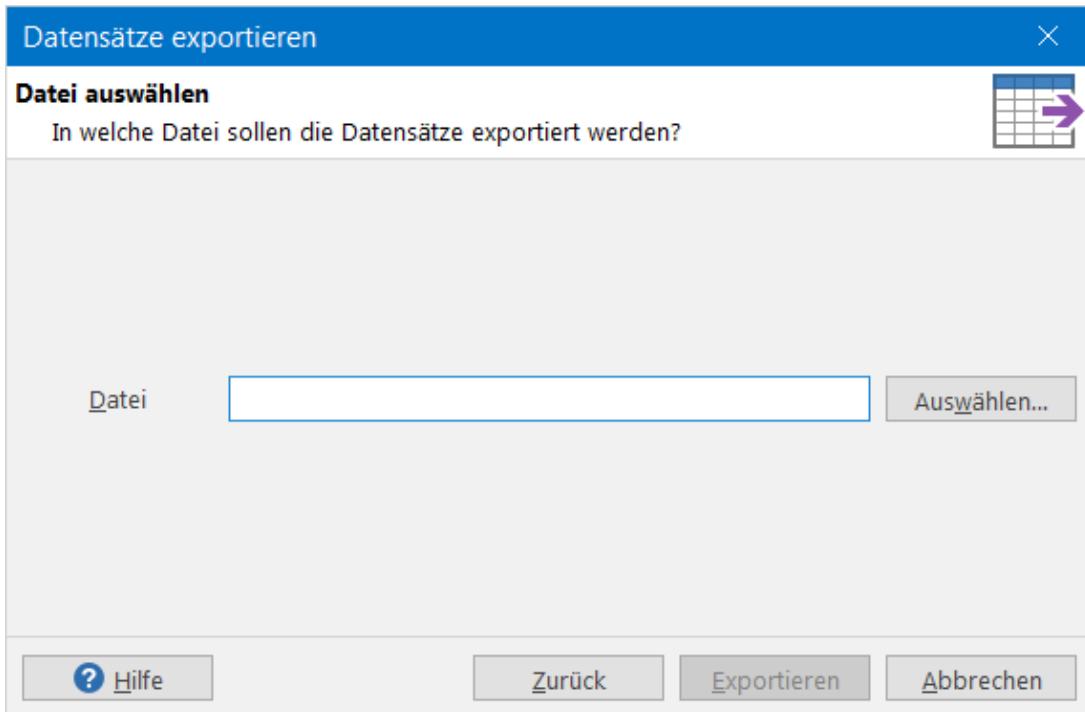
- **Umbenennen**

Name des Feldes in der Datei ändern.

Datei auswählen

Nach einem Klick auf "Weiter" fordert der Assistent zur Auswahl der Zielfile auf:

Kontakt



Diese kann über die Schaltfläche "Auswählen" oder durch direkte Eingabe gewählt werden. Als Dateiname wird der Name der Tabelle vorgeschlagen. Ist eine Datei ausgewählt, wird die Schaltfläche "Exportieren" aktiv, mit der der Exportvorgang gestartet werden kann.

Fortschritt

Auf der folgenden Seite wird der Fortschritt des Exportvorgangs angezeigt:

Ist der Exportvorgang abgeschlossen, werden die Schaltflächen "Zurück" und "Abbrechen" entfernt und die Schaltfläche "OK" erscheint. Sind Fehler beim Export aufgetreten, wird der Benutzer darüber informiert.

Datensatzverwaltung Import

Import von Datensätze in eine Tabelle in der Datensatzverwaltung

Siehe [Blog](#)

Format auswählen

Sollen Datensätze in der [Datensatzverwaltung](#) importiert werden, öffnet sich der folgende Assistent:

Im ersten Schritt wird das Quellformat ausgewählt. Der Aufbau der zu importierenden Datei ist bei den möglichen [Transferformaten der Datensatzverwaltung](#) beschrieben.

Beim CSV-Format können noch zusätzlich Zeichen für die Feld- und Dezimaltrennung definiert werden. Zusätzlich muss angegeben werden, ob in der ersten Zeile der Datei die Spaltennamen enthalten sind.

Ein Klick auf "Hilfe" zeigt die Dokumentation zur aktuellen Seite des Assistenten.

Datei auswählen

Im nächsten Schritt fordert der Assistent zur Eingabe der Quelldatei auf.

Diese kann mit der Schaltfläche "Auswählen" oder durch direkte Eingabe gewählt werden. Ist eine Datei ausgewählt, wird die Schaltfläche "Weiter" aktiv.

Felder zuordnen

Nach einem Klick auf "Weiter" kann im Assistenten die Feldzuordnung geändert werden:

Die ersten Spalte zeigt die Position an. In der zweiten Spalte kann ausgewählt werden, ob das Feld importiert werden soll. Es können nur Felder ausgewählt werden, die zugewiesen sind. Die Auswahl kann mit der Taste  oder per Doppelklick gesetzt oder aufgehoben werden.

In der Spalte "Externer Name" wird das Quellfeld in der Datei angezeigt.

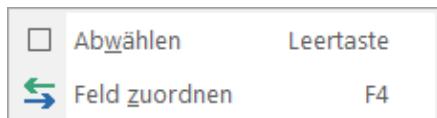
Die Spalte "Externer Typ" gibt den Quelldatentyp wieder. Bei alpha-Feldern steht in Klammern die Anzahl der Zeichen dahinter. Die Spalte ist beim Import von CSV-Dateien nicht vorhanden, da dort für die in der Datei enthaltenen Werte kein Datentyp hinterlegt ist.

In der Spalte "Name" wird das Zielfeld in der Datenbank angezeigt. Dies kann über einen Doppelklick oder die Taste  neu zugewiesen werden.

Die letzte Spalte enthält den Zieldatentyp. Dieser muss mit dem Quelldatentyp kompatibel sein. Eine genaue Übersicht der Konvertierungsmöglichkeiten befindet sich unter [Datensatzverwaltung - Typkonvertierung](#).

Kontakt

Wird in der Liste der Felder mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



- **Auswählen / Abwählen**

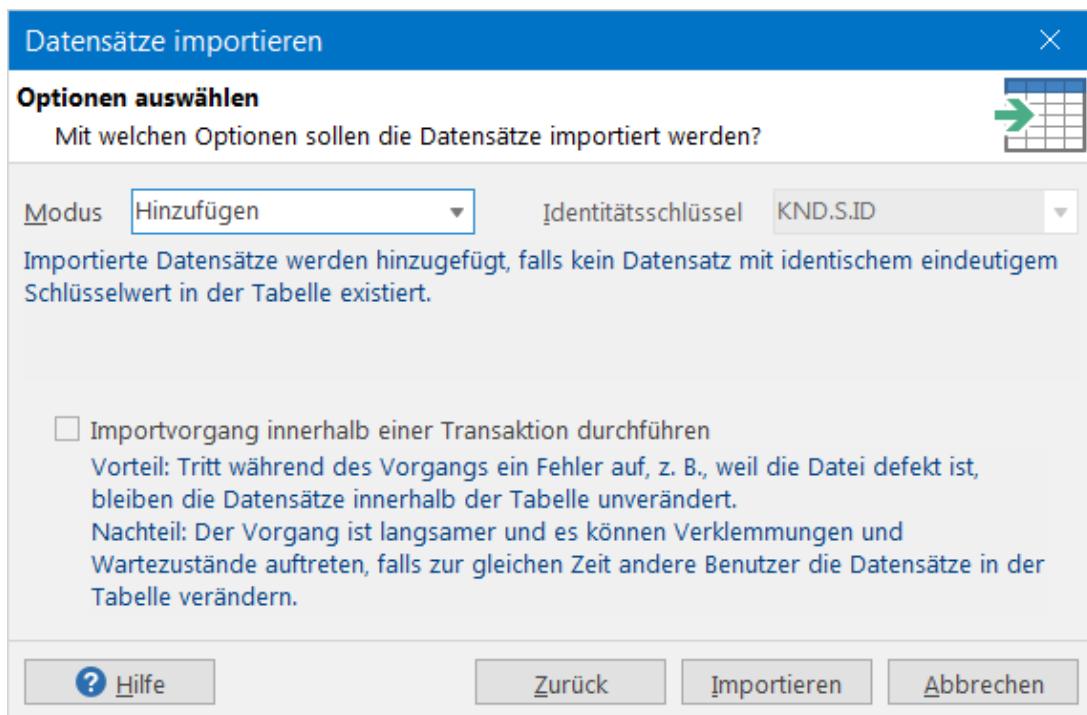
Aktuelles Feld zum Import auswählen oder abwählen.

- **Feld zuordnen**

Feldzuordnung der aktuellen Zeile ändern.

Optionen auswählen

Im nächsten Schritt werden die Importoptionen definiert:



Hier kann einer der folgenden Modi ausgewählt werden.

- **Ersetzen**

Die Datensätze aus der Quelldatei werden importiert.



Zuvor werden alle bestehenden Datensätze gelöscht.

- **Hinzufügen**

Kontakt

Neue Einträge werden aus der Quelldatei importiert. Datensätze, deren eindeutige Schlüsselwerte bereits vorhanden sind, werden nicht aktualisiert.

- **Aktualisieren**

Vorhandene Einträge werden aus der Quelldatei importiert. Datensätze, deren eindeutige Schlüsselwerte nicht vorhanden sind, werden nicht importiert.

- **Hinzufügen/Aktualisieren**

Neue Einträge werden aus der Quelldatei importiert und Datensätze, deren eindeutige Schlüsselwerte bereits vorhanden sind, werden aktualisiert.

Bei den aktualisierenden Modi kann zusätzlich ein Identitätsschlüssel angegeben werden. Alle Felder, die nicht Bestandteil des Schlüssels sind, werden durch die importierten Daten ersetzt, wenn ein Datensatz mit den gleichen Schlüsselwerten bereits vorhanden ist.

Ist eine Aktion nicht möglich, weil der Benutzer unzureichende Rechte hat, oder die Tabelle keinen Schlüssel besitzt, wird ein Fehlersymbol angezeigt, welches beim Überfahren mit der Maus den Fehler erläutert.

Wird die zweite Option ausgewählt, findet der Import in einer Transaktion statt. Treten Lesefehler in der Quelldatei auf, behält die Tabelle ihren Ausgangszustand.

Fortschritt

Auf der folgenden Seite wird der Fortschritt des Importvorgangs angezeigt:

Ist der Importvorgang abgeschlossen, werden die Schaltflächen "Zurück" und "Abbrechen" entfernt und die Schaltfläche "OK" erscheint. Sind Fehler beim Import aufgetreten, wird der Benutzer darüber informiert.

Datensatzverwaltung - Typkonvertierung
 Typkonvertierung beim Import von Datensätzen
Datensatzverwaltung,
 Siehe Datensatzverwaltung
- Import

Datenformat CSV

Beim CSV-Import ist kein Quelldatentyp bekannt. Die in der Datei enthaltenen Daten werden als alpha interpretiert und in den jeweiligen Zieldatentyp konvertiert. Die Konvertierung ist fehlertolerant, kann jedoch auch fehlschlagen. In diesem Fall wird die Meldung "Datei ist defekt" ausgegeben.

Datenformat C16R

Beim C16R-Import ist der Quelldatentyp bekannt. Sofern die Konvertierung von Quell- zu Zieldatenfeld nicht kompatibel ist (ungleiche Datentypen oder kürzere alpha-Länge), wird ein Warnsymbol bei Auswahl des Zielfeldes in der Auswahlliste angezeigt. In diesem Fall wird beim Import eine Konvertierung vom Quelldatentyp zum Zieldatentyp vorgenommen. Auf diese Weise lassen sich Daten importieren, deren Datentypen nicht übereinstimmen.

Die folgende Tabelle zeigt, welche Typkonvertierungen beim Import von C16R-Dateien möglich sind bzw. Einschränkungen unterliegen:

	Ziel	<u>alpha</u>	<u>bigint</u>	<u>date</u>	<u>decimal</u>	<u>float</u>	<u>int</u>	<u>logic</u>	<u>time</u>	<u>word</u>	Ziel	Quelle
Quelle		☒	✗	✗	✗	✗	✗	✗	✗	✗	✗	Quelle
<u>alpha</u>		☒	✗	✗	✗	✗	✗	✗	✗	✗	✗	<u>alpha</u>
<u>bigint</u>		✗		✗		⚠	⚠	⚠	✗	⚠		<u>bigint</u>
<u>date</u>		✗	✗	✗		✗	✗	✗	✗	✗	✗	<u>date</u>
<u>decimal</u>		✗	⚠	✗		⚠	⚠	⚠	✗	⚠		<u>decimal</u>
<u>float</u>		✗	⚠	✗	⚠		⚠	⚠	✗	⚠		<u>float</u>
<u>int</u>		✗		✗			⚠		✗		⚠	<u>int</u>
<u>logic</u>		✗		✗					✗			<u>logic</u>
<u>time</u>		✗	✗	✗	✗		✗	✗	✗		✗	<u>time</u>
<u>word</u>		✗		✗			⚠		✗			<u>word</u>
Quelle		☒	✗	✗	✗	✗	✗	✗	✗	✗	✗	Quelle

Ziel alpha bigint date decimal float int logic time word Ziel
Symbolerklärung

- kompatibel (ohne Einschränkung konvertierbar)
- ✗ nicht kompatibel (nicht konvertierbar)
- ⚠ konvertierbar (ggfls. verlustbehaftet)
- ☒ konvertierbar (bei gleicher oder größerer Länge des Zieldatenfeldes, ansonsten wird abgeschnitten)

Datensatzverwaltung Transferformate
 Dateiformate für den Im- und Export von Datensätzen
 Datensätze können in der Datensatzverwaltung in den folgenden Formaten für den Im-
und Export verwendet werden:

- CONZEPT 16-Datensätze (*.c16r)
- Komma-separierte Werte (*.csv)

CONZEPT 16-Datensätze (*.c16r)

Dateien im internen Format sind wie folgt aufgebaut:

```
<signatur/> = 4-Byte string : "c16r"  Formatsignatur (CONZEPT 16 records)
<version/> = 4-Byte integer (little-endian) : 0x01000000  Formatversion (1.0.0.0)
<fields> = MSX-Message : ID = 1  Felder <field> = MSX-Item : ID = 1
Felddefinition <name/> = MSX-Data (Typ: alpha)  Feldbezeichner (max. 80 Zeichen)
<type/> = MSX-Data (Typ: int)  Feldtyp (vgl. FldType) <size/> = MSX-Data (Typ:
int)  Feldgröße (vgl. FldLen) </field> ...  weitere Felddefinitionen </fields>
<records> = MSX-Message : ID = 2  Datensätze <record> = MSX-Item : ID = 1
Datensatz <field/> = MSX-Data (Typ: Feldtyp)  Feldwert (gleiche Reihenfolge wie
Felddefinition) ...  weitere Felder </record> ...  weitere Datensätze </records>
```

Komma-separierte Werte (*.csv)

Die Datensatzverwaltung kann Komma-separierte Werte mit den Feldtrennzeichen Komma (,), Semikolon(;) und Tabulator erzeugen. Satztrennzeichen ist immer der Zeilenumbruch.

In der ersten Zeile werden optional die Feldbezeichnungen angegeben.

Die Feldinhalte werden je nach Feldtyp (vgl. FldType) in folgender Notation geschrieben:

Feldtyp	Notation	Anmerkung
<u>alpha</u>	"<Wert>"	Anführungszeichen werden durch doppelte Anführungszeichen ersetzt
<u>word</u> , <u>int</u> , <u>bigint</u>	<Wert> / -<Wert>	keine Tausendertrennzeichen
<u>float</u> , <u>decimal</u>	<Wert> / -<Wert>	wahlweise Punkt (.) oder Komma (,) als Dezimaltrennzeichen. Sind keine Dezimalstellen vorhanden, werden nur die Vorkommastellen geschrieben.
<u>logic</u>	0 / 1	<u>false</u> = 0 / <u>true</u> = 1
<u>date</u>	yyyy-MM-dd	
<u>time</u>	HH:mm[:ss[(. ,)ff]]	wahlweise Punkt (.) oder Komma (,) als Trennzeichen (siehe Dezimaltrennzeichen für <u>float</u> / <u>decimal</u>) für Hundertstelsekunden. Die Hundertstelsekunden werden nur geschrieben, wenn sie nicht 0 und die Sekunden nur, wenn sie und die Sekunden nicht 0 sind.

Kontakt

Es kann entweder das Feldtrennzeichen oder das Dezimaltrennzeichen ein Komma (,) sein. Beides zusammen kann nicht ausgewählt werden, da sie sonst nicht unterschieden werden können.

BLOB-Verwaltung

Verwalten von binären Objekten und Verzeichnissen

Siehe [Blog](#)

Über den Menüeintrag **Extras / BLOB-Verwaltung...** werden die binären Objekte und Verzeichnisse der Datenbank angezeigt.



Zum Aufrufen benötigt der Benutzer das Recht `UrmIdcPermRead` auf der Standard-Elementgruppe _blob.

In der BLOB-Verwaltung können binäre Objekte und Verzeichnisse angelegt, umbenannt, importiert, exportiert, gelöscht, geleert, kopiert und verschoben werden. Weiterhin ist es möglich sich die Eigenschaften eines binären Objektes oder Verzeichnisses anzusehen. Für bekannte Datentypen z. B. Bilder (.png, .jpg, .gif, ...) oder Text (.txt, .rtf, .pdf, ...) gibt es außerdem die Möglichkeit sich die Objekte innerhalb der BLOB-Verwaltung anzeigen zu lassen.

Nach dem Aufruf des Menüpunktes "Extras" / "BLOB-Verwaltung..." wird folgender Dialog angezeigt:

The screenshot shows the 'BLOB-Verwaltung - [Wurzelverzeichnis]' dialog window. On the left, a tree view displays the root folder 'Wurzelverzeichnis' containing four subfolders: 'Bilder', 'Customer', 'Documents', and 'Ressourcen'. On the right, a list view shows the same four items in a table format with columns: Name, Größe, Zuletzt geändert, and MIME-Typ. At the bottom, a status bar indicates '0 Objekte (0 Bytes), 4 Verzeichnisse'.

Name	Größe	Zuletzt geändert	MIME-Typ
Bilder			
Customer			
Documents			
Ressourcen			

Kontakt

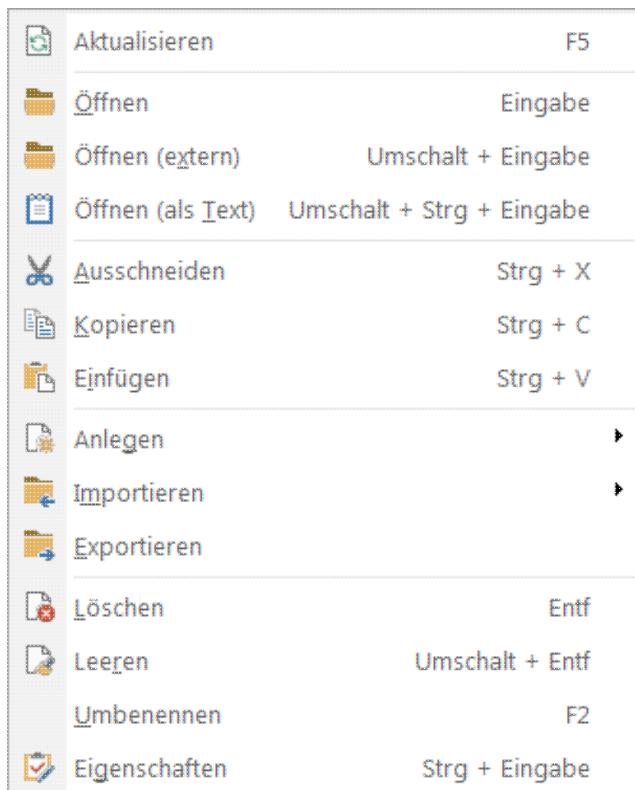
Das Layout der BLOB-Verwaltung orientiert sich am Windows Explorer. Die obere Leiste des Dialogs dient zur Navigation und bietet die Möglichkeit zum vorherigen , zum nächsten und zum übergeordneten Verzeichnis zu springen. Daneben ist eine Adressleiste, die den Pfad des aktuellen Verzeichnisses anzeigt. Über diese kann ebenfalls navigiert werden. In einer Popup-Liste werden Vorschläge angezeigt. Existiert ein angegebenes Verzeichnis nicht, wird eine Fehlermeldung ausgegeben. Ganz rechts in dieser Leiste befindet sich eine Aktualisierungsschaltfläche . Diese baut den Baum und die Liste unterhalb der Navigation neu auf.

Auf der linken Seite des Dialogs ist die Verzeichnisstruktur, der Datenbank, in dem die binären Objekte abgelegt sind, zu sehen. Auf der rechten Seite wird der Inhalt (Verzeichnisse und Objekte) des aktuell ausgewählten Verzeichnisses dargestellt. Die Aktionen werden über das Kontextmenü gestartet.

Das Kontextmenü kann sowohl in dem Baum, als auch in der Liste mit der rechten Maustaste aufgerufen werden. Es erscheint folgendes Menü:

Wird zusätzlich die -Taste gedrückt, erscheint das erweiterte Kontextmenü. Dieses variiert, je nachdem ob man sich auf einem Objekt oder einem Verzeichnis befindet.

Erweitertes Kontextmenü Objekt:



Erweitertes Kontextmenü Verzeichnis:

Kontakt

	Aktualisieren	F5
	Öffnen	Eingabe
	Öffnen (neues Fenster)	Umschalt + Eingabe
	Ausschneiden	Strg + X
	Kopieren	Strg + C
	Einfügen	Strg + V
	Anlegen	▶
	Importieren	▶
	Exportieren	
	Löschen	Entf
	Leeren	Umschalt + Entf
	Umbenennen	F2
	Eigenschaften	Strg + Eingabe

- **Aktualisieren**

Aktualisiert den Baum und die Liste. Diese Aktion wird ebenfalls über die Taste ausgelöst.

- **Öffnen**

Objekt: Es wird eine Vorschau zu dem Objekt angezeigt. Es können nur bekannte Datentypen angezeigt werden. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

Verzeichnis: Das Verzeichnis wird geöffnet. Die Liste neu geladen.

- **Öffnen (extern)**

Das Objekt wird mit dem in Windows definierten Standard-Programm geöffnet. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Öffnen (als Text)**

Das Objekt wird intern als Text angezeigt. Nicht darstellbare Zeichen in binär-codierten Dateien werden ignoriert. Diese Aktion wird ebenfalls über die Tasten + + ausgelöst.

- **Öffnen (neues Fenster)**

Es öffnet sich ein neues MDI-Fenster mit dem ausgewählten Verzeichnis. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Ausschneiden**

Schneidet das aktuell selektierte binäre Objekt bzw. Verzeichnis aus. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Kopieren**

Kopiert das aktuell selektierte binäre Objekt bzw. Verzeichnis. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Einfügen**

Fügt das binäre Objekt bzw. Verzeichnis der Zwischenablage in das aktuelle oder derzeit selektierte Verzeichnis ein. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Anlegen**

Legt ein binäres Objekt oder Verzeichnis an. Ist kein Verzeichnis selektiert, wird das Objekt bzw. Verzeichnis im aktuellen Verzeichnis angelegt. Ansonsten im selektierten. Dazu öffnet sich ein Eingabefenster in der ein Name für das binäre Objekt bzw. Verzeichnis definiert werden muss. Diese Aktion wird ebenfalls über die -Taste (für Datei) bzw. die Tasten + (für Verzeichnis) ausgelöst.

- **Importieren**

Importiert eine Datei oder ein Verzeichnis in die Datenbank.

- **Exportieren**

Exportiert das ausgewählte Objekt bzw. Verzeichnis aus der Datenbank.

- **Löschen**

Löscht das ausgewählte Objekt bzw. Verzeichnis. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

- **Leeren**

Objekt: Leert das Objekt, der Inhalt ist danach leer. Das Objekt besitzt dann eine Größe von 0 Bytes.

Verzeichnis: Alle untergeordneten Objekte und Verzeichnisse werden gelöscht.

Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Umbenennen**

Benennt das aktuell ausgewählte Objekt bzw. Verzeichnis um. Diese Aktion wird ebenfalls über die Taste ausgelöst.

- **Eigenschaften**

Zeigt die Eigenschaften zu einem binären Objekt bzw. Verzeichnis an. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

In der Statusleiste des Dialogs werden Informationen zum Inhalt des aktuellen Verzeichnisses dargestellt. Dabei werden nur die Dateien und Verzeichnisse gezählt, die direkt in dem aktuellen Verzeichnis liegen.

4 Objekte (667,70 KB), 1 Verzeichnis

Kontakt

Die BLOB-Verwaltung unterstützt weiterhin Drag & Drop zwischen MDI-Frames und dem Windows Explorer. Allerdings ist die Anzahl der per Drag & Drop übertragbaren Objekte auf ein Limit von 50 beschränkt.

Kontakt

BLOB-Verwaltung Eigenschaften

Eigenschaften von binären Objekten und Verzeichnissen

Sollen die Eigenschaften eines binären Objektes in der BLOB-Verwaltung angezeigt werden, öffnet sich der folgende Dialog:

Auf der Seite "Allgemein" werden die Objektinformationen angezeigt, die für jedes binäre Objekt zur Verfügung stehen. Die Felder Name, Benutzerdefiniert (System), MIME und Benutzerdefiniert (Typ) können verändert werden. Über die Schaltfläche können mehrzeilige Texte eingegeben werden.

Bei den Dateitypen .bmp, .jpg, jpeg, .gif, .png, .tif, .tiff, .ico, .emf, .wmf, .pdf existiert zusätzlich die Seite "Details", auf der Dateispezifische Eigenschaften aufgelistet werden. Dies können bei Bildern beispielsweise Informationen zu den Ausmaßen oder eventuell vorhandene Exif-Informationen sein.

Für Verzeichnisse sieht der Dialog wie folgt aus:

Hier können die Felder Name und Benutzerdefiniert (System) verändert werden. Über die Schaltfläche wird die Größe der enthaltenen Objekte (rekursiv) berechnet. Dieser Vorgang kann, abhängig von der darunterliegenden Struktur, einige Zeit in Anspruch nehmen.

BLOB-Verwaltung Export

Exportieren eines Objektes oder eines Verzeichnisses

Speicherort auswählen

Soll ein Objekt bzw. ein Verzeichnis in der BLOB-Verwaltung exportiert werden, öffnet sich der folgende Assistent.

Im ersten Schritt fordert der Assistent zur Auswahl eines Speicherortes auf. Diese kann mit der Schaltfläche "Auswählen..." oder durch direkte Eingabe gewählt werden. Ist eine Datei oder ein Verzeichnis ausgewählt, wird die Schaltfläche "Weiter" aktiv.

Ein Klick auf Hilfe zeigt die Dokumentation zur aktuellen Seite des Assistenten an.

Wird ein einzelnes Objekt exportiert, beendet sich der Assistent im erfolgreichen Fall bereits nach dieser Seite. Ein Fortschritt ist bei einer Datei nicht notwendig.

Fortschritt

Auf der folgenden Seite wird der Fortschritt des für Verzeichnisse Importvorgangs angezeigt.

Ist der Exportvorgang abgeschlossen, werden die Schaltflächen "Zurück" und "Abbrechen" entfernt und die Schaltfläche "OK" erscheint. Sind Fehler beim Import aufgetreten, wird der Benutzer darüber informiert.

Export-Einstellungen vornehmen

Sofern eine oder mehrere Dateien verschlüsselt sind wird nach dem Fortschritt ein weitere Seite zum definieren des Entschlüsselungs-Code angezeigt. Der Entschlüsselungs-Code wird standardmäßig mit Sternen angezeigt. Dieser kann über einen Klick auf den Hacken "Sichtschutz" im Klartext dargestellt werden.

Dies sieht bei binären Objekten wie folgt aus:

Wenn der Entschlüsselungs-Code falsch ist, färbt sich das Eingabefeld rot.

Für binäre Verzeichnisse sehen die Export-Einstellungen folgendermaßen aus:

Besitzen mehrere Objekte einen Entschlüsselungscode wird eine Liste der verschlüsselten Dateien angezeigt. Hier kann für jedes Objekt ein Entschlüsselungscode angegeben werden. Mehrere Einträge können mit Multi-Select markiert werden. Der gesetzte Entschlüsselungs-Code wird mit Klick auf für die markierten Einträge übernommen. Einträge, bei denen bereits ein Entschlüsselungs-Code hinterlegt ist, werden in der Liste mit dem -Symbol gekennzeichnet. An diesem Punkt wurden bereits die Objekte, die keinen Entschlüsselungscode besitzen, exportiert. Danach wird wieder die Seite Fortschritt angezeigt.

Kontakt

BLOB-Verwaltung Import

Importieren einer Datei oder eines Verzeichnisses

Datei / Verzeichnis auswählen

Soll eine Datei oder ein Verzeichnis in der BLOB-Verwaltung importiert werden, öffnet sich der folgende Assistent:

Im ersten Schritt fordert der Assistent zur Auswahl einer Datei bzw. eines Verzeichnisses, das importiert werden soll, auf. Diese kann mit der Schaltfläche "Auswählen..." oder durch direkte Eingabe gewählt werden. Ist eine Datei ausgewählt, wird die Schaltfläche "Weiter" aktiv.

Ein Klick auf "Hilfe" zeigt die Dokumentation zur aktuellen Seite des Assistenten an.

Import-Einstellungen vornehmen

Nach dem Klick auf "Weiter" kann konfiguriert werden, wie die Datei oder das Verzeichnis importiert werden soll.

Dies sieht bei Dateien wie folgt aus:

Neben dem Namen und einem MIME-Typ kann hier die Kompressionsstufe und der Verschlüsselungs-Code definiert werden. Der MIME-Typ wird dabei automatisch vorgeschlagen, sofern die Dateiendung bekannt ist. Der Verschlüsselungs-Code wird standardmäßig mit Sternen angezeigt. Dieser kann über einen Klick auf den Hacken "Sichtschutz" im Klartext dargestellt werden.

Für Verzeichnisse sehen die Import-Einstellungen folgendermaßen aus:

Wird ein Verzeichnis importiert, wirken sich die Einstellungen auf alle Dateien und Verzeichnisse aus, die in diesem Verzeichnis liegen. Dabei behalten die binären Objekte den Namen der externen Datei. Mit dem Setzen des Hakens der Option "MIME-Typ automatisch erkennen" wird der MIME-Typ der binären Objekte für bekannte Dateiendungen automatisch gesetzt.

Der hier angegebene Verschlüsselungs-Code wirkt sich nur auf die importierten Dateien aus.

Mit einem Klick auf "Weiter" wird der Importvorgang gestartet. Dieser kann je nach Menge und Größe der Dateien einige Zeit in Anspruch nehmen.

Fortschritt

Auf der folgenden Seite wird der Fortschritt des Importvorgangs angezeigt.

Ist der Importvorgang abgeschlossen, werden die Schaltflächen "Zurück" und "Abbrechen" entfernt und die Schaltfläche "OK" erscheint. Sind Fehler beim Import

Kontakt

aufgetreten, wird der Benutzer darüber informiert.

BLOB-Verwaltung Objektanzeige

Anzeige eines Objektes in der BLOB-Verwaltung

In der BLOB-Verwaltung können verschiedene Objekte angezeigt werden. Ob ein Objekt angezeigt werden kann oder nicht hängt vom Dateityp ab. Um diesen zu bestimmen, wird zuerst die Dateiendung verwendet. Ist diese unbekannt oder nicht gesetzt, wird der MIME-Typ ausgelesen. Ist dieser auch unbekannt oder nicht gesetzt, kann das Objekt nicht angezeigt werden.

Für jeden Dateityp gibt es einen eigenen Dialog:

Bilder

MIME-Typ: image/*

Dateiendungen: .bmp, .jpg, .jpeg, .gif, .png, .tif, .tiff, .ico, .emf, .wmf

Texte

MIME-Typ: text/*, application/pdf

Dateiendungen: .txt, .rtf, .htm, .html, .xhtml, .pdf

Office-Dateien

MIME-Typ: application/doc, application/msword, application/xls, application/vnd.ms-excel, application/ppt, application/mspowerpoint

Dateiendungen: .doc, .docx, .xls, .xlsx, .ppt, .pptx



Zum Anzeigen von Office-Dateien muss die Office-Erweiterung installiert sein.
Mit der DocEdit-Erweiterung können nur Word-Dateien angezeigt werden.



Änderungen an Inhalten der Objektanzeige werden nicht in das binäre Objekt übernommen.

Benutzerverwaltung

Vergabe und Verwaltung von benutzerabhängigen Rechten

Siehe [Blog](#)

Die Benutzerverwaltung wird im Designer über das Menü **Extras /**

Benutzerverwaltung... aufgerufen. Im folgenden wird das Benutzersystem und der Umgang damit erläutert:

- [Benutzer](#)
- [Benutzergruppen](#)
- [Elementgruppen](#)
- [Zusammenarbeit der Objekte](#)
- [Berechtigungen](#)
- [Verwaltung](#)

Benutzer

Ausgangspunkt eines jeden Benutzersystems ist der Benutzer selbst. In CONZEPT 16 wird der Benutzer durch ein [Benutzer-Objekt](#) repräsentiert. Jedes Benutzer-Objekt verfügt über verschiedene [Eigenschaften](#) (Properties). Die wichtigsten Eigenschaften sind der Name ([_UrmPropName](#)) und das Kennwort ([_UrmPropPassword](#)), mit denen sich der Anwender an der Datenbank anmeldet. Der Name muss innerhalb einer Datenbank eindeutig sein. Neben den vordefinierten Eigenschaften kann das Benutzer-Objekt um zusätzliche Eigenschaften erweitert werden. Die Applikation kann dadurch zusätzliche Daten im Benutzer-Objekt speichern.

In einer neu erstellten Datenbank sind bereits die beiden Benutzer SUPERUSER und USER mit leerem Kennwort angelegt. Bei der Anmeldung als SUPERUSER reicht die Angabe von SU als Benutzer aus. Der Benutzer SUPERUSER kann nicht gelöscht werden.

Unabhängig vom verwendeten Client ist zum Öffnen einer Datenbank immer die Angabe eines Benutzers erforderlich. Alle Aktionen (Ausführen von Prozeduren, Laden von Dialogen usw.) finden somit immer in einem Benutzerkontext statt.

Die Berechtigungen eines Benutzers ergeben sich aus seiner Zugehörigkeit zu einer oder mehreren [Benutzergruppen](#), im Benutzer-Objekt selbst werden keine Rechte definiert.

Benutzergruppen

Eine Benutzergruppe fasst mehrere Benutzer zusammen, um eine einfachere Vergabe von Rechten zu gewährleisten. Die Berechtigungen werden immer für eine Benutzergruppe definiert, wodurch alle Mitglieder dieser Gruppe die selben Rechte erhalten.

Ein Benutzergruppen-Objekt wird durch seinen Namen identifiziert und verfügt über eine Reihe von vordefinierten [Eigenschaften](#). Genau wie beim Benutzer-Objekt können zusätzliche Eigenschaften angelegt werden. Neben seinen Eigenschaften verfügt das Benutzergruppen-Objekt über Rechte auf mehrere [Elementgruppen](#). Ein Benutzer-Objekt kann Mitglied in mehreren Benutzergruppen sein. Die effektiven Rechte des Benutzers ergeben sich in diesem Fall durch die Addition der Rechte aller

Kontakt

Benutzergruppen, zu denen er gehört.

In einer leeren Datenbank sind bereits die beiden Benutzergruppen `_Everyone` und `_Administrators` vorhanden. Die Gruppe `_Everyone` dient als Basisgruppe für alle Benutzer, um die minimal benötigten Rechte für die Applikation zur Verfügung zu stellen. Die Gruppe `_Administrators` verfügt automatisch über alle Rechte in der Datenbank. Beide Gruppen können nicht gelöscht werden. Der Benutzer `SUPERUSER` ist Mitglied von `_Administrators` und der Benutzer `USER` Mitglied von `_Everyone`.

Elementgruppen

Genauso wie Benutzergruppen mehrere Benutzer mit den gleichen Benutzerrollen zusammenfassen, fassen Elementgruppen mehrere Applikationselemente zusammen, für die die selben Rechte benötigt werden. Die Elementgruppe kann sowohl reale Objekte aus der Datenbank (bestimmte Dialoge, einzelne Prozeduren, ein oder mehrere Datensatz-Tabellen usw.) als auch virtuelle Objekte der Applikation (Programm-Module, Funktionalitäten usw.) beinhalten. Durch die Zusammenfassung von mehreren Elementen zu Elementgruppen wird die Rechtevergabe wesentlich übersichtlicher. Gerade wenn es um die Berechtigungen eines Anwendungsbereichs geht, können alle Prozeduren, Dialoge, Tabellen usw. aus diesem Bereich in eine Elementgruppe zusammengefasst werden. Für alle Elemente werden dann gleiche Rechte vergeben. Jedes reale oder virtuelle Element kann Mitglied in bis zu 128 verschiedenen Elementgruppen sein (maximal 32 bei Texten und Prozeduren).

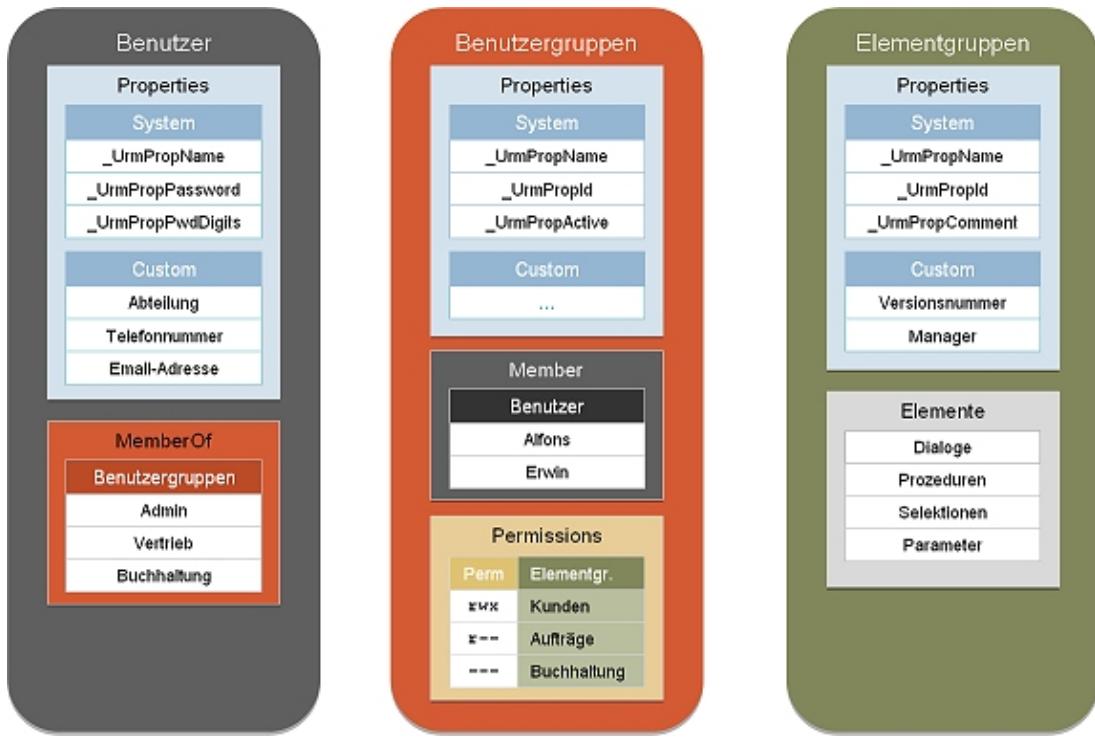
Auch ein Elementgruppen-Objekt wird durch seinen Namen identifiziert und verfügt sowohl über vordefinierte als auch über anwendungsdefinierte Eigenschaften.

In jeder Datenbank sind mehrere sogenannte Standard-Elementgruppen vordefiniert. In diesen Elementgruppen sind jeweils alle Elemente eines bestimmtem Typs enthalten. Zum Beispiel gehören alle Prozeduren automatisch zur Gruppe `_procedure`. Eine Liste aller Standard-Elementgruppen befindet sich im Abschnitt [Verwaltung der Elementgruppen](#).

Zusammenarbeit der Objekte

Die drei Objekte der Benuterverwaltung treten über Listen miteinander in Beziehung. Innerhalb des Benutzer-Objekts gibt es eine Mitgliedsliste von Benutzergruppen. Bei den Benutzergruppen gibt es eine Liste mit den Mitgliedern (Benutzer-Objekte) und eine Liste mit Elementgruppen einschließlich deren Rechte.

Kontakt



Ob ein Benutzer ein bestimmtes Recht an einem Applikationselement besitzt, hängt also davon ab, ob er Mitglied einer Benutzergruppe ist, die das entsprechende Recht besitzt. Innerhalb der Benutzergruppe muss eine Elementgruppe mit dem entsprechenden Recht eingetragen sein, in der das Element enthalten ist. Ein Recht wird letztendlich in der Benutzergruppe definiert, wo die Elementgruppe eingetragen ist.

Berechtigungen

In CONZEPT 16 werden folgende Berechtigungen verarbeitet:

Read (R)	<u>_UrmPermRead</u>	Lesen und Anzeigen
Create (C)	<u>_UrmPermCreate</u>	Anlegen
Modify (M)	<u>_UrmPermModify</u>	Ändern
Modify Owner (MO)	<u>_UrmPermModifyOwner</u>	Ändern bei Objektbesitz
Delete (D)	<u>_UrmPermDelete</u>	Löschen
Delete Owner (DO)	<u>_UrmPermDeleteOwner</u>	Löschen bei Objektbesitz
Execute (X)	<u>_UrmPermExecute</u>	Ausführen
Config (CF)	<u>_UrmPermConfig</u>	Konfiguration

Da sich die Rechte jeweils auf die **Elemente** in einer Elementgruppe beziehen, gibt es für die Elementgruppe selbst drei zusätzliche Rechte:

Read Element (RE)	<u>_UrmPermElmGroupRead</u>	Berechtigung Elementgruppen lesen
Insert Element (IE)	<u>_UrmPermElmGroupInsert</u>	Berechtigung Element zur Gruppe hinzufügen <u>_UrmPermElmGroupDelete</u>

Kontakt

Delete Element
(DE)

Berechtigung Element aus Gruppe
löschen

Speziell für das Zuordnen von **Benutzern** zu **Benutzergruppen** gibt es die folgenden Berechtigungen:

Insert Member [UrmPermElmGroupRead](#) Berechtigung Benutzer zu Benutzergruppe hinzufügen
(IM)

Delete Member [UrmPermElmGroupInsert](#) Berechtigung Benutzer aus Benutzergruppe entfernen
(DM)

Eine Berechtigung für ein Element entsteht, wenn einer Benutzergruppe eine Elementgruppe hinzugefügt wird und damit gleichzeitig die Rechte auf diese Elementgruppe definiert werden. Ist ein Element in keiner Elementgruppe einer Benutzergruppe enthalten, hat diese Benutzergruppe keine Berechtigung für dieses Element. Die effektiven Rechte für einen Benutzer entstehen durch die Kombination aller Rechte aus den Benutzergruppen, in denen der Benutzer Mitglied ist.

Eine Berechtigung kann erteilt oder entzogen werden. Auf diese Weise können auch "Negativ-Gruppen" erstellt werden, die bestimmte Rechte wieder entfernen. Gibt es für ein Applikationselement sowohl das erteilte, wie das entzogene Recht, setzt sich das entzogene Recht durch.

Beispiel:

In einer Elementgruppe "ElmTblSales" sind alle Tabellen zusammengefasst, die zum Erstellen von Angeboten, Abwickeln von Aufträgen usw. notwendig sind. Darüber hinaus existieren die Elementgruppen "ElmProcCom" und "ElmProcSales" in denen alle Bibliotheken und alle Prozeduren, die zur Abwicklung von Angeboten und Aufträgen benötigt werden, zusammengefasst sind. Es werden drei Benutzergruppen erstellt: "Sales", "Developer" und "Visitor". Folgende Rechte werden vergeben:

	_tables	ElmTblSales	ElmProcSales	ElmProcCom
Visitor	M- D- CF-		C- M- D-	C- M- D-
Sales		R+ C+ M+ D+	X+	X+
Developer		R+ C+ M+ D+ X+ CF+	R+ C+ M+ D+ X+	R+ C+ M+ D+ X+

Die Elementgruppe "**_tables**" wird von CONZEPT 16 zur Verfügung gestellt und enthält alle Tabellen der Datenbank. Die mit einem Minuszeichen versehenen Rechte werden der Benutzergruppe entzogen, die Rechte mit einem Pluszeichen werden erteilt.

Ein Benutzer der Benutzergruppe "Sales" hat innerhalb der relevanten Tabellen alle Rechte, die er benötigt, um Datensätze zu lesen, zu erzeugen, zu ändern und zu löschen. Die notwendigen Prozeduren darf er ausführen. Ein Einsehen der Prozeduren oder gar Ändern und neu übersetzen ist ihm nicht möglich. Da der Benutzer nur in einer Benutzergruppe Mitglied ist, bestehen seine effektiven Rechte auch nur aus den in der Benutzergruppe angegebenen Rechten.

Ein weitere Benutzer wird angelegt. Dieser Benutzer ist nur vorübergehend Gast im Verkauf der Firma. Der Benutzer wird den Benutzergruppen "Sales" und "Visitor" zugeordnet. Seine effektiven Rechte sind durch die Besucher-Gruppe eingeschränkt,

Kontakt

so kann er keine Datensätze Ändern oder löschen. Die entzogenen Rechte in bezug auf die Prozeduren wirken sich nicht weiter aus, der schon die "Sales"-Gruppe hier nur ausführende Rechte besitzt.

Die effektiven Rechte des Besuchers sehen wie folgt aus:

<u>tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Effektiven Rechte	M- D- CF- R+ C+ M- D- CF-	C- M- D- X+	C- M- D- X+
Bekommt die Firma einen weiteren Gast in der Entwicklungsabteilung, so bekommt dessen Benutzer die Mitgliedschaft in den Benutzergruppen "Developer" und "Visitor". Die effektiven Rechte sind dann wie folgt:			

<u>tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Effektiven Rechte	M- D- CF- R+ C+ M- D- X+ CF-	R+ C- M- D- X+	R+ C- M- D- X+

Verwaltung

Die Verwaltung des Benutzersystems kann über den Designer oder prozedural vorgenommen werden. Die Verwaltung über den Designer gliedert sich in die folgenden Bereiche:

- Verwaltung der Benutzer
- Verwaltung der Benutzergruppen
- Verwaltung der Elementgruppen

Für die prozedurale Steuerung des Benutzersystems werden die folgenden Objekte benötigt:

- User (Benutzer)
- UserGroup (Benutzergruppe)
- ElmGroup (Elementgruppe)

Über die Funktion WinUrmDialog() kann die Benutzerverwaltung in die eigene Anwendung integriert werden.

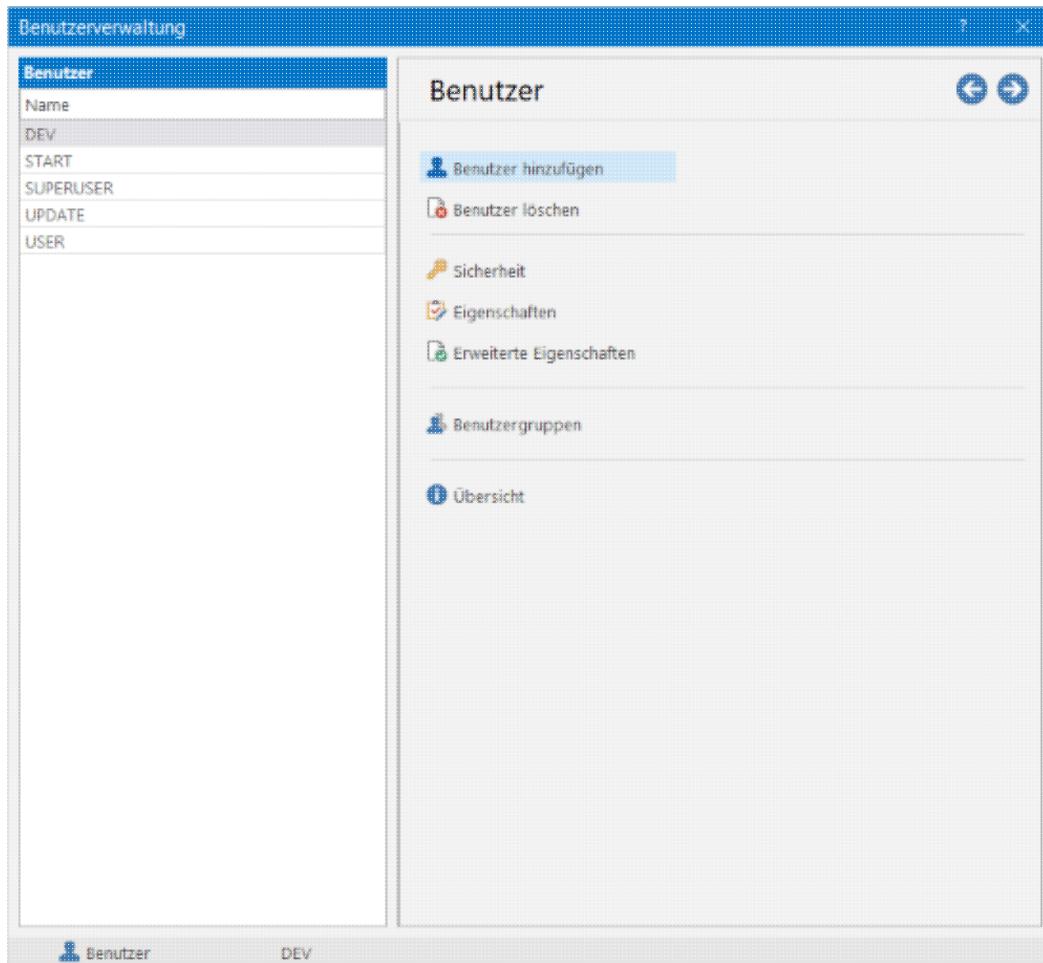
Kontakt

Verwaltung der Benutzer

Anlegen, ändern und löschen von Benutzern

Die Verwaltung der Benutzer wird durch Drücken der Schaltfläche [Benutzer] gestartet. Auf der linken Seite werden in einer Liste alle bereits angelegten Benutzer angezeigt. Auf der rechten Seite befinden sich die zur Verfügung stehenden Funktionen.

Standardmäßig gibt es die zwei Benutzer SUPERUSER und USER als Hauptadministrator und als Vorlage für einen Standardanwender. Der Super-User kann nicht gelöscht und nur von sich selbst oder Mitgliedern der vordefinierten Benutzergruppe _Administrators verändert werden. Der Super-User hat immer eine Mitgliedschaft in der Administratorengruppe.



Neue Benutzer werden durch Drücken der Schaltfläche [Benutzer hinzufügen] angelegt. Vorhandene Benutzer können durch Auswahl des Benutzers auf der linken Seite und Auswahl der entsprechenden Funktion verändert oder gelöscht werden. Folgende Funktionen stehen zur Verfügung:

- Benutzer hinzufügen
- Benutzer löschen
- Sicherheit
- Eigenschaften

- [Erweiterte Eigenschaften](#)
- [Benutzergruppen](#)
- [Übersicht](#)



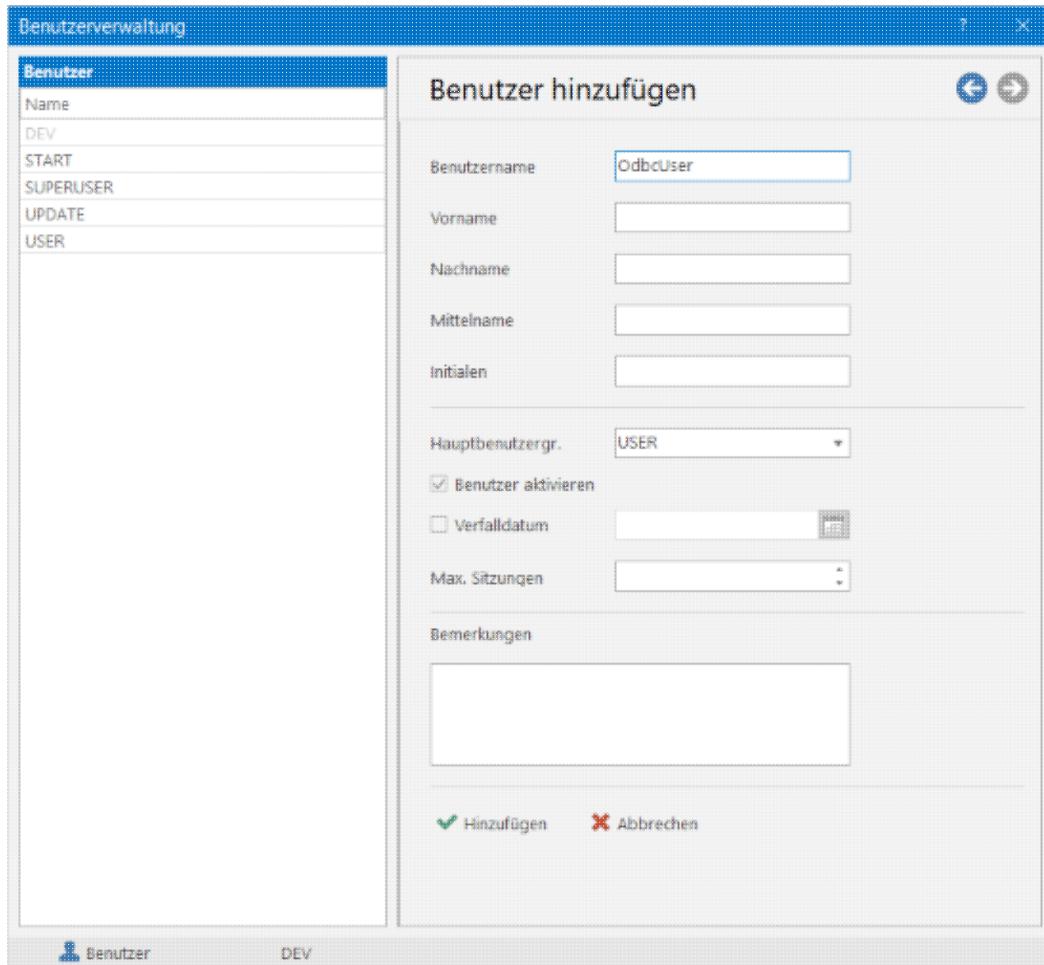
Um Änderungen an einem Benutzer vornehmen zu können wird das Recht UrmPermModify benötigt. Der eigene Benutzer bildet dabei eine Ausnahme. Vom eigenen Benutzer darf die Eigenschaft UrmPropPassword (solange die Eigenschaft UrmPropPwdLocked nicht gesetzt ist) und alle benutzerdefinierten Eigenschaften geändert werden.

Kontakt

Benutzer - Benutzer hinzufügen

Neuen Benutzer erzeugen

Mit dieser Schaltfläche wird ein neuer Benutzer erzeugt. Anschließend können die Eigenschaften des Benutzers angegeben werden.



The screenshot shows the 'Benutzerverwaltung' application interface. On the left, there's a sidebar with a tree view under 'Benutzer' containing nodes like 'Name', 'DEV', 'START', 'SUPERUSER', 'UPDATE', and 'USER'. Below the sidebar are buttons for 'Benutzer' and 'DEV'. The main area is titled 'Benutzer hinzufügen' and contains the following fields:

- Benutzername: OdbcUser
- Vorname: (empty)
- Nachname: (empty)
- Mittlename: (empty)
- Initialen: (empty)
- Hauptbenutzergr.: USER *
- Benutzer aktivieren
- Verfalldatum: (with calendar icon)
- Max. Sitzungen: (dropdown menu)
- Bemerkungen: (text area)

At the bottom are two buttons: a green checkmark icon followed by 'Hinzufügen' and a red cross icon followed by 'Abbrechen'.

Folgende Eigenschaften des Benutzers können gesetzt werden:

- Benutzername / UrmPropName Anmeldename des Benutzers
- Vorname / UrmPropFirstName Vorname des Benutzers
- Nachname / UrmPropLastName Nachname des Benutzers
- Mittlename / UrmPropInitials Zweiter Vorname des Benutzers
- Initialen / UrmPropShortcut Initialen des Benutzers
- Hauptbenutzergr. / UrmPropUserGroup Hauptbenutzergruppe
- Benutzer aktivieren / UrmPropActive Benutzeraccount aktivieren
- Verfallsdatum / UrmPropExpirationDate Verfallsdatum des Accounts
- Sessions / UrmPropMaxSessions Maximale Anzahl der Sessions
- Bemerkung / UrmPropComment Bemerkung zum Benutzer

Der Benutzername muss angegeben werden. Alle anderen Angaben sind optional. Für die Eingaben gelten die gleichen Beschränkungen, wie für die entsprechenden Eigenschaften.

Kontakt

Der Benutzer wird erst angelegt, wenn die Schaltfläche [Hinzufügen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird der Benutzer angelegt. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

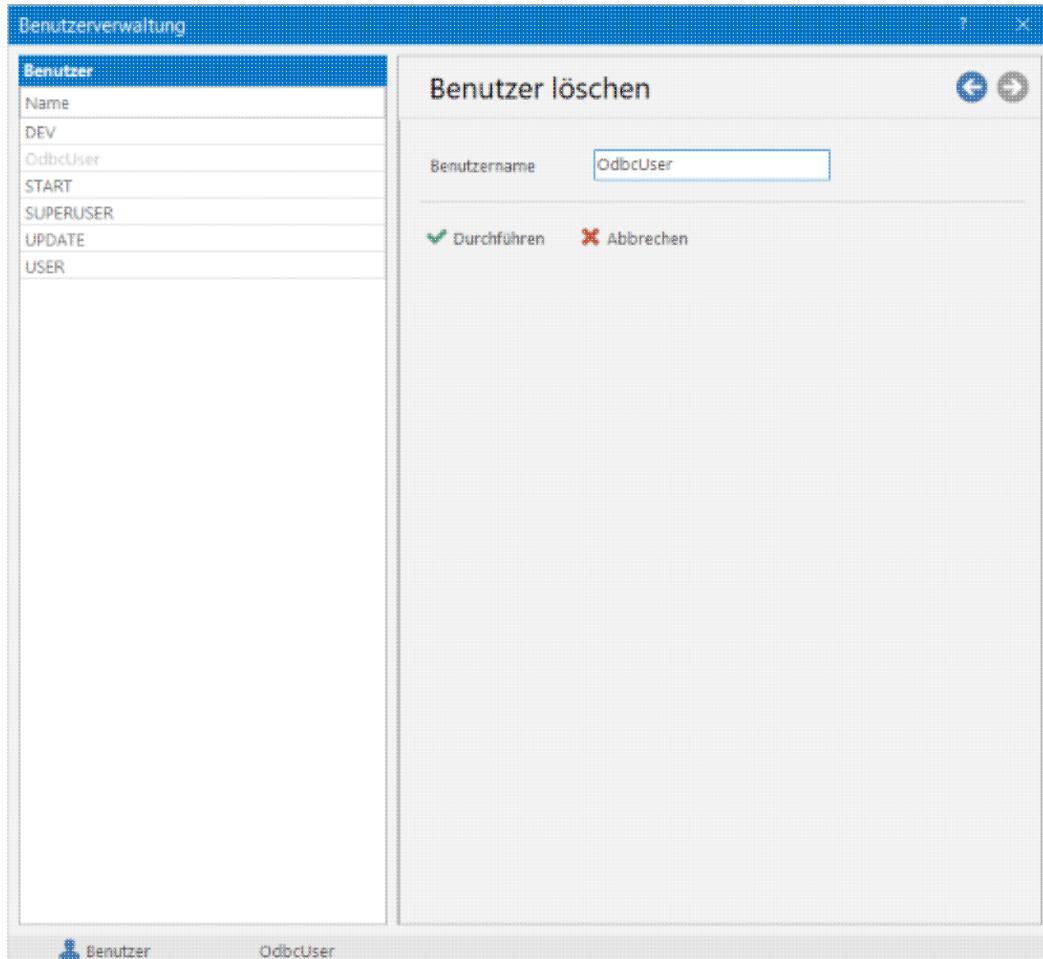
Diese Funktion entspricht der Anweisung [UrmCreate\(\)](#)(0, [UrmTypeUser](#), ...) wobei anschließend der Benutzer mit [UrmOpen\(\)](#) geöffnet und die entsprechenden Eigenschaften gesetzt werden.

Kontakt

Benutzer - Benutzer löschen

Löschen eines Datenbankbenutzers

Mit dieser Schaltfläche wird der links ausgewählte Benutzer gelöscht.



Der Benutzer wird erst gelöscht, wenn die Schaltfläche [Durchführen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird der Benutzer gelöscht. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

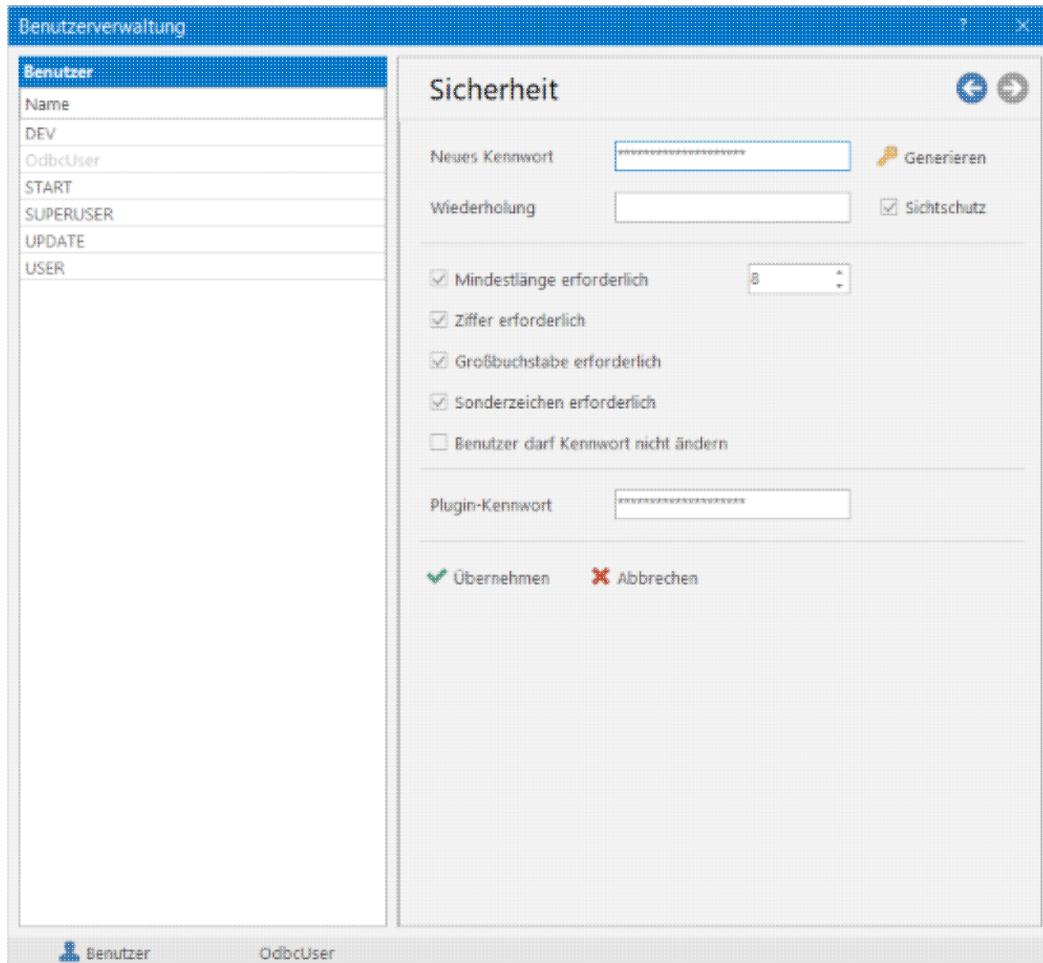
Die Funktion entspricht dem Befehl UrmDelete(0, _UrmTypeUser, <Name>). Die Mitgliederlisten von Benutzergruppen werden automatisch aktualisiert. Der Benutzer SUPERUSER kann nicht gelöscht werden.

Kontakt

Benutzer - Sicherheit

Voraussetzungen für ein gültiges Kennwort

In diesem Bereich werden die Einstellungen bezüglich des Kennwortes des ausgewählten Benutzers vorgenommen.



Folgende Eigenschaften des Benutzer-Objekts können hier gesetzt werden:

- Neues Kennwort und Wiederholung / _UrmPropPassword Neues Kennwort des Benutzers
- Mindestlänge / _UrmPropPwdMinLength Mindestlänge des Kennworts. Die Länge wird in dem Eingabefeld angegeben.
- Ziffer erforderlich / _UrmPropPwdDigits Das Kennwort muss mindestens eine Zahl beinhalten.
- Großbuchstabe erforderlich / _UrmPropPwdCapitals Das Kennwort muss mindestens einen Großbuchstaben beinhalten.
- Sonderzeichen erforderlich / _UrmPropPwdSpecials Das Kennwort muss mindestens ein Sonderzeichen beinhalten.
- Benutzer darf Kennwort nicht ändern / _UrmPropPwdLocked Das Kennwort ist gesperrt und kann nicht vom Anwender geändert werden.
- Plugin-Kennwort / _UrmPropDsgPlugInPwd Das Kennwort für die Plugin-Schnittstelle des Designers.

Kontakt

Über die Schaltfläche [Generieren] kann ein zufälliges Kennwort, das die angegebenen Regeln erfüllt, erzeugt werden. Die Angabe eines neuen Kennworts kann mit einem Sichtschutz erfolgen. Ist diese Checkbox angehakt, werden die eingegebenen Zeichen unkenntlich gemacht. Wird das Kennwort des Benutzers SUPERUSER geändert, erfolgt eine zusätzlich Sicherheitsabfrage.

Die Eigenschaften für die Voraussetzungen des Kennwertes sind alle vom Typ int. Wird in der Benutzeroberfläche ein Haken gesetzt, bekommt die entsprechende Eigenschaft den Wert 1. Prozedural können auch höhere Werte gesetzt werden.

Durch Drücken der Schaltfläche [Übernehmen] werden die Berechtigungen des angemeldeten Benutzers geprüft und bei ausreichenden Rechten die Änderungen wirksam.

Kontakt

Benutzer - Eigenschaften

Eigenschaften des Benutzer

Über die Schaltfläche [Eigenschaften] werden die Eigenschaften des in der Liste ausgewählten Benutzers angezeigt. Die Eigenschaften, die hier geändert werden können, entsprechen weitestgehend den Eigenschaften beim Hinzufügen eines Benutzers.

Folgende Eigenschaften werden beim Anlegen automatisch vorbelegt und können hier nachträglich verändert werden:

- Besitzer / _UrmPropOwner Besitzer
- Besitzergruppe / _UrmPropOwnerGroup Besitzergruppe

The screenshot shows the 'Benutzerverwaltung' interface. On the left, a list of users is displayed: DEV, OdbcUser (selected), START, SUPERUSER, UPDATE, and USER. The main window is titled 'Eigenschaften' and contains the following fields:

- Benutzername: OdbcUser
- Vorname: (empty)
- Nachname: (empty)
- Mittename: (empty)
- Initialen: (empty)
- Besitzer: SUPERUSER
- Besitzergruppe: (empty)
- Hauptbenutzergr.: USER
- Benutzer aktivieren
- Verfalldatum: (calendar icon)
- Max. Sitzungen: (empty)
- Bemerkungen: (empty text area)

At the bottom of the dialog are two buttons: a green checkmark labeled 'Übernehmen' and a red X labeled 'Abbrechen'.



Für den Benutzer SUPERUSER können die Eigenschaften _UrmPropActive und _UrmPropExpirationDate nicht verändert werden.

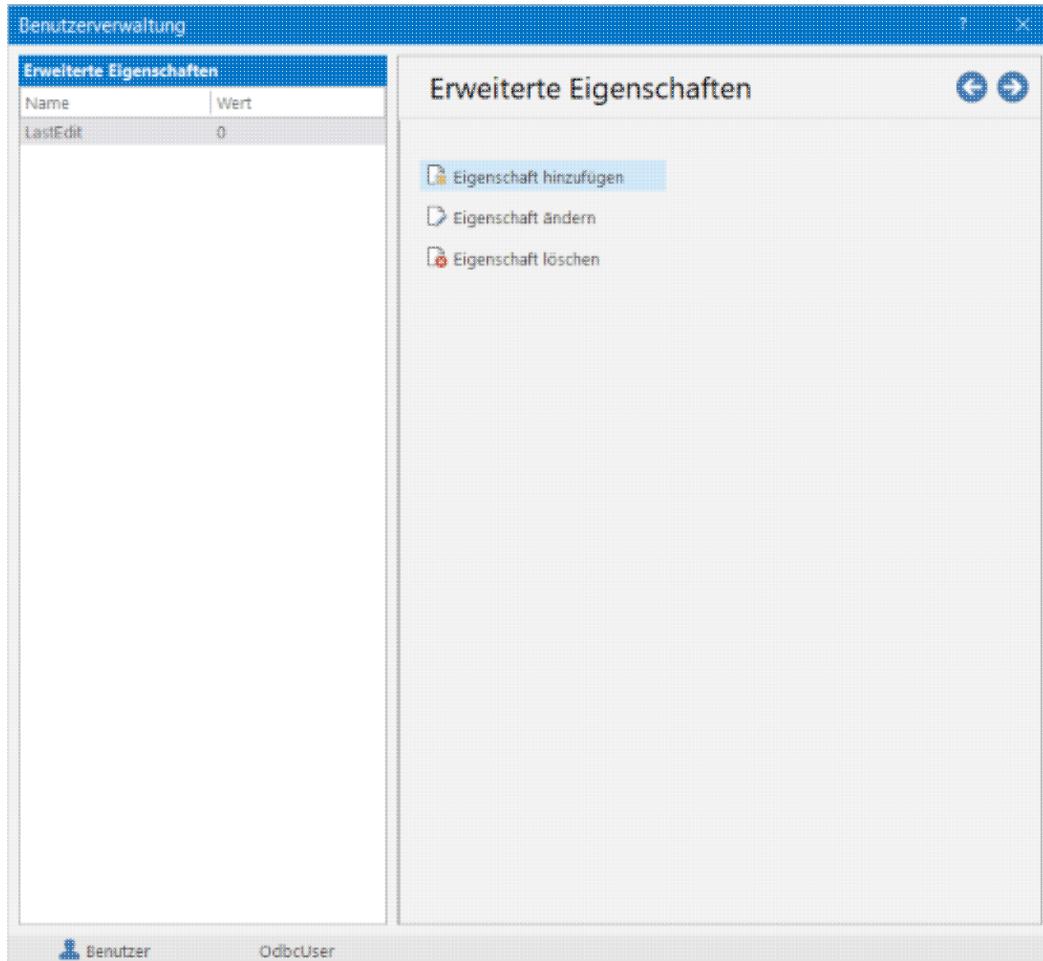
Wird die Hauptbenutzergruppe des Benutzers auf eine Benutzergruppe geändert, in der der Benutzer nicht Mitglied ist, erfolgt eine Abfrage, ob der Benutzer dieser Benutzergruppe hinzugefügt werden soll. Ist der Benutzer nicht Mitglied seiner Hauptbenutzergruppe, können ihm unter Umständen notwendige Berechtigungen in Bezug auf die Startprozedur fehlen.

Kontakt

Benutzer - Erweiterte Eigenschaften

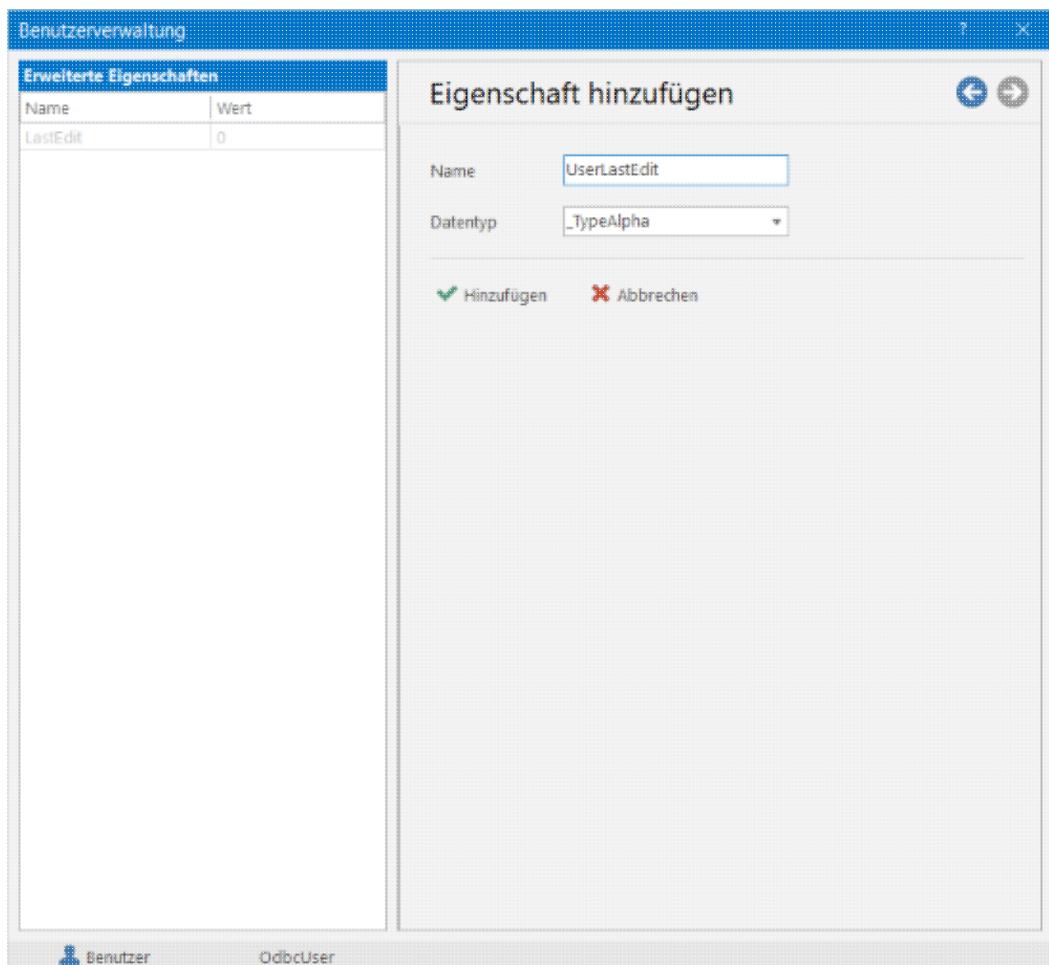
Eigene Eigenschaften definieren

Durch Drücken der Schaltfläche [Erweiterte Eigenschaften] wird diese Seite geöffnet.



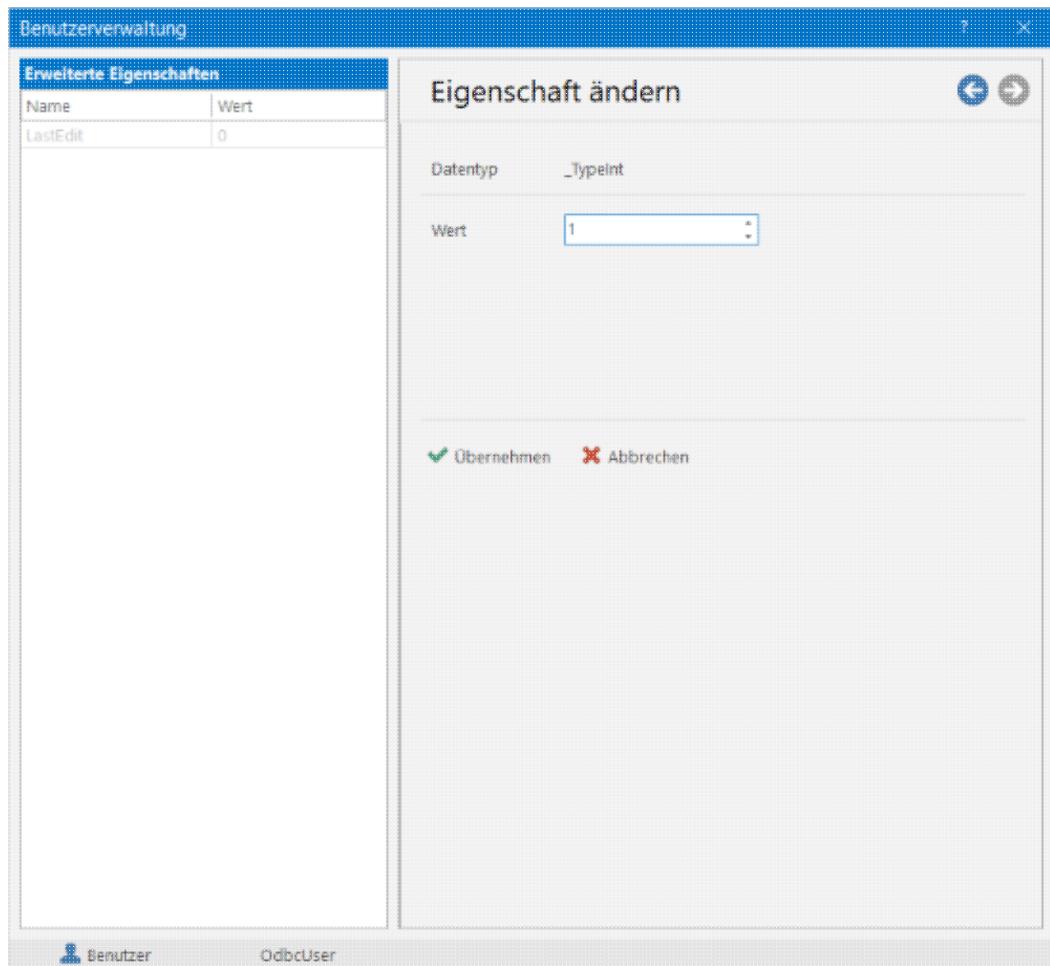
In diesem Bereich können dem ausgewählten Benutzer selbst definierte Eigenschaften zugeordnet werden. Mit der Schaltfläche [Eigenschaft hinzufügen] wird folgender Dialog aufgerufen:

Kontakt



Der Eigenschaft muss ein eindeutiger Name gegeben werden. Der Typ der Eigenschaft wird über die ComboBox ausgewählt. Die Eigenschaft wird erst dann erzeugt, wenn die Schaltfläche [Hinzufügen] gedrückt wird. Zu diesem Zeitpunkt werden auch die Berechtigungen des Benutzers geprüft. Sind diese ausreichend, wird die Eigenschaft angelegt. In der Programmierung entspricht das der Anweisung [UrmCreate\(tHdlUser, _UrmTypeProperty, <Name>, <Type>\)](#). Der Wert der Eigenschaft kann entweder über die Schaltfläche [Eigenschaft ändern] oder zur Laufzeit mit der Anweisung [UrmPropSet\(\)](#) gesetzt werden.

Kontakt



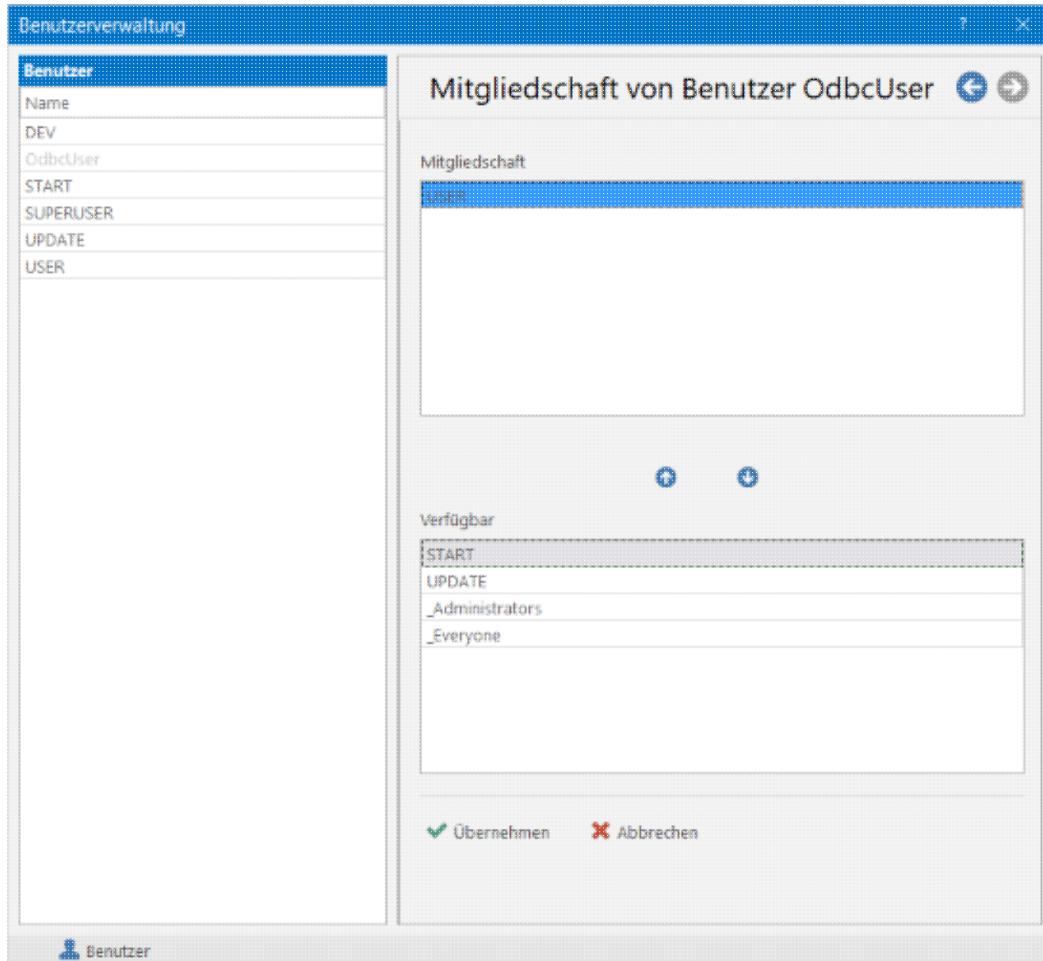
Wird die Eigenschaft nicht mehr benötigt, kann sie mit der Schaltfläche [Eigenschaft löschen] wieder entfernt werden. Zur Laufzeit kann die Eigenschaft mit `UrmDelete()` gelöscht werden.

Kontakt

Benutzer - Benutzergruppen

Zuordnung des Benutzers zu Benutzergruppen

Wird die Schaltfläche [Benutzergruppen] angeklickt, erscheint folgender Dialog:



Hier kann der Benutzer einer oder mehreren Benutzergruppen zugewiesen werden. Innerhalb der Programmierung erfolgt diese Zuweisung über die Anweisung [UrmCreate\(tHdlUser, UrmTypeMember, <Name>\)](#). Die Mitgliedschaft kann mit [UrmDelete\(\)](#) wieder entfernt werden. In der oberen Liste sind alle Benutzergruppen aufgeführt, zu denen der Benutzer bereits gehört, alle anderen Benutzergruppen sind in der unteren Liste aufgeführt. Ein Benutzer kann einer Gruppe hinzugefügt oder aus einer Gruppe entfernt werden, indem die Gruppe in der entsprechenden Liste doppelt geklickt wird oder mit Drag & Drop Benutzergruppen von einer Liste in die andere Liste verschoben werden. Es können auch mehrere Gruppen ausgewählt werden. Alternativ dazu können Einträge (auch mehrere gleichzeitig) auch mit Hilfe der Schaltflächen zwischen den Listen verschoben werden.

Durch Drücken der Schaltfläche [Übernehmen] werden die Berechtigungen des angemeldeten Benutzers geprüft und bei ausreichenden Rechten die Änderungen wirksam.

Kontakt

Benutzer - Übersicht

Anzeige der Eigenschaften und Berechtigungen eines Benutzers

Über die Funktion "Übersicht" können die Berechtigungen und die Eigenschaften eines Benutzers eingesehen werden.

The screenshot shows the 'Benutzerverwaltung' application interface. On the left, a sidebar lists users with 'OdbcUser' selected. The main window has a title 'Übersicht' and contains several sections: 'Benutzer' (Name: OdbcUser, Aktiviert: Ja, Max. Sitzungen: Unbegrenzt, Verfalldatum: Nie), 'Erweitert' (Besitzer: SUPERUSER, Besitzergruppe: null, Hauptbenutzergruppe: START), 'Mitgliedschaft' (START, _Everyone), and 'Berechtigungen' (START, <Leer>). Below these, a detailed view of permissions for '_Everyone' is shown for various element groups: dialog (R C M D X), _elementgroup (RE), menu (R C M D X), metapicture (R C M D X), and picture (R C M D X). At the bottom right is a 'Schließen' (Close) button.

Die Anzeige erfolgt in verschiedenen Bereichen:

- **Benutzer und Erweitert**

Hier werden einige Informationen zu dem Benutzer ausgegeben. Die Informationen werden im Bereich Eigenschaften bzw. Benutzer hinzufügen gesetzt.

- **Mitgliedschaft**

Hier werden alle Benutzergruppen angezeigt, in denen der Benutzer Mitglied ist. Die Zuweisung von Benutzern zu Benutzergruppen erfolgt bei Benutzer / Benutzergruppen bzw. bei Benutzergruppen / Benutzer.

- **Berechtigungen**

Dieser Bereich wird für jede Benutzergruppe angezeigt. Dahinter werden alle Elementgruppen mit ihren Berechtigungen aufgelistet. Die Berechtigungen werden im Bereich Benutzergruppen / Elementgruppen gesetzt.

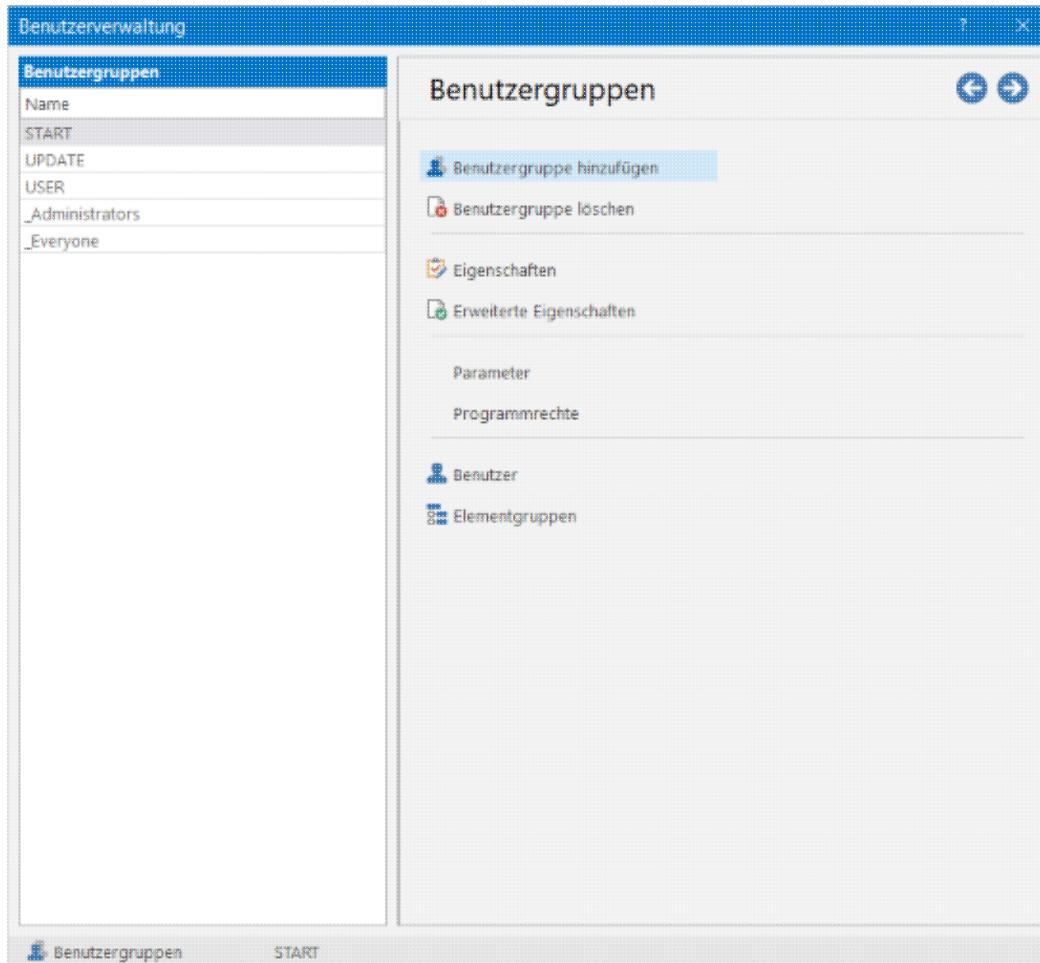
Kontakt

Verwaltung der Benutzergruppen

Anlegen, ändern und löschen von Benutzergruppen

Die Verwaltung der Benutzergruppen wird durch Drücken der Schaltfläche [Benutzergruppen] gestartet. Auf der linken Seite werden in einer Liste alle bereits angelegten Benutzergruppen angezeigt. Auf der rechten Seite befinden sich die zur Verfügung stehenden Funktionen.

Standardmäßig gibt es die zwei Benutzergruppen _Administrators und _Everyone als Administratorengruppe und als Gruppe mit den gewünschten Minimalrechten für Datenbank und Datenbankapplikation. Beide Benutzergruppen können nicht gelöscht werden. Die Administratorengruppe kann zusätzlich nur von Mitgliedern der Gruppe verändert werden. Die Rechte der Administratorengruppe umfassen das gesamte System.



Die Funktionen entsprechen weitgehend den Funktionen der Benutzerverwaltung.

- Benutzergruppe hinzufügen
- Benutzergruppe löschen
- Eigenschaften
- Erweiterte Eigenschaften
- Parameter
- Programmrechte

Kontakt

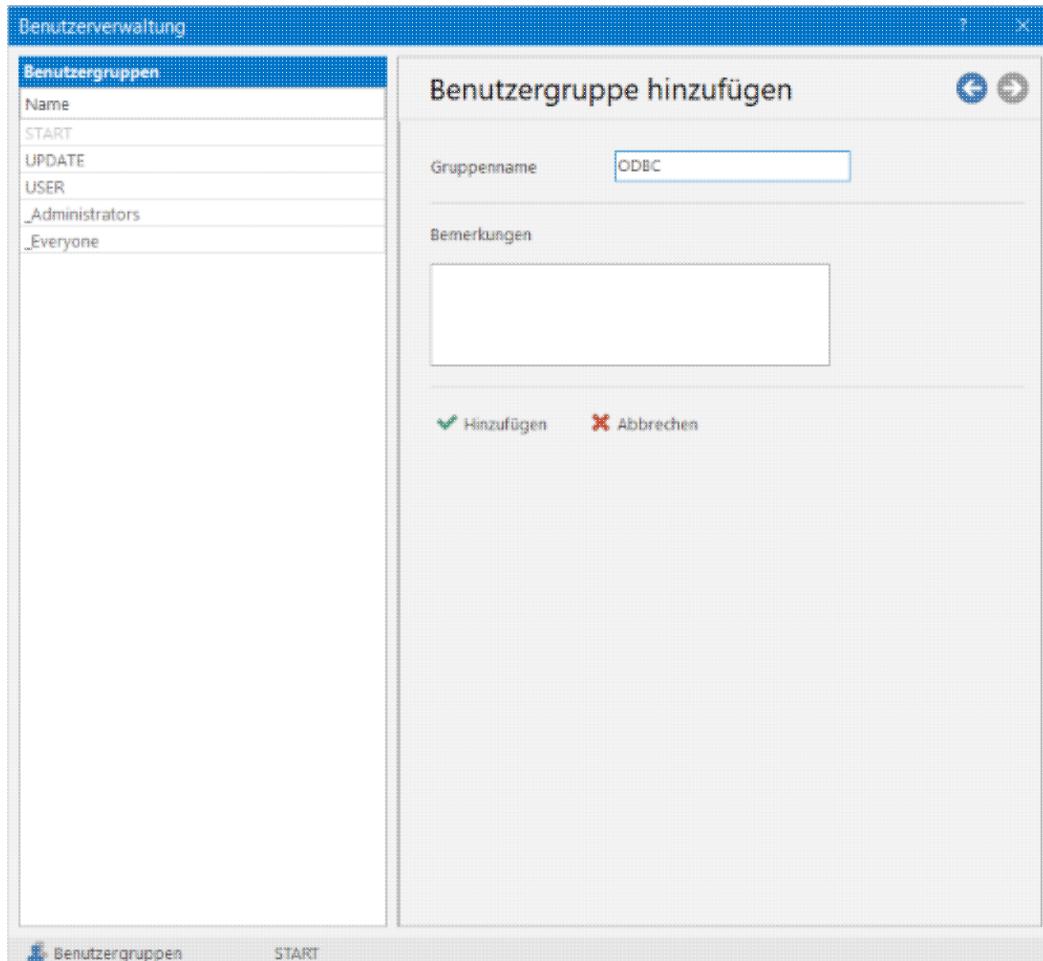
- Benutzer
- Elementgruppen

Kontakt

Benutzergruppen - Benutzergruppe hinzufügen

Neue Benutzergruppe erzeugen

Mit dieser Schaltfläche wird eine neue Benutzergruppe erzeugt. Anschließend werden die Eigenschaften der Benutzergruppe angezeigt und können ausgefüllt werden.



Folgende Eigenschaften der Benutzergruppe können gesetzt werden:

- Benutzergruppenname / _UrmPropName Name der Benutzergruppe
- Bemerkung / _UrmPropComment Bemerkung zur Benutzergruppe

Der Name der Benutzergruppe muss angegeben werden. Die Angabe einer Bemerkung ist optional. Für die Eingaben gelten die gleichen Beschränkungen, wie für die entsprechenden Eigenschaften.

Die Benutzergruppe wird erst angelegt, wenn die Schaltfläche [Hinzufügen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird die Benutzergruppe angelegt. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

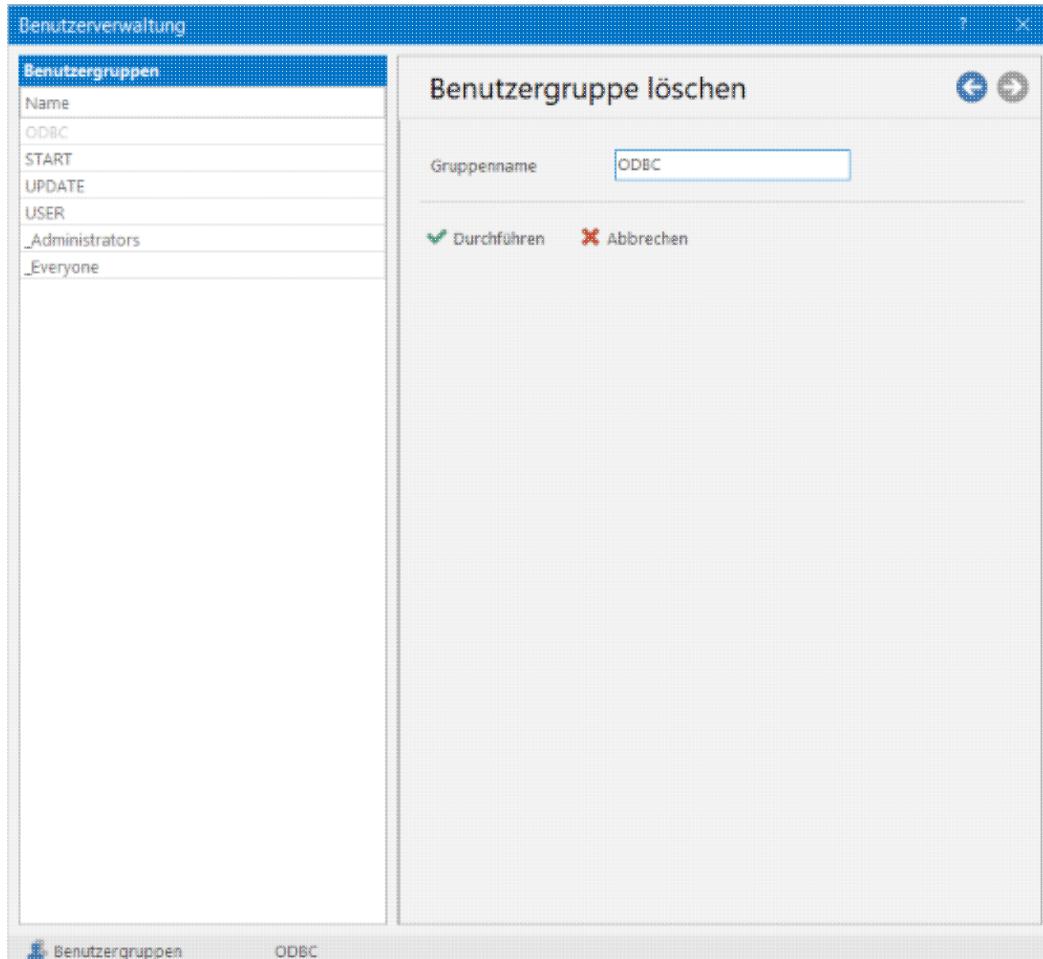
Diese Funktion entspricht der Anweisung `UrmCreate(0, _UrmTypeUserGroup, ...)` wobei anschließend die Benutzergruppe mit `UrmOpen()` geöffnet und die entsprechenden Eigenschaften gesetzt werden.

Kontakt

Benutzergruppen - Benutzergruppe löschen

Löschen einer Benutzergruppe

Mit dieser Schaltfläche wird die links ausgewählte Benutzergruppe gelöscht.



Die Benutzergruppe wird erst gelöscht, wenn die Schaltfläche [Durchführen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird die Benutzergruppe gelöscht. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

Die Funktion entspricht dem Befehl [UrmDelete\(0, _UrmTypeUserGroup, <Name>\)](#). Die Mitgliederlisten der Benutzer werden automatisch aktualisiert. Die Benutzergruppen _Administrator und _Everyone können nicht gelöscht werden.

Kontakt

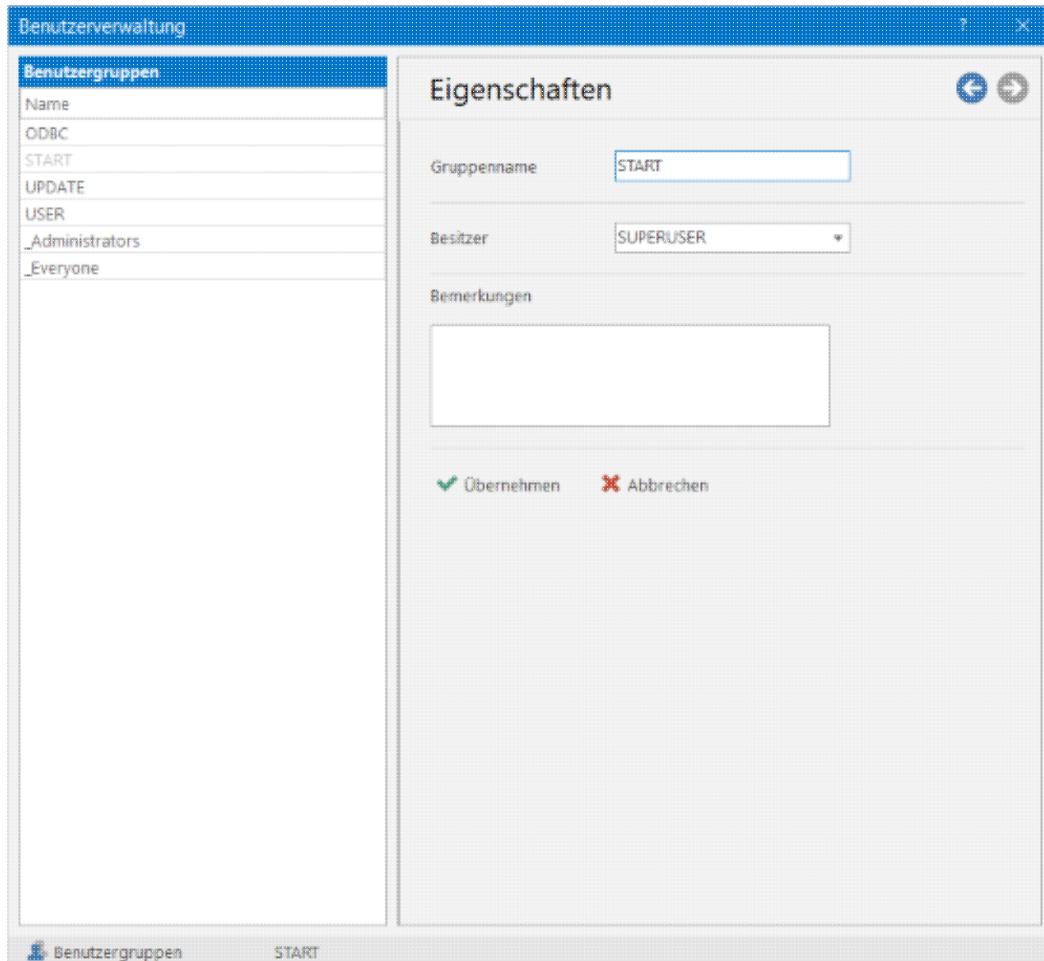
Benutzergruppen - Eigenschaften

Eigenschaften von Benutzergruppen

Über die Schaltfläche [Eigenschaften] werden die Eigenschaften der in der Liste ausgewählten Benutzergruppe angezeigt. Die Eigenschaften, die hier geändert werden können, entsprechen weitestgehend den Eigenschaften beim Hinzufügen einer Benutzergruppe.

Folgende Eigenschaft wird beim Anlegen automatisch vorbelegt und kann hier nachträglich verändert werden:

- Besitzer / _UrmPropOwner Besitzer der Benutzergruppe

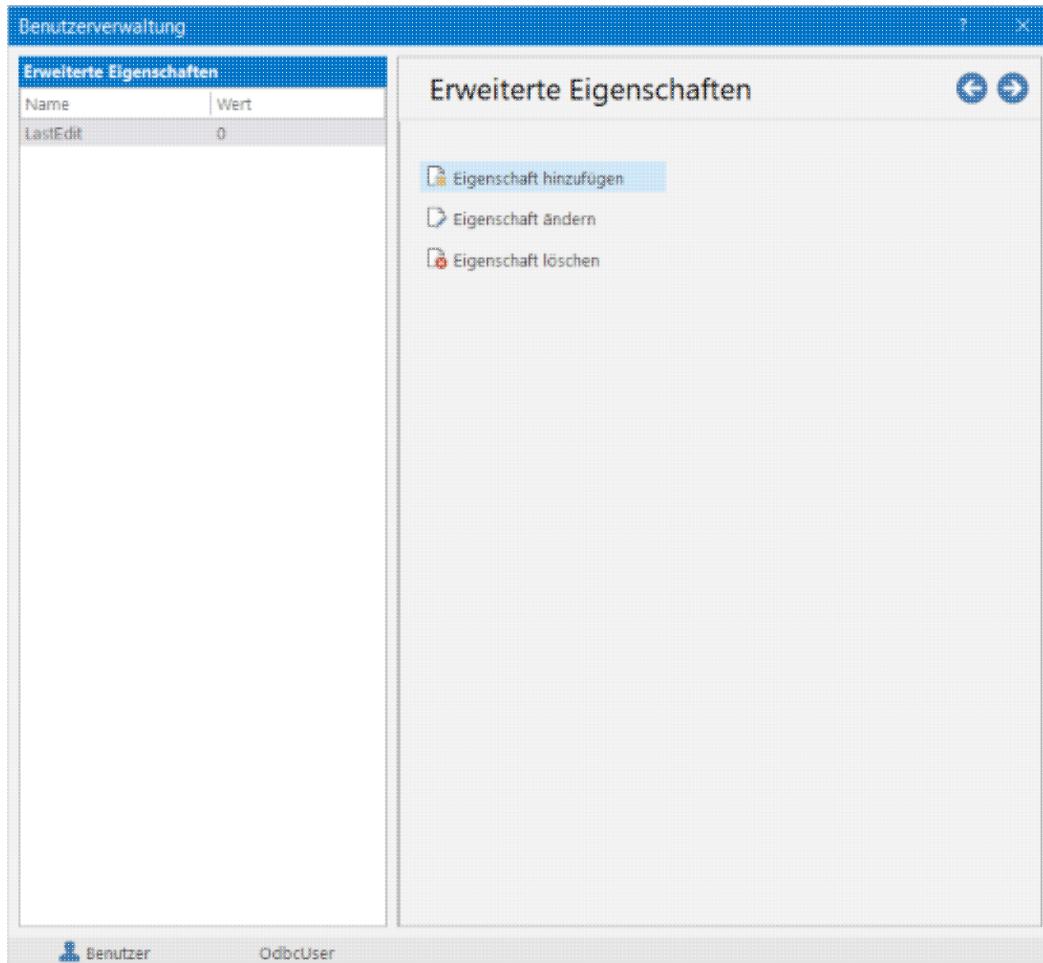


Kontakt

Benutzergruppen - Erweiterte Eigenschaften

Eigene Eigenschaften definieren

Mit der Schaltfläche [Erweiterte Eigenschaften] können vom Benutzer Eigenschaften von Benutzergruppen definiert werden. Es wird folgender Dialog aufgerufen:



Die Funktionen zum Hinzufügen, Ändern und Löschen entsprechen den Funktionen, die bei den Benutzern erläutert sind.

Besondere Eigenschaften

Durch spezielle vordefinierte Eigenschaften ist es möglich, den externen Zugriff auf die Datenbank zu beschränken. Ist die Eigenschaft [UrmOldPropExtAccess](#) der Hauptbenutzergruppe eines Benutzer gesetzt und keine der zusätzlichen Eigenschaften definiert, kann sich der Benutzer mit der jeweiligen Komponente anmelden. Um den Zugriff für einzelne Komponenten zu unterbinden, muss die jeweilige Eigenschaft mit dem Datentyp [TypeLogic](#) angelegt und auf true gesetzt werden. Folgende Eigenschaften werden beachtet:

Eigenschaftsname	Komponente
ExtAccessDeniedAPI	<u>Externe Windows Programmierschnittstelle</u>
ExtAccessDeniedODBC	<u>ODBC-Treiber</u>
ExtAccessDeniedPHP	<u>PHP-Schnittstelle</u>

Kontakt

ExtAccessDeniedPrinter [Druckprozessor](#)

ExtAccessDeniedSOA [SOA-Service](#)

ExtAccessDeniedWeb [Web-Schnittstelle](#)

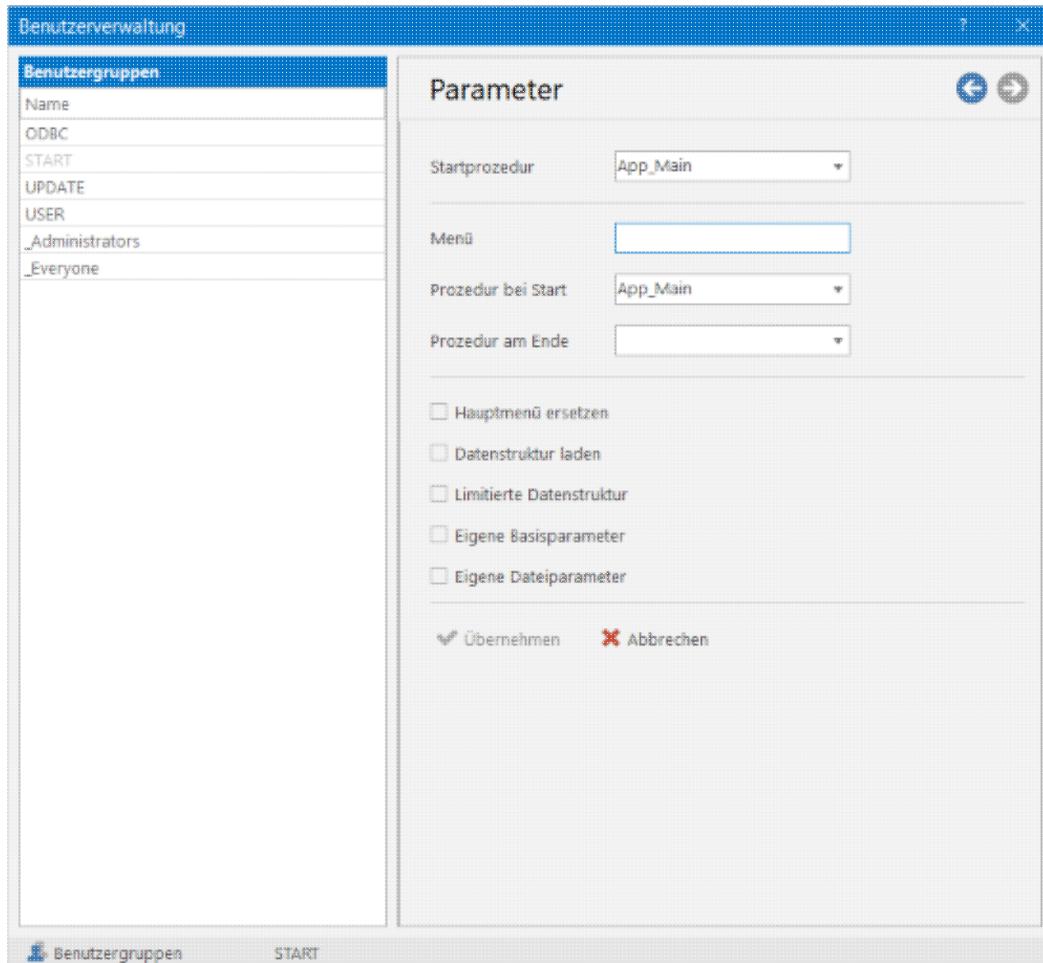
ExtAccessDeniedClient [DbaConnect\(\)](#) aus [Standard-](#) und [Advanced-Client](#)

Kontakt

Benutzergruppen - Parameter

Benutzerparameter des alten Benutzersystems

Durch Drücken der Schaltfläche [Parameter] werden die Benutzerparameter des alten Benutzersystems aufgerufen. Siehe auch [Konvertierung des alten Benutzersystems](#).



Folgende Eigenschaften können gesetzt werden:

- [Startprozedur / _UrmPropStartProc](#)
- [Menü / _UrmOldPropMenu](#)
- [Prozedur bei Start / _UrmOldPropInitProc](#)
- [Prozedur am Ende / _UrmOldPropTermProc](#)
- [Hauptmenü ersetzen / _UrmOldPropOwnMainMenu](#)
- [Datenstruktur laden / _UrmOldPropAutoLoadDataStruct](#)
- [Limitierte Datenstruktur / _UrmOldPropLimitedDataStruct](#)
- [Eigene Basisparameter / _UrmOldPropOwnBaseParams](#)
- [Eigene Dateiparameter / _UrmOldPropOwnFileParams](#)

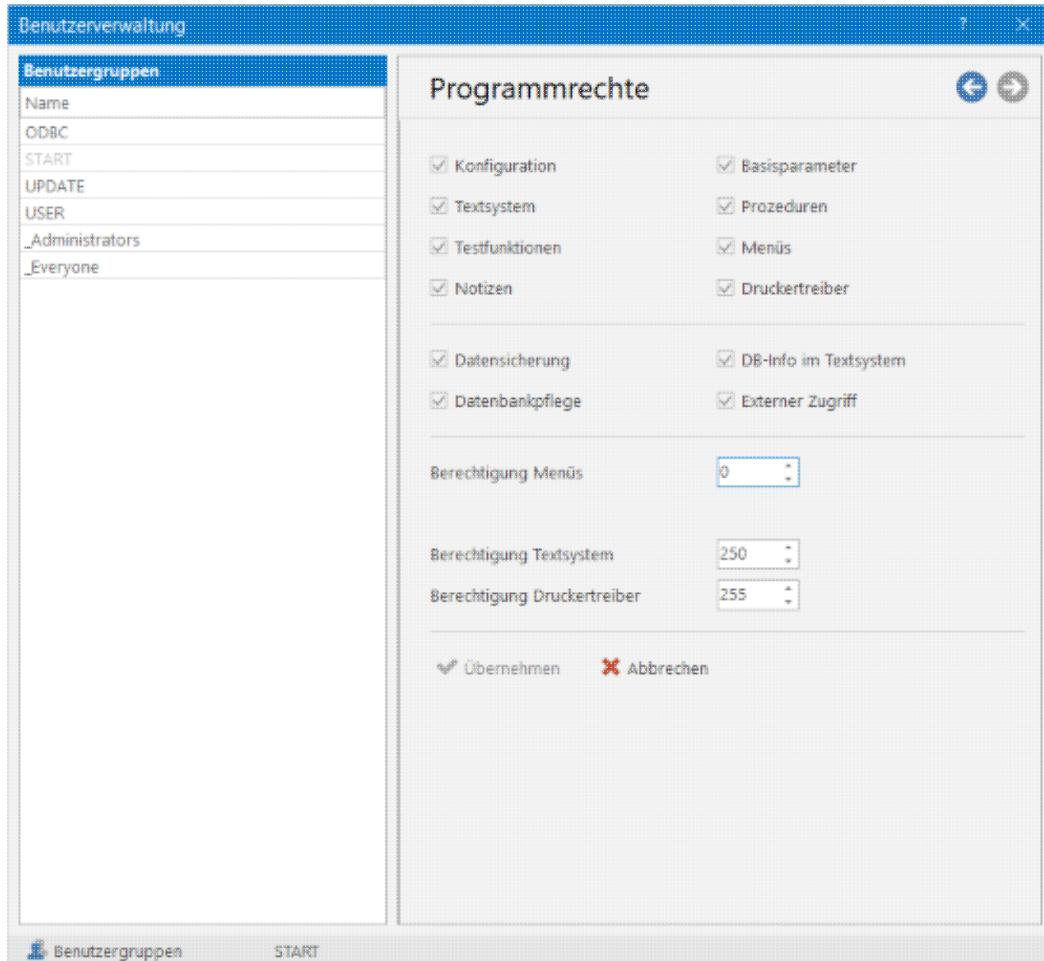
Änderungen an den Rechten werden erst übernommen, wenn die Schaltfläche [Übernehmen] geklickt wird. Zu diesem Zeitpunkt werden auch die Rechte des Benutzers überprüft. Sind die Rechte ausreichend, werden die Änderungen übernommen.

Kontakt

Benutzergruppen - Programmrechte

Programmberechtigungen des alten Benutzersystems

Wird die Schaltfläche [Programmrechte] gedrückt, wird folgender Dialog aufgerufen:



- Konfiguration / _UrmOldPropConfiguration
- Textsystem / _UrmOldPropTextSystem
- Testfunktion / _UrmOldPropTestFunction
- Notizen / _UrmOldPropNotes
- Datensicherung / _UrmOldPropLoadDbDefinition
- Datenbankpflege / _UrmOldPropDbMaintenance
- Basisparameter / _UrmOldPropEditBaseParams
- Prozeduren / _UrmOldPropProcedures
- Menüs / _UrmOldPropEditMenus
- Druckertreiber / _UrmOldPropEditPdv
- DB-Info im Textsystem / _UrmOldPropDbInfo
- Externer Zugriff / _UrmOldPropExtAccess
- Berechtigung Menüs / _UrmOldPropMenuLevel
- Berechtigung Textsystem / _UrmOldPropTextLevel
- Berechtigung Druckertreiber / _UrmOldPropPdvLevel

Die Berechtigungen zur Benutzersteuerung können nicht konvertiert werden und müssen dem neuen Benutzermodell angepasst werden (siehe Definition von

Kontakt

Berechtigungen).

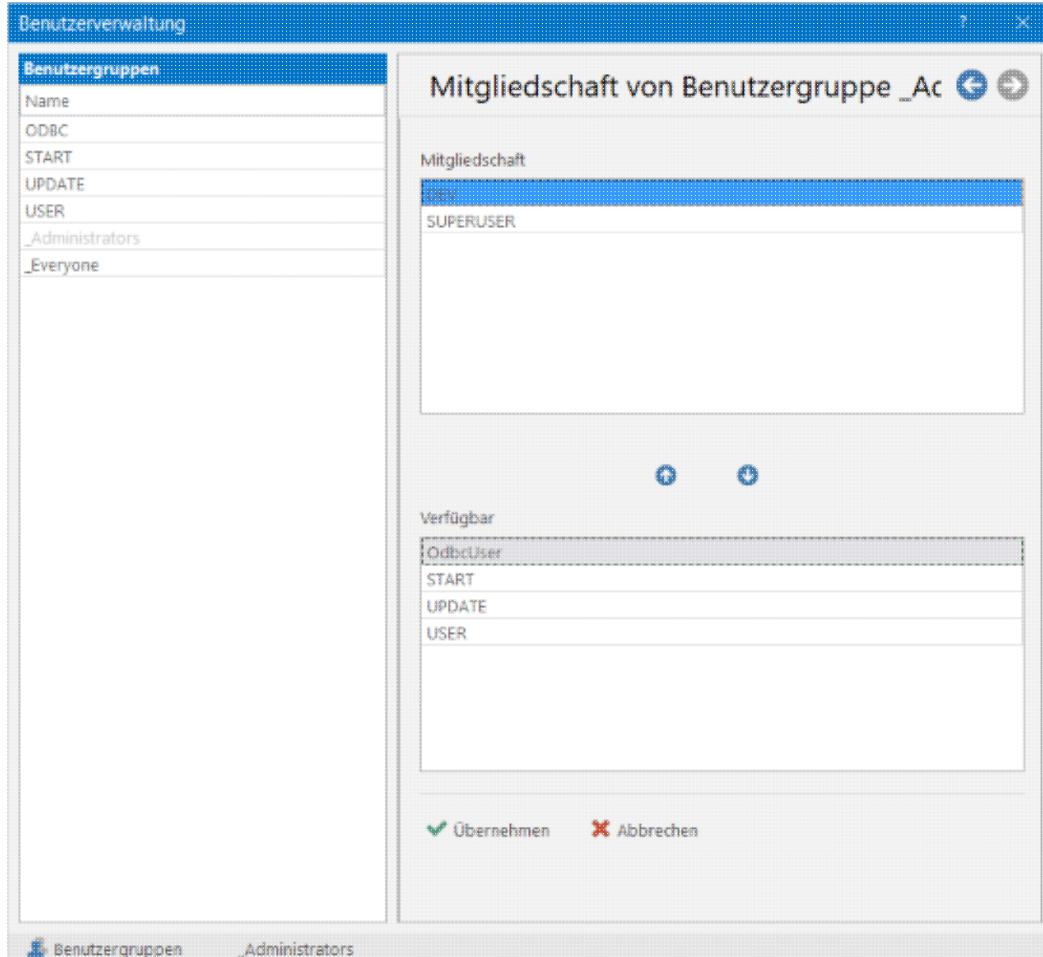
Änderungen an den Rechten werden erst übernommen, wenn die Schaltfläche [Übernehmen] geklickt wird. Zu diesem Zeitpunkt werden auch die Rechte des Benutzers überprüft. Sind die Rechte ausreichend, werden die Änderungen übernommen.

Kontakt

Benutzergruppen - Benutzer

Zuweisung von Benutzern einer Benutzergruppe

Durch Drücken der Schaltfläche [Benutzer] wird folgender Dialog aufgerufen:



Hier können der Benutzergruppe ein oder mehrere Benutzer zugewiesen werden. Innerhalb der Programmierung erfolgt diese Zuweisung über die Anweisung `UrmCreate(tHdlUserGroup, UrmTypeMember, <Name>)`. Die Mitgliedschaft kann mit `UrmDelete()` wieder entfernt werden.

In der oberen Liste sind alle Benutzer aufgeführt, die bereits zu der Benutzergruppe gehören, alle anderen Benutzer sind in der unteren Liste aufgeführt. Ein Benutzer kann der Gruppe hinzugefügt oder aus der Gruppe entfernt werden, indem der Benutzer in der entsprechenden Liste doppelt geklickt wird. Ebenso können über Drag & Drop ein oder mehrere selektierte Benutzer von der einen Liste in die andere Liste verschoben werden. Alternativ dazu kann der Benutzer ausgewählt und mit Hilfe der Schaltflächen zwischen den Listen verschoben werden.

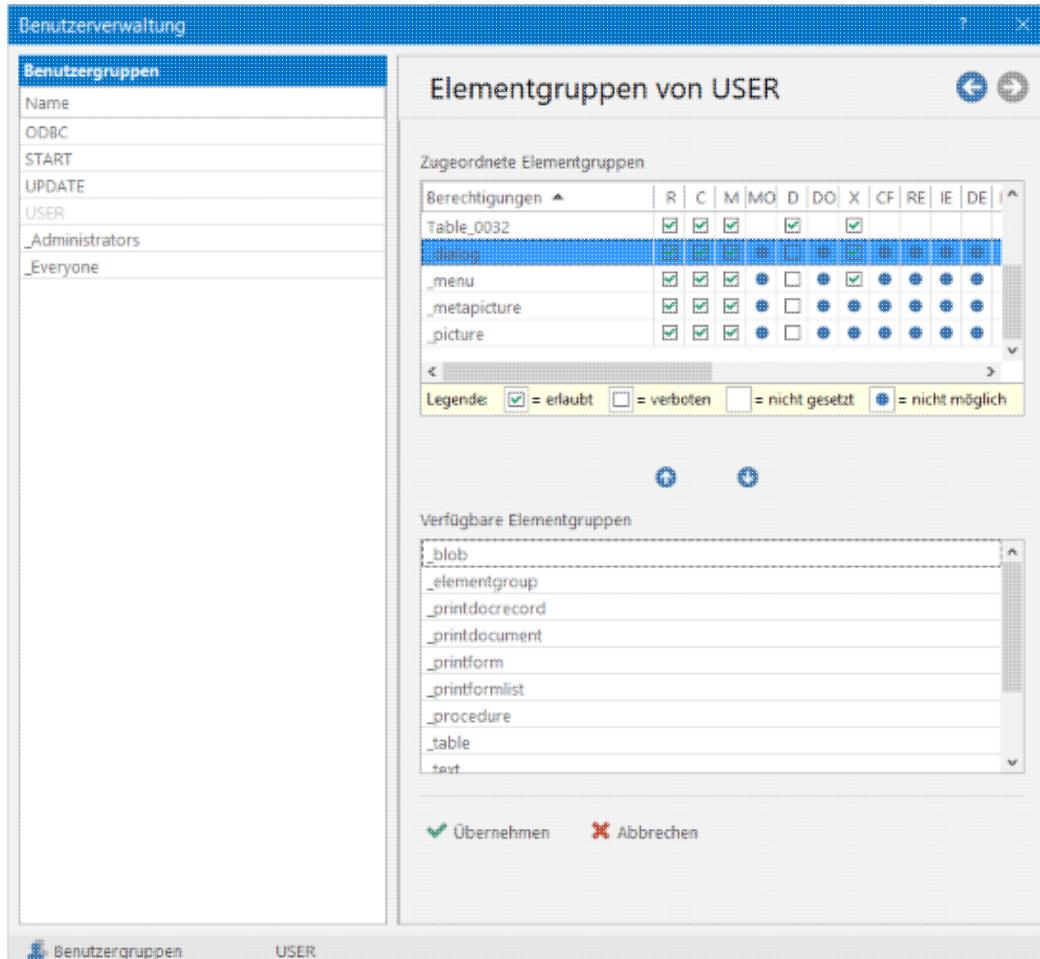
Durch Drücken der Schaltfläche [Übernehmen] werden die Berechtigungen des angemeldeten Benutzers geprüft und bei ausreichenden Rechten die Änderungen wirksam.

Kontakt

Benutzergruppen - Elementgruppen

Zuweisung von Elementgruppen zu Benutzergruppen

Durch Drücken der Schaltfläche [Elementgruppen] wird folgender Dialog aufgerufen:



Hier können der Benutzergruppe eine oder mehrere Elementgruppen zugewiesen und deren Rechte definiert werden. Innerhalb der Programmierung erfolgt die Rechtevergabe über die Anweisungen [UrmPermSet\(tHdlUserGroup, <Name>, UrmAllow, tRights\)](#) und [UrmPermSet\(tHdlUserGroup, <Name>, UrmDeny, tRights\)](#).

In der oberen Liste werden alle Elementgruppen, die bereits zugewiesen sind, aufgeführt. In der unteren Liste werden die noch nicht zugewiesenen Elementgruppen angezeigt. Elementgruppen können durch einen Doppelklick von der einen in die andere Liste verschoben werden. Ebenso können über Drag & Drop ein oder mehrere selektierte Gruppen von der einen Liste in die andere Liste verschoben werden.

In der oberen Liste können die Rechte der Benutzergruppe auf die Elementgruppe angegeben werden. Folgende Rechte sind das im einzelnen:

R - Read	<u>UrmPermRead</u>	Lesen und Anzeigen
C - Create	<u>UrmPermCreate</u>	Anlegen
M - Modify	<u>UrmPermModify</u>	Ändern
	<u>UrmPermModifyOwner</u>	Ändern bei Objektbesitz

Kontakt

MO - Modify owner

D - Delete

DO - Delete owner

X - Execute

CF - Config

[UrmPermDelete](#)

[UrmPermDeleteOwner](#)

[UrmPermExecute](#)

[UrmPermConfig](#)

Löschen

Löschen bei Objektbesitz

Ausführen

Konfiguration

RE - Read elements

[UrmPermElmGroupRead](#)

Berechtigung Elementgruppen lesen

IE - Insert elements

[UrmPermElmGroupInsert](#)

Berechtigung Element zur Gruppe hinzufügen

DE - Delete elements

[UrmPermElmGroupDelete](#)

Berechtigung Element aus Gruppe entfernen

IM - Insert members

[UrmPermMemberInsert](#)

Berechtigung Benutzer zu Benutzergruppe hinzufügen

DM - Delete members

[UrmPermMemberDelete](#)

Berechtigung Benutzer aus Benutzegruppe entfernen

Hier werden ebenfalls die Dateirechte des alten Benzersystems angegeben. Sie sind farblich in gelb von den neuen Rechten abgehoben:

RD - Read (development environment)

[UrmIdePermRead](#) /
[UrmOldPermAccess](#)

Lesen (Entwicklungsumgebung)

CD - Create (development environment)

[UrmIdePermCreate](#) /
[UrmOldPermSave](#)

Erstellen (Entwicklungsumgebung)

MD - Modify (development environment)

[UrmIdePermModify](#) /
[UrmOldPermModify](#)

Ändern (Entwicklungsumgebung)

DD - Delete (development environment)

[UrmIdePermDelete](#) /
[UrmOldPermDelete](#)

Löschen (Entwicklungsumgebung)

AD - Manage (development environment)

[UrmOldPermDelete](#)

Verwalten (Entwicklungsumgebung)

DT - Mix data and text

[UrmOldPermTextMix](#)

Daten und Text mischen

AL - Access links

[UrmOldPermLink](#)

Zugriff auf Verknüpfungen

MS - Modify selections

[UrmOldPermSelections](#)

Ändern von Selektionen

XS - Execute selections

[UrmOldPermExecSelections](#)

Ausführen von Selektionen

MT - Modify transfers

[UrmOldPermTransfers](#)

Ändern von Transfers

Kontakt

XT - Execute transfers	<u>UrmOldPermExecTransfers</u>	Ausführen von Transfers
XL - Execute lists	<u>UrmOldPermExecLists</u>	Ausführen von Listen
MF - Modify list formats	<u>UrmOldPermListFormats</u>	Änderung von Listenformaten
MP - Modify file parameters	<u>UrmOldPermParameters</u>	Änderung von Dateiparametern
MA - Modify access lists	<u>UrmOldPermRecLists</u>	Änderung von Zugriffslisten
Level - File permission level		Berechtigungslevel (Siehe auch <u>UrmPermLevelSet()</u> / <u>UrmPermLevelGet()</u>)

Ein Recht kann durch Doppelklick der entsprechenden Spalte gesetzt und (durch erneuten Doppelklick) entzogen werden. Sollen die Rechte von mehreren

Elementgruppen gesetzt werden, können diese mit + angeklickt werden. Beim Doppelklick mit den gedrückten Tasten wird das Recht für alle markierten Elementgruppen gesetzt. Die Rechte der Benutzergruppe Administrators können nicht verändert werden.



Rechte die in Bezug auf die ausgewählte Gruppe keine Auswirkung haben, werden durch einen schwarzen Punkt gesperrt. Dies ist beispielsweise beim Execute-Recht (X) in der Standard-Elementgruppe printform der Fall.

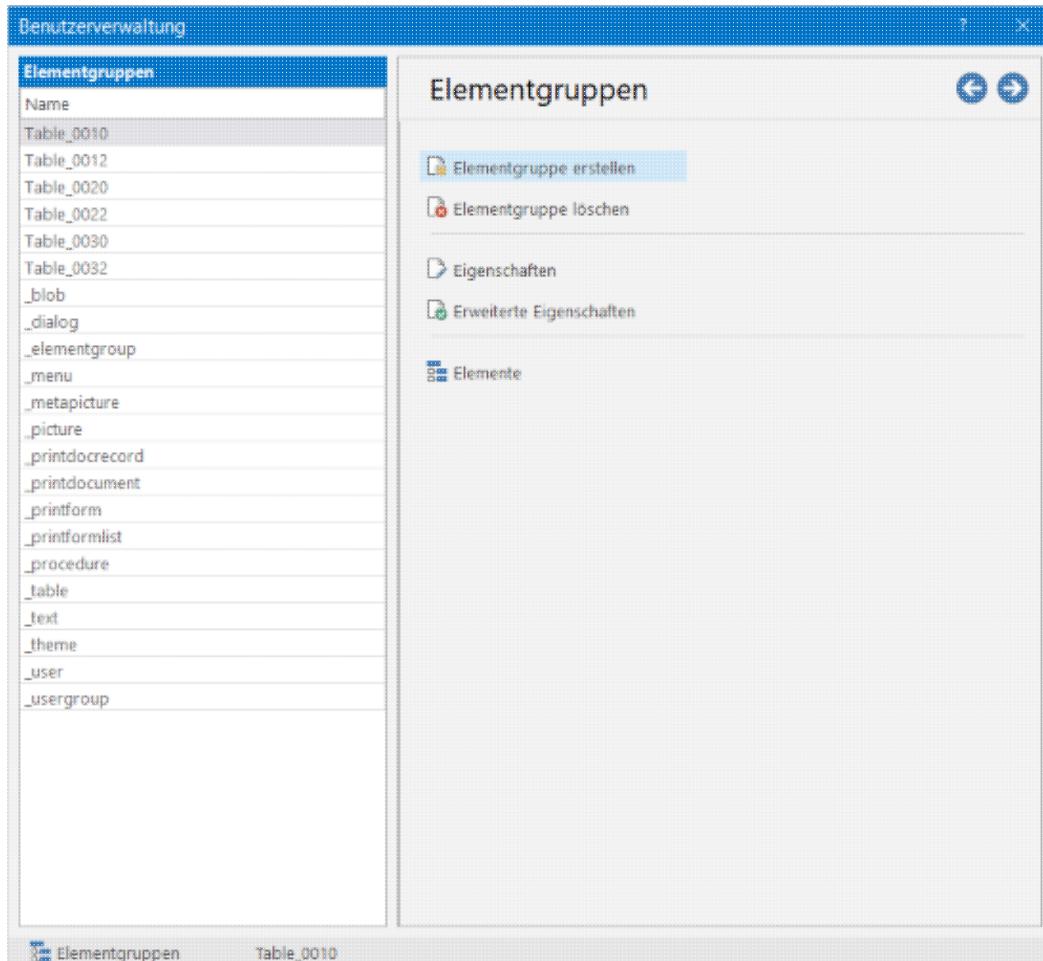
Durch Drücken der Schaltfläche [Übernehmen] werden die Berechtigungen des angemeldeten Benutzers geprüft und bei ausreichenden Rechten die Änderungen wirksam.

Kontakt

Verwalten der Elementgruppen

Anlegen, ändern und löschen von Elementgruppen

Die Verwaltung der Elementgruppen wird durch Drücken der Schaltfläche [Elementgruppen] aufgerufen. Folgender Dialog wird dargestellt:



Folgende Funktionen stehen zur Verarbeitung der Elementgruppen zur Verfügung:

- [Elementgruppe erstellen](#)
- [Elementgruppe löschen](#)
- [Eigenschaften](#)
- [Erweiterte Eigenschaften](#)
- [Elemente](#)

Die Funktionen entsprechen weitgehend den Funktionen der Benutzerverwaltung. In der Liste werden alle definierten Elementgruppen angezeigt. Alle Elementgruppen, deren Name mit einem Unterstrich beginnen, sind vordefinierte Standard-Elementgruppen. Diese Gruppen können nicht geändert oder gelöscht werden.

Standard-Elementgruppen

_blob

[Binäre Objekte in der BLOB-Verwaltung](#)

Kontakt

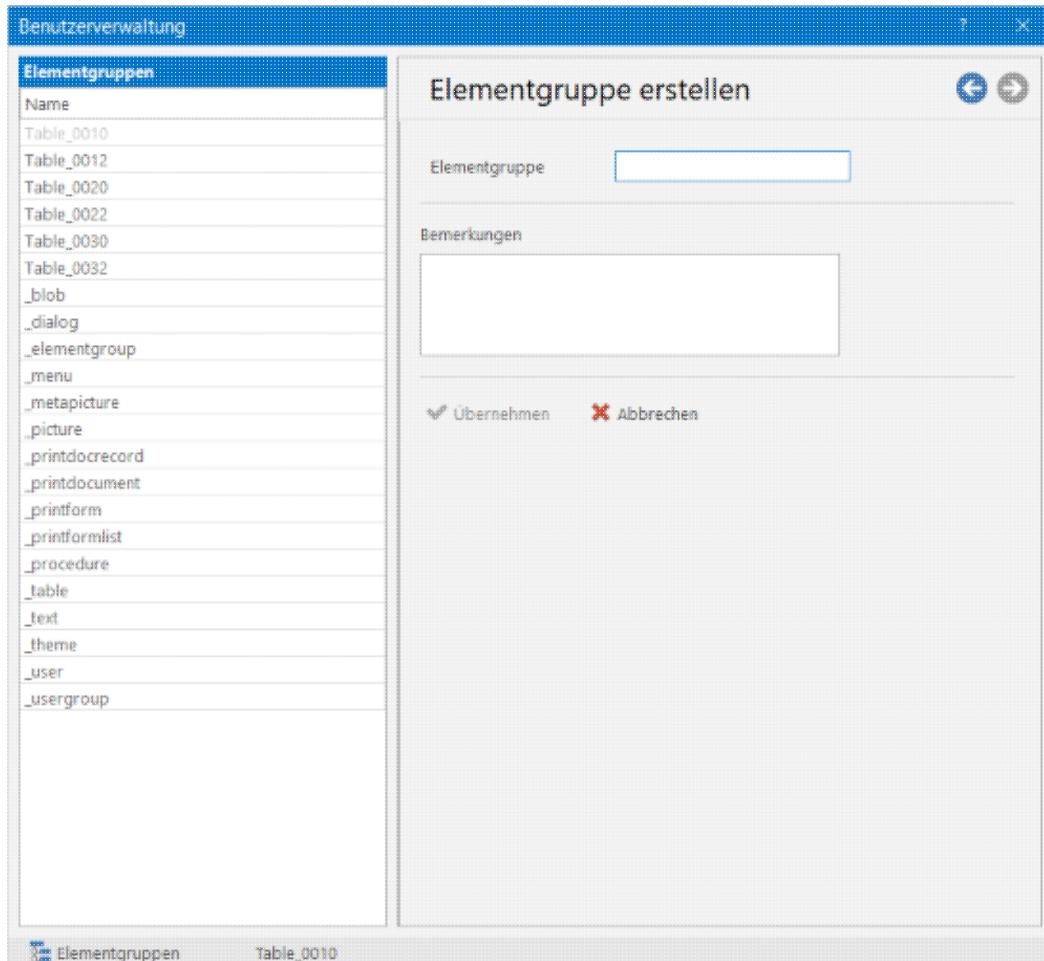
_dialog	alle Dialoge
_elementgroup	alle Elementgruppen
_menu	alle Menüs
_metapicture	alle Bilder im Bereich <u>MetaPicture</u>
_picture	alle Bilder im Bereich <u>Picture</u>
_printdocrecord	alle <u>PrintDocRecord</u> -Objekte
_printdocument	alle <u>PrintDoc</u> -Objekte
_printform	alle <u>PrintForm</u> -Objekte
_printformlist	alle <u>PrintFormList</u> -Objekte
_procedure	alle Prozeduren
_table	alle Dateien
_text	alle Texte
_theme	alle <u>Themes</u>
_user	alle Benutzer
_usergroup	alle Benutzergruppen

Kontakt

Elementgruppen - Elementgruppe erstellen

Neue Elementgruppe erstellen

Durch Drücken der Schaltfläche [Elementgruppe erstellen] wird folgender Dialog aufgerufen:



Folgende Eigenschaften der Elementgruppe können gesetzt werden:

- Elementgruppenname / _UrmPropName Name der Elementgruppe
- Bemerkung / _UrmPropComment Bemerkung zur Elementgruppe

Der Name der Elementgruppe muss angegeben werden. Die Angabe einer Bemerkung ist optional. Für die Eingaben gelten die gleichen Beschränkungen, wie für die entsprechenden Eigenschaften.

Die Elementgruppe wird erst angelegt, wenn die Schaltfläche [Hinzufügen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird die Gruppe angelegt. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

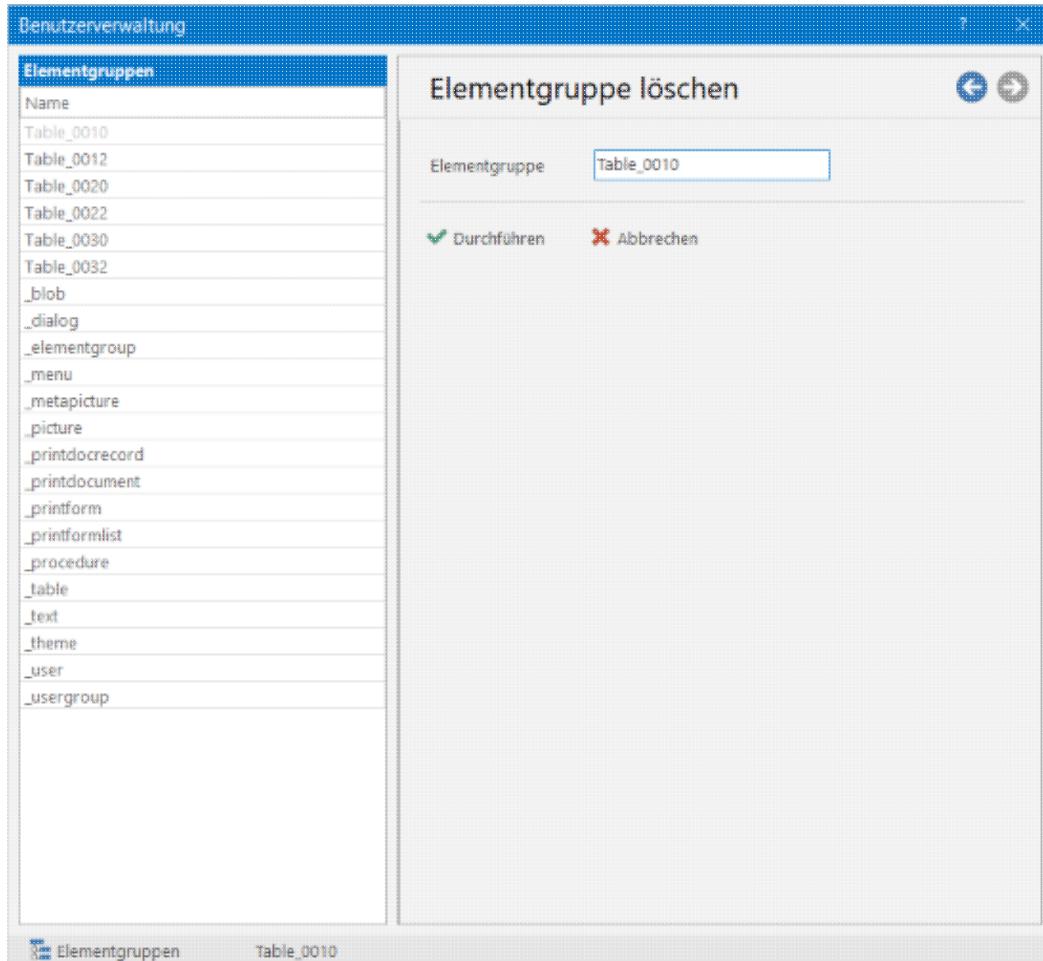
Diese Funktion entspricht der Anweisung UrmCreate(0, _UrmTypeElmGroup, <Name>) wobei anschließend die Elementgruppe mit UrmOpen() geöffnet und die entsprechenden Eigenschaften gesetzt werden.

Kontakt

Elementgruppen - Elementgruppe löschen

Elementgruppe löschen

Mit dieser Schaltfläche wird die links ausgewählte Elementgruppe gelöscht.



Die Elementgruppe wird erst gelöscht, wenn die Schaltfläche [Durchführen] geklickt wird. Die Überprüfung der Berechtigungen findet ebenfalls erst zu diesem Augenblick statt. Sind ausreichende Rechte vorhanden, wird die Gruppe gelöscht. Der Vorgang kann jederzeit durch Klicken von [Abbrechen] abgebrochen werden.

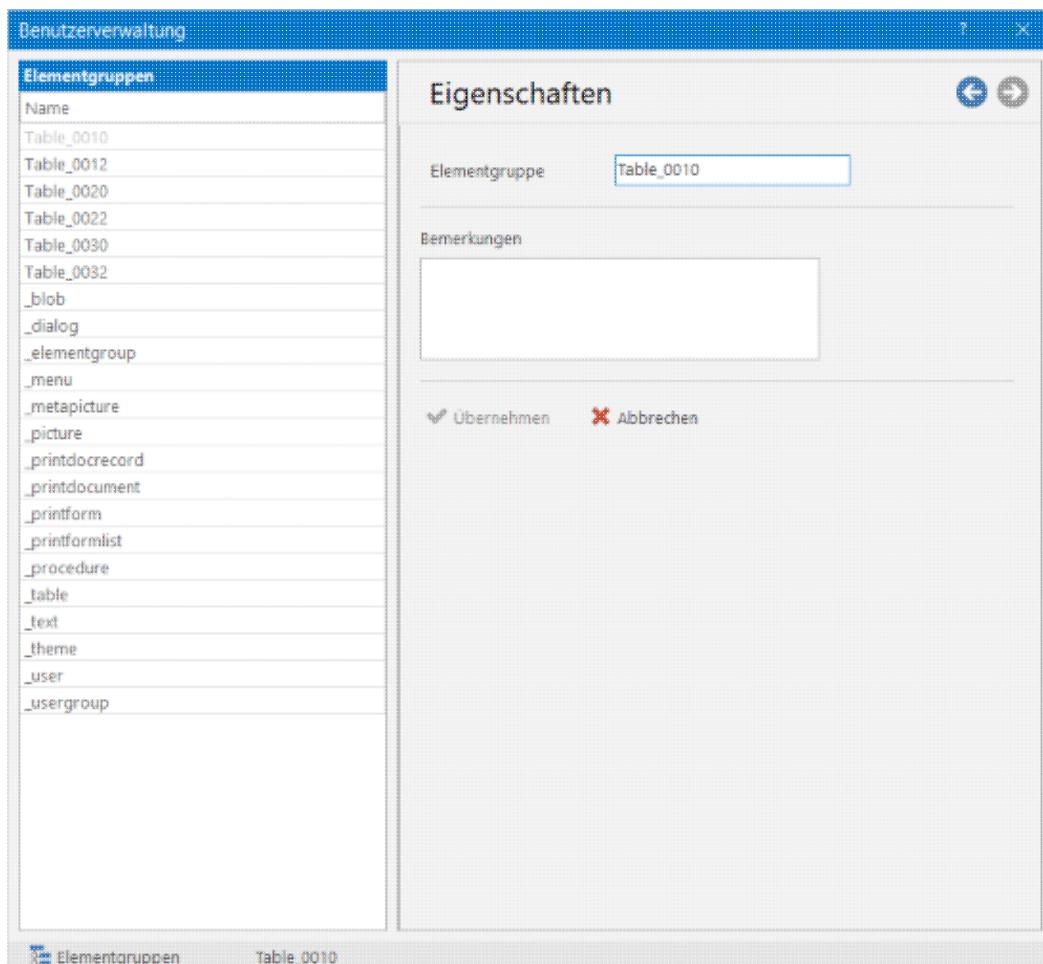
Die Funktion entspricht dem Befehl `UrmDelete(0, _UrmTypeElmGroup, <Name>)`. Die Mitgliederlisten der Benutzer werden automatisch aktualisiert. Die Standard-Elementgruppen können nicht gelöscht werden.

Kontakt

Elementgruppen - Eigenschaften

Eigenschaften einer Elementgruppe ändern

Über die Schaltfläche [Eigenschaften] werden die Eigenschaften der in der Liste ausgewählten Elementgruppe angezeigt. Die Eigenschaften, die hier geändert werden können, entsprechen den Eigenschaften beim Erstellen einer Elementgruppe.

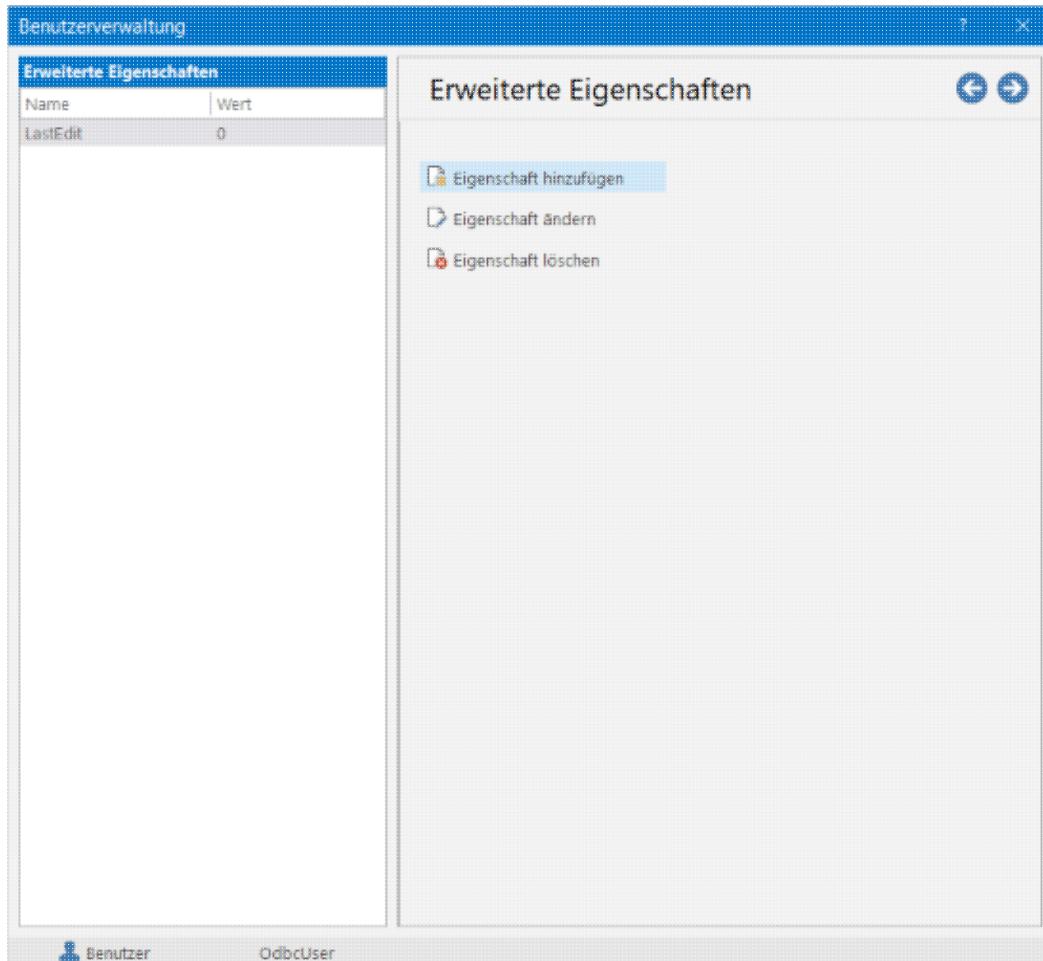


Kontakt

Elementgruppen - Erweiterte Eigenschaften

Eigene Eigenschaften definieren

Mit der Schaltfläche [Erweiterte Eigenschaften] können vom Benutzer Eigenschaften von Elementgruppen definiert werden. Es wird folgender Dialog aufgerufen:



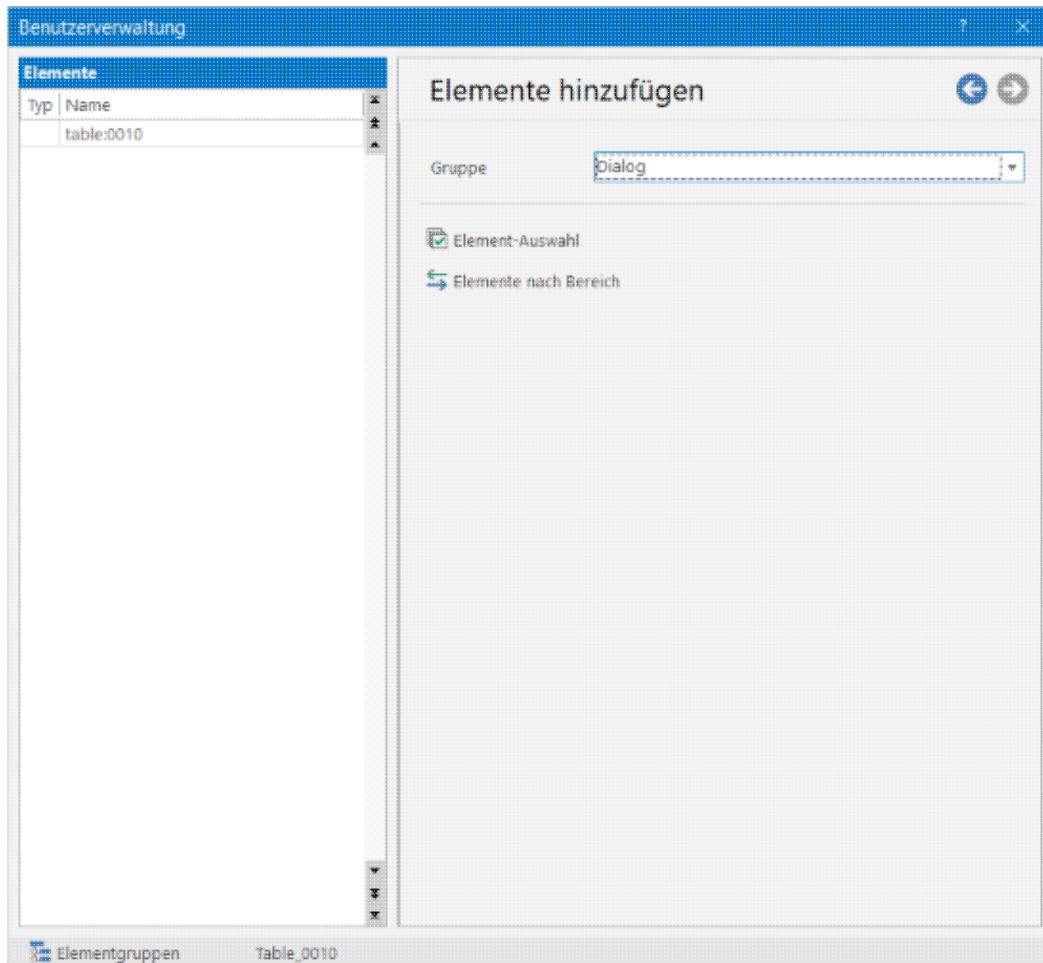
Die Funktionen zum Hinzufügen, Ändern und Löschen entsprechen den Funktionen, die bei den Benutzern erläutert sind.

Elementgruppen - Elemente

Elemente einer Elementgruppe ändern

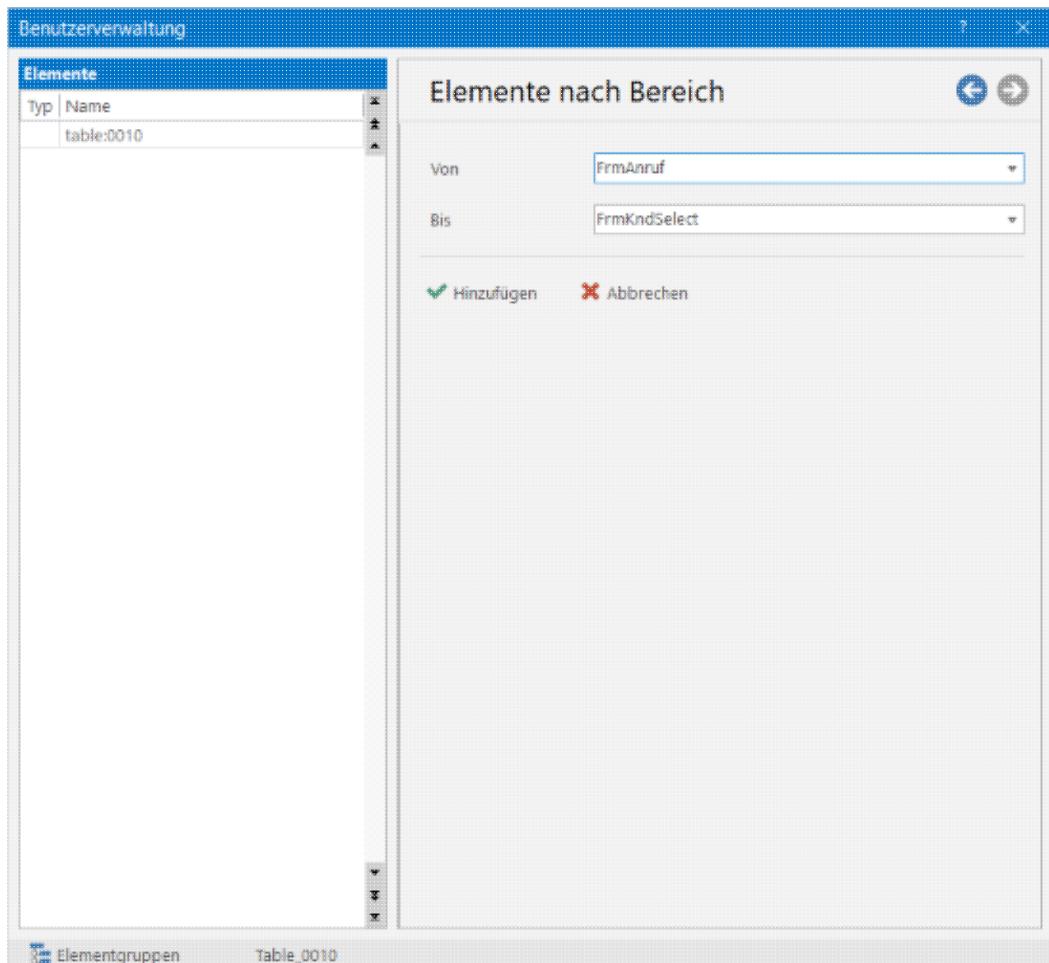
Nach dem Drücken der Schaltfläche [Elemente] werden in der Liste die Elemente der zuvor selektierten Elementgruppe angezeigt. Standardmäßig werden Text-Elemente zunächst ausgeblendet. Sollen die in der Elementgruppe enthaltenen Texte angezeigt werden, muss die Option "Text-Elemente anzeigen" aktiviert werden. Abhängig von der Anzahl der in der Datenbank vorhandenen Texte, kann der Vorgang einige Zeit in Anspruch nehmen. Der Elementgruppe können neue Elemente hinzugefügt oder bestehende Elemente entfernt werden.

Soll ein neues Element hinzugefügt werden, muss zunächst die Gruppe ausgewählt werden (Tabellen, Dialoge, Menüs ...), aus dem das Element stammt. Anschließend kann über die Schaltfläche [Element Auswahl] oder über [Elemente nach Bereich] ein oder mehrere Elemente in die Elementgruppe aufgenommen werden. Die Elemente einer Elementgruppe können aus unterschiedlichen Gruppen stammen, so kann eine Elementgruppe aus Dialogen, Menüs, Prozeduren, Druckformularen usw. bestehen.



Soll ein Bereich von Elementen angegeben werden, muss nur das erste und letzte Element angegeben werden. Bei der Element-Auswahl kann auch eine Mehrfachselektion in der Liste der Elemente erfolgen.

Kontakt



In der Programmierung wird zur Zuweisung von Elementen zu einer Elementgruppe die Anweisung `UrmCreate(tHdlElmGroup, _UrmTypeElm..., <Entryname>)` verwendet.

Durch Drücken der Schaltfläche [Durchführen] werden die Berechtigungen des angemeldeten Benutzers geprüft und bei ausreichenden Rechten die Änderungen wirksam.

Kontakt

Objekte des Benutzersystems
Benutzer-, Benutzergruppe- und Elementgruppe

- User
- UserGroup
- ElmGroup

User

 Benutzer-Objekt der Benutzerverwaltung

[Liste](#),

Siehe [Eigenschaften](#),

[Benutzerbefehle](#)

Dieses Objekt entspricht einem Benutzer der Datenbank. Mit diesem Benutzer ist eine Anmeldung an der Datenbank möglich, vorausgesetzt das Kennwort ist bekannt und andere Eigenschaften ([_UrmPropActive](#) und [_UrmPropExpirationDate](#)) verhindern die Anmeldung nicht.

Das Objekt User ist nicht zu verwechseln mit dem angemeldeten Benutzer. In einer leeren Datenbank sind zwei Benutzer vorhanden SUPERUSER und USER. Der Benutzer USER kann über die verschiedenen Befehle der Benutzerpflege manipuliert werden. Der Benutzer SUPERUSER kann seine Eigenschaften nur selbst verändern, andere Benutzer haben dazu keine Berechtigung. Der SUPERUSER kann die Eigenschaften [_UrmPropActive](#) und [_UrmPropExpirationDate](#) sowie seine Zugehörigkeit zu Benutzergruppen nicht verändern. Alle anderen Benutzer können das eigene Kennwort (sofern nicht durch [_UrmPropPwdLocked](#) unterbunden) und alle eigenen benutzerdefinierten Eigenschaften ändern. Zum Ändern ihrer eigenen Systemeigenschaften werden entsprechende [_UrmPermModify](#)-Rechte benötigt.

Ein Benutzer kann einer oder mehreren [Benutzergruppen](#) zugeordnet werden (siehe [_UrmCreate\(\)](#)). Er bekommt dadurch die dort definierten Rechte an einer oder mehreren [Elementgruppen](#). Details zur Rechtevergabe befinden sich im Abschnitt [Definition von Berechtigungen](#).

Ein Benutzer wird mit der Anweisung [_UrmCreate\(0, _UrmTypeUser, ...\)](#) erzeugt. Ein vorhandener Benutzer kann mit der Anweisung [_UrmOpen\(0, _UrmTypeUser, ...\)](#) geöffnet werden. Nach dem Öffnen des Benutzers steht ein Deskriptor zur Verfügung, mit dem auf die [Eigenschaften](#) des Benutzers zugegriffen werden kann. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch [Verwaltung der Benutzer](#)).

Beispiel

```
// Aktiviert den Benutzer Sales_Temp für drei Monate
HdlUser # UrmOpen(_UrmTypeUser, _UrmLock, 'S...
```

UserGroup

 Objekt der Benutzergruppe in der Benutzerverwaltung

[Liste](#),

Siehe [Eigenschaften](#),

[Benutzerbefehle](#)

Dieses Objekt entspricht einer Benutzergruppe in der Datenbank. Die Benutzergruppe verfügt über eine Liste von Benutzern ([User](#)-Objekte) die Mitglieder dieser Gruppe sind. In einer Benutzergruppe können auch Rechte für [Elementgruppen](#) definiert werden. Die effektiven Rechte eines Benutzer an einer Elementgruppe entstehen aus der Kombination aller Rechte der Benutzergruppen, in denen Benutzer Mitglied ist. Details zur Rechtevergabe befinden sich im Abschnitt [Definition von Berechtigungen](#).

In einer leeren Datenbank sind bereits die beiden Benutzergruppen `_Administrators` und `_Everyone` definiert.

Standardmäßig sind in der Benutzergruppe `_Everyone` alle vordefinierten Elementgruppen mit Ausnahme von `_user` und `_usergroup` mit den Rechten zum Erzeugen, Lesen, Ändern, Löschen, Ausführen und Konfigurieren eingetragen. Soll einer anderen Benutzergruppe die Ausführung aller Prozeduren erlaubt werden, muss der Benutzergruppe die Elementgruppe `_procedure` mit dem Recht [`UrmPermExecute`](#) zugewiesen werden.

Eine Benutzergruppe wird mit der Anweisung [`UrmCreate\(0, _UrmTypeUserGroup, ...\)`](#) erzeugt. Eine vorhandene Benutzergruppe kann mit der Anweisung [`UrmOpen\(0, _UrmTypeUser, ...\)`](#) geöffnet werden. Beim Öffnen der Benutzergruppe wird ein Deskriptor zurückgegeben, mit dem auf die [Eigenschaften](#) der Benutzergruppe zugegriffen werden kann. Die Rechte auf Elementgruppen können dann mit dem Befehl [`UrmPermSet\(\)`](#) festgelegt werden. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch [Verwaltung der Benutzergruppen](#)).

Die Eigenschaften der Benutzergruppe `_Administrators` können mit Ausnahme der Eigenschaft [`UrmPropComment`](#) nicht verändert werden. Insbesondere können die Berechtigungen der Gruppe nicht geändert werden.

Beispiel

```
// Benutzergruppe erzeugenErg # UrmCreate(0, _UrmTypeUserGroup, 'UserGrpLogin');if (tErg = _Err0)
```

ElmGroup

 Elementgruppe der Benutzerverwaltung

[Liste](#),

Siehe [Eigenschaften](#),

[Benutzerbefehle](#)

Dieses Objekt beinhaltet eine Reihe von Einträgen. Jeder Eintrag repräsentiert entweder ein reales Element aus der Datenbank (ein Dialog, eine Prozedur, eine Datensatz-Tabelle usw.) oder ein durch den Programmierer definiertes virtuelles Objekt. Eine Elementgruppe fasst somit ein oder mehrere Datenbank- und/oder virtuelle Objekte zu einer Gruppe zusammen. Für die Gruppe können dann Rechte bei den [Benutzergruppen](#) definiert werden. Durch die Gruppenbildung erhält der Programmierer eine bessere Übersicht über das Rechtesystem.

Um einen Eintrag in eine Elementgruppe zu schreiben wird die Anweisung [UrmCreate\(\)](#) verwendet. Der Deskriptor der Elementgruppe und eine der [UrmTypeElm...-Konstanten](#) müssen zusammen mit dem Namen des Elements übergeben werden. Folgende Typen können angegeben werden:

UrmTypeElmCustom	Benutzerdefiniert / virtuelles Objekt
UrmTypeElmUser	Benutzer
UrmTypeElmUsergroup	Benutzergruppe
UrmTypeElmTable	Datei
UrmTypeElmDialog	Dialog
UrmTypeElmMenu	Menü
UrmTypeElmPicture	Bild im Bereich Picture
UrmTypeElmMetapicture	Bild im Bereich MetaPicture
UrmTypeElmPrintform	PrintForm -Objekt
UrmTypeElmPrintformlist	PrintFormList -Objekt
UrmTypeElmPrintdocument	PrintDoc -Objekt
UrmTypeElmPrintdocrecord	PrintDocRecord -Objekt
UrmTypeElmProcedure	Prozedur
UrmTypeElmText	interner Text
UrmTypeElmBlob	Binäre Objekte in der BLOB-Verwaltung
UrmTypeElmTheme	Theme -Objekt

Um zum Beispiel einen Dialog in eine Elementgruppe aufzunehmen, muss ein Eintrag für den Typ (in diesem Fall [UrmTypeElmDialog](#)) in die Elementgruppe vorgenommen werden. Als (alpha2) wird der Name des Dialogs 'Beispiel1' übergeben.

Über den Typ [UrmTypeElmCustom](#) können Einträge durch den Programmierer vorgenommen werden: 'ModulArticle'. Die Rechte auf ein Element können in der Programmierung mit den Anweisungen [UrmPermElementGet\(\)](#) oder [UrmPermElementGetRaw\(\)](#) ermittelt werden.

Eine Elementgruppe wird mit der Anweisung [UrmCreate\(0, UrmTypeElmGroup, ...\)](#) erzeugt. Eine vorhandene Elementgruppe kann mit der Anweisung [UrmOpen\(0, UrmTypeElmGroup, ...\)](#) geöffnet werden. Beim Öffnen wird ein Deskriptor zurückgegeben, mit dem auf die [Eigenschaften](#) der Elementgruppe zugegriffen

Kontakt

werden kann. Ebenso können weitere Objekte der Elementgruppe hinzugefügt oder aus der Gruppe entfernt werden. Die Möglichkeiten stehen zur Laufzeit und zur Entwurfszeit zur Verfügung (siehe auch Verwaltung der Elementgruppen).

Beispiel

```
// Elementgruppe erzeugen tErg # UrmCreate(0, _UrmTypeElmGroup, 'ElmGrpLoginExamples');if (tErg =
```

Kontakt

Definition von Berechtigungen

Benutzergruppen Rechte geben und entziehen

Ein Recht kann zugesichert ([UrmAllow](#)) oder entzogen ([UrmDeny](#)) werden. Ein entzogenes Recht hat dabei Vorrang vor einem zugesicherten Recht.

Folgende Rechte können vergeben und entzogen werden:

Read (R)	<u>UrmPermRead</u>	Lesen und Anzeigen
Create (C)	<u>UrmPermCreate</u>	Anlegen
Modify (M)	<u>UrmPermModify</u>	Ändern
Modify Owner (MO)	<u>UrmPermModifyOwner</u>	Ändern bei Objektbesitz
Delete (D)	<u>UrmPermDelete</u>	Löschen
Delete Owner (DO)	<u>UrmPermDeleteOwner</u>	Löschen bei Objektbesitz
Execute (X)	<u>UrmPermExecute</u>	Ausführen
Config (CF)	<u>UrmPermConfig</u>	Konfiguration

Für die Verarbeitung von Benutzerrechten werden weitere Rechte benötigt:

Group read (RE)	<u>UrmPermElmGroupRead</u>	Berechtigung Elementgruppen lesen
Element insert (IE)	<u>UrmPermElmGroupInsert</u>	Berechtigung Element zur Gruppe hinzufügen
Element delete (DE)	<u>UrmPermElmGroupDelete</u>	Berechtigung Element aus Gruppe löschen

Für das Zuordnen von Benutzern zu Benutzergruppen gibt es die folgenden Berechtigungen:

Insert Member (IM)	<u>UrmPermMemberInsert</u>	Berechtigung Benutzer zu Benutzergruppe hinzufügen
Delete Member (DM)	<u>UrmPermMemberDelete</u>	Berechtigung Benutzer aus Benutzergruppe entfernen

Neben diesen vom System vorgegebenen Berechtigungen können vom Programmierer noch 24 weitere Rechte vergeben werden. Die Rechte können über [UrmPermUser](#)-Konstanten angesprochen werden.

Neue Berechtigungen können entweder im Designer über den Menüpunkt **Bearbeiten / Benutzeroberfläche...** oder über die entsprechenden [Benutzerbefehle](#) angelegt bzw. geändert werden.

Prinzipiell gibt es zwei Vorgehensweisen, wenn im Rahmen eines Rechtesystems innerhalb der Datenbank gearbeitet wird:

1. Aufruf der Funktion und Überprüfung des Rückgabewertes
2. Überprüfung der Rechte und Aufruf der Funktion

Werden keine benutzerdefinierten Rechte und Elemente verwendet, kann die erste Methode verwendet werden. Über den Rückgabewert der aufgerufenen Funktion kann dann festgestellt werden, ob die Funktion ausgeführt werden konnte (siehe [ErrGet\(\)](#) und bei den entsprechenden Anweisungen).

Kontakt

Soll vor dem Aufruf einer Funktion überprüft werden, ob entsprechende Rechte vorhanden sind, kann dies mit den Anweisungen [UrmPermGet\(\)](#) und [UrmPermGetRaw\(\)](#) bzw. [UrmPermElementGet\(\)](#) und [UrmPermElementGetRaw\(\)](#) erfolgen. Die Anweisungen unterscheiden sich einerseits dadurch, dass sie die Rechte auf Ebene der Elementgruppen bzw. auf Ebene der Elemente überprüfen, andererseits eine Rechteauswahl prüfen oder alle Rechte zurück geben.

Welche Anweisung verwendet wird, um die Rechte zu überprüfen hängt letztlich von der Situation ab. Soll nur für einen Moment überprüft werden, ob ein Recht vorhanden ist, können die Anweisungen [UrmPermGet\(\)](#) und [UrmPermElementGet\(\)](#) leicht in eine [if](#)-Abfrage integriert werden. Sollen im weiteren Verlauf noch andere Rechte des gleichen Objekts überprüft werden, können mit den anderen Anweisungen gleich alle Rechte abgefragt und zwischengespeichert werden.

```
if (UrmPermGet(0, 'ElmGrpCustomer', _UrmAllow, _UrmPermRead)){ tErg # RecRead(tblCstCustomer, ke
```

Das bisherige Beispiel erfordert die Kenntnis, welche Elementgruppe abgefragt werden muss, um die korrekte Berechtigung zu lesen. Ist die Elementgruppe nicht bekannt, können die UrmPermElement...-Befehle verwendet werden. Hier wird das Datenbank-Element angegeben, um die Berechtigung zu ermitteln.

```
if (UrmPermElementGet(_UrmTypeElmTable, 'tblCstCustomer', _UrmPermRead)){ tErg # RecRead(tblCstCu
```

In diesem Fall wird das Leserecht in einer Datei überprüft. Die Vergabe der Rechte ist aber nicht an Elemente der Datenbank gebunden. Der Programmierer kann eigene Einträge definieren. So kann zum Beispiel die Ausführung von Druck-Funktionen durch Rechte beschränkt werden. Dazu wird mit [UrmCreate\(tHdlElmGrp, UrmTypeElmCustom, 'ElmPrintOutput'\)](#) einer geöffneten Elementgruppe der Eintrag ElmPrintOutput hinzugefügt. Mit der Anweisung [UrmPermSet\(tHdlUserGrp, <Name Elmementgruppe>, _UrmRead | _UrmExecute\)](#) wird einer Benutzergruppe das Lese- und das Ausführungsrecht zugesprochen. Mit der Abfrage [UrmPermElementGet\(_UrmTypeElmCustom, 'ElmPrintOutput', _UrmPermExecute\)](#) kann für den eigenen Benutzer das Ausführungsrecht überprüft werden.

Mit diesen Befehlen können nur die effektiven Rechte des eigenen Benutzers ermittelt werden. Die Rechte von anderen Benutzern können nur mit den Anweisungen [UrmPermGet\(\)](#) und [UrmPermGetRaw\(\)](#) ermittelt werden.

Die neuen Berechtigungen werden vom CONZEPT 16-Server verwaltet. Änderungen an diesen Rechten wirken sich sofort aus. Die Berechtigungen aus dem alten Benutzersystem werden durch den Client verarbeitet. Diese Rechte werden nur bei der Anmeldung des Benutzers vom Server gelesen. Änderungen wirken sich somit erst nach einer erneuten Anmeldung aus. Alle Änderungen an den Rechten und Eigenschaften von Benutzern, Benutzergruppen und Elementgruppen bleiben persistent in der Datenbank erhalten.

Kontakt

Limitationen des Benutzersystems

Maximale Anzahl von Benutzern, Benutzergruppen usw.

Die Anzahl der verschiedenen Objekte des Benutzersystems sind beschränkt:

Maximale Anzahl der <u>User</u> -Objekte	8000
Maximale Anzahl der <u>UserGroup</u> -Objekte	8000
Maximale Anzahl der <u>ElmGroup</u> -Objekte	32000
Maximale Anzahl der Einträge in einer <u>ElmGroup</u>	unbeschränkt
Maximale Anzahl der Mitglieder einer <u>UserGroup</u>	1000
Maximale Anzahl der Mitgliedschaften eines <u>User</u> -Objekts	1000
Maximale Anzahl der Rechte auf <u>Elementgruppen</u> in einer <u>UserGroup</u>	8000
Maximale Anzahl von zusätzlichen Eigenschaften	1000

Kontakt

Konvertierung des Benutzersystems

Verwendung des alten Benutzersystems in Versionen ab 5.3.00

Bei der Umstellung einer Datenbank von den Versionen 4.7 bis 5.2 auf die Version 5.3 oder der Versionen 3.5 bis 3.7 auf Version 3.8, wird das bis dahin gültige Benutzersystem in das neue Benutzersystem konvertiert. Der Zugriff auf die Benutzer ist anschließend nur noch über das neue Benutzersystem aus dem Designer aufrufbar. Die Benutzerverwaltung wird über das Menü **Bearbeiten / Benutzerpflege...** aufgerufen.

Dieser Abschnitt geht davon aus, dass der Aufbau des neuen Benutzersystems bekannt ist. Das neue Benutzersystem ist im Abschnitt Benutzerpflege erläutert.

Bei der Konvertierung wird für jeden Benutzer der Datenbank ein User-Objekt angelegt. Ist dieser Benutzer keinem Hauptbenutzer untergeordnet, wird zusätzlich noch ein UserGroup-Objekt erzeugt. Die Namen des Benutzer- und des Benutzergruppen-Objekts entsprechen dem Namen des Benutzers. Der Benutzer wird automatisch seiner Benutzergruppe (entweder der Benutzergruppe mit dem gleichen Namen oder seinem Hauptbenutzer - jetzt eine Benutzergruppe) zugeordnet. Bei den Eigenschaften des Benutzers verbleiben nur noch das Kennwort, der Kommentar und die Benutzergruppe. Alle anderen Eigenschaften können bei der Benutzergruppe abgefragt werden. Das betrifft insbesondere die Rechte des Benutzers. Die Eigenschaft UrmPropUserGroup beinhaltet nur die Hauptbenutzergruppe, wenn er im alten Benutzersystem vorhanden war.

Beispiel:

In einer Datenbank der Version 5.2 sind folgende Benutzer angelegt: SUPERUSER, USER, SALES, SALES1, SALES2, UPDATE und DEV. Die Benutzer SALES1 und SALES2 sind dem Benutzer SALES, der Benutzer DEV ist dem SUPERUSER untergeordnet. Bei der Konvertierung des Benutzersystems werden jetzt folgende Benutzer angelegt: SUPERUSER, USER, SALES, SALES1, SALES2, UPDATE und DEV. Von jedem Benutzer, der keinen Hauptbenutzer hat, wird eine Benutzergruppe angelegt: USER, SALES, UPDATE. Statt der Benutzergruppe SUPERUSER, wird die Gruppe _Administrators erstellt.

Nach der Zuordnung der Benutzer zu den Benutzergruppen ergibt sich folgendes Benutzermodell:

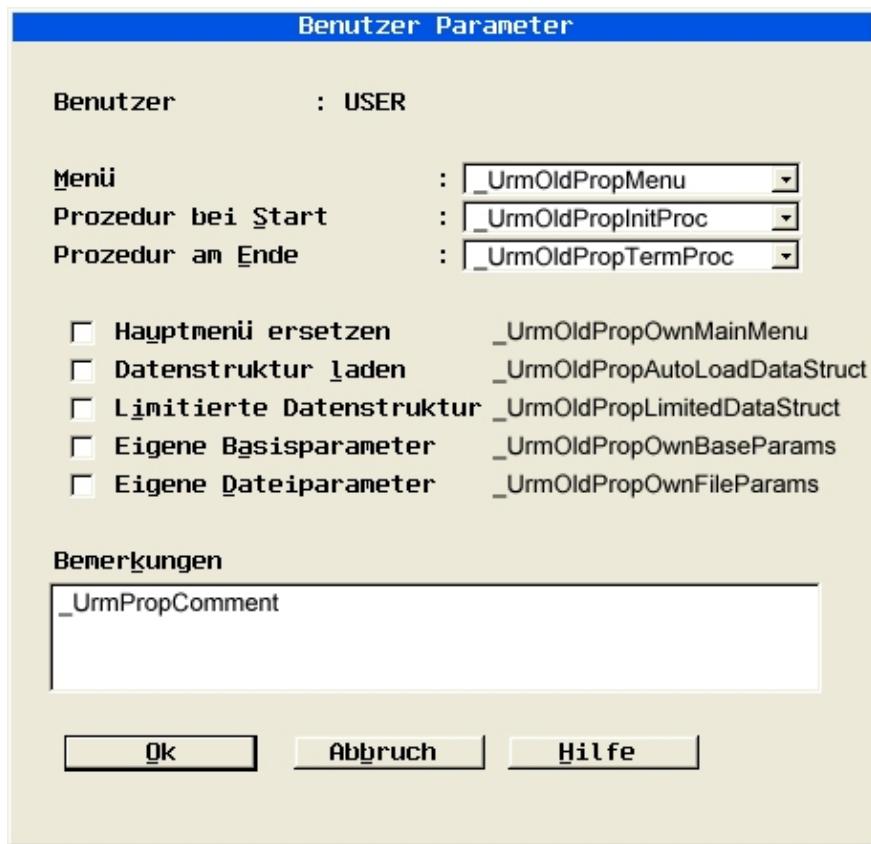
Benutzer	Benutzergruppe
SUPERUSER	_Administrators
USER	USER
SALES	SALES
SALES1	SALES
SALES2	SALES
UPDATE	UPDATE
DEV	_Administrators

Darüber hinaus befinden sich alle Benutzer, die nicht zur Benutzergruppe _Administrators zugewiesen sind, in der Benutzergruppe _Everyone (siehe Benutzerpflege). Die Benutzergruppe hat auf alle Standardelemente alle Rechte

Kontakt

(lesen, schreiben, ändern, ausführen und löschen).

Die Parameter des Benutzers im alten Benutzersystem können über folgende Eigenschaften von der Benutzergruppe abgefragt werden:



[_UrmOldPropMenu](#)

Menü

[_UrmOldPropInitProc](#)

Prozedur bei Start

[_UrmOldPropTermProc](#)

Prozedur bei Ende

[_UrmOldPropOwnMainMenu](#)

Hauptmenü ersetzen

[_UrmOldPropAutoLoadDataStruct](#)

Datenstruktur laden

[_UrmOldPropLimitedDataStruct](#)

Limitierte Datenstruktur

[_UrmOldPropOwnBaseParams](#)

Eigene Basisparameter

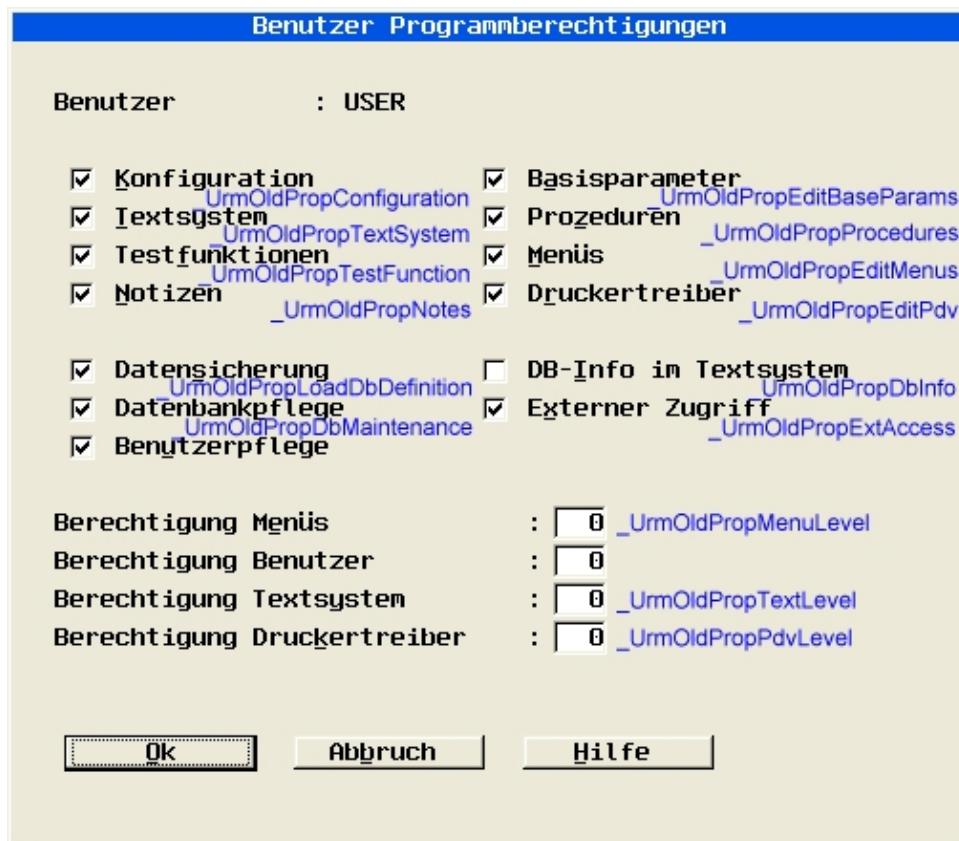
[_UrmOldPropOwnFileParams](#)

Eigene Dateiparameter

Die Bemerkung zu dem Benutzer wird in der neuen Eigenschaft [_UrmPropComment](#) abgelegt.

Die Programmberichtigungen des Benutzers stehen bei der Benutzergruppe über folgende Eigenschaften zur Verfügung:

Kontakt



UrmOldPropConfiguration	Konfiguration
UrmOldPropTextSystem	Textsystem
UrmOldPropTestFunctions	Testfunktion
UrmOldPropNotes	Notizen
UrmOldPropLoadDbDefinition	Datensicherung
UrmOldPropDbMaintenance	Datenbankpflege
UrmOldPropEditBaseParams	Basisparameter
UrmOldPropProcedures	Prozeduren
UrmOldPropEditMenus	Menüs
UrmOldPropEditPdv	Druckertreiber
UrmOldPropDbInfo	DB-Info im Textsystem
UrmOldPropExtAccess	Externer Zugriff
UrmOldPropMenuLevel	Berechtigung Menüs
UrmOldPropTextLevel	Berechtigung Textsystem
UrmOldPropPdvLevel	Berechtigung Druckerteiber

Die Berechtigungen zur Benutzersteuerung können nicht konvertiert werden und müssen dem neuen Benutzermodell angepasst werden (siehe [Benutzerpflege](#)).

Die neuen [UserGroup](#)-Objekte beinhalten ebenfalls noch Berechtigungen aus Datenbanken den Version 3.x:

[UrmOldPropEditColors](#) Programmbenutzung "Farbpaletten"

Kontakt

[UrmOldPropBatches](#) Programmbenutzung "Abläufe"

[UrmOldPropProtocol](#) Programmbenutzung "Protokoll starten"

[UrmOldPropStartBatch](#) Benutzerparameter "Ablauf bei Start"

Diese Berechtigungen werden von den Clients der Version 5.3 nicht verwendet, sie werden nur vom Client für Datenbanken der Version 3.x ausgewertet.

Die Dateiberechtigungen können ebenfalls abgefragt werden. Um die Kompatibilität zu dem alten Benutzersystem zu gewährleisten, gibt es für jede Datei eine eigene Elementgruppe. Die Elementgruppe hat den Namen Table_<Dateinummer>, wobei <Dateinummer> die vierstellige Nummer der Datei mit führenden Nullen darstellt.

 Diese Aufteilung wurde alleine aus der Notwendigkeit einer zum alten Benutzersystem kompatiblen Darstellung der Dateirechte gewählt. Es ist nicht sinnvoll, wenn eine Elementgruppe nur aus einem Element besteht, da so eine unübersichtliche Menge an Elementgruppen entsteht. Sinnvoll ist hier funktionelle Gruppen von Tabellen zu bilden (zum Beispiel: Tabellen für Auftragsköpfe, Auftragspositionen, Lieferadressen usw.), die dann mit gemeinsamen Rechten ausgestattet werden.

Die Dateiberechtigungen stehen über folgende Eigenschaften zur Verfügung:

Benutzer Dateiberechtigung			
Benutzer	: USER		
Datei	: 100 / <code>tblCstCustomer</code>		
Berechtigung	: 1 <code>UrmPermLevelGet()</code> / <code>UrmPermLevelSet()</code>		
Verwaltung		Parameter	
<input type="checkbox"/> Eingabe	<code>_UrmOldPermEntry</code>	<input type="checkbox"/> Selektionen	<code>_UrmOldPermSelections</code>
<input type="checkbox"/> Zugriff	<code>_UrmOldPermAccess</code>	<input type="checkbox"/> Listenformate	<code>_UrmOldPermListFormats</code>
<input type="checkbox"/> Speichern	<code>_UrmOldPermSave</code>	<input type="checkbox"/> Zugriffslisten	<code>_UrmOldPermRecLists</code>
<input type="checkbox"/> Ändern	<code>_UrmOldPermModify</code>	<input type="checkbox"/> Transfers	<code>_UrmOldPermTranfers</code>
<input type="checkbox"/> Löschen	<code>_UrmOldPermDelete</code>	<input type="checkbox"/> Allgemeine	<code>_UrmOldPermParameters</code>
Bearbeiten			
<input type="checkbox"/> Selektieren	<code>_UrmOldPermExecSelections</code>	<input type="checkbox"/> Konfiguration	<code>_UrmPermConfig</code>
<input type="checkbox"/> Listen	<code>_UrmOldPermExecLists</code>		
<input type="checkbox"/> Datenübernahme	<code>_UrmOldPermExecTransfers</code>		
Ok		Abbruch	
		Hilfe	

[UrmPermLevelGet\(\)](#) Abfragen des Dateiberechtigungslevel

[UrmPermLevelSet\(\)](#) Setzen des Dateiberechtigungslevel

[_UrmOldPermEntry](#) Dateiberechtigung "Eingabe"

Kontakt

<u>UrmOldPermAccess</u>	Dateiberechtigung "Zugriff"
<u>UrmOldPermSave</u>	Dateiberechtigung "Speichern"
<u>UrmOldPermModify</u>	Dateiberechtigung "Ändern"
<u>UrmOldPermDelete</u>	Dateiberechtigung "Löschen"
<u>UrmOldPermSelections</u>	Dateiberechtigung "Selektionen"
<u>UrmOldPermListFormats</u>	Dateiberechtigung "Listenformate"
<u>UrmOldPermRecLists</u>	Dateiberechtigung "Zugriffslisten"
<u>UrmOldPermTransfers</u>	Dateiberechtigung "Transfers"
<u>UrmOldPermParameters</u>	Dateiberechtigung "Allgemeine"
<u>UrmOldPermExecSelections</u>	Dateiberechtigung "Selektieren"
<u>UrmOldPermExecLists</u>	Dateiberechtigung "Listen"
<u>UrmOldPermExecTransfers</u>	Dateiberechtigung "Datenübernahme"
<u>UrmPermConfig</u>	Dateiberechtigung "Konfiguration"



Die Berechtigungen eines Benutzers werden ab der Version 5.3 bereits beim CONZEPT 16-Server verarbeitet.

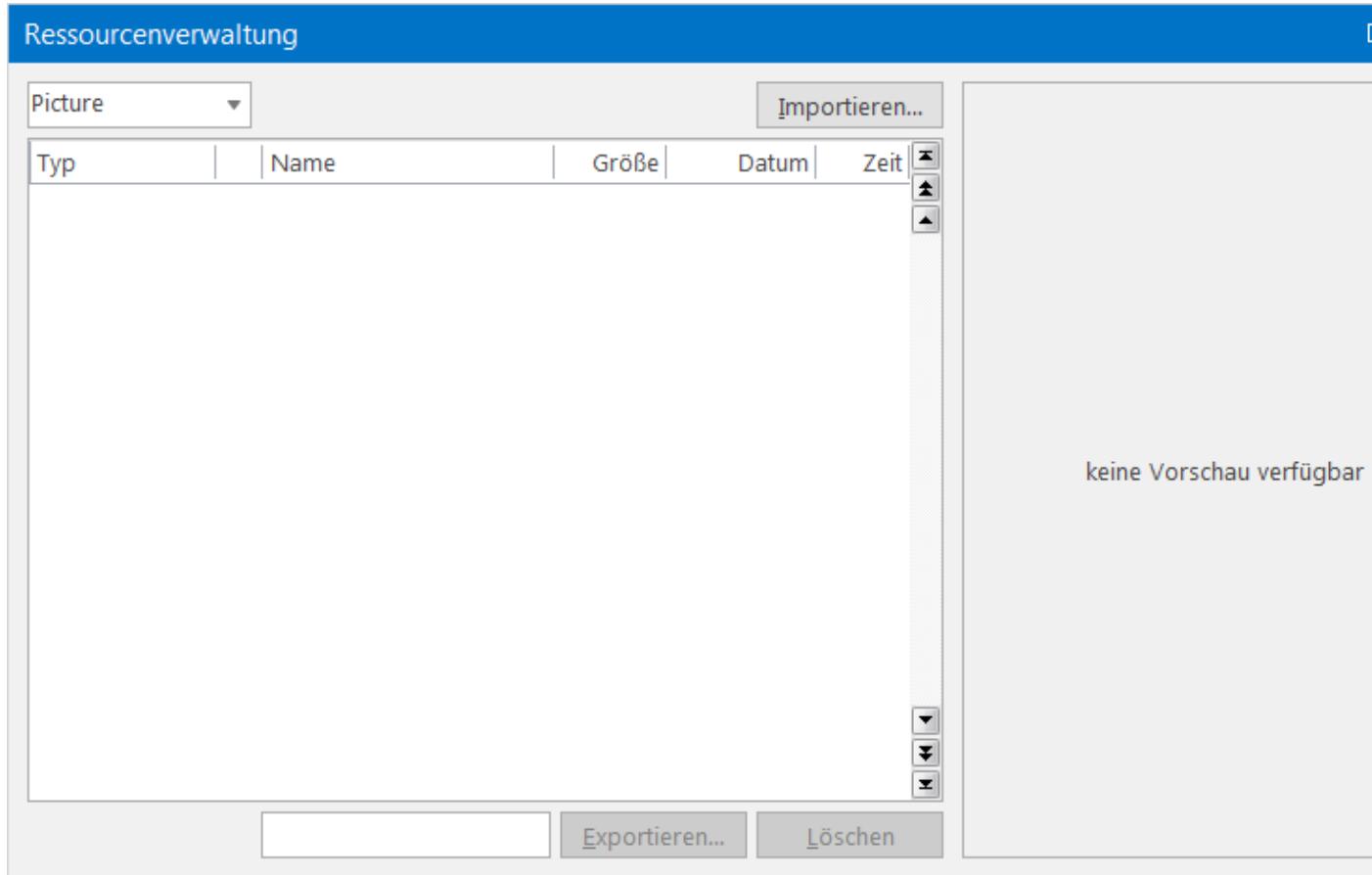
Kontakt

Ressourcenverwaltung

Verwaltung von Ressourcen

Über diesen Menüeintrag werden unterschiedliche Ressourcen, die innerhalb der Datenbank gespeichert wurden, angezeigt. Weiterhin gibt es dort die Möglichkeit weitere Ressourcen zu importieren, vorhandene zu exportieren oder zu löschen.

Nach dem Aufruf des Menüpunktes "Extras" / "Ressourcenverwaltung..." wird folgender Dialog angezeigt:



In dem Dropdown Menü oben links wird die Art der Ressource ausgewählt. Die Liste passt sich an die Auswahl an.

Mit der Schaltfläche Importieren wird eine neue Ressource hinzugefügt.

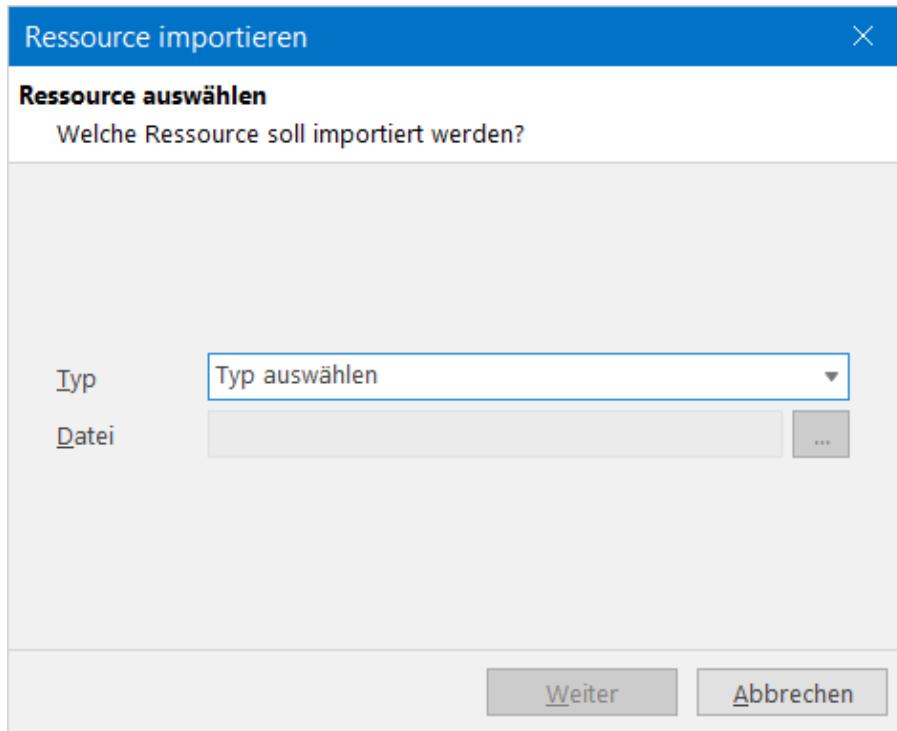
Über die Schaltflächen Exportieren und Löschen wird die momentan selektierte Ressource exportiert bzw. gelöscht.

Ressource importieren
Import von Ressourcen
Import von
Siehe Kachelgrafiken
(Blog)

Ressource auswählen

Über diesen Dialog kann eine Ressource für die Applikation innerhalb der Datenbank gespeichert werden.

Nach dem Klick auf "Importieren..." in der wird folgender Dialog angezeigt:



Als erstes muss hier ein Ressourcentyp ausgewählt werden. Folgende Typen stehen zur Auswahl:

- **Rastergrafik**

Die Grafiken die beispielsweise im Picture- oder Icon-Objekt angezeigt werden.

Formate: *.bmp, *.ico, *.jpg, *.jpeg, *.jpe, *.jfif, *.png, *.gif, *.tif, *.tiff

- **Vektorgrafik**

Die Grafiken die im MetaPicture-Objekt angezeigt werden.

Formate: *.emf, *.wmf

- **Kachelgrafik für Schaltflächen- und Listenobjekte**

Formate: *.bmp, *.png

- **Kachelgrafik für Baumobjekte**

Formate: *.bmp, *.png

- **Kachelgrafik für Menüobjekte**

Formate: *.bmp, *.png

Kachelgrafiken

Eine Kachelgrafik besteht aus mehreren Kacheln, die über ihre Nummer angesprochen werden können. Eine Referenz zu einer bestimmten Kachel besteht aus dem Importnamen und der Nummer der Kachel.

Es wird eine Farbtiefe von 4 Bit (16 Farben), 8 Bit (256 Farben) oder 32 Bit (16,7 Millionen Farben) empfohlen.

 Bei Kachelgrafiken für Menü, Schaltflächen- und Listenobjekte müssen die einzelnen Kacheln in einem Bild durch einen Rand von einem Pixel von einander getrennt werden. Als Größe wird die verwendete Größe einer Kachel (ohne Rand) angegeben.

Ist eine Kachelgrafik in unterschiedlichen Farbtiefen vorhanden, wählt CONZEPT 16 automatisch die zu verwendende Kachelgrafik anhand der angezeigten Farben des Systems. Bei Systemen mit 256 Farben oder weniger wird die Grafik ohne Erweiterung verwendet. Bei Systemen mit 16-, 24- und 32-Bit Farbtiefe wird die Grafik mit der Erweiterung .hcm (High Color Mode) verwendet. Auf Systemen mit 24- und 32-Bit Farbtiefe wird die Grafik mit der Erweiterung .tcm (True Color Mode) ausgewählt.

Existiert keine Grafik mit der Erweiterung .tcm wird die mit der Erweiterung .hcm geladen. Existiert auch diese nicht, wird die Tile-Grafik ohne Erweiterung verwendet.

Zusätzlich kann nach der Erweiterung (.tcm bzw. .hcm) noch eine Tile-Größe (.16, .24, .32 bzw. .48) angegeben werden.

Ist die Eigenschaft TileName... auf den Wert "BtnTile" gesetzt, wird je nach Farbtiefe des Systems und Wert der Eigenschaft TileSize folgende Kachelgrafik ausgewählt:

- **TrueColor-Modus (24, 32 BPP)**

- ◆ BtnTile.tcm.<TileSize>
- ◆ BtnTile.tcm
- ◆ BtnTile.hcm.<TileSize>
- ◆ BtnTile.hcm
- ◆ BtnTile.<TileSize>
- ◆ BtnTile

- **HiColor-Modus (16 BPP)**

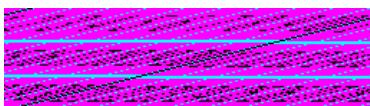
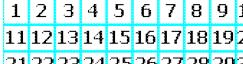
- ◆ BtnTile.hcm.<TileSize>
- ◆ BtnTile.hcm
- ◆ BtnTile<TileSize>
- ◆ BtnTile

- **Sonstige Modi**

- ◆ BtnTile<TileSize>
- ◆ BtnTile

 Die Grafiken mit der Erweiterung .<TileSize> werden nur verwendet, wenn die Eigenschaft TileTheme auf WinTileThemeEnhanced gesetzt ist.

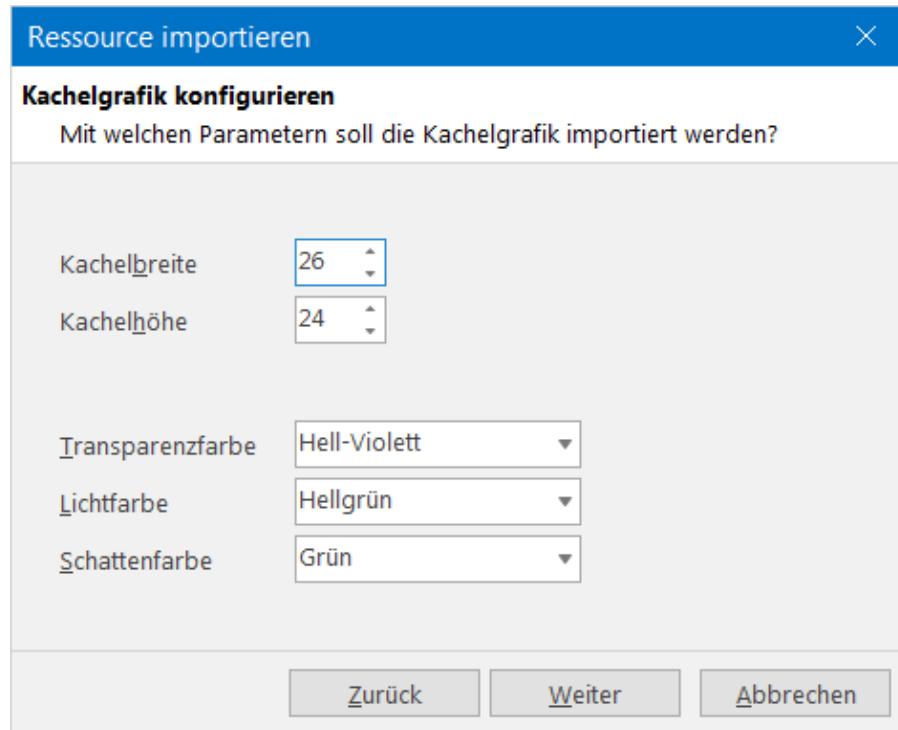
Beispiele:

Typ	8-Bit	32-Bit	Breite	Höhe
Kachelgrafik für Schaltflächen- und Listenobjekte		 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	24	22
Kachelgrafik für Baumobjekte			16	16
Kachelgrafik für Menüobjekte		 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	16	16

 Die Bilder für Schaltflächen- und Menüobjekte werden hier aus Platzgründen in 75%iger Größe angezeigt.

Kachelgrafik konfigurieren

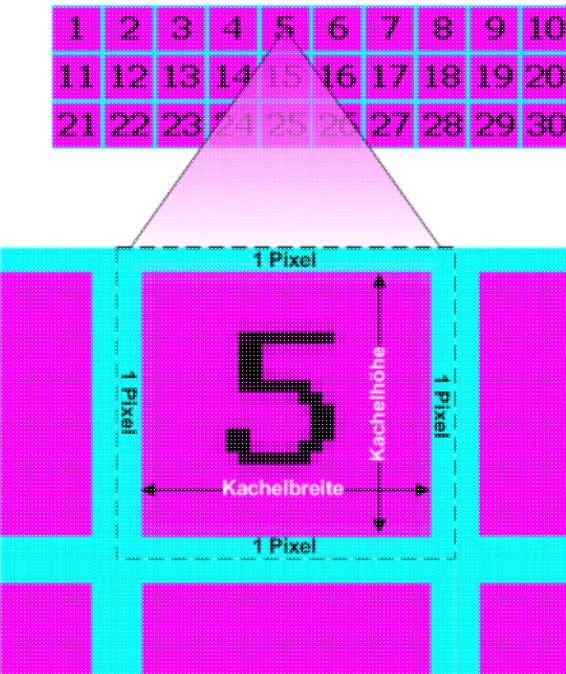
Wird eine Kachelgrafik ausgewählt sieht die nächste Seite wie folgt aus:



Es wird nun bestimmt wie groß ein einzelnes Bild in der Kachelgrafik ist. Die Größe wird dabei ohne Rand angegeben. Bei Kachelgrafiken für Schaltflächen-, Listen- und

Kontakt

Menü-Objekte muss um jede Kachel ein Rand von einem Pixel sein. Somit ist zwischen zwei Kacheln ein Rand von 2 Pixeln. Siehe Bild:



Weiterhin können hier die Transparenzfarbe, die Lichtfarbe und die Schattenfarbe definiert werden. Bei Bildern mit Alpha-Kanal (32 Bit Farbtiefe) werden diese Farben nicht berücksichtigt.

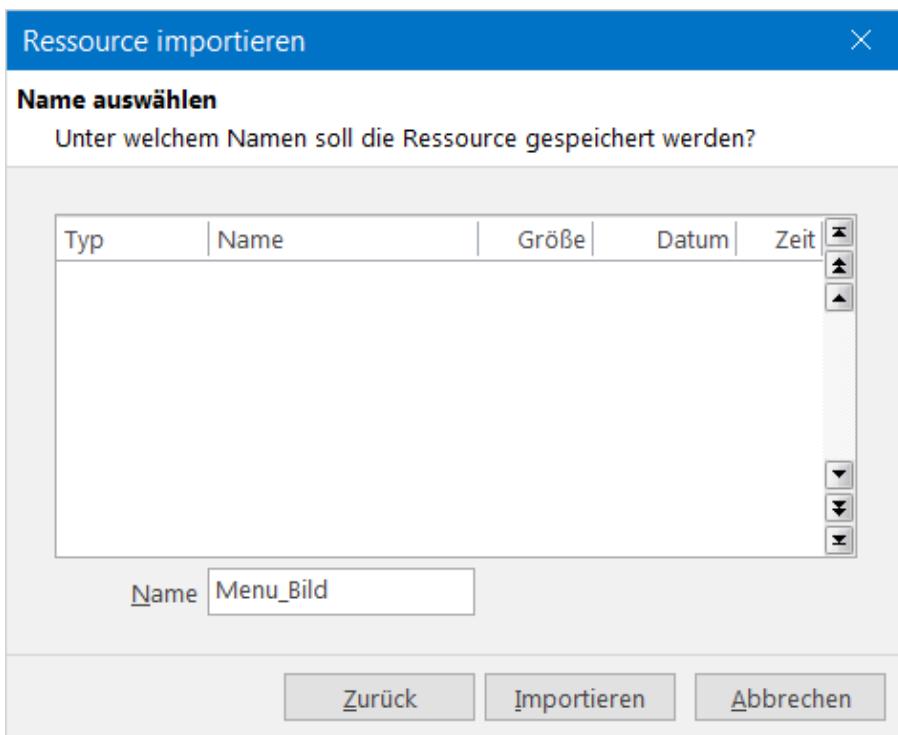
- i** Bei Kachelgrafiken für Menüs können maximal 16 x 16 Pixel angezeigt werden. Größere Kacheln werden abgeschnitten.
Bei Bildern ohne Alpha-Kanal müssen in den Feldern Transparenzfarbe, Schattenfarbe und Lichtfarbe die Farben angegeben werden, die bei der Anzeige der Kachel durch die entsprechenden Systemfarben ersetzt werden. Die Transparenzfarbe wird durch den Hintergrund ersetzt. Die Schattenfarbe und die Lichtfarbe werden durch [WinColBtnShadow](#) und [WinColBtnHighLight](#) ersetzt. Bei Grafiken für Baumobjekte entfällt die Angabe der Schatten- und Lichtfarbe.

Name auswählen

Wird keine Kachelgrafik ausgewählt oder die Kachelgrafik wurde bereits konfiguriert, erscheint diese Seite.

Der Importname darf nur aus den Buchstaben A...z den Ziffern 0...9 sowie dem "_" und dem "." bestehen, wobei das erste Zeichen ein Buchstabe sein muss. Zwischen Groß- und Kleinschreibung wird nicht unterschieden. Die externe Datei wird nach dem Import nicht mehr benötigt.

Kontakt



Hier wird nun der Name der Datei angegeben. Es wird standardmäßig der Name der Datei ohne Endung vorgeschlagen. Wird ein bereits verwendeter Name angegeben. Wird die alte Ressource überschrieben.

Kontakt

Validierungs-Editor

Verwalten von binären Objekten und Verzeichnissen

Siehe [Validierungsbefehle](#)

Über den Menüeintrag **Extras / Validierungs-Editor...** wird der Validierungs-Editor angezeigt. Der Validierungs-Editor erlaubt es eine Validierung für Datenbank-Elemente vorzunehmen.



Zum Aufrufen benötigt der Benutzer das Recht [UrmIdePermRead](#) auf der [Standard-Elementgruppe validation](#).

Nach dem Aufruf des Menüpunktes "Extras" / "Validierungs-Editor..." wird folgender Dialog angezeigt:

Typ	Name	Validierung Status	Validierung Datum	Validierung Uhrzeit	Letzte Änderung Benutzer	Letzte Änderung Datum
Avi	Beispiel	✓				
	BinData	✓				
	BinRename	✓				
	ChartDesigner	✓				
	ComDlgMultiSelect	✓				
	ComError	✓				
	ComOutlook	✓				
	ComWord1	✓				
	ComWord2	✓				

Der Validierungs-Editor besteht aus der [Validierungsliste](#) und dem Ribbon mit den Seiten [Filter](#) und [Ansicht](#).

Validierungsliste

Die Validierungsliste zeigt die [Validierungselemente](#) gemäß dem eingestellten Filter an. Liegen keine [Validierungselemente](#) vor, ist die Liste leer und es wird eine Schaltfläche angezeigt, mit der ein Validierungslauf durchgeführt werden kann. Beim Validierungslauf werden [Validierungselemente](#) des aktuell eingestellten Typs (Dialog, Menü, Prozedur oder Tabelle) erstellt und zur Validierungsliste hinzugefügt. Der Vorgang kann je nach Umfang der in der Datenbank enthaltenen Elemente länger dauern, kann aber jederzeit abgebrochen werden. Die Liste enthält dann nur die Elemente, die bis zu diesem Zeitpunkt erstellt wurden.

Die Validierungsliste zeigt folgende Informationen an:

- **Typ**

Hierbei handelt es sich um den Typ des Validierungselementes (Dialog, Menü, Prozedur oder Tabelle).

- **Name**

Der Name des Validierungselementes. Dieser ist beim Zeitpunkt der Erstellung (Validierungslauf) identisch mit dem Name des referenzierten Objektes (Dialog, Menü, Prozedur oder Tabelle).

- **Validierung Status**

Zeigt den aktuellen Validierungsstatus des Elementes an. Folgende Status sind möglich:

Icon	Bedeutung	Beschreibung
	Unbestimmt	Es wurde nicht festgelegt, ob das Element für die Validierung relevant ist oder nicht.
	Nicht relevant	Das Element ist nicht für die Validierung relevant.
	Relevant, jedoch nicht überprüft	Das Element ist zwar relevant für die Validierung, es wurde jedoch noch nicht validiert. Dies ist der Initialzustand nach dem Validierungslauf für neue <u>Validierungselemente</u> .
	Validiert	Das Referenzelement wurde validiert. Die Einträge in den Spalten "Validierung Datum" und "Validierung Uhrzeit" geben den Validierungszeitpunkt wieder.
	Validiert, jedoch modifiziert	Das Referenzelement wurde modifiziert, nachdem es zuvor validiert wurde. Die Zeitpunkte für Validierungselement und Referenzelement können in den Spalten "Validierung Datum", "Validierung Uhrzeit" und "Letzte Änderung Datum", "Letzte Änderung Uhrzeit" abgelesen werden. Der Benutzer, der das Referenzelement geändert hat, wird in der Spalte "Letzte Änderung Benutzer" angezeigt.
		Bei Validierungselementen vom Typ "Tabelle" sind die Spalten leer, da hier diese Informationen nicht vorhanden sind. Anstelle dessen wird hier ein Hash-Wert für die Ermittlung des Status herangezogen.
	Validiert, jedoch gelöscht	Das Referenzelement wurde nach der Validierung gelöscht.

- **Validierung Datum**

Datum, an dem der Status des Elementes auf Überprüft gesetzt wurde.

- **Validierung Uhrzeit**

Uhrzeit, zu der der Status des Elementes auf Überprüft gesetzt wurde.

- **Letzte Änderung Benutzer**

Kontakt

Benutzer, der das Referenzelement zuletzt geändert hat.

- **Letzte Änderung Datum**

Datum, an dem das Referenzelement zuletzt geändert wurde.

- **Letzte Änderung Uhrzeit**

Uhrzeit, zu der das Referenzelement zuletzt geändert wurde.



Die in der Liste angezeigten Informationen (bezüglich der Validierung) geben den Zustand zum Zeitpunkt des letzten Validierungslaufes wieder. Zum Aktualisieren der Informationen ist ein erneuter Validierungslauf (Aktualisierung) erforderlich.

In der Validierungsliste können mehrere Einträge mittels ausgewählt und mittels der Auswahl hinzugefügt oder aus dieser entfernt werden.

Kontextmenü der Validierungsliste

Über das Kontextmenü kann der Validierungsstatus für die selektierten Elemente gesetzt werden.



Der Status für als gelöscht markierte Elemente ist nicht änderbar. Automatisch hinzugefügte Elemente, die in der Datenbank gelöscht wurden, können nicht aus der Liste entfernt werden.

Filter

Hierüber kann die Ansicht der Validierungselemente gefiltert werden. Die Filterung unterteilt sich in die Bereiche "Elemente", "Validierungs-Zeitraum", "Validierungs-Benutzer" und "Custom". Ein Filter ist aktiv, wenn das Häkchen vor dem entsprechenden Filterelement gesetzt ist.

- **Elemente**

Über die obere Liste kann der Element-Typ gefiltert werden. Sofern das Häkchen nicht aktiv ist, bestimmt der ausgewählte Eintrag den Typ der in der Liste angezeigten Validierungselemente. Bei gesetztem Häkchen, können in der Popup-Liste mehrere Typen kombiniert werden (mit). Über die untere Liste kann nach Elementstatus gefiltert werden.

- **Validierungs-Zeitraum**

Hierüber kann der Validierungs-Zeitraum eingeschränkt werden. Die Liste enthält bei aktiverter Filterung nur noch Elemente, die im eingestellten Zeitraum validiert wurden.

- **Validierungs-Benutzer**

Hierüber kann der Validierungs-Benutzer eingeschränkt werden. Die Liste enthält bei aktiverter Filterung nur noch Elemente, die von den eingestellten Benutzern validiert wurden.

- **Custom**

Kontakt

Hierüber kann eine Filterung der Elemente über die Custom-Eigenschaft des Validierungselementes vorgenommen werden.



Die Eigenschaft Custom kann nur prozedural gesetzt werden.

Ansicht



Über die Schaltfläche kann ein Validierungslauf durchgeführt werden.

Menüzeile

Menüzeile der Entwicklungsumgebung

In der Menüzeile sind alle Grundfunktionen des Designers enthalten. Über die verschiedenen Menüs können Dialoge bearbeitet und die Werkzeugfenster des Designers aktiviert bzw. deaktiviert werden.



Die Funktionen werden auf folgende Menüs verteilt:

- **Datei**

Hier können neue Dialoge angelegt bzw. vorhandene Dialoge geladen oder gespeichert werden. Zudem befinden sich die Menüeinträge zur Konfiguration des Designers und zum Import- oder Exportieren von Dialogen in diesem Abschnitt.

- **Bearbeiten**

Unter diesem Menüpunkt befinden sich Funktionen zur Verwendung der Windows Zwischenablage und zum Löschen von markierten Objekten. Hier können ebenfalls Objekte markiert werden.

- **Extras**

Diesem Menüpunkt beinhaltet Funktionen zum Bearbeiten der Datenstruktur, die Benutzer- und die Ressourcenverwaltung.

- **Anzeige**

In diesem Menü können alle Werkzeugfenster aktiviert bzw. deaktiviert und angeordnet werden. Über dieses Menü wird ebenfalls die Anzeige im Bereich nach den Menüeinträgen gesteuert.

- **Fenster**

Hier werden alle zur Zeit bearbeiteten Dialoge und Fenster angezeigt. Das Menü erlaubt das schnelle wechseln zwischen den einzelnen Fenstern.

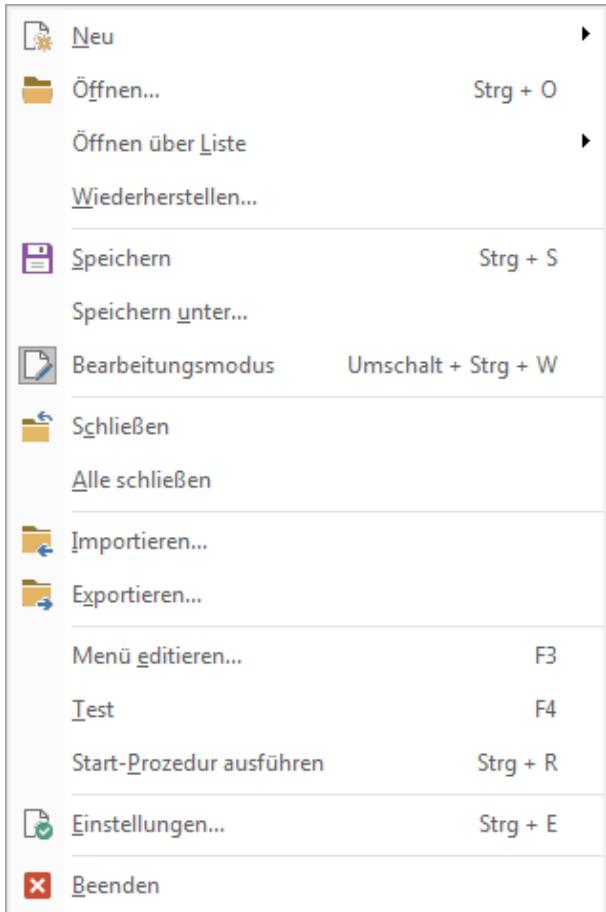
- **?**

In diesem Menü kann die Hilfe und der Releasestand aufgerufen werden.

Menü Datei

Menü zum Erstellen von Dialogen und Menüs

In diesem Menü befinden sich alle Funktionen, um neue Dialoge und Menüs anzulegen und vorhandene zu laden oder zu speichern. Darüber hinaus, können Einstellungen an der Entwicklungsumgebung vorgenommen werden.



- **Neu**

Unter diesem Menüeintrag können neue Dialoge, Druck-Formen oder ein neues Menü angelegt werden.

- **Öffnen...**

Ein bestehender Dialog kann mit diesem Eintrag geöffnet werden.

- **Öffnen über Liste**

In einem untergeordneten Menü werden die neun Dialoge angezeigt, die zuletzt geöffnet wurden. Durch Anklicken kann ein entsprechender Dialog geladen werden.

- **Wiederherstellen...**

Wurde ein bestehender Dialog geändert, kann er über diesen Menüeintrag erneut aus der Datenbank gelesen werden.

- **Speichern**

Kontakt

Mit diesem Eintrag wird der aktuelle Dialog in der Datenbank gespeichert.

- **Speichern unter**

Der aktuelle Dialog wird in der Datenbank gespeichert. Vor der Speicherung wird der Name, unter dem der Dialog gespeichert werden soll, abgefragt.

- **Bearbeitungsmodus**

Der aktuelle Dialog wird zwischen dem "Nur lesen"-Modus und dem Lese-/Schreibmodus umgeschaltet.

- **Schließen**

Der aktuelle Dialog wird geschlossen.

- **Alle Schließen**

Alle Dialoge werden geschlossen.

- **Importieren...**

Über diesen Menüeintrag können Dialoge und Menüs in die Datenbank importiert werden.

- **Exportieren...**

Hier können Dialoge oder Menüs aus der Datenbank in eine externe Datei geschrieben werden.

- **Menü editieren...**

Der Menü-Editor wird geöffnet und das Menü des aktuellen Dialoges geladen.

- **Test**

Der aktuelle Dialog wird in der Testumgebung gestartet. In dieser Umgebung wird keine Funktion ausgeführt.

- **Start-Prozedur ausführen...**

Die in der Einstellungen festgelegte Start-Prozedur wird ausgeführt.

- **Einstellungen...**

Über diesen Menüeintrag können die Einstellungen des Designers geändert werden. Die Änderungen sind nur für einen Benutzer gültig.

- **Beenden**

Die Entwicklungsumgebung wird mit diesem Menüeintrag beendet.

Kontakt

Menüeintrag Neu

Dialoge, Menüs und Themes anlegen

Durch Auswahl dieses Menüeintrages wird ein weiteres Menüfenster geöffnet.

Über dieses Menüfenster kann ein neues Fenster oder Menü angelegt werden. Das Fenster kann anschließend mit weiteren Objekten aus dem Fenster Palette gefüllt werden.

Wird ein Menü angelegt, kann das Menü mit dem Menü-Editor bearbeitet werden. Es stehen vier unterschiedliche Fenster zur Verfügung: Frame, MdiFrame, AppFrame und TrayFrame. Die Objekte unterscheiden sich in ihrer Verwendung und ihren Eigenschaften. Das Frame-Objekt ist ein Fenster, das keine weiteren Fenster-Objekte beinhalten kann. Es wird zur Darstellung von einfachen Dialogen verwendet. Das MdiFrame ist ein Fenster, das einem AppFrame-Objekt untergeordnet werden muss. In einem AppFrame-Objekt können ein oder mehrere MdiFrame-Objekte oder Frame-Objekte angezeigt werden. Das TrayFrame dient zur Erzeugung einer Applikation, die nur im Infobereich der Taskleiste erscheint.

Bestehende Frame-Objekte können über den Menüeintrag **Konvertieren zu MdiFrame** des Kontextmenüs, bzw. Bearbeiten / Konvertieren zu MdiFrame im Menü in ein MDI-Fenster umgewandelt werden. Entsprechend andersrum können über den Menüeintrag **Konvertieren zu Frame** des Kontextmenüs, bzw. Bearbeiten / Konvertieren zu Frame im Menü MdiFrame-Objekte in Frame-Objekte konvertiert werden.

Die Eigenschaften eines Fensters können jederzeit verändert werden. Durch Anklicken des Fensters werden die Eigenschaften im Fenster Eigenschaften angezeigt und können dort verändert werden. Änderungen wirken sich sofort auf das Fenster aus. Die Beschreibung der Fenster-Objekte befindet sich im Abschnitt Form-Palette.

Ähnlich wie Fenster können Druckformulare angelegt werden. Es können vier unterschiedliche Druckformulare erzeugt werden. Die PrintForm und die PrintFormList sind Textbausteine. Aus mehreren dieser Bausteine kann ein Druckjob zusammengestellt werden.

Mit dem Objekt PrintDoc wird ein Druckjob Seitenweise zusammengestellt. Die Position der einzelnen Ausgabebereiche werden auf der Seite fest vorgegeben.

Das PrintDocRecord erstellt eine Liste, bestehend aus Datensätzen einer Datei. Der Aufbau der Liste wird im Designer fest vorgegeben.

Menüeintrag Öffnen...

Objekt öffnen

Menüeintrag

Neu,

Siehe Menüeintrag

Öffnen über

Liste

Ein bereits gespeichertes Objekt kann über diesen Menüeintrag geöffnet werden.

In der Liste werden alle Informationen zu den Objekten angezeigt. Bei Dialogen, Druckformularen und Menüs wird in der Spalte UCS (Universal Character Set) angezeigt, ob für das Objekt die Unicode-Unterstützung aktiviert ist. Die Angabe der Größe erfolgt in Byte.

Über den Eintrag "Typ" kann der Typ des Objekts, das geöffnet werden soll, bestimmt werden. In der Liste werden die Namen der Objekte angezeigt. Nach Auswahl des Objektes und Drücken der Schaltfläche [OK] wird das Objekt geöffnet. Die zuletzt bearbeiteten Objekte werden unter dem Menüeintrag Öffnen über Liste aufgeführt.

Der Name des Objektes kann direkt in das Feld "Name" eingetragen werden. Der Name wird solange grau angezeigt, bis der Name eines gespeicherten Objektes angegeben wurde. Die Schaltfläche [Ok] wird erst dann aktiviert, wenn ein vorhandener Objektname eingegeben wurde.

Wird der Text in dem Feld "Name" rot dargestellt, ist das Objekt von einem anderen CONZEPT 16-Client in Bearbeitung.

Soll das Objekt nicht geändert werden, kann die Option "Nur lesend" aktiviert werden. In diesem Fall können keine Änderungen vorgenommen werden. Die Markierungsrahmen im Anwendungsfenster sowie alle Eigenschaften und Ereignisse der Objekte sind rot dargestellt. Lediglich die Größe und die Position eines geöffneten Dialogs können geändert werden, was sich aber nicht auf das gespeicherte Objekt auswirkt.

Über die Populiste "Unicode" kann für ein bestehendes Fenster oder Menü die Unicode-Unterstützung aktiviert oder deaktiviert werden. Vor der Umwandlung der Zeichenketten erfolgt eine Abfrage. Bei der Umwandlung eines Objekts mit Unicode in ein Objekt ohne Unicode kann es zu Verlusten von Zeichen kommen, die nicht im ANSI-Zeichensatz vorhanden sind.

Folgende Modi stehen zur Verfügung:

Kein Unicode Das Objekt besitzt keine Unicode-Unterstützung.

Unicode-Ausgabe Das Objekt besitzt Unicode-Unterstützung in Ausgabe-Objekten. Objekte, die in diesem Modus gespeichert wurden, haben in der Spalte UCS das Symbol .

Unicode-Eingabe Das Objekt besitzt Unicode-Unterstützung in Ausgabe-Objekten und dem Edit-Objekt. Objekte, die in diesem Modus gespeichert wurden, haben in der Spalte UCS das Symbol .

Kontakt

Weitere Informationen zu Unicode befinden sich im Abschnitt [Unicode-Unterstützung](#).

Bestehende Objekte können mit der -Taste oder über die entsprechende Schaltfläche gelöscht werden.

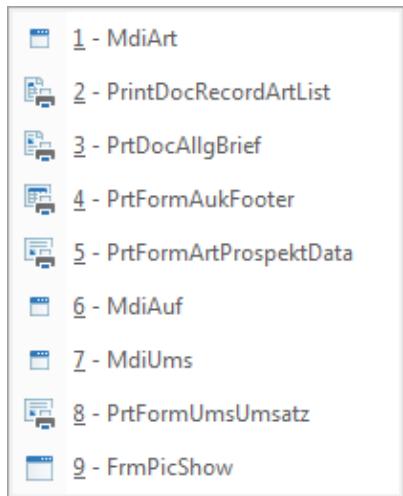
-  Wurde das Objekt mit einer neueren Version des CONZEPT 16-Clients bearbeiten und verfügt in Folge dessen über Eigenschaften und Ereignisse, die in dieser Version des Clients nicht vorhanden sind, erscheint vor dem Öffnen eine entsprechende Warnung. Die betreffenden Eigenschaften und Ereignisse werden nicht geladen.

Kontakt

Menüeintrag Öffnen über Liste

Bestehenden Dialog öffnen

In dem Menüfenster werden die neun zuletzt geöffneten Dialoge angezeigt.



Durch Anklicken eines Eintrages wird das entsprechende Objekt geladen.

Kontakt

Menüeintrag Wiederherstellen...

Lädt einen Dialog erneut aus der Datenbank

Das zur Zeit bearbeitete Fenster kann über diesen Menüpunkt erneut geladen werden.
Alle Änderungen, die seit dem letzten Speichern getätigt wurden gehen dabei verloren.

Kontakt

Menüeintrag Speichern

Dialog speichern

Ein Fenster kann über diesen Menüeintrag gespeichert werden. Hat das Fenster noch keinen Namen, wird in einem Dialog der Name des Fensters abgefragt.



Wurde beim Öffnen des Objekts festgestellt, dass es über Eigenschaften und Ereignisse aus einer neueren Version von CONZEPT 16 verfügt, wird vor dem Speichern des Objekts erneut eine Warnung ausgegeben. Wird das Objekt gespeichert, gehen die neuen Eigenschaften und Ereignisse verloren.

Kontakt

Menüeintrag Speichern unter...

Dialog unter neuem Namen speichern

Mit diesem Menüeintrag wird das aktuelle Fenster gespeichert. Dabei kann in einem Dialog ein neuer Name oder ein bereits bestehender Name angegeben werden.

In der angezeigten Liste kann ein bereits vergebener Name ausgewählt oder ein neuer Name im Feld "Name" eingegeben werden. Wird ein Name aus der Liste ausgewählt, wird das gespeicherte Fenster durch das neue Fenster überschrieben. Durch Klicken auf die Schaltfläche [OK] wird das Fenster gespeichert.

Die Schaltfläche [Ok] ist solange deaktiviert, bis ein korrekter Name in das Feld "Name" eingetragen wurde. Ungültige Namen werden durchgestrichen dargestellt.



Wurde beim Öffnen des Objekts festgestellt, dass es über Eigenschaften und Ereignisse aus einer neueren Version von CONZEPT 16 verfügt, wird vor dem Speichern des Objekts erneut eine Warnung ausgegeben.

Kontakt

Menüeintrag Bearbeitungsmodus

Bearbeitungsmodus umschalten

Das zur Zeit geöffnete Fenster kann über diesen Menüeintrag zwischen dem "Nur lesen"- und dem Lese-/Schreibmodus umgeschaltet werden. Ist das Fenster von einem anderen Benutzer gesperrt, wird der Name und der Rechner des Benutzers in einer Meldung angezeigt.

Kontakt

Menüeintrag Schließen

Dialog schließen

Das zur Zeit bearbeitete Fenster kann über diesen Menüeintrag oder die Schließen-Schaltfläche des Fensters geschlossen werden. Das Fenster wird gespeichert. Ist in den Einstellungen der Entwicklungsumgebung die "Abfrage beim Speichern" (siehe [Einstellungen](#)) aktiviert, erscheint vor dem Speichern eine Abfrage. Sollen alle Fenster geschlossen werden, kann der Menüeintrag Alle Schließen verwendet werden.

Kontakt

Menüeintrag Alle schließen

Kurzbeschreibung

Einzelne Fenster können über den Menüeintrag **Datei / Schließen** geschlossen werden. Sollen alle Fenster geschlossen werden, kann dieser Menüeintrag verwendet werden. Die Fenster werden gespeichert. Ist in den Einstellungen der Entwicklungsumgebung die "Abfrage beim Speichern" (siehe [Einstellungen](#)) aktiviert, erscheint vor dem Speichern jedes Fensters eine Abfrage.

Kontakt

Menüeintrag Importieren...

Dialog oder Menü importieren

Menüeintrag

Exportieren....

Siehe Menüeintrag

Menü

editieren...

Mit diesem Menüeintrag können Menü- und Frame-Objekte sowie Masken, Zugriffslisten und Listenformate aus der textbasierten Oberfläche importiert werden.

Als Resource können Frame- und Menü-Objekte, die aus anderen Datenbanken exportiert wurden, importiert werden.

In dem Feld "Dateityp" kann zwischen den Typen *.rsc (Menü- und Frame-Objekte) und *.exp (Masken, Zugriffslisten oder Listenformate) gewählt werden.

Nach dem Import der Resource ist diese noch nicht in der Datenbank gespeichert.

Kontakt

Menüeintrag Exportieren...

Dialog oder Menü exportieren

Menüeintrag

Importieren...

Siehe Menüeintrag

Menü

editieren...

Das aktuelle Fenster kann über diesen Menüeintrag als separate Datei aus der Datenbank ausgelagert werden. In einem Dialog wird ein entsprechender Dateiname abgefragt.

Das Objekt kann in einem CONZEPT 16-spezifischen Format (.rsc) oder als XML-Datei gespeichert werden. Bei der Speicherung als XML werden alle Eigenschaften und Ereignisse (falls Ereignisse eingetragen wurden) in folgender Struktur geschrieben:

```
<Object Class="Frame" Unicode="false" UCS="none"> <Properties> <Property> <Name>Name</Na
```

Ist das Objekt als Resource (.rsc) gespeichert, kann es in eine andere Datenbank über den Menüeintrag Datei / Importieren... eingelesen werden.

Kontakt

Menüeintrag Menü editieren...

Menü ändern

Zu einem Fenster oder Applikationsfenster kann ein Menü erzeugt werden. Dieses Menü kann über diesen Menüeintrag oder über das Kontextmenü des Fensters angelegt oder verändert werden. Die Änderungen werden mit dem Menü-Editor vorgenommen.

Kontakt

Menüeintrag Test

Dialog testen

Mit diesem Menüeintrag kann ein Fenster überprüft werden. Der Test wird mit dem Schließen des Fensters beendet. Bei diesem Test werden keine Funktionen ausgeführt. Ein umfassender Test kann mit der Testfunktion des Editors oder dem Menüeintrag Start-Prozedur ausführen durchgeführt werden.

Kontakt

Menüeintrag Start-Prozedur ausführen

Prozedur starten

Menüeintrag

Siehe Test,

Menüeintrag

Einstellungen...

Die in den Einstellungen der Entwicklungsumgebung festgelegte Startprozedur wird gestartet. Über diese Startprozedur kann auf einfache Weise eine Prozedur gestartet werden, die nicht im Editor bearbeitet wird. Die Prozedur wird in den Einstellungen der Entwicklungsumgebung angegeben. Die Prozedur kann mit der

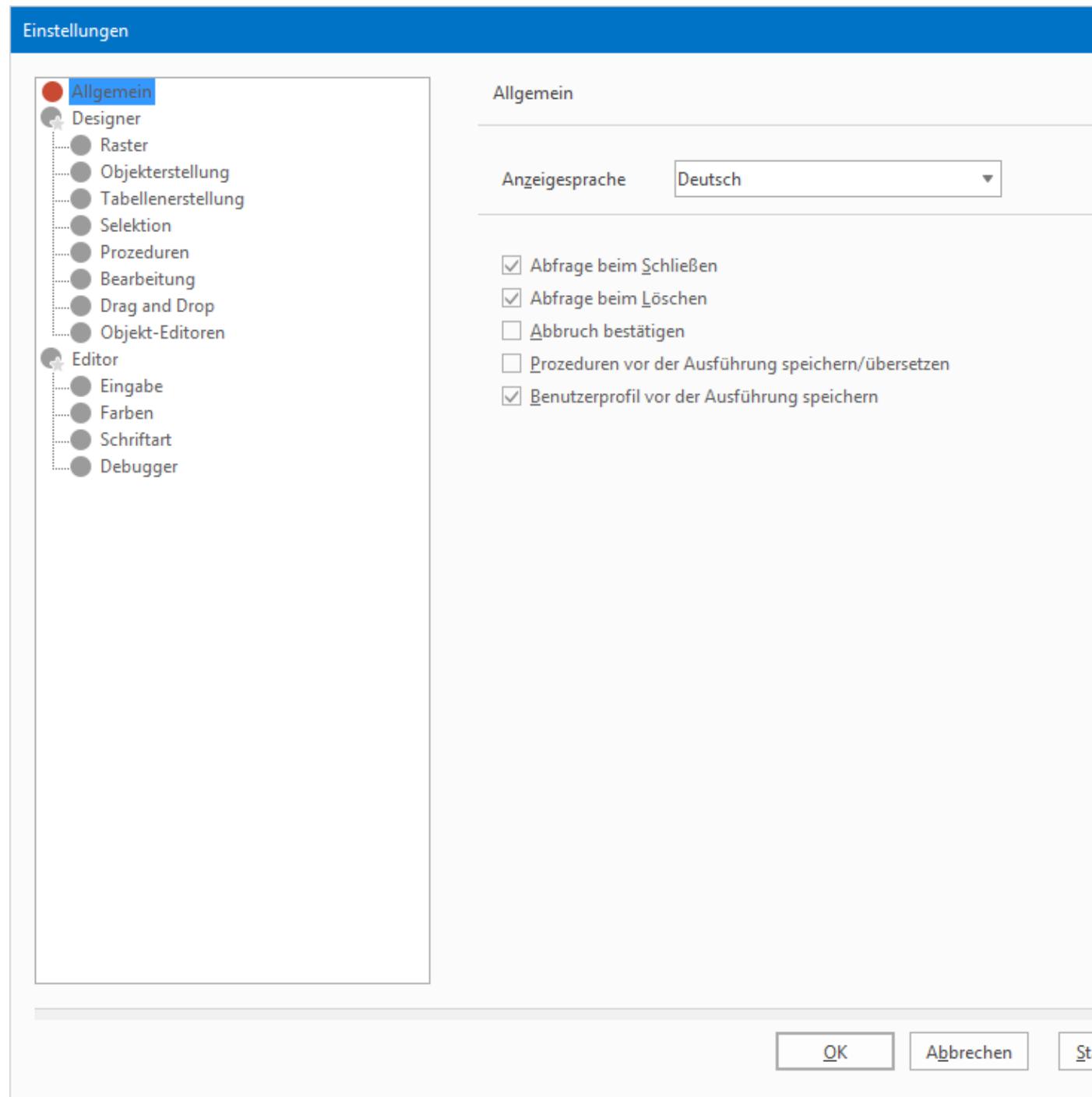
Tastenkombination  +  gestartet werden.

Kontakt

Menüeintrag Einstellungen...

Einstellungen der Entwicklungsumgebung

Einige Eigenschaften der Entwicklungsumgebung können geändert werden. Die Änderungen werden in den Einstellungen vorgenommen. Jede Änderung in den Einstellungen ist nur für den Datenbankbenutzer gültig, der in der Menüzeile angezeigt wird.



Wird in dem Baum der Bereiche die Taste **F3** oder die Tastenkombination **[strg] + [F]** gedrückt, kann in einem Eingabefeld ein Name aus dem Baum angegeben werden.

Kontakt

Nach Abschließen der Eingabe mit wird auf den entsprechenden Bereich positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in dem Baum gesucht. Ist das Ende bzw. der Anfang des Baumes erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Einstellungen können in folgenden Bereichen vorgenommen werden:

- **Allgemein**
- **Designer**
 - ◆ **Raster**
 - ◆ **Objekterstellung**
 - ◆ **Tabellenerstellung**
 - ◊ **Tabellenkopf**
 - ◊ **Seitenkopf**
 - ◊ **Datenkopf**
 - ◊ **Tabellenfuß**
 - ◊ **Seitenfuß**
 - ◊ **Datenfuß**
 - ◊ **Datenbereich**
 - ◆ **Selektion**
 - ◆ **Prozeduren**
 - ◆ **Copy and Paste**
 - ◆ **Drag and Drop**
 - ◆ **Objekt-Editoren**
- **Editor**
 - ◆ **Eingabe**
 - ◆ **Farben**
 - ◆ **Schriftart**
 - ◆ **Debugger**

Kontakt

Einstellungen - Allgemein

Allgemein

Allgemein

The screenshot shows a user interface for setting language preferences. At the top, there is a dropdown menu labeled "Anzeigesprache" with "Deutsch" selected. Below this, a list of checkboxes allows users to enable or disable specific behaviors:

- Abfrage beim Schließen
- Abfrage beim Löschen
- Abbruch bestätigen
- Prozeduren vor der Ausführung speichern/übersetzen
- Benutzerprofil vor der Ausführung speichern
- Automatische Einblendung der Info-Seite deaktivieren

Im Bereich "Allgemein" können folgende Einstellungen vorgenommen werden:

- **Anzeigesprache**

Die Ausgaben des Designers erfolgen in der hier angegebenen Sprache. Die Sprache kann per Kommandozeilenargument bzw. Umgebungsvariable c16lang oder in den Konfigurationsdateien C16.CFG bzw. C16_APGI.CFG in der Einstellung Lang_Display voreingestellt werden.

- **Abfrage beim Schließen**

Beim Schließen des Designers bzw. eines Fensters wird bei nicht gespeichertem Fenster gefragt, ob es gespeichert werden soll.

- **Abfrage beim Löschen**

Wird in einem Objekt-Editor (Menü-Editor, RecList-Editor usw.) ein Objekt gelöscht, erfolgt vorher eine Abfrage.

- **Abbruch bestätigen**

Die Bearbeitung in einem Objekt-Editor abgebrochen, wird der Dialog erst nach einer weiteren Bestätigung geschlossen.

- **Prozeduren vor der Ausführung speichern/übersetzen**

Vor der Ausführung wird die aktuelle Prozedur immer gespeichert und übersetzt. Ist diese Einstellung gesetzt, werden zusätzlich alle anderen geöffneten und geänderten Prozeduren automatisch gespeichert bzw. übersetzt.

- **Benutzerprofil vor der Ausführung speichern**

Standardmäßig ist diese Option aktiviert und bewirkt das Speichern der benutzerspezifischen Einstellungen vor der Ausführung des Befehls "Testen" im Editor (+) bzw. vor dem Durchführen der Startprozedur (+).

- **Automatische Einblendung der Info-Seite deaktivieren**

Standardmäßig wird die Info-Seite im Editor angezeigt, wenn kein anderes Dokument geladen ist. Wird diese Einstellung gesetzt, wird die Info-Seite nur

Kontakt

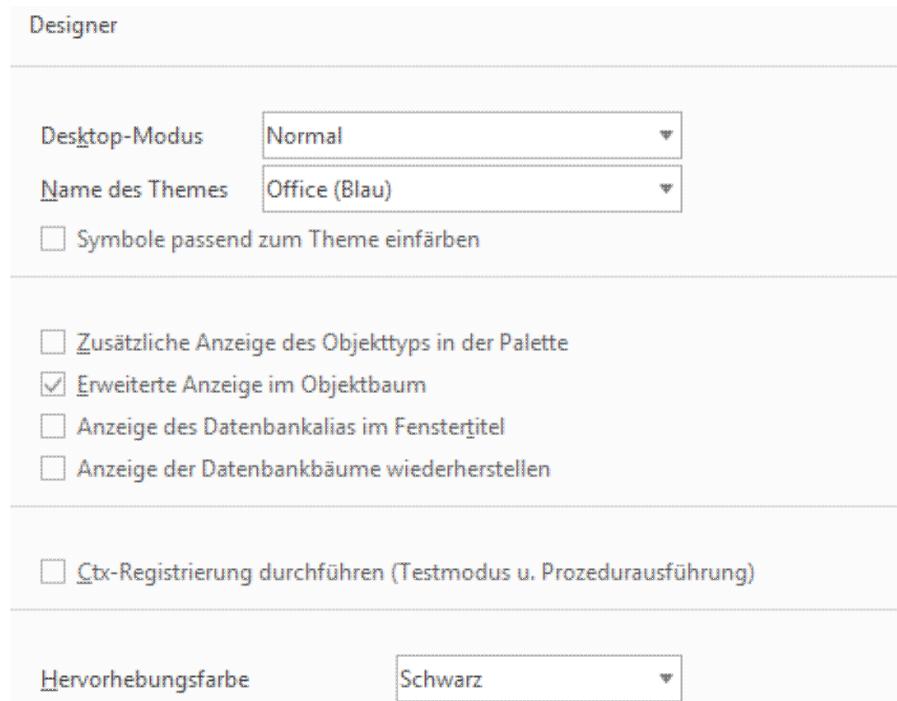
noch über den Menüpunkt Info im Editor geöffnet.

Kontakt

Einstellungen - Designer

Designer

In dem Bereich "Designer" kann die Darstellung des Designers beeinflusst werden.



• Desktop-Modus

Folgende Einstellungen sind hier möglich:

- ◆ Normal Der Hintergrund und noch laufende Applikationen bleiben sichtbar.
- ◆ Desktop-Farbe Der Hintergrund wird mit der Farbe des Desktop gefüllt. Laufende Applikationen sind nicht mehr sichtbar.
- ◆ Desktop-System Der Hintergrund wird mit dem Desktopbild ausgefüllt. Laufende Anwendungen sind nicht mehr sichtbar.
- ◆ Integriert Der Designer stellt sich als MDI-Anwendung dar. Alle geöffneten Frame- und PrintForm-Objekte sind im Designer ingegriffen. Der Designer kann in diesem Modus frei positioniert werden.

• Name des Themes

Hier kann ein Standard-Theme ausgewählt werden, in dem der Designer dargestellt wird. Folgende Einstellungen sind hier möglich:

- ◆ Office (Blau) Blaues Theme, welches an Office 2016 angelehnt ist.
- ◆ Office (Dunkel) Dunkles Theme, welches an Office 2016 angelehnt ist.



Wird Windows im Modus "Hoher Kontrast" ausgeführt, ist diese Einstellung deaktiviert. Der Designer verwendet dann die in Windows konfigurierten Farben.

• Symbole passend zum Theme einfärben

Kontakt

Ist diese Einstellung gesetzt, werden die Symbole im Designer in der zum Theme passenden monochromen Farbe eingefärbt, andernfalls werden mehrfarbige Symbole verwendet.

- **Zusätzliche Anzeige des Objekttyps in der Palette**

Ist diese Einstellung gesetzt, werden in der Palette zusätzlich zu den Icons auch die Objekttypen angezeigt.

- **Erweiterte Anzeige im Objektbaum**

Ist diese Einstellung gesetzt, werden im Objekt-Baum zusätzlich zu dem Objektnamen noch ein Icon, der Objekttyp und der Inhalt dargestellt.

- **Anzeige des Datenbankalias im Fenstertitel**

Ist diese Einstellung gesetzt, wird in der Titelleiste im Designer geöffneter Frame- und PrintForm-Objekte der Datenbank-Alias angezeigt. Im integrierten Modus wird die Option nicht berücksichtigt.

- **Anzeige der Datenbankbäume wiederherstellen**

Ist diese Einstellung gesetzt, werden die offenen Knoten in dem Datenstruktureditor und in der Datenbankinfo beim Beenden des Designers gespeichert und beim nächsten Start wiederhergestellt.

- **Ctx-Registrierung durchführen (Testmodus u. Prozedurausführung)**

Ist diese Einstellung gesetzt, wird beim Starten des Designers, falls notwendig, die Ctx-Registrierung für alle Objekte (siehe beispielsweise InstallCtxOffice und InstallCtxDocEdit) durchgeführt. Ist die Einstellung nicht gesetzt, wird die Komponente nicht automatisch registriert. Auf Systemen, auf denen die Komponente nicht registriert ist, können dann keine Office-Dokumente im Test-Modus (siehe CtxOffice) angezeigt werden.

- **Hervorhebungsfarbe**

Mit dieser Einstellung wird die Farbe für die Hervorhebung der Umrandung von Container-Objekten im Designer definiert. Die Hervorhebung kann im Menü Anzeige mit dem Menüpunkt Objektumrandung hervorheben aktiviert werden.

Einstellungen - Designer

Raster

Im Bereich "Raster" werden Einstellungen vorgenommen, welche die Rasterpunkte im Designer betreffen. Alle Objekte innerhalb eines Applikationsfensters bzw. des Print-Formulars werden auf diese Rasterpunkte positioniert.

• **Frames (Horizontal / Vertikal)**

Die Rasterpunkte bei Fenster-Objekten liegen in dem hier angegebenen Abstand auseinander.

• **PrintForms (Horizontal / Vertikal)**

Die Rasterpunkte bei Druck-Objekten liegen in dem hier angegebenen Abstand auseinander.

• **Container-Raster**

Ist diese Option aktiviert, werden die Rasterpunkte auch innerhalb von Container-Objekten (z. B. GroupBox) dargestellt.

• **Objektposition am Raster ausrichten**

Ist diese Einstellung gesetzt, orientiert sich die Position des Objekts bei Erstellung, Größenänderung und Verschieben an den Rasterpunkten.

• **Objektgröße am Raster ausrichten**

Ist diese Einstellung gesetzt, orientiert sich die Größe des Objekts beim Einfügen, Ändern der Größe und Verschieben an den Rasterpunkten.

• **Beim Einfügen am Raster ausrichten**

Ist diese Option aktiv, werden aus der Zwischenablage eingefügte Objekte automatisch am Raster ausgerichtet, auch wenn diese in den Eigenschaften eine andere Position gesetzt haben.

Einstellungen - Designer Objekterstellung

Im Bereich "Objekterstellung" werden die Abstände zwischen den Objekten bei ihrer Erstellung angegeben. Das neue Objekt wird in dem Abstand zum selektierten Objekt erstellt, der in den Feldern angegeben ist. Sind beide Werte 0, wird das neue Objekt um die Hälfte der Breite und Höhe zum aktuell selektierten Objekt versetzt eingefügt.

Über die Einstellung des "Frame Font" und des "Print Font" kann ein Zeichensatz definiert werden, der bei der Erstellung eines Frame-Objekts ([AppFrame](#), [MdiFrame](#) oder [Frame](#)) bzw. eines Print-Objekts ([PrintForm](#), [PrintFormList](#), [PrintDoc](#) oder [PrintDocRecord](#)) in die Eigenschaft [Font](#) eingetragen wird. Über die Schaltflächen können alle auf dem System installierten Fonts ausgewählt werden.

Über die Einstellung "Unicode-Objekte" kann für Oberflächen- und Druck-Objekte getrennt definiert werden, ob Frame-, Menü- oder Druck-Objekte mit Unicode-Unterstützung angelegt werden.

Bei Frames gibt es folgende Einstellungsmöglichkeiten:

Kein Unicode Neue Frames werden ohne Unicode-Unterstützung erstellt.

Unicode-Ausgabe Neue Frames werden mit Unicode-Unterstützung in Ausgabe-Objekten erstellt.

Unicode-Eingabe Neue Frames werden mit Unicode-Unterstützung in Ausgabe-Objekten und [Edit](#)-Objekten erstellt.

Ist die Eigenschaft für PrintForms gesetzt, werden alle Druck-Objekte, die ab diesem Zeitpunkt neu angelegt werden, mit Unicode-Unterstützung erzeugt.

Im Titel des Fensters bzw. im Titel des [Menü-Editors](#) wird "Unicode" angezeigt, wenn die Unterstützung aktiviert ist.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Tabellenerstellung

Die "Tabellenerstellung" gliedert sich in folgende Bereiche:

- **Tabellenkopf**
- **Seitenkopf**
- **Datenkopf**
- **Tabellenfuß**
- **Seitenfuß**
- **Datenfuß**
- **Datenbereich**

Kontakt

Einstellungen - Designer - Tabellenerstellung

Tabellenkopf

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Druck-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Druck-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Seitenkopf

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Druck-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Print-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Datenkopf

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Print-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, so werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, so wird das Tabellen-Objekt bei der Erstellung des Druck-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Tabellenfuß

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Druck-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Print-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Seitenfuß

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Druck-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Print-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Datenfuß

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Print-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Print-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Kontakt

Einstellungen - Designer - Tabellenerstellung

Datenbereich

Im Bereich "Tabellenerstellung" werden Einstellungen vorgenommen, welche die Tabellenelemente bei der Erstellung von Druck-Formularen betreffen.

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Einstellung aktiv, so werden die Anzahl der Spalten, die Anzahl der Seiten, die Zellbreite, die Zellhöhe und der Zellhöhenmodus automatisch bei der Erstellung des jeweiligen Tabellen-Objekts berücksichtigt. Andernfalls erscheint vor der Erstellung des Objekts das oben dargestellte Einstellungsfenster.

- **Mit Tabelle erstellen**

Ist diese Einstellung aktiv, wird das Tabellen-Objekt bei der Erstellung des Druck-Objekts TblCommon sofort mit erstellt.

- **Spalten / Zeilen**

An dieser Stelle wird die Anzahl der Spalten und Zeilen definiert, die bei der Erstellung des Tabellen-Objekts verwendet werden.

- **Zellbreite / Zellhöhe / Zellhöhe-Modus**

In diesen Einstellungen wird die Zellenbreite und Zellenhöhe des Tabellenobjekts bei der Erstellung definiert. In der Einstellung Zellenhöhen-Modus wird festgelegt, ob sich die Zellenhöhe dem Inhalt automatisch anpasst, oder ob dieser auf der konstanten Zellenhöhe bleibt.

Einstellungen - Designer

Selektion

Der Bereich "Selektion" enthält Einstellungen zum Verhalten von selektierten Objekten.

• Container-Drag

Ist diese Option aktiviert, können innerhalb eines Container-Objektes (z. B. GroupBox) durch Ziehen mit der Maus, Objekte selektiert werden. Zum Verschieben kann der Rand des Objektes verwendet werden. Bei deaktivierter Option wird das Container-Objekt auch dann verschoben, wenn innerhalb des Objektes die Maus mit gedrückter Taste gezogen wird.

• Tastatursteuerung

Die Cursor-Tasten können zum Verschieben von selektierten Objekten verwendet werden. Die Cursortasten verschieben den Bereich um einen Rasterpunkt (Bereich "Raster"). In Verbindung mit der -Taste wird der Bereich um drei Rasterpunkte verschoben.

• Klicken & Ziehen

Ist diese Option aktiv, so ist es nicht mehr notwendig, vor der Verschiebung eines Objektes mit der Maus, dem Objekt explizit den Fokus zu geben. Das Klicken & Ziehen kann ohne Unterbrechung durchgeführt werden.

Kontakt

Einstellungen - Designer

Prozeduren

In dem Feld "Startprozedur" kann eine Prozedur eingetragen werden, die über den Menüpunkt Datei / Start-Prozedur ausführen oder mit der Tastenkombination  +  gestartet werden kann.



Über die Pfeil-Schaltfläche wird eine Liste aller Prozeduren aufgerufen.

Einstellungen - Designer

Bearbeitung

Im Bereich "Bearbeitung" werden Einstellungen vorgenommen, welche die Objektbearbeitung im Designer betreffen.

Bearbeitung

Löschspeicher

10

Rückgängig Schritte

50

Dialog- und Druckformulare nur-lesend öffnen

• **Löschspeicher**

Anzahl der Objekte, die nach dem Löschen wieder hergestellt werden können.

Dies betrifft den Menüpunkt **Bearbeiten / Löschspeicher abrufen**

• **Rückgängig Schritte**

Anzahl der Bearbeitungsschritte, die mit dem Menüpunkt **Bearbeiten / Rückgängig** rückgängig gemacht werden können.

• **Dialog- und Druckformulare nur-lesend öffnen**

Ist diese Option aktiviert, werden Dialoge und Druckformulare standardmäßig nur-lesend geöffnet. Dies betrifft die zuletzt geöffneten Objekte. Zusätzlich wird die Option "Nur lesend" im Öffnen-Dialog vorbelegt.

Einstellungen - Designer

Drag and Drop

Aus dem Fenster "Datenbankinfo" können mit Hilfe der Drag & Drop-Funktionalitäten automatisch Eingabeobjekte zu Feldern der Datenbank erzeugt werden.

Standardmäßig werden beim Ziehen eines Datenbankfeldes in ein Fenster ein Label-Objekt mit dem Namen des Feldes und ein dem Feldtyp entsprechenden Eingabeobjekt erzeugt. In diesem Dialog können die erzeugten Objekte beeinflusst werden.

• **Eingabeobjekte mit Labels erzeugen**

Ist diese Einstellung gesetzt, wird bei der Übernahme von Datenbankfeldern in den Dialog ein Edit-Objekt mit Label (Bezeichner) erzeugt. In die Caption

des Label-Objektes wird der Name des Feldes übernommen. Ist die Einstellung nicht gewählt, wird ein Edit-Objekt ohne Label erzeugt.>

• **Text bei logischen Feldern links anordnen**

Bei logischen Feldern wird eine CheckBox erzeugt.

Der Name des Feldes wird rechts von der Checkbox geschrieben. Bei aktiverter Option wird der Name des Feldes links von der Checkbox geschrieben.

• **Eingabehilfe bei typgebundenen Objekten aktivieren**

Die Eigenschaft InputCtrl des Eingabeobjektes wird auf true gesetzt.

Durch Deaktivieren der Funktion wird kein Input-Control erzeugt.

• **Objektnamen aus Feldnamen automatisch bilden**

Der Name des Eingabeobjektes wird aus einem Kürzel für den Typ des Eingabeobjekts und dem Namen des Datenbankfeldes gebildet.

Ist diese Funktion ausgeschaltet, bleibt der Name des Eingabeobjekts und des Label-Objektes leer.

• **Eingabeobjekte mit Datenstruktur verknüpfen**

In die Eigenschaft DbFieldName wird der Name des Datenbankfeldes eingetragen.

Die Verbindung von Eingabeobjekt und Datenbankfeld erfolgt nicht, wenn die Option deaktiviert ist.

• **Setzen von DbFieldName setzt auch Eingabelimit**

Ist diese Option gesetzt, wird beim Setzen der Eigenschaft DbFieldName eines Eingabe-Objekts, die Eingabemöglichkeiten auf die Möglichkeiten des Datenbankfeldes beschränkt, falls die definierten Eingabemöglichkeiten nicht bereits enger gefasst sind.

Neben der Eigenschaft DbFieldName werden somit abhängig von Typ des Eingabe-Objektes folgende Eigenschaften gesetzt:

Typ	Eigenschaften Wert
<u>Edit</u>	<u>LengthMax</u> maximale Länge des Datenbankfeldes
<u>TextEdit</u>	<u>LengthMax</u> maximale Länge des Datenbankfeldes
<u>IntEdit</u>	<u>MinInt</u> 0 (bei Ganzzahlig-Kurz) / -2.147.483.647 (bei Ganzzahlig-Lang) <u>MaxInt</u> 65.535 (bei Ganzzahlig-Kurz) / 2.147.483.647 (bei Ganzzahlig-Lang)
<u>BigIntEdit</u>	<u>MinBigInt</u> 0 (bei Ganzzahlig-Kurz) / -2.147.483.647 (bei Ganzzahlig-Lang) / -9.223.372.036.854.775.807 (bei Ganzzahlig 64) <u>MaxBigInt</u> 65.535 (bei Ganzzahlig-Kurz) / 2.147.483.647 (bei Ganzzahlig-Lang) / 9.223.372.036.854.775.807 (bei Ganzzahlig 64)

Beispiel:

In einem Fenster befindet sich ein IntEdit-Objekt. Wird diesem Objekt ein Datenbankfeld vom Typ Ganzzahlig-kurz zugewiesen, werden die Eigenschaften MinInt und MaxInt auf 0 und 65535 gesetzt. Wurden diese Eigenschaften bereits zuvor auf 0 und 3000 gesetzt, werden die Eigenschaften nicht verändert.

Kontakt

Einstellungen - Designer Objekt-Editoren

Mit der Tastenkombination  +  (vom Nummernblock) können innerhalb der Objekt-Editoren einzelne Objekte der Zwischenablage hinzugefügt werden. Im Bereich "Objekt-Editoren" kann die maximale Anzahl der hinzufügbaren Objekte angegeben werden. Werte von 1 bis 100 werden akzeptiert.

Einstellungen - Editor
Editor

• **Editor als Toolfenster darstellen**

Das Fenster des Editors wird standardmäßig nicht in der Taskleiste dargestellt. Wird diese Option deaktiviert, erfolgt die Darstellung des Editors als Symbol in der Taskleiste.

• **Lesezeichen laden und speichern**

Ist diese Option aktiviert, werden Lesezeichen von Dokumenten beim Designer-Start geladen und beim Verlassen gespeichert. Ist die Option deaktiviert sind die Lesezeichen des Dokuments zwar noch vorhanden, sie werden jedoch nicht geladen und nicht gespeichert. Standardmäßig ist diese Option aktiviert.

• **Dokumente nur-lesend öffnen**

Ist diese Option aktiviert, werden Dokumente im Editors standardmäßig nur-lesend geöffnet. Dies betrifft den Assistenten, die Eigenschaftsliste,  + Doppelklick auf Includes und Prozeduren und die zuletzt geöffneten Dokumente. Zusätzlich wird die Option "Nur lesend" im Öffnen-Dialog vorbelegt.

• **Zeilennummern anzeigen**

Ist diese Option aktiviert, werden im Editor Zeilennummern angezeigt.

Einstellungen - Editor

Eingabe

Im Bereich "Eingabe" wird das Verhalten des Editors bei der Eingabe bestimmt. Hier kann festgelegt werden, wann eine automatische Einrückung stattfindet.

Eingabe

Einrückung aus

Nach Bereich

Nach vorheriger Zeile

Einrücktiefe

Breakpoint/Lesezeichen setzen mit Strg-Taste

Klammerung hervorheben

Einrückhilfe anzeigen

Automatische Klammerung

Folding-Punkte beim Öffnen von Dokumenten schließen

• Einrücken aus

Nach dem Drücken der Taste  wird der Cursor in der ersten Spalte positioniert.

• Nach Bereich

Nach der Eingabe von "{" wird die nächste Zeile um die Anzahl von Spalten, die in dem Feld "Einrücktiefe" angegeben ist, eingerückt. Bei der Eingabe von "}" wird das Zeichen die gleiche Anzahl von Spalten eingerückt, wie die zugehörige öffnende Klammer. Das Drücken von  positioniert auf die gleiche Spalte, wie in der vorigen Zeile.

• Nach vorheriger Zeile

Der Cursor wird in der gleichen Spalte positioniert, wie in der vorigen Zeile.

• Einrücktiefe

Die Einrücktiefe gibt die Anzahl der Leerzeichen an, die beim Drücken der -Taste eingefügt werden. Hier stehen ebenfalls die Anzahl der Leerzeichen bei einer Einrückung "Nach Bereich".

• Breakpoint / Lesezeichen setzen mit Strg-Taste

Um ein Lesezeichen oder einen Breakpoint in einer Prozedur zu setzen, wird standardmäßig die Taste  bzw.  verwendet. Aus Kompatibilitätsgründen zu früheren Versionen kann ebenfalls die Tastenkombination  +  bzw.  +  dazu verwendet werden, wenn diese Option aktiviert ist.

- **Klammerung hervorheben**

Ist diese Option aktiviert, werden im Editor zusammengehörende Klammern hervorgehoben, wenn der Cursor vor einer Klammer (({} [])) steht. Die Option kann über den Menüpunkt Klammerung hervorheben aktiviert und deaktiviert werden.

- **Einrückhilfe anzeigen**

Ist diese Option aktiviert, wird im Editor am Zeilenanfang die Einrückung durch eine senkrechte, gepunktete Linie gekennzeichnet. Die Linien werden zur Orientierung auch in leeren Zeilen dargestellt.

- **Automatische Klammerung**

Ist diese Option aktiviert, werden im Editor automatisch passende schließende Klammern () }]) ergänzt, wenn eine öffnende Klammer (({ [) getippt wird.

Steht der Cursor in einer Zeile mit hinzugefügter, öffnender runder Klammer, bevor der Cursor die Zeile verlassen hat und das letzte Zeichen in der Zeile ist eine runde schließende Klammer, wird mit der Cursor hinter die runde schließende Klammer gesetzt. Mit der Tastenkombination + wird an der aktuellen Position ein Zeilenumbruch eingefügt. Die Überprüfung auf die runde schließende Klammer am Zeilenende bei wirkt sich dann auf die neue Zeile aus. Hat der Cursor nach Einfügen der runden öffnenden Klammer die aktuelle Zeile verlassen, wird mit immer ein Zeilenumbruch eingefügt.

- **Folding-Punkte beim Öffnen von Dokumenten schließen**

Ist diese Option aktiviert, werden alle Folding-Punkte der Dokumente geschlossen, wenn der Designer geöffnet oder Dokumente geladen werden. Standardmäßig sind alle Punkte offen. Befand sich der Cursor vor dem Schließen des Designers innerhalb einer Funktion, wird der Funktionsblock, in dem der Cursor steht, immer geöffnet.

Kontakt

Einstellungen - Editor

Farben

Im Bereich "Farben" werden die Farben des Editors eingetragen.

The screenshot shows the 'Farben' (Colors) settings panel. At the top, there is a table with two columns: 'Farbe' (Color) and 'Wert' (Value). The table lists several color categories with their corresponding WinCol values:

Farbe	Wert
Hintergrund	_WinColWindow
Text	_WinColWindowText
Kommentar	_WinColGreen
Numerische Werte	_WinColLightRed
Zeichenkette	_WinColCyan
Schlüsselwort	_WinColBlue
Markierung	_WinColHighLight

Below the table is a code editor window. It contains the following C code:

```
1  @A+
2  @C+
3
4  Markierter Text
5  #define DEFINE_EXAMPLE
6
7  0x80000000 1.75 4711 // Numerische Werte
8  'Zeichenkette'      /* Kommentar */
9
10 sub SubExample(aHandle : handle)
11 {
12     aHandle->wpName # 'MyName';
13     WinDialog('MyForm', _WinDialogCenter);
14 }
```

• Farben

Hier werden die Farben für die einzelnen Syntaxtypen im Editor eingestellt. Durch Anklicken eines Wertes kann eine neue Farbe aus einer ComboBox ausgewählt werden.

• Benutzerdefinierte Farben

Ist ein Hacken bei *Benutzerdefinierte Farben* gesetzt, werden die Farben aus der Liste verwendet. Andernfalls werden die Farben des definierten Themes (siehe Einstellungen - Designer) benutzt.

• Schaltfläche [Standard]

Über diese Schaltfläche wird das Standard Farb-Schema eingestellt.

Kontakt

Im unteren Bereich befindet sich eine Vorschau der aktuell eingestellten Farben.

Kontakt

Einstellungen - Editor

Schriftart

Der Text im Editor wird in der im Bereich "Schriftart" eingestellten Schrift angezeigt. Der Zeichensatz kann aus allen Windowszeichensätzen mit einer festen Zeichenbreite ausgewählt werden. Neben dem Font kann die Größe der Zeichen angegeben werden. Die zur Verfügung stehenden Font-Größen sind von dem ausgewählten Font abhängig.

Kontakt

Einstellungen - Editor

Debugger

Im Bereich "Debugger" können die Adressen von Debuggern angegeben werden.

Bei der normalen Verarbeitung wird eine Prozedur in den Puffer gelegt, bevor sie ausgeführt wird. Sind nicht ausreichend unbenutzte Puffersegmente vorhanden, wird die Prozedur aus dem Puffer entfernt, die am längsten nicht aufgerufen wurde. Auf diese Weise kann die Verarbeitung von Prozeduren beschleunigt werden, wenn eine Prozedur aufgerufen wird, die sich bereits im Puffer befindet. Sie muss dann nicht extra aus der Datenbank gelesen werden.

Läuft der CONZEPT 16-Debugger nicht auf der gleichen Maschine, wie der CONZEPT 16-Client, kann die IP-Adresse oder der Name des Rechners mit dem Debugger in dieser Liste eingetragen werden. Um eine neue IP-Adresse oder einen neuen Namen einzutragen, wird der Eintrag "<leer>" angeklickt und die Adresse eingegeben. Mit Drücken der Taste  wird die Adresse in die Liste übernommen und der Eintrag "<leer>" rückt in die nächste Zeile.

Wird ein Eintrag in der Liste ausgewählt und die Schaltfläche [Standard-Verbindung] gedrückt, wird der entsprechende Eintrag in das dahinterstehende Feld übernommen. Es wird dann auf diesem Rechner nach einem laufenden Debugger gesucht. Ist kein Standard definiert, wird der Debugger auf der lokalen Maschine gestartet und eine Verbindung aufgebaut. In der Liste können die Rechner eingetragen werden, die häufig zum Debuggen verwendet werden.

Um die Standardverbindung zu entfernen, muss entweder der entsprechende Eintrag in der Liste gelöscht oder der Eintrag "<Leer>" als Standardverbindung definiert werden.

Wird die Ausführung einer Funktion durch den Debugger unterbrochen, erscheint beim Client ein entsprechendes Hinweisfenster. Das Fenster erscheint auch, wenn der Debugger auf der gleichen Maschine, wie der Client gestartet wurde. Durch Deaktivieren der Option "Status-Meldungsfenster auch für lokalen Debugger anzeigen" kann das Meldungsfenster unterdrückt werden.

Kontakt

Menüeintrag Beenden

Programmierumgebung verlassen

Mit diesem Menüeintrag wird die Programmierumgebung verlassen. Sind noch nicht gespeicherte Editierfenster oder Prozeduren geöffnet, erfolgt eine Abfrage, ob gespeichert werden soll.

Die zu diesem Zeitpunkt geöffneten Dialoge und Prozeduren, sowie die Position der Werkzeugfenster werden für diesen Benutzer gespeichert. Beim nächsten Öffnen des Designers, wird die Umgebung wieder hergestellt.

Wurde der Advanced-Client (c16_apgi.exe) gestartet, wird der CONZEPT 16-Client beendet. Wurde der Designer über den Menüpunkt **Parameter / Designer** aus dem Standard-Client (c16_winc.exe) heraus gestartet, gelangt der Benutzer in die Umgebung des Standard-Clients zurück.

Kontakt

Menü Bearbeiten

Menü zum Bearbeiten und Markieren von Objekten

Menü

Siehe Datei,

Menü

Anzeige

Die meisten der hier enthaltenen Menüeinträge beziehen sich auf einen markierten Bereich oder auf markierte Objekte.

Einzelne Objekte können durch Anklicken mit der Maus markiert werden. Die Eigenschaften des markierten Objektes werden im Fenster Eigenschaften angezeigt. Handelt es sich bei dem Objekt nicht um ein Form-Objekt, bekommt es einen Rahmen, deren Ecken hervorgehoben sind.

Durch Ziehen mit der Maus kann ein Rahmen erzeugt werden, mit dem mehrere Objekte durch Umschließung markiert werden können. Dabei werden nur Objekte selektiert, die vollständig von dem Rahmen eingeschlossen werden. Eine bestehende Selektion kann durch Drücken der Tasten + und Anklicken eines Objekts erweitert werden. Werden dabei bereits selektierte Objekte angeklickt, werden diese aus der Selektion entfernt.

Der Inhalt eines Container-Objekts (zum Beispiel GroupBox oder NotebookPage) kann erst dann durch Ziehen mit der Maus markiert werden, wenn ein anderes Objekt als der Container selektiert ist. Im Fall, dass der Container selektiert ist, wird das Ziehen mit der Maus als Verschieben des Containers interpretiert.

Alle markierten Objekte werden mit einem blauen oder roten Rahmen versehen. Der rote Rahmen kennzeichnet das erste Objekt der Markierung.

Ein Fenster kann durch Klicken in den Clientbereich markiert werden. Dies ist allerdings nur dann möglich, wenn kein Objekt den gesamten Clientbereich ausfüllt. Ein Fenster kann ebenfalls durch das Anklicken der Titelzeile bei gedrückter -Taste selektiert werden.

Ein Label kann zusammen mit seinem Eingabeobjekt (Eigenschaft ObjLink) durch einen Doppelklick auf das Label markiert werden.

Weitere Möglichkeiten, Objekte zu markieren, sind im folgenden beschrieben.

Kontakt

	Rückgängig: Objekt erstellen	Strg + Z
	Wiederherstellen: Löschen	Strg + Y
	Ausschneiden	Strg + X
	Kopieren	Strg + C
	Einfügen	Strg + V
	Löschen	Entf
	Löschspeicher abrufen	Strg + B
	Alles markieren	Strg + A
	Typisiert markieren	Umschalt + Strg + A
	Unterobjekte markieren	Strg + Q
	Selektion aufheben	Strg + D
	Verschieben	
	Größe ändern	
	Konvertieren zu MdiFrame	

• Rückgängig

Die letzte Änderung an einem Objekt wird rückgängig gemacht. Die Anzahl der Änderungen, die rückgängig gemacht werden können, werden in den Einstellungen eingetragen.

• Wiederherstellen

Werden zuviele Schritte rückgängig gemacht, kann über die Funktion "Wiederherstellen" der Zustand des Objektes wieder hergestellt werden.

• Ausschneiden

Die ausgewählten Elemente im Fenster werden in die Windows-Zwischenablage verschoben.

• Kopieren

Die ausgewählten Elemente im Fenster werden in die Windows-Zwischenablage kopiert.

• Einfügen

Der Inhalt der Windows-Zwischenablage wird in das Fenster eingefügt. Der Inhalt der Zwischenablage bleibt für weiteres Einfügen bestehen.

• Löschen

Die ausgewählten Elemente im Fenster werden gelöscht. Versehentlich gelöschte Elemente können über den Menüeintrag **Bearbeiten / Löschspeicher abrufen** wieder hergestellt werden.

• Löschspeicher abrufen

Kontakt

Versehentlich gelöschte Elemente aus dem Fenster werden mit diesem Menüeintrag wieder hergestellt. Die Größe des Löschspeichers kann im Menüeintrag Datei / Einstellungen verändert werden.

- **Alles markieren**

Alle Elemente im Fenster werden markiert.

- **Typisiert markieren**

Dieser Menüeintrag ist aktiviert, wenn ein Objekt in einem Fenster markiert ist. Über diese Funktion werden alle Objekte des gleichen Types auf der gleichen Objektebene markiert. Es können so zum Beispiel alle Labels innerhalb einer GroupBox selektiert werden.

- **Unterobjekte markieren**

Es werden alle Objekte markiert, die dem ausgewählten untergeordnet sind. Das ausgewählte Objekt wird selbst nicht selektiert. Dieser Menüeintrag ist nur dann aktiviert, wenn ein Objekt selektiert wurde, das untergeordnete Objekte enthalten kann, wie zum Beispiel GroupBox, Frame-Objekt oder NotebookPage.

- **Selektion aufheben**

Alle selektierten Objekte werden deselektiert.

- **Verschieben**

Das selektierte Objekt kann mit den Cursortasten verschoben werden. Das Objekt bewegt sich innerhalb der vom Raster vorgegebenen Punkte und kann nicht aus dem übergeordneten Objekt herausgeschoben werden.

- **Größe ändern**

Die Größe des selektierten Objektes kann mit den Cursortasten verändert werden. Die linke untere Ecke des Objektes wird dabei verschoben. Die Ecke kann nur auf die Punkte des Rasters verschoben werden. Eine Vergrößerung nach außerhalb des übergeordneten Objektes ist nicht möglich.

- **Konvertieren zu Frame**

Der aktuell ausgewählte MdiFrame wird in einen Frame umgewandelt. Der Menüpunkt ist nur verfügbar, wenn ein MdiFrame ausgewählt wurde.

- **Konvertieren zu MdiFrame**

Der aktuell ausgewählte Frame wird in einen MdiFrame umgewandelt. Der Menüpunkt ist nur verfügbar, wenn ein Frame ausgewählt wurde.

Menü Extras

Menü zum Bearbeiten der Datenstruktur sowie Verwalten der Benutzer und Ressourcen

In diesem Menü befinden sich alle Funktionen, um die Datenstruktur zu editieren und die Benutzer und Ressourcen zu verwalten.



- **Datenstruktureditor...**

Über diesen Menüeintrag kann die Datenstruktur der Datenbank verändert werden.

- **Datensatzverwaltung...**

Über diesen Menüeintrag können die Datensätze der Datenbank verändert werden. Zum Aufrufen benötigt der Benutzer die Rechte UrmPermRead und UrmIdePermRead auf der Standard-Elementgruppe _table oder einer Elementgruppe, die mindestens eine Tabelle enthält.

- **BLOB-Verwaltung...**

Über diesen Menüeintrag können die binären Objekte und Verzeichnisse verwaltet werden. Zum Aufrufen benötigt der Benutzer das Recht UrmIdePermRead auf der Standard-Elementgruppe _blob.

- **Benutzerverwaltung...**

Über diesen Menüeintrag können die Benutzer der Datenbank und ihre Berechtigungen verändert werden.

- **Ressourcenverwaltung...**

Über diesen Menüeintrag können Bilder in die Datenbank importiert werden.

- **Anwendungsprotokoll öffnen**

Über diesen Menüeintrag kann das Protokoll des Designers geöffnet werden.

Datenstruktureditor

Erstellen und ändern der Datenstruktur

Menü

Siehe Bearbeiten,

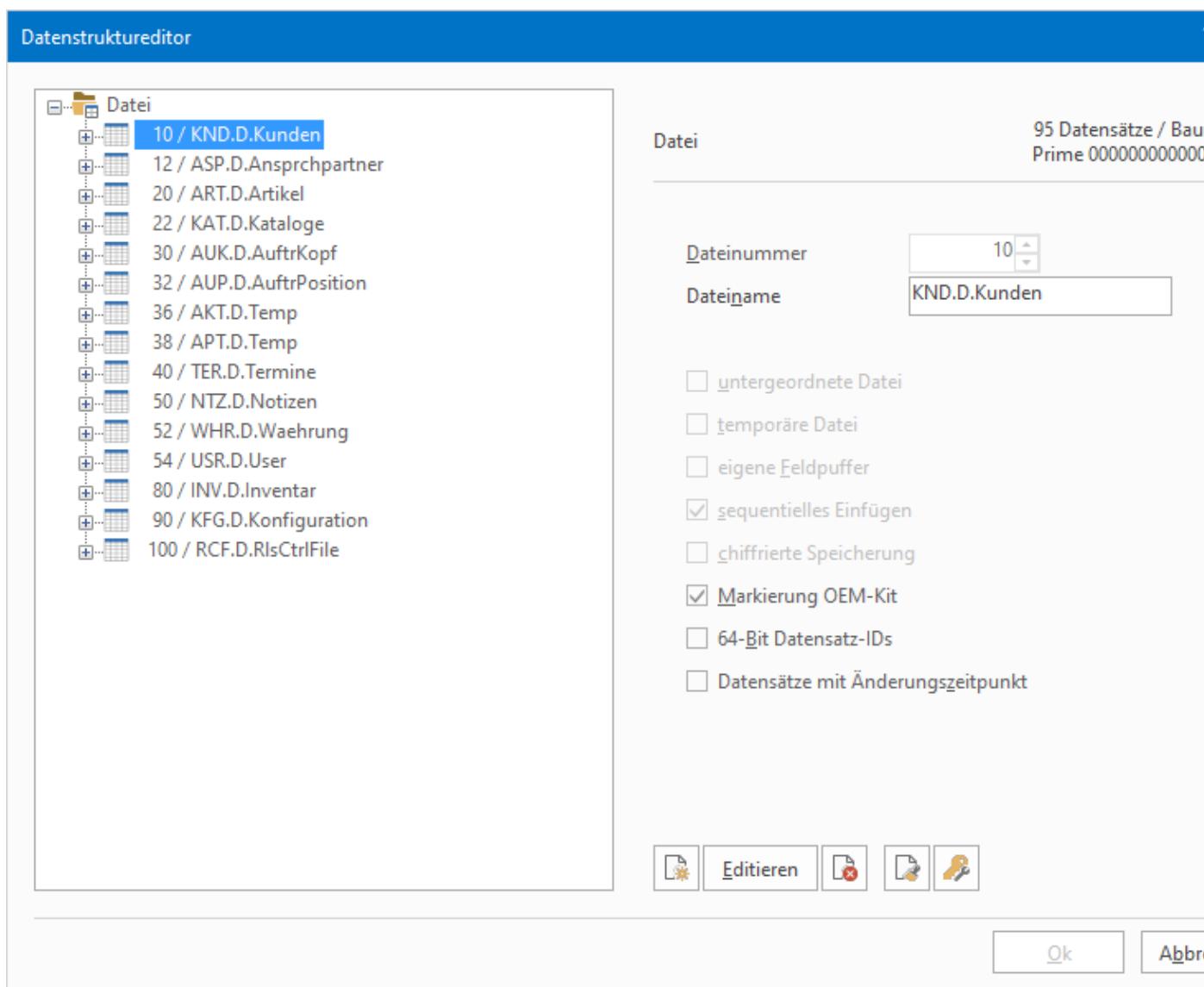
Konventionen

Die Datenstruktur kann über den Menüeintrag **Extras / Datenstruktureditor...** aufgerufen werden.

Bei Änderungen in der Datenstruktur ist zu beachten, dass die neuen Elemente erst nach einem Aufbau der Datenstruktur zur Verfügung stehen. Die Datenstruktur wird durch Schließen des Datenstruktureeditors neu aufgebaut. Die neue Datenstruktur wird dann an jeden Client übertragen, der sich ab diesem Zeitpunkt an der Datenbank anmeldet. Eine Ausnahme bildet der Client, der die Datenstruktur neu aufgebaut hat, er besitzt automatisch die neue Struktur. Damit im laufenden Betrieb die neue Datenstruktur auch von anderen Clients genutzt werden kann, müssen sich diese Clients neu anmelden.

Nach dem Aufruf des Menüeintrages wird folgender Dialog dargestellt:

Kontakt



In der linken Hälfte des Dialoges kann auf die Elemente der Datenstruktur zugegriffen werden. Wird ein Element selektiert, werden in der rechten Hälfte Informationen über das Element dargestellt. Bei Auswahl von "Datei" werden Informationen zur Datenstruktur (Anzahl der Dateien, der Schlüssel, Größe der Datenstruktur usw.) dargestellt. Der Eintrag kann durch Doppelklicken aufgeklappt werden und zeigt dann alle bereits existierenden Dateien an.

Wird in dem Baum die Taste **F3** oder die Tastenkombination **Strg**+**F** gedrückt, kann in einem Eingabefeld ein Name aus der Datenstruktur angegeben werden. Nach Abschließen der Eingabe mit **Enter** wird auf das entsprechende Strukturelement (Datei, Teildatensatz, Feld, Schlüssel usw.) positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten **F3** und **Home**+**F3** wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Datenstruktur gesucht. Ist das Ende bzw. der Anfang der Datenstruktur erreicht, wird die Suche am jeweils anderen Ende

Kontakt

fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Alle Elemente in denen Änderungen vorgenommen wurden, die noch nicht in der Datenstruktur enthalten sind, werden in einem anderen Font und mit einem grünen Symbol gekennzeichnet. Erst nach Neuaufbau der Datenstruktur können diese Elemente von einem neu angemeldetem Client verwendet werden.

Wurden Änderungen an Schlüsseln in Dateien mit Datensätzen vorgenommen, müssen entweder alle Schlüssel oder die betroffenen Schlüssel über die Schaltfläche reorganisiert werden.

Das Erstellen oder Ändern der Datenstruktur unterteilt sich in folgende Bereiche:

- Datei bearbeiten
- Teildatensatz bearbeiten
- Feld bearbeiten
- Schlüssel bearbeiten
- Verknüpfung bearbeiten

Mit dem Schließen des Dialoges mit der Schaltfläche [Übernehmen] wird die Datenstruktur neu aufgebaut. Ab diesem Zeitpunkt wird die neue Datenstruktur an neu angemeldete Clients übertragen.



Um sicher zu gehen, dass alle mit der gleichen Datenstruktur arbeiten, müssen sich alle Clients neu anmelden.

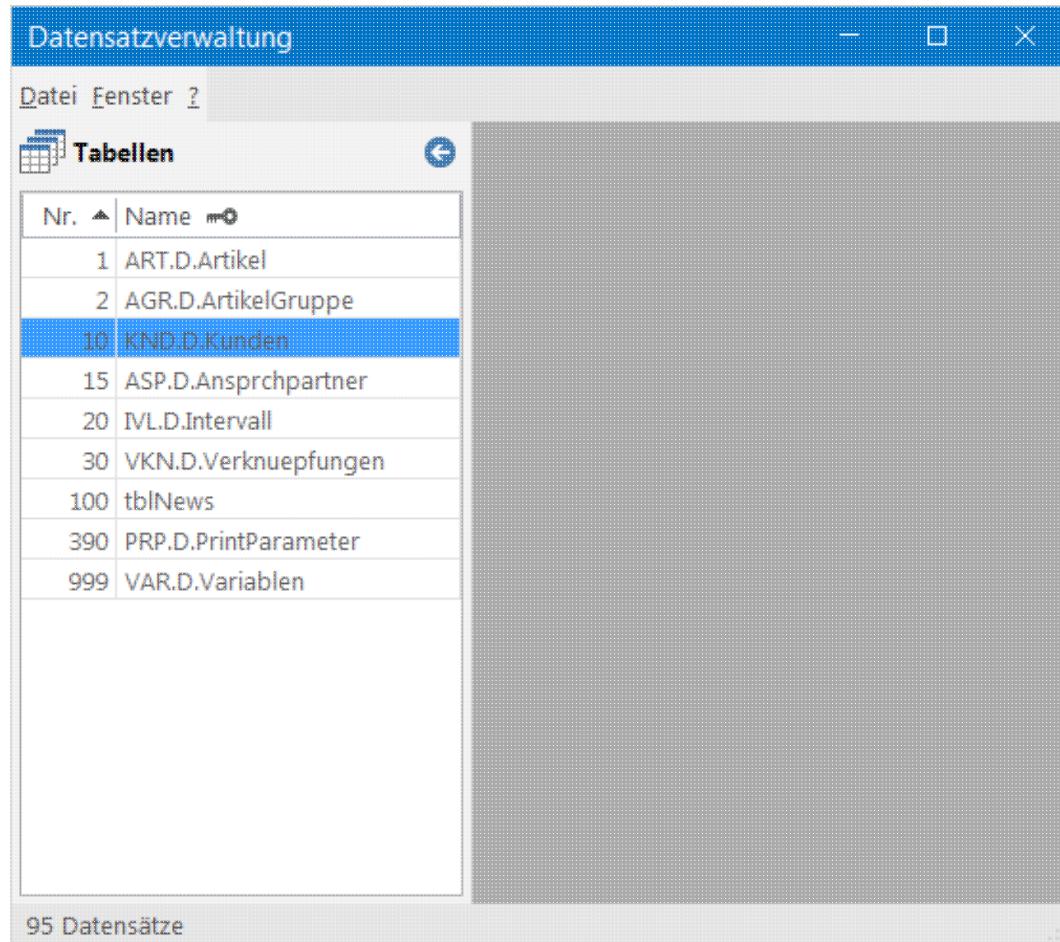
Datensatzverwaltung
Verwaltung von Datensätzen
Siehe [Blog](#)

Über den Menüeintrag **Extras / Datensatzverwaltung...** werden die Tabellen und die dazugehörigen Datensätze der Datenbank angezeigt.

- i** Zum Aufrufen benötigt der Benutzer die Rechte [UrmPermRead](#) und [UrmIdPermRead](#) auf der [Standard-Elementgruppe](#) _table oder einer Elementgruppe, die mindestens eine Tabelle enthält.

Es gibt die Möglichkeit Datensätze anzulegen, zu ändern und zu löschen, sowie Tabellen zu leeren. Weiterhin können direkt die verknüpften Datensätze in einer anderen Tabelle angezeigt werden.

Nach dem Aufruf des Menüpunktes "Extras" / "Datensatzverwaltung..." wird folgender Dialog angezeigt:



Auf der linken Seite werden die Tabellen angezeigt. Die Anzeige kann über die Schaltfläche ausgeblendet und mit der Schaltfläche wieder eingeblendet werden.

In der Liste kann mit **strg** + **F** nach Tabellennamen und Nummern gesucht werden. Mit **strg** + **F3** wird auf den nächsten und mit **strg** + **↑** + **F3** auf den vorherigen Suchtreffer positioniert. Die Groß- / Kleinschreibung wird bei der Suche nicht

Kontakt

beachtet. Die Verwendung von den Wildcards * und ? ist möglich. Standardmäßig wird nach <Suchbegriff>* gesucht.

Durch Doppelklick bzw.  wird der Inhalt der ausgewählten Tabelle angezeigt.

Wird in der Liste der Tabellen mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:



Wird zusätzlich die -Taste gedrückt, erscheint das erweiterte Kontextmenü:



• Suchen

Öffnet die Suche nach Tabellen. Diese Aktion wird ebenfalls über die Tasten  +  ausgelöst.

• Vorwärts weitersuchen

Sucht die nächste Tabelle, die dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die -Taste ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

• Rückwärts weitersuchen

Kontakt

Sucht die vorherige Tabelle, die dem Suchbegriff entspricht. Diese Aktion wird ebenfalls über die Tasten + ausgelöst. Der Menüpunkt wird nur angezeigt, wenn zuvor eine Suche durchgeführt und der Suchbegriff gefunden wurde.

- **Öffnen**

Der Inhalt der Tabelle wird auf der rechten Seite angezeigt. Ist die Tabelle bereits geöffnet, wird das Fenster in den Vordergrund geholt. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

- **Öffnen (neues Fenster)**

Der Inhalt der Tabelle wird auf der rechten Seite in einem neuen Fenster angezeigt. Diese Aktion wird ebenfalls über die Tasten + ausgelöst. Dieser Menüpunkt wird nur angezeigt, wenn das Menü mit gedrückter -Taste geöffnet wurde.

- **Leeren**

Der Inhalt der Datei wird nach Bestätigung einer Abfrage geleert.

- **Datensatz einfügen...**

Ein Fenster mit den Feldern der Tabelle wird angezeigt, in dem ein neuer Datensatz angelegt werden kann.

- **Datensätze exportieren...**

Export von Datensätzen.

- **Datensätze importieren...**

Import von Datensätzen.

Kontakt

BLOB-Verwaltung

Verwalten von binären Objekten und Verzeichnissen

Siehe [Blog](#)

Über den Menüeintrag **Extras / BLOB-Verwaltung...** werden die binären Objekte und Verzeichnisse der Datenbank angezeigt.



Zum Aufrufen benötigt der Benutzer das Recht `UrmIdePermRead` auf der Standard-Elementgruppe _blob.

In der BLOB-Verwaltung können binäre Objekte und Verzeichnisse angelegt, umbenannt, importiert, exportiert, gelöscht, geleert, kopiert und verschoben werden. Weiterhin ist es möglich sich die Eigenschaften eines binären Objektes oder Verzeichnisses anzusehen. Für bekannte Datentypen z. B. Bilder (.png, .jpg, .gif, ...) oder Text (.txt, .rtf, .pdf, ...) gibt es außerdem die Möglichkeit sich die Objekte innerhalb der BLOB-Verwaltung anzeigen zu lassen.

Nach dem Aufruf des Menüpunktes "Extras" / "BLOB-Verwaltung..." wird folgender Dialog angezeigt:

The screenshot shows the 'BLOB-Verwaltung - [Wurzelverzeichnis]' dialog window. On the left, there is a tree view of the root directory structure:

- Wurzelverzeichnis
 - Bilder
 - Customer
 - Documents
 - Ressourcen

On the right, there is a list view of files:

Name	Größe	Zuletzt geändert	MIME-Typ
Bilder			
Customer			
Documents			
Ressourcen			

At the bottom of the dialog, the status bar displays: '0 Objekte (0 Bytes), 4 Verzeichnisse'.

Kontakt

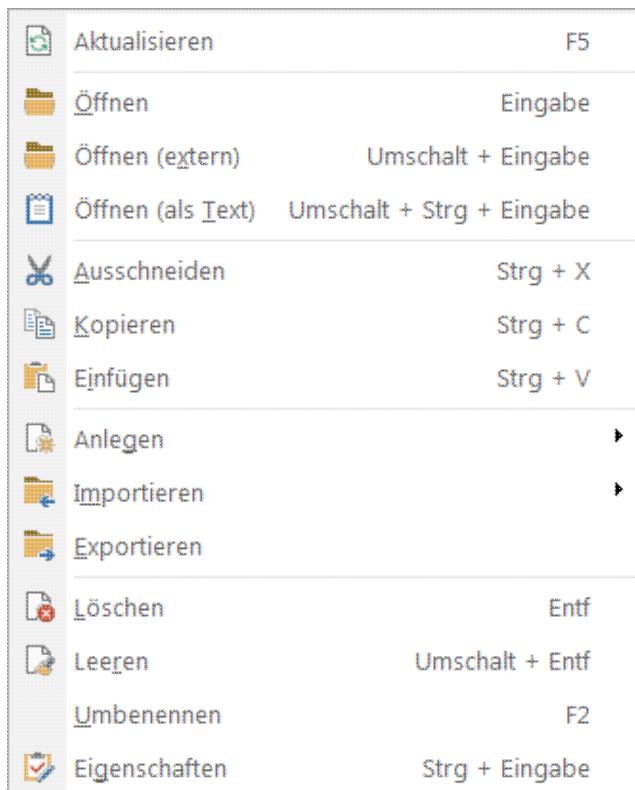
Das Layout der BLOB-Verwaltung orientiert sich am Windows Explorer. Die obere Leiste des Dialogs dient zur Navigation und bietet die Möglichkeit zum vorherigen , zum nächsten und zum übergeordneten Verzeichnis zu springen. Daneben ist eine Adressleiste, die den Pfad des aktuellen Verzeichnisses anzeigt. Über diese kann ebenfalls navigiert werden. In einer Popup-Liste werden Vorschläge angezeigt. Existiert ein angegebenes Verzeichnis nicht, wird eine Fehlermeldung ausgegeben. Ganz rechts in dieser Leiste befindet sich eine Aktualisierungsschaltfläche . Diese baut den Baum und die Liste unterhalb der Navigation neu auf.

Auf der linken Seite des Dialogs ist die Verzeichnisstruktur, der Datenbank, in dem die binären Objekte abgelegt sind, zu sehen. Auf der rechten Seite wird der Inhalt (Verzeichnisse und Objekte) des aktuell ausgewählten Verzeichnisses dargestellt. Die Aktionen werden über das Kontextmenü gestartet.

Das Kontextmenü kann sowohl in dem Baum, als auch in der Liste mit der rechten Maustaste aufgerufen werden. Es erscheint folgendes Menü:

Wird zusätzlich die -Taste gedrückt, erscheint das erweiterte Kontextmenü. Dieses variiert, je nachdem ob man sich auf einem Objekt oder einem Verzeichnis befindet.

Erweitertes Kontextmenü Objekt:



Erweitertes Kontextmenü Verzeichnis:

Kontakt

	Aktualisieren	F5
	Öffnen	Eingabe
	Öffnen (neues Fenster)	Umschalt + Eingabe
	Ausschneiden	Strg + X
	Kopieren	Strg + C
	Einfügen	Strg + V
	Anlegen	▶
	Importieren	▶
	Exportieren	
	Löschen	Entf
	Leeren	Umschalt + Entf
	Umbenennen	F2
	Eigenschaften	Strg + Eingabe

- **Aktualisieren**

Aktualisiert den Baum und die Liste. Diese Aktion wird ebenfalls über die Taste ausgelöst.

- **Öffnen**

Objekt: Es wird eine Vorschau zu dem Objekt angezeigt. Es können nur bekannte Datentypen angezeigt werden. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

Verzeichnis: Das Verzeichnis wird geöffnet. Die Liste neu geladen.

- **Öffnen (extern)**

Das Objekt wird mit dem in Windows definierten Standard-Programm geöffnet.

Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Öffnen (als Text)**

Das Objekt wird intern als Text angezeigt. Nicht darstellbare Zeichen in binär-codierten Dateien werden ignoriert. Diese Aktion wird ebenfalls über die Tasten + + ausgelöst.

- **Öffnen (neues Fenster)**

Es öffnet sich ein neues MDI-Fenster mit dem ausgewählten Verzeichnis. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Ausschneiden**

Schneidet das aktuell selektierte binäre Objekt bzw. Verzeichnis aus. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Kopieren**

Kopiert das aktuell selektierte binäre Objekt bzw. Verzeichnis. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Einfügen**

Fügt das binäre Objekt bzw. Verzeichnis der Zwischenablage in das aktuelle oder derzeit selektierte Verzeichnis ein. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Anlegen**

Legt ein binäres Objekt oder Verzeichnis an. Ist kein Verzeichnis selektiert, wird das Objekt bzw. Verzeichnis im aktuellen Verzeichnis angelegt. Ansonsten im selektierten. Dazu öffnet sich ein Eingabefenster in der ein Name für das binäre Objekt bzw. Verzeichnis definiert werden muss. Diese Aktion wird ebenfalls über die -Taste (für Datei) bzw. die Tasten + (für Verzeichnis) ausgelöst.

- **Importieren**

Importiert eine Datei oder ein Verzeichnis in die Datenbank.

- **Exportieren**

Exportiert das ausgewählte Objekt bzw. Verzeichnis aus der Datenbank.

- **Löschen**

Löscht das ausgewählte Objekt bzw. Verzeichnis. Diese Aktion wird ebenfalls über die -Taste ausgelöst.

- **Leeren**

Objekt: Leert das Objekt, der Inhalt ist danach leer. Das Objekt besitzt dann eine Größe von 0 Bytes.

Verzeichnis: Alle untergeordneten Objekte und Verzeichnisse werden gelöscht.

Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

- **Umbenennen**

Benennt das aktuell ausgewählte Objekt bzw. Verzeichnis um. Diese Aktion wird ebenfalls über die Taste ausgelöst.

- **Eigenschaften**

Zeigt die Eigenschaften zu einem binären Objekt bzw. Verzeichnis an. Diese Aktion wird ebenfalls über die Tasten + ausgelöst.

In der Statusleiste des Dialogs werden Informationen zum Inhalt des aktuellen Verzeichnisses dargestellt. Dabei werden nur die Dateien und Verzeichnisse gezählt, die direkt in dem aktuellen Verzeichnis liegen.

4 Objekte (667,70 KB), 1 Verzeichnis

Kontakt

Die BLOB-Verwaltung unterstützt weiterhin Drag & Drop zwischen MDI-Frames und dem Windows Explorer. Allerdings ist die Anzahl der per Drag & Drop übertragbaren Objekte auf ein Limit von 50 beschränkt.

Benutzerverwaltung

Vergabe und Verwaltung von benutzerabhängigen Rechten

Siehe [Blog](#)

Die Benutzerverwaltung wird im Designer über das Menü **Extras /**

Benutzerverwaltung... aufgerufen. Im folgenden wird das Benutzersystem und der Umgang damit erläutert:

- [Benutzer](#)
- [Benutzergruppen](#)
- [Elementgruppen](#)
- [Zusammenarbeit der Objekte](#)
- [Berechtigungen](#)
- [Verwaltung](#)

Benutzer

Ausgangspunkt eines jeden Benutzersystems ist der Benutzer selbst. In CONZEPT 16 wird der Benutzer durch ein [Benutzer-Objekt](#) repräsentiert. Jedes Benutzer-Objekt verfügt über verschiedene [Eigenschaften](#) (Properties). Die wichtigsten Eigenschaften sind der Name ([UrmPropName](#)) und das Kennwort ([UrmPropPassword](#)), mit denen sich der Anwender an der Datenbank anmeldet. Der Name muss innerhalb einer Datenbank eindeutig sein. Neben den vordefinierten Eigenschaften kann das Benutzer-Objekt um zusätzliche Eigenschaften erweitert werden. Die Applikation kann dadurch zusätzliche Daten im Benutzer-Objekt speichern.

In einer neu erstellten Datenbank sind bereits die beiden Benutzer SUPERUSER und USER mit leerem Kennwort angelegt. Bei der Anmeldung als SUPERUSER reicht die Angabe von SU als Benutzer aus. Der Benutzer SUPERUSER kann nicht gelöscht werden.

Unabhängig vom verwendeten Client ist zum Öffnen einer Datenbank immer die Angabe eines Benutzers erforderlich. Alle Aktionen (Ausführen von Prozeduren, Laden von Dialogen usw.) finden somit immer in einem Benutzerkontext statt.

Die Berechtigungen eines Benutzers ergeben sich aus seiner Zugehörigkeit zu einer oder mehreren [Benutzergruppen](#), im Benutzer-Objekt selbst werden keine Rechte definiert.

Benutzergruppen

Eine Benutzergruppe fasst mehrere Benutzer zusammen, um eine einfachere Vergabe von Rechten zu gewährleisten. Die Berechtigungen werden immer für eine Benutzergruppe definiert, wodurch alle Mitglieder dieser Gruppe die selben Rechte erhalten.

Ein Benutzergruppen-Objekt wird durch seinen Namen identifiziert und verfügt über eine Reihe von vordefinierten [Eigenschaften](#). Genau wie beim Benutzer-Objekt können zusätzliche Eigenschaften angelegt werden. Neben seinen Eigenschaften verfügt das Benutzergruppen-Objekt über Rechte auf mehrere [Elementgruppen](#). Ein Benutzer-Objekt kann Mitglied in mehreren Benutzergruppen sein. Die effektiven Rechte des Benutzers ergeben sich in diesem Fall durch die Addition der Rechte aller

Kontakt

Benutzergruppen, zu denen er gehört.

In einer leeren Datenbank sind bereits die beiden Benutzergruppen _Everyone und _Administrators vorhanden. Die Gruppe _Everyone dient als Basisgruppe für alle Benutzer, um die minimal benötigten Rechte für die Applikation zur Verfügung zu stellen. Die Gruppe _Administrators verfügt automatisch über alle Rechte in der Datenbank. Beide Gruppen können nicht gelöscht werden. Der Benutzer SUPERUSER ist Mitglied von _Administrators und der Benutzer USER Mitglied von _Everyone.

Elementgruppen

Genauso wie Benutzergruppen mehrere Benutzer mit den gleichen Benutzerrollen zusammenfassen, fassen Elementgruppen mehrere Applikationselemente zusammen, für die die selben Rechte benötigt werden. Die Elementgruppe kann sowohl reale Objekte aus der Datenbank (bestimmte Dialoge, einzelne Prozeduren, ein oder mehrere Datensatz-Tabellen usw.) als auch virtuelle Objekte der Applikation (Programm-Module, Funktionalitäten usw.) beinhalten. Durch die Zusammenfassung von mehreren Elementen zu Elementgruppen wird die Rechtevergabe wesentlich übersichtlicher. Gerade wenn es um die Berechtigungen eines Anwendungsbereichs geht, können alle Prozeduren, Dialoge, Tabellen usw. aus diesem Bereich in eine Elementgruppe zusammengefasst werden. Für alle Elemente werden dann gleiche Rechte vergeben. Jedes reale oder virtuelle Element kann Mitglied in bis zu 128 verschiedenen Elementgruppen sein (maximal 32 bei Texten und Prozeduren).

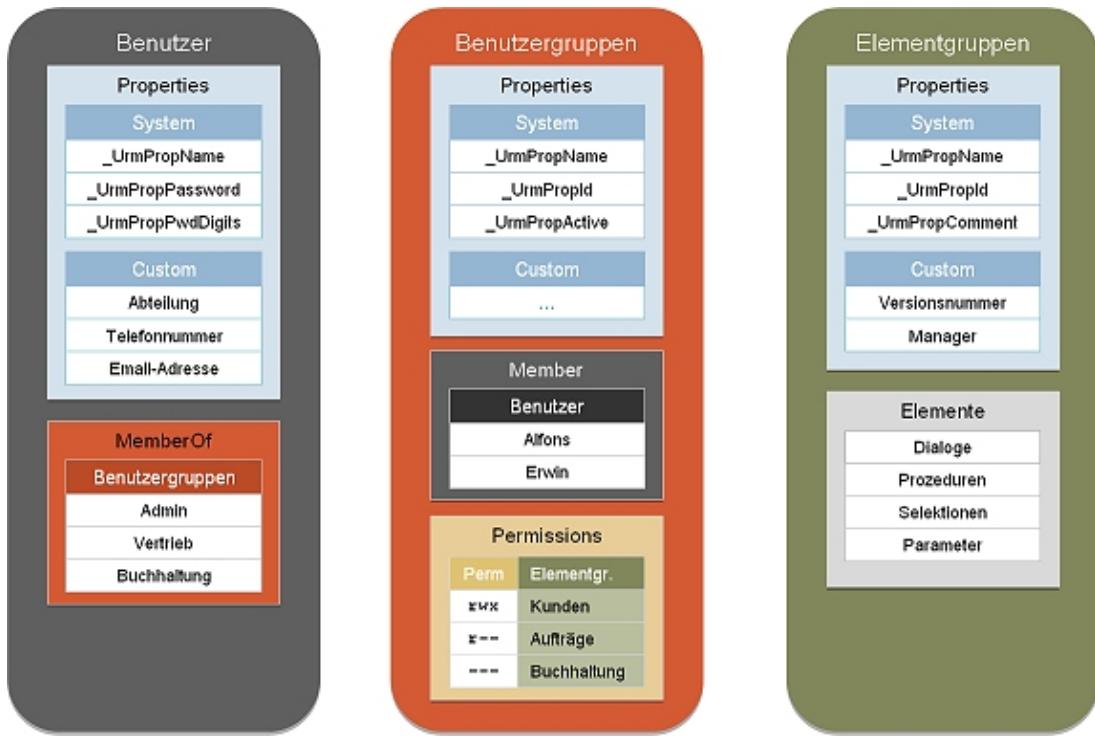
Auch ein Elementgruppen-Objekt wird durch seinen Namen identifiziert und verfügt sowohl über vordefinierte als auch über anwendungsdefinierte Eigenschaften.

In jeder Datenbank sind mehrere sogenannte Standard-Elementgruppen vordefiniert. In diesen Elementgruppen sind jeweils alle Elemente eines bestimmtem Typs enthalten. Zum Beispiel gehören alle Prozeduren automatisch zur Gruppe _procedure. Eine Liste aller Standard-Elementgruppen befindet sich im Abschnitt Verwaltung der Elementgruppen.

Zusammenarbeit der Objekte

Die drei Objekte der Benuterverwaltung treten über Listen miteinander in Beziehung. Innerhalb des Benutzer-Objekts gibt es eine Mitgliedsliste von Benutzergruppen. Bei den Benutzergruppen gibt es eine Liste mit den Mitgliedern (Benutzer-Objekte) und eine Liste mit Elementgruppen einschließlich deren Rechte.

Kontakt



Ob ein Benutzer ein bestimmtes Recht an einem Applikationselement besitzt, hängt also davon ab, ob er Mitglied einer Benutzergruppe ist, die das entsprechende Recht besitzt. Innerhalb der Benutzergruppe muss eine Elementgruppe mit dem entsprechenden Recht eingetragen sein, in der das Element enthalten ist. Ein Recht wird letztendlich in der Benutzergruppe definiert, wo die Elementgruppe eingetragen ist.

Berechtigungen

In CONZEPT 16 werden folgende Berechtigungen verarbeitet:

Read (R)	<u>UrmPermRead</u>	Lesen und Anzeigen
Create (C)	<u>UrmPermCreate</u>	Anlegen
Modify (M)	<u>UrmPermModify</u>	Ändern
Modify Owner (MO)	<u>UrmPermModifyOwner</u>	Ändern bei Objektbesitz
Delete (D)	<u>UrmPermDelete</u>	Löschen
Delete Owner (DO)	<u>UrmPermDeleteOwner</u>	Löschen bei Objektbesitz
Execute (X)	<u>UrmPermExecute</u>	Ausführen
Config (CF)	<u>UrmPermConfig</u>	Konfiguration

Da sich die Rechte jeweils auf die **Elemente** in einer Elementgruppe beziehen, gibt es für die Elementgruppe selbst drei zusätzliche Rechte:

Read Element (RE)	<u>UrmPermElmGroupRead</u>	Berechtigung Elementgruppen lesen
Insert Element (IE)	<u>UrmPermElmGroupInsert</u>	Berechtigung Element zur Gruppe hinzufügen

[UrmPermElmGroupDelete](#)

Kontakt

Delete Element
(DE)

Berechtigung Element aus Gruppe
löschen

Speziell für das Zuordnen von **Benutzern** zu **Benutzergruppen** gibt es die folgenden Berechtigungen:

Insert Member [UrmPermElmGroupRead](#) Berechtigung Benutzer zu Benutzergruppe hinzufügen
(IM)

Delete Member [UrmPermElmGroupInsert](#) Berechtigung Benutzer aus Benutzergruppe entfernen
(DM)

Eine Berechtigung für ein Element entsteht, wenn einer Benutzergruppe eine Elementgruppe hinzugefügt wird und damit gleichzeitig die Rechte auf diese Elementgruppe definiert werden. Ist ein Element in keiner Elementgruppe einer Benutzergruppe enthalten, hat diese Benutzergruppe keine Berechtigung für dieses Element. Die effektiven Rechte für einen Benutzer entstehen durch die Kombination aller Rechte aus den Benutzergruppen, in denen der Benutzer Mitglied ist.

Eine Berechtigung kann erteilt oder entzogen werden. Auf diese Weise können auch "Negativ-Gruppen" erstellt werden, die bestimmte Rechte wieder entfernen. Gibt es für ein Applikationselement sowohl das erteilte, wie das entzogene Recht, setzt sich das entzogene Recht durch.

Beispiel:

In einer Elementgruppe "ElmTblSales" sind alle Tabellen zusammengefasst, die zum Erstellen von Angeboten, Abwickeln von Aufträgen usw. notwendig sind. Darüber hinaus existieren die Elementgruppen "ElmProcCom" und "ElmProcSales" in denen alle Bibliotheken und alle Prozeduren, die zur Abwicklung von Angeboten und Aufträgen benötigt werden, zusammengefasst sind. Es werden drei Benutzergruppen erstellt: "Sales", "Developer" und "Visitor". Folgende Rechte werden vergeben:

	<u>_tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Visitor	M- D- CF-		C- M- D-	C- M- D-
Sales		R+ C+ M+ D+	X+	X+
Developer		R+ C+ M+ D+ X+ CF+	R+ C+ M+ D+ X+	R+ C+ M+ D+ X+

Die Elementgruppe "_tables" wird von CONZEPT 16 zur Verfügung gestellt und enthält alle Tabellen der Datenbank. Die mit einem Minuszeichen versehenen Rechte werden der Benutzergruppe entzogen, die Rechte mit einem Pluszeichen werden erteilt.

Ein Benutzer der Benutzergruppe "Sales" hat innerhalb der relevanten Tabellen alle Rechte, die er benötigt, um Datensätze zu lesen, zu erzeugen, zu ändern und zu löschen. Die notwendigen Prozeduren darf er ausführen. Ein Einsehen der Prozeduren oder gar Ändern und neu übersetzen ist ihm nicht möglich. Da der Benutzer nur in einer Benutzergruppe Mitglied ist, bestehen seine effektiven Rechte auch nur aus den in der Benutzergruppe angegebenen Rechten.

Ein weitere Benutzer wird angelegt. Dieser Benutzer ist nur vorübergehend Gast im Verkauf der Firma. Der Benutzer wird den Benutzergruppen "Sales" und "Visitor" zugeordnet. Seine effektiven Rechte sind durch die Besucher-Gruppe eingeschränkt,

Kontakt

so kann er keine Datensätze Ändern oder löschen. Die entzogenen Rechte in bezug auf die Prozeduren wirken sich nicht weiter aus, der schon die "Sales"-Gruppe hier nur ausführende Rechte besitzt.

Die effektiven Rechte des Besuchers sehen wie folgt aus:

<u>tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Effektiven Rechte	M- D- CF- R+ C+ M- D- CF-	C- M- D- X+	C- M- D- X+
Bekommt die Firma einen weiteren Gast in der Entwicklungsabteilung, so bekommt dessen Benutzer die Mitgliedschaft in den Benutzergruppen "Developer" und "Visitor". Die effektiven Rechte sind dann wie folgt:			

<u>tables</u>	ElmTblSales	ElmProcSales	ElmProcCom
Effektiven Rechte	M- D- CF- R+ C+ M- D- X+ CF-	R+ C- M- D- X+	R+ C- M- D- X+

Verwaltung

Die Verwaltung des Benutzersystems kann über den Designer oder prozedural vorgenommen werden. Die Verwaltung über den Designer gliedert sich in die folgenden Bereiche:

- Verwaltung der Benutzer
- Verwaltung der Benutzergruppen
- Verwaltung der Elementgruppen

Für die prozedurale Steuerung des Benutzersystems werden die folgenden Objekte benötigt:

- User (Benutzer)
- UserGroup (Benutzergruppe)
- ElmGroup (Elementgruppe)

Über die Funktion WinUrmDialog() kann die Benutzerverwaltung in die eigene Anwendung integriert werden.

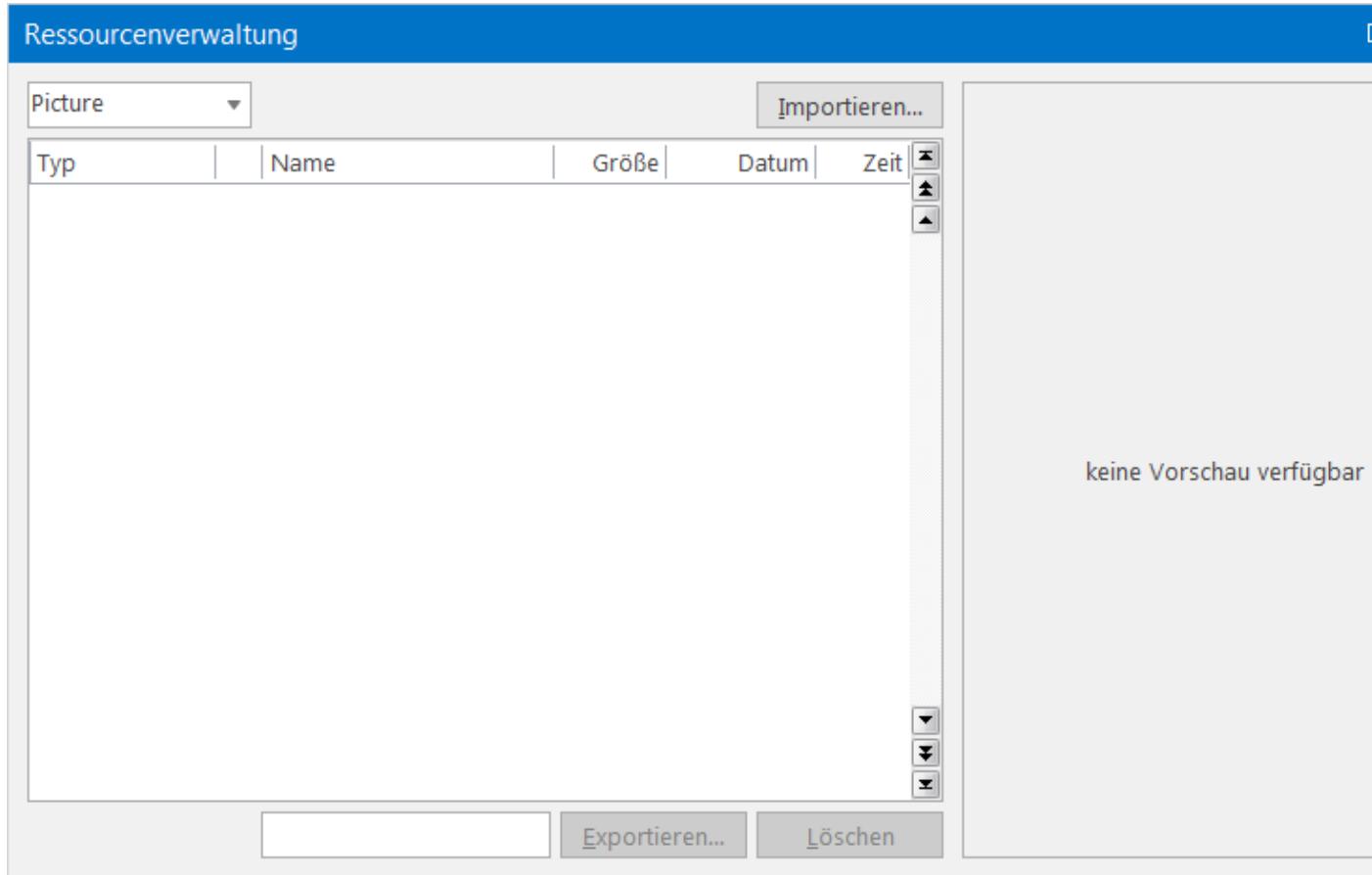
Kontakt

Ressourcenverwaltung

Verwaltung von Ressourcen

Über diesen Menüeintrag werden unterschiedliche Ressourcen, die innerhalb der Datenbank gespeichert wurden, angezeigt. Weiterhin gibt es dort die Möglichkeit weitere Ressourcen zu importieren, vorhandene zu exportieren oder zu löschen.

Nach dem Aufruf des Menüpunktes "Extras" / "Ressourcenverwaltung..." wird folgender Dialog angezeigt:



In dem Dropdown Menü oben links wird die Art der Ressource ausgewählt. Die Liste passt sich an die Auswahl an.

Mit der Schaltfläche Importieren wird eine neue Ressource hinzugefügt.

Über die Schaltflächen Exportieren und Löschen wird die momentan selektierte Ressource exportiert bzw. gelöscht.

Validierungs-Editor

Verwalten von binären Objekten und Verzeichnissen

Siehe [Validierungsbefehle](#)

Über den Menüeintrag **Extras / Validierungs-Editor...** wird der Validierungs-Editor angezeigt. Der Validierungs-Editor erlaubt es eine Validierung für Datenbank-Elemente vorzunehmen.



Zum Aufrufen benötigt der Benutzer das Recht [UrmIdePermRead](#) auf der [Standard-Elementgruppe validation](#).

Nach dem Aufruf des Menüpunktes "Extras" / "Validierungs-Editor..." wird folgender Dialog angezeigt:

Typ	Name	Validierung Status	Validierung Datum	Validierung Uhrzeit	Letzte Änderung Benutzer	Letzte Änderung Datum
	Avi	<input checked="" type="checkbox"/>				
	Beispiel	<input checked="" type="checkbox"/>				
	BinData	<input checked="" type="checkbox"/>				
	BinRename	<input checked="" type="checkbox"/>				
	ChartDesigner	<input checked="" type="checkbox"/>				
	ComDlgMultiSelect	<input checked="" type="checkbox"/>				
	ComError	<input checked="" type="checkbox"/>				
	ComOutlook	<input checked="" type="checkbox"/>				
	ComWord1	<input checked="" type="checkbox"/>				
	ComWord2	<input checked="" type="checkbox"/>				

136 Elemente

Der Validierungs-Editor besteht aus der [Validierungsliste](#) und dem Ribbon mit den Seiten [Filter](#) und [Ansicht](#).

Validierungsliste

Die Validierungsliste zeigt die [Validierungselemente](#) gemäß dem eingestellten Filter an. Liegen keine [Validierungselemente](#) vor, ist die Liste leer und es wird eine Schaltfläche angezeigt, mit der ein Validierungslauf durchgeführt werden kann. Beim Validierungslauf werden [Validierungselemente](#) des aktuell eingestellten Typs (Dialog, Menü, Prozedur oder Tabelle) erstellt und zur Validierungsliste hinzugefügt. Der Vorgang kann je nach Umfang der in der Datenbank enthaltenen Elemente länger dauern, kann aber jederzeit abgebrochen werden. Die Liste enthält dann nur die Elemente, die bis zu diesem Zeitpunkt erstellt wurden.

Die Validierungsliste zeigt folgende Informationen an:

- **Typ**

Hierbei handelt es sich um den Typ des Validierungselementes (Dialog, Menü, Prozedur oder Tabelle).

- **Name**

Der Name des Validierungselementes. Dieser ist beim Zeitpunkt der Erstellung (Validierungslauf) identisch mit dem Name des referenzierten Objektes (Dialog, Menü, Prozedur oder Tabelle).

- **Validierung Status**

Zeigt den aktuellen Validierungsstatus des Elementes an. Folgende Status sind möglich:

Icon	Bedeutung	Beschreibung
	Unbestimmt	Es wurde nicht festgelegt, ob das Element für die Validierung relevant ist oder nicht.
	Nicht relevant	Das Element ist nicht für die Validierung relevant.
	Relevant, jedoch nicht überprüft	Das Element ist zwar relevant für die Validierung, es wurde jedoch noch nicht validiert. Dies ist der Initialzustand nach dem Validierungslauf für neue <u>Validierungselemente</u> .
	Validiert	Das Referenzelement wurde validiert. Die Einträge in den Spalten "Validierung Datum" und "Validierung Uhrzeit" geben den Validierungszeitpunkt wieder.
	Validiert, jedoch modifiziert	Das Referenzelement wurde modifiziert, nachdem es zuvor validiert wurde. Die Zeitpunkte für Validierungselement und Referenzelement können in den Spalten "Validierung Datum", "Validierung Uhrzeit" und "Letzte Änderung Datum", "Letzte Änderung Uhrzeit" abgelesen werden. Der Benutzer, der das Referenzelement geändert hat, wird in der Spalte "Letzte Änderung Benutzer" angezeigt.
		Bei Validierungselementen vom Typ "Tabelle" sind die Spalten leer, da hier diese Informationen nicht vorhanden sind. Anstelle dessen wird hier ein Hash-Wert für die Ermittlung des Status herangezogen.
	Validiert, jedoch gelöscht	Das Referenzelement wurde nach der Validierung gelöscht.

- **Validierung Datum**

Datum, an dem der Status des Elementes auf Überprüft gesetzt wurde.

- **Validierung Uhrzeit**

Uhrzeit, zu der der Status des Elementes auf Überprüft gesetzt wurde.

- **Letzte Änderung Benutzer**

Kontakt

Benutzer, der das Referenzelement zuletzt geändert hat.

- **Letzte Änderung Datum**

Datum, an dem das Referenzelement zuletzt geändert wurde.

- **Letzte Änderung Uhrzeit**

Uhrzeit, zu der das Referenzelement zuletzt geändert wurde.



Die in der Liste angezeigten Informationen (bezüglich der Validierung) geben den Zustand zum Zeitpunkt des letzten Validierungslaufes wieder. Zum Aktualisieren der Informationen ist ein erneuter Validierungslauf (Aktualisierung) erforderlich.

In der Validierungsliste können mehrere Einträge mittels ausgewählt und mittels der Auswahl hinzugefügt oder aus dieser entfernt werden.

Kontextmenü der Validierungsliste

Über das Kontextmenü kann der Validierungsstatus für die selektierten Elemente gesetzt werden.



Der Status für als gelöscht markierte Elemente ist nicht änderbar. Automatisch hinzugefügte Elemente, die in der Datenbank gelöscht wurden, können nicht aus der Liste entfernt werden.

Filter

Hierüber kann die Ansicht der Validierungselemente gefiltert werden. Die Filterung unterteilt sich in die Bereiche "Elemente", "Validierungs-Zeitraum", "Validierungs-Benutzer" und "Custom". Ein Filter ist aktiv, wenn das Häkchen vor dem entsprechenden Filterelement gesetzt ist.

- **Elemente**

Über die obere Liste kann der Element-Typ gefiltert werden. Sofern das Häkchen nicht aktiv ist, bestimmt der ausgewählte Eintrag den Typ der in der Liste angezeigten Validierungselemente. Bei gesetztem Häkchen, können in der Popup-Liste mehrere Typen kombiniert werden (mit). Über die untere Liste kann nach Elementstatus gefiltert werden.

- **Validierungs-Zeitraum**

Hierüber kann der Validierungs-Zeitraum eingeschränkt werden. Die Liste enthält bei aktiverter Filterung nur noch Elemente, die im eingestellten Zeitraum validiert wurden.

- **Validierungs-Benutzer**

Hierüber kann der Validierungs-Benutzer eingeschränkt werden. Die Liste enthält bei aktiverter Filterung nur noch Elemente, die von den eingestellten Benutzern validiert wurden.

- **Custom**

Kontakt

Hierüber kann eine Filterung der Elemente über die Custom-Eigenschaft des Validierungselementes vorgenommen werden.



Die Eigenschaft Custom kann nur prozedural gesetzt werden.

Ansicht



Über die Schaltfläche kann ein Validierungslauf durchgeführt werden.

Anwendungsprotokoll
Log-Datei des Clients
Siehe Log-Viewer

Der Designer erstellt ein Anwendungsprotokoll. Es handelt sich hierbei um eine binäre Log-Datei, welche mit dem Log-Viewer betrachtet werden kann. Optional kann auch beim Starten des Standard- und Advanced-Client das Anwendungsprotokoll gestartet werden. Hierzu muss der Konfigurationseintrag AppLogOptions oder das Kommandozeilenargument C16AppLogOptions den Wert 4 enthalten.

Die Protokolldatei hat den Namen c16_dlgw.lgb. Sofern der aktuell installierte Client verwendet wird, wird sie im Datenverzeichnis des Clients abgelegt, andernfalls wird sie im Pfad des ausführten Clients gespeichert.

-  Bei Netzwerkpfaden, benötigen die Anwender Schreibrechte auf das Client-Verzeichnis, da sonst das Anwendungsprotokoll nicht erstellt werden kann.

Alternativ kann der Pfad des Anwendungsprotokolls wie folgt definiert werden:

1. Als Kommandozeilenargument /C16AppLogPath beim Starten des Standard- bzw. Advanced-Client.
2. In der Konfigurationsdatei des Standard- bzw. Advanced-Client mit dem Eintrag AppLogPath.
3. Als Umgebungsvariable C16AppLogPath.

Zum Öffnen des Protokolls kann im Designer-Menüpunkt "Extras" der Menüpunkt Anwendungsprotokoll öffnen ausgewählt werden. Das Anwendungsprotokoll kann für die Fehleranalyse sinnvoll sein.

-  Beim Öffnen der Log-Datei über den Designer-Menüpunkt wird der Log-Viewer zunächst im Verzeichnis des Clients gesucht. Ist er dort nicht vorhanden, wird der über die Setup-Routine installierte Log-Viewer verwendet. Die Reihenfolge ist wichtig, wenn der Log-Viewer nicht auf der Maschine installiert wurde und der Client aus dem Netzwerk gestartet wird, da ohne Log-Viewer das Anwendungsprotokoll nicht angezeigt werden kann.

Folgende Einträge werden protokolliert:

- **Application started (<Client Typ> <Version> [/ Terminal session])**

Dieser Eintrag wird beim Starten des Designers protokolliert. <Client Typ> gibt an, über welchen Client der Designer gestartet wurde:

- ◆ Client 3.x
- ◆ Standard client
- ◆ Advanced client

<Version> ist das CONZEPT 16-Release des gestarteten Clients.

Der Eintrag "Terminal Session" wird nur protokolliert, wenn der Client in einer Terminal-Session läuft.

Die Spalte "ID" benennt die Benutzer-ID.

• Application stopped

Dieser Eintrag wird beim Beenden des Clients protokolliert.

Die Spalte "ID" benennt die Benutzer-ID.

• System information

Wird unmittelbar nach dem Starten protokolliert und enthält Informationen zum verwendeten Betriebssystem sowie Prozessor und Arbeitsspeicher.

• Exception



Unter Windows XP und Windows 2003 werden keine Einträge zu Exceptions generiert.

Folgende Ausnahmetypen gibt es:

◆ Vom System generierte Ausnahmen (ab Windows Vista)

◊ **Exception <System-Code> / <Code-Text> / code-ptr=<code-addr> / module=(<module>) / version=(<version>) / base=(<base-addr>) [/data=(<data>)]**

Dieser Eintrag wird generiert, wenn der Designer eine unbehandelte Ausnahme generiert, die zum Beenden des Designers führt. Es werden alle Ausnahmen innerhalb der Verarbeitung des Designers protokolliert. Dazu gehören auch Ausnahmen werden der Ausführung von Prozeduren im Designer.

Der Designer zeigt vor der Terminierung ein Hinweisfenster mit folgendem Text an:

"Die Anwendung verursachte einen Ausnahmefehler und muss beendet werden.

Das Anwendungsprotokoll enthält weiterführende Informationen."

<System-Code> gibt die Art der Ausnahme an und <Code-Text> den zugehörigen Fehlertext:

<Code-Text>	Beschreibung	<data> vorhanden?
INT DIVIDE BY ZERO	Integer-Division durch Null	Nein
STACK OVERFLOW	Stapelüberlauf	Nein
ACCESS VIOLATION	Lesender oder schreibender Zugriff auf eine ungültige Speicheradresse	optional
IN PAGE ERROR	Es wurde versucht auf eine nicht vorhandene Seite im Speicher zuzugreifen	optional

<code-addr> gibt die Adresse der Instruktion an, welche zum

Kontakt

Auslösen der Ausnahme führte. <module> ist die ausführbare Datei (DLL oder EXE) in welcher die Ausnahme auftrat. Version liefert (sofern vorhanden) die Version der DLL/EXE und <base-addr> seine Basis-Adresse.

Der optionale Eintrag <data> protokolliert folgende Informationen:

<data>=<access-kind> / addr=<access-addr>

<access-addr> gibt die ungültige Adresse an, auf die versucht wurde zuzugreifen.

<access-kind> gibt die Art des Zugriffs an:

READ Es wurde versucht, die Adresse zu lesen

WRITE Es wurde versucht, die Adresse zu beschreiben

DEP Es liegt eine Datenausführungsverhinderung vor (Data Execution Prevention)

♦ Von der Anwendung generierte Ausnahmen (ab Windows Vista)

◊ **Exception ASSERTION FAILED / file=<file> / line=<line> / proc=<proc> / proc-line=<proc-line> / assertion=<assertion>**

Eine vom Programm angenommene Bedingung ist nicht erfüllt.

<file> und <line> geben Datei und Zeile des Quellcodes wider, an der die angenommene Bedingung auftrat.

<proc> und <proc-line> geben die Zeile und den Name der Prozedur/Funktion in der CONZEPT 16-Laufzeitumgebung wider.
<assertion> ist die fehlgeschlagene Bedingung.



Die Liste der Ausnahmen ist nicht vollständig. Es wurden nur die am häufigsten zu erwartenden Ausnahmen aufgeführt.

• Minidump

Tritt eine Exception auf, wird versucht eine Minidump-Datei zu erstellen. Hierbei können folgende Einträge protokolliert werden:

♦ Minidump created successfully [Path=<Path>]

<Path> gibt an, wo die Minidump-Datei abgelegt wurde.

◊ **Minidump creation failed (<Text>) [Path=<Path>] [Native result=<Native>]**

<Text> gibt Aufschluss über die Operation, die fehlschlug:

◊ **Load debug library failed**

Die Windows Debug-Library (DBGHELP.DLL) konnte nicht geladen werden. <Native> enthält den vom System gelieferten nativen

- Fehlercode.
◊ **Write of minidump failed**

Das Schreiben des Minidump schlug fehl. <Path> enthält den Dateiname, an dem die Datei abgelegt werden sollte. <Native> enthält den nativen Fehlercode.

-  Die Liste der Texte ist nicht vollständig. Es wurden nur die am häufigsten zu erwartenden Texte aufgeführt.

- **Plugin-Schnittstelle**

Folgende Einträge werden von der Plugin-Schnittstelle angelegt:

- ◆ **Plugin manager init : ...**

Die Plugin-Schnittstelle wurde initialisiert. Es werden Informationen zur Datenbank, Benutzer und Port angegeben.

- ◆ **Connect request**

Ein Plugin-Client hat eine Verbindung zum Designer aufgebaut. Der Benutzer muss sich authentifizieren.

- ◆ **... access granted : ...**

Einem Plugin-Client wurde der Zugriff gewährt. Es werden der Plugin-Name, der Datenbankname und der Benutzer angegeben.

- ◆ **... access denied : ...**

Einem Plugin-Client wurde der Zugriff verweigert. Es werden der Plugin-Name, der Datenbankname und der Benutzer angegeben.

- ◆ **... required argument missing : ...**

Bei einem Designer-Befehl wurde ein erwartetes Argument nicht angegeben. Es werden der Plugin-Name und das erwartete Argument angegeben.

- ◆ **... required argument invalid : ...**

Bei einem Designer-Befehl wurde bei einem Argument ein ungültiger Wert verwendet. Es werden der Plugin-Name, das Argument und der ungültige Wert angegeben.

- ◆ **... failed to decode/send command : ...**

Eine Nachricht konnte nicht gesendet oder dekodiert werden. Es werden der Plugin-Name, Befehlsname und Fehlerwert angegeben.

- ◆ **... command reply mismatch : ...**

Antwort passt nicht zum Befehl oder Ereignis. Es werden der Plugin-Name und der Befehlsname angegeben.

- ◆ **... command processing failed : ...**

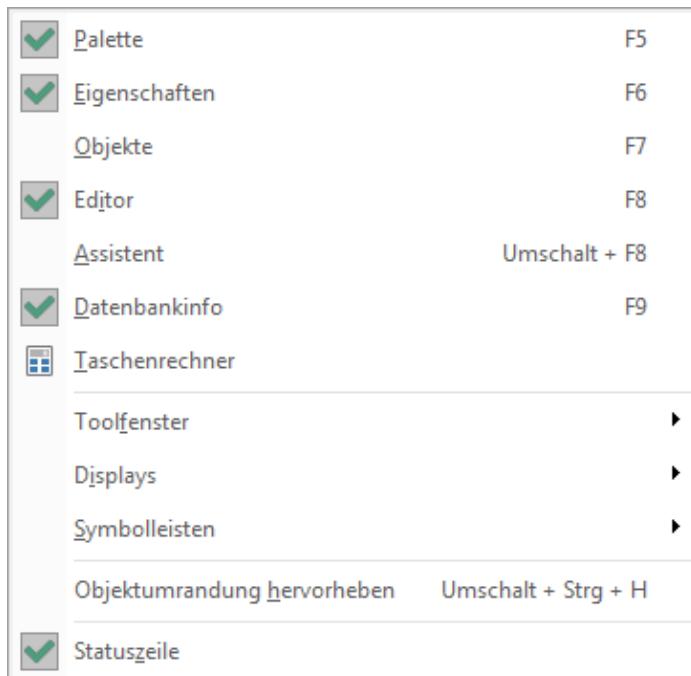
Fehler bei der Ausführung eines Designer-Befehls. Es werden der Plugin-Name, der Fehlerwert und die gesendeten Daten angegeben.

Menü Anzeige

Werkzeugfenster ein- und ausblenden

In diesem Menü können verschiedene Fenster der Entwicklungsumgebung ein- und ausgeblendet werden. Alle Fenster die zur Zeit angezeigt werden, sind mit einem Haken versehen. Die Entwicklungsumgebung wird mit den Fenstern geöffnet, mit denen sie zuletzt geschlossen wurde.

Die in dem Menü angezeigten Funktionstasten haben noch eine zusätzliche Funktion. Befindet sich das Toolfenster nicht im Vordergrund, wird durch Drücken der entsprechenden Funktionstaste das Toolfenster zunächst in den Vordergrund geholt. Durch erneutes Drücken der Funktionstaste wird das Fenster ausgeblendet.



- **Palette**

In diesem Werkzeugfenster sind alle Objekte enthalten, die in ein Fenster integriert werden können.

- **Eigenschaften**

In diesem Fenster werden die Eigenschaften des ausgewählten Objektes dargestellt.

- **Objekte**

Hier wird die Objektstruktur des Fenster dargestellt. Das Werkzeugfenster kann ebenfalls durch eine Schaltfläche im Fenster "Eigenschaften" ein- und ausgeblendet werden.

- **Editor**

Fenster zum Erstellen und Verändern von Dokumenten wie Prozeduren, Funktionen und Texten.

- **Assistent**

Der Assistent dient zur Verwaltung bestehender Dokumente.

- **Datenbankinfo**

Darstellung der Datenstruktur und der Online-Hilfe.

- **Taschenrechner**

Ein Taschenrechner mit Grundrechenarten.

- **Toolfenster**

- ◆ **Links anordnen / Rechts anordnen**

Die eingeblendeten Fenster können über den Menüpunkt **Anzeige / Toolfenster** automatisch auf der linken oder rechten Bildschirmseite angeordnet werden.

- ◆ **Links anordnen - Objektbaum rechts / Rechts anordnen - Objektbaum links**

Der Menüeintrag entspricht "Links bzw. rechts anordnen", jedoch wird zusätzlich der Objektbaum am rechten Bildschirmrand positioniert. Neu angelegte Frame-Objekte erhalten die Position und Größe des freien Platzes zwischen den Toolfenstern.

Der Taschenrechner ist von dieser Funktion nicht betroffen.

- ◆ **Im Arbeitsbereich anordnen**

Alle Toolfenster und das Suchen-Fenster des Editors werden in den Bereich des primären Bildschirms verschoben.

- **Displays**

In der Titelzeile werden Informationen über die Datenbank angezeigt. Weitere Informationen können in der Statuszeile über den Menüeintrag **Anzeige / Display** ein- bzw. ausgeblendet werden (siehe auch Abschnitt Statuszeile).



Hierzu muss ebenfalls der Menüpunkt Statuszeile aktiviert sein.

- ◆ **Benutzer** Der in der Datenbank angemeldete Benutzer wird angezeigt.
- ◆ **Datenbank** Der Alias der Datenbank wird angezeigt.
- ◆ **Server** Der Name oder IP-Adresse des Servers.
- ◆ **Änderungs-/Speicherstatus** Der Änderungs- und Speicherstatus wird angezeigt.
- ◆ **Relative Mausposition** Befindet sich die Maus im Applikationsfenster, wird in der Menüzeile die Position relativ zur linken oberen Ecke des Arbeitsbereiches des Fensters angezeigt.
- ◆ **Absolute Mausposition** Befindet sich die Maus im Applikationsfenster, wird in der Menüzeile die Position relativ zur linken oberen Ecke des Bildschirms angezeigt.
- ◆ **Objekt-Typ** In der Menüleiste wird der Objekt-Typ, über dem die Maus steht angezeigt.

- **Symbolleisten**

Kontakt

Die unter der Menüzeile dargestellten Symbolleisten können nach Bedarf ein- bzw. ausgeblendet werden.

- ◆ **Datei**
- ◆ **Fenster**
- ◆ **Zoom**
- ◆ **Seite**
- ◆ **Lineal**

Über den Eintrag "Anordnen" werden die Symbolleisten wieder in ihrer ursprünglichen Reihenfolge in der Toolbar angeordnet.

• **Objektumrandung hervorheben**

Ist der Menüpunkt "Objektumrandung hervorheben" aktiviert, wird die Umrandung von Container-Objekten (GroupBox, NotebookPage, Scrollbox) mit einem gestrichelten Rahmen hervorgehoben. Hierdurch können die Objektbegrenzungen beim Positionieren, Vergrößern, etc. besser erkannt werden, insbesondere bei kontrastschwachen Windows-Themes. Die Funktion wirkt sich auf alle im Designer geöffneten Frames und PrintForms aus.

Die Farbe der Umrandung kann in den Einstellungen im Bereich Designer über die Auswahl "Hervorhebungsfarbe" angepasst werden.

• **Statuszeile**

In der Statuszeile werden die Displays angezeigt.

Durch Anklicken eines Toolfensters mit der rechten Maustaste in der Titelzeile, wird ein Kontextmenü geöffnet. Durch das Kontextmenü kann das betreffende Fenster verschoben, in der Größe verändert oder geschlossen werden. Mit dem Menüeintrag **Immer im Vordergrund** wird es nicht mehr vom Applikationsfenster verdeckt.

Kontakt

Menü Fenster

Wechseln zwischen verschiedenen Fenstern

In diesem Menü werden alle zur Zeit bearbeiteten Fenster angezeigt. Über die Menüeinträge kann schnell das aktuelle Fenster gewechselt werden.

Kontakt

Menü Hilfe (?)

Aufruf der Online-Hilfe und der Versionsinformationen

Menü

Siehe Fenster,

Anzeige

Objekt

Über das Hilfe-Menü (?) können die Online-Hilfe und die Versionsinformationen aufgerufen werden.

Hilfe

Über diesem Menüpunkt wird die Online-Hilfe aufgerufen.

FAQs

Mit diesem Menüpunkt wird die Hilfe aufgerufen und auf den Bereich der FAQ positioniert. Dort finden sich Antworten auf häufig gestellte Fragen zu unterschiedlichen Themen.

Info

• Programm

Hier finden sich allgemeine Informationen über den CONZEPT 16-Client. Es werden die Version und Links, die direkt zur vectorsoft AG verweisen, angezeigt.

• Bibliothek

In diesem Abschnitt werden die verwendeten Bibliotheken mit ihren Versionen angezeigt. Bei alle Dateinamen, die mit c16_ beginnen handelt es sich um CONZEPT 16-Bibliotheken. Die weiter unten aufgeführten Bibliotheken werden durch das Betriebssystem zur Verfügung gestellt. Von allen Bibliotheken werden die jeweiligen Versionsnummern angezeigt. Bei den CONZEPT 16-Bibliotheken ist darauf zu achten, dass alle Dateien die gleiche Versionsnummer besitzen.

• Datenbank

Hier stehen Informationen über die Datenbank.

Datenraum Pfad Verzeichnis auf dem Datenbank-Server in dem die Datenbank abgelegt ist

Datenraum Name Dateiname der Datenbank

Datenraum Alias Symbolischer Name der Datenbank

Datenraum gesamt Größe des Datenraums in KB

Datenraum frei Freier Speicher innerhalb der Datenbank in KB

Kontakt

Benutzer	Anzahl der angemeldeten Benutzer in der Datenbank
• Server	

Informationen über den Server können auf dieser Seite abgefragt werden.

Name	Name oder IP-Adresse des Datenbankservers
Lizenz	Lizenznummer
Release	Releasenummer des Servers
Benutzerlimit	Maximale Benutzeranzahl in allen Datenbanken

• Parameter

Unter dem Registerreiter "Parameter" werden Informationen zum Speicherort der verwendeten Konfigurationsdatei, sowie zu gesetzten Umgebungsvariablen bzw. Kommandozeilenargumenten angezeigt:

Konfiguration	Speicherort der verwendeten Client-Konfigurationsdatei
C16	Kommandozeilenargument oder Umgebungsvariable zum Auffinden der Konfigurationsdatei
C16CFG	Kommandozeilenargument oder Umgebungsvariable zum Auffinden der CONZEPT 16-Programmdateien
C16LANG	Kommandozeilenargument oder Umgebungsvariable zum Setzen der Anzeigesprache

Konnte die Konfigurationsdatei nicht gelesen werden, wird vor dem Dateinamen "[?]" angezeigt.

• System

In diesem Abschnitt werden Informationen zum System angezeigt.

Betriebssystem	Das Betriebssystem des Clients
Codepage	Die aktive Windows Codepage.
Logische Prozessoren	Anzahl der logischen Prozessoren auf dem Client
Arbeitsspeicher (MB)	In dem Computer verbauter RAM in MB
Speicher Verbrauch (KB)	Zur Zeit verwandelter Speicher in KB
Speicher Peak (KB)	Maximal verwandelter Speicher in KB
Bildschirm-Auflösung	Breite, Höhe und Farbtiefe des Bildschirms
Verbrauch	Aktueller Verbrauch der Benutzer-Objekte /
Benutzer-Objekte	Beschränkung von Windows für die Anzahl der Benutzer-Objekte (Aktueller Verbrauch in Prozent)
Verbrauch GDI-Objekte	Aktueller Verbrauch der GDI-Objekte / Beschränkung von Windows für die Anzahl der GDI-Objekte (Aktueller Verbrauch in Prozent)

Die Benutzer- und GDI-Objekte sind im Zusammenhang mit dem Laden von Fenster-Objekten wichtig. Zusätzliche Informationen befinden sich bei den

Kontakt

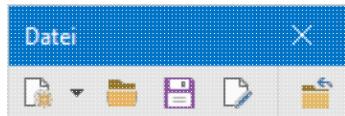
Anweisungen WinOpen(), WinDialog() und WinAddByName() .

Symbolleisten

Symbolleisten der Entwicklungsumgebung

Folgende Symbolleisten stehen zur Verfügung:

Symbolleiste "Datei"



- **Neu**

Im linken Bereich des Buttons wird ein Frame erstellt, im rechten Bereich wird ein Menü zum Anlegen von Objekten geöffnet.

- **Öffnen**

Es wird ein Dialog zum Laden von Objekten geöffnet.

- **Speichern**

Es wird ein Dialog zum Speichern von Objekten geöffnet.

- **Bearbeitungsmodus**

Der Bearbeitungsmodus des zur Zeit bearbeiteten Fensters wird zwischen dem "Nur lesen"- und dem Lese-/Schreibmodus umgeschaltet.

- **Schließen**

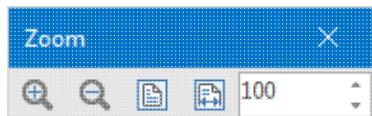
Das zur Zeit bearbeitete Fenster wird geschlossen.

Symbolleiste "Fenster"



In dieser Symbolleiste werden alle geöffneten Fenster als Button dargestellt. Durch das Anklicken eines Buttons wird das jeweilige Fenster in den Vordergrund geholt.

Symbolleiste "Zoom"



- **Ansicht vergrößern**

Das Druck-Formular wird vergrößert.

- **Ansicht verkleinern**

Das Druck-Formular wird verkleinert.

- **Ganze Seite**

Kontakt

Das Druck-Formular wird so skaliert, dass die gesamte Seite zu sehen ist.

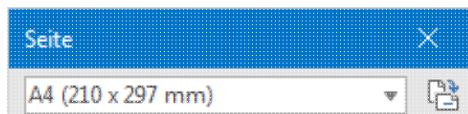
- **Seitenbreite**

Das Druck-Formular wird so skaliert, dass die Seitenbreite der Fensterbreite entspricht.

- **Zoomfaktor**

Einstellen des Zoomfaktors. Es kann ein Zoomfaktor von 25% bis 500% eingestellt werden.

Symbolleiste "Seite"



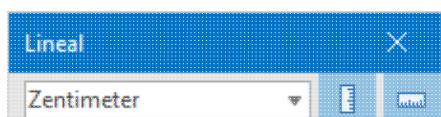
- **Seitenformat**

Über eine Combobox kann das gewünschte Seitenformat ausgewählt werden.

- **Hoch- oder Querformat**

Darstellung der Seite im Hoch- oder Querformat.

Symbolleiste "Lineal"



- **Maßeinheit**

Über eine Combobox kann ausgewählt werden, welche Maßeinheit (Zentimeter oder Inch) das Lineal verwendet.

- **Lineal links**

Über diesen Button wird das Lineal an der linken Seite des Druck-Objekts eingeschaltet.

- **Lineal oben**

Über diesen Button wird das Lineal an der oberen Seite des Druck-Objekts eingeschaltet.

Statuszeile

Statuszeile der Entwicklungsumgebung

In der Statuszeile sind Statusinformationen des Designers enthalten.



• **Zustand**

In diesem Bereich wird der Zustand des bearbeiteten Fensters angezeigt.

• **Server**

Hier wird der Name oder IP-Adresse des Servers angezeigt. Wurde beim Starten des Clients *, localhost oder 127.0.0.1 angegeben, steht hier Local.

• **Alias der Datenbank**

Hier wird der Alias der Datenbank angezeigt.

• **Benutzer**

Der Name des angemeldeten Benutzers wird im letzten Bereich dargestellt.

• **Mausposition absolut**

Beim ersten Starten der Entwicklungsumgebung ist die Anzeige nicht aktiv. Hier wird die Position des Mauszeigers relativ zur oberen linken Ecke des Bildschirms angezeigt.

• **Mausposition relativ**

Beim ersten Starten der Entwicklungsumgebung ist diese Anzeige nicht aktiv. In diesem Bereich wird die Position des Mauszeigers relativ zur oberen linken Ecke des aktuell bearbeiteten Fensters angezeigt.

• **Objekt**

Beim ersten Starten der Entwicklungsumgebung ist die Anzeige nicht aktiv. Sie zeigt den Typ des ausgewählten Objektes an.

Dialogzustand

Anzeige des Dialogzustandes

Siehe Statuszeile

Der Zustand des Dialoges wird mit folgenden Symbolen dargestellt:

Der gerade bearbeitete Dialog ist neu angelegt oder importiert worden. Er ist nicht in der Datenbank gespeichert.

-  Der gerade bearbeitete Dialog ist in der Datenbank geöffnet worden. Der Dialog ist somit für andere Benutzer der Datenbank gesperrt. Der Dialog wurde nicht verändert.

Der Dialog ist gegenüber der in der Datenbank gespeicherten Version verändert worden.

Der Dialog ist "nur-lesend" geöffnet worden.

Die Anzeige des Status wird über den Menüpunkt **Anzeige / Displays / Änderungs-/Speicherstatus** aktiviert oder deaktiviert.

Mausposition absolut

Anzeige der absoluten Mausposition

Anzeige

Objekt,

Siehe Anzeige

Mausposition

(relativ)

Die Anzeige der Mausposition wird über den Menüpunkt **Anzeige / Displays / absolute Mausposition** aktiviert oder deaktiviert.

In dem Bereich wird die Position des Mauszeigers relativ zur oberen linken Bildschirmecke angezeigt. Die Anzeige ist leer, wenn sich der Mauszeiger nicht im aktuellen Fenster befindet.

Mausposition relativ

Anzeige der relativen Mausposition

Anzeige

Mausposition

Siehe (absolut),

Anzeige

Zustand

Die Anzeige der Mausposition wird über den Menüpunkt **Anzeige / Displays / relative Mausposition** aktiviert oder deaktiviert.

In dem Bereich wird die Position des Mauszeigers relativ zur oberen linken Ecke des Arbeitsbereiches angezeigt. Die Anzeige ist leer, wenn sich der Mauszeiger nicht im Arbeitsbereich des aktuellen Fensters befindet.

Kontakt

Objekt-Typ

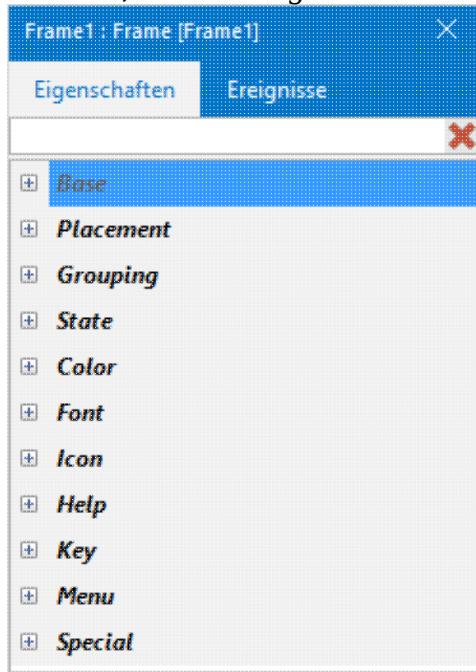
Typ eines Objekts anzeigen

In einem Abschnitt der Menüzeile wird der Typ des Objekts angezeigt, über dem der Mauszeiger steht. Die Anzeige des Objekt-Typen kann über das Menü Anzeige aktiviert und deaktiviert werden.

Kontakt

Designer - Eigenschaften

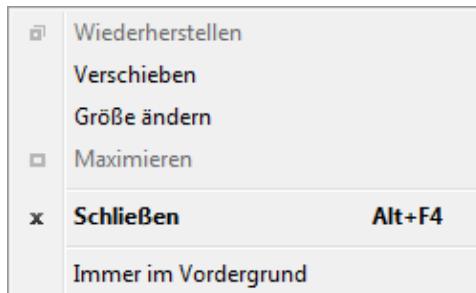
Fenster, das die Eigenschaften und Ereignisse des selektierten Objekts anzeigt.



In diesem Fenster werden die Eigenschaften und die Ereignisse des markierten Objekts dargestellt. Das selektierte Objekt wird in der Titelzeile mit Name und Caption ausgegeben. Die Anzeige kann über die Registerreiter "Eigenschaften" und "Ereignisse" gewechselt werden. Ist kein Objekt markiert, sind die Registerreiter ausgegraut und die Liste leer.

Werte von Eigenschaften und Ereignissen können über die Zwischenablage kopiert werden. Dazu muss die Eigenschaft nicht im Editiermodus sein. Durch Markierung der Eigenschaft und Drücken der Tastenkombination + wird der Inhalt der Eigenschaft in die Zwischenablage kopiert. Ein Ausschneiden des Eigenschaftswertes ist nicht möglich.

Das Fenster verfügt über ein Kontextmenü, das durch Klicken mit der rechten Maustaste auf den Fenstertitel aufgerufen werden kann.



• Verschieben

Über diesem Menüpunkt kann das Fenster verschoben werden. Das Fenster kann ebenfalls durch Ziehen der Titelzeile verschoben werden. Die

Kontakt

ursprüngliche Position kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

- **Größe ändern**

Mit der Maus kann nur die Länge des Fensters verändert werden. Mit dem Menüpunkt kann ebenfalls die Breite geändert werden. Die ursprüngliche Größe kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

- **Schließen**

Dieser Menüeintrag hat die gleiche Wirkung, wie die Schließen-Schaltfläche des Eigenschaftsfenster. Das Fenster kann jederzeit durch den Menüeintrag **Anzeige / Eigenschaften** in der Menüzeile wieder aktivieren.

- **Immer im Vordergrund**

Ist dieser Menüeintrag mit einem Haken versehen, kann das Eigenschaftsfenster nicht von einem anderen Fenster verdeckt werden. Durch Anklicken des Menüintrages kann die Anzeige "Immer im Vordergrund" aktiviert und deaktiviert werden.

Eigenschaften

Die Liste der Eigenschaften teilt sich in zwei Spalten. In der Spalte Beschreibung sind die Namen der Eigenschaften angegeben. In der Spalte Inhalt die zu den Eigenschaften gesetzten Werte.

Eigenschaften, die rot oder gelb dargestellt sind, können nicht geändert werden. Rot sind Eigenschaften, die entweder nur gelesen werden können oder über andere Funktionen geändert werden. Der Name eines Fensters wird beim Speichern des Objekts in der Datenbank vergeben, daher ist der Name in der Eigenschaftsliste rot. Alle gelben Eigenschaften stehen erst zur Laufzeit zur Verfügung. Sie sind der Vollständigkeit wegen in der Eigenschaftenliste enthalten, können aber zum Zeitpunkt des Entwurfs nicht geändert werden.

Eigenschaften können per Drag & Drop direkt in den Editor eingefügt werden, indem bei gedrückter -Taste die Eigenschaft in das Dokument gezogen wird.

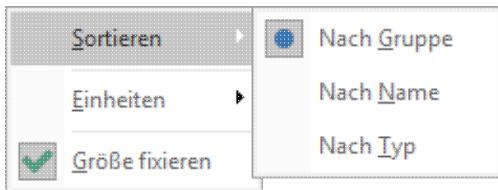
Die Liste der Eigenschaften kann über das Eingabefeld am oberen Rand durchsucht werden. In dem Feld kann der Name einer Eigenschaft eingegeben werden, um danach zu filtern. Die Suche ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Suchbegriff>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

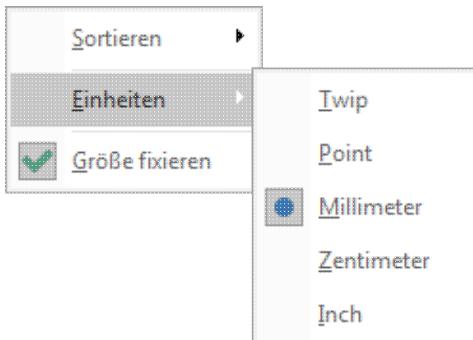
Innerhalb der Liste kann mit der rechten Maustaste ein Kontextmenü aufgerufen werden. In diesem Kontextmenü kann die Sortierung der Eigenschaften und die Darstellung der Position des Objektes verändert werden. Ist "Größe fixieren" aktiviert, wird beim Ändern der Eigenschaften AreaLeft und AreaTop die Größe des Objektes beibehalten und somit die Eigenschaften AreaRight und AreaBottom verändert. Ist es

Kontakt

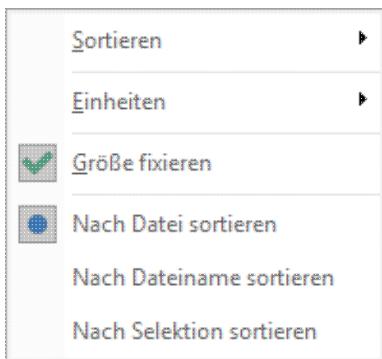
nicht aktiviert werden statt dessen die Eigenschaften AreaWidth und AreaHeight verändert. Die Ersetzung findet auch beim Doppelklick auf den Namen einer der vier Eigenschaften statt.



In dem Kontextmenüpunkt Einheiten können MaÙeinheiten für Druck-Objekte definiert werden. Alle Angaben zur Position, Größe usw. des jeweiligen Druck-Objektes werden in der eingestellten Einheit vorgenommen.



Bei der Eigenschaft DbRecBuf kommt folgendes Kontextmenü:



- **Nach Datei sortieren**

Die Einträge in der Liste der Eigenschaft werden nach Dateinummer sortiert.

- **Nach Dateiname sortieren**

Die Einträge in der Liste der Eigenschaft werden nach Dateiname sortiert.

- **Nach Selektion sortieren**

Die Einträge in der Liste der Eigenschaft werden nach selektierten (angehakten) Einträgen sortiert.

Bei der Eigenschaft DbVar kommt folgendes Kontextmenü:



- **Nach Variablenname sortieren**

Die Einträge in der Liste der Eigenschaft werden nach Variablenname sortiert.

- **Nach Selektion sortieren**

Die Einträge in der Liste der Eigenschaft werden nach selektierten (angehakten) Einträgen sortiert.

Werden mehrere Objekte selektiert, können die Eigenschaften, die für alle ausgewählten Objekte existieren, für alle Objekte geändert werden. Eigenschaften, die nicht bei allen Objekten den gleichen Wert haben, werden ausgegraut. Sie enthalten den Wert vom Ausgangsobjekt. Das Ausgangsobjekt ist im Dialog mit einem roten Rahmen versehen.

Die angezeigten Eigenschaften sind vom ausgewählten Objekt abhängig. Eine Liste aller Eigenschaften befindet sich im Abschnitt Liste aller Eigenschaften. Ein Objekt kann durch Anklicken mit der Maus selektiert werden. Mit bzw. + kann das Objekt gewechselt werden ohne den Eingabefokus aus dem Eigenschaftsfenster zu entfernen. Wird dabei zusätzlich die -Taste gedrückt, wird nicht in untergeordnete Objekte eines Container-Objektes (zum Beispiel einer GroupBox) gewechselt.

Der Inhalt einer bestimmten Eigenschaft kann durch Anklicken des entsprechenden Feldes neben dem Namen der Eigenschaft geändert werden. Je nach Typ der Eigenschaft stehen dazu verschiedene Controls zur Verfügung:

- **Alphanumerische Werte**



Alphanumerische Werte können ohne weitere Besonderheiten in das Feld eingetragen werden. Eine Eigenschaft kann bis zu 8192 Zeichen aufnehmen.

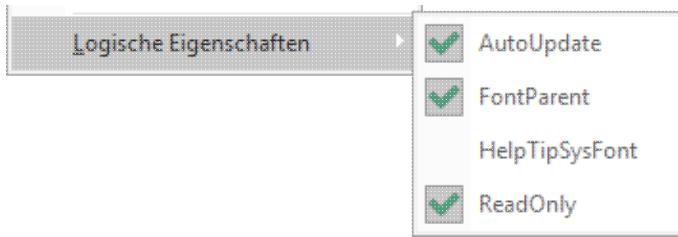
- **Numerische Werte**



Numerische Daten können entweder direkt über die Tastatur oder über den Rollknopf eingetragen werden.

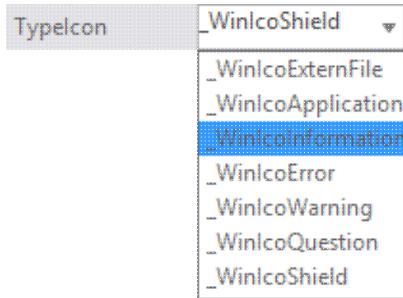
- **Logische Werte**

Kontakt



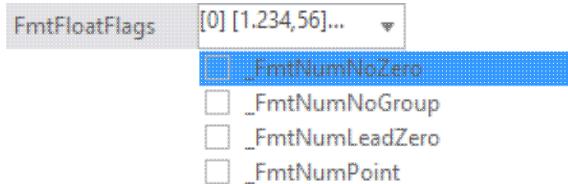
Logische Werte können auch über das Kontextmenü des Objekts angezeigt und geändert werden. Alle logischen Eigenschaften werden an das Ende des Kontextmenüs angehängt. Gesetzte Eigenschaften werden durch einen Haken vor der Eigenschaft angezeigt.

• Konstanten



Kann in einem Feld nur eine Konstante eingetragen werden, erfolgt die Eingabe entweder durch Aufklappen der ComboBox und Auswahl eines Eintrags oder durch einen Doppelklick auf den Namen der Eigenschaft. In diesem Fall werden die Auswahlmöglichkeiten zyklisch zugewiesen. Dies betrifft im Besonderen die Eigenschaften, die nur einen Wahrheitswert aufnehmen können. Hier wird dann abwechselnd true und false zugewiesen.

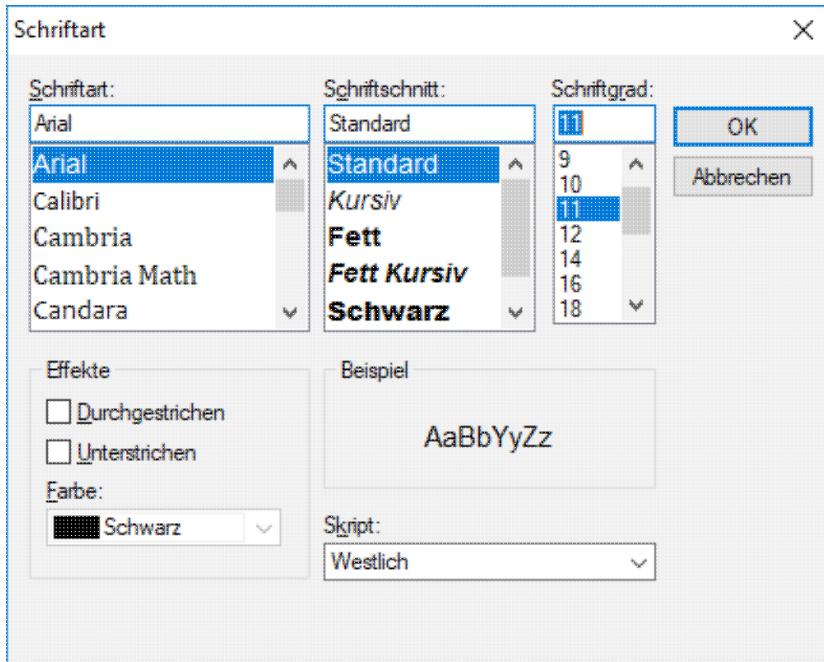
• Mehrere Konstanten



Können in einer Eigenschaft mehrere Konstanten miteinander kombiniert werden, erfolgt die Auswahl der Konstanten über Checkboxen. Diese Checkboxen können durch einen Doppelklick aktiviert oder deaktiviert werden.

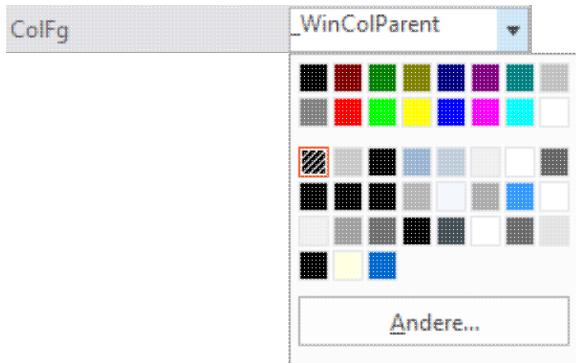
• Zeichensatz

Kontakt



Wird die Eigenschaft Font angeklickt, erscheint ein Auswahldialog mit entsprechenden Einstellungen für die Schriftart, Schriftschnitt und Schriftgröße. Im Eintrag "Skript" kann ein Schriftskript gewählt werden, um mehrsprachige Dialoge zu ermöglichen.

- **Farbe**



Bei Farbeinstellungen können aus der aufgeklappten ComboBox entweder eine der 16 voreingestellten Farben oder eine der darunter aufgeführten Systemfarben verwendet werden. Die Systemfarben werden aus den Darstellungseigenschaften des Bildschirms ermittelt. Sollte diese Farbauswahl nicht ausreichen, kann eine individuelle Farbe über die Schaltfläche [Andere...] zusammengestellt werden.

- **Datei-, Schlüssel- und Verknüpfungsnummer**

DbFileNo	
AGR.D.ArtikelGruppe	2
ART.D.Artikel	1
ASP.D.Ansprchpartner	15
BKV.D.BspKatVerkn	320
BSP.D.Beispiele	300
FKT.D.Funktionen	340
IVL.D.Intervall	20
KAT.D.Kategorien	310
KFG.D.Konfiguration	380
KND.D.Kunden	10

Einige Eigenschaften beziehen sich auf Objekte in der Datenbank (zum Beispiel MenuItem oder TileNameNormal). In diesen Fällen kann durch Doppelklick auf den Namen der Eigenschaft ein Dialog geöffnet werden, der die Auswahl des entsprechenden Objektes erlaubt. Der Name des Objektes wird in die Eigenschaft eingetragen. Ist der Name bereits bekannt, kann er auch direkt eingetragen werden.

Eine Liste aller Eigenschaften befindet sich im Abschnitt Liste aller Eigenschaften.

Ereignisse

Jedes Objekt verfügt über eine Reihe von Ereignissen, die bei bestimmten Aktionen des Benutzers oder des Systems ausgelöst werden.

Die Liste der Ereignisse teilt sich in zwei Spalten. In der Spalte "Beschreibung" sind die Namen der Ereignisse angegeben. In der Spalte "Prozedur" wird die Prozedur oder Funktion angegeben, die beim Eintreten des Ereignisses ausgeführt wird.

Eine Ereignisfunktion kann durch Ziehen des Ereignisnamens in den Editor mit gedrückter -Taste in das Ereignis eingetragen werden. Im Editor wird der vollständige Funktionskopf mit einem leeren Funktionskörper eingefügt.

Ist bei einem Ereignis eine Funktion hinterlegt, kann diese durch Doppelklicken der Beschreibung im Editor geöffnet werden. Dabei wird die eingetragene Prozedur geladen und der Cursor auf die Funktion positioniert.

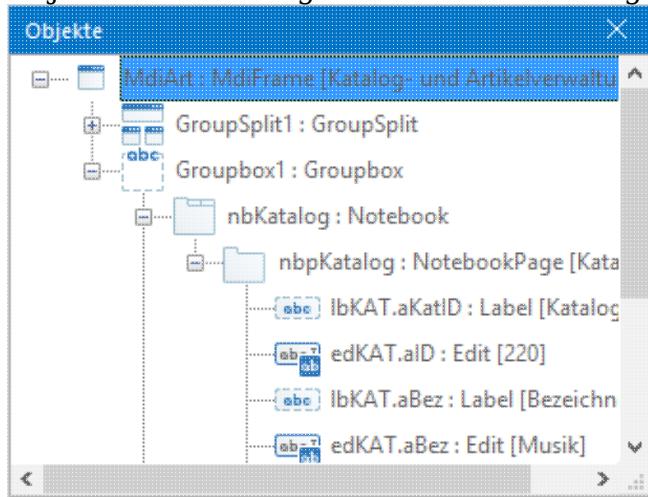
Innerhalb der Liste kann mit der rechten Maustaste ein Kontextmenü aufgerufen werden. In diesem Kontextmenü kann die Sortierung der Ereignisse verändert werden.

Auch die Liste der Ereignisse kann über das Eingabefeld am oberen Rand durchsucht werden. Das Verhalten ist das gleiche, wie bei den Eigenschaften.

Eine Liste aller Ereignisse befindet sich im Abschnitt Liste aller Ereignisse.

Designer - Objekte

Objektstruktur der geöffneten Anwendungsfenster.



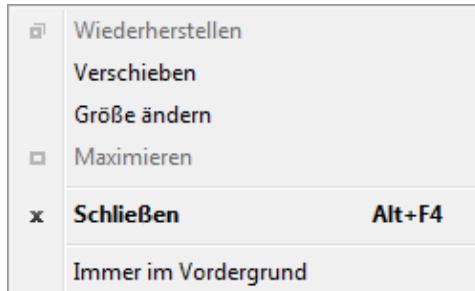
In diesem Fenster ist die Objektstruktur aller geladenen Editierfenster aufgeführt. Die Anzeige ist wie folgt aufgebaut: "<Symbol> <Objektname> : <Objekttyp> [<Caption des Objektes>]".

Über ein Kontextmenü lassen sich alle untergeordneten Knoten auf- und zusammenklappen. Zusätzlich kann der komplette Baum zusammengeklappt werden.



Das Kontextmenü enthält zudem alle Menüpunkte des Objektspezifischen Kontextmenüs.

Zusätzlich verfügt das Fenster über ein Kontextmenü, das durch Klicken mit der rechten Maustaste auf den Fenstertitel aufgerufen werden kann.



• Verschieben

Über diesem Menüeintrag kann das Fenster verschoben werden. Das Fenster kann ebenfalls durch Ziehen der Titelzeile verschoben werden. Die ursprüngliche Position kann über den Menüeintrag **Anzeige / Toolfenster**

wieder hergestellt werden.

- **Größe ändern**

Mit der Maus kann nur die Länge des Fensters verändert werden. Mit dem Menüeintrag kann ebenfalls die Breite geändert werden. Die ursprüngliche Größe kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

- **Schließen**

Dieser Menüeintrag hat die gleiche Wirkung, wie die Schließen-Schaltfläche des Eigenschaftsfensters. Das Fenster kann jederzeit durch den Menüeintrag **Anzeige / Eigenschaften** in der Menüzeile wieder aktiviert werden.

- **Immer im Vordergrund**

Ist dieser Menüeintrag mit einem Haken versehen, kann das Eigenschaftsfenster nicht von einem anderen Fenster verdeckt werden. Durch Anklicken des Menüintrages kann die Anzeige "Immer im Vordergrund" aktiviert und deaktiviert werden.

Anzeigen von untergeordneten Objekten

Sind einem Objekt weitere Objekte untergeordnet, zum Beispiel beim Objekt Groupbox, wird vor dem Objektnamen ein Plus dargestellt. Durch Anklicken des Plus-Zeichens werden die untergeordneten Objekte eingeblendet. Durch Anklicken des jetzt angezeigten Minus-Zeichens können die untergeordneten Objekte wieder ausgeblendet werden.

Objekte kopieren

Ein Objekt kann in diesem Fenster kopiert werden. Dazu wird das Objekt bei gedrückter -Taste mit der Maus an eine andere Stelle im Objektbaum gezogen. Das kopierte Objekt befindet sich an der gleichen Fensterposition und kann im Anwendungsfenster verschoben werden.

Objekte löschen

Ein Objekt kann in diesem Fenster markiert und anschließend über den Menüpunkt **Bearbeiten / löschen** oder durch Drücken der -Taste entfernt werden.

Objektreihenfolge ändern

Alle Objekte werden in der Reihenfolge ihrer Erstellung im Dialog angesprungen (zum Beispiel bei einem Feldwechsel mit der -Taste). Diese Reihenfolge kann durch Verschieben der Objekte verändert werden. Das Objekt wird dazu mit der Maus an seine neue Position gezogen. Solange die Maustaste gedrückt ist, erscheint an der neuen Position ein schwarzer Strich.

Objekt suchen

Kontakt

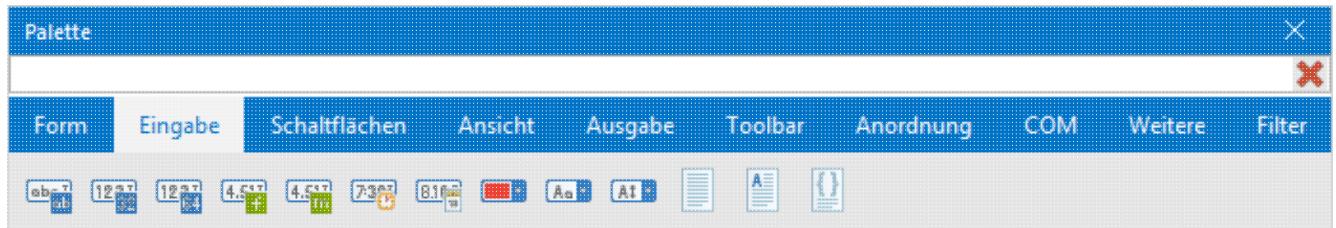
Wird in diesem Fenster die Taste oder die Tastenkombination + gedrückt, kann in einem Eingabefeld ein Name aus dem Objektbaum angegeben werden. Nach Abschließen der Eingabe mit wird auf das entsprechende Objekt positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in dem Objektbaum gesucht. Ist das Ende bzw. der Anfang des Objektbaumes erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

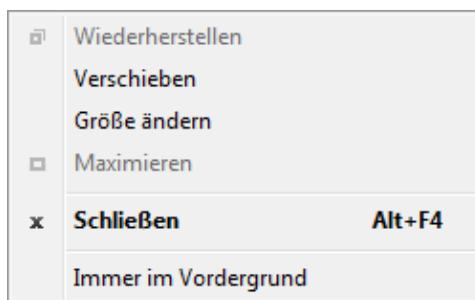
Kontakt

Designer - Palette

In der Palette sind alle Objekte enthalten, die in einem Editierfenster erstellt werden können.



Das Fenster verfügt über ein Kontextmenü, das durch Klicken mit der rechten Maustaste auf den Fenstertitel aufgerufen werden kann.



• Verschieben

Über diesem Menüeintrag kann das Fenster verschoben werden. Das Fenster kann ebenfalls durch Ziehen der Titelzeile verschoben werden. Die ursprüngliche Position kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

• Größe ändern

Mit der Maus kann nur die Länge des Fensters verändert werden. Mit dem Menüeintrag kann ebenfalls die Breite geändert werden. Die ursprüngliche Größe kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

• Schließen

Dieser Menüeintrag hat die gleiche Wirkung, wie die Schließen-Schaltfläche des Eigenschaftsfensters. Das Fenster kann jederzeit durch den Menüeintrag **Anzeige / Eigenschaften** in der Menüzeile wieder aktiviert werden.

• Immer im Vordergrund

Ist dieser Menüeintrag mit einem Haken versehen, kann das Eigenschaftsfenster nicht von einem anderen Fenster verdeckt werden. Durch Anklicken des Menüeintrages kann die Anzeige "Immer im Vordergrund" aktiviert und deaktiviert werden.

Filter

Die Palette kann über das Eingabefeld am oberen Rand durchsucht werden. In dem Feld kann der Name eines Objekttyps eingegeben werden, um danach zu filtern. Die

Kontakt

Suche ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Suchbegriff>* gesucht. Die Objekttypen, die dem Filterkriterium entsprechen, werden auf der Seite Filter angezeigt.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

In der Palette sind alle Objekte enthalten die erstellt werden können. Jedes Objekt ist durch ein Symbol dargestellt. Durch einen Klick auf das entsprechende Symbol wird ein Objekt im Editierfenster erstellt. War zuvor ein Frame-Objekt oder ein GroupBox-Objekt selektiert, wird das Objekt in der linken oberen Ecke des Frame- oder GroupBox-Objektes erzeugt. Im anderen Fall wird das Objekt relativ zum zuvor selektierten Objekt angeordnet. Dabei werden die Einstellungen / Objekterstellung der Entwicklungsumgebung verwendet.

Die Objekte sind in folgende Bereiche untergliedert:

- **Form**

Im Bereich Form befinden sich Dialog- und Druckformulare.

- **Eingabe**

Im Bereich Eingabe befinden sich die ganzen Eingabe-Objekte für die unterschiedlichen Datentypen. Jedes Eingabeobjekt kann mit einem Feld in der Datenbank mit gleichem Typ verknüpft werden.

- **Schaltflächen**

Im Bereich Schaltflächen befinden sich alle Schaltflächen-Objekte.

- **Ansicht**

Im Bereich Ansicht befinden sich Objekte zur Datenausgabe in Listen- und Baumform.

- **Ausgabe**

Im Bereich Ausgabe befinden sich alle Ausgabe-Objekte.

- **Toolbar**

Im Bereich Toolbar sind alle Objekte enthalten, die für die Verwendung der Toolbar notwendig sind.

- **Anordnung**

Im Bereich Anordnung befinden sich Objekte zur Gruppierung und Anordnung von anderen Objekten.

- **COM**

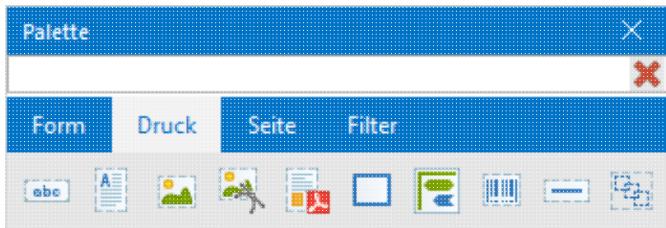
Im Bereich COM sind Objekte enthalten, die den Zugriff auf Dokumente anderer Applikationen innerhalb der CONZEPT 16-Anwendung erlauben.

- **Weitere**

Kontakt

Im Bereich Weitere befinden sich Objekte, die in keinen anderen Bereich passen.

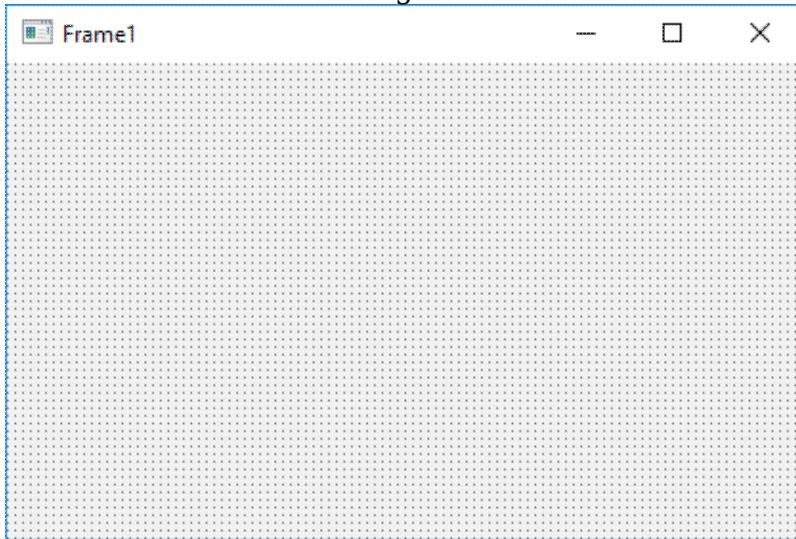
Bei Bearbeitung von Druckformular-Objekten steht folgende Palette zur Verfügung:



Die Objekte sind in folgende Bereiche untergliedert:

- **Druck-Objekte**
- **Seiten-Objekte**
- **Tabellen-Objekte**

Designer - Anwendungsfenster Fenster für die Anwendung



Im Editierfenster wird das Fenster der Anwendung zusammengestellt. Dieses Fenster kann geladen, gespeichert und in der Anwendung verwendet werden.

Einfügen neuer Objekte

Neue Objekte werden durch Anklicken des Objekts in der Palette in das Fenster eingefügt. Das Objekt wird nach rechts unten versetzt vom selektierten Objekt eingefügt. Ist kein Objekt selektiert, erscheint es in der oberen linken Ecke des übergeordneten Objekts. Die Eigenschaften des Objekts werden im Fenster Eigenschaften dargestellt.

Objekte markieren

Einzelne Objekte können durch Anklicken mit der Maus markiert werden. Mehrere Objekte werden bei gedrückter -Taste nacheinander angeklickt. Die Markierung wird um das entsprechende Objekt erweitert. Sollen mehrere zusammenstehende Objekte markiert werden, kann mit der Maus ein Rahmen um die betreffenden Objekte gezogen werden. Es werden nur die Objekte selektiert, die vollständig vom Selektionsrahmen umschlossen werden.

Der Inhalt eines Container-Objektes (zum Beispiel GroupBox oder NotebookPage) kann erst dann durch Ziehen mit der Maus markiert werden, wenn ein anderes Objekt als der Container selektiert ist. Im Fall, dass der Container selektiert ist, wird das Ziehen mit der Maus als Verschieben des Containers interpretiert. Ein Objekt innerhalb eines selektierten Containers kann bei gedrückter -Taste selektiert werden.

Mehrfachselektion

Werden mehrere Objekte hintereinander durch Halten der -Taste selektiert, gibt es zusätzlich die Möglichkeit durch Betätigen von (also +) ein Objekt, welches innerhalb des Selektionsrahmens liegt, zu selektieren.

Kontakt

Sofern mehr als ein Objekt selektiert ist, wird um jedes selektierte Objekt ein doppelter blau gestrichelter Rahmen gezeichnet. Das so selektierte Objekt kann durch die gleiche Vorgehensweise auch wieder einzeln deseletiert werden.

Alle Objekte selektieren

Über die Tastenkombination  +  werden alle Objekte des aktiven Fensters markiert.

Alle Objekte deseletieren

Über die Tastenkombination  +  werden alle Objekte des aktiven Fensters demarkiert.

Markierte Objekte verschieben oder löschen

Markierte Objekte können mit der Maus verschoben oder über das Menü **Bearbeiten** gelöscht oder ausgeschnitten werden. Das Kontextmenü stellt noch weitere Funktionen zum Ausrichten der markierten Objekte zur Verfügung.

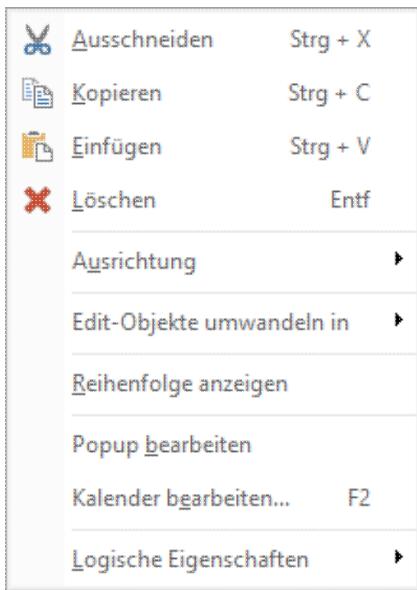
Wird beim Verschieben eines Objekts die  gedrückt gehalten, wird an der Stelle, wo die Maustaste losgelassen wird, eine Kopie der markierten Objekte erstellt.

Bewegung des Selektionsrahmens mit der Tastatur

Wird die Option "Tastatursteuerung" in den Designer-Einstellungen ausgewählt, kann die Selektion auch per Pfeiltasten bewegt werden. Unter Zuhilfenahme der  -Taste wird die Selektion jeweils um drei Rastereinheiten bewegt. Mit gedrückter  -Taste wird die Selektion jeweils um eine halbe Rastereinheit bewegt, ansonsten um eine Rastereinheit.

Kontextmenü für Objekte

Für jedes Objekt steht ein Kontextmenü zur Verfügung, welches je nach Objekttyp unterschiedliche Optionen zur Verfügung stellt:



• Ausrichtung

- ◆ **Links**
Die markierten Objekte werden linksbündig ausgerichtet.
- ◆ **Rechts**
Die markierten Objekte werden rechtsbündig ausgerichtet.
- ◆ **Open**
Die markierten Objekte werden am obersten selektierten Objekt ausgerichtet.
- ◆ **Unten**
Die markierten Objekte werden untersten selektierten Objekt ausgerichtet.
- ◆ **Zentriert untereinander**
Die markierten Objekte werden untereinander zentriert.
- ◆ **Zentriert nebeneinander**
Markierte Objekte, die nebeneinander liegen, werden zueinander zentriert.
- ◆ **Gleiche Breite**
Alle markierten Objekte erhalten die gleiche Breite. Maßgebend ist das zuerst gewählte Objekt (roter Selektionsrahmen).
- ◆ **Gleiche Höhe**
Alle markierten Objekte erhalten die gleiche Höhe. Maßgebend ist das zuerst gewählte Objekt (roter Selektionsrahmen).
- ◆ **Gleicher Abstand horizontal**
Alle markierten Objekte werden auf den gleichen horizontalen Abstand

Kontakt

gesetzt. Maßgebend ist der Abstand zwischen dem ersten und zweiten ausgewählten Objekt.

- ◆ **Gleicher Abstand vertikal**

Alle markierten Objekte werden auf den gleichen vertikalen Abstand gesetzt. Maßgebend ist der Abstand zwischen dem ersten und zweiten ausgewählten Objekt.

- ◆ **Breite wie übergeordnetes Objekt**

Das bzw. die selektierten Objekte werden an den linken Rand des übergeordneten Objekts verschoben und bekommen die gleiche Länge wie das übergeordnete Objekt.

- ◆ **Höhe wie übergeordnetes Objekt**

Das bzw. die selektierten Objekte werden an den oberen Rand des übergeordneten Objekts verschoben und bekommen die gleiche Höhe wie das übergeordnete Objekt.

- **Edit-Objekte umwandeln in**

- ◆ **Edit**

Umwandlung in ein Edit-Objekt

- ◆ **IntEdit**

Umwandlung in ein IntEdit-Objekt

- ◆ **BigIntEdit**

Umwandlung in ein BigIntEdit-Objekt

- ◆ **FloatEdit**

Umwandlung in ein FloatEdit-Objekt

- ◆ **DecimalEdit**

Umwandlung in ein DecimalEdit-Objekt

- ◆ **DateEdit**

Umwandlung in ein DateEdit-Objekt

- ◆ **TimeEdit**

Umwandlung in ein TimeEdit-Objekt

- **Reihenfolge anzeigen**

Für jedes Objekt wird die Tab-Position innerhalb der Applikation angezeigt. Die Anzeige kann ebenfalls bei einer 3-Tasten-Maus mit der mittleren Taste und bei einer Wheelmaus durch Drücken des Rades aktiviert werden. Die Reihenfolge wird nur bei den markierten Objekten angezeigt.

Durch Doppelklick auf die Ziffer kann die Reihenfolge geändert werden.

Alternativ ist das Ändern der Reihenfolge auch in dem Fenster möglich.

- **Popup bearbeiten**

Für die Objekte Edit, IntEdit, BigIntEdit, FloatEdit, DecimalEdit, TimeEdit und DateEdit steht diese Menüoption zur Verfügung. Bei Anwahl der Option wird für das Eingabeobjekt ein PopupList-Objekt erzeugt.

- **Kalender bearbeiten**

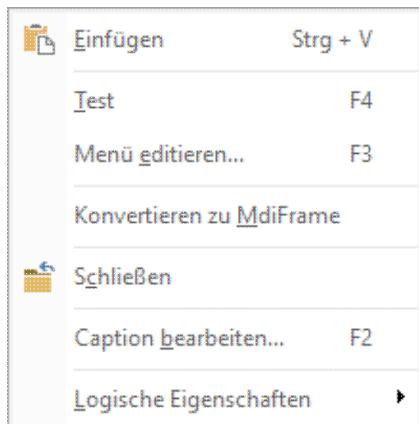
Ist das Objekt vom Typ DateEdit, wird ein Editor für die Konfiguration des Kalenders aufgerufen.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften des Objekts zur direkten Bearbeitung.

Kontextmenü des Fensters

Durch Betätigen der rechten Maustaste auf dem Hintergrund des Fensters, wird das Kontextmenü aufgerufen.



Folgende Funktionen stehen zur Verfügung:

- **Einfügen**

Einfügen ausgeschnittener oder kopierter Objekte.

- **Test**

Ausführen des Fensters im Testmodus. Im Testmodus werden keine Funktionen (beispielsweise bei Ereignissen) ausgeführt.

- **Menü editieren**

Aufruf des Menü-Editors

- **Konvertieren zu MdiFrame**

Bei einem Fenster vom Typ Frame

- **Konvertieren zu Frame**

Bei einem Fenster vom Typ MdiFrame

- **Schließen**

Schließen des Fensters

- **Caption bearbeiten...**

Caption des Fensters bearbeiten

- **Logische Eigenschaften**

Liste der logischen Eigenschaften des Fensters zur direkten Bearbeitung.

Tastaturkommandos

Folgende Tasten und Tastenkombinationen können im Fenster verwendet werden:

- **Navigation**

- ◆ und + selektiert das nächste bzw. vorhergehende Objekt. Die Reihenfolge wird von der Fokusreihenfolge bestimmt (auch wenn das entsprechende Objekt den Fokus nicht bekommen kann).

- **Markieren**

- ◆ + selektiert alle Objekte in dem Fenster.
- ◆ + + selektiert alle Objekte in dem Fenster.
- ◆ + selektiert alle untergeordneten Objekte des selektierten Objekts.
- ◆ + hebt die Selektion auf.
- ◆ + + erweitert die Selektion um das angeklickte Objekt. Ist das Objekt bereits selektiert, wird es aus der Selektion entfernt.

- **Eigenschaften ändern**

- ◆ , , und positioniert das Objekt einen Rasterpunkt (siehe Einstellungen) weiter in die angegebene Richtung.
- ◆ + , + , + und + positioniert das Objekt drei Rasterpunkte (siehe Einstellungen) weiter in die angegebene Richtung.
- ◆ , , + und , positioniert das Objekt einen halben Rasterpunkt (siehe Einstellungen) weiter in die angegebene Richtung.
- ◆ ruft den Objekt-Editor des Objektes auf, sofern vorhanden. Bei Objekten, die eine PopupList enthalten können, wird diese hinzugefügt bzw. geöffnet. Andernfalls wird ein Fenster zur Bearbeitung der Caption des Objektes geöffnet.

- **Ausschneiden / Kopieren / Einfügen**

- ◆ + oder + schneidet die markierten Objekte aus.
- ◆ + oder + kopiert die markierten Objekte.
- ◆ + oder + fügt die ausgeschnittenen/kopierten Objekte ein.
- ◆ löscht die selektierten Objekte.
- ◆ + stellt die zuletzt gelöschten Objekte wieder her.
- ◆ + und + macht die letzte Aktion rückgängig, bzw. führt die zuletzt rückgängig gemachte Aktion wieder aus.

• **Speichern**

- ◆  +  speichert das aktuelle Fenster.

• **Hilfe**

- ◆  aktiviert die kontextsensitive Hilfe.

Kontakt

Editor

Mit dem Editor können die Prozeduren für die Anwendung geschrieben und interne Texte verwaltet werden.

Siehe [Tastenkürzel](#)
[\(Blog\)](#)

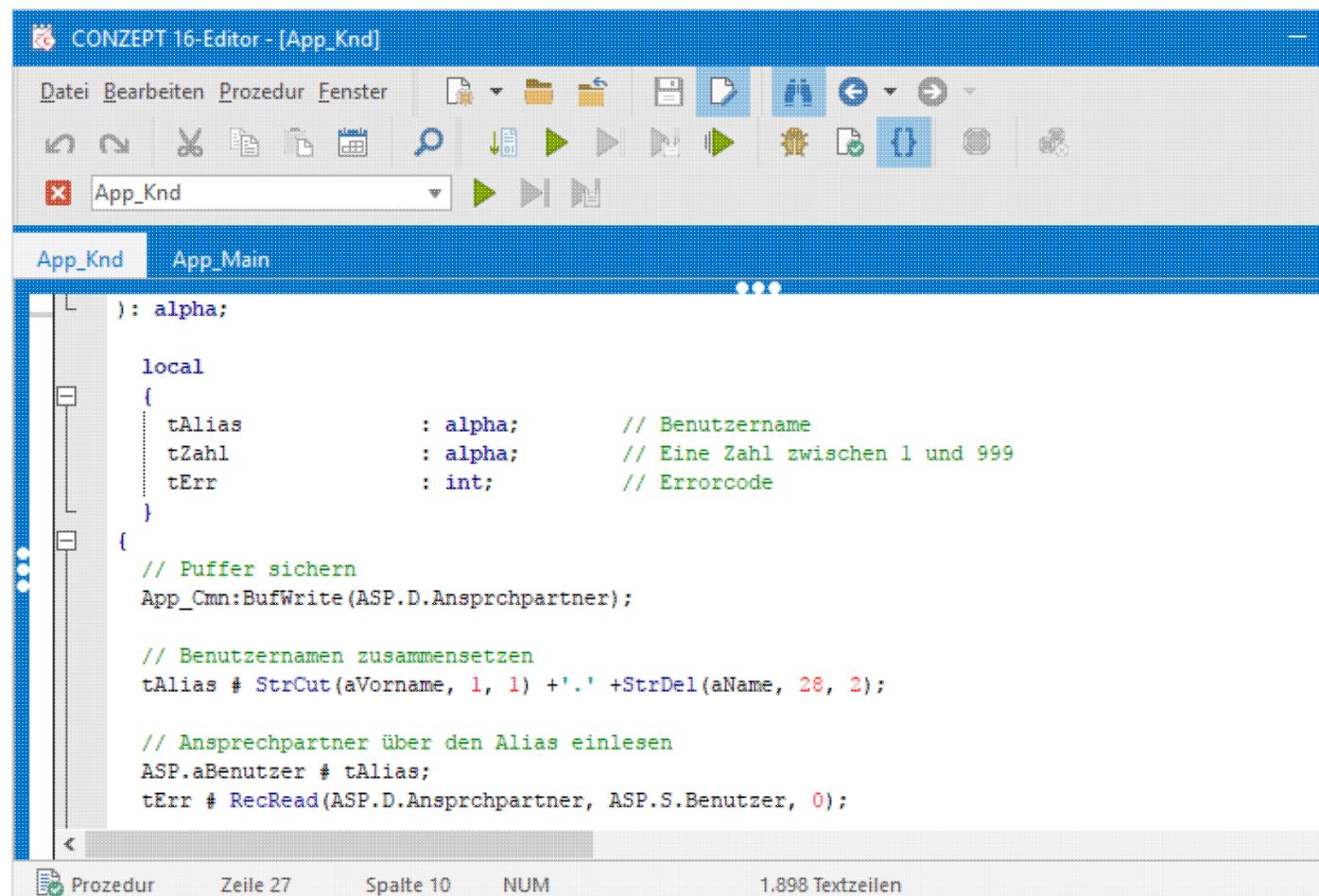
Der Editor kann über den Menüeintrag **Anzeige / Editor** aktiviert werden.

Prozeduren können in der CONZEPT 16 eigenen Sprache, die sich an C/C++ oder PASCAL (siehe [Compileranweisungen](#)) anlehnt, geschrieben werden. Reservierte Worte, Kommentare und Konstanten werden in unterschiedlichen Farben dargestellt (Syntax-Highlighting).

Im Editor können mehrere Dokumente gleichzeitig bearbeitet werden. Es steht eine mehrstufige UNDO-Funktion zur Verfügung.

Geschriebene Prozeduren können mit Hilfe des [Source-Level Debuggers](#) überprüft werden. Die Prozeduren enthalten Funktionen zur vollständigen Dialogsteuerung. Somit ist es möglich, dass sämtliche Objekteigenschaften oder auch ganze Dialoge dynamisch zur Laufzeit geändert werden können.

Der Editor kann über den Menüeintrag **Anzeige / Editor**, die Schließen-Schaltfläche oder den Menüeintrag **Datei / Beenden** des Editor-Fensters ausgeblendet werden. Der Zustand des Editors und der zu diesem Zeitpunkt geöffneten Dokumente bleibt erhalten.



The screenshot shows the CONZEPT 16-Editor interface. The title bar reads "CONZEPT 16-Editor - [App_Knd]". The menu bar includes "Datei", "Bearbeiten", "Prozedur", "Fenster". The toolbar contains various icons for file operations like Open, Save, Print, and search. A dropdown menu shows "App_Knd". Below the toolbar is a tab bar with "App_Knd" and "App_Main". The main area is a code editor with the following content:

```
    ): alpha;

    local
    {
        tAlias      : alpha;          // Benutzername
        tZahl       : alpha;          // Eine Zahl zwischen 1 und 999
        tErr        : int;            // Errorcode
    }
    {
        // Puffer sichern
        App_Cmn:BufWrite(ASP.D.Anspchpartner);

        // Benutzernamen zusammensetzen
        tAlias # StrCut(aVorname, 1, 1) + '.' +StrDel(aName, 28, 2);

        // Ansprechpartner über den Alias einlesen
        ASP.aBenutzer # tAlias;
        tErr # RecRead(ASP.D.Anspchpartner, ASP.S.Benutzer, 0);
    }
```

The status bar at the bottom shows "Prozedur", "Zeile 27", "Spalte 10", "NUM", and "1.898 Textzeilen".

Der Editor besteht aus folgenden Bereichen:

- **Menü**
- **Werkzeugleiste**
- **Geöffnete Dokumente**
- **Editierbereich**
- **Suchoptionen**
- **Statuszeile**

Funktionen im Kontextmenü

Direkte Befehlshilfe	In der Datenbankinfo wird die Hilfe zu dem angeklickten Befehl angezeigt.
<u>Ausführen</u>	Die aktuelle Prozedur wird ausgeführt.
<u>Ausführen mit Stopp</u>	Die aktuelle Prozedur wird ausgeführt und vor der ersten Anweisung im Debugger angehalten.
<u>Ausschneiden</u>	Die Markierung wird in die Zwischenablage verschoben.
<u>Kopieren</u>	Die Markierung wird in die Zwischenablage kopiert.
<u>Einfügen</u>	Der Inhalt der Zwischenablage wird eingefügt.
<u>UTF-8-Text</u>	Der Inhalt der Zwischenablage wird als UTF-8 -Zeichenkette eingefügt.
<u>Einfügen</u>	Suchen eines Textes in dem aktuellen Dokument.
<u>Suchen...</u>	Ersetzen eines Textes in dem aktuellen Dokument.
<u>Ersetzen...</u>	Direktes Anspringen einer bestimmten Zeile oder Sub-Funktion.
<u>Gehe zu...</u>	Alle Leerzeichen werden durch einen Punkt angezeigt.
Leerzeichen anzeigen	
Großwandlung	Der markierte Text wird in Großbuchstaben umgewandelt.
Kleinwandlung	Der markierte Text wird in Kleinbuchstaben umgewandelt.
<u>Rückgängig</u>	Die vorhergehende Änderung wird rückgängig gemacht.
<u>Wiederholen</u>	Die letzte rückgängig gemachte Operation wird wieder hergestellt.

Menü des Editors

Menüzeile des Editors

Der Editor wird über die Menüzeile und die Werkzeugleiste gesteuert. Einige der Menüeinträge können ebenfalls über Schaltflächen der Werkzeugleiste ausgeführt werden.

Die Menüzeile besteht aus folgenden Menüs:

- **Datei**

Dokumente öffnen, schließen oder speichern und den Editor schließen.

- **Bearbeiten**

Dokumenttext bearbeiten und durchsuchen.

- **Prozedur**

Dokumenttext übersetzen und ausführen.

- **Fenster**

Zwischen geöffneten Dokumenten wechseln.

- **Status**

Status der MergeTool-Ansicht.

- **Vergleich**

Aktionen des MergeTools.

- **Info**

Willkommensseite öffnen.

Menüeintrag Datei des Editors
Menüeinträge im Menü Datei des Editors
Menü des
Editors,
Siehe Menüeintrag
Bearbeiten
des Editors

Menü Datei für Dokumente

Menü Datei der Willkommensseite

Neu

Ein neues Dokument kann über den Menüeintrag **Datei / Neu** angelegt werden. Zuvor muss als Typ entweder Prozedur oder Text ausgewählt werden. Es wird ein Editierfenster ohne Inhalt geöffnet. In der Liste der Dokumente erscheint ein Registerreiter mit dem Namen "Unbenannt" mit einer fortlaufenden Nummer. Der Name des Dokuments muss erst beim Speichern angegeben werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Öffnen

Ein vorhandenes Dokument kann über den Menüeintrag **Datei / Öffnen** geöffnet werden. Das Dokument kann anschließend aus einer Liste ausgewählt werden.

In der Liste werden alle Informationen zu den Dokumenten angezeigt. Die Angabe der Größe erfolgt in Byte.

In diesem Dialog können ebenfalls nicht mehr benötigte Dokumente gelöscht werden. Nach der Auswahl des Dokuments und Drücken der -Taste wird das Dokument gelöscht.

Das Öffnen eines Dokuments erzeugt ein neues Editierfenster, in dem der Inhalt des Dokuments dargestellt wird.

Der Name des Dokuments kann im Feld "Name" eingegeben werden. Gibt es kein Dokument mit dem angegebenen Namen, wird der Text grau dargestellt und die [Ok]-Schaltfläche ist deaktiviert. Ist das Dokument zur Zeit von einem anderen Client in Bearbeitung, wird der Text rot dargestellt. Dokumente, die von einem anderen Benutzer bearbeitet werden, können im Nur-Lesen-Modus geöffnet werden. Zusätzlich wird der Name, die ID und der Rechnername des sperrenden Benutzers in der Titelzeile angezeigt.

Soll das Dokument unabhängig von anderen Benutzern im Nur-Lesen-Modus geöffnet werden, kann die Option "Nur lesend" aktiviert werden.

Kontakt

Prozeduren, die über die Include-Anweisung in eine Prozedur eingebunden wurden, können durch Doppelklicken des Prozedurnamens hinter der Include-Anweisung geöffnet werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

MergeTool öffnen

Über diesen Menüpunkt wird ein neues MergeTool geöffnet.

Schließen

Ein Dokument wird über den Menüeintrag **Datei / Schließen** geschlossen. Ist das Dokument noch nicht gespeichert, erfolgt eine Abfrage:

Durch Drücken der Schaltfläche [Ja] wird ein Dialog aufgerufen, in dem der Name des Dokuments eingetragen werden kann. [Nein] schließt das Editierfenster, ohne das Dokument zu speichern. [Abbrechen] schließt das Editierfenster nicht.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Alle schließen

Über diesen Menüeintrag können alle geöffneten Dokumente geschlossen werden. Bei geänderten und noch nicht gespeicherten Dokumenten erfolgt für jedes der Dokumente, eine Abfrage, ob es gespeichert werden soll.

Speichern und Speichern unter

Über den Menüeintrag **Datei / Speichern** wird das Dokument, das zur Zeit bearbeitet wird, unter dem vorhandenen Namen gespeichert. Hat das Dokument noch keinen Namen, wird ein Name in einem Dialog abgefragt.

Der Menüeintrag wird erst nach einer Änderung in dem Dokument aktiviert.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche  aufgerufen werden.

Über den Menüeintrag **Datei / Speichern unter...** wird das Dokument, das zur Zeit bearbeitet wird unter einem neuen Namen gespeichert. Der Name wird in einem Dialog abgefragt.

Bei der Angabe eines Namens mit ungültigen Zeichen, wird der Name durchgestrichen dargestellt und die [Ok]-Schaltfläche deaktiviert. Der Name darf keine Sonderzeichen beinhalten. Der Doppelpunkt kann im Prozedurnamen verwendet werden, es können dann allerdings keine Sub-Funktionen aus dieser Prozedur aufgerufen werden, da die Trennung zwischen Prozedurname und Funktionsname

Kontakt

nicht mehr eindeutig ist. Entsprechend verhält es sich mit anderen Sonderzeichen, die zum Beispiel für mathematische Funktionen verwendet werden. Ein Prozedurname sollte sich daher ebenso wie ein Funktionsname nur aus Buchstaben, Zahlen und dem Unterstrich "_" zusammensetzen.

Bearbeitungsmodus

Das zur Zeit geöffnete Dokument kann über diesen Menüeintrag zwischen dem "Nur lesen"- und dem Lese-/Schreibmodus umgeschaltet werden. Ist das Dokument von einem anderen Benutzer gesperrt, wird der Name und der Rechner des Benutzers in einer Meldung angezeigt.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Assistent

Mit diesem Menüpunkt wird der Assistent ein- und ausgeblendet. Der Assistent kann ebenfalls über die Schaltfläche in der Werzeugleiste und über den Menüpunkt "Anzeige / Assistent" des Designers aktiviert und deaktiviert werden.

Importieren... / Exportieren...

Über den Menüeintrag **Datei / Importieren...** bzw. **Exportieren...** kann das zur Zeit bearbeitete Dokument als ASCII-Datei importiert bzw. exportiert werden.

Einstellungen

Dieser Menüpunkt öffnet den Einstellungsdialog des Designers.

Zurück (zu "...") / Vorwärts (zu "...")

Mit den Menüpunkten der Navigationshistorie kann zwischen den zuletzt betrachteten Dokumenten hin und her navigiert werden. In der Werkzeugleiste kann ebenfalls mit den Schaltflächen bzw. zwischend en Dokumenten navigiert werden.

Zuletzt geöffnete Dokumente

In diesem Bereich werden die zuletzt geöffneten Dokumente angezeigt und können durch Anklicken geöffnet werden. Der Editor merkt sich bis zu zehn Dokumente.

Beenden

Dieser Menüeintrag schließt den Editor. Über den Menüeintrag Anzeige / Editor wird der Editor wieder in dem Zustand angezeigt, in dem er geschlossen wurde.

Kontakt

Menüeintrag Bearbeiten des Editors

Menüeinträge im Menü Bearbeiten des Editors

Menüeintrag

Datei des

Siehe Editors,

Menüeintrag

Prozedur des

Editors

In diesem Menü sind alle Funktionen enthalten, die zum Bearbeiten und Durchsuchen von Dokumenten benötigt werden.

Rückgängig

Mit diesem Menüeintrag können Änderungen im Editor rückgängig gemacht werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Wiederholen

Versehentlich rückgängig gemachte Änderungen können mit diesem Menüeintrag wieder hergestellt werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Ausschneiden

Der markierte Bereich wird in die Windows-Zwischenablage verschoben. Der Inhalt der Zwischenablage kann über den Menüeintrag **Bearbeiten / Einfügen** wieder in das Dokument kopiert werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Kopieren

Der markierte Bereich wird in die Windows-Zwischenablage kopiert. Der Inhalt der Zwischenablage kann über den Menüeintrag **Bearbeiten / Einfügen** wieder in das Dokument kopiert werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Einfügen

Der Inhalt der Windows-Zwischenablage wird in das zur Zeit bearbeitete Dokument an der derzeitigen Cursorposition eingefügt. Die Zwischenablage bleibt erhalten und kann ein weiteres mal eingefügt werden.

Kontakt

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Datum einfügen

An der Cursorposition wird das aktuelle Datum in das Dokument eingefügt.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Alles markieren

Mit diesem Menüeintrag wird das gesamte Dokument markiert.

Suchen

Über diesen Menüeintrag wird folgende Werkzeugleiste eingeblendet:

Im Feld "Suche" wird der Suchbegriff eingetragen. Der Cursor springt innerhalb des Dokumenttextes sofort an das erste Vorkommen dieses Textes. Über die entsprechenden Schaltflächen kann nach einem weiteren Vorkommen des Textes nach oben oder nach unten gesucht werden. Folgende Optionen können bei der Suche angegeben werden:

- **Groß-/Kleinschreibung beachten**

Durch Aktivieren dieser Schaltfläche wird eine Groß-/Kleinschreibung des Suchbegriffs beachtet. Der Suchbegriff wird dann gefunden, wenn er im Dokument genau so geschrieben wurde.

- **Begriffsorientiert**

Ist diese Schaltfläche aktiviert, wird der Suchbegriff nur als ganzes Wort und nicht als Wortbestandteil gesucht.

- **Regulärer Ausdruck**

Bei Verwendung dieser Option, wird der Suchtext als regulärer Ausdruck interpretiert.

- **Im selektierten Bereich suchen**

Ist diese Schaltfläche aktiviert, wird nur im selektierten Bereich gesucht.

- **Wortvorkommen fixieren**

Ist diese Schaltfläche aktiviert, bleiben die markierten Wortvorkommen markiert, bis eine neue Suche durchgeführt wird, oder doppelt in einen leeren Bereich geklickt wird. Ist sie nicht aktiviert, wird die Hervorhebung mit einem einfachen Klick aufgehoben.

Der Vorgang kann über die -Taste abgebrochen werden. Soll der Text weiter durchsucht werden, kann die Tastenkombination + oder die Schaltfläche

Kontakt

verwendet werden. Mit der Kombination + bzw. der Schaltfläche wird in umgekehrter Richtung nach dem Suchbegriff gesucht.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Ersetzen

Über diesen Menüeintrag wird folgende Werkzeugleiste eingeblendet:

Das Ersetzen entspricht weitestgehend der Suchen-Funktion. Nachdem der angegebene Suchbegriff gefunden wurde, kann er durch den im Feld "Ersetzen mit" angegebenen Begriff ersetzt werden. Die Optionen entsprechen denen beim Suchen.

Durch Betätigen der Schaltfläche "Text ersetzen" wird die markierte Fundstelle durch den im Feld "Ersetzen mit" eingetragenen Text ersetzt. Der Vorgang kann jederzeit mit der -Taste abgebrochen werden.

Die Schaltfläche "Alles ersetzen" ersetzt alle Suchbegriffe.

Gehe zu Zeile

Über diesen Menüeintrag wird folgende Werkzeugleiste eingeblendet:



Mit dem Menüeintrag **Gehe zu Zeile** kann auf eine bestimmte Zeile positioniert werden. Die Nummer der Zeile wird in dem Feld eingegeben. Durch Drücken der Schaltfläche [go] wird auf die entsprechende Zeile positioniert.

Gehe zu Funktion

Über diesen Menüeintrag wird folgende Werkzeugleiste eingeblendet:

Mit dem Menüeintrag **Gehe zu Funktion** kann auf eine bestimmte Funktion positioniert werden. Die Funktion wird in dem Feld eingegeben oder aus der ComboBox ausgewählt. Durch Drücken der Schaltfläche [go] wird auf die entsprechende Zeile positioniert.

In dem Feld können die Wildcards '*' und '?' verwendet werden. Sind Wildcards angegeben, werden in der Liste der Funktionen nur die Funktionen angezeigt, die zu der Zeichenkette passen.

Aktuelle Funktion

Über diesen Menüeintrag wird die aktuelle Funktion, in der der Textcursor steht in der Statuszeile angezeigt.

Lesezeichen setzen / entfernen

Mit diesem Menüpunkt kann ein Lesezeichen gesetzt werden. Mit den Lesezeichen kann die Navigation innerhalb des Dokuments erleichtert werden.

Nächstes Lesezeichen und Vorheriges Lesezeichen

Mit diesen Menüpunkten kann zwischen den einzelnen Lesezeichen gesprungen werden. Die Einfügemarkie befindet sich dabei immer in der ersten Spalte der markierten Zeile.

Alle Lesezeichen löschen

Mit diesem Menüpunkt werden alle Lesezeichen des aktuellen Dokuments gelöscht.

Zeilennummern anzeigen

Mit diesem Menüpunkt wird am linken Rand des Editors eine Leiste mit den Zeilennummern angezeigt / ausgeblendet.

Menüeintrag Prozedur des Editors
Prozeduren übersetzen und ausführen

Menüeintrag

Bearbeiten

Siehe des Editors,

Menüeintrag

Fenster des

Editors

In diesem Menü sind alle Funktionen vorhanden, die mit dem Übersetzen und Ausführen der Prozeduren zusammenhängen.

• Übersetzen

Über den Menüeintrag **Prozedur / Übersetzen** wird die zur Zeit bearbeitete Prozedur übersetzt. Ist der Prozedurtext geändert oder noch nicht gespeichert worden, wird die Prozedur zuvor in der Datenbank gespeichert. Tritt beim Übersetzen ein Fehler auf, wird die fehlerhafte Zeile mit einem roten Balken markiert und in der Statuszeile eine Fehlermeldung ausgegeben. Bei einer korrekten Übersetzung erfolgt keine Meldung.

Die Funktion kann ebenfalls in der Werkzeugeiste mit der Schaltfläche aufgerufen werden.

• Ausführen

Dieser Menüeintrag startet die zur Zeit bearbeitete Prozedur. Es wird die main-Funktion ausgeführt. Ist in der Prozedur keine main-Funktion enthalten, wird die Ausführung sofort wieder beendet.



Diese Funktion darf nicht mit der Funktion **Datei / Test** der Entwicklungsumgebung verwechselt werden. Mit der Funktion im Editor wird eine Prozedur gestartet, die Funktion in der Entwicklungsumgebung testet einen Dialog. Prozeduren oder Funktionen werden dort nicht ausgeführt.

Die Funktion kann ebenfalls in der Werkzeugeiste mit der Schaltfläche aufgerufen werden.

• Ausführen mit Stopp

Besteht eine Verbindung mit einem Debugger, kann diese Funktion ausgewählt werden. Nach dem Start der Prozedur wird im Debugger die Verarbeitung vor der ersten Anweisung gestoppt.

Die Funktion kann ebenfalls in der Werkzeugeiste mit der Schaltfläche aufgerufen werden.

• Ausführen mit Stopp (nur Trace)

Besteht eine Verbindung mit einem Debugger, kann diese Funktion ausgewählt werden. Nach dem Start der Prozedur wird im Debugger die Verarbeitung vor der ersten Anweisung gestoppt. Die Anweisungen DbgStop() und DbgControl(_DbgStop), sowie Breakpoints werden ignoriert.

Kontakt

Debugger-Protokollausgaben werden wie gewohnt ausgeführt.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

- **Ausführen Startprozedur**

In den Einstellungen kann eine Prozedur definiert werden, die beim Aufrufen dieses Menüpunktes gestartet wird. Über diesen Menüpunkt kann zum Beispiel die Applikation gestartet werden, auch wenn im Editor nur ein Modul bearbeitet wird.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

- **Ausführen andere...**

Dieser Menüpunkt aktiviert die Toolbar Startleiste.

Dort kann eine Prozedur ausgewählt werden. Die Schaltflächen , und führen die, in der Leiste, ausgewählte Prozedur aus. Sie verhalten sich, wie unter Ausführen, Ausführen mit Stopp bzw. Ausführen mit Stopp (nur Trace) beschrieben.

- **Debugger verbinden**

Über diesen Menüpunkt wird der Kontakt zum Debugger hergestellt. Die Maschine auf dem der Debugger läuft ist entweder der lokale Rechner oder ist in den Einstellungen angegeben. Ist in den Einstellungen kein Rechner eingetragen, wird der Debugger auf der lokalen Maschine gesucht. Ist der Debugger nicht gestartet, wird er aus dem gleichen Verzeichnis gestartet, aus dem auch der CONZEPT 16-Client gestartet wurde.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

- **Einstellungen für Debugger...**

Mit diesem Menüpunkt werden die Einstellungen des Debuggers aufgerufen. Der Dialog entspricht dem Dialog unter dem Menüpunkt Datei / Einstellungen des Designers.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

- **Klammerung hervorheben**

Über diesem Menüpunkt kann die Anzeige der öffnenden bzw. schließenden Klammer aktiviert und deaktiviert werden. Die zusammengehörenden Klammern werden mit einem grauen Rahmen angezeigt.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

- **Breakpoints**

Kontakt

Sobald eine Verbindung mit einem Debugger besteht, können in bereits übersetzte Prozeduren Breakpoints gesetzt werden. Der Debugger hält die Verarbeitung der Prozedur an dem Breakpoint an. Die Anweisung in der betreffenden Zeile wurden zu diesem Zeitpunkt noch nicht durchgeführt.

Die Breakpoints können ebenfalls über das Kontextmenü im linken Rand des Editors gesetzt und entfernt werden.

In diesem Untermenü befinden sich alle Funktionen zur Verwaltung von Breakpoints:

◆ **Breakpoint ein / aus**

Dieser Menüpunkt setzt einen Breakpoint in der aktuellen Zeile. Ist in dieser Zeile bereits ein Breakpoint gesetzt, wird dieser entfernt. Ist in einer Prozedur ein Breakpoint gesetzt, kann diese nicht mehr editiert werden. Das Editieren ist erst wieder nach dem Entfernen aller Breakpoints in der Prozedur möglich.

Breakpoints können nur in Zeilen gesetzt werden, in denen auch P-Code erzeugt wird. Wird nach dem Setzen eines Breakpoints die Prozedur weiter editiert, kann der Breakpoint an eine Stelle verschoben werden, die keinen P-Code erzeugt. Der Punkt würde dann einfach übersprungen werden. Um ein solches Verhalten zu verhindern, kann nach dem Setzen eines Breakpoints die Prozedur nicht mehr verändert werden.

Soll eine Funktion verändert werden, müssen zunächst alle Breakpoints in dieser Prozedur entfernt werden.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

◆ **Nächster Breakpoint und Vorheriger Breakpoint**

Mit diesen Menüpunkten kann zwischen den Breakpoints gesprungen werden. Die Breakpoints werden innerhalb der Prozedur zyklisch angesprungen.

◆ **Prozedur-Breakpoints löschen**

Alle Breakpoints innerhalb der Prozedur werden entfernt. Die Prozedur kann anschließend wieder verändert werden.

◆ **Alle Breakpoints löschen**

Alle gesetzten Breakpoints werden entfernt.

Die Funktion kann ebenfalls in der Werkzeugleiste mit der Schaltfläche aufgerufen werden.

Menüeintrag Fenster des Editors

Anordnen und Wechseln der Dokument-Fenster

Menüeintrag

Prozedur des

Siehe Editors,

Menü des

Editors

Mit diesen Menüpunkten können die Editierfenster mehrerer Dokumente im Editor angeordnet werden. Ebenso kann schnell zwischen verschiedenen Editierfenstern gewechselt werden.

Anordnen

Alle Editierfenster werden kaskadenförmig angeordnet. Von jedem Fenster ist die Titelzeile zu sehen.

Symbole anordnen

Alle zu Symbolen verkleinerte Editierfenster werden in der untersten Fensterzeile des Editors angeordnet. Reicht die Zeilenbreite nicht aus, werden weitere Zeilen von unten nach oben gefüllt.

Übereinander

Alle geöffneten Editierfenster werden übereinander angeordnet. Bei mehreren Fenstern werden diese neben- und übereinander angeordnet.

Nebeneinander

Alle geöffneten Editierfenster werden nebeneinander angeordnet. Bei mehreren Fenstern werden diese neben- und übereinander angeordnet.

Nächstes und Vorheriges

Über diese Menüinträge kann in das nächste bzw. vorhergehende Editierfenster gesprungen werden. Das Wechseln der Fenster ist ebenfalls über die Leiste der Dokumente oder der Tastenkombination  +  ( + ) bzw.  +  +  ( +  + ) möglich.

Fensterliste

Alle geöffneten Editierfenster werden im Untermenü "Fensterliste" angezeigt. Durch Anklicken des entsprechenden Menüpunktes wird direkt in dieses Fenster gewechselt. Zwischen den einzelnen Dokumenten kann ebenfalls mit Hilfe der Registerreiter in der Leiste der Dokumente gewechselt werden.

Das aktuelle Fenster kann mit der Tastenkombination  +  geschlossen werden.

Symbolleisten

Kontakt

Über dieses Untermenü können die verschiedenen Symbolleisten aktiviert und deaktiviert werden. Alle aktiven Symbolleisten werden mit einem Haken vor der Bezeichnung angezeigt. Über die Menüpunkte "Alle sichtbar" und "Alle unsichtbar" können alle Symbolleisten angezeigt oder ausgeblendet werden.

Dokumente

Über diesen Menüeintrag kann die Leiste der Dokumente, in der die Registerreiter für die geöffneten Dokumente angezeigt werden, ein- oder ausgeblendet werden.

Suchleiste

Mit diesem Menüeintrag kann die Suchleiste des Editors aus- oder eingeblendet werden.

Startleiste

Mit diesem Menüeintrag kann die Startleiste des Editors aus- oder eingeblendet werden.

Statuszeile

Mit diesem Menüeintrag kann die Statuszeile des Editors aus- oder eingeblendet werden.

Menüeintrag Status des Editors

Status der MergeTool-Ansicht

Menüeintrag

Fenster des

Siehe Editors,

Menüeintrag

Vergleich

des Editors

Mit diesen Menüpunkten kann die Anzeige des MergeTools beeinflusst werden.

Scrollbars arretieren

Beide Seiten des MergeTools werden so gescrollt, dass die ausgewählte Änderung auf beiden Seiten in der gleichen Höhe angezeigt wird.

Vergleich deaktivieren

Der Vergleich zwischen beiden Seiten des MergeTools wird de-/aktiviert.

Menüeintrag Vergleich des Editors

MergeTool-Aktionen

Menüeintrag

Status des

Siehe Editors,

Menü des

Editors

Mit diesen Menüpunkten können Aktionen mit den verglichenen Dokumenten im MergeTool durchgeführt werden.

Aktuelle Änderung von rechts übernehmen

Die aktuell hervorgehobene Änderung wird von dem Dokument auf der rechten Seite in das Dokument auf der linken Seite übernommen.

Alle Änderungen von rechts übernehmen

Alle Änderungen von dem Dokument auf der rechten Seite werden in das Dokument auf der linken Seite übernommen.

Vorherige Differenz

Sprung zur vorherigen Differenz. Falls es keine vorherige Differenz gibt, wird zur letzten Differenz gesprungen.

Nächste Differenz

Sprung zur nächsten Differenz. Falls es keine nächste Differenz gibt, wird zur ersten Differenz gesprungen.

Vergleich aktualisieren

Die Anzeige der Differenzen wird aktualisiert.

Texte vertauschen

Die Seite der Dokumente wird vertauscht.

Werkzeugleiste des Editors

Häufig genutzte Funktionen des Editors

Menü des
Editors,

Siehe Geöffnete
Dokumente
im Editor

In der Werkzeugleiste sind häufig verwendete Funktionen zusammengefasst. Die Verwendung der Werkzeugleiste ist in der Regel einfacher, als das entsprechende Menü zu öffnen und den Menüeintrag anzuklicken.

Die Werkzeugleiste teilt sich in folgende Bereiche:

.

.

.

.

.

Die einzelnen Funktionen sind in den entsprechenden Menüs beschrieben.

Geöffnete Dokumente im Editor
Anzeige der geöffneten Dokumente
Siehe [Werkzeugleiste](#),
[Editerbereich](#)

In der Leiste der Dokumente werden alle vom Benutzer geöffnete Prozeduren und Texte angezeigt. Durch Anklicken der entsprechenden Reiter kann schnell zwischen den einzelnen Dokumenten gewechselt werden.

Die Leiste der Dokumente kann über den Menüeintrag [Fenster / Dokumente](#) oder das Kontextmenü ein- und ausgeblendet werden.



Das Kontextmenü

Das Kontextmenü wird durch Anklicken der Leiste der Dokumente mit der rechten Maustaste geöffnet.

Folgende Funktionen stehen im Kontextmenü zur Verfügung:

- **Schließen**

Das Dokument, auf dessen Tab rechts geklickt wurde, wird geschlossen. Wurde im freien Bereich rechts geklickt, wird das aktive Dokument geschlossen.

- **Restliche schließen**

Alle Dokumente außer dem, auf dessen Tab rechts geklickt wurde, werden geschlossen. Wurde im freien Bereich rechts geklickt, werden alle außer das aktive Dokument geschlossen.

- **Tabs nach Links schließen** oder **Tabs nach Rechts schließen**

Alle Dokumente, die sich links bzw. rechts von dem Tab befinden, auf den rechts geklickt wurde, werden geschlossen. Wurde im freien Bereich rechts geklickt, werden alle Dokumente, die sich links bzw. rechts vom außer aktiven Dokument befinden, geschlossen.

- **Oben, Unten, Links oder Rechts**

Mit diesen Einträgen kann angegeben werden, auf welcher Seite des Editors die Reiter der Dokumente angezeigt werden sollen. Der Ort der derzeitigen Darstellung ist mit einem Punkt markiert.

- **Sortieren**

Die Reiter werden in einer bestimmten Reihenfolge angezeigt. Die Reihenfolge kann über die Auswahl der Sortierung unter diesem Menüeintrag geändert werden. Die derzeit aktive Sortierung ist mit einem Punkt markiert.

- **Ausblenden**

Die Leiste der Dokumente wird ausgeblendet. Die Reiter für die geöffneten Dokumente werden nicht mehr angezeigt. Eine ausgeblendete Dokumentleiste kann über den Menüeintrag [Fenster / Dokumente](#) wieder eingeblendet werden.

Editierbereich des Editors

Darstellung und Editieren der geöffneten Dokumente

Im Editierbereich des Editors werden alle geöffneten Dokumente in eigenen Fenstern dargestellt. Die Dokumente können hier editiert und über die Funktionen des Editors gespeichert und übersetzt werden.

Titelzeile

In der Titelzeile wird der Name des Dokuments, das im Editierfenster angezeigt wird, dargestellt. Ist das Fenster maximiert, erscheint der Name des Dokuments in der Titelzeile des Editors. Ebenso verschieben sich, wie bei MDI-Fenstern üblich, die Systemschaltflächen in die Menüzeile.

Anmerkungsleiste

In der Anmerkungsleiste werden Positionen von Lesezeichen und Suchtreffern angezeigt. Mit einem Klick auf die Leiste wird im Editierbereich zu der jeweiligen Zeile gesprungen. Der aktuell angezeigte Code-Ausschnitt wird mit einem glauen Bereich hinterlegt.

Zeilennummern

Die Zeilennummern können zwischen der Anmerkungsleiste und dem Informationsrand über die Tastenkombination  +  ein- / ausgeblendet werden.

Informationsrand

Der Rand wird links vom Dokumenttext angezeigt. Er wird durch eine andere Hintergrundfarbe vom Text getrennt. In diesem Rand werden Lesezeichen und Breakpoints dargestellt. Die Lesezeichen werden mit der Taste  vor die Zeile mit dem Cursor gesetzt oder gelöscht. Die gesetzten Lesezeichen können mit den Tastenkombinationen  +  und  +  angesprungen werden. Breakpoints werden mit der Taste  gesetzt oder gelöscht und mit den Tastenkombinationen  +  und  +  angesprungen. Existieren mehrere Markierungen, wird mit  zur folgenden und mit  zur vorherigen Markierung gesprungen. Mit den Plus- und Minus-Symbolen kann der Prozedurtext an Klammern zu- bzw. aufgeklappt werden.

Bildlaufleiste

Über die horizontalen und vertikalen Bildlaufleisten kann der Text geblättert werden.

Bereichstrenner

Die Bereichstrenner befinden sich am linken und oberen Rand. Durch Ziehen der Bereichstrenner wird der Text geteilt und kann unabhängig voneinander gescrollt werden. Durch die Bereichstrenner können bis zu vier Bereiche entstehen. Zwischen den Bereichen kann mit den Tastenkombinationen  +  ...  gewechselt

werden.

Editierbereich

Im Editierbereich kann das Dokument editiert werden. Nach einer Übersetzung der Prozedur wird der erste gefundene Fehler durch einen roten Balken gekennzeichnet. Erfolgt die Übersetzung ohne Fehler, erscheint keine Markierung.

Wird im Editierbereich mit der rechten Maustaste geklickt, erscheint folgendes Kontextmenü:

 Direkte Befehlshilfe	Strg + F1
 Ausführen	Strg + T
 Ausführen und anhalten	Umschalt + Strg + T
 Ausführen und anhalten (nur Trace)	
 Ausschneiden	Strg + X
 Kopieren	Strg + C
 Einfügen	Strg + V
 Suchen	Strg + F
Ersetzen	Strg + H
Gehe zu Zeile	Strg + G
Gehe zu Funktion	Umschalt + Strg + G
Aktuelle Funktion	Umschalt + Strg + F
Leerzeichen anzeigen	Alt + Strg + T
Großwandlung	Umschalt + Strg + U
Kleinwandlung	Strg + U
Zuklappen	Alt + Strg + C
Aufklappen	Alt + Strg + O
Alles zuklappen	Umschalt + Strg + C
Alles aufklappen	Umschalt + Strg + O
Untergeordnete zuklappen	Umschalt + Strg + S
Untergeordnete aufklappen	Alt + Strg + S
 Rückgängig	Strg + Z
 Wiederholen	Strg + Y
 Schließen	Strg + F4

Kontakt

Die Funktionen entsprechen im wesentlichen den gleichnamigen Funktionen im Menü bzw. in der Werkzeugeiste. Im linken Rand des Editerbereiches wird ein anderes Kontextmenü aufgerufen, in dem die Funktionen für Breakpoints aufgerufen werden können.

Mit dem Eintrag **Direkte Befehlshilfe** kann die Hilfe zu einem Befehl auf der Seite "Hilfe" des Datenbankinfo-Fensters angezeigt werden. Steht der Eingabefokus auf einem Datenbankfeld, wird auf der Seite "Datenstruktur" der Baum bis zu diesem Feld aufgeklappt. Wird der Eintrag **Leerzeichen anzeigen** mit einem Haken versehen, werden alle Leerzeichen im Prozedurtext durch einen Punkt dargestellt. Über die Einträge **Großwandlung** bzw. **Kleinwandlung** kann ein markierter Bereich in Großbuchstaben bzw. Kleinbuchstaben umgewandelt werden. Mit den Menüpunkten (Alles / Untergeordnete) zuklappen bzw. aufklappen können einzelne, alle oder alle untergeordneten Folding-Punkte zu- bzw. aufgeklappt werden.

Fehler beim Übersetzen der Prozedur werden direkt im Editierfenster angezeigt:

Die Fehlermeldung und der genaue Ort des Fehlers wird in einer Ballon-Hilfe angezeigt. In der markierten Zeile sind zu diesem Zeitpunkt alle Defines aufgelöst.

Folgende Tasten und Tastenkombinationen können im Editor verwendet werden:

Navigation

- Der Cursor kann mit den Pfeiltasten im Text bewegt werden.
- und positioniert den Cursor an den Anfang des folgenden oder des vorigen Wortes.
- und scrollt den Text spaltenweise nach rechts bzw. nach links.
- und positioniert innerhalb einer Zeile auf das erste oder letzte Zeichen, das kein Leerzeichen ist. Beim erneuten Drücken der gleichen Taste wird auf das erste oder letzte Zeichen der Zeile positioniert.
- und blättert eine Bildschirmseite.
- und positioniert auf den Anfang bzw. auf das Ende des gesamten Textes.
- , , und scrollt den Inhalt des Editors, ohne den Cursor zu verschieben.
- versieht die aktuelle Zeile mit einem Lesezeichen zu dem zu einem späteren Zeitpunkt mit bzw. gesprungen werden kann.
- oder wechselt zwischen den geöffneten Dokumenten.
- oder wechselt zwischen den geöffneten Dokumenten.
- wechselt zum vorherigen Dokument in der Navigationshistorie.
- wechselt zum nächsten Dokument in der Navigationshistorie.
- und wechselt zwischen der öffnenden und schließenden Klammer, wenn der Cursor auf einer Klammer steht (unabhängig davon, welche

Kontakt

Klammern verwendet werden).

- **[alt] + [1]** ... **[alt] + [4]** wechselt zwischen den Bereichen, wenn ein Text in verschiedene Bereiche aufgeteilt wurde.

Markieren

- Zum Markieren können die Tastenkombinationen zur Navigation bei gedrückter **[Home]**-Taste verwendet werden.
- Ein Bereich kann durch Ziehen mit der Maus markiert werden. Soll ein rechteckiger Bereich (nicht zeilenweise) markiert werden, kann die Maus mit gedrückten **[alt]** + **[Home]**- bzw. **[strg]** + **[Home]**-Tasten verwendet werden.
- **[strg]** + **[A]** markiert den gesamten Text.
- **[alt]** + **[Home]** markiert einen rechteckigen Bereich.
- **[strg]** + **[Home]** markiert einen rechteckigen Bereich (entspricht **[alt]** + **[Home]**).
- Sind mehrere Zeilen markiert, können diese durch **[Hilf]** oder **[strg]** + **[Hilf]** ein- bzw. ausgerückt werden.
- Mit **[Home]** oder **[Ende]** kann eine rechteckige Markierung links oder rechts verschoben werden.
- **[strg]** + **[alt]** + **[pos1]** oder **[strg]** + **[alt]** + **[ende]** markiert zwischen der öffnenden und schließenden Klammer, wenn der Cursor auf einer Klammer steht (unabhängig davon, welche Klammern verwendet werden).

Suchen / Ersetzen

- **[strg]** + **[F]** Aktiviert die Suchleiste mit der Funktion "Suchen".
- **[strg]** + **[Home]** + **[G]** Aktiviert die Suchleiste mit der Funktion "Gehe zu Funktion".
- **[strg]** + **[G]** Aktiviert die Suchleiste mit der Funktion "Gehe zu Zeile".
- **[strg]** + **[H]** Aktiviert die Suchleiste mit der Funktion "Ersetzen".
- **[strg]** + **[F3]** übernimmt aktuelle Markierung als Suchbegriff und positioniert auf den nächsten Suchtreffer (Vorwärts).
- **[F3]** positioniert auf den nächsten Suchtreffer (Vorwärts).
- **[Home]** + **[F3]** positioniert auf den nächsten Suchtreffer (Rückwärts).
- **[F4]** positioniert im nächsten gefundenen Dokument im Assistenten auf den ersten Suchtreffer (Vorwärts).
- **[Home]** + **[F4]** positioniert im vorherigen gefundenen Dokument im Assistenten auf den ersten Suchtreffer (Rückwärts).

Ausschneiden / Kopieren / Einfügen

- **[strg]** + **[X]** oder **[Home]** + **[entf]** schneidet den markierten Textbereich aus.
- **[strg]** + **[C]** oder **[strg]** + **[einfgl]** kopiert den markierten Textbereich.
- **[strg]** + **[V]** oder **[Home]** + **[einfgl]** fügt den ausgeschnittenen/kopierten Textbereich ein.
- **[strg]** + **[Home]** + **[D]** löscht die aktuelle Zeile.

Öffnen / Speichern / Übersetzen

Kontakt

- zeigt den Öffnen-Dialog für Dokumente an.
- legt eine neue Prozedur an.
- + + legt einen neuen Text an.
- speichert das aktuelle Dokument.
- speichert und übersetzt die aktuelle Prozedur.

Hilfe

- + öffnet eine Liste aller Prozedurbefehle und Konstanten. Die Befehle und Konstanten können direkt in den Editor übernommen werden.
- aktiviert die kontextsensitive Hilfe bzw. zeigt die Datenstruktur an. Dabei muss sich der Cursor auf einem Prozedurbefehl, einer Konstanten oder auf einem Datenbankfeld befinden.

Folding

- einzelnen Bereich zuklappen.
- einzelnen Bereich aufklappen.
- alle Bereiche zuklappen.
- alle Bereiche aufklappen.

Zoom

- oder Ansicht vergrößern.
- oder Ansicht verkleinern.
- Ansicht zurücksetzen.

Sonstiges

- löscht den Text bis zum Ende des Wortes.
- oder schließt das Dokument.
- führt die Startprozedur aus.
- blendet Leerzeichen ein/aus
- + wandelt den markierten Bereich in Großbuchstaben.
- wandelt den markierten Bereich in Kleinbuchstaben.
- setzt/löscht ein Lesezeichen.
- springt zum nächsten Lesezeichen.
- springt zum vorherigen Lesezeichen.
- setzt/löscht einen Breakpoint.
- springt zum nächsten Breakpoint.
- springt zum vorherigen Breakpoint.
- aktiviert und deaktiviert die Anzeige der zusammengehörenden Klammern.
- zeigt die aktuelle Funktion in der Statuszeile an.
- fügt das aktuelle Datum an der Cursorposition ein.

Kontakt

-  +  blendet die Anzeige der Zeilenummern ein bzw. aus.
-  +  kommentiert die markierten Zeilen aus oder entfernt die Kommentarzeichen. Ist mindestens eine Zeile nicht auskommentiert, werden alle Zeilen auskommentiert.
-  +  +  schaltet den Bearbeitungsmodus aus bzw. ein.
-  +  fügt unabhängig vom automatischen Klammerungsmodus einen Zeilenumbruch ein.

Kontakt

Statuszeile des Editors

Statuszeile des Editors

Editierbereich

Siehe des Editors,

Editor

In der Statuszeile werden Informationen zum Editor angezeigt.



Im ersten Bereich wird der Dokumenttyp angezeigt. Anhand des dargestellten Symbols kann der Status des Dokumentes unterschieden werden:

- - Geändertes Dokument
- - Gespeichertes Dokument
- - Kompilierte Prozedur
- - Prozedur mit Compiler-Fehler

In den folgenden beiden Bereichen wird die Position des Cursors im aktuellen Text dargestellt. In den folgenden Feldern wird "NUM" bzw. "CAPS" angezeigt, wenn die Taste bzw. eingerastet ist. Über die Taste kann zwischen dem Einfügen-Modus und dem Überschreiben-Modus gewechselt werden. Wird der bestehende Text überschrieben ist im sechsten Bereich "ÜB" eingetragen.

Im letzten Feld steht die Anzahl der Zeilen. Ist das Dokument eine Prozedur, stehen hier nach der Übersetzung zusätzliche Informationen: Die Anzahl der übersetzten Textzeilen, die Größe des erzeugten Prozedurcodes, Referenzen und Symbole.

Fehlermeldungen beim Übersetzen von Prozeduren

Fehlermeldungen beim Übersetzen mit dem Editor oder der Anweisung ProcCompile()

Werden fehlerhafte Prozeduren übersetzt, erscheint eine Ballonhilfe mit der entsprechenden Fehlermeldung. In der Beschreibung befindet sich der deutsche Fehlertext, der englische Fehlertext, sowie die Konstante und der Wert des Rückgabewertes von ProcCompile(). Der Fehlertext kann mit der Anweisung ErrMapText() aus dem Fehlerwert ermittelt werden.

Folgende Fehler werden ausgegeben:

- **@ENDIF fehlt**

Fehlermeldung (deutsch) @ENDIF fehlt.

Fehlermeldung (englisch) Missing @ENDIF directive.

Fehlerkonstante _ErrCplNoENDIF

Wert -154

Nach einer @ifdef-Anweisung steht kein @endif.

- **Anweisung fehlt**

Fehlermeldung (deutsch) Anweisung fehlt.

Fehlermeldung (englisch) Missing instruction.

Fehlerkonstante _ErrCplNoStatement

Wert -139

An der betreffenden Stelle wird eine Anweisung erwartet. Eine leere Anweisung kann mit ; angegeben werden. Die Meldung wird ebenfalls ausgegeben, wenn zum Beispiel ein globaler Datenbereich den gleichen Namen wie eine Prozedur besitzt.

- **Anweisung unzulässig**

Fehlermeldung (deutsch) Anweisung unzulässig.

Fehlermeldung (englisch) Invalid instruction.

Fehlerkonstante _ErrCplInvalidStatement

Wert -119

Dieser Fehler tritt auf, wenn innerhalb eines try-Blocks ein weiterer try-Block beginnt oder die Befehle break bzw. cycle außerhalb einer Schleife verwendet werden.

- **Argumente fehlen**

Fehlermeldung (deutsch) Argumente fehlen.

Fehlermeldung (englisch) Missing Arguments.

Fehlerkonstante _ErrCplNoArguments

Wert -127

Dieser Fehler kann nur bei Definitionen auftreten. Die Definition ist mit Argumenten definiert. In der Prozedur wird sie aber ohne Argumente aufgerufen.

- **Ausdruck ist nicht konstant**

Fehlermeldung (deutsch) Ausdruck ist nicht konstant.

Fehlermeldung (englisch) Expression not constant.

Fehlerkonstante _ErrCplExpressionNotConst

Wert -109

An der betreffenden Stelle muss ein konstanter Ausdruck stehen. In einem konstanten Ausdruck kommt keine Variable vor. Ein konstanter Ausdruck wird bei der Deklaration von der Länge von Zeichenketten und bei der Anzahl von Array-Elementen benötigt.

• **BEGIN fehlt**

Fehlermeldung (deutsch) BEGIN fehlt.

Fehlermeldung (englisch) Missing BEGIN.

Fehlerkonstante _ErrCplNoBegin

Wert -129

Zu Beginn einer Sektion (zum Beispiel local, global oder define muss ein BEGIN (PASCAL-Stil) bzw. { (C-Stil) stehen. Ebenso wird nach den Anweisungen main, sub und try ein BEGIN bzw. { erwartet.

• **CASE fehlt**

Fehlermeldung (deutsch) CASE fehlt.

Fehlermeldung (englisch) Missing CASE.

Fehlerkonstante _ErrCplNoCASE

Wert -150

Die Fehlermeldung tritt nur bei Verwendung des C-Stils auf. In einem switch-Konstrukt steht vor einem Fall kein case.

• **Compileranweisung ungültig**

Fehlermeldung (deutsch) Compileranweisung ungültig.

Fehlermeldung (englisch) Compiler instruction invalid.

Fehlerkonstante _ErrCplInvalidMetaCommand

Wert -124

Es wurde eine ungültige Anweisung mit vorangestelltem @ angegeben. Die gültigen Compiler-Anweisungen sind im Abschnitt Compiler-Anweisungen beschrieben.

Der Fehler tritt ebenfalls auf, wenn @else oder @endif ohne vorheriges @ifdef oder @ifndef ohne Bedingung aufgerufen wird. Er wird auch bei der Überschreitung der maximalen Schachtelungstiefe der Befehle ausgegeben.

• **Datenlimit überschritten**

Fehlermeldung (deutsch) Datenlimit überschritten.

Fehlermeldung (englisch) Data limit exceeded.

Fehlerkonstante _ErrCplDataOverflow

Wert -146

In einem Datenbereich können maximal 4 MB zur Deklaration von Variablen verwendet werden. Dieser Wert wurde überschritten. Die Fehlermeldung kommt auch dann, wenn die Startposition einer Variable nicht innerhalb der ersten 2 MB des Bereichs liegt. Möglicherweise müssen große Arrays dynamisch angelegt werden.

• **Definition unvollständig**

Fehlermeldung (deutsch) Definition unvollständig.

Fehlermeldung (englisch) Definition incomplete.

Fehlerkonstante _ErrCplIncompleteDefine

Wert -110

Eine Definition (define) ist vollständig definiert, wird aber im Prozedurtext nicht vollständig angegeben. Der Fehler tritt nur bei Definitionen mit Argumenten auf, wenn im Prozedurtext die Definition ohne schließende Klammer bzw. mit zu wenigen Parametern angegeben ist.

• **DO fehlt**

Fehlermeldung (deutsch) DO fehlt.

Fehlermeldung (englisch) Missing DO.

Fehlerkonstante _ErrCplNoDO

Wert -132

Die Fehlermeldung tritt nur bei Verwendung des PASCAL-Stils auf. Die Anweisungen for bzw. while erfordern die Angabe von DO.

• **Doppelpunkt fehlt**

Fehlermeldung (deutsch) Doppelpunkt fehlt.

Fehlermeldung (englisch) Colon missing.

Fehlerkonstante _ErrCplNoColon

Wert -147

Dieser Fehler kann folgende Ursachen haben:

- ◆ Innerhalb eines Define ist der Name von dem Ersetzungstext nicht durch ein Doppelpunkt getrennt.
- ◆ In einem switch-Konstrukt steht nach dem Ausdruck hinter case kein Doppelpunkt vor der ersten Anweisung.
- ◆ In einer Variablen-Deklaration steht zwischen dem Namen der Variablen und dem Typ kein Doppelpunkt.

• **Eckige Klammer fehlt**

Fehlermeldung (deutsch) Eckige Klammer fehlt.

Fehlermeldung (englisch) Missing Bracket.

Fehlerkonstante _ErrCplNoBracket

Wert -130

Bei der Definition eines Array oder in einem Ausdruck mit einem Array ist nur eine oder keine eckige Klammer angegeben. Bei einer Zuweisung oder bei der Parameterübergabe wird nur ein Element des Arrays erwartet.

• **END fehlt**

Fehlermeldung (deutsch) END fehlt.

Fehlermeldung (englisch) Missing END.

Fehlerkonstante _ErrCplNoEndCase

Wert -133

Das switch-Konstrukt wurde nicht mit einem END bzw. } abgeschlossen.

• **Falsche Version der Übersetzungstabelle**

Fehlermeldung (deutsch) Falsche Version der Übersetzungstabelle.

Fehlermeldung (englisch) Wrong compile table version.

Fehlerkonstante _ErrCplTokenTableVersion

Wert -101

Die Version der Übersetzungstabelle c16.tla passt nicht zur Version des Clients.

• **Fehler beim Lesen der Übersetzungstabelle**

Fehlermeldung (deutsch) Fehler beim Lesen der Übersetzungstabelle.

Kontakt

Fehlermeldung (englisch) Error reading compile table.

Fehlerkonstante _ErrCplTokenTableRead

Wert -102

Beim Lesen der Übersetzungstabelle c16.tla ist ein Fehler aufgetreten.

• Feld nicht vorhanden

Fehlermeldung (deutsch) Feld nicht vorhanden.

Fehlermeldung (englisch) Field not existing.

Fehlerkonstante _ErrCplFldNotFound

Wert -148

Wird ein Feld über seine Nummer ("100,1,3") angesprochen, erscheint der Fehler, wenn keine Datei, kein Teildatensatz oder kein Feld mit der Nummer existiert oder nicht alle Nummern angegeben wurden.

• Funktionscode fehlt

Fehlermeldung (deutsch) Funktionscode fehlt.

Fehlermeldung (englisch) function code missing.

Fehlerkonstante _ErrCplUndefinedProcedure

Wert -151

In der Prozedur ist eine Funktion mit declare deklariert, die in der gleichen Prozedur vorhanden sein soll. In der Prozedur ist sie allerdings nicht definiert.

• Hauptspeicher nicht ausreichend

Fehlermeldung (deutsch) Hauptspeicher nicht ausreichend.

Fehlermeldung (englisch) Not enough memory.

Fehlerkonstante _ErrCplTokenTableOutOfMemory

Wert -103

Beim Lesen der Übersetzungstabelle c16.tla konnte nicht genügend Hauptspeicher allokiert werden.

• Konstante ungültig

Fehlermeldung (deutsch) Konstante ungültig.

Fehlermeldung (englisch) Invalid constant.

Fehlerkonstante _ErrCplInvalidConstant

Wert -113

Der Typ der Konstante konnte nicht ermittelt werden, oder die Konstante enthält ungültige Werte. Zum Beispiel darf in einer Zeitkonstanten für die Anzahl der Stunden kein Wert über 24, für die Anzahl der Minuten und Sekunden kein Wert über 59 angegeben werden.

• Komma fehlt

Fehlermeldung (deutsch) Komma fehlt.

Fehlermeldung (englisch) Missing Comma.

Fehlerkonstante _ErrCplNoComma

Wert -131

Bei einem Befehl oder einer Funktion wurden weniger Argumente übergeben, als notwendig sind.

• Kommentarende fehlt

Fehlermeldung (deutsch) Kommentarende fehlt.

Kontakt

Fehlermeldung (englisch) Missing end of comment.

Fehlerkonstante _ErrCplOpenComment
Wert -141

Ein Kommentar wurde mit /* eingeleitet aber nicht wieder beendet.

• Label unbekannt

Fehlermeldung (deutsch) Label unbekannt.

Fehlermeldung (englisch) Unknown label.

Fehlerkonstante _ErrCplUndefinedLabel
Wert -121

In einem Ausdruck befindet sich ein nicht zuvor definiertes Label (vgl. [try](#)).

• Label unzulässig

Fehlermeldung (deutsch) Label unzulässig.

Fehlermeldung (englisch) Invalid label.

Fehlerkonstante _ErrCplInvalidLabel
Wert -115

Das angegebene Label enthält ungültige Zeichen.

• LOOP fehlt

Fehlermeldung (deutsch) LOOP fehlt.

Fehlermeldung (englisch) Missing LOOP.

Fehlerkonstante _ErrCplNoLOOP
Wert -134

In einer for-Schleife wurde keine loop-Anweisung angegeben.

• Name unbekannt

Fehlermeldung (deutsch) Name unbekannt.

Fehlermeldung (englisch) Unknown name.

Fehlerkonstante _ErrCplUnknownName
Wert -126

Der angegebene Name ist nicht definiert. In der Regel handelt es sich hier um einen Schreibfehler oder eine vergessene Variablendeclaration. Der Fehler tritt ebenfalls auf, wenn eine Funktion aufgerufen werden soll, die erst zu einem späteren Zeitpunkt definiert wird (siehe [declare](#)).

Wird ein Feldname mit Sonderzeichen oder Umlauten in den Prozedurtext geschrieben, erscheint ebenfalls diese Fehlermeldung. Feldnamen mit Umlauten sollten nicht verwendet werden, da es beim Zugriff von anderen Programmen auf die CONZEPT 16-Datenbank zu Problemen kommen kann. Diese Feldnamen können aber in CONZEPT 16 verwendet werden, wenn sie in " eingeschlossen werden (z. B. "ADR.STRÄßE").

• Name ungültig

Fehlermeldung (deutsch) Name ungültig.

Fehlermeldung (englisch) Invalid name.

Fehlerkonstante _ErrCplInvalidName
Wert -116

Es wurde ein Name angegeben, der bereits definiert ist. Der Fehler kann überall auftreten, wo Namen definiert werden (in Defines, Deklaration von

Kontakt

Funktionen oder Variablen usw.). Der Namenskonflikt entsteht entweder durch Verwendung eines Namens, der bereits im Sprachumfang von CONZEPT 16 enthalten ist, oder durch nochmaliges Verwenden eines Namens einer bereits definierten Variablen oder eines Feldes.

Der Fehler tritt ebenfalls auf, wenn in einem try-Block ein Label mehrfach definiert oder bei einem zusammengesetzten Datentyp auf eine nicht vorhandene Komponente referenziert wird.

• **Name zu lang**

Fehlermeldung (deutsch) Name zu lang.

Fehlermeldung (englisch) Name too long.

Fehlerkonstante _ErrCplNameTooLong

Wert -125

Prozedurnamen haben eine maximale Länge von 20 Zeichen. Funktionsnamen dürfen 40 Zeichen nicht überschreiten. Für alle anderen Namen gilt eine Beschränkung auf maximal 64 Zeichen. Wird die entsprechende Länge überschritten, erfolgt dieser Fehler.

• **OF fehlt**

Fehlermeldung (deutsch) OF fehlt.

Fehlermeldung (englisch) Missing OF.

Fehlerkonstante _ErrCplNoOF

Wert -135

Die Fehlermeldung tritt nur bei Verwendung des PASCAL-Stils auf. Die Anweisungen CASE erfordert die Angabe von OF.

• **Operanden sind ungleich**

Fehlermeldung (deutsch) Operanden sind
ungleich.

Fehlermeldung (englisch) Operands are different.

Fehlerkonstante _ErrCplOperandTypes

Wert -145

Die Typen zweier Ausdrücke sind nicht gleich. Dies tritt zum Beispiel auf, wenn ein Wert vom Typ float mit einem Wert vom Typ int multipliziert wird. Einer der beiden Werte muss mit dem entsprechenden Konvertierungsbefehle umgewandelt werden.

• **Operand fehlt**

Fehlermeldung (deutsch) Operand fehlt.

Fehlermeldung (englisch) Missing operand.

Fehlerkonstante _ErrCplNoOperand

Wert -136

In einer switch-Anweisung steht nach dem case kein Ausdruck. Eine Funktion ohne Rückgabewert wird in einer Zuweisung verwendet.

• **Prozedurcode ist zu lang**

Fehlermeldung (deutsch) Prozedurcode ist zu
lang.

Fehlermeldung (englisch) Procedure code too
long.

Fehlerkonstante _ErrCplCodeOverflow

Wert -106

Die Prozedur kann nicht übersetzt werden, da sie zu lang ist. Bei der Übersetzung wird mehr Code erzeugt, als in den Puffer für Prozeduren passt.

Der Übersetzungsfehler tritt auch dann auf, wenn eine Sprungadresse mehr als 32 KB entfernt ist. Dies ist dann der Fall, wenn innerhalb eines switch-, if- oder Schleifen-Konstruktes sehr viel Prozedurcode steht.

• **Prozedur unbekannt**

Fehlermeldung (deutsch) Prozedur unbekannt.

Fehlermeldung (englisch) Procedure unknown.

Fehlerkonstante _ErrCplUnknownProcedure

Wert -142

Die angegebene Prozedur konnte nicht gefunden werden, da sie noch nicht übersetzt wurde. Wird die Prozedur übersetzt oder mit declare deklariert, tritt der Fehler nicht mehr auf.

• **Prozedur in Bearbeitung**

Fehlermeldung (deutsch) Prozedur in Bearbeitung.

Fehlermeldung (englisch) Procedure currently processed.

Fehlerkonstante _ErrCplProcedureLocked

Wert -105

Die Prozedur wird gerade von einem Benutzer verändert.

• **Prozedur nicht vorhanden**

Fehlermeldung (deutsch) Prozedur nicht vorhanden.

Fehlermeldung (englisch) Procedure does not exist.

Fehlerkonstante _ErrCplProcedureNotFound

Wert -104

Die Prozedur wurde nicht gefunden. Häufig ist nur ein falscher Name angegeben.

• **Runde Klammer fehlt**

Fehlermeldung (deutsch) Runde Klammer fehlt.

Fehlermeldung (englisch) Missing parenthesis.

Fehlerkonstante _ErrCplNoParenthesis

Wert -137

- ◆ Eine öffnende runde Klammer bedingt immer eine schließende runde Klammer.
- ◆ Die Argumente eines Befehls oder einer Funktion werden in einer runden Klammer hinter dem Namen angegeben. Bei dieser Fehlermeldung fehlt die schließende Klammer, oder es wurden mehr Argumente als zulässig übergeben.
- ◆ Bei der Verwendung des C-Stils muss der Ausdruck hinter einer if- oder switch-Anweisung immer geklammert werden.
- ◆ Die Eigenschaften LclDateDay... und LclDateMonth... erwarten einen Parameter, der eingeklammert werden muss.

• **Sektion unzulässig**

Fehlermeldung (deutsch) Sektion unzulässig.

Fehlermeldung (englisch) Invalid section.

Kontakt

Fehlerkonstante	_ErrCplInvalidSection
Wert	-118

Dieser Fehler tritt auf, wenn innerhalb einer Prozedur oder einer Funktion zwei lokale Datenbereiche angelegt wurden oder in einer Prozedur zweimal eine main-Funktion deklariert wurde. Ebenfalls ist die Verwendung eines bereits existierenden Namens als Sektion unzulässig.

• Semikolon fehlt

Fehlermeldung (deutsch) Semikolon fehlt.

Fehlermeldung (englisch) Missing semicolon.

Fehlerkonstante	_ErrCplNoSemicolon
-----------------	--------------------

Wert	-138
------	------

Die Deklaration einer Variablen muss mit einem Semikolon enden. Bei der Deklaration von Argumenten (sub, declare usw.) müssen die Argumente mit Semikolon getrennt werden.

• THEN fehlt

Fehlermeldung (deutsch) THEN fehlt.

Fehlermeldung (englisch) Missing THEN.

Fehlerkonstante	_ErrCplNoTHEN
-----------------	---------------

Wert	-149
------	------

Die Fehlermeldung tritt nur bei Verwendung des PASCAL-Stils auf. Die Anweisungen IF erfordert die Angabe von THEN.

• Typ des Arguments ist falsch

Fehlermeldung (deutsch) Typ des Arguments ist falsch.

Fehlermeldung (englisch) Wrong argument type.

Fehlerkonstante	_ErrCplInvalidArgumentType
-----------------	----------------------------

Wert	-111
------	------

Dieser Fehler wird immer dann ausgegeben, wenn als Argument ein Wert von einem anderen Typ erwartet wurde. Beispiele:

- ◆ Vor dem Operator -> steht keine Variable vom Typ int bzw. handle oder kein Objekt-Deskriptor.
- ◆ Das Argument ist als var-Parameter deklariert, wird aber als "Call-by-Value" übergeben.
- ◆ Bei der Funktionsdeklaration ist ein Parameter als optional (opt) angegeben und ein nachfolgender Parameter ist nicht als opt definiert.

• Typ des Ausdrucks ist falsch

Fehlermeldung (deutsch) Typ des Ausdrucks ist falsch.

Fehlermeldung (englisch) Type of expression is wrong.

Fehlerkonstante	_ErrCplInvalidExpressionType
-----------------	------------------------------

Wert	-114
------	------

Bei einer Zuweisung hat die linke Seite (vom Zuweisungsoperator #) einen anderen Typ als die rechte Seite.

Bei den Funktionen Min() oder Max() sind bei den Argumenten unterschiedliche Typen (zum Beispiel int und float) angegeben worden.

• Typ des Operanden ist falsch

Fehlermeldung (deutsch) Typ des Operanden ist falsch.

Kontakt

Fehlermeldung (englisch) Wrong operand type.

Fehlerkonstante _ErrCplInvalidOperatorType

Wert -117

Es soll eine Operation durchgeführt werden, die auf dem entsprechenden Typ nicht definiert ist.

• Typ unbekannt

Fehlermeldung (deutsch) Typ unbekannt.

Fehlermeldung (englisch) Unknown type.

Fehlerkonstante _ErrCplUnknownType

Wert -144

Bei der Programmierung der COM-Schnittstelle wird auf eine Eigenschaft mit einem unbekannten Typ (nicht cpa, cpi ... siehe COM-Befehle) zugegriffen. Bei der Deklaration einer Variablen wurde für den Typ ein Name verwendet, der keinem Typen entspricht.

• Übersetzungstabelle nicht definiert

Fehlermeldung (deutsch) Übersetzungstabelle nicht definiert.

Fehlermeldung (englisch) Compile table not defined.

Fehlerkonstante _ErrCplTokenTableUndefined

Wert -100

Die Übersetzungstabelle c16.tla konnte nicht geöffnet werden.

• Unerwartetes Textende

Fehlermeldung (deutsch) Unerwartetes Textende.

Fehlermeldung (englisch) Unexpected end of text.

Fehlerkonstante _ErrCplUnexpectedEndOfText

Wert -108

Eine Sektion innerhalb der Prozedur ist nicht abgeschlossen. Eine Sektion (zum Beispiel global, local, define, main oder sub) wurde eingeleitet aber nicht mit der geschweiften Klammer abgeschlossen. An der betreffenden Stelle fehlen ein oder mehrere }`.

• Ungültige Eigenschaft

Fehlermeldung (deutsch) Ungültige Eigenschaft.

Fehlermeldung (englisch) Invalid property.

Fehlerkonstante _ErrCplInvalidProperty

Wert -153

In der Anweisung ist eine Konstante angegeben worden, die keine Eigenschaft repräsentiert.

• Variable abweichend

Fehlermeldung (deutsch) Variable abweichend.

Fehlermeldung (englisch) Variable differs.

Fehlerkonstante _ErrCplArgumentMismatch

Wert -152

Die Definition einer sub-Funktion weicht von der Deklaration in der declare-Anweisung ab. Der Fehler tritt auf, wenn zum Beispiel ein anderer Return-Wert, andere Variablen als optionale Parameter oder Variablen mit unterschiedlichem Typ definiert sind.

• Wert unzulässig

Fehlermeldung (deutsch) Wert unzulässig.

Fehlermeldung (englisch) Invalid value.

Fehlerkonstante _ErrCplConstantValue

Wert -107

Bei der Deklaration von alpha-Werten kann die Länge von Zeichenketten im Bereich 1 bis 8192 angegeben werden. Arrays müssen mindestens mit einem Element definiert werden. Werden diese Vorgaben nicht eingehalten, erscheint diese Fehlermeldung.

• WHILE fehlt

Fehlermeldung (deutsch) WHILE fehlt.

Fehlermeldung (englisch) Missing WHILE.

Fehlerkonstante _ErrCplNoWHILE

Wert -140

Bei einer for-Schleife fehlt die while-Anweisung.

• Wort unbekannt

Fehlermeldung (deutsch) Wort unbekannt.

Fehlermeldung (englisch) Unknown word.

Fehlerkonstante _ErrCplInvalidWord

Wert -120

Aus gültigen Zeichen wurde eine Kombination gebildet, die nicht Bestandteil des Sprachumfangs ist (zum Beispiel <>).

• Zeichen ungültig

Fehlermeldung (deutsch) Zeichen ungültig.

Fehlermeldung (englisch) Invalid character.

Fehlerkonstante _ErrCplInvalidChar

Wert -112

In der Prozedur wurde ein ungültiges Zeichen angegeben. Ungültige Zeichen sind alle Zeichen der ASCII-Tabelle mit einem Wert kleiner als 32 und größer als 126. Dazu kommt noch das Zeichen @, wenn es nicht am Anfang der Zeile steht.

• Zeile zu lang

Fehlermeldung (deutsch) Zeile zu lang.

Fehlermeldung (englisch) Line too long.

Fehlerkonstante _ErrCplLineOverflow

Wert -123

Durch die Ersetzung von Defines ist eine Zeile entstanden, die länger als 2048 Zeichen ist. Ohne die Ersetzung von Defines darf die Zeile nur 250 Zeichen lang sein.

• Zuviele Argumente

Fehlermeldung (deutsch) Zuviele Argumente.

Fehlermeldung (englisch) Too many arguments.

Fehlerkonstante _ErrCplTooManyArguments

Wert -143

Ein Define ist mit mehr Argumenten definiert worden, als es aufgerufen wird.

Der gleiche Fehler tritt auf, wenn ein Define oder eine Funktion mit mehr als 64 Argumenten definiert wurde.

• **Zuweisung fehlt**

Fehlermeldung (deutsch) Zuweisung fehlt.

Fehlermeldung (englisch) Missing assignment.

Fehlerkonstante _ErrCplNoAssign

Wert -128

Die Anweisung besteht nur aus einer Variable, einem Feld oder einer Eigenschaft.

• **Zuwenig Argumente**

Fehlermeldung (deutsch) Zuwenig Argumente.

Fehlermeldung (englisch) Too few arguments.

Fehlerkonstante _ErrCplMissingArgument

Wert -122

Bei einem Funktionsaufruf sind zu wenig Argumente angegeben.

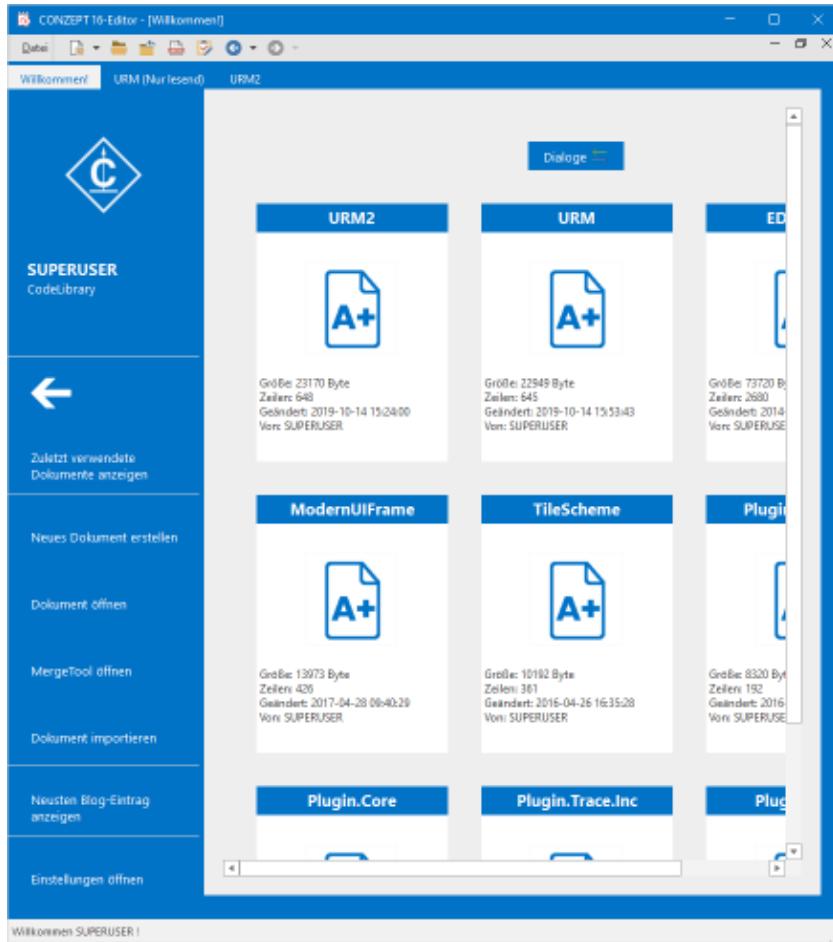
Kontakt

Willkommenseite des Editors

Schnellzugriff auf wichtige Funktionen des Editors

Siehe [Editor](#)

Die Willkommenseite fasst wichtige Aktionen des [Editors](#) zusammen. Sie wird immer gestartet, wenn kein Dokument im [Editor](#) aktiv ist. Dieses Verhalten kann in den [Einstellungen](#) deaktiviert werden.



Im linken Bereich der Willkommenseite werden der aktuelle Benutzer und die Datenbank angezeigt. Darunter befinden sich die wichtigsten Aktionen des [Editors](#). Folgende Aktionen sind möglich:

Pfeil links

War zuvor ein anderes Dokument ausgewählt, wird eine Pfeil-Schaltfläche angezeigt. Durch Anklicken der Schaltfläche wird zum zuvor ausgewählten Dokument gewechselt.

Zuletzt verwendete Dokumente anzeigen

Durch Druck dieser Schaltfläche wird im rechten Bereich eine Liste der zuletzt verwendeten Prozeduren und Texte angezeigt. Diese können durch Klick direkt geöffnet werden. Über die Schaltfläche Dialoge / Prozeduren, welche sich oben im rechten Bereich befindet, wird die Ansicht zwischen Dokumenten und Dialogen und

Kontakt

Druckformularen gewechselt.

Neues Dokument erstellen

Bei Klick der Schaltfläche öffnet sich ein Kontextmenü, in dem der Typ des neuen Dokumentes ausgewählt werden kann.

Dokument öffnen

Diese Schaltfläche öffnet den Dokument öffnen-Dialog, in dem die vorhandenen Prozeduren und Texte in einer Liste zur Auswahl stehen.

MergeTool öffnen

Ein Klick dieser Schaltfläche öffnet eine neue Ansicht des MergeTools. Dieses dient dem Vergleich von Prozeduren und Texten.

Dokument importieren

Durch Klick dieser Schaltfläche wird ein Dialog zum Importieren von Dokumenten gestartet.

Neusten Blog-Eintrag anzeigen

Diese Schaltfläche bewirkt, dass im rechten Bereich der aktuellste Blog-Eintrag aus dem CONZEPT 16-Blog angezeigt wird.

Einstellungen öffnen

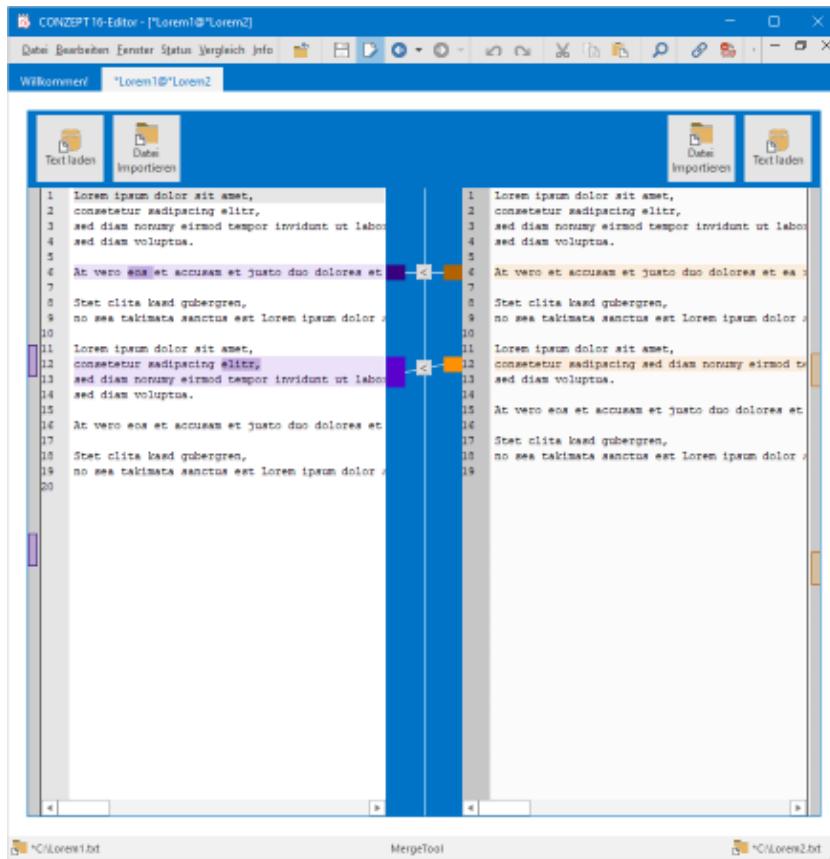
Ein Klick dieser Schaltfläche öffnet den Einstellungsdialog von CONZEPT 16.

Kontakt

MergeTool

Tool zum Vergleichen von Prozeduren und Texten

Im Editor kann mit der Tastenkombination **strg** + **alt** + **M** das MergeTool aufgerufen werden. Dieses dient dem Vergleich von Prozeduren und Texten. Die Prozeduren und Texte können hierbei sowohl in der Datenbank, als auch als externe Dateien vorliegen. Das MergeTool sieht wie folgt aus:



Das MergeTool ist in zwei Bereiche aufgeteilt. In jedem Bereich kann ein Dokument angezeigt werden. Dies können z. B. zwei unterschiedliche Versionen einer Prozedur sein. Das Dokument auf der rechten Seite kann nicht bearbeitet werden. Auf der linken Seite kann das Dokument bearbeitet werden, wenn der Benutzer über entsprechende Rechte auf dem Dokument verfügt, und dieses nicht bereits gesperrt ist. Ist das Dokument bereits durch den eigenen Benutzer gesperrt, erfolgt eine Abfrage, ob die Sperre übernommen werden soll.

Zur Verwendung des MergeTools werden die Menüpunkte Prozedur und Fenster durch die Menüpunkte Status und Vergleich ersetzt. Die Funktionen stehen auch über entsprechende Symbolleisten zur Verfügung.

Durch die Schaltfläche **Text laden** kann auf der jeweiligen Seite eine Prozedur oder ein Text aus der Datenbank geladen werden. Die Schaltfläche **Datei importieren** öffnet hingegen eine externe Datei.

Werden in beiden Ansichten Dokumente angezeigt, werden die Unterschiede ermittelt und hervorgehoben. Dabei werden die Zeilen mit Änderungen mit einem helleren

Kontakt

Farbton markiert. Die detaillierten Änderungen werden durch einen dunkleren Farbton zusätzlich hervorgehoben.

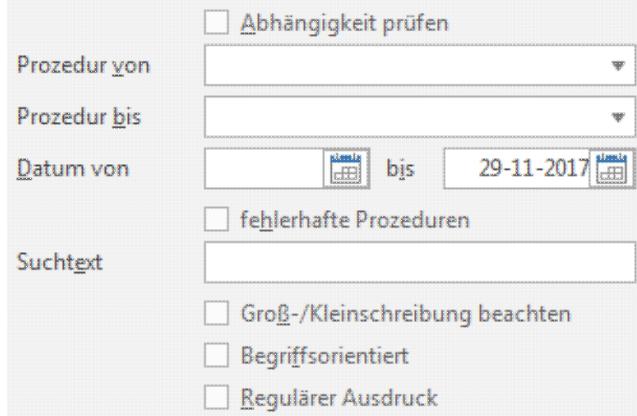
Einzelne Änderungen können über die -Schaltflächen zwischen den zwei Ansichten übernommen werden.

Designer - Assistent

Verwaltung bestehender Dokumente

Der Assistent kann über den Menüeintrag **Anzeige / Assistent** oder mit dem Tastenkürzel  +  bzw. aus dem Editor aktiviert werden.

Dokumentsuche



• Abhängigkeit prüfen (nur Prozeduren)

Ist diese Option gewählt, werden alle Prozeduren aufgelistet, bei denen folgende Bedingungen gegeben sind:

- ◆ Der Prozedurcode ist älter als der Quelltext, d. h. die Prozedur wurde nach Änderung gespeichert aber noch nicht übersetzt.
- ◆ Eine Prozedur bindet über die Include-Anweisung (@I:<name>) eine andere Prozedur ein und die eingebundene Prozedur wurde nach dem letzten Übersetzen der Hauptprozedur geändert.
- ◆ Eine Prozedur ruft die Funktion einer anderen Prozedur auf (über <Prozedurname>:<Funktionsname>) und die Prozedur der aufgerufenen Funktion wurde nach dem letzten Übersetzen der aufrufenden Prozedur geändert.

• Prozedur von / Prozedur bis / Text von / Text bis

An dieser Stelle lässt sich der Bereich der zu bearbeitenden Dokumente vorgeben. Wird keine Auswahl getroffen, werden alle Dokumente berücksichtigt.

• Datum von / Datum bis

Alle Dokumente die innerhalb des vorgegebenen Datumsbereiches liegen, werden berücksichtigt. Ist kein Datum vorgegeben, werden alle Prozeduren unabhängig des Datums bei der Verarbeitung berücksichtigt.

• fehlerhafte Prozeduren (nur Prozeduren)

Es werden alle Prozeduren aufgelistet, die aufgrund eines Fehlers nicht übersetzt werden können. Fehlerhafte Prozeduren werden in der Spalte "!"

Kontakt

entsprechend gekennzeichnet.

!	Name	Letzte Änderung
!	App_Main	29.11.2017 - 13:27
!	App_Knd	27.04.2016 - 14:42

- **Suchtext**

Es werden alle Dokumente aufgelistet, in denen der hier angegebene Text vorhanden ist.

- **Groß-/Kleinschreibung beachten**

Bei Verwendung dieser Option, wird eine Groß-/Kleinschreibung des Suchtextes beachtet. Der Suchbegriff wird dann gefunden, wenn er im Dokumenttext genauso geschrieben wurde.

- **begriffsorientiert**

Bei Verwendung dieser Option, wird der Suchtext nur als ganzes Wort und nicht als Wortbestandteil gesucht.

- **Regulärer Ausdruck**

Bei Verwendung dieser Option, wird der Suchtext als regulärer Ausdruck interpretiert.

Wird keine der Optionen verwendet, wird eine Liste mit allen Prozeduren bzw. Texte in der Datenbank erstellt.

!	Name	Letzte Änderung
!	App_Knd	27.04.2016 - 14:42
!	App_Main	29.11.2017 - 13:27
	App_Art	07.03.2011 - 12:05
	App_ArtSelect	28.08.2006 - 15:37
	App_Auf	23.03.2011 - 11:24
	App_Cmn	08.03.2011 - 13:55
	App_ComChart	08.03.2011 - 13:56
	App_ComExcel	09.03.2011 - 10:10
	App_ComMapPoint	09.09.2004 - 10:24
	App_ComOutlook	28.02.2011 - 09:36
	App_ComWord	10.03.2011 - 10:26

Spalten

In der Liste werden folgende Informationen angezeigt:

- **Spalte ! (nur Prozeduren)**

Kontakt

In dieser Spalte werden Statusinformationen ausgegeben. Folgende Symbole sind möglich:

- ◆ Prozedur wurde ohne Fehler übersetzt.
- ◆ Prozedur kann aufgrund eines Fehlers nicht übersetzt werden.
- ◆ Prozedur kann nicht übersetzt werden, da die Prozedur im Editor geändert, aber nicht gespeichert ist.
- ◆ Prozedur kann nicht übersetzt werden, da diese von einem anderen Benutzer in Bearbeitung ist.

• Spalte Folge

Diese Spalte steht nur dann zur Verfügung, wenn die Option "Abhängigkeit prüfen" gewählt ist. Es wird die Reihenfolge angezeigt, in welcher der Compiler die Prozeduren übersetzt, um möglichst in einem Durchlauf alle Abhängigkeiten auflösen zu können.

• Spalte Name

Name des Dokuments.

• Spalte Letzte Änderung

Datum und Uhrzeit der letzten Änderung.

• Spalte Zeilen

Anzahl der Zeilen.

• Spalte Größe

Größe in Byte.

• Spalte Geändert von

Benutzer der letzten Änderung.

• Spalte Letzte Übersetzung

Datum und Uhrzeit der letzten Übersetzung.

Die Sortierung der Liste kann durch einen Klick auf den jeweiligen Spaltenkopf geändert werden. Durch einen Doppelklick auf ein Dokument wird dieses automatisch in den Editor übernommen. Über den Button "Übersetzen" werden alle in der Liste befindlichen Prozeduren übersetzt.

Kontextmenü

Mit einem Rechtsklick auf eine der Prozeduren kann ein Kontextmenü geöffnet werden. Hier stehen die folgenden Funktionen zur Verfügung:

- **Öffnen** - Die selektierte Prozedur wird geöffnet.
- **Offnen (Nur lesend)** - Die selektierte Prozedur wird im Lesemodus geöffnet. Änderungen sind nicht möglich.
- **Kopieren** - Die Liste der Prozeduren wird in die Zwischenablage kopiert. Für jede Prozedur wird dabei eine Zeile angelegt. Die Spalten entsprechen den Spalten in der Ergebnisliste. Getrennt werden die einzelnen Spalten durch ein Semikolon (";"). Die Reihenfolge der Liste wird ebenfalls eingehalten. Beispiel:

OK;AFrmMain;2008-06-22;16:45;4343;128.7550K;MfrmArticles;2008-06-24;11:41;3486;102.8270K;M

- **Neue Suche** - Die Suchefelder im Kopfbereich des Assistenten werden geleert und das aktuelle Suchergebnis entfernt.
- **Starten** - Die Suche wird anhand der eingestellten Parameter erneut gestartet.
- **Übersetzen** - Alle angezeigten Prozeduren werden durch den Prozedur-Compiler übersetzt.
- **Prozeduren löschen** - Alle angezeigten Prozeduren werden gelöscht.
- **Selektierte Prozedur löschen** - Die selektierte Prozedur wird gelöscht.

Folgende Tasten und Tastenkombinationen können im Assistenten verwendet werden:

- **Assistent**

- ◆ schließt den Assistenten.
- ◆ führt die Suche des Assistenten aus.
- ◆ + positioniert im Editor in dem nächsten gefundenen Dokument auf den ersten Suchbegriff (Vorwärts).
- ◆ + im Editor in dem vorherigen gefundenen Dokument auf den ersten Suchbegriff (Rückwärts).

- **Liste der Dokumente**

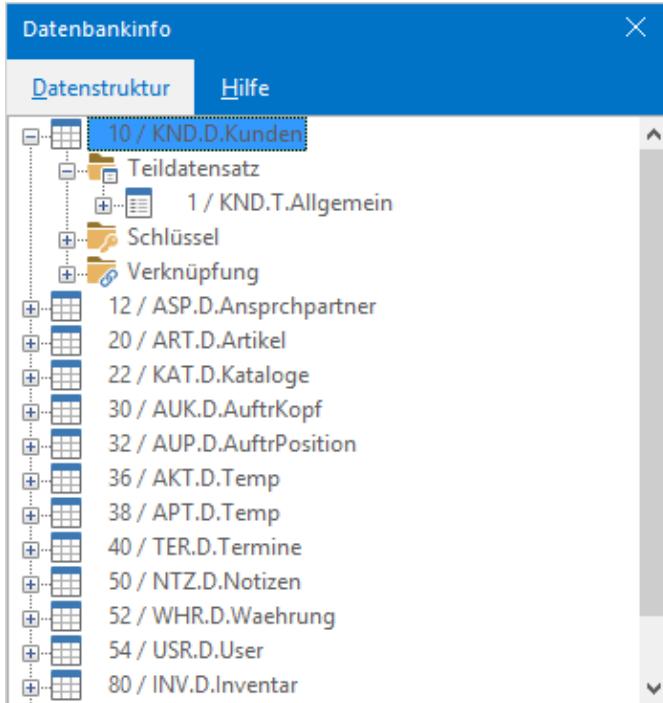
- ◆ + bzw. + kopiert die Liste der Dokumente in die Zwischenablage.
- ◆ löscht das selektierte Dokument.

Kontakt

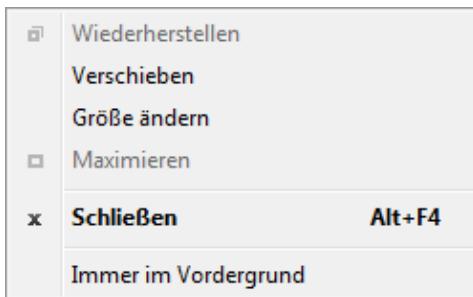
Designer - Datenbankinfo

Fenster in dem Elemente der Datenbank aufgeführt sind.

Im Fenster "Datenbankinfo" wird die Datenstruktur und die Online-Hilfe angezeigt.



Das Fenster verfügt über ein Kontextmenü, das durch Klicken mit der rechten Maustaste auf den Fenstertitel aufgerufen werden kann.



• Verschieben

Über diesem Menüeintrag kann das Fenster verschoben werden. Das Fenster kann ebenfalls durch Ziehen der Titelzeile verschoben werden. Die ursprüngliche Position kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

• Größe ändern

Mit der Maus kann nur die Länge des Fensters verändert werden. Mit dem Menüeintrag kann ebenfalls die Breite geändert werden. Die ursprüngliche Größe kann über den Menüeintrag **Anzeige / Toolfenster** wieder hergestellt werden.

• Schließen

Kontakt

Dieser Menüeintrag hat die gleiche Wirkung, wie die Schließen-Schaltfläche des Eigenschaftsfenster. Das Fenster kann jederzeit durch den Menüeintrag **Anzeige / Eigenschaften** in der Menüzeile wieder aktivieren.

• Immer im Vordergrund

Ist dieser Menüeintrag mit einem Haken versehen, kann das Eigenschaftsfenster nicht von einem anderen Fenster verdeckt werden. Durch Anklicken des Menüeintrages kann die Anzeige "Immer im Vordergrund" aktiviert und deaktiviert werden.

Datenstruktur

Auf dieser Notizbuchseite wird die Datenstruktur der Datenbank angezeigt. Mit können Elemente aus der Datenstruktur in das Editierfenster gezogen werden. Ohne Zusatztaste werden die entsprechenden Eingabefelder mit einem vorangestellten Label im Editierfenster erzeugt (vgl. Einstellungen). In den Eingabe-Objekten werden die Eingabemöglichkeiten an das zugrunde liegende Datenbankfeld angepasst (Eingabelänge bei alpha-Feldern, zulässiger Wertebereich bei Ganzahlig-lang oder -kurz usw.). Wird das Element der Datenstruktur mit + in das Editierfenster gezogen, wird um die Eingabefelder eine Groupbox erzeugt. Mit + wird eine RecList und mit + + ein RecView mit den entsprechenden Feldern erstellt.

Ein Datenbankfeld kann ebenfalls auf die Eigenschaft DbFieldName gezogen werden.

Drag & Drop-Operationen sind mit folgenden Elementen möglich:

- Dateien
- Teildatensätze
- Felder

Auf die gleiche Weise können Namen der Datenstruktur in den Editor übernommen werden.

Bei untergeordneten Dateien wird der Dateiname der Hauptdatei in eckigen Klammern hinter dem Namen der Datei angegeben.

Wird in diesem Fenster die Taste oder die Tastenkombination + gedrückt, kann in einem Eingabefeld ein Name aus der Datenstruktur angegeben werden. Nach Abschließen der Eingabe mit wird auf das entsprechende Strukturelement (Datei, Teildatensatz, Feld, Schlüssel usw.) positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Datenstruktur gesucht. Ist das Ende bzw. der Anfang der Datenstruktur erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Kontakt

Über ein Kontextmenü lassen sich alle untergeordneten Knoten auf- und zusammenklappen. Zusätzlich kann der komplette Baum zusammengeklappt werden.



Hilfe

Der Designer verfügt über ein kontextsensitives Hilfesystem. Wird ein Objekt im Editierfenster ausgewählt, eine Eigenschaft angeklickt oder im Editor die Hilfe zu einem Befehl aufgerufen, erscheint die Hilfe auf dieser Notizbuchseite.

Bei der Benutzung einer Wheel-Maus kann mit der Tasten-Kombination + in der Hilfe seitenweise zurück und mit + seitenweise vorwärts geblättert werden. Der Schriftgrad lässt sich mit gedrückter -Taste und bzw. einstellen.

Mit der Drag & Drop-Funktionalität können Teile des Hilfetextes markiert und in den Editor gezogen werden. So können die Funktionsköpfe für Ereignisfunktionen oder ganze Beispiele in den Editor übernommen werden.

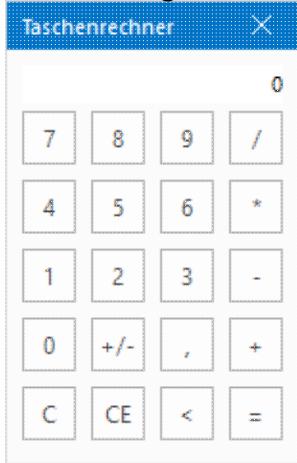
Kontakt

Designer - Taschenrechner

Taschenrechner zum Durchführen von Grundrechenoperationen

Siehe [Datenbankinfo](#),

[Designer](#)



Mit dem Taschenrechner können Grundrechenoperationen durchgeführt werden. Die Bedienung erfolgt mit der Maus oder der Tastatur.

Designer - Objekt öffnen

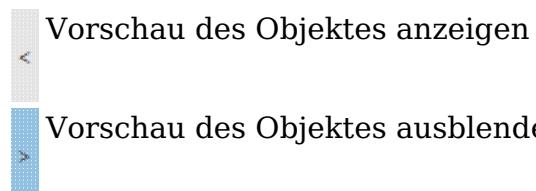
Dialog zum Öffnen von Objekten

Der Dialog hat folgendes Aussehen. Durch das Klicken auf ein Element wird die zugehörige Hilfe angezeigt.

Innerhalb des Designers sind alle Öffnen- und Speichern-Dialoge gleich aufgebaut. In diesem Dialog können Objekte, die in der Datenbank gespeichert sind, zum Öffnen oder Speichern ausgewählt werden.

Vorschau

Auf der linken Seite des Dialogs kann eine Vorschau des Objektes eingeblendet werden.



Bei Dialogen und Druckobjekten wird die Vorschau erstellt, wenn diese schreibend geöffnet oder nach einer Veränderung gespeichert werden. Für Prozeduren wird die Vorschau beim Selektieren erzeugt. Dabei werden maximal die ersten 100 Zeilen gelesen. Die Vorschau von Bildern zeigt immer den aktuellen Stand.

Liste der Objekte

In dieser Liste werden alle Objekte angezeigt die von diesem Typ in der Datenbank vorhanden sind. Ein Objekt wird durch Anklicken selektiert. Wenn das selektierte Objekt verfügbar ist, wird die OK-Schaltfläche aktiv. Mit einem Doppelklick auf ein Objekt, wird dieses geöffnet bzw. gespeichert.

Auf der rechten Seite der Liste befindet sich der Scrollbalken mit folgenden zusätzlichen Funktionalitäten:

- Auf das erste Objekt positionieren
- Eine Seite nach oben blättern
- Auf das vorherige Objekt positionieren
- Auf das nächste Objekt positionieren
- Eine Seite nach unten blättern
- Auf das letzte Objekt positionieren

Name der Objekte

In diesem Feld ist das Objekt einzugeben, das geladen bzw. gespeichert werden soll. Wenn das angegebene Objekt verfügbar ist, wird die OK-Schaltfläche aktiv.

Abhängig von der Verfügbarkeit des Objektes wird der Objektname unterschiedlich dargestellt:

Kontakt

Schwarz	Objekt ist verfügbar
Grau	Objekt ist nicht vorhanden
Rot	Objekt ist gesperrt
Durchgestrichen	Objektnname ist nicht zulässig

Ok und Abbrechen

Mit der Schaltfläche OK wird das angegebene Objekt geöffnet bzw. gespeichert.

Diese Schaltfläche ist grau gesetzt, wenn das Objekt von einem anderen Client gesperrt ist. Beim Öffnen ist sie ebenfalls grau, wenn das Objekt bereits geöffnet oder nicht vorhanden ist.

Objekt löschen

Über die Entf-Taste wird das markierte Objekt gelöscht.

Nur lesend

Soll das Objekt nicht geändert werden, kann die Option "Nur lesend" aktiviert werden. In diesem Fall können keine Änderungen vorgenommen werden. Die Markierungsrahmen im Anwendungsfenster sowie alle Eigenschaften und Ereignisse der Objekte sind rot dargestellt. Lediglich die Größe und die Position eines geöffneten Dialogs können geändert werden, was sich aber nicht auf das gespeicherte Objekt auswirkt.

Unicode

Über die Populiste "Unicode" kann für ein bestehendes Fenster oder Menü die Unicode-Unterstützung aktiviert oder deaktiviert werden. Vor der Umwandlung der Zeichenketten erfolgt eine Abfrage. Bei der Umwandlung eines Objekts mit Unicode in ein Objekt ohne Unicode kann es zu Verlusten von Zeichen kommen, die nicht im ANSI-Zeichensatz vorhanden sind.

Folgende Modi stehen zur Verfügung:

Kein Unicode Das Objekt besitzt keine Unicode-Unterstützung.

Unicode-Ausgabe Das Objekt besitzt Unicode-Unterstützung in Ausgabe-Objekten. Objekte, die in diesem Modus gespeichert wurden, haben in der Spalte UCS das Symbol .

Unicode-Eingabe Das Objekt besitzt Unicode-Unterstützung in Ausgabe-Objekten und dem Edit-Objekt. Objekte, die in diesem Modus gespeichert wurden, haben in der Spalte UCS das Symbol .

 Im Modus "Unicode-Eingabe" können die Eigenschaften Password, PasswordChar und EditMask nicht gesetzt werden. Bei einer Konvertierung in diesen Modus werden die Eigenschaften auf den Standardwert zurückgesetzt.

Wird der ausgewählte Modus nicht verändert, wird das Objekt mit oder ohne Unicode-Unterstützung geladen, abhängig davon wie das Objekt gespeichert ist. Ansonsten wird die Unicode-Unterstützung beim Laden des Objektes entweder

Kontakt

aktiviert oder deaktiviert. Vorhandene Sonderzeichen werden dabei gewandelt. Sind beim Deaktivieren im ANSI-Zeichensatz nicht vorhandene Unicode-Zeichen im Dialog enthalten, werden diese in Fragezeichen gewandelt.

Weitere Informationen zu Unicode befinden sich im Abschnitt [Unicode-Unterstützung](#).

Kontakt

Objekt-Editoren

Editoren zum Bearbeiten von Objekten

Siehe [Oberflächen-Objekte](#)

Folgende Objekte werden mit Objekt-Editoren bearbeitet:

Objekt	Editor
DataList	DataList-Editor
DataListPopup	DataListPopup-Editor
DateEdit	DateEditCalendar-Editor
GanttGraph	GanttGraph-Editor
Menü	Menü-Editor
PrtGanttGraph	PrtGanttGraph-Editor
RecList	RecList-Editor
RecListPopup	RecListPopup-Editor
RecView	RecView-Editor
Statusbar	Statusbar-Editor
Toolbar	Toolbar-Editor
TreeView	TreeView-Editor

Die Objekt-Editoren können mit der Taste  aus dem Designer heraus gestartet werden.

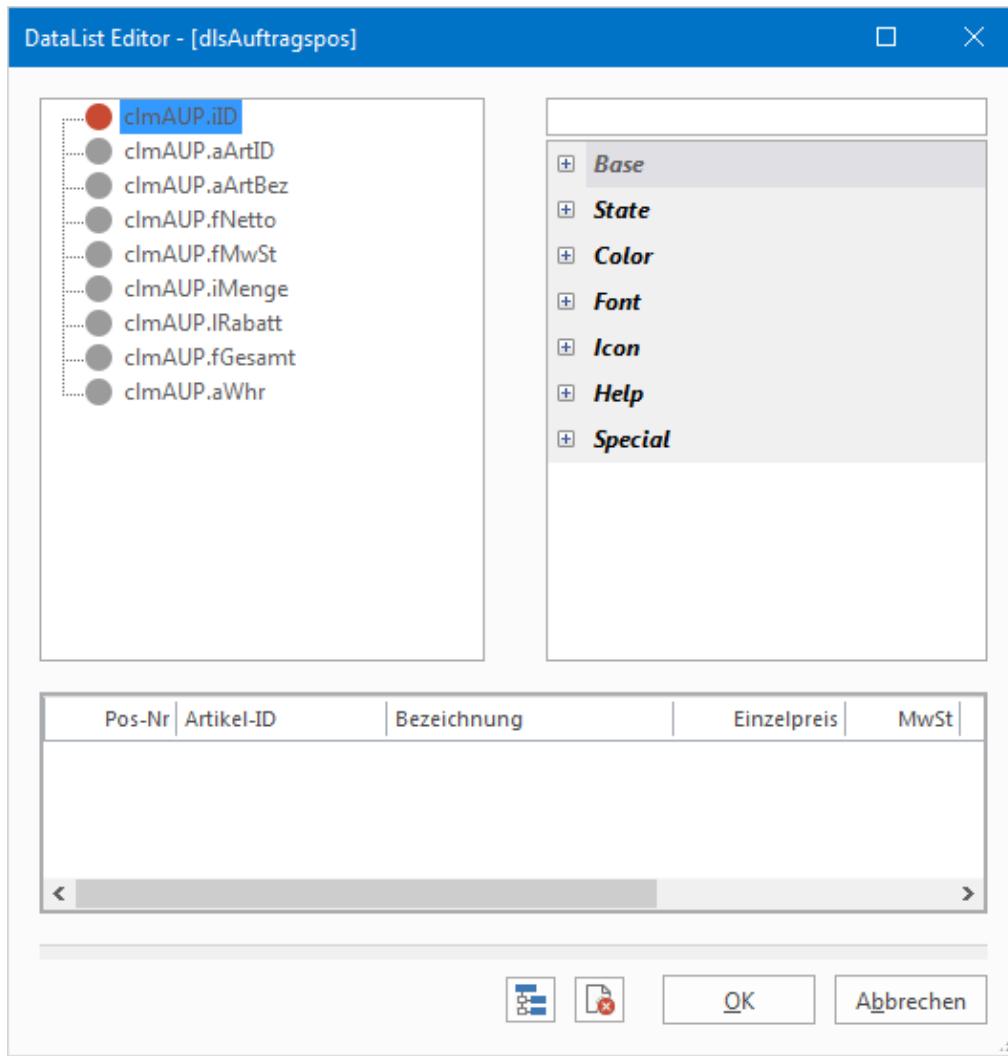
Kontakt

DataList-Editor

Editor zum Bearbeiten der DataList in den Anwendungen

Der DataList-Editor wird im Kontextmenü des DataList-Objekts über die Menüoption **Spalten bearbeiten** oder durch einen Doppelklick in das selektierte Objekt aufgerufen.

Ein ähnlicher Editor wird zum Bearbeiten des Objekts DataListPopup verwendet.



Der DataList-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Liste nicht gespeichert. Mit [OK] wird sie gespeichert.

Liste der Spalten

Einzelne Spalten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden. Die Spalten des DataList-Objekts sind im linken Bereich des Editors aufgeführt. Die Eigenschaften der markierten Spalten können im rechten Bereich eingesehen und editiert werden.

Kontakt

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Spalten können mit Hilfe der Zwischenablage aus der Datenliste kopiert und in die Zugriffsliste eingefügt werden. Ebenso können Spalten aus der Zugriffsliste in die Datenliste kopiert werden.

Mehrere Objekte können nacheinander mit der Tastenkombination  +  ( des Zahlenblocks) in die Zwischenablage kopiert werden.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Spalte wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Spalteneintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



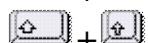
Der kopierte Eintrag wird eingefügt.



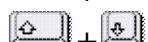
Die Spalte wird in der Reihenfolge nach links verschoben.



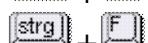
Die Spalte wird in der Reihenfolge nach rechts verschoben.



Die Selektion der Spalten wird nach oben erweitert.



Die Selektion der Spalten wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Spalteneintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

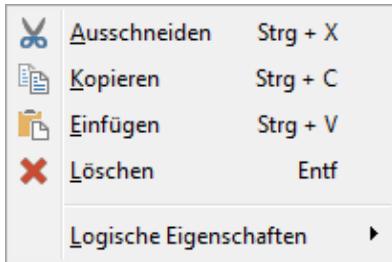
Wird in der Liste der Spalten die Taste  oder die Tastenkombination  +  gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit  wird auf die entsprechende Spalte positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten  und  +  wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird

Kontakt

bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Spalten.

- **Kopieren**

Kopieren der selektierten Spalten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Spalten.

- **Löschen**

Löschen der selektierten Spalten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Spalte zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen der Spalte eines DataList-Objektes zur Verfügung:

<u>Name</u>	Name der Spalte
<u>Custom</u>	Benutzerdefinierte Eigenschaft

Kontakt

<u>Caption</u>	Überschrift der Spalte
<u>Visible</u>	Spalte anzeigen
<u>ClmColFg</u>	Vordergrundfarbe der Spalte
<u>ClmColBkg</u>	Hintergrundfarbe der Spalte
<u>ClmColFocusFg</u>	Vordergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusBkg</u>	Hintergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusOffFg</u>	Vordergrundfarbe der ausgewählten Zeile ohne Fokus
<u>ClmColFocusOffBkg</u>	Hintergrundfarbe der ausgewählten Zeile ohne Fokus
<u>FontParent</u>	Zeichensatz des RecList-Objekts verwenden
<u>FontAttr</u>	Schriftattribute der Spalte
<u>HelpTip</u>	Bubble-Help angeben
<u>ClmType</u>	Datentyp der Spalte
<u>ClmWidth</u>	Breite der Spalte in Pixel
<u>ClmAlign</u>	Ausrichtung der Spalte
<u>ClmSortImage</u>	Bild zur Anzeige der Sortierreihenfolge
<u>ClmOrder</u>	Reihenfolge der Spalten
<u>ClmSortFlags</u>	Optionen der Spalte / Automatische Sortierung
<u>ClmStretch</u>	Automatische Spaltenbreite
<u>ClmStretchWidth</u>	Vorherige Spaltenbreite

Vorschau

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster. Spalten in denen ganze Zahlen, Gleitkommazahlen, Zeit- oder Datumswerte dargestellt werden, können über weitere Eigenschaften formatiert werden. In der Eigenschaft werden ein oder mehrere Beispielwerte angezeigt. In der ComboBox zu der Eigenschaft können nach dem Aufklappen der Liste mehrere Einträge mit Doppelklick aktiviert oder deaktiviert werden. Der Beispieleintrag wird sofort verändert.

Ganze Zahlen

- FmtIntFlags Formatoptionen
- ClmTypeImage Darstellung als Zahl oder Bild

Gleitkommazahlen

- FmtFloatFlags Formatoptionen
- FmtPostComma Anzahl der Nachkommastellen

Zeitwerte

- FmtTimeFlags Formatoptionen

Datumswerte

- FmtDateFormat Formatoptionen
- FmtDateString Darstellungsformat

Logische Werte

- FmtBoolStyle Darstellung des Wertes
- FmtBoolString Zeichenketten zur Darstellung des Wahrheitswertes

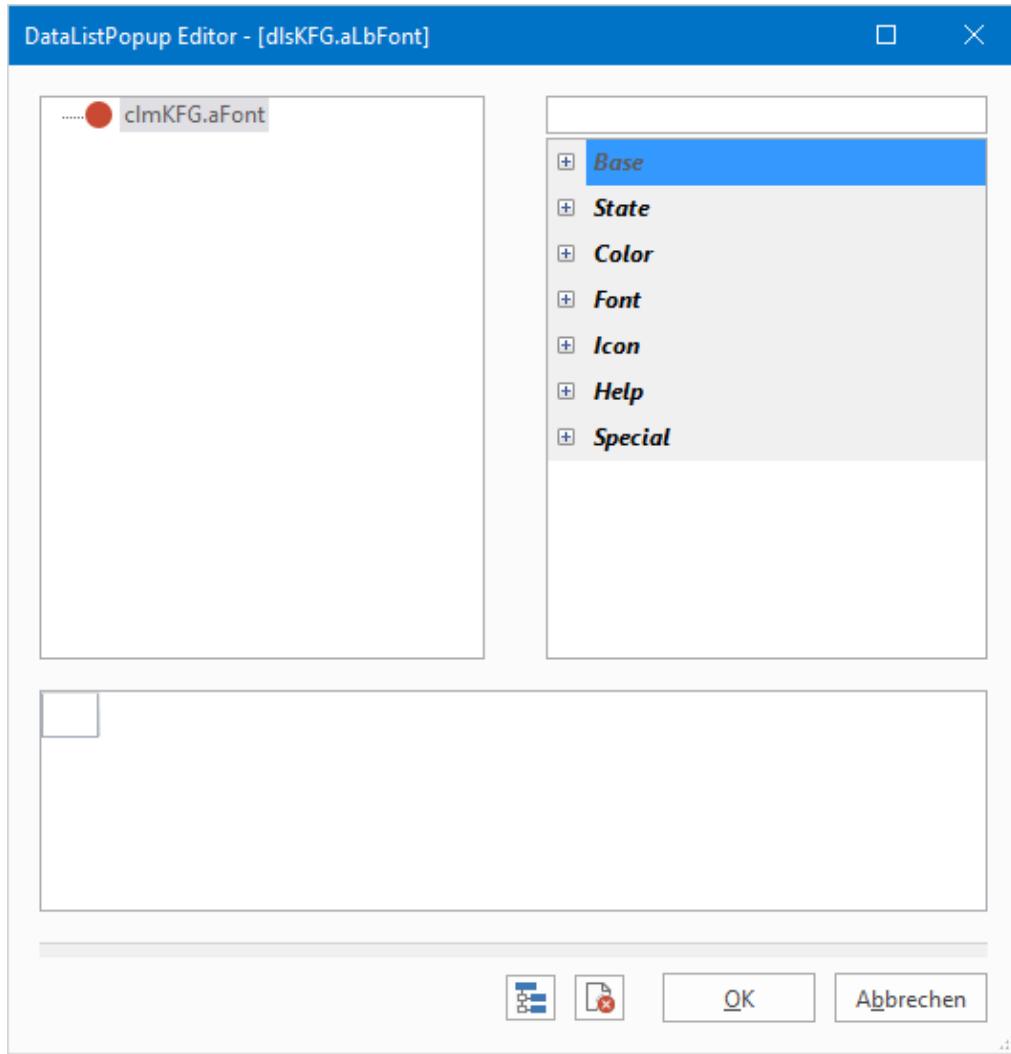
Kontakt

DataListPopup-Editor

Editor zum Bearbeiten der DataListPopUp in den Anwendungen

Der DataListPopup-Editor wird im Kontextmenü des DataListPopUp-Objekts über die Menüoption **Spalten bearbeiten** oder durch einen Doppelklick in das selektierte Objekt aufgerufen.

Ein ähnlicher Editor wird zum Bearbeiten des Objekts DataList verwendet.



Der DataList-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Liste nicht gespeichert. Mit [OK] wird sie gespeichert.

Liste der Spalten

Einzelne Spalten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden. Die Spalten der DataList sind im linken Bereich des Editors aufgeführt. Die Eigenschaften der markierten Spalten können im rechten Bereich eingesehen und editiert werden.

Kontakt

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Spalten können mit Hilfe der Zwischenablage aus der Datenliste kopiert und in die Zugriffsliste eingefügt werden. Ebenso können Spalten aus der Zugriffsliste in die Datenliste kopiert werden.

Mehrere Objekte können nacheinander mit der Tastenkombination  +  ( des Zahlenblocks) in die Zwischenablage kopiert werden.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Spalte wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Spalteneintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



Der kopierte Eintrag wird eingefügt.



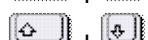
Die Spalte wird in der Reihenfolge nach links verschoben.



Die Spalte wird in der Reihenfolge nach rechts verschoben.



Die Selektion der Spalten wird nach oben erweitert.



Die Selektion der Spalten wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Spalteneintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

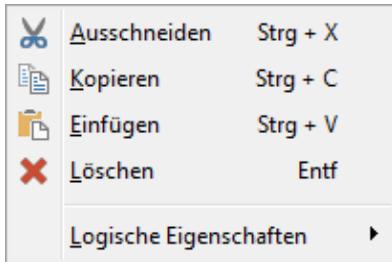
Wird in der Liste der Spalten die Taste  oder die Tastenkombination  +  gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit  wird auf die entsprechende Spalte positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten  und  +  wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird

Kontakt

bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Spalten.

- **Kopieren**

Kopieren der selektierten Spalten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Spalten.

- **Löschen**

Löschen der selektierten Spalten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Spalte zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen der Spalte eines DataList-Objektes zur Verfügung:

<u>Name</u>	Name der Spalte
<u>Custom</u>	Benutzerdefinierte Eigenschaft

<u>Caption</u>	Überschrift der Spalte
<u>Visible</u>	Spalte anzeigen
<u>ClmColFg</u>	Vordergrundfarbe der Spalte
<u>ClmColBkg</u>	Hintergrundfarbe der Spalte
<u>ClmColFocusFg</u>	Vordergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusBkg</u>	Hintergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusOffFg</u>	Vordergrundfarbe der ausgewählten Zeile ohne Fokus
<u>ClmColFocusOffBkg</u>	Hintergrundfarbe der ausgewählten Zeile ohne Fokus
<u>FontParent</u>	Zeichensatz des RecList-Objekts verwenden
<u>FontAttr</u>	Schriftattribute der Spalte
<u>HelpTip</u>	Bubble-Help angeben
<u>ClmType</u>	Datentyp der Spalte
<u>ClmWidth</u>	Breite der Spalte in Pixel
<u>ClmAlign</u>	Ausrichtung der Spalte
<u>ClmOrder</u>	Reihenfolge der Spalten
<u>ClmSortFlags</u>	Optionen der Spalte / Automatische Sortierung
<u>ClmStretch</u>	Automatische Spaltenbreite
<u>ClmStretchWidth</u>	Vorherige Spaltenbreite
<u>ClmPopup</u>	Rückgabewert in Eingabe-Objekt
Vorschau	

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster. Spalten in denen ganze Zahlen, Gleitkommazahlen, Zeit- oder Datumswerte dargestellt werden, können über weitere Eigenschaften formatiert werden. In der Eigenschaft werden ein oder mehrere Beispielwerte angezeigt. In der ComboBox zu der Eigenschaft können nach dem Aufklappen der Liste mehrere Einträge mit Doppelklick aktiviert oder deaktiviert werden. Der Beispieleintrag wird sofort verändert.

Ganze Zahlen

- FmtIntFlags Formatoptionen
- ClmTypeImage Darstellung als Zahl oder Bild

Gleitkommazahlen

- FmtFloatFlags Formatoptionen
- FmtPostComma Anzahl der Nachkommastellen

Zeitwerte

- FmtTimeFlags Formatoptionen

Datumswerte

- FmtDateFormat Formatoptionen
- FmtDateString Darstellungsformat

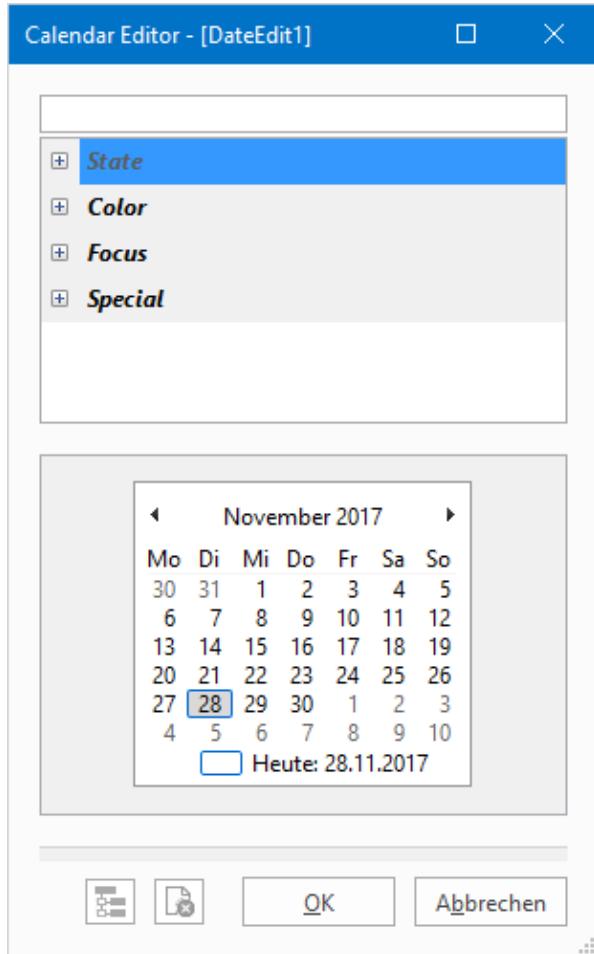
Logische Werte

- FmtBoolStyle Darstellung des Wertes
- FmtBoolString Zeichenketten zur Darstellung des Wahrheitswertes

DateEditCalendar-Editor

Editor zum Bearbeiten des Kalenders eines DateEdit-Objekts

Der DateEditCalendar-Editor kann im Kontextmenü des DateEdit-Objekts über die Menüoption **Kalender bearbeiten** aufgerufen werden.



Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

Kontakt

- ColDateMonth Hintergrundfarbe des Monats
- ColDateFg Vordergrundfarbe
- ColDateTitleBkg Hintergrundfarbe des Monatstitels
- ColDateTitleFg Vordergrundfarbe des Monatstitels
- ColDateTrailing Trailing-Farbe
- StyleDisplay Darstellung des Kalenders



Um die Eigenschaften prozedural zu setzen, werden diese beim DateEdit-Objekt angegeben. Beim Öffnen des Popups durch den Anwender, werden die Eigenschaften automatisch an das Calendar-Objekt übertragen.

Vorschaufenster

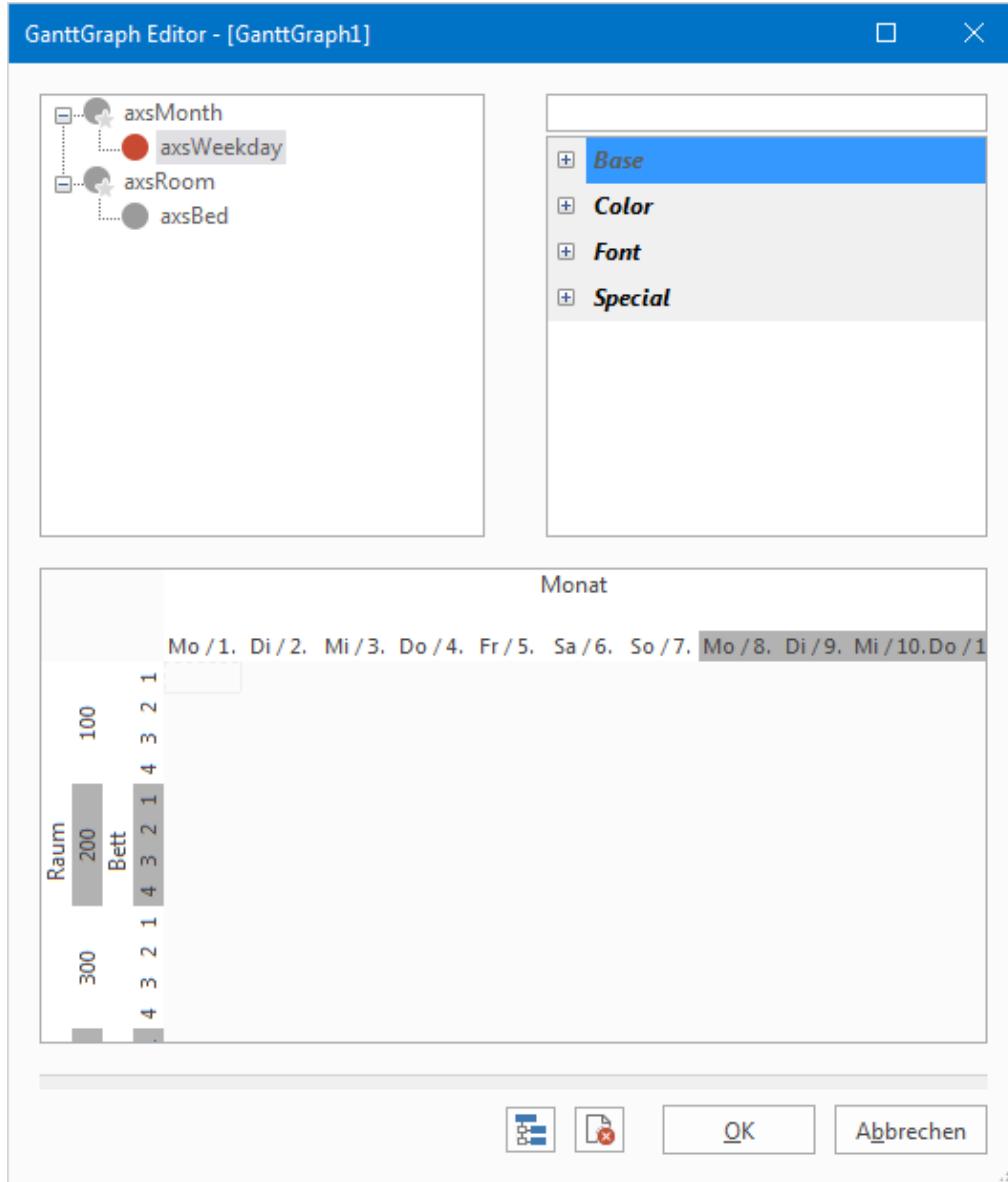
Im Vorschaufenster kann überprüft werden, ob der Kalender das gewünschte Aussehen hat.

Kontakt

GanttGraph-Editor

Editor zum Bearbeiten eines Gantt-Diagramms

Der GanttGraph-Editor kann im Kontextmenü des GanttGraph-Objekts über den Menüeintrag **Achsen bearbeiten...** oder mit einem Doppelklicken auf das Objekt aufgerufen werden.



Das GanttGraph-Objekt besteht aus horizontalen und vertikalen Achsen und der Anzeigefläche für Intervalle. Das Objekt besteht mindestens aus einer horizontalen und einer vertikalen Achse. Weitere Achsen werden diesen untergeordnet. Der GanttGraph-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird der GanttGraph nicht gespeichert. Mit [OK] wird der GanttGraph gespeichert.

Einzelne Achsen können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Kontakt

Nach dem Aufruf werden im oberen Bereich die Achsen und deren Eigenschaften dargestellt. Im unteren Bereich wird das Gantt-Diagramm angezeigt. Alle Änderungen in den Eigenschaften wirken sich sofort auf die Darstellung im unteren Bereich aus.

Eine Achse besteht aus einem Titel und einer Skala. In der Skala werden die Überschriften für jeden Bereich gesetzt und wieviele Spalten ein Bereich umfasst. Zum Beispiel kann eine Monatsübersicht aus zwei Achsen bestehen: Den Wochentagen und den Kalenderwochen. In der untergeordneten Achse können die Wochentage in der Eigenschaft ScalaLabels eingetragen werden. In der übergeordneten Achse werden jeweils sieben Tage (Eintrag in SubDivisions) zu einer Kalenderwoche (Eintrag in ScalaLabels) zusammengefasst.

Liste der Achsen

Die Achsen des GanttGraphen sind im linken Teil des Editors aufgeführt.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Unterachse wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften der Achse oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



Der kopierte Eintrag wird eingefügt.



Die Selektion wird nach oben erweitert.



Die Selektion wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Wird in der Liste der Achsen die Taste oder die Tastenkombination + gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Achse positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Eigenschafts-Filter

Kontakt

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name der Achse
- Custom benutzerdefinierte Eigenschaft
- TitleColorBkg Hintergrundfarbe der Achsenüberschrift
- TitleColorFg Vordergrundfarbe der Achsenüberschrift
- ScalaColorBkg Hintergrundfarbe der Unterteilungen
- ScalaColorEnd Farbe des nicht verwendeten Skala-Bereichs
- ScalaColorText Vordergrundfarbe der Unterteilungen
- ScalaColorEdge Farbe der Begrenzung zwischen den Achsenbeschriftungen und dem Anzegebereich der Intervalle
- FontParentTitle Zeichensatz des GanttGraph-Objekts für die Achsenüberschrift verwenden
- FontTitle Zeichensatz der Achsenüberschrift
- FontParentScala Zeichensatz des GanttGraph-Objekts für die Unterteilungen verwenden
- FontScala Zeichensatz der Unterteilungen
- TitleText Achsenüberschrift
- TitleJustify Ausrichtung der Achsenüberschrift
- ScalaTextJustify Ausrichtung der Unterteilungsüberschrift
- ScalaTextOrigin Anordnung der Skalen
- SubDivisions Anzahl der untergeordneten Zellen
- ScalaLabels Unterteilungsüberschriften
- VerticalText Überschrift vertikal darstellen
- Size Höhe bzw. Breite der Unterteilungen
- SizeScala Höhe bzw. Breite der Achsenüberschrift

In der Eigenschaft ScalaLabels können verschiedene Überschriften angegeben werden. Die unterschiedlichen Überschriften werden durch Kommata getrennt. So können zum Beispiel in einem GanttGraphen Schulungs- und Konferenzräume angezeigt werden, ohne dass es zu Verwechslungen kommen kann ('Schulungsraum 1.1, Schulungsraum 1.2, Konferenzraum 2 ...'). Existieren mehr Zeilen bzw. Spalten als angegebene Labels, werden diese wiederholt. Die Angabe der Liste 'A, B, C' führt zur Ausgabe von A B C A B C A Eine Wiederholung kann mit dem Zeichen '~' unterbunden werden. Das ScalaLabel 'A, B, C~' führt zur Ausgabe von A B C.

Kontakt

Für gängige Bezeichnungen sind Makros definiert, die in der Eigenschaft eingetragen werden können. Die Makros sind bei den Eigenschaften näher erläutert.

Vorschau

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster.

Menü-Editor

Editor zum Bearbeiten der Menüs in den Anwendungen

Der Menüeditor wird über den Menüeintrag **Datei / Menü editieren...** oder über das Kontextmenü des Applikationsfensters aufgerufen. Durch Doppelklicken in den Frame wird ebenfalls der Menüeditor gestartet.

Der Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird das Menü nicht gespeichert. Mit [OK] wird das Menü gespeichert. Einzelne Menüpunkte können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Liste der Menüeinträge

Die Menüeinträge sind links in der Baumstruktur aufgeführt. Menüeinträge eines Menüs stehen untereinander. Die Untermenüs lassen sich bei dem jeweiligen Menüeintrag durch Doppelklicken aufklappen.

Folgende Kommandos stehen im Menüeditor zur Verfügung:

- | | |
|--|--|
| | Ein neuer Menüeintrag wird eingefügt. |
| | Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Menüeintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden. |
| | Der ausgewählte Eintrag wird gelöscht. |
| | Der ausgewählte Eintrag wird kopiert. |
| | Der kopierte Eintrag wird eingefügt. |
| | Der Menüeintrag wird dem darüberliegenden Eintrag untergeordnet. |
| | Der Menüeintrag wird in der Hierarchie nach oben verschoben. |
| | Der Menüeintrag wird noch oben verschoben. |
| | Der Menüeintrag wird nach unten verschoben. |
| | Die Selektion wird noch oben erweitert. |
| | Die Selektion wird nach unten erweitert. |
| | Die Suche wird gestartet. |
| | Suche nach dem nächsten Vorkommen. |
| | Suche nach dem vorherigen Vorkommen. |

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Menüeintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Die Position eines Eintrags in der Hierarchie kann verändert werden, indem der Menüeintrag auf den übergeordneten Eintrag gezogen wird.

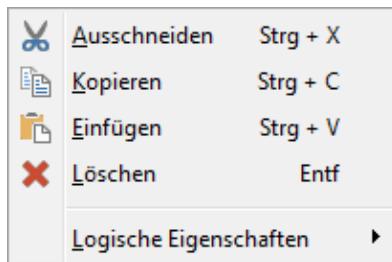
Kontakt

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

Wird in der Liste der Menüeinträge die Tastenkombination + gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf den entsprechenden Menüeintrag positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



• Ausschneiden

Ausschneiden der selektierten Menüpunkte.

• Kopieren

Kopieren der selektierten Menüpunkte.

• Einfügen

Einfügen von vorher ausgeschnittenen bzw. kopierten Menüpunkten.

• Löschen

Löschen der selektierten Menüpunkte.

• Logische Eigenschaften

Liste der logischen Eigenschaften des Menüpunktes zur direkten Bearbeitung.

Menüfunktionen

- Neuen Menüeintrag anlegen
- Markierten Menüeintrag löschen
- Importieren eines Menüs
- Exportieren eines Menüs

Erstellen eines neuen Menüs
Gespeichertes Menü laden
Menü speichern
Menü unter anderen Namen speichern
Ein Menü kann als Resource (*.rsc) oder XML-Datei (*.xml) exportiert werden.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Menüeintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name des Menüeintrags
- Caption Text des Menüeintrags
- MenuID Nummer des Menüeintrags
- MenuSeparator Darstellung als Separator
- MenuDefault Default-Menüeintrag
- ImageTile Verwendetes Bild
- ImageTileUser Verwendetes selbstdefiniertes Bild
- Icon verwendetes Icon
- Disabled Eintrag ausgegraut
- MenuCheck Check-Markierung
- MenuRadioCheck Darstellung der Check-Markierung
- MenuBreak Menü mehrspaltig darstellen
- MenuJustifyRight Menüeintrag rechtsbündig
- MenuAction Menüaktion festlegen
- MenuKey Tastenkombinationen
- StatusItemText Ausgabetext in der Statuszeile
- Custom benutzerdefinierte Eigenschaft

Vorschau

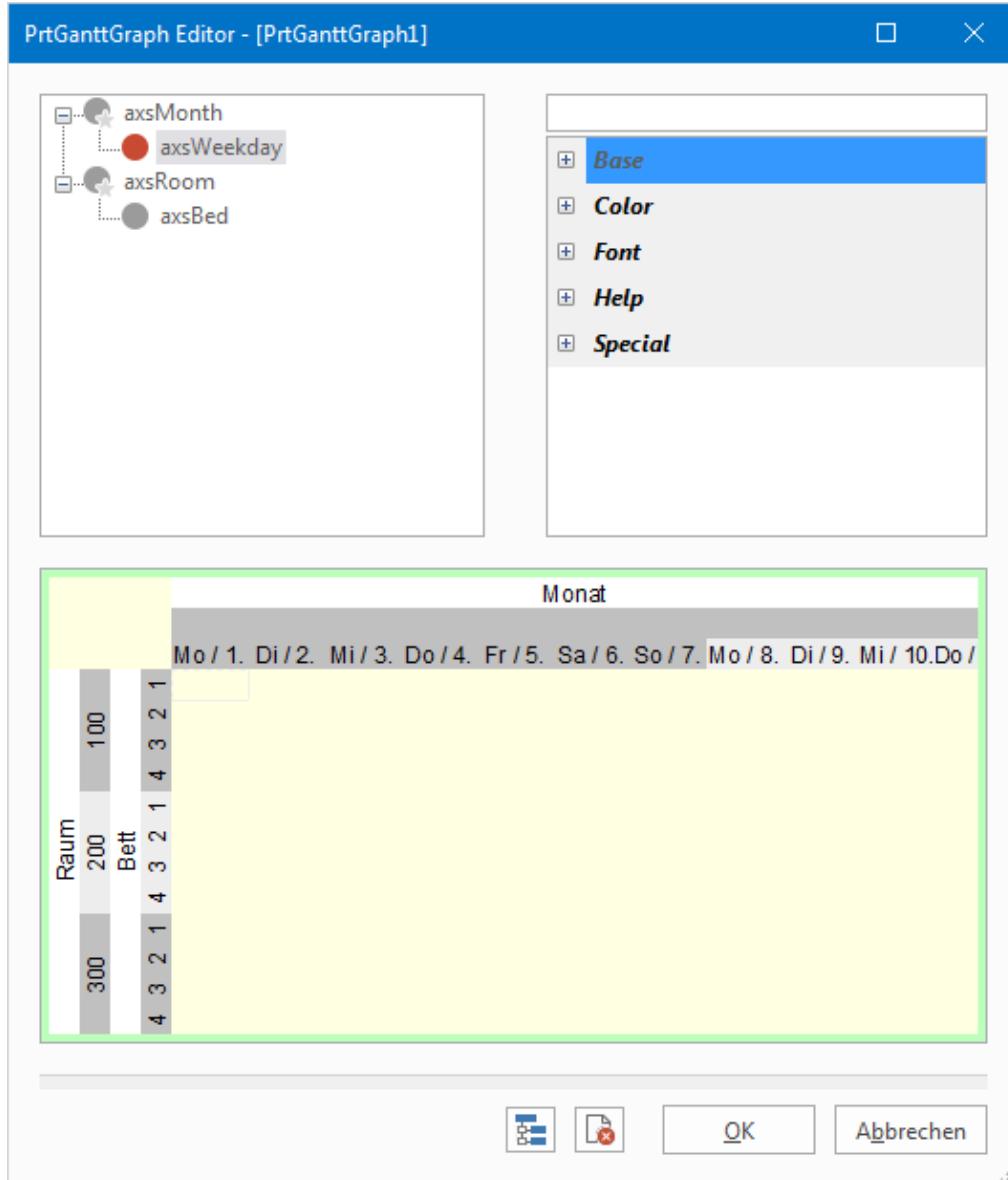
Im Vorschaufenster wird das bearbeitete Menü als Menüzeile angezeigt. Durch Klicken mit der rechten Maustaste im Testfenster, wird das Menü als Kontextmenü angezeigt. Änderungen können sofort im Vorschaufenster überprüft werden.

Kontakt

PrtGanttGraph-Editor

Editor zum Bearbeiten eines druckbaren Gantt-Diagramms

Der PrtGanttGraph-Editor kann im Kontextmenü des PrtGanttGraph-Objektes über den Menüeintrag **Achsen bearbeiten...** oder mit einem Doppelklicken auf das Objekt aufgerufen werden.



Das PrtGanttGraph-Objekt besteht aus horizontalen und vertikalen Achsen und der Anzeigefläche für Intervalle. Das Objekt besteht mindestens aus einer horizontalen und einer vertikalen Achse. Weitere Achsen werden diesen untergeordnet. Der GanttGraph-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird der GanttGraph nicht gespeichert. Mit [OK] wird der GanttGraph gespeichert.

Einzelne Achsen können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Kontakt

Nach dem Aufruf werden im oberen Bereich die Achsen und deren Eigenschaften dargestellt. Im unteren Bereich wird das PrtGantt-Diagramm angezeigt. Alle Änderungen in den Eigenschaften wirken sich sofort auf die Darstellung im unteren Bereich aus.

Eine Achse besteht aus einem Titel und einer Skala. In der Skala werden die Überschriften für jeden Bereich gesetzt und wieviele Spalten ein Bereich umfasst. Zum Beispiel kann eine Monatsübersicht aus zwei Achsen bestehen: Den Wochentagen und den Kalenderwochen. In der untergeordneten Achse können die Wochentage in der Eigenschaft ScalaLabels eingetragen werden. In der übergeordneten Achse werden jeweils sieben Tage (Eintrag in SubDivisions) zu einer Kalenderwoche (Eintrag in ScalaLabels) zusammengefasst.

Liste der Achsen

Die Achsen des GanttGraphen sind im linken Teil des Editors aufgeführt.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Unterachse wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften der Achse oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



Der kopierte Eintrag wird eingefügt.



Die Selektion wird nach oben erweitert.



Die Selektion wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Wird in der Liste der Achsen die Taste oder die Tastenkombination + gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Achse positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination + gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Eigenschafts-Filter

Kontakt

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Über das Kontextmenü kann die Sortierung der Liste und die verwendete Maßeinheit geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name der Achse
- Custom benutzerdefinierte Eigenschaft
- TitleColorBkg Hintergrundfarbe der Achsenüberschrift
- TitleColorFg Vordergrundfarbe der Achsenüberschrift
- ScalaColorBkg Hintergrundfarbe der Unterteilungen
- ScalaColorText Vordergrundfarbe der Unterteilungen
- ScalaColorEdge Farbe der Begrenzung zwischen den Achsenbeschriftungen und dem Anzeigebereich der Intervalle
- FontParentTitle Zeichensatz des GanttGraph-Objekts für die Achsenüberschrift verwenden
- FontTitle Zeichensatz der Achsenüberschrift
- FontParentScala Zeichensatz des GanttGraph-Objekts für die Unterteilungen verwenden
- FontScala Zeichensatz der Unterteilungen
- TitleText Achsenüberschrift
- TitleJustify Ausrichtung der Achsenüberschrift
- ScalaTextJustify Ausrichtung der Unterteilungsüberschrift
- ScalaTextOrigin Anordnung der Skalen
- SubDivisions Anzahl der untergeordneten Zellen
- ScalaLabels Unterteilungsüberschriften
- VerticalText Überschrift vertikal darstellen
- Size Höhe bzw. Breite der Unterteilungen
- SizeScala Höhe bzw. Breite der Achsenüberschrift

In der Eigenschaft ScalaLabels können verschiedene Überschriften angegeben werden. Die unterschiedlichen Überschriften werden durch Kommata getrennt. So können zum Beispiel in einem GanttGraphen Schulungs- und Konferenzräume angezeigt werden, ohne dass es zu Verwechslungen kommen kann ('Schulungsraum 1.1, Schulungsraum 1.2, Konferenzraum 2 ...'). Existieren mehr Zeilen bzw. Spalten als angegebene Labels, werden diese wiederholt. Die Angabe der Liste 'A, B, C' führt zur Ausgabe von A B C A B C A Eine Wiederholung kann mit dem Zeichen '~'

Kontakt

unterbunden werden. Das ScalaLabel 'A, B, C~' führt zur Ausgabe von A B C.

Für gängige Bezeichnungen sind Makros definiert, die in der Eigenschaft eingetragen werden können. Die Makros sind bei den Eigenschaften näher erläutert.

Vorschau

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster.

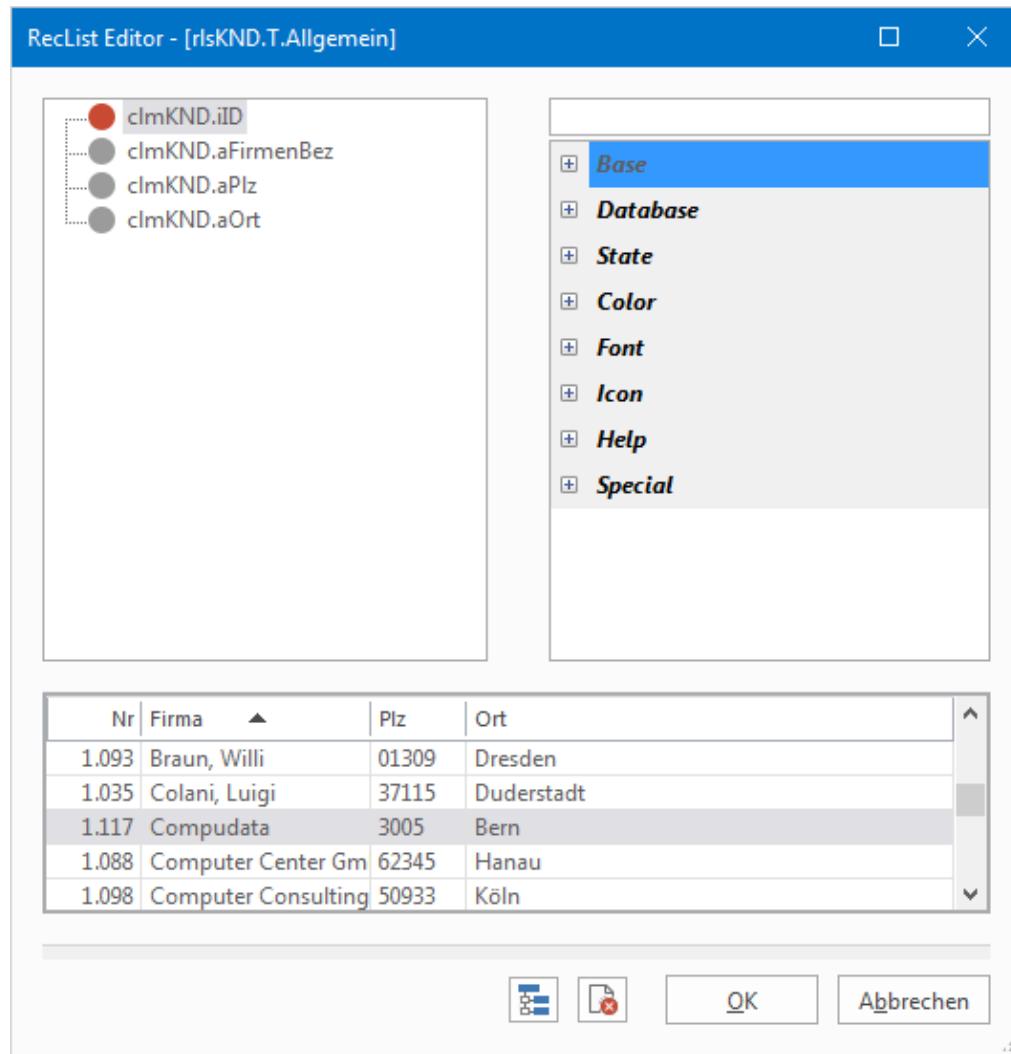
Kontakt

RecList-Editor

Editor zum Bearbeiten der RecList in den Anwendungen

Der RecList-Editor wird im Kontextmenü des RecList-Objekts über die Menüoption **Spalten bearbeiten** oder durch einen Doppelklick in das selektierte Objekt aufgerufen.

Ein ähnlicher Editor wird zum Bearbeiten des Objekts RecListPopup verwendet.



Der RecList-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Liste nicht gespeichert. Mit [OK] wird sie gespeichert.

Liste der Spalten

Einzelne Spalten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden. Die Spalten der RecList sind im linken Bereich des Editors aufgeführt. Die Eigenschaften der markierten Spalten können im rechten Bereich eingesehen und editiert werden.

Kontakt

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Spalten können mit Hilfe der Zwischenablage aus der Datenliste kopiert und in die Zugriffsliste eingefügt werden. Ebenso können Spalten aus der Zugriffsliste in die Datenliste kopiert werden.

Mehrere Objekte können nacheinander mit der Tastenkombination (des Zahlenblocks) in die Zwischenablage kopiert werden.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Spalte wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Spalteneintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



Der kopierte Eintrag wird eingefügt.



Die Spalte wird in der Reihenfolge nach links verschoben.



Die Spalte wird in der Reihenfolge nach rechts verschoben.



Die Selektion der Spalten wird nach oben erweitert.



Die Selektion der Spalten wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Spalteneintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

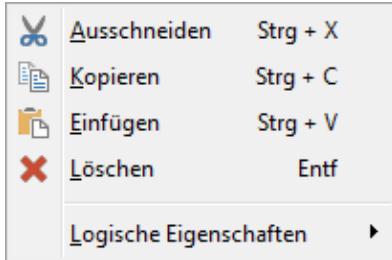
Wird in der Liste der Spalten die Taste oder die Tastenkombination gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Spalte positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und + wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination gestartet. Die Groß-Kleinschreibung wird

Kontakt

bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Spalten.

- **Kopieren**

Kopieren der selektierten Spalten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Spalten.

- **Löschen**

Löschen der selektierten Spalten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Spalte zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen der Spalte eines RecList-Objekts zur Verfügung:

<u>Name</u>	Name der Spalte
<u>Custom</u>	Benutzerdefinierte Eigenschaft

Kontakt

<u>Caption</u>	Überschrift der Spalte
<u>DbFieldName</u>	Feldname
<u>DbKeyNo</u>	Nummer des Schlüssels
<u>DbKeyReverseSort</u>	Schlüssel in umgekehrter Reihenfolge
<u>Visible</u>	Spalte anzeigen
<u>ClmColFg</u>	Vordergrundfarbe der Spalte
<u>ClmColBkg</u>	Hintergrundfarbe der Spalte
<u>ClmColFocusFg</u>	Vordergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusBkg</u>	Hintergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusOffFg</u>	Vordergrundfarbe der ausgewählten Zeile ohne Fokus
<u>ClmColFocusOffBkg</u>	Hintergrundfarbe der ausgewählten Zeile ohne Fokus
<u>FontParent</u>	Zeichensatz des RecList-Objekts verwenden
<u>FontAttr</u>	Schriftattribute der Spalte
<u>HelpTip</u>	Bubble-Help angeben
<u>ClmWidth</u>	Breite der Spalte in Pixel
<u>ClmAlign</u>	Ausrichtung der Spalte
<u>ClmSortImage</u>	Bild zur Anzeige der Sortierreihenfolge
<u>ClmOrder</u>	Reihenfolge der Spalten
<u>ClmSortFlags</u>	Optionen der Spalte / Automatische Sortierung
<u>ClmStretch</u>	Automatische Spaltenbreite
<u>ClmStretchWidth</u>	Vorherige Spaltenbreite

Vorschau

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster.

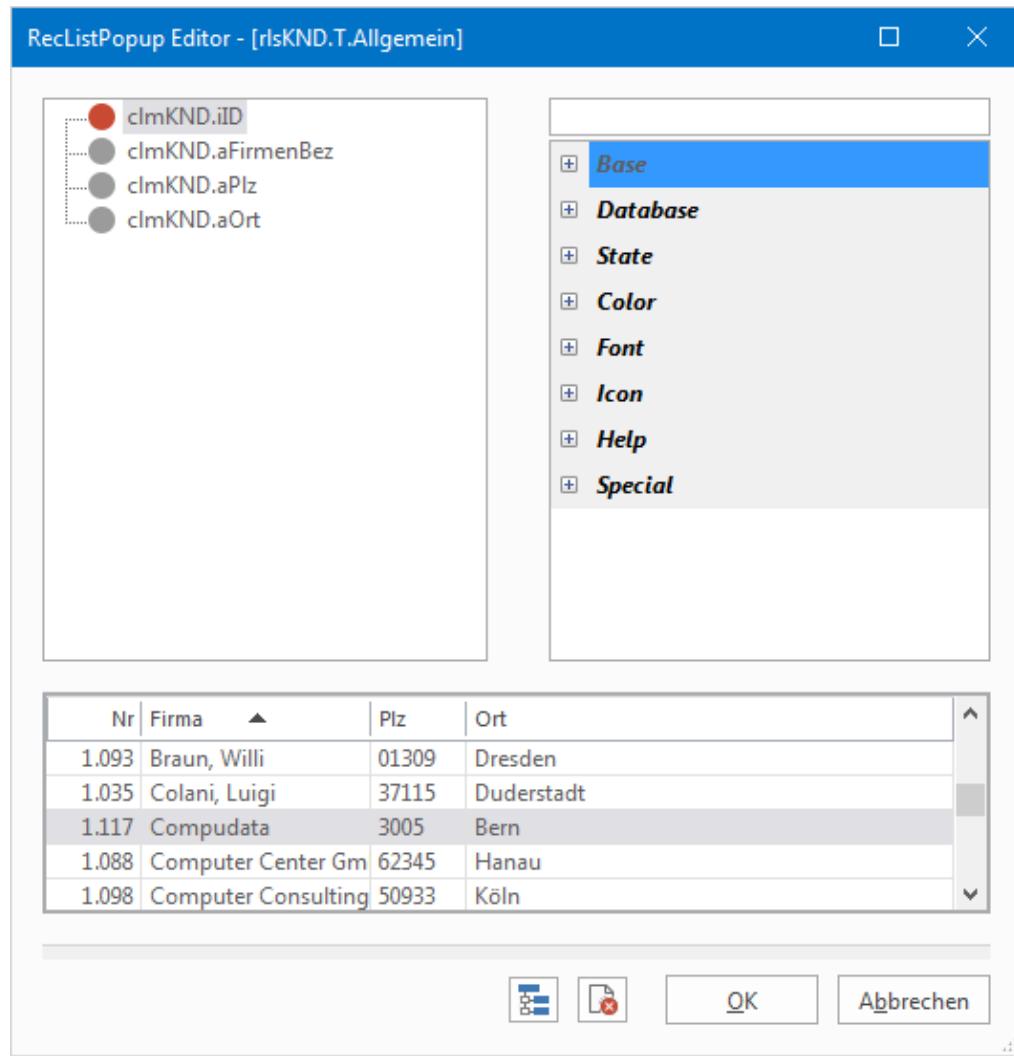
Kontakt

RecListPopup-Editor

Editor zum Bearbeiten der RecListPopup in den Anwendungen

Der RecList-Editor wird im Kontextmenü des RecListPopup-Objekts über die Menüoption **Spalten bearbeiten** oder durch einen Doppelklick in das selektierte Objekt aufgerufen.

Ein ähnlicher Editor wird zum Bearbeiten des Objekts RecList verwendet.



Der RecListPopup-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Liste nicht gespeichert. Mit [OK] wird sie gespeichert.

Liste der Spalten

Einzelne Spalten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden. Die Spalten der RecList sind im linken Bereich des Editors aufgeführt. Die Eigenschaften der markierten Spalten können im rechten Bereich eingesehen und editiert werden.

Kontakt

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Spalten können mit Hilfe der Zwischenablage aus der Datenliste kopiert und in die Zugriffsliste eingefügt werden. Ebenso können Spalten aus der Zugriffsliste in die Datenliste kopiert werden.

Mehrere Objekte können nacheinander mit der Tastenkombination  +  ( des Zahlenblocks) in die Zwischenablage kopiert werden.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Spalte wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Spalteneintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



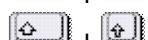
Der kopierte Eintrag wird eingefügt.



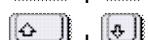
Die Spalte wird in der Reihenfolge nach links verschoben.



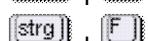
Die Spalte wird in der Reihenfolge nach rechts verschoben.



Die Selektion der Spalten wird nach oben erweitert.



Die Selektion der Spalten wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Spalteneintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

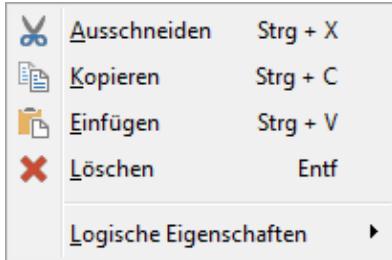
Wird in der Liste der Spalten die Taste  oder die Tastenkombination  +  gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit  wird auf die entsprechende Spalte positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten  und  +  wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird

Kontakt

bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Spalten.

- **Kopieren**

Kopieren der selektierten Spalten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Spalten.

- **Löschen**

Löschen der selektierten Spalten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Spalte zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen der Spalte eines RecList-Objektes zur Verfügung:

<u>Name</u>	Name der Spalte
<u>Custom</u>	Benutzerdefinierte Eigenschaft

Kontakt

<u>Caption</u>	Überschrift der Spalte
<u>DbFieldName</u>	Feldname
<u>DbKeyNo</u>	Nummer des Schlüssels
<u>DbKeyReverseSort</u>	Schlüssel in umgekehrter Reihenfolge
<u>Visible</u>	Spalte anzeigen
<u>ClmColFg</u>	Vordergrundfarbe der Spalte
<u>ClmColBkg</u>	Hintergrundfarbe der Spalte
<u>ClmColFocusFg</u>	Vordergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusBkg</u>	Hintergrundfarbe der ausgewählten Zeile mit Fokus
<u>ClmColFocusOffFg</u>	Vordergrundfarbe der ausgewählten Zeile ohne Fokus
<u>ClmColFocusOffBkg</u>	Hintergrundfarbe der ausgewählten Zeile ohne Fokus
<u>FontParent</u>	Zeichensatz des RecList-Objekts verwenden
<u>FontAttr</u>	Schriftattribute der Spalte
<u>HelpTip</u>	Bubble-Help angeben
<u>ClmWidth</u>	Breite der Spalte in Pixel
<u>ClmAlign</u>	Ausrichtung der Spalte
<u>ClmOrder</u>	Reihenfolge der Spalten
<u>ClmStretch</u>	Automatische Spaltenbreite
<u>ClmStretchWidth</u>	Vorherige Spaltenbreite
<u>ClmPopup</u>	Rückgabewert in Eingabe-Objekt
Vorschau	

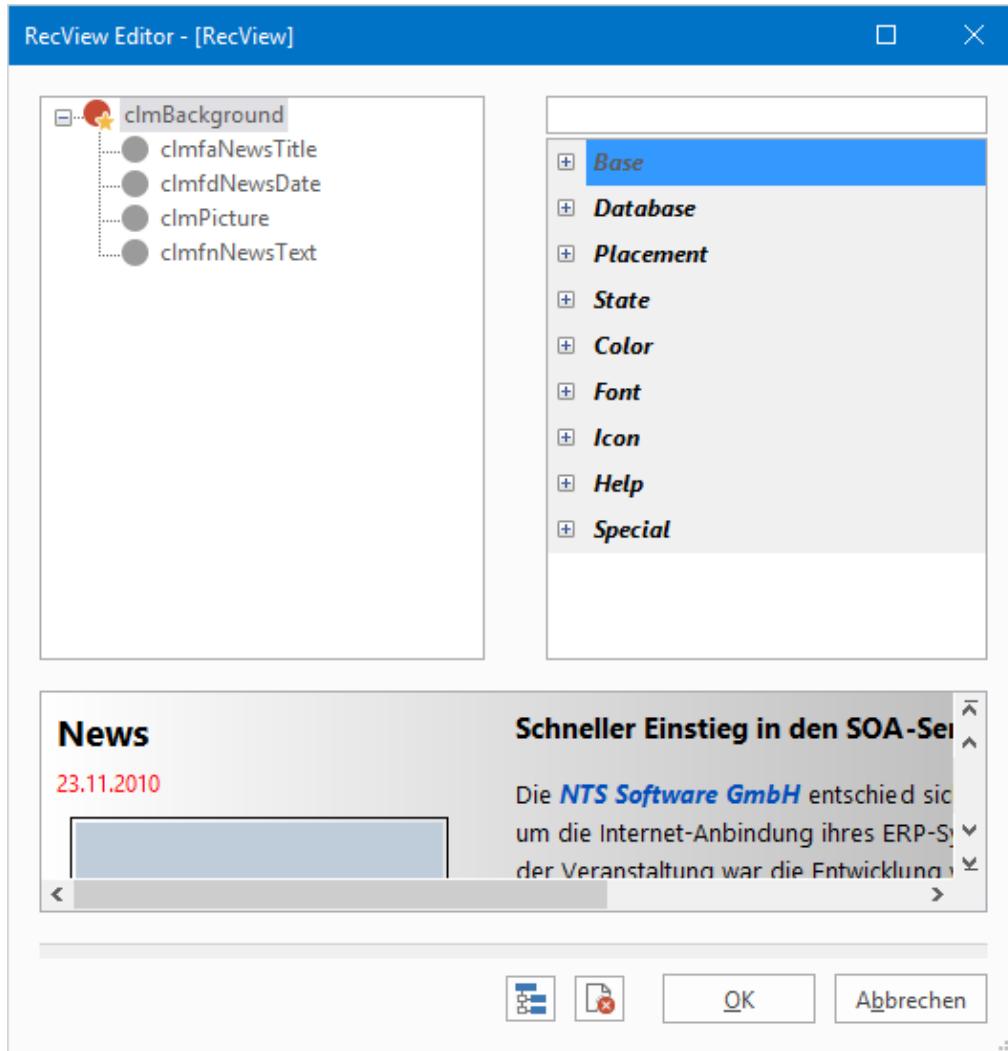
Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster.

Kontakt

RecView-Editor

Editor zum Bearbeiten des RecView in den Anwendungen

Der RecView-Editor wird im Kontextmenü des RecView-Objekts über die Menüoption **Spalten bearbeiten** oder durch einen Doppelklick in das selektierte Objekt aufgerufen.



Der RecView-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Liste nicht gespeichert. Mit [OK] wird sie gespeichert.

Liste der Spalten

Einzelne Spalten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden. Die Spalten des RecViews sind im linken Bereich des Editors aufgeführt. Die Eigenschaften der markierten Spalten können im rechten Bereich eingesehen und editiert werden.

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Spalten können mit Hilfe der Zwischenablage aus der Datenliste kopiert und in die Zugriffsliste eingefügt

Kontakt

werden. Ebenso können Spalten aus der Zugriffsliste in die Datenliste kopiert werden.

Mehrere Objekte können nacheinander mit der Tastenkombination (des Zahlenblocks) in die Zwischenablage kopiert werden.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Spalte wird eingefügt.



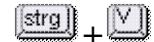
Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Spalteneintrags oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



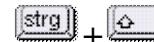
Der ausgewählte Eintrag wird gelöscht.



Der ausgewählte Eintrag wird kopiert.



Der kopierte Eintrag wird eingefügt.



Die Spalte wird dem darüberliegenden Eintrag untergeordnet.



Die Spalte wird in der Hierarchie nach oben verschoben.



Die Spalte wird in der Reihenfolge nach links verschoben.



Die Spalte wird in der Reihenfolge nach rechts verschoben.



Die Selektion der Spalten wird nach oben erweitert.



Die Selektion der Spalten wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

Wird ein Spalteneintrag mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

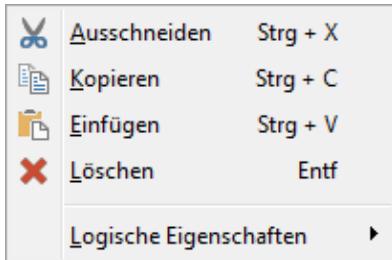
Wird in der Liste der Spalten die Taste oder die Tastenkombination gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Spalte positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination gestartet. Die Groß-Kleinschreibung wird

Kontakt

bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Spalten.

- **Kopieren**

Kopieren der selektierten Spalten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Spalten.

- **Löschen**

Löschen der selektierten Spalten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Spalte zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen der Spalte eines RecView-Objekts zur Verfügung:

AreaLeft X-Position der Spalte
AreaMarginBottom Unterer Randabstand

Kontakt

<u>AreaMarginLeft</u>	Linker Randabstand
<u>AreaMarginRight</u>	Rechter Randabstand
<u>AreaMarginTop</u>	Oberer Randabstand
<u>AreaTop</u>	Y-Position der Spalte
<u>BorderColFg</u>	Rahmenfarbe
<u>BorderType</u>	Rahmentyp
<u>BorderWidth</u>	Rahmenbreite in Pixeln
<u>Caption</u>	Überschrift der Spalte
<u>ClmAlign</u>	Ausrichtung der Spalte
<u>ClmColBkg</u>	Hintergrundfarbe der Spalte
<u>ClmColFg</u>	Vordergrundfarbe der Spalte
<u>ClmStretch</u>	Automatisches Anpassen der Breite des Objektes
<u>ClmWidth</u>	Breite der Spalte in Pixel
<u>ContentHeightMax</u>	Maximale Höhe einer Zeile in der Spalte
<u>ContentSource</u>	Quelle des Spalteninhalts
<u>ContentType</u>	Typ des Spalteninhalts
<u>CryptKey</u>	Passwort für Verschlüsselung von binären Objekten
<u>DbFieldName</u>	Feldname
<u>DrawBkgAlways</u>	Hintergrundfarbe der Spalte auch bei selektierter Gruppe anzeigen
<u>FileName</u>	Name der anzuseigenden Datei
<u>FmtBigIntFlags</u>	Formatierungsattribute für große Integer-Werte
<u>FmtBoolString</u>	Formatierungsattribute für logische Werte
<u>FmtBoolStyle</u>	Formatierungsattribute für logische Werte
<u>FmtDateString</u>	Formatierungsattribute für Datums-Werte
<u>FmtDateStyle</u>	Formatierungsattribute für Datums-Werte
<u>FmtDecimalFlags</u>	Formatierungsattribute für Dezimal-Werte
<u>FmtFloatFlags</u>	Formatierungsattribute für Gleitkomma-Werte
<u>FmtIntFlags</u>	Formatierungsattribute für Integer-Werte
<u>FmtPostComma</u>	Anzahl der Nachkommastellen
<u>FmtTimeFlags</u>	Formatierungsattribute für Zeit-Werte
<u>Font</u>	Schriftart der Spalte
<u>HelpTip</u>	Text des HelpTip
<u>CurrentInt</u>	Nummer des zu verwendenden Bildes
<u>Name</u>	Name der Spalte
<u>ReadOnly</u>	Spalte nur lesbar
<u>Visible</u>	Spalte anzeigen
<u>VisibleOrder</u>	Reihenfolge der sichtbaren Spalten
<u>WordBreak</u>	Text der Spalte automatisch umbrechen, falls er nicht in die Zeile passt

Vorschau

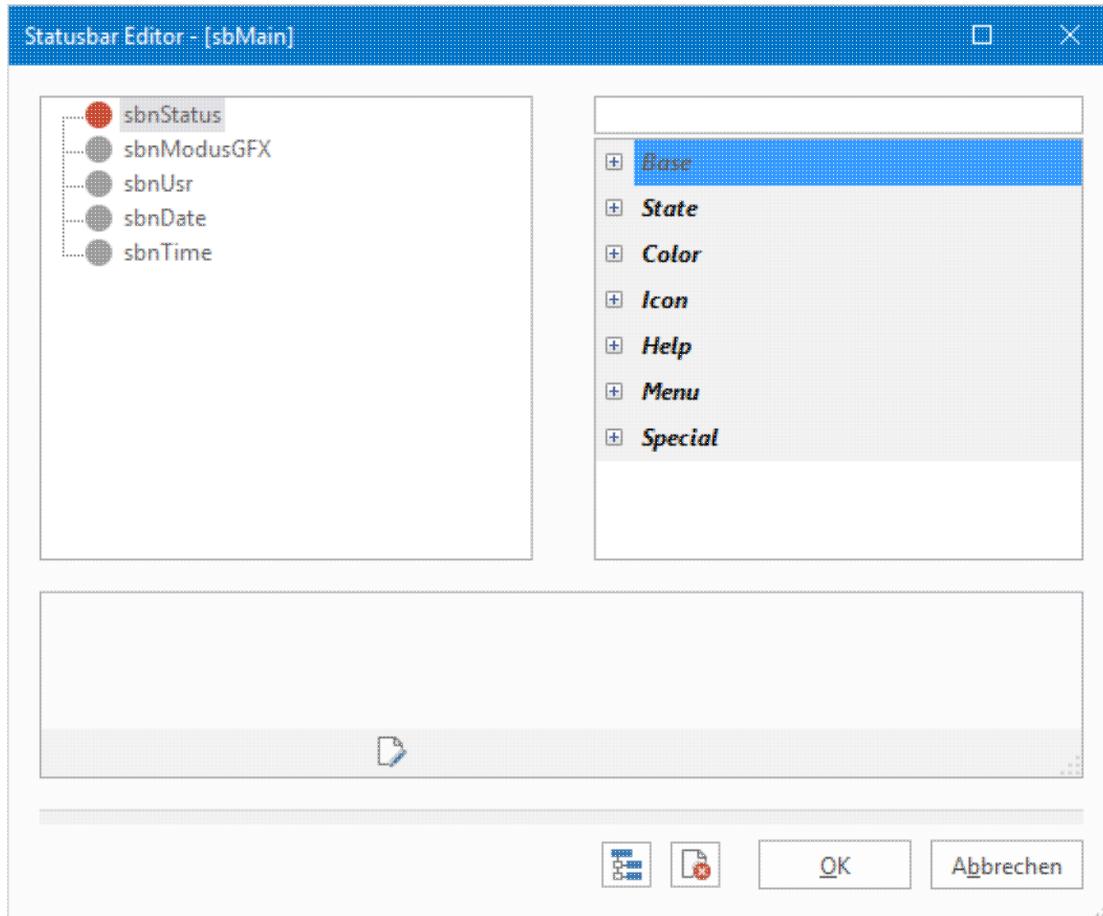
Kontakt

Änderungen in den Eigenschaften führen sofort zu Veränderungen im Vorschaufenster.

Statusbar-Editor

Editor zum Bearbeiten der Statusbar

Der Statuszeilen-Editor kann im Kontextmenü des Statuszeilen-Objektes über die Menüoption **Statusbar bearbeiten...** oder mit einem Doppelklicken auf das Objekt aufgerufen werden.



Der Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Statusbar nicht gespeichert. Mit [OK] wird die Statuszeile gespeichert. Einzelne Schaltflächen können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Liste der Schaltflächen

Die Schaltflächen der Statusbar sind im linken Teil des Editors aufgeführt.

Folgende Tastaturkommandos stehen zur Verfügung:



Ein neuer Button wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften der Schaltfläche oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.

Kontakt

- Die Schaltfläche wird noch oben verschoben.
- Die Schaltfläche wird nach unten verschoben.
- Die Selektion wird nach oben erweitert.
- Die Selektion wird nach unten erweitert.
- Die Suche wird gestartet.
- Suche nach dem nächsten Vorkommen.
- Suche nach dem vorherigen Vorkommen.

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Schaltflächen können mit Hilfe der Zwischenablage in andere Statusbar-Objekte eingefügt werden. Mehrere Objekte können nacheinander mit der Tastenkombination (des Zahlenblocks) in die Zwischenablage kopiert werden.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

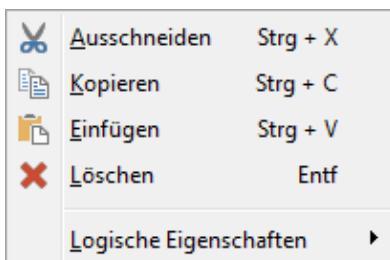
Wird eine Schaltfläche mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

Wird in der Liste der Schaltflächen die Taste oder die Tastenkombination gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Schaltfläche positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Menüpunkte.

- **Kopieren**

Kopieren der selektierten Menüpunkte.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Menüpunkten.

- **Löschen**

Löschen der selektierten Menüpunkte.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften des Menüpunktes zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Buttons sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name des Objekts
- Custom Benutzerdefinierte Eigenschaft
- Caption Ausgabetext
- ImageTileUser Verwendetes benutzerdefiniertes Bild
- Visible Sichtbarkeit der Schaltfläche
- ColFg Vordergrundfarbe
- ColBkg Hintergrundfarbe
- HelpTip Bubble-Help angeben
- MenuSeparator Objekt als Separator
- Group Gruppe
- JustifyView Ausrichtung des Textes in der Schaltfläche
- ImageTile Verwendetes Bild
- ImageOption Darstellungsoption des Bildes
- ClmStretch Automatisches Anpassen der Breite

Vorschau

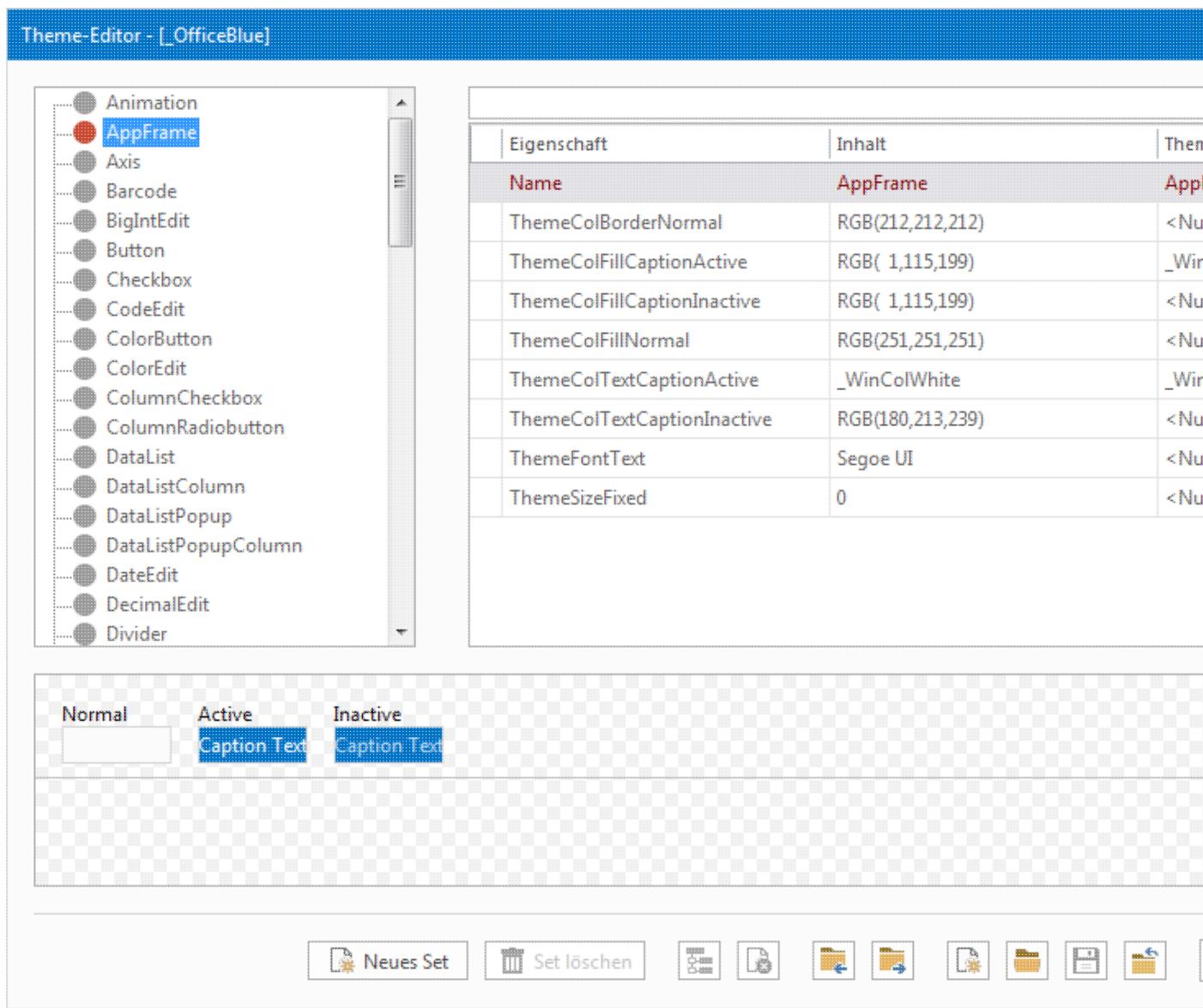
Im Vorschaufenster wird die Statusbar dargestellt, die momentan editiert wird. In diesem Fenster kann überprüft werden, ob die Statusbar das gewünschte Aussehen hat.

Kontakt

Theme-Editor

Editor zum Bearbeiten der Themes in den Anwendungen

Der Theme-Editor wird über den Menüeinträge **Datei / Neu / Theme..., Datei / Öffnen** nach Auswahl des Typs Themes oder über Doppelklick auf die Eigenschaft ThemeName und anschließenden Klick auf **Erweitert...** geöffnet.



Der Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird das Theme nicht gespeichert. Mit [OK] wird das Theme gespeichert.

Liste der Objekttypen

Die Objekttypen, für die Theme-Eigenschaften definiert werden können, sind links in der Baumstruktur aufgeführt.

Folgende Kommandos stehen im Menüeditor zur Verfügung:

Kontakt

-  +  Die Selektion wird nach oben erweitert.
-  +  Die Selektion wird nach unten erweitert.
-  +  Die Suche wird gestartet.
-  Suche nach dem nächsten Vorkommen.
-  +  Suche nach dem vorherigen Vorkommen.

Wird in der Liste der Objekttypen die Tastenkombination  +  gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit  wird auf den entsprechenden Objekttyp positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten  und  +  wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination  +  gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Werkzeugleiste

-  Neues ThemeSet anlegen
 - Ausgewähltes ThemeSet löschen
 - Importieren eines Themes
 - Exportieren eines Themes
 - Erstellen eines neuen Themes
 - Gespeichertes Theme laden
 - Theme speichern
 - Theme unter anderen Namen speichern
- Ein Theme kann als Resource (*.rsc) oder XML-Datei (*.xml) exportiert werden.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Objekttyps sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name
- ThemeBaseName
- ThemeColArrowChecked
- ThemeColArrowDisabled
- ThemeColArrowDropDisabled
- ThemeColArrowDropError
- ThemeColArrowDropHighlight
- ThemeColArrowDropNormal
- ThemeColArrowDropPressed
- ThemeColArrowHighlight
- ThemeColArrowIvl
- ThemeColArrowNormal
- ThemeColArrowPressed
- ThemeColArrowScrollHighlight
- ThemeColArrowScrollNormal
- ThemeColArrowScrollPressed
- ThemeColBorderActive
- ThemeColBorderBase
- ThemeColBorderBoundary
- ThemeColBorderCellActiveHorz
- ThemeColBorderCellActiveVert
- ThemeColBorderCheckDisabled
- ThemeColBorderChecked
- ThemeColBorderCheckHighlight
- ThemeColBorderCheckNormal
- ThemeColBorderDisabled
- ThemeColBorderError
- ThemeColBorderFocussed
- ThemeColBorderHdrBase
- ThemeColBorderHighlight
- ThemeColBorderHover
- ThemeColBorderInactive
- ThemeColBorderIvl
- ThemeColBorderNormal
- ThemeColBorderPageNormal
- ThemeColBorderPressed
- ThemeColBorderReadOnly
- ThemeColBorderSelectedActive
- ThemeColBorderSelectedInactive
- ThemeColBorderVisited
- ThemeColCheckActive
- ThemeColCheckDisabled
- ThemeColCheckHighlight
- ThemeColCheckInactive
- ThemeColCheckNormal
- ThemeColCheckPressed
- ThemeColCheckSelectedActive
- ThemeColCheckSelectedInactive
- ThemeColEditorAlpha

- [ThemeColEditorCKeywords](#)
- [ThemeColEditorComment](#)
- [ThemeColEditorConstants](#)
- [ThemeColEditorCurrentLine](#)
- [ThemeColEditorCustomHighlight](#)
- [ThemeColEditorDataTypes](#)
- [ThemeColEditorFunctions](#)
- [ThemeColEditorKeywords](#)
- [ThemeColEditorNumber](#)
- [ThemeColEditorOccurrences](#)
- [ThemeColEditorPreProcessor](#)
- [ThemeColEditorSelection](#)
- [ThemeColEditorText](#)
- [ThemeColFillActive](#)
- [ThemeColFillBarDisabled](#)
- [ThemeColFillBarNormal](#)
- [ThemeColFillBarShadeDisabled](#)
- [ThemeColFillBarShadeNormal](#)
- [ThemeColFillBase](#)
- [ThemeColFillCaption](#)
- [ThemeColFillCaptionActive](#)
- [ThemeColFillCaptionInactive](#)
- [ThemeColFillChannelDisabled](#)
- [ThemeColFillChannelHighlight](#)
- [ThemeColFillChannelNormal](#)
- [ThemeColFillChannelPressed](#)
- [ThemeColFillCheckDisabled](#)
- [ThemeColFillChecked](#)
- [ThemeColFillCheckHighlight](#)
- [ThemeColFillCheckNormal](#)
- [ThemeColFillDisabled](#)
- [ThemeColFillDropDisabled](#)
- [ThemeColFillDropError](#)
- [ThemeColFillDropHighlight](#)
- [ThemeColFillDropNormal](#)
- [ThemeColFillDropPressed](#)
- [ThemeColFillEnd](#)
- [ThemeColFillError](#)
- [ThemeColFillFocussed](#)
- [ThemeColFillHdrBase](#)
- [ThemeColFillHdrDisabled](#)
- [ThemeColFillHdrHighlight](#)
- [ThemeColFillHdrNormal](#)
- [ThemeColFillHdrPressed](#)
- [ThemeColFillHighlight](#)
- [ThemeColFillHover](#)
- [ThemeColFillInactive](#)
- [ThemeColFillIndentNormal](#)
- [ThemeColFillNormal](#)
- [ThemeColFillPressed](#)
- [ThemeColFillReadOnly](#)

- ThemeColFillRulerNormal
- ThemeColFillScrollHighlight
- ThemeColFillScrollNormal
- ThemeColFillScrollPressed
- ThemeColFillSelectedActive
- ThemeColFillSelectedInactive
- ThemeColFillShadeChecked
- ThemeColFillShadeDisabled
- ThemeColFillShadeHighlight
- ThemeColFillShadeNormal
- ThemeColFillShadePressed
- ThemeColFillSpinArrowDisabled
- ThemeColFillSpinArrowError
- ThemeColFillSpinArrowHighlight
- ThemeColFillSpinArrowNormal
- ThemeColFillSpinArrowPressed
- ThemeColFillSpinDisabled
- ThemeColFillSpinError
- ThemeColFillSpinHighlight
- ThemeColFillSpinNormal
- ThemeColFillSpinPressed
- ThemeColFillSplitterDisabled
- ThemeColFillSplitterNormal
- ThemeColFillTabNormal
- ThemeColFillThumbDisabled
- ThemeColFillThumbHighlight
- ThemeColFillThumbNormal
- ThemeColFillThumbPressed
- ThemeColFillVisited
- ThemeColForeSplitterDisabled
- ThemeColForeSplitterNormal
- ThemeColGrip
- ThemeColImageNormal
- ThemeColSeparator
- ThemeColSeparatorClmBase
- ThemeColSeparatorClmFixedBase
- ThemeColSeparatorHdrBase
- ThemeColSeparatorLineBase
- ThemeColSortHdrDisabled
- ThemeColSortHdrHighlight
- ThemeColSortHdrNormal
- ThemeColSortHdrPressed
- ThemeColTextActive
- ThemeColTextCaption
- ThemeColTextCaptionActive
- ThemeColTextCaptionInactive
- ThemeColTextChecked
- ThemeColTextDisabled
- ThemeColTextError
- ThemeColTextFocussed
- ThemeColTextHdrDisabled

- [ThemeColTextHdrHighlight](#)
- [ThemeColTextHdrNormal](#)
- [ThemeColTextHdrPressed](#)
- [ThemeColTextHighlight](#)
- [ThemeColTextHover](#)
- [ThemeColTextInactive](#)
- [ThemeColTextNormal](#)
- [ThemeColTextPressed](#)
- [ThemeColTextRulerNormal](#)
- [ThemeColTextSelectedActive](#)
- [ThemeColTextSelectedInactive](#)
- [ThemeColTextVisited](#)
- [ThemeColTickDisabled](#)
- [ThemeColTickNormal](#)
- [ThemeFontBox](#)
- [ThemeFontCaption](#)
- [ThemeFontIvl](#)
- [ThemeFontText](#)
- [ThemeFontTextHdr](#)
- [ThemeOptFillShade](#)
- [ThemeSizeBorderChecked](#)
- [ThemeSizeBorderDefault](#)
- [ThemeSizeBorderDisabled](#)
- [ThemeSizeBorderHighlight](#)
- [ThemeSizeBorderNormal](#)
- [ThemeSizeBorderPressed](#)
- [ThemeSizeFixed](#)



Je nach ausgewähltem Objekttyp stehen unterschiedliche Eigenschaften zur Verfügung.

In der Liste der Eigenschaften werden bis zu 10 ThemeSets angezeigt. Sind mehr ThemeSets im Theme vorhanden, bleiben die nicht angezeigten Themesets beim Speichern erhalten. Wird in der Liste eine Eigenschaft eines ThemeSets ausgewählt, wird das entsprechende ThemeSet hervorgehoben. Die Vorschau zeigt jeweils das ausgewählte ThemeSet an. Ist kein ThemeSet ausgewählt, wird die Vorschau des Theme angezeigt.

Eigenschaften, die bei den ThemeSets auf den Null-Wert gesetzt sind, werden vom Theme übernommen. Je nach Typ kann bei den ein anderer Wert als Null-Wert ausgewählt werden:

<u>WinColUndefined</u> (_WinColUndefined.png)	Farb-Eigenschaften
<Null>	Font-Eigenschaften
leeres Feld	Integer-Eigenschaften
Vorschau	

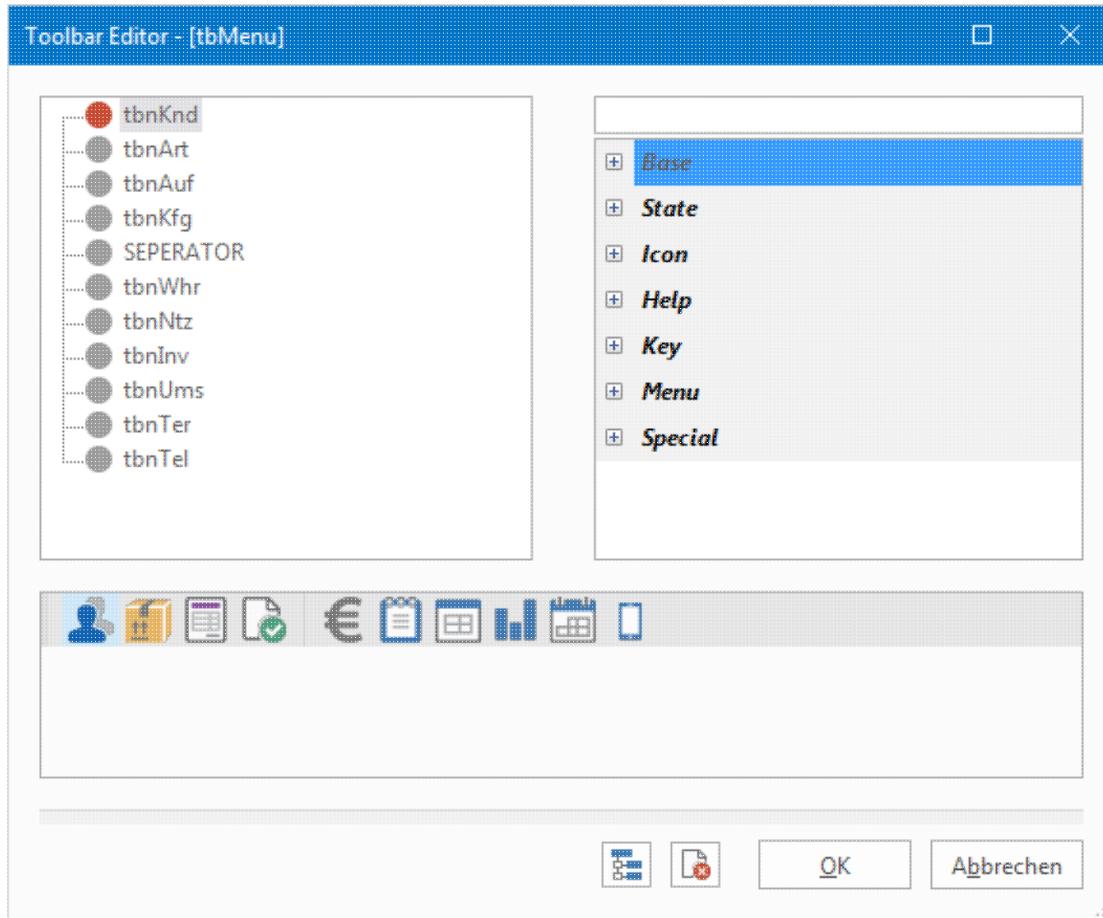
Im Vorschaufenster wird der Objekttyp in allen möglichen Darstellungsformen angezeigt. Änderungen können sofort im Vorschaufenster überprüft werden.

Kontakt

Toolbar-Editor

Editor zum Bearbeiten der Toolbar

Der Toolbar-Editor kann im Kontextmenü des Toolbar-Objektes über die Menüoption **Toolbar bearbeiten...** oder mit einem Doppelklicken auf das Objekt aufgerufen werden.



Der Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird die Toolbar nicht gespeichert. Mit [OK] wird die Werkzeugleiste gespeichert. Einzelne Schaltflächen können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Liste der Schaltflächen

Die Schaltflächen der Toolbar sind im linken Teil des Editors aufgeführt.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neue Schaltfläche wird eingefügt.



Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften der Schaltfläche oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.

Kontakt

- Die Schaltfläche wird noch oben verschoben.
- Die Schaltfläche wird nach unten verschoben.
- Die Selektion wird nach oben erweitert.
- Die Selektion wird nach unten erweitert.
- Die Suche wird gestartet.
- Suche nach dem nächsten Vorkommen.
- Suche nach dem vorherigen Vorkommen.

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Schaltflächen können mit Hilfe der Zwischenablage in andere Toolbar-Objekte eingefügt werden. Mehrere Objekte können nacheinander mit der Tastenkombination (des Zahlenblocks) in die Zwischenablage kopiert werden.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

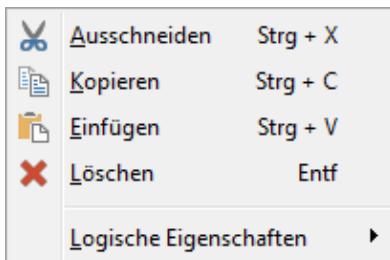
Wird eine Schaltfläche mit der Maus gezogen, kann er an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Eintrages kann durch Anklicken des ausgewählten Eintrags geändert werden.

Wird in der Liste der Schaltflächen die Taste oder die Tastenkombination gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf die entsprechende Schaltfläche positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Menüpunkte.

- **Kopieren**

Kopieren der selektierten Menüpunkte.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Menüpunkten.

- **Löschen**

Löschen der selektierten Menüpunkte.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften des Menüpunktes zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften der selektierten Schaltfläche sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name des Objekts
- Custom Benutzerdefinierte Eigenschaft
- Caption Ausgabetext
- ImageTileUser Verwendetes benutzerdefiniertes Bild
- Disabled Schaltfläche deaktiviert
- Visible Sichtbarkeit der Schaltfläche
- ReadOnly Drücken nicht möglich
- HelpTip Bubble-Help angeben
- StatustItemText Ausgabetext in der Statuszeile
- MenuKey Tastenkombination
- MenuItemID ID der Schaltfläche
- MenuAction Menüaktion
- MenuSeparator Objekt als Separator
- MenuCheck Check-Markierung
- Group Gruppe

- Size Breite der Schaltfläche
- ImageTile Verwendetes Bild
- ImageOption Darstellungsoption des Bildes

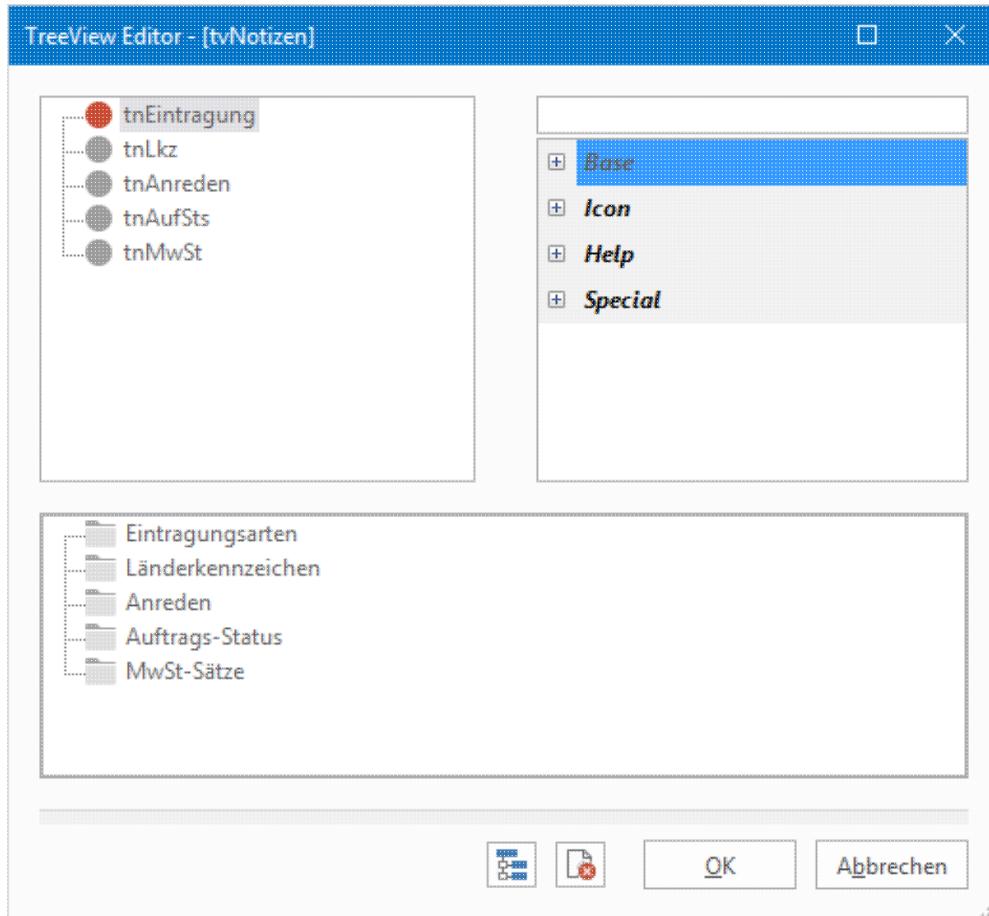
Vorschau

Im Vorschaufenster wird die Toolbar dargestellt, die momentan editiert wird. In diesem Fenster kann überprüft werden, ob die Toolbar das gewünschte Aussehen hat.

TreeView-Editor

Editor zum Bearbeiten eines TreeViews

Der TreeView-Editor kann im Kontextmenü des TreeView-Objekts über die Menüoption **Knoten bearbeiten...** oder mit einem Doppelklicken auf das Objekt aufgerufen werden.



Das TreeView-Objekt besteht aus einem Container (das TreeView-Objekt selbst und einem oder mehreren Knoten). Die Knoten werden durch den Editor oder prozedural erzeugt.

Der TreeView-Editor wird mit den Schaltflächen [OK] oder [Abbrechen] verlassen. Durch Drücken der Schaltfläche [Abbrechen] wird das TreeView nicht gespeichert. Mit [OK] wird das TreeView gespeichert. Einzelne Knoten können mit der Schaltfläche generiert und mit der Schaltfläche gelöscht werden.

Liste der Knoten

Die Knoten des TreeViews sind im linken Teil des Editors aufgeführt.

Folgende Tastaturkommandos stehen im Editor zur Verfügung:



Eine neuer Knoten wird eingefügt.

Kontakt

Namen des Eintrags ändern. Der Name kann ebenfalls bei den Eigenschaften des Knoten oder durch Anklicken des ausgewählten Eintrags mit der Maus geändert werden.



Der ausgewählte Eintrag wird gelöscht.



Der Knoten wird dem darüberliegenden Eintrag untergeordnet.



Der Knoten wird in der Hierarchie nach oben verschoben.



Der Knoten wird in der Reihenfolge nach oben verschoben.



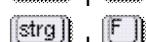
Der Knoten wird in der Reihenfolge nach unten verschoben.



Die Selektion wird nach oben erweitert.



Die Selektion wird nach unten erweitert.



Die Suche wird gestartet.



Suche nach dem nächsten Vorkommen.



Suche nach dem vorherigen Vorkommen.

Darüber hinaus stehen die in Windows üblichen Tastaturkommandos zum Ausschneiden, Kopieren und Einfügen zur Verfügung. Einzelne Knoten können mit Hilfe der Zwischenablage in andere TreeView-Objekte eingefügt werden. Mehrere Objekte können nacheinander mit der Tastenkombination (des Zahlenblocks) in die Zwischenablage kopiert werden.

Die Operationen lassen sich ebenfalls über verschiedene Mauskommandos durchführen.

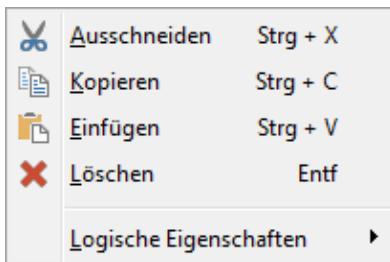
Wird ein Knoten mit der Maus gezogen, kann sie an die neue Stelle verschoben werden, wenn die neue Position mit einem schwarzen Balken markiert ist. Wird dabei die -Taste gedrückt gehalten, wird der Eintrag kopiert.

Der Name eines Knotens kann durch Anklicken des ausgewählten Knotens geändert werden.

Wird in der Liste der Knoten die Taste oder die Tastenkombination gedrückt, kann in einem Eingabefeld ein Name aus der Liste angegeben werden. Nach Abschließen der Eingabe mit wird auf den entsprechenden Knoten positioniert. Die Verwendung von Wildcards (*, ?) ist bei der Suche möglich. Sind keine Wildcards angegeben wird nach allen Stellen gesucht, die den Suchbegriff enthalten (*<Begriff>*).

Mit den Tasten und wird nach dem nächsten oder vorherigen Vorkommen des Suchbegriffs in der Liste gesucht. Ist das Ende bzw. der Anfang der Liste erreicht, wird die Suche am jeweils anderen Ende fortgesetzt. Eine neue Suche wird mit der Tastenkombination gestartet. Die Groß-Kleinschreibung wird bei der Suche ignoriert.

Über das Kontextmenü stehen folgende Optionen zur Verfügung:



- **Ausschneiden**

Ausschneiden der selektierten Knoten.

- **Kopieren**

Kopieren der selektierten Knoten.

- **Einfügen**

Einfügen von vorher ausgeschnittenen bzw. kopierten Knoten.

- **Löschen**

Löschen der selektierten Knoten.

- **Logische Eigenschaften**

Liste der logischen Eigenschaften der Knoten zur direkten Bearbeitung.

Eigenschafts-Filter

In dem Feld kann der Name einer Eigenschaft eingegeben werden, um nach dieser Eigenschaft zu filtern. Die Suche nach Eigenschaften ist auch in Kombination mit Wildcards möglich. Sind keine Wildcards angegeben, wird nach <Eigenschaft>* gesucht.

Wurde ein Suchbegriff angegeben, erscheint auf der rechten Seite des Feldes eine -Schaltfläche um den Filter zu entfernen. Alternativ kann auch das Feld geleert werden.

Liste der Eigenschaften

Die Eigenschaften des selektierten Eintrags sind in einer Liste auf der rechten Seite aufgeführt. Durch Anklicken eines Wertes kann dieser geändert werden. Die Sortierung der Liste kann über ein Kontextmenü geändert werden:

Folgende Eigenschaften stehen zur Verfügung:

- Name Name des Objektes
- Caption Text des Knotens
- ID ID des Knotens
- NodeStyle Verwendetes Bild
- NodeExpanded Knoten geöffnet
- NodeDynamic Dynamischer Knoten

Kontakt

- ImageTileUser Verwendetes benutzerdefiniertes Bild
- HelpTip Text des HelpTip
- Custom benutzerdefinierte Eigenschaft

Testbereich

Im Vorschaufenster wird das bearbeitete TreeView angezeigt. Änderungen können sofort im Vorschaufenster überprüft werden.

Kontakt

Designer - Formular-Designer

Erstellen von Reporten und anderen Dokumenten.

Der Formular-Designer ist integraler Bestandteil des Designers. Er wird automatisch aufgerufen, wenn ein Druck-Objekt angelegt oder selektiert wird.

Das Ausdrucken von Dokumenten teilt sich in drei Bereiche:

- Druckobjekte erstellen
- Anzeige der Druckvorschau
- Ausgabe auf einen Drucker

Die Druckobjekte werden im Designer mit dem Formular-Designer erstellt. Die Anzeige in der Druckvorschau und die Ausgabe auf einen Drucker werden durch Prozedurbefehle durchgeführt.

Formular-Designer - Druckobjekte erstellen
Erstellen und Ändern von Druckvorlagen

Formular-Designer,

Siehe Anzeige der
Druckvorschau

Mit dem Formular-Designer können drei unterschiedliche Formular-Objekte bearbeitet werden:

- **PrintDoc**

Mit diesem Objekt kann ein Dokument mit umfangreicherem Text erstellt werden. Dieses Objekt sollte bei der Generierung eines Briefes oder ähnlichem verwendet werden.

- **PrintDocRecord**

Soll eine Liste gedruckt werden bietet sich dieses Objekt an. Es besteht aus einer Tabelle, die entsprechend konfiguriert werden kann.

- **PrintForm**

Ein umfangreicher Druckjob kann aus mehreren Textbausteinen (PrintForm-Objekten) zusammengesetzt werden.

Erstellen eines Dokumentes (PrintDoc)

Das PrintDoc-Objekt wird über den Menüpunkt **Datei / Neu / PrintDoc** angelegt. In dieses Objekt können nur Seiten-Objekte eingefügt werden.

Innerhalb eines Seiten-Objektes können weitere Objekte definiert werden. Mit den Objekten PrtPageHeader und PrtPageFooter können entsprechende Bereiche für eine Kopf- und eine Fußzeile reserviert werden. In diesen Objekten und in der Seite selbst können weitere Objekte zur Darstellung von Text, Bildern usw. (siehe Druck-Objekte) angelegt und frei auf der Seite platziert werden.

Ein so zusammengestelltes Formular kann in der Datenbank über den Menüpunkt **Datei / Speichern** oder **Datei / Speichern unter** gespeichert werden.

Dieses Objekt kann mit dem Befehl PrtJobOpen() geladen und anschließend die Eigenschaften aller Objekte des Druck-Jobs verändert werden. Die gespeicherte Vorlage kann so angepasst und mit den Befehlen PrtJobWrite() und PrtJobClose() geschrieben bzw. ausgedruckt werden.

Beispiel:

```
// Druckjob öffnetPrtJob # PrtJobOpen('Letter', '', _PrtJobOpenWrite, _PrtTypePrintDoc); // Druck
```

Sollen Änderungen innerhalb des Dokumentes vorgenommen werden, müssen diese Änderungen vor den Schreiben des Dokumentes mit PrtJobWrite() erfolgen.

Der Suchpfad für die Objekte wird beim Laden des Druckobjektes nicht umgesetzt. Soll also über die Namen der Objekte innerhalb des Objekts Letter referenziert

Kontakt

werden, muss der Suchpfad mit dem Befehl WinSearchPath() auf den Druckjob gesetzt werden. Anschließend wird zum Beispiel das für die Empfängeradresse vorgesehene PrtText-Objekt verändert:

```
tPrtJob->WinSearchPath(); $ptReceiver->ppCaption # 'Firma vectorsoft AG' + StrChar(13) + StrChar(10)
```

Zum Schluss wird der Suchpfad wieder gelöscht.

Erstellen einer Liste (PrintDocRecord)

Beim Anlegen eines PrintDocRecord-Objekts wird ein Dialog aufgerufen, in dem verschiedene Einstellungen der Liste vorgenommen werden können.

Über diesen Dialog können die einzelnen Bereiche der Liste konfiguriert werden:

- **Tabellenkopf**

Der Tabellenkopf wird zu Beginn der Tabelle gedruckt.

- **Seitenkopf**

Der Seitenkopf wird zu Beginn jeder Seite gedruckt. Sollten Tabellenkopf und Seitenkopf auf einer Seite gedruckt werden, wird der Tabellenkopf vor dem Seitenkopf gedruckt.

- **Datenkopf**

Nach dem Seitenkopf wird der Datenkopf gedruckt. Hier können Spaltenüberschriften hinterlegt werden. Der Datenkopf wird vor jedem Datenbereich gedruckt.

- **Tabellenfuß**

Der Tabellenfuß wird am Ende der Tabelle gedruckt. Der Ausdruck erfolgt nach dem Seitenfuß der letzten Seite.

- **Seitenfuß**

Der Seitenfuß wird am Ende jeder Seite gedruckt.

- **Datenfuß**

Der Datenfuß wird nach jedem Datenbereich gedruckt. Der Ausdruck erfolgt vor dem Seitenfuß.

- **Datenbereich**

Der Datenbereich wird für jeden Datensatz der Tabelle gedruckt.

Folgende Einstellungen können vorgenommen werden:

- **Mit diesen Einstellungen automatisch erstellen**

Ist diese Checkbox aktiviert, werden die Einstellungen zu diesem Element nicht mehr abgefragt.

- **Mit Tabelle erstellen**

Kontakt

Das auf der linken Seite ausgewählte Objekt wird mit der Tabelle erzeugt. Soll zum Beispiel eine Tabelle ohne Seitenkopf erstellt werden, kann das Element über diese Checkbox deaktiviert werden.

• Spalten

Das ausgewählte Objekt wird mit der angegebenen Anzahl von Spalten generiert. Die Anzahl der Spalten ist von der Anzahl der Spalten in den anderen Bereichen unabhängig.

• Zeilen

Das ausgewählte Objekt wird mit der angegebenen Anzahl von Zeilen generiert.

• Zellbreite

Jede Zelle des ausgewählten Objektes wird mit der hier angegebenen Breite angelegt. Dies kann auch dazu führen, dass Zeilen breiter als die Seite angelegt werden.

Wird hier nichts angegeben, werden die Zellen in einer Breite angelegt, die die Seitenbreite ausnutzt.

• Zellhöhe

Hier kann die Höhe einer Zelle in dem ausgewählten Bereich angegeben werden.

• Zellhöhe-Modus

Die Höhe der Zelle kann entweder automatisch erfolgen oder konstant angegeben werden. Wird die Zellhöhe automatisch bestimmt, ist sie abhängig von der Höhe des enthaltenen Objekts.

Nach Bestätigen der Einstellungen mit [OK] wird die Tabelle erzeugt. Alle Elemente können auch zu einem späteren Zeitpunkt in die Tabelle integriert werden. Im Fenster "Palette" stehen diese Objekte auf der Seite "Tabelle" zur Verfügung.

Die angegebene Anzahl von Spalten werden gleichmäßig über die gesamte Seitenbreite angeordnet. Die Breite der Spalten kann entweder mit der Maus oder über die Eigenschaft Width verändert werden. Wird beim Ändern der Breite einer Spalte mit der Maus die -Taste gedrückt, werden alle übereinander liegenden Spalten mit verändert.

Bestehende Spalten oder Zeilen können über das Kontextmenü kopiert werden. Wird der Mauszeiger direkt vor eine Zeile positioniert, verändert er sich zu einem schwarzen Pfeil. Durch Drücken der linken Maustaste wird dann die gesamte Zeile markiert. Zur Unterscheidung, ob ein ganzer Bereich (zum Beispiel Data-Bereich oder HD-Bereich) oder eine Zeile markiert ist, erfolgt die Bereichsmarkierung in grau und die Zeilenmarkierung in blau.

Ebenso wie Zeilen können auch Spalten markiert werden. Dabei ist zu beachten, dass immer nur Spalten eines Bereiches markiert werden können. Der Mauszeiger verändert sich zu einem Pfeil nach unten, wenn er über einer Spalte positioniert wird.

Kontakt

Ist ein PrtTblCell-Objekt selektiert, kann über das Kontextmenü die Markierung auf die gesamte Zeile bzw. auf die gesamte Spalte ausgedehnt werden.

Das Kopieren des markierten Bereiches erfolgt über die Einträge "Zeilen duplizieren" bzw. "Spalten duplizieren". Alle in den Zellen enthaltenen Objekte werden ebenfalls kopiert.

Ist eine Spalte oder eine Zelle markiert, kann diese mit "Spalten teilen" aus den Kontextmenü in der Mitte geteilt werden.

Soll das gleiche Objekt in mehrere Zellen eingefügt werden, können die entsprechenden Zellen markiert und anschließend das Objekt aus der Palette eingefügt werden. In jeder Zelle wird das entsprechende Objekt erzeugt.

Die vorgenommenen Einstellungen werden in den Einstellungen des Designers gespeichert. Beim Erstellen der nächsten Tabelle werden die entsprechenden Einstellungen übernommen.

Um in den erzeugten Zellen Texte oder Bilder ausgeben zu können, müssen entsprechende Objekte in den Zellen erzeugt werden. Diese Objekte stehen im Fenster "Palette" auf der Seite "Druck" zur Verfügung. Werden die Objekte über die "Palette" eingefügt, muss eine Verbindung zur Datenstruktur über die Eigenschaft DbFieldName hergestellt werden. Die Verbindung wird automatisch hergestellt, wenn aus der Datenstruktur die entsprechenden Felder in die Zellen gezogen werden.

Das PrintDocRecord-Objekt liest alle Datensätze einer Datei oder einer Verknüpfung, wenn die Eigenschaften DbFileNo, DbKeyNo und DbLinkFileNo entsprechend gesetzt sind.

Eine Liste aller Datensätze kann dann mit folgenden Befehlen ausgegeben werden:

```
// Druckjob öffnetPrtJob # PrtJobOpen('Tabelle', '', _PrtJobOpenWrite, _PrtTypePrintDocRecord);
```

Zur Aufbereitung von komplexeren Listen, zum Beispiel die Ausgabe aller Datensätze einer Datei und den dazugehörigen verknüpften Datensätzen, ist das PrintForm-Objekt besser geeignet.

Erstellen von Textbausteinen (PrintForm)

Soll ein Dokument aus vielen unterschiedlichen Bestandteilen zusammengesetzt werden, kommen Textbausteine zum Einsatz. Mit Hilfe dieser Textbausteine können, wie mit dem Objekt PrintDocRecord Listen oder andere Formulare zusammengestellt werden.

Zunächst werden ein oder mehrere PrintForm-Objekte erstellt. Jedes dieser Objekte kann Texte, Bilder usw. enthalten. Die Höhe des PrintDoc-Objekts wird mit einem blauen Rahmen angezeigt. Werden Objekte aus dem PrintDoc-Objekt entfernt, verkleinert sich nicht automatisch die Höhe des Objekts. Dies kann über das Kontextmenü mit dem Eintrag **Formgröße reduzieren** oder über Einträge in den Eigenschaften FormHeight und FormWidth erfolgen. Die einzelnen PrintForm-Objekte werden prozedural zu einem ganzen Druckjob zusammengestellt.

Kontakt

Um einen solchen Druckjob zusammenzustellen, wird mit dem Befehl PrtJobOpen() ein Druckjob geöffnet. Anschließend wird mit PrtFormOpen() die PrintForm geladen. Diese kann jetzt mit dem Befehl PrtAdd() mehrfach zu dem Druckjob hinzugefügt werden.

Abschließend wird die PrintForm mit PrtFormClose() entladen und der Druckjob mit PrtJobClose() beendet.

Beispiel:

```
// Druckjob und Textbaustein öffnenJob # PrtJobOpen(_PrtDocDinA4, '', _PrtJobOpenWrite, _PrtType)
```

Auf diese Weise kann ein Druckjob aus ein oder mehreren PrintForm-Objekten zusammengesetzt werden. Der Seitenumbruch findet in diesem Beispiel erst statt, wenn eine PrintForm nicht mehr auf eine Seite gedruckt werden kann. Soll am Ende einer Seite ein Seitenfuß gedruckt werden, muss über die Eigenschaften BoundAdd und BoundMax der von PrtJobWrite(_PrtJobPageStart) zurückgegebenen Druckseite der noch auf der Seite zur Verfügung stehende Platz mit der Höhe (Eigenschaft FormHeight) der PrintForm und des Seitenfußes verglichen werden. Reicht der Platz nicht mehr aus, muss der Seitenfuß gedruckt und mit dem Befehl PrtJobWrite(_PrtJobPageBreak) ein Seitenumbruch eingefügt werden. Vor dem Abschluss der Druckaufbereitung muss die aktuelle Seite mit dem Befehl PrtJobWrite(_PrtJobPageEnd) abgeschlossen werden. Ein ausführlicheres Beispiel zur Programmierung befindet sich [hier](#).

Kontakt

Formular-Designer - Anzeige der Druckvorschau
Druckvorschau anzeigen und anpassen

Druckobjekte

erstellen,

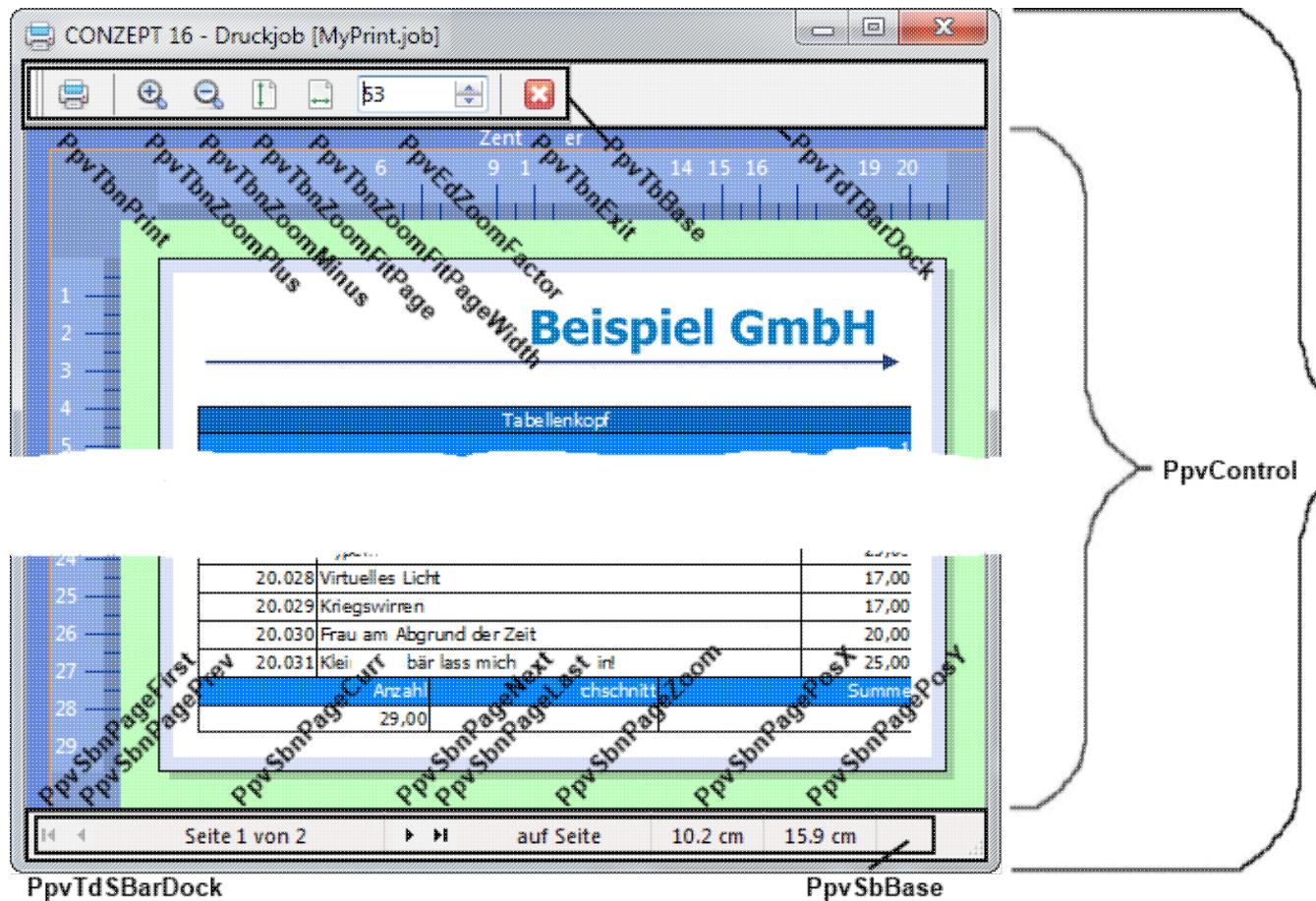
Siehe Ausgabe auf
einen
Drucker

Beim Schließen eines Druckjobs wird die Druckvorschau angezeigt, wenn entweder ein temporärer Druckjob (Angabe von PrtJobOpenTemp bei PrtJobOpen()) erzeugt oder beim Befehl PrtJobClose() die Option PrtJobPreview angegeben wurde.

Wird die Druckvorschau angezeigt, bleibt die Verarbeitung der Prozedur bei dem Befehl PrtJobClose() stehen. Die Elemente der Druckvorschau können zuvor angepasst werden.

Nach dem Öffnen eines Druckjobs mit dem Befehl PrtJobOpen() kann der Deskriptor der Druckvorschau mit dem Befehl PrtInfo(..., PrtFrame) ermittelt werden.

Ausgehend von diesem Deskriptor können die Eigenschaften des Fensters verändert oder die Deskriptoren der Objekte innerhalb der Vorschau ermittelt werden. Die einzelnen Objekte können über ihre Namen angesprochen werden. Die Sprache des Dialogs kann über die Eigenschaft LangDisplay des Application-Objekts geändert werden. Der Dialog verfügt über Unicode-Unterstützung.



Kontakt

Die Druckvorschau besteht aus folgenden Objekten:

Name	Objekttyp	Beschreibung
PreviewDlg	<u>Frame</u>	Frame-Objekt der Druckvorschau
PpvControl	<u>PrtJobPreview</u>	Preview-Objekt
PpvTdTBarDock	<u>ToolbarDock</u>	ToolbarDock-Objekt mit den Steuerschaltflächen
PpvTbBase	<u>Toolbar</u>	Toolbar-Objekt mit den Steuerschaltflächen
PpvTbnPrint	<u>Toolbar-Button</u>	Schaltfläche "Drucken"
PpvTbnZoomPlus	<u>Toolbar-Button</u>	Schaltfläche "Vergrößern"
PpvTbnZoomMinus	<u>Toolbar-Button</u>	Schaltfläche "Verkleinern"
PpvTbnZoomFitPage	<u>Toolbar-Button</u>	Schaltfläche "Ganze Seite"
PpvTbnZoomFitPageWidth	<u>Toolbar-Button</u>	Schaltfläche "Seitenbreite"
PpvEdZoomFactor	<u>IntEdit</u>	Eingabefeld Zoomfaktor
PpvTbnExit	<u>Toolbar-Button</u>	Schaltfläche "Druckvorschau beenden"
PpvTdSBarDock	<u>ToolbarDock</u>	StatusbarDock-Objekt mit den Steuerschaltflächen
PpvSBase	<u>Statusbar</u>	Statusbar-Objekt mit den Steuerschaltflächen
PpvSbnPageFirst	<u>Statusbar-Button</u>	Schaltfläche zum Blättern auf die erste Seite
PpvSbnPagePrev	<u>Statusbar-Button</u>	Schaltfläche zum Blättern auf die vorherige Seite
PpvSbnPageCurr	<u>Statusbar-Button</u>	Anzeige aktuelle Seitennummer
PpvSbnPageNext	<u>Statusbar-Button</u>	Schaltfläche zum Blättern auf die nächste Seite
PpvSbnPageLast	<u>Statusbar-Button</u>	Schaltfläche zum Blättern auf die letzte Seite
PpvSbnPageZoom	<u>Statusbar-Button</u>	Anzeige Zoom-Faktor
PpvSbnPagePosX	<u>Statusbar-Button</u>	Anzeige der X-Koordinate des Mauszeigers
PpvSbnPagePosY	<u>Statusbar-Button</u>	Anzeige der Y-Koordinate des Mauszeigers
PpvComPrint	<u>Frame</u>	Dialog des Druckers. Hier können die gleichen Eigenschaften wie beim <u>Standard-Dialog</u> gesetzt werden.

Alle Elemente der Druckvorschau können angepasst werden.

Beispiel:

```
// Druckjob öffnetPrtJob # PrtJobOpen('Letter', '', PrtJobOpenWrite, PrtTypePrintDoc); // Zoom
$PpvTbnZoomFitPage->wpCaption # 'Page &full'$PpvTbnZoomFitPage->wpImageOption # _Wi
```

Bestimmte Eigenschaften müssen vor dem Schreiben des Druckjobs beim Dokument selbst angegeben werden. Dazu gehört auch der Zoomfaktor. Über die Objekte

Kontakt

\$PpvEdZoomFactor und \$PpvSbnPageZoom kann der Vergrößerungsfaktor nur ausgelesen werden. Der Vergrößerungsfaktor muss beim Dokument selber in der Eigenschaft ZoomFactor gesetzt werden. Genauso wird dort unabhängig vom verwendeten Drucker das Papierformat (PageFormat) und das Hoch- oder Querformat (Orientation) festgelegt.

Gedruckt wird das Dokument über die Schaltflächen [Drucken]. Sie öffnet einen Auswahldialog, in dem einer, der auf dem System installierten Drucker ausgewählt werden kann. Dies kann sowohl ein Drucker, als auch ein Fax-Drucker, PDF-Writer usw. sein. Voreingestellt ist der Standarddrucker des Systems. Andere Drucker können durch die Angabe einer Device beim Befehl PrtJobClose() voreingestellt werden (siehe Ausgabe auf einen Drucker).

Formular-Designer - Ausgabe auf den Drucker
Auswahl und Einstellungen eines Druckers.

Anzeige der

Siehe Druckvorschau,

Formular-Designer

Ein Dokument kann entweder in der Druckvorschau durch Drücken der Schaltfläche [Drucken] oder ohne die Druckvorschau durch Angabe einer Ausgabeeinheit beim Befehl PrtJobClose() ausgedruckt werden. Werden sowohl die Druckvorschau angezeigt, als auch die Ausgabeeinheit gesetzt, wird der entsprechende Drucker im Auswahldialog voreingestellt.

Beispiel:

```
tPrtDevice # PrtDeviceOpen('Acrobat PDFWriter', _PrtDeviceSystem); tPrtJob # PrtJobOpen('Letter',
```

In diesem Beispiel wird die Druckvorschau angezeigt und der Druckertreiber "Acrobat PDFWriter" als Ausgabeeinheit voreingestellt. Der Benutzer hat vor dem Drucken noch die Möglichkeit einen anderen Drucker auszuwählen.

Wird beim Befehl PrtJobClose() die Konstante PrtJobPreview durch PrtJobPrint ersetzt, wird ohne Anzeige der Vorschau der Druckjob sofort auf den angegebenen Drucker ausgedruckt:

```
tPrtDevice # PrtDeviceOpen('Acrobat PDFWriter', _PrtDeviceSystem); tPrtJob # PrtJobOpen('Letter',
```

Der Rückgabewert der Anweisung PrtJobClose() gibt Auskunft darüber, ob der Druckjob an den Drucker übergeben werden konnte. Hat zum Beispiel der Druckjob ein Seitenformat, das vom Druckertreiber nicht unterstützt wird, wird der Fehler ErrPrtPaperFormat zurückgegeben.

In diesem speziellen Fall wird noch ein Dialog zur Eingabe des Dateinamens der erzeugten PDF-Datei angezeigt. Dieser Dialog wird nicht angezeigt, wenn beim Druckjob die Eigenschaft PdfFileName gesetzt ist.

Die Angabe des Druckernamens und der Konstante kann entfallen, wenn der Standarddrucker geöffnet werden soll.

Die Namen der installierten Drucker können mit dem Befehl PrtInfo() der Druckerliste ermittelt werden. Die Druckerliste ist beim Applikations-Objekt (App) abgelegt. Ein DataList-Objekt kann mit folgenden Anweisungen mit den Namen aller installierten Druckern gefüllt werden:

```
// Druckerliste ermittelntPrinterList # _App->PrinterList(_PrtListRefresh); // Ersten Drucker les
```

Ist der Name des Druckers ermittelt, kann die Ausgabeeinheit mit dem Befehl PrtDeviceOpen() geöffnet werden. Dabei muss die Konstante PrtDeviceSystem angegeben werden. Das Printer-Objekt stellt Informationen zu dem Drucker (zum Beispiel den Status und die Anzahl der gerade verarbeiteten Druckjobs) zur Verfügung, während über das PrintDevice-Objekt die Einstellungen des Druckers (zum Beispiel das Papierformat und die Anzahl der zu druckenden Kopien) abgefragt und gesetzt werden können.

Kontakt

Die Verarbeitung eines Druckjobs erfolgt auf zwei Ebenen:

- **Dokument-Ebene**
- **Drucker-Ebene**

Bei der Erstellung eines Dokumentes mit Hilfe der Formular-Objekte wird mit dem Schließen des Druckjobs (PrtJobClose()) ein Drucker unabhängiger Job erzeugt. Wird dieser Job in einer externen Datei gespeichert, wird die Drucker-Ebene nicht angesprochen.

Erst wenn der unabhängige Job an eine Ausgabeeinheit (PrintDevice), welche die für diesen Ausdruck notwendigen Einstellungen aufweist (zum Beispiel des Seitenformates), weitergeleitet wird, erfolgt die Weitergabe an die Drucker-Ebene. Hier findet unter anderem die Verbindung vom Papierformat des Druckjobs und der von Drucker unterstützten Papierformate statt.

Befehle und Funktionen

Befehle und Funktionen

Alphabetische

Siehe Liste aller

Befehle

Die Befehle und Funktionen gliedern sich in folgende Gruppen:

- Befehle für binäre Objekte
- Befehle für dynamische Strukturen
- Befehle für Memory-Objekte
- Befehle und Konstanten für Fehlermeldungen
- Druckfunktionen
- Funktionen auf Datenbankelemente
- Funktionen der Benutzeroberfläche
- Funktionen der Systemumgebung
- Funktionen und Operatoren
- Liste aller Prozedurbefehle
- Schnittstellen und Kommunikation
- Standardsprachelemente
- Validierungsbefehle

Die CONZEPT 16-Anweisungen sind alle in dem gleichen Schema beschrieben. Im Kopf der Beschreibung befindet sich ein Syntax-Diagramm mit den Datentypen der Übergabeparameter und des Rückgabewertes angegeben werden. Eine kurze Beschreibung befindet sich unter dem Syntax-Diagramm.

RecRead(int1, int2, int3[, bigint4[, int5]]) : int	
Datensatz lesen	
int1	Dateinummer oder Deskriptor eines Datensatzpuffers
int2	Schlüsselnummer / Selektions-Deskriptor
int3	Optionen (siehe Text)
bigint4	Datensatzposition / Filter-Deskriptor / Datensatz-ID (optional)
int5	Anzahl der Sätze bei <u>_RecPrev</u> und <u>_RecNext</u> (optional)
Resultat	int Leseresultat <u>_rOk</u> Lesen erfolgreich <u>_rLocked</u> Datensatz gesperrt <u>_rMultiKey</u> Schlüssel nicht eindeutig <u>_rNoKey</u> Schlüssel nicht vorhanden, nächsten Datensatz geladen <u>_rLastRec</u> Schlüssel nicht vorhanden, letzten Datensatz geladen <u>_rNoRec</u> Kein (weiterer) Datensatz vorhanden <u>_rNoLock</u> Datensatz ist nicht gesperrt <u>_rNoRights</u> Benutzerrechte nicht ausreichend
Siehe	<u>Verwandte Befehle</u> , <u>RecBufCreate()</u> , <u>RecLink()</u> , <u>SelOpen()</u> , <u>Datensatzpuffer</u> , <u>Datensatzzugriff</u>

Kontakt

Das Syntax-Diagramm zeigt in diesem Fall maximal sechs Parameter an. Bis auf Parameter 4 sind alle Parameter vom Datentyp int. Die letzten beiden Parameter sind optional, sie werden in [...] eingeklammert. Danach wird der Kontext angezeigt, in dem die Anweisung ausgeführt werden kann. Nicht in jedem Kontext stehen alle Prozedurbefehle zur Verfügung. Die angezeigten Symbole haben folgende Bedeutung:

-  Ausführbar von Clients mit grafischer Oberfläche. Dies sind der Standard- und der Advanced-Client.
-  Ausführbar vom Server (siehe RmtCall()).
-  Ausführbar von Clients mit Programmierschnittstelle. Diese sind die DLL- (siehe C16_ProcCall()) und die PHP-Schnittstelle (siehe c16_call()).
-  Ausführbar von der DLL-Schnittstelle (siehe C16_ProcCall()) mit aktiverter grafischer Erweiterung.
-  Ausführbar von der Web-Schnittstelle (siehe C16_CALL()).
-  Ausführbar vom Druckprozessor.
-  Ausführbar von der ODBC-Schnittstelle.
-  Ausführbar vom SOA-Service.
-  Ausführbar vom SOA-Service mit aktiverter grafischer Erweiterung.

In der nächsten Zeile wird eine kurze Beschreibung der Anweisung ausgegeben. Anschließend erfolgt eine Übersicht über die Bedeutung der einzelnen Übergabeparameter. Hier finden sich ebenfalls mögliche Konstanten, die in dem Parameter übergeben werden können.

Innerhalb der Resultat-Zeile, wird angezeigt, welchen Datentyp der Rückgaberwert hat und wie der Rückgabewert auszuwerten ist. Mit dem Symbol  wird angezeigt, dass die Anweisung den globalen Fehlerwert setzt.

Darunter befinden sich Verweise zu weiteren Hilfeseiten. Der Verweis "Verwandte Befehle" führt dabei immer zu einer Liste von Anweisungen, die in den gleichen Bereich fallen (in dem Beispiel zu einer Liste mit weiteren Datensatz-Befehlen).

Standardsprachelemente

Standardsprachelemente

Siehe [Befehlsgruppen](#),
[Befehlsliste](#)

- [Befehle für globale Variablen](#)
- [Befehle zur Prozedursteuerung](#)
- [Compiler-Anweisungen](#)
- [Datentypen](#)
- [Debugger-Befehle](#)
- [Konvertierungsbefehle](#)
- [Schlüsselwörter](#)
- [Style-Unterschiede](#)

Compiler-Anweisungen

Verwendung von Compiler-Anweisungen

Über Compiler-Anweisungen wird die Übersetzung der Prozedur gesteuert. Alle Compileranweisungen beginnen mit dem Zeichen @ und müssen in der ersten Spalte einer Zeile stehen.

@A+ oder @A-

Diese Anweisung schaltet auf den A+ Compiler um (@A+). Im Designer wird standardmäßig der A+ Compiler verwendet.

@I:<Name>

Mit dieser Anweisung wird ein anderer Prozedurtext an der entsprechenden Stelle zur Übersetzungszeit eingebunden. <Name> ist dabei der Name der einzubindenden Prozedur. Die Anzahl von @I-Anweisungen ist nicht beschränkt.

Eine eingebundene Prozedur kann wiederum @I-Anweisungen enthalten. Die maximale Verschachtelungstiefe beträgt dabei 16 Ebenen. Sofern mehrmals dieselbe Prozedur angegeben ist, wird die Prozedur tatsächlich nur einmal eingebunden, da wiederholte Einbindungen ignoriert werden.

@C+ oder @C-

In den Prozeduren kann mit zwei unterschiedlichen 'Styles' gearbeitet werden. Dabei gibt es zum einen den Pascal-Style, bei dem die Syntax ähnlich der Sprache Pascal ist. Dieser Stil entspricht dem der 4.0 Prozeduren und ist standardmäßig eingestellt. Mit @C+ wird der C-Style aktiviert, bei dem die Syntax eher der Sprache C ähnelt.

Im Wesentlichen wirkt sich das auf den Standardsprachumfang aus. Zum Beispiel wird aus dem if-Konstrukt

IF <Bedingung>THEN <Befehl>

im PASCAL-Style,

if (<Bedingung>) <Befehl>

im C-Style. Weitere Unterschiede sind im Kapitel Style-Unterschiede beschrieben.

Innerhalb des Prozedurtextes können Bereiche in Abhängigkeit von Bedingungen übersetzt werden. Dazu können verschiedene Compileranweisungen verwendet werden.

@define und @undef

Mit der Anweisung @define kann eine Metavariablen definiert werden. Die Anzahl der Metavariablen ist nicht beschränkt. Die Variable kann mit @undef wieder entfernt werden.

Syntax:

Kontakt

```
@define <Name>@undef <Name>
```

Im weiteren Verlauf der Prozedur kann mit den Anweisungen @ifdef und @ifndef die Existenz der Metavariablen abgefragt werden. Innerhalb der Prozedur kann eine Variable wieder entfernt werden. Auf diese Weise kann zum Beispiel ein Debugmodus für nur eine Funktion aktiviert werden.

@ifdef oder @ifndef ... @else ... @endif

Mit diesen Anweisungen kann die Existenz bzw. die Nicht-Existenz einer Metavariablen geprüft werden.

Syntax:

```
@ifdef <Name>           @ifndef <Name>  <Prozedurtext>      <Prozedurtext>[@else    [...] ]
```

Ist die Metavariable mit @define definiert, wird mit @ifdef der Prozedurtext nach der Anweisung übersetzt. Gibt es eine @else-Anweisung, wird der Prozedurtext im ELSE-Fall übersetzt, wenn die Metavariable nicht definiert ist.

Im Falle der Compileranweisung @ifndef wird der Prozedurtext nach der Anweisung nur dann übersetzt, wenn die angegebene Metavariable nicht definiert ist.

Der Prozedurtext kann beliebigen Text beinhalten, insbesondere weitere Compiler-Anweisungen. @ifdef und @ifndef können in bis zu 30 Ebenen geschachtelt werden.

Beispiele

Im folgenden Beispiel wird gezeigt, wie Debug-Anweisungen auch nach der Fehlersuche im Prozedurtext stehen bleiben können, um die relevanten Informationen auch noch bei einer späteren Fehlersuche ausgeben zu können.

```
@A+@C+@define debugmodemain{@ifdef debugmode  DbgConnect('*', false, false);@endif  ...}sub Funktion()
```

Es können auch innerhalb einer Include-Prozedur mehrere Varianten implementiert werden. In den Prozeduren müssen lediglich vor dem Include die entsprechenden Meta-Variablen definiert werden.

```
@A+@C+@ifdef ungenau#define{  mPi : 3.14}@endif@ifdef genau5#define{  mPi : 3.14159}@endif@ifdef genau7#define{  mPi : 3.141592653589793}@endif
```

In folgender Prozedur wird die Definition eingebunden:

```
@A+@C+@define genau5@I:PI.def...
```

Style-Unterschiede

Unterschiede zwischen dem PASCAL- und dem C-Style

Der Style einer Prozedur wird durch die Compileranweisung @C+ bzw. @C- festgelegt. Er kann innerhalb einer Prozedur nicht verändert werden.

Die Unterschiede in der Syntax betreffen folgende Befehle:

- **Anweisungsblock**

- ♦ **PASCAL-Style**

BEGIN...END

(es kann auch {...} verwendet werden)

- ♦ **C-Style**

{...}

- **Verzweigung**

- ♦ **PASCAL-Style**

IF <Bedingung>
THEN <Anweisung>
ELSE <Anweisung>

- ♦ **C-Style**

if (<Bedingung>)
 <Anweisung>
else
 <Anweisung>

- **Fallabfrage**

- ♦ **PASCAL-Style**

CASE <Ausdruck>
 OF <Ausdruck> :
 <Anweisung>
 OTHERWISE :
 <Anweisung>
 END

- ♦ **C-Style**

switch (<Ausdruck>)
{
 case <Ausdruck> :
 <Anweisung>
 default :
 <Anweisung>
}

- **Zählschleife**

- ◆ **PASCAL-Style**

```
FOR <Anweisung>
LOOP <Anweisung>
WHILE <Bedingung>
DO <Anweisung>
```

oder

```
FOR <Anweisung>
LOOP <Anweisung>
UNTIL <Bedingung>
DO <Anweisung>
```

- ◆ **C-Style**

```
for <Anweisung>
loop <Anweisung>
while (<Bedingung>)
    <Anweisung>
```

oder

```
for <Anweisung>
loop <Anweisung>
until (<Bedingung>)
    <Anweisung>
```

- **Schleife mit Eintrittsbedingung**

- ◆ **PASCAL-Style**

```
WHILE <Bedingung>
DO <Anweisung>
```

- ◆ **C-Style**

```
while (<Bedingung>)
    <Anweisung>
```

- **Schleife mit Austrittsbedingung**

- ◆ **PASCAL-Style**

```
REPEAT <Anweisung>
UNTIL <Bedingung>
```

- ◆ **C-Style**

```
do <Anweisung>
while (<Bedingung>)
```



Alle Beispiele innerhalb der Dokumentation sind im C-Style geschrieben.

Kommentare

Verwendung von Kommentaren

Siehe [Editor](#)

In den Prozeduren sind zwei verschiedene Arten von Kommentaren möglich:

Kommentarblock: /* ... */

Alle in diesen Zeichenblock eingeschlossenen Zeichen werden bei der Übersetzung ignoriert. Eine Verschachtelung von Kommentaren ist zulässig.

Kommentarzeile: //

Durch diese Zeichenfolge werden alle nachfolgenden Zeichen bis zum Zeilenende ignoriert. Diese Form des Kommentars ist daher besonders für einzeilige Kommentare geeignet, da kein explizites Kommentarenden notwendig ist. Die Kommentarzeile ist unabhängig vom Kommentarblock und kann somit beliebig in Kommentarblöcke eingeschlossen werden.

Shell-Kommando: <....>

In einer Kommentarzeile kann ein spezieller Kommentar eingegeben werden. Dieser Kommentar führt beim  + Doppelklick ein Shell-Kommando aus. Dadurch ist es z. B. möglich, direkt auf eine Web-Adresse zu verweisen oder direkt eine E-Mail zu versenden.

Beispiel:

```
// *****/* / *
```

Das in <....> eingefasste Shell-Kommando wird ausgeführt, wenn ein Doppelklick auf das Kommando erfolgt. Pro Zeile ist ein Kommando erlaubt, die Zeile muss außerdem eine Kommentarzeile sein, die durch // eingeleitet wird.

Schlüsselwörter

Schlüsselwörter

Siehe [Befehlsgruppen](#),
[Befehlsliste](#)

- [declare](#)
- [define](#)
- [global](#)
- [handle](#)
- [local](#)
- [NULL](#)



define
Deklaration von Definitionen

Siehe [Verwandte Befehle](#)

Syntax:

```
define{ [[ <name> [[ <argument>, ]] : <Zeichenfolge> ]]}
```

Mittels einer Definition wird einem Namen eine frei definierbare Zeichenfolge zugewiesen. Innerhalb der Prozedur kann dann anstelle der jeweiligen Zeichenfolge dieser Name eingesetzt werden. Damit können zum Einen symbolische Konstanten benutzt werden, bei denen einem konstanten Wert ein symbolischer Name zugeordnet wird, zum Anderen Makros benutzt werden, bei denen einem Namen mit zugehörigen Argumenten ein entsprechender Ausdruck oder Befehlsfolge zugeordnet wird.

Der Name ist die Bezeichnung der Definition, mit der später auf sie zugegriffen wird (Namensvergabe unter [global](#)). Nach dem Doppelpunkt folgt der Definitionstext bis zum Ende der Zeile. Eine mehrzeilige Definition ist daher nicht möglich.

Beispiel:

```
define{ mMaxPositions           : 64   mMaxElements          : MaxPositions * 2 - 1   mCRL
```

In den ersten beiden Definitionen werden MaxPositions und MaxElements als symbolische Konstanten definiert. Bei den nachfolgenden Definitionen werden zusätzlich Argumente angegeben, für die dann später die entsprechenden Werte eingesetzt werden. Bis zu 24 Argumente pro Definition sind zulässig, wobei der Name eines Arguments frei wählbar ist, solange er nicht mit bereits deklarierten Namen konfliktiert. Kommentare innerhalb des Definitionstextes werden ignoriert.

Bei Definitionen handelt es sich immer um einen reinen Textersatz, das heißt, dass bei Verwendung einer Definition diese innerhalb der Zeile durch den Definitionstext (inklusive der Parameter) ersetzt wird. Daher können Definitionen für fast jeden Zweck benutzt werden, allerdings finden keinerlei syntaktische Prüfungen durch den Compiler bei der Deklaration einer Definition statt. So kann beispielsweise in einer Definition mit Variablen gearbeitet werden, die an dieser Stelle noch gar nicht deklariert wurden. Bei der Verwendung eines der obigen Beispiele passiert also folgendes:

```
aus if mRecReadLock(1, 2)
wird if (RecRead(1, 2, _RecLock) = _rOk)
```

Es ist grundsätzlich darauf zu achten, dass Anzahl, Art und Umfang von Definitionen ein noch überschaubares Maß nicht übersteigen und Definitionen nur in sinnvollen Fällen eingesetzt werden, da sonst der positive Nutzen ins Gegenteil verkehrt wird; Prozeduren werden bei unvernünftigem Einsatz von Definitionen wartungsunfreundlich, schlecht überschaubar und schwierig zu testen.

Innerhalb einer Prozedur können beliebig viele Definitionen als auch Definitionssektionen deklariert werden. Daher ist es empfehlenswert, häufig benutzte Definitionen in einer Prozedur zusammenzufassen, die dann von anderen Prozeduren aus per [@I](#)-Anweisung eingebunden wird.

Kontakt

Bei geschachtelten Definitionen (der Definitionstext einer Definition enthält andere Definitionen) ist darauf zu achten, dass bei der Verarbeitung eines Definitionsaufrufs eine oder mehrere Ersetzungen in der jeweiligen Zeile vorgenommen werden. Dabei verlängert sich die Zeile entsprechend. Die Länge einer Zeile kann dabei maximal 2000 Zeichen betragen.

Compiler-Makros:

Im Compiler sind folgende Makros vordefiniert:

<u>PROC</u>	Aktuelle Prozedur als Zeichenkette
<u>FUNC</u>	Aktuelle Funktion als Zeichenkette
<u>PROCFUNC</u>	Aktuelle Prozedur und Funktion als Zeichenkette
<u>LINE</u>	Aktuelle Zeile als Zahl



declare

Forward-Deklaration von sub-Funktionen

Verwandte

Siehe [Befehle](#),

sub

Mit Hilfe von declare lassen sich sub-Funktionen deklarieren und verwenden, ohne dass von diesen bereits Code existiert.

```
declare [<procedure>:]<function>(<varname>:<vartype> [<varname>:<vartype>])[:<vartype>]
```

Die Angaben entsprechen somit exakt dem Kopf der jeweiligen Subfunktion. Bei der Verwendung von declare müssen bei der zugehörigen Funktion genau die gleichen Argumente bezüglich Namen, Typ und Optionen verwendet werden.

declare kann hauptsächlich für folgende Zwecke verwendet werden:

- **Deklaration einer lokalen Subfunktion**

Dies ermöglicht die Verwendung einer Funktion im Prozedurtext vor ihrer Definition.

Beispiel:

```
declare fnc2 ( aID : int; ) : int; sub fnc1 : int; { return(fnc2(215)); // dies ist ohne d
```

Die deklarierte Funktion muss dabei im Prozedurtext an späterer Stelle definiert sein, da die Prozedur sonst nicht übersetzt werden kann.

- **Deklaration einer externen Subfunktion**

Dies ermöglicht die Verwendung einer Funktion aus einer anderen Prozedur, ohne dass diese zum Zeitpunkt der Übersetzung vorhanden sein muss.

Die Deklaration kann sich dabei auch in einem anderen Prozedurtext befinden, der mittels Include-Anweisung (@I:<Prozedurname>) eingebunden wird.

Beispiel:

```
declare Lib:CheckDate ( aLimit : date; ) : logic; main{ if (Lib:CheckDate(tStart)) ... }
```

Bei der Ausführung der Prozedur muss allerdings die deklarierte Funktion in compilierter Form vorhanden sein, sonst erfolgt ein Laufzeitfehler.



global

Deklaration eines globalen Datenbereichs

Verwandte

Befehle,

local,

Siehe VarAllocate(),

VarInstance(),

VarFree()

Die Definition eines globalen Datenbereiches wird mit dem Schlüsselwort **global** eingeleitet. Jeder globale Datenbereich muss einen eindeutigen Namen besitzen, der nach dem Schlüsselwort angegeben wird. Der Name kann unabhängig von den im Datenbereich deklarierten Variablen definiert werden.

Pro Prozedur können bis zu 65.000 globale und lokale Datenbereiche verwendet werden. Da der Speicher eines globalen Datenbereichs erst zur Laufzeit mit dem Befehl VarAllocate() angelegt wird, sollten für unterschiedliche Bereiche einer Applikation gegebenenfalls mehrere globale Datenbereiche definiert werden. Die Anzahl von Variablen in einem Datenbereich ist nicht beschränkt, allerdings kann ein einzelner Datenbereich nicht mehr als 4 MB Speicher belegen.

Von einem globalen Datenbereich können mehrere 'Instanzen' angelegt werden, indem VarAllocate() mehrfach aufgerufen wird. Der Wechsel zwischen den Speicherbereichen erfolgt dann mit VarInstance().

Der globale Datenbereich muss in allen Prozeduren definiert sein, in denen er benutzt werden soll. Die Definition des Datenbereiches muss somit in einer Prozedur erfolgen, die in alle Prozeduren mit der Compileranweisung @I eingebunden wird. Der Datenbereich behält solange seine Gültigkeit, bis er mit dem Befehl VarFree() gelöscht wird.

Eine Variable wird deklariert durch <Variablename> : <Variablentyp>;. Nach dem Typ der Variablen muss ein Semikolon stehen. Es können auch mehrere Variable gleichzeitig mit demselben Typ deklariert werden, indem einfach mehrere Variablennamen durch Komma getrennt angegeben werden.

Beispiel:

```
gString : alpha;gWert    : float;gDate    : date;gTime    : time;
```

Folgende Typen können verwendet werden:

- alpha
- byte
- word
- int
- float
- date
- time
- logic

Kontakt

Die Typen byte und word sind im wesentlichen nur bei Arrays von Bedeutung, da mit diesen Typen im Gegensatz zu int Speicherplatz gespart werden kann, sofern die zu erwartenden Werte im jeweiligen Wertebereich von byte bzw. word liegen.

Arrays

Eine Variable kann als Array deklariert werden. In diesem Fall folgt nach dem Typ die Anzahl der Elemente in eckigen Klammern. Sofern sich die Anzahl der benötigten Elemente erst zur Laufzeit ergibt, kann ein Array auch als dynamisch deklariert werden, indem keine Anzahl angegeben wird.

Beispiel:

```
global Common{ gWerte      : int[1000];           // 1000 Elemente von int gNamensTab : alpha(100)
```

Ein einzelnes Array kann nicht größer als 4 MB deklariert werden. Ein dynamisches Array muss zur Laufzeit per Prozedurbefehl mit der gewünschten Anzahl von Elementen im Speicher angelegt werden (VarAllocate()). Die einzelnen Elemente eines Array werden später durch <namen>[<index>] angesprochen (zum Beispiel bSortTab[22]). Dabei hat das erste Element immer die Nummer 1, die Nummer des letzten Elements entspricht der Anzahl von Elementen im Array.

Benutzung konstanter Ausdrücke

Bei der Alphalänge und den Arrayelementen kann anstatt einem einzelnen Wert ein konstanter Ausdruck benutzt werden. Konstante Ausdrücke müssen vom Typ int sein und dürfen keine Felder, Variablen oder Funktionen enthalten. Symbolische Konstanten und konstante Werte sind zulässig.

Beispiel:

```
gSortTab      : byte[iSortSize * 4 + 100];
```

Dabei ist iSortSize eine symbolische Konstante. Zulässig sind alle ganzzahligen Operationen und die Verwendung von Klammerebenen.

Gültigkeitsbereich

Der Gültigkeitsbereich eines globalen Datenbereichs lässt sich am sinnvollsten mit einem kleinen Beispiel verdeutlichen:

Prozedur A

global x{ gTest : int;}...gTest # 122; global x{ gTest : int;}...gTest # 45;
Hierbei wird in den beiden Prozeduren A und B der globale Datenbereich x deklariert. Intern wird aber ein globaler Datenbereich durch <Prozedurname>:<Datename> angesprochen. Dadurch liegt die Variable 'gTest' in der Prozedur A im Datenbereich 'A:x', in der Prozedur B dagegen im Datenbereich 'B:x'. Somit ist die Variable 'gTest' in A und B nicht dieselbe. Um in beiden Prozeduren mit 'gTest' auch tatsächlich dieselbe Variable zu benutzen, ist folgende Konstruktion notwendig:

Prozedur B

Prozedur A

Prozedur B

Prozedur C

Kontakt

```
@I:B...gTest # 45; global x{ gTest : int;} @I:B...gTest # 122;
```

Jetzt wird der Datenbereich x in der Prozedur B deklariert, wodurch die Variable 'gTest' im Datenbereich 'B:x' liegt. Durch das Einbinden von B in den Prozeduren A und C greifen beide Prozeduren jetzt auf dieselbe Variable 'gTest' zu.

Dieses Verfahren garantiert, dass von gemeinsam benutzten globalen Datenbereich nur eine einzige Deklaration existiert, wodurch nicht identische Mehrfachdeklarationen vermieden werden. Globale Datenbereiche müssen daher immer dann in gesonderten Prozeduren deklariert werden, wenn mehrere Prozeduren denselben Datenbereich verwenden wollen. Dies ist der Regelfall, jedoch macht es auch durchaus Sinn, einen Datenbereich direkt in einer Prozedur zu deklarieren, anstatt die Deklaration einzubinden. Dadurch wird der Datenbereich praktisch 'privat', dass heißt keine andere Prozedur hat Zugriff auf diesen Datenbereich.

Damit sichergestellt ist, dass bei Änderungen einer Deklaration keine ungültigen Prozeduren entstehen (zum Beispiel wenn Variablen aus dem Datenbereich entfernt wurden), existieren zwei zusätzliche Mechanismen:

- **Laufzeitkontrolle**

Beim Starten einer Prozedur wird automatisch überprüft, ob die von der Prozedur benutzten Datenbereiche bereits im Speicher vorhanden sind. Ist dies der Fall, wird überprüft, ob der vorhandene Datenbereich in Bezug auf Größe und Anzahl der Variablen mit der in der Prozedur benutzten Deklaration übereinstimmt. Ist dies nicht der Fall, so entsteht ein Laufzeitfehler. Mit dieser Überprüfung können die meisten Inkompatibilitäten zur Laufzeit entdeckt werden.

- **Übersetzen abhängiger Prozeduren**

Zur völligen Sicherheit können nach Änderung einer Prozedur alle diejenigen Prozeduren übersetzt werden, die die geänderte Prozedur einbinden. Eine entsprechende Funktion ist bei der Übersetzung von Prozeduren vorhanden.

Lebensdauer

Die Lebensdauer eines globalen Datenbereichs beginnt mit der Speicheranforderung per Prozedurbefehl ([VarAllocate\(\)](#)). Bei Datenbereichen, die ständig verfügbar sein müssen empfiehlt es sich, diese im Ereignis [EvtInit](#) des Frames anzulegen oder in der Eigenschaft [DbVar](#) eines Applikations- oder MDI-Fensters anzugeben. Die Lebensdauer des Bereiches endet beim Löschen des Datenbereiches mit [VarFree\(\)](#) (zum Beispiel im Ereignis [EvtTerm](#)) oder beim Entladen des Applikations- oder MDI-Frames. Spätestens beim Entladen der Datenstruktur werden ebenfalls die globalen Datenbereiche freigegeben.

handle
Deskriptor
Siehe [Datentypen](#),
[ComCall\(\)](#)

Der Begriff handle hat im CONZEPT 16-Editor die folgenden Bedeutungen:

- [Datentyp](#)
- [Typumwandlung bei COM-Aufrufen](#)

Datentyp

Der Datentyp handle ist ein spezieller Datentyp für Deskriptoren. Variablen vom Typ handle sind vollständig kompatibel zum Datentyp [int](#). Alle entsprechenden Spezifikationen zum Datentyp handle können der Dokumentation unter [int](#) entnommen werden.

Durch die Verwendung des eigenen Datentyps für Deskriptoren ist es möglich, eigene [sub](#)-Funktionen, die einen Deskriptor als ersten Parameter erwarten, mit dem Pfeiloperator aufzurufen. Aufrufe in Form von

Function(tHandle, tA, tB)

können also auch im folgenden Format angegeben werden:

tHandle->Function(tA, tB)

Typumwandlung bei COM-Aufrufen

Mit dem Schlüsselwort handle wird bei der Übergabe von Parametern an Methoden oder Eigenschaften von COM-Objekten ein Deskriptor gekennzeichnet.

Beispiel:

```
tCellStart # tComWorksheet->cphCell(1, 1);tCellEnd    # tComWorkSheet->cphCell(10, 2);tComRange #
```

In diesem Beispiel wird ein Bereich markiert, der anschließend mit Werten gefüllt oder ausgelesen werden kann. Wird in diesem Fall handle weggelassen, würde der Deskriptor falsch interpretiert werden.