



# Session V-VFX2

## Produktiv mit VFX Builder Einsatz in der Praxis

*Venelina Jordanova, Uwe Habermann*  
V&U Ltd. - [www.VandU.eu](http://www.VandU.eu)

---

### **Einführung**

Viele Wizards, Builders und andere Produktivitätswerkzeuge unterstützen den VFX Entwickler bei der Arbeit. Nicht nur bei der Erstellung neuer Projekte, Formulare, Grids oder Auswahllisten bietet VFX leistungsfähige Builder, auch die Erstellung von Cursoradaptern, Einstellungen für die Produktaktivierung und Lokalisierung bis hin zur Erstellung von Hilfeprojekten sind Tools vorhanden. In dieser Session wird gezeigt, was alles durch den Einsatz der Builder von VFX und ohne Programmierung möglich ist.

---

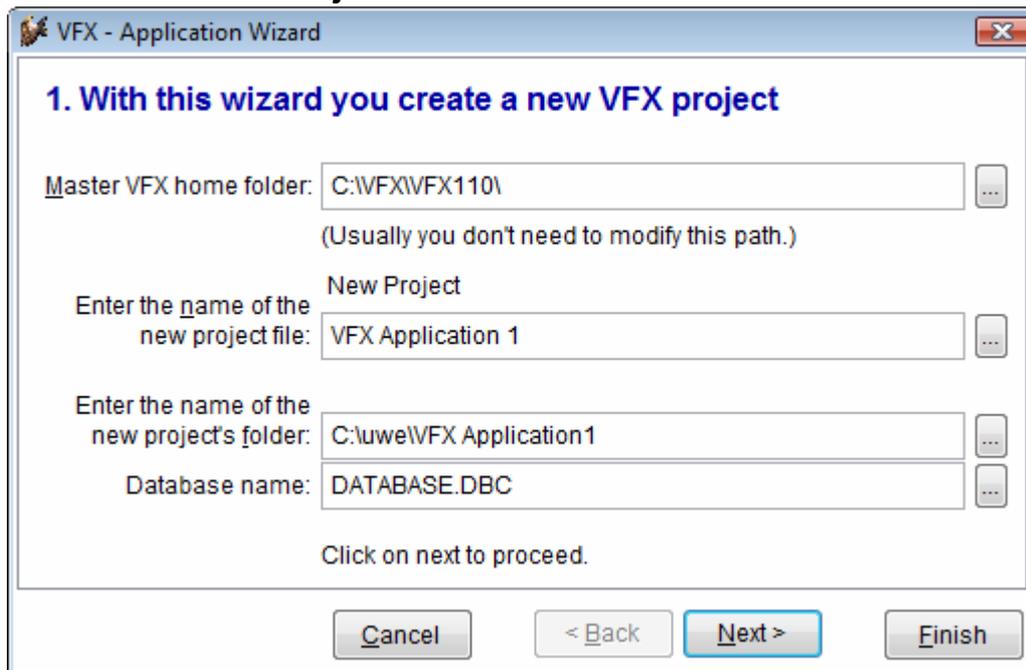
# Projekt

## VFX – Application Wizard

Eine neue Anwendung wird mit dem VFX – Application Wizard erstellt. Der VFX – Application Wizard kann aus dem VFX Menü und aus der VFX – Task Pane gestartet werden.

Die Einstellungen, die im VFX – Application Wizard gemacht werden, werden zur Verwendung in späteren Projekten gespeichert.

### 1. Neues VFX Projekt



Geben Sie die folgenden Daten ein, bevor Sie eine neue Anwendung generieren lassen:

**Master VFX home folder:** Tragen Sie hier den VFX-Ordner ein, in dem sich Ihre VFX-Installation befindet. Normalerweise ist der vorgegebene Wert des Assistenten richtig und Sie brauchen keine Änderung zu machen.

**Enter the name of the new project file:** Geben Sie hier den Namen für Ihre neue Projektdatei ein. Fügen Sie keinen Pfad und keine Namenserweiterung hinzu. Geben Sie nur den Namen des neuen Projekts ein.

**Enter the name of the new project's folder:** Geben Sie den Ordner für Ihr neues Projekt ein. Wenn der Ordner noch nicht existiert, so wird er von dem VFX – Application Wizard erstellt. Der Standardpfad, in dem neue Projekte angelegt werden, ist „Eigene Dateien\VFX Projects“. Wenn ein anderer Pfad zum Erstellen eines Projektes gewählt wird, werden auch alle folgenden Projekte unter diesem Pfad gespeichert. Standardmäßig wird ein Projektordner mit dem Namen *VFX Application* gefolgt von einer fortlaufenden Nummer erstellt.

**Database Name:** Geben Sie den Namen Ihres Datenbank-Containers an (DBC). Geben Sie nur den Namen des Datenbank-Containers ohne Pfad und ohne Namenserweiterung ein. Wenn Ihre Anwendung auf eine Remote Datenquelle zugreifen soll und ausschließlich CursorAdapter für den Datenzugriff verwenden soll, können Sie dieses Feld leer lassen.

## 2. About

**2. About**

Enter the title of your application and the data which will be displayed in the About... dialog of your application.

Application title: Visual Extend Application

Version: Version #

Copyright: Copyright © dFPUG c/o ISYS GmbH

Click on next to proceed.

Cancel < Back **Next >** Finish

**Application title:** Geben Sie die Überschrift für das Hauptfenster Ihrer Anwendung an. Wenn nicht mit Lokalisierung zur Laufzeit gearbeitet wird, wird diese Überschrift als Konstante CAP\_APPLICATION\_TITLE in der Include-Datei USERTXT.H gespeichert. Bei Lokalisierung zur Laufzeit wird diese Überschrift aus der Tabelle Vfxmsg.dbf gelesen. Die Tabelle Vfxmsg.dbf ist bei Lokalisierung zur Laufzeit in das Projekt eingeschlossen.

**Version:** Geben Sie die Versionsnummer für den Infodialog Ihrer Anwendung ein. Wenn nicht mit Lokalisierung zur Laufzeit gearbeitet wird, wird die Nummer in der Konstante CAP\_LBLVERSION in der Include-Datei USERTXT.H gespeichert.

**Copyright:** Geben Sie Ihre Copyright-Information für den Infodialog Ihrer Anwendung ein. Wenn nicht mit Lokalisierung zur Laufzeit gearbeitet wird, wird diese Information in der Konstante CAP\_LBLCOPYRIGHT-INFORMATION in der Include-Datei USERTXT.H gespeichert.

## 3. Optionen

**3. Options**

The following options are general settings for your application.

You can modify these settings later using the VFX Application Builder

Ask to save when close:       Toolbar style: CAppNavBar

Enable autoedit mode:       Language: German

Enter on the grid means edit:       AutoFit grids on first load:

Enable hooks:       Enable product activation:

Use DBCX compliant products:       Use "FirstInstall.txt" file:

Copy Loader.exe to new project:       Advanced

Click on next to proceed.

Cancel < Back **Next >** Finish

**Ask to save when close:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *nAsktoSave* des Anwendungsobjekts auf 1. Diese Eigenschaft bestimmt das Verhalten von VFX wenn der Benutzer ein Formular schließt, nachdem er Änderungen am aktuellen Datensatz gemacht hat.

**Enable autoedit mode.** Die Auswahl dieser Option setzt den Wert der Eigenschaft *nAutoEditMode* des Anwendungsobjekts auf 1. Das bedeutet, dass der Benutzer jederzeit mit der Bearbeitung der Daten beginnen kann, ohne vorher in den Bearbeitungsmodus wechseln zu müssen.

**Enter on the grid means edit:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *nEnterisEditinGrid* des Anwendungsobjekts auf 1. Das bedeutet, dass durch Drücken der Enter-Taste auf dem Grid einer Suchseite in den Bearbeitungsmodus gewechselt wird.

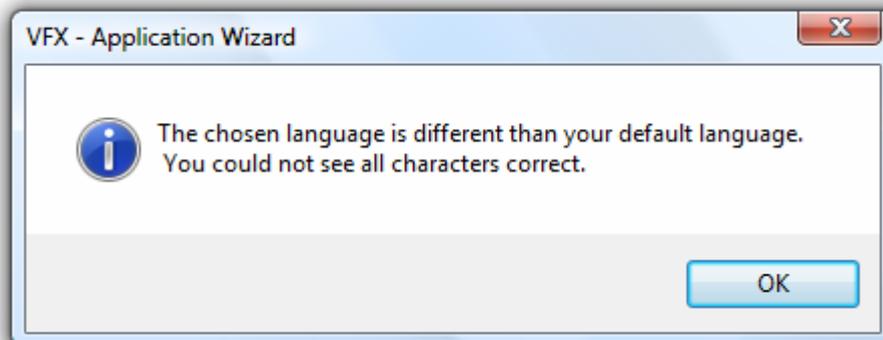
**Enable hooks:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *nEnableHook* des Anwendungsobjekts auf 1. Das bedeutet, dass die Hooks aktiviert werden.

**Use DBCX compliant products:** Wenn der Stonefield Database Toolkit mit der zu erstellenden VFX-Anwendung eingesetzt werden soll, muss diese Option markiert werden.

**Copy Loader.exe to new project:** Zur Aktualisierung der Anwendung beim Kunden über das Internet wird die Datei Loader.exe benötigt. Wenn Sie das Loader-Projekt für Ihre Anwendung individuell anpassen möchten, markieren Sie diese Option.

**Toolbar style:** Wählen Sie hier die Symbolleistenklasse, die Sie verwenden wollen. *CAppNavBar* enthält Schaltflächen zur Bewegung des Datensatzzeigers und andere Schaltflächen zur Bearbeitung in der Standard-Symbolleiste. *CAppToolBar* enthält keine Schaltflächen zur Bewegung des Datensatzzeigers und zur Bearbeitung.

**Language:** Wählen Sie die gewünschte Sprache für Ihr neues Projekt. Bei der Auswahl einer Sprache für die generierte Anwendung prüft VFX die aktuellen Unicode-Einstellungen des Betriebssystems. Wenn die Zeichen der gewählten Sprache mit den aktuellen Einstellungen nicht angezeigt werden können, erscheint eine Warnung.



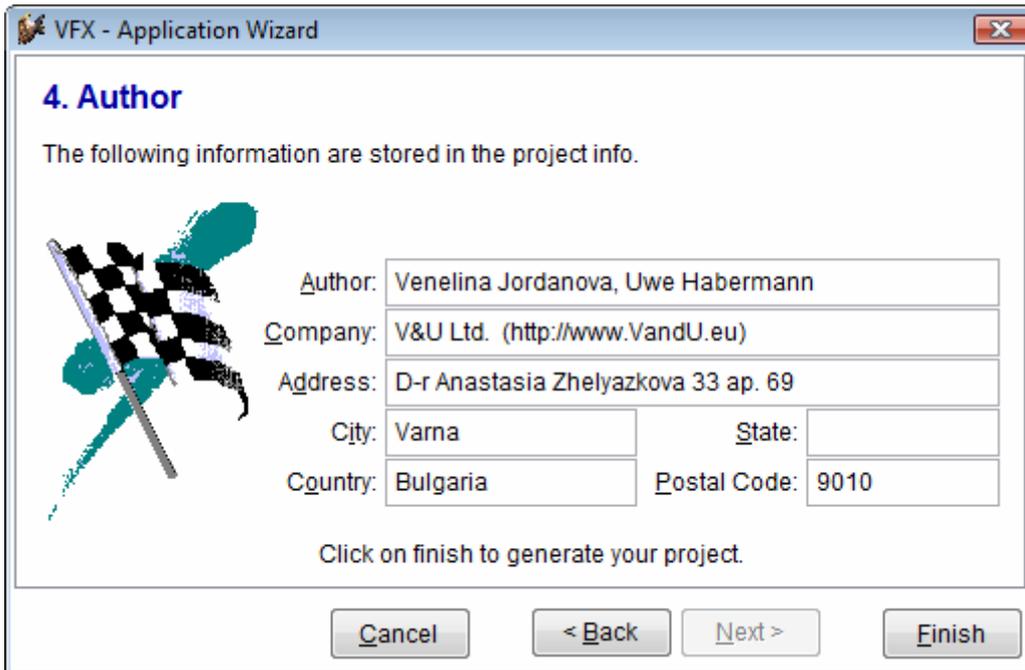
**AutoFit grids on first load:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *nUseAutofit* des Anwendungsobjekts auf 1. Das bedeutet, dass bei der ersten Initialisierung von Grids das AutoFit-Ereignis aufgerufen wird.

**Enable product activation:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *IUseActivation* des Anwendungsobjekts auf .T. Das bedeutet, dass die Anwendung eine Produktaktivierung erfordert.

**Use „Firstinstall.txt“ file:** Die Auswahl dieser Option setzt den Wert der Eigenschaft *lActivationType* des Anwendungsobjekts auf .T. Das bedeutet, dass die Produktaktivierung die Datei *Firstinstall.txt* erfordert. Der Schutz Ihrer Anwendung wird dadurch weiter verbessert.

**Advanced:** Über diese Schaltfläche wird der VFX – Application Builder gestartet, der eine Vielzahl weiterer Einstellmöglichkeiten des Anwendungsobjekts bietet.

## 4. Autor



**4. Autor**

The following information are stored in the project info.



Author: Venelina Jordanova, Uwe Habermann  
Company: V&U Ltd. (<http://www.VandU.eu>)  
Address: D-r Anastasia Zhelyazkova 33 ap. 69  
City: Varna State:   
Country: Bulgaria Postal Code: 9010

Click on finish to generate your project.

Diese Informationen werden in der Projektdatei gespeichert.

Wenn Sie *Finish* auswählen, wird der VFX – Application Wizard ein neues Projekt entsprechend den von Ihnen eingegebenen Parametern erstellen. Dabei wird die Musteranwendung aus der VFX-Installation in den neuen Projektordner kopiert. Die Include-Dateien werden entsprechend der ausgewählten Sprache generiert. Anschließend wird das gesamte Projekt kompiliert, damit die in den Include-Dateien enthaltenen Konstanten zur Anwendung kommen. Eine abschließende Meldung zeigt an, dass Ihre neue Anwendung erfolgreich vorbereitet wurde.

---

**ANMERKUNG:** Da Sie sicher sofort mit der Arbeit an Ihrem neuen Projekt beginnen wollen, hat der VFX-Anwendungs-Assistent bereits automatisch den Standardordner auf den Startordner des neuen Projektes gesetzt. Um die Anwendung aus dem Projekt-Manager zu starten, wählen Sie das Hauptprogramm *VFXMAIN.PRG* und wählen Sie „ausführen“.

---

## VFX – Application Builder

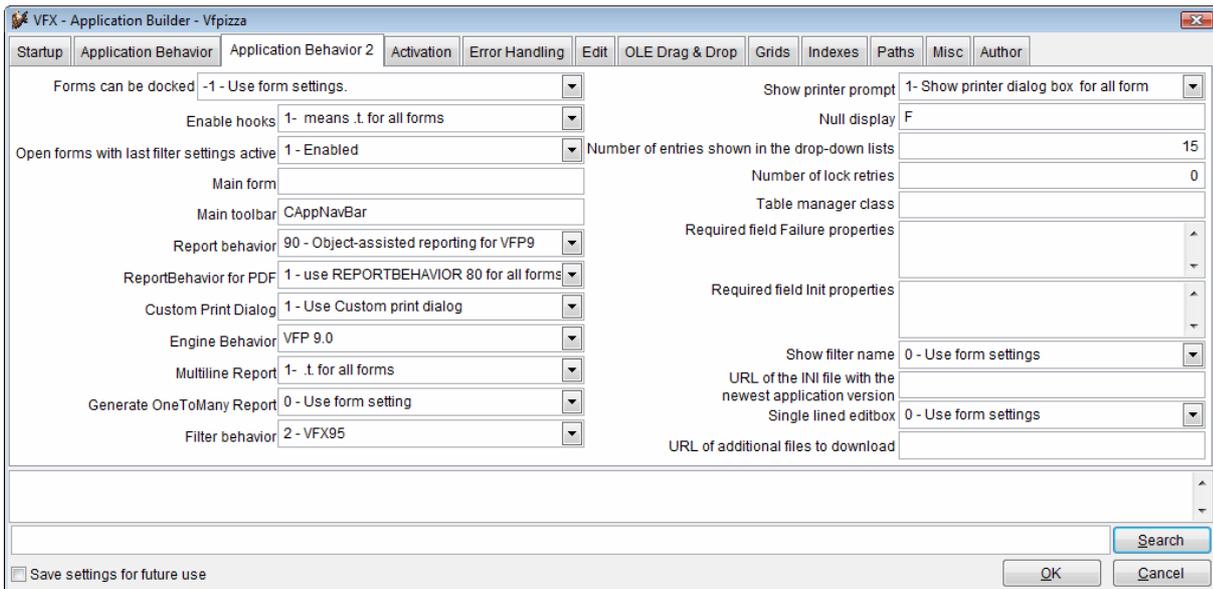
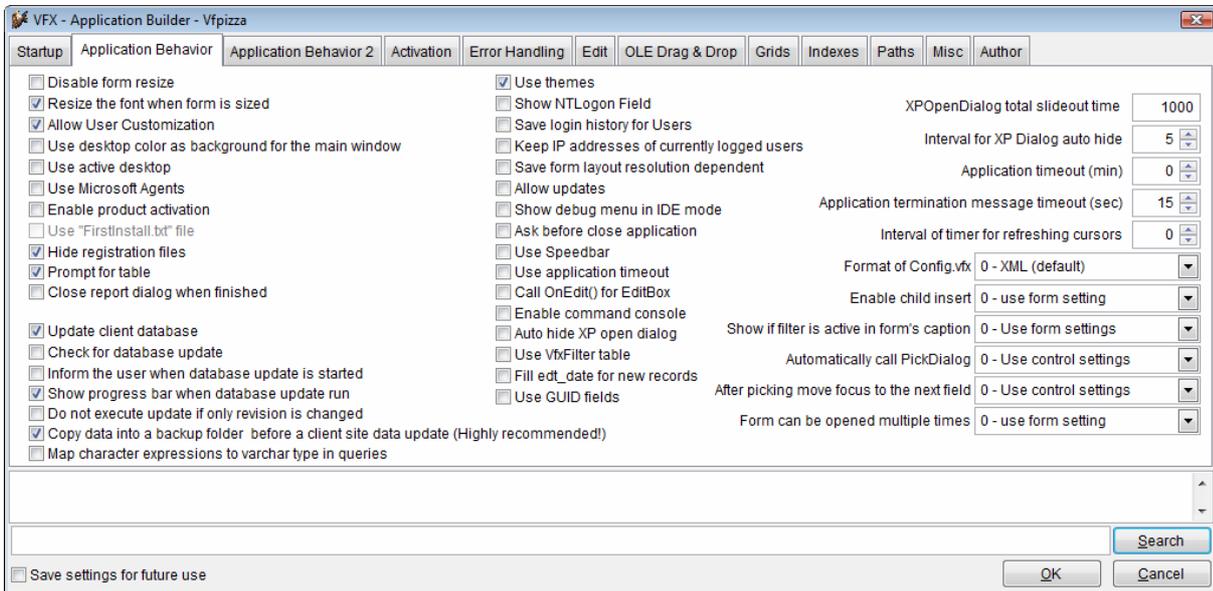
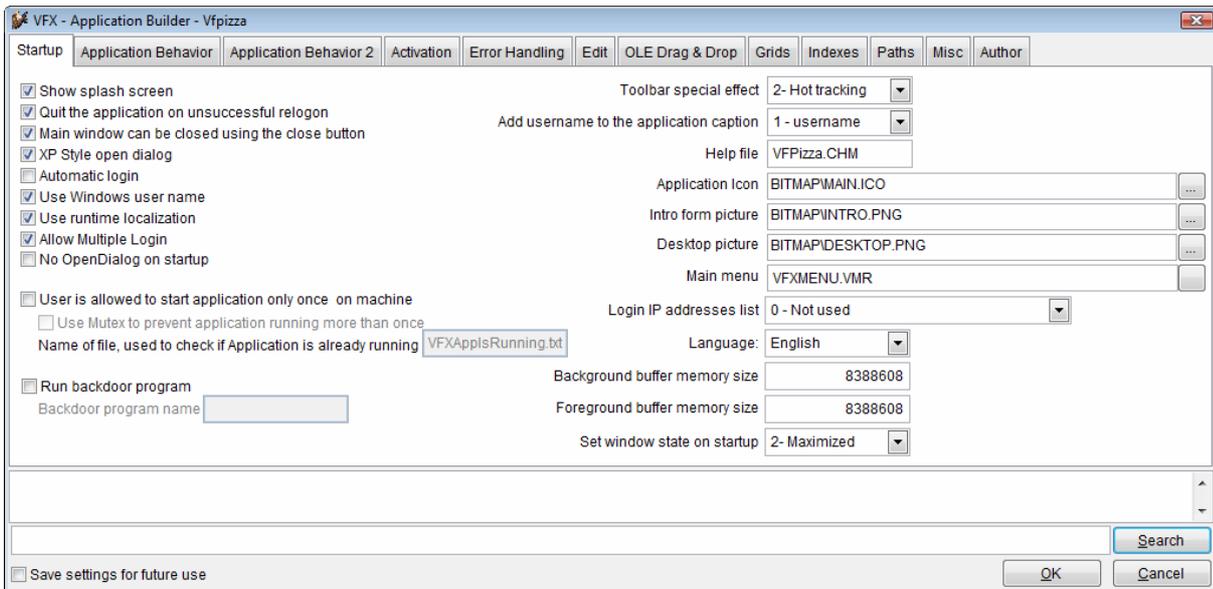
Dieser Dialog kann jederzeit aus dem VFX Menü über den Menüpunkt *Project, Application Builder* aufgerufen werden, um Einstellungen des Anwendungsobjekts zu ändern.

Wenn ein Steuerelement im VFX – Application Builder den Fokus erhält, wird in einer Editbox im unteren Bereich des Builders ein Hilfetext mit einer Klärung zu der Einstellung angezeigt. Zu jedem Steuerelement wird in einem Tooltip der Name der Klasse und Eigenschaft angezeigt, die mit dem entsprechenden Steuerelement eingestellt wird.

---

**ANMERKUNG:** Die mit dem VFX – Application Builder gemachten Einstellungen können für das nächste neue Projekt übernommen werden, wenn das Kontrollkästchen „Save settings for future use“ markiert wird.

---



VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Number of changes accepted, when using hardware parameters tolerance

Hardware parameters file

Encrypt password for hardware parameters

Store activation data to

Activation key validity in days

Activation key type

Time limited activation key

Start date of activation keys

Method to send registration number to the developer

Server name for HTTP registration

Object name for HTTP registration

Filename for registration number

Email to send registration number to

Name for the Register form

Web service

Web Service name

Web Service link

Web Service Register method name

Save settings for future use

Search

OK Cancel

VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Error processing

Log error details

Web Service ErrorReport method

Name of application

Company

Save settings for future use

Search

OK Cancel

VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Show century in date fields

Century for rollover

Year for rollover

Date format

Null is valid value

Always ask prior any save operation

Hide controls when table is empty

Autoedit mode

Name of the field in any table to be automatically used to store:

the "user" who inserted this record

the "user" who last modified this record

the "date" when this record has been inserted

the "last edit" date

the "time" when this record has been inserted

the "last edit" time

the "date" when this record has been modified

the "time" when this record has been modified

check sum value for the record

the deletion status of the record

the readonly status of the record

Specifies the source table name for Auto Complete data

Don't hide list page while editing

Allow save empty records

Save without transaction

Use memo form

Move the focus to the next object on Enter key for cCheckBox

Refresh all pages before the form valid event on Save

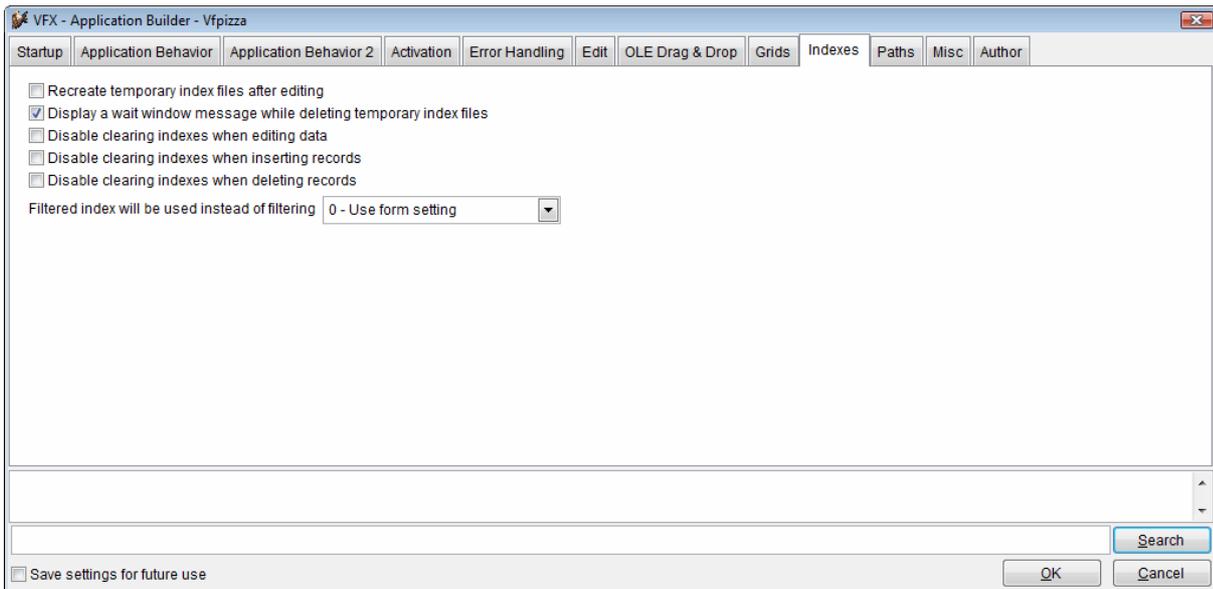
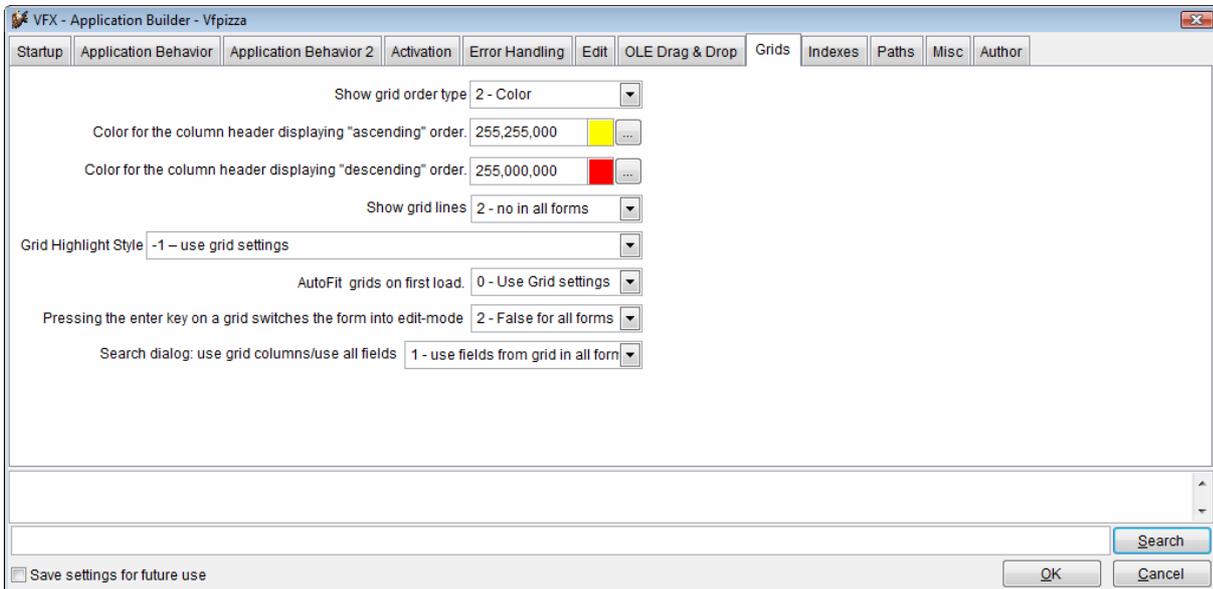
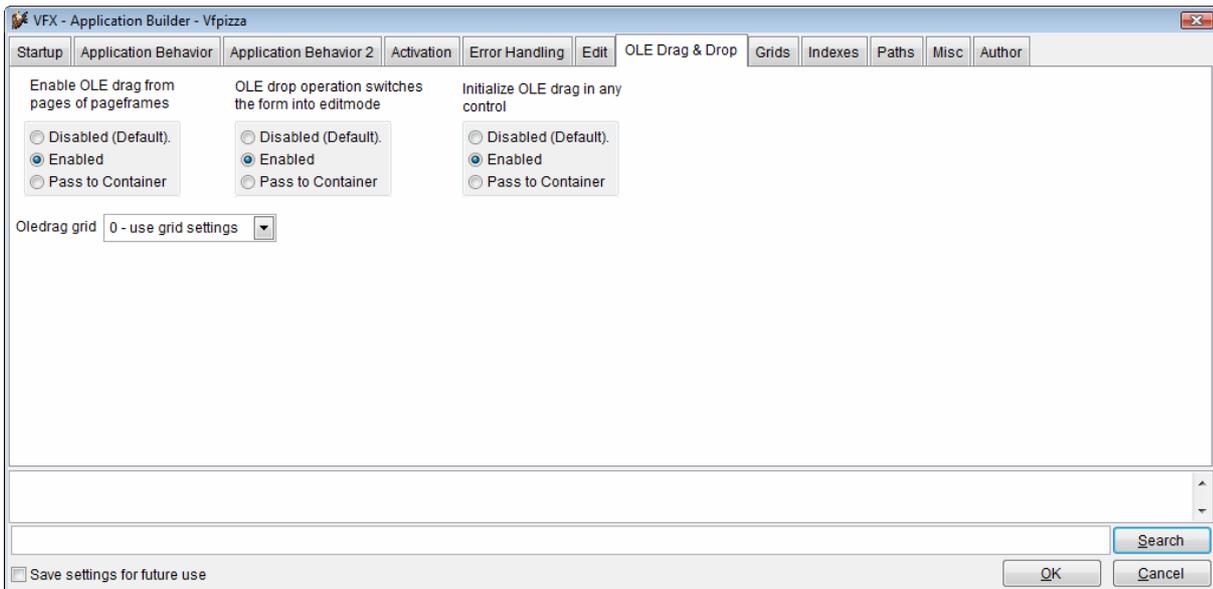
Allow to delete child data even if the deletion of parent records is not allowed

User is allowed to send BCC E-Mail

Save settings for future use

Search

OK Cancel



VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Database folder

Database name

Metadata folder

Name of metadata table

Default import folder

Current export folder

Path to the external report files (\*.frx)

Save Export files folder per user

Save settings for future use

Search OK Cancel

VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Name of Postscript printer to be installed when necessary

Always install PS printer

Name of Fax printer driver to be used when sending fax reports

URL used when checking for internet connection existence

Password to be used for encrypting config.vfx file

Support URL

Support e-mail

List separator chars

Security tables list

Install ClickYes

Save settings for future use

Search OK Cancel

VFX - Application Builder - Vfpizza

Startup Application Behavior Application Behavior 2 Activation Error Handling Edit OLE Drag & Drop Grids Indexes Paths Misc Author

Author:

Company:

Address:

City:

State:

PostalCode:

Country:

Company web site URL:

Feedback email address:

Save settings for future use

Search OK Cancel

Die meisten Einstellmöglichkeiten des VFX – Application Builder beziehen sich auf Eigenschaften des Anwendungsobjekts. Die Klasse *CApplication* ist die Klasse des Anwendungsobjekts. Die Eigenschaften und Methoden des Anwendungsobjekts stehen global in der gesamten Anwendung zur Verfügung.

Die Klasse *CApplication* wird in *Vfxmain.prg* programmatisch von der visuellen Klasse *CFoxappl* aus der Klassenbibliothek *Appl.vcx* abgeleitet. In der Klasse *CFoxappl* macht der VFX – Application Builder die Einstellungen. Für die Steuerung der Anwendung wichtige Eigenschaften des Anwendungsobjekts sollen hier erläutert werden.

*cAscOrderRGB* – RGB-Wert einer Farbe, die verwendet wird um eine aufsteigende Sortierung in einer Grid-Spalte in der Überschrift anzuzeigen. Der Standardwert ist "*RGB(255,255,0)*". Dies entspricht der Farbe gelb.

*cDataDir* – Der Pfad in dem sich die Datenbank befindet. In der Entwicklungsumgebung ist dies der Pfad *Data* und sollte nicht verändert werden. Standardmäßig wird dieser Pfad aus der Konstanten *datapath\_loc* aus der Datei *Userdef.h* gelesen. Weisen Sie dieser Eigenschaft einen Leerstring zu, wenn Sie Multi-Client-Database-Eigenschaft von VFX nutzen möchten. Zur Laufzeit der Anwendung beim Kunden kann die Datenbank dann in einem beliebigen Ordner verwendet werden.

*cDateFormat* – Das Datumsformat, das standardmäßig in der Anwendung verwendet wird. Der Wert dieser Eigenschaft wird als Parameter dem Befehl SET DATE übergeben. Der Wert dieser Eigenschaft wird normalerweise in der Methode *setlangid* des Anwendungsobjekts entsprechend der eingestellten Sprache zugewiesen.

*cDescOrderRGB* – RGB-Wert einer Farbe, die verwendet wird um eine absteigende Sortierung in einer Grid-Spalte in der Überschrift anzuzeigen. Der Standardwert ist "*RGB(255,0,0)*". Dies entspricht der Farbe rot.

*cEdt\_Date* – Der Name eines Feldes in einer beliebigen Tabelle. Wenn ein Datensatz gespeichert wird und in der betreffenden Tabelle ein Feld mit diesem Namen gefunden wird, werden hier das Datum und ggf. die Uhrzeit der Bearbeitung gespeichert. Der Typ des Feldes kann *Date* oder *Datetime* sein. Der Standardwert ist ein Feld mit dem Namen *edt\_date*.

*cEdt\_Usr* – Der Name eines Feldes in einer beliebigen Tabelle. Wenn ein Datensatz gespeichert wird und in der betreffenden Tabelle ein Feld mit diesem Namen gefunden wird, wird hier der Name des Benutzers gespeichert, der den Datensatz verändert hat. Das Feld muss vom Typ *Zeichen* sein. Der Standardwert ist ein Feld mit dem Namen *edt\_usr*.

*cExcludeFiles* – Hier kann eine durch Komma separierte Liste von Dateinamen eingegeben werden. Die hier aufgeführten Dateien erscheinen nicht im Dialog Datenbankwartung und sind von der Datenbankwartung ausgeschlossen. Der Standardwert ist "*DBCXREG.DBF;CDBKMETA.DBF;SDTMETA.DBF;SDTUSER.DBF;COREMETA.DBF*".

*cHelpFile* – Der Name der Hilfedatei, die beim drücken der Taste F1 geöffnet werden soll. Die Namensweiterung (*chm* oder *hlp*) muss mit angegeben werden. Der Standardwert ist der Name des Projekts mit der Namensweiterung *chm*.

*cIns\_Date* – Der Name eines Feldes in einer beliebigen Tabelle. Wenn ein neuer Datensatz gespeichert wird und in der betreffenden Tabelle ein Feld mit diesem Namen gefunden wird, werden hier das Datum und ggf. die Uhrzeit der Neuanlage gespeichert. Der Typ des Feldes kann *Date* oder *Datetime* sein. Der Standardwert ist ein Feld mit dem Namen *ins\_date*.

*cIns\_Usr* – Der Name eines Feldes in einer beliebigen Tabelle. Wenn ein neuer Datensatz gespeichert wird und in der betreffenden Tabelle ein Feld mit diesem Namen gefunden wird, wird hier der Name des Benutzers gespeichert, der den Datensatz neu angelegt hat. Das Feld muss vom Typ *Zeichen* sein. Der Standardwert ist ein Feld mit dem Namen *ins\_usr*.

*cIntroBitmap* – Der Name einer Bilddatei, die als Splashscreen angezeigt werden soll. Es sind alle von VFP unterstützten Grafikformate zulässig, also zum Beispiel *bmp*, *jpg*, *gif* oder *png*. Der Standardwert ist *Bitmap\Intro.png* und wird aus der Include-Datei *Userdef.h* gelesen.

*cIntroForm* – Der Name einer Formulareklasse, die den Splashscreen anzeigen soll. Eine Änderung dieses Wertes ist nur erforderlich, wenn ein Splashscreen mit besonderen Eigenschaften verwendet werden soll. Der Standardwert ist *CSplashDialog*.

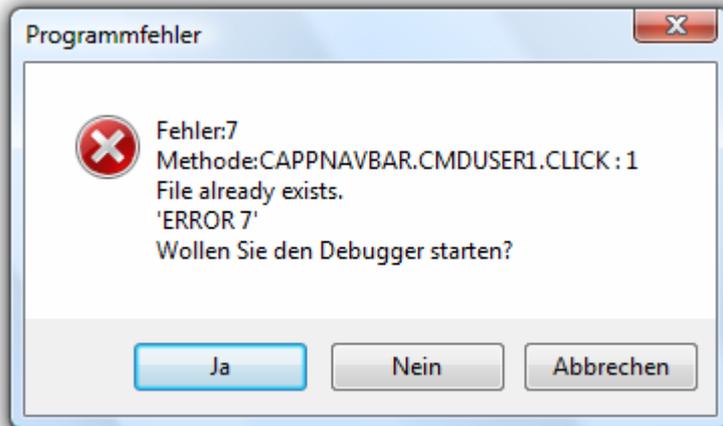
*cLoginForm* – Der Name einer Formularedatei, die den Anmeldedialog enthält. Eine Änderung dieser Eigenschaft ist nur erforderlich, wenn die Benutzerverwaltung von VFX nicht verwendet soll und ein eigenes Verfahren zur Benutzerverwaltung zum Einsatz kommt. Der Standardwert ist *Vfxlogin.scx*.

- cMainDatabase* – Der Name der Datenbank. Der Wert wird aus der Konstanten *database\_loc* aus der Datei *Userdef.h* gelesen. Der Standardwert wurde mit dem VFX – Application Wizard beim Erstellen des Projekts festgelegt.
- cMainForm* – Der Name eines Formulars, das beim Start der Anwendung nach der Benutzeranmeldung angezeigt werden soll. Der Standardwert ist eine leere Zeichenkette.
- cMainIcon* – Das Symbol der Anwendung. Standardmäßig wird dieses Symbol in allen Formularen verwendet. Der Standardwert ist *Bitmap\Main.ico* und wird aus der Konstanten *mainicon\_loc* aus der Include-Datei *Userdef.h* gelesen
- cMainTitle* – Der Name der Anwendung. Dieser Name erscheint in der Titelzeile der Anwendung. Der Name der Anwendung kann auch beim Befehl *CREATEOBJECT(„application“, <Name der Anwendung>)* als zweiter Parameter angegeben werden. In diesem Fall wird der Wert dieser Eigenschaft überschrieben.
- cMainToolbar* – Der Name der Standard-Symboleiste. Der Standardwert wurde mit dem VFX – Application Wizard beim Anlegen des Projekts festgelegt. VFX stellt zwei Klassen mit Symboleisten zur Verfügung. Die Klasse *CAppToolbar* enthält keine Schaltflächen zur Bewegung des Datensatzzeigers in Formularen. Die Klasse *CAppNavBar* enthält Schaltflächen zur Bewegung des Datensatzzeigers in Formularen. Bei Verwendung der Multifunktionsleiste bleibt diese Einstellung unberücksichtigt.
- cvfxpath* – In dieser Eigenschaft kann der Name der Tabelle angegeben werden, die die Informationen zu den Pfaden der Datenbanken der Anwendung enthält. Der Standardwert ist *Vfxpath.dbf*.
- FileMnuOffset* – Dies ist die Nummer des Eintrags im Menü „Datei“, das für den ersten Eintrag eines zuletzt verwendeten Formulars verwendet wird. Wenn Sie dem Menü „Datei“ eigene Einträge hinzufügen wollen, muss dieser Wert entsprechend erhöht werden.
- lAllowDeleteChildData* – Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt wird, dürfen Benutzer, die in OneToMany-Formulare keine Datensätze löschen dürfen, trotzdem Child-Datensätze löschen. Wenn dieser Wert auf *falsch* gesetzt wird, dürfen auch keine Child-Datensätze gelöscht werden.
- lAutoLogin* – Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt wird, werden Benutzer, die in der Benutzerverwaltung registriert sind, beim Anwendungsstart ohne Aufforderung zur Eingabe eines Kennworts automatisch angemeldet. Der Standardwert dieser Eigenschaft ist *falsch*.
- lCentury* – Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt ist, wird in allen Datumsfeldern der Anwendung die Jahreszahl 4stellig angezeigt. Der Standardwert ist *falsch*, Jahreszahlen werden 2stellig angezeigt.
- lDisableFormResize* – Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt wird, ist das Ändern der Größe aller Formulare der Anwendung nicht möglich. Der Standardwert ist *falsch*, die Größe von Formularen kann vom Benutzer verändert werden.
- lNoClearIdxOnDelete* – Standardmäßig löscht VFX temporäre Indexdateien, wenn ein Datensatz gelöscht werden soll. Setzen Sie den Wert dieser Eigenschaft auf *wahr*, wenn temporäre Indexdateien in dieser Situation nicht gelöscht werden sollen. Beachten Sie, dass temporäre Indexdateien nicht geöffnet sein dürfen, wenn Transaktionen ausgeführt werden. Der Standardwert ist *falsch*.
- lNoClearIdxOnEdit* – Standardmäßig löscht VFX temporäre Indexdateien, wenn ein Datensatz bearbeitet werden soll. Setzen Sie den Wert dieser Eigenschaft auf *wahr*, wenn temporäre Indexdateien in dieser Situation nicht gelöscht werden sollen. Beachten Sie, dass temporäre Indexdateien nicht geöffnet sein dürfen, wenn Transaktionen ausgeführt werden. Der Standardwert ist *falsch*.
- lNoClearIdxOnInsert* – Standardmäßig löscht VFX temporäre Indexdateien, wenn ein Datensatz neu angelegt werden soll. Setzen Sie den Wert dieser Eigenschaft auf *wahr*, wenn temporäre Indexdateien in dieser Situation nicht gelöscht werden sollen. Beachten Sie, dass temporäre Indexdateien nicht geöffnet sein dürfen, wenn Transaktionen ausgeführt werden. Der Standardwert ist *falsch*.
- lRelogonQuit* – Steuert das Verhalten der Anwendung, wenn ein Benutzer versucht sich während die Anwendung läuft erneut anzumelden und den Vorgang abbricht. Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt wird, wird die Anwendung beendet. Wenn der Wert dieser Eigenschaft auf *falsch* gesetzt wird, bleibt der zuletzt angemeldete Benutzer angemeldet.
- lRemakeIdxAfterClear* – Wenn der Wert dieser Eigenschaft auf *wahr* gesetzt wird, werden temporäre Indexdateien nach dem Abschluss eines Speichervorgangs automatisch wieder angelegt. Vergleichen Sie auch mit den Eigenschaften *lNoClearIdxOnDelete*, *lNoClearIdxOnEdit*, *lNoClearIdxOnInsert*. Der Standardwert dieser Eigenschaft ist *falsch*.
- nAppOnErrorBehavior* – Diese Eigenschaft steuert das Verhalten der Anwendung im Fehlerfall.

0 – Laufzeitfehler werden ignoriert.

1 – Es wird eine Fehlermeldung angezeigt (Standardwert).

Bei einem Laufzeitfehler in der Entwicklungsumgebung hat der Entwickler die Möglichkeit den Debugger zu starten:



Bei einem Laufzeitfehler in der ausführbaren Datei wird ein Laufzeitfehler so angezeigt:



2 – Die Ausführung der Anwendung wird nach Anzeige eines Hinweises beendet.

*ErrorDetailLevel* – Diese Eigenschaft steuert welche Informationen im Fehlerfall in der Tabelle *Vfxlog.dbf* protokolliert werden.

0 – Nur die Fehlermeldung aber keine Information über den Aufrufstapel.

1 – Die Fehlermeldung und Informationen über den Aufrufstapel (Standardwert).

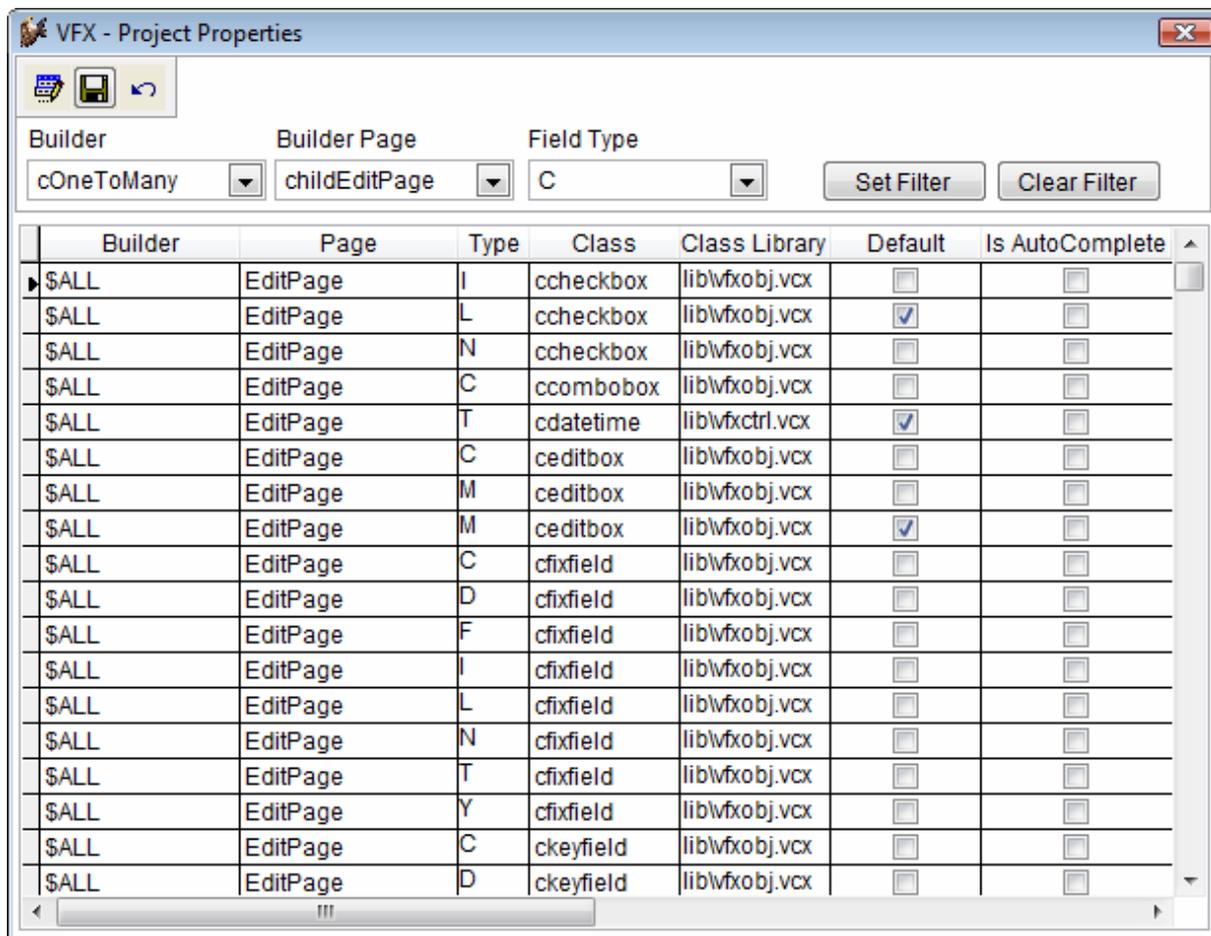
2 – Vollständige, detaillierte Fehlerinformationen.

*PSPrinterToInstall* – Diese Eigenschaft enthält den Namen eines Postscript-Druckertreibers. Dieser Drucker-treiber wird automatisch installiert, wenn noch kein Postscript-Druckertreiber installiert ist und die Anwendung einen Postscript-Druckertreiber braucht um eine PDF-Datei zu erstellen.

- cConnectionCheckURL* – Diese Eigenschaft enthält die Adresse einer Internetseite, die verwendet wird um zu testen, ob eine Internet-Verbindung besteht. Wenn keine Verbindung mit dem Internet besteht und auch keine DFÜ-Netzwerkverbindung eingerichtet ist, wird von VFX ein Eintrag im DFÜ-Netzwerk angelegt. Alle Eigenschaften der DFÜ-Verbindung können vom Entwickler vorgegeben werden. Der Anwender kann bei Bedarf in einem Dialog die Telefonnummer, den Benutzernamen und das Kennwort ändern.
- lUseActivation* – Über diese Eigenschaft wird die Produktaktivierung ein- bzw. ausgeschaltet. Der Standardwert ist *falsch*, die Produktaktivierung wird nicht verwendet.
- lActivationType* – Wenn diese Eigenschaft auf *.T.* gesetzt wird, überprüft die Klasse *CVFXActivate* ob die Datei *FirstInstall.txt* existiert, wenn die Anwendung gestartet wird. Der Standardwert ist *.F.*, es wird nicht auf das Vorhandensein der Datei *FirstInstall.txt* geprüft.
- cConfigPassword* – Kennwort für die Verschlüsselung der Datei *Config.vfx*. Dieses Kennwort wird aus Sicherheitsgründen benötigt. Die Verbindungsinformationen zur Datenquelle, wie Benutzername und Kennwort, sind so auch für versierte Anwender nicht im Klartext einsehbar.
- cFaxPrinterName* – Der Name des Fax-Druckertreibers, der zur Versendung von Berichten als Fax verwendet werden soll. Wenn dieser Wert leer ist, versucht VFX einen Druckertreiber mit „Fax“ im Namen zu finden. Bevorzugt werden die Druckertreiber Winfax und Fritz!fax verwendet.
- cMetadataTableName* – Name der Tabelle mit den Metadaten. Diese Tabelle wird zur Aktualisierung einer SQL Server-Datenbank beim Kunden benötigt. Der Standardwert ist *Datadict*.
- lAllowMultipleLogin* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, dürfen sich Benutzer mehrmals gleichzeitig an der Anwendung anmelden. Der Standardwert ist *.T.*
- lAllowUserCustomization* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, können die Umgebungseinstellungen je Benutzer gespeichert werden. Der Standardwert ist *.T.*
- lInformUserForUpdate* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, wird vor der Aktualisierung der Kundendatenbank eine Meldung angezeigt. Der Standardwert ist *.F.*
- lSaveExportPathPerUser* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, wird der Exportpfad für PDF-, BMP-, HTML- und TIFF-Dateien je Benutzer in der Ressourcentabelle *Vfxres.dbf* gespeichert. Der Standardwert ist *.T.*
- lShowProgressOnUpdate* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, wird während der Aktualisierung der Kundendatenbank eine Fortschrittsanzeige angezeigt. Der Standardwert ist *.T.*
- lUseBCCRecipients* – Wenn diese Eigenschaft auf *.T.* eingestellt ist, wird im Dialog zur Eingabe von E-Mail-adressen eine Textbox zur Eingabe von BCC-Empfängern angezeigt. Der Standardwert ist *.F.* Dieses Feature wird nicht von allen E-Mailprogrammen unterstützt. BCC funktioniert zum Beispiel mit Outlook Express, nicht jedoch mit Outlook.
- nDockable* – Einstellung des Dock-Verhaltens von Formularen. -1 – Die Einstellung des Formulars wird verwendet, 0 – Alle Formulare können nicht gedockt werden, 1 – Alle Formulare unterstützen „Docking“, 2 – Alle Formulare unterstützen „Docking“ sind aber nicht dockbar. In diesem Fall können Formulare nur ineinander gedockt werden. Modale Formulare können grundsätzlich nicht gedockt werden.
- nHighLightStyle* – Mit dieser Eigenschaft kann die Eigenschaft *HighlightStyle* von der Klasse *CGrid* global eingestellt werden.
- nIndexInsteadOfFilter* – Mit dieser Eigenschaft kann eingestellt werden, ob anstelle von Filtern mit gefilterten, temporären Indexdateien gearbeitet werden soll. 0 - Die Einstellung des Formulars wird verwendet, 1 - immer gefilterten, temporären Indexdateien verwenden, 2 – es wird immer mit Filtern gearbeitet.
- nNullValid* – Mit dieser Eigenschaft kann eingestellt werden, ob Eingaben in Auswahlfeldern erforderlich sind. 0 - Die Einstellung des Auswahlfeldes wird verwendet, 1 – eine leere Eingabe ist erlaubt, 2 - eine leere Eingabe ist nicht erlaubt.
- nSearchOnInit* – Mit dieser Eigenschaft kann eingestellt werden, ob beim Start eines Formulars der zuletzt verwendete Filter gesetzt werden soll.

## VFX – Project Properties

In VFX Anwendungen können eigene Ableitungen der VFX Klassen verwendet werden. Im Dialog VFX – Project Properties können die von den VFX Buildern zu verwendenden Klassen für die einzelnen Steuerelement-Typen eingetragen werden. Als Vorgabe stehen hier die Klassen aus der Klassenbibliothek *Vfxobj.vcx*. Der VFX Entwickler kann diese Vorgaben ändern und eigene Klassen, die vorzugsweise in der Klassenbibliothek *Appl.vcx* gespeichert sind, eintragen. Diese Klassen können dann von den VFX Buildern bei der Erstellung neuer Formulare verwendet werden.

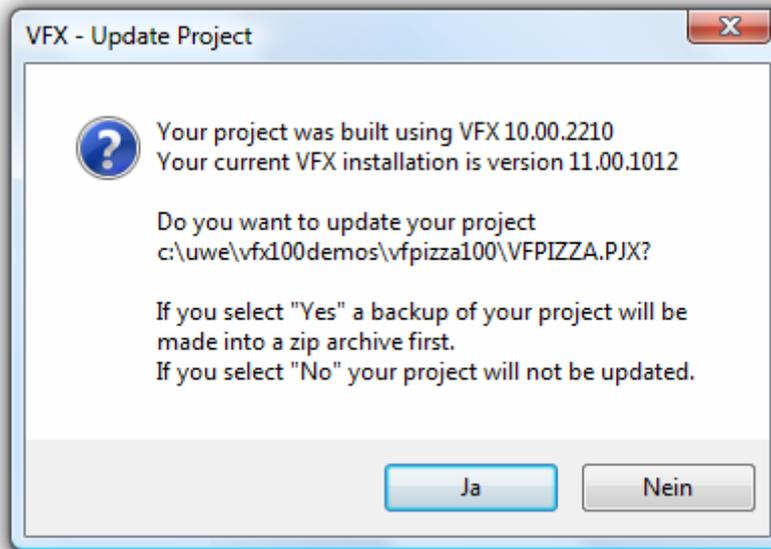


## VFX – Project Backup

Erstellt eine Zip-Datei vom selektierten Projekt mit allen Unterordnern. Der Dateiname besteht aus dem Namen der Projektdatei, gefolgt von einem Unterstrich, gefolgt von einem Zeitstempel im ANSI Format, gefolgt von einem Unterstrich, gefolgt von „VFX“, gefolgt von der Buildnummer von VFX ohne Trennzeichen.

## VFX – Update Project

Projekte, die mit älteren Versionen von VFX oder mit älteren Builds von VFX erstellt wurden, können automatisch auf die neueste Version aktualisiert werden.



Der VFX – Project Update Wizard kann direkt aus dem VFX Menü über den Menüpunkt *Project, Update Project* gestartet werden. Der VFX – Update Project Wizard vergleicht die Version des geöffneten Projekts mit der installierten VFX Version. Wenn das Projekt mit einer älteren VFX-Version erstellt wurde, wird der Entwickler gefragt, ob das Projekt aktualisiert werden soll.

Nach einem Klick auf Ja beginnt der Wizard mit der Arbeit. Zunächst wird zur Sicherheit eine Sicherungskopie des Projekts in einer Zip-Datei angelegt. Die Zip-Datei wird im Projektordner angelegt und erhält den Namen der Projektdatei. Wenn das Archiv bereits existiert oder nicht angelegt werden kann, beendet der Wizard seine Arbeit sofort.

Der VFX – Update Project Wizard aktualisiert die VFX-Klassenbibliotheken, VFX-Berichtsvorlagen und die Datei *Vfxfunc.prg*. Der Tabelle *Vfxmsg.dbf* werden gegebenenfalls neu hinzugekommene Datensätze hinzugefügt. Alle Include-Dateien werden neu erstellt.

Die Struktur der freien VFX-Tabellen wird aktualisiert. Fehlende Felder oder Indexschlüssel werden automatisch ergänzt.

Fehlende Dateien werden dem Projekt automatisch hinzugefügt, wie zum Beispiel neue Bitmap-Dateien oder freie Tabellen.

In der Regel werden die so aktualisierten Projekte sofort mit der neuen VFX Version lauffähig sein. Trotzdem sollte der Entwickler das Projekt sorgfältig prüfen und bei Bedarf manuelle Ergänzungen machen.

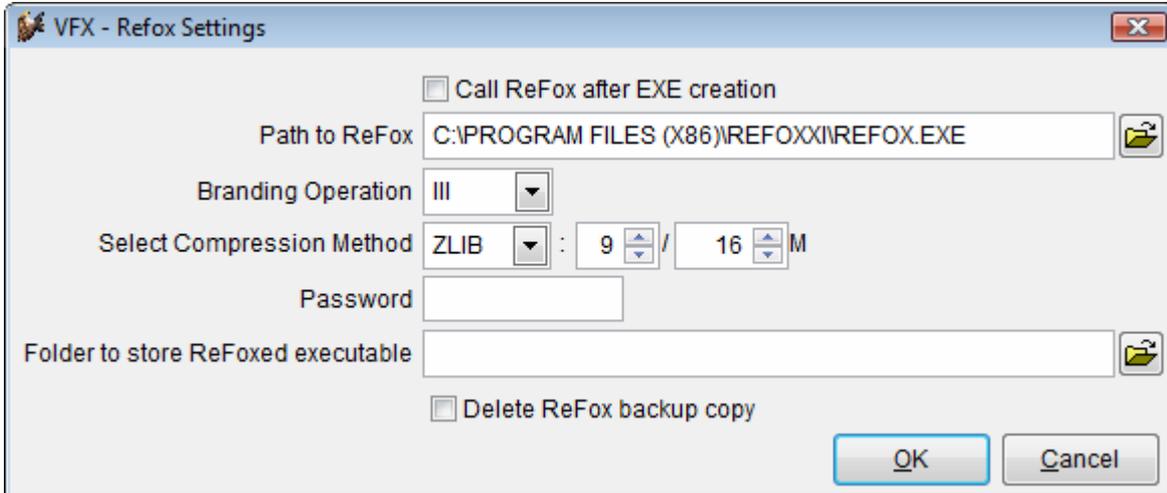
Die meisten Anwendungen werden zum Beispiel ein speziell angepasstes Menü *Vfxmenu.vmx* haben. Der VFX – Update Projekt Wizard kann nicht wissen, welche Menüeinträge der Entwickler vielleicht absichtlich entfernt hat. Der Wizard kann daher keine neuen Menüeinträge hinzufügen. Durch einen Vergleich mit dem Menü aus der VFX Installation können Menüeinträge für neue Funktionen aber schnell ergänzt werden.

Prüfen Sie das neue *Vfxmain.prg* und machen Sie von Hand die für Ihr Projekt erforderlichen Änderungen.

## VFX – Refox Settings

Aus dem VFX Menü kann der Dialog VFX – Refox Settings gestartet werden. Wenn in diesem Dialog das Kontrollkästchen “Call ReFox after EXE creation” markiert ist und ein Pfad zur Refox Installation eingetragen ist, wird nach dem Erstellen einer Exe Datei das Programm Refox gestartet, um die Exe Datei vor Dekompilierung zu schützen. Der Start von Refox wird über den Project Hook gesteuert.

In dem Dialog können außerdem einige Parameter von Refox eingestellt werden.



## VFX – Vista Form Border Fix

Voraussetzung für den Lauf des Vista Form Border Fix ist Visual FoxPro mit SP 2. In VFP 9 SP 2 wurde das Format geändert, in dem Formulare und Formularklassen gespeichert werden. Der Vista Form Border Fix öffnet alle Formulare und Formularklassen und speichert diese ohne Änderung. Nach Anwendung des Vista Form Border Fix werden alle Formulare und Formularklassen, die mit älteren Versionen von VFP erstellt wurden, auf Windows Vista richtig angezeigt. Der Vista Form Border Fix muss nur einmal je Projekt ausgeführt werden.

## VFX – Project Toolbox

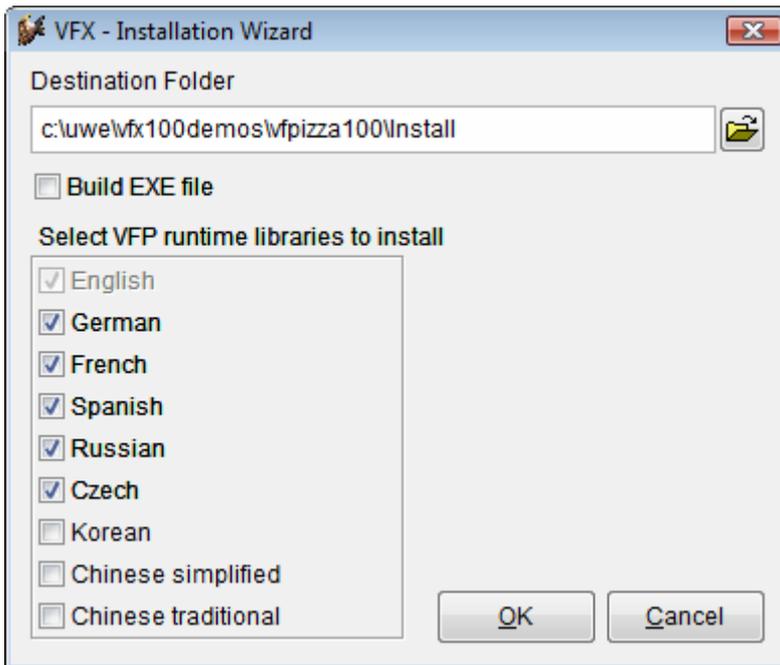
VFX unterstützt die Verwendung der VFP Toolbox für Entwickler. Wenn ein Projekt geöffnet wird, können die zu diesem Projekt gehörenden Klassen in die Toolbox geladen werden.

## VFX – Installation Wizard

Der VFX – Installation Wizard erstellt im Projektordner aus den aktuellen Quellen eine Exe Datei, wenn keine Exe Datei vorhanden ist. Anschließend werden alle Dateien, die zur Ausführung beim Kunden benötigt werden, in den Zielordner kopiert. Die Standardeinstellung ist der Ordner Install unterhalb des Projektordners.

Für die Ausführung beim Kunden werden die Exe Datei, die Laufzeitumgebung von VFP, die VFX.fll sowie einige weitere Dateien, insbesondere ActiveX Steuerlemente benötigt. Alle diese Dateien sowie eine leere Datenbank werden in den Ordner Install kopiert. Die Dateien im Ordner Install können als Vorlage für ein Installationsprogramm verwendet werden oder auch direkt in einen Ordner auf einem Kunden-PC kopiert und von dort ohne Installation ausgeführt werden.

Für die Laufzeitumgebung von VFP können die gewünschten Sprachversionen ausgewählt werden.

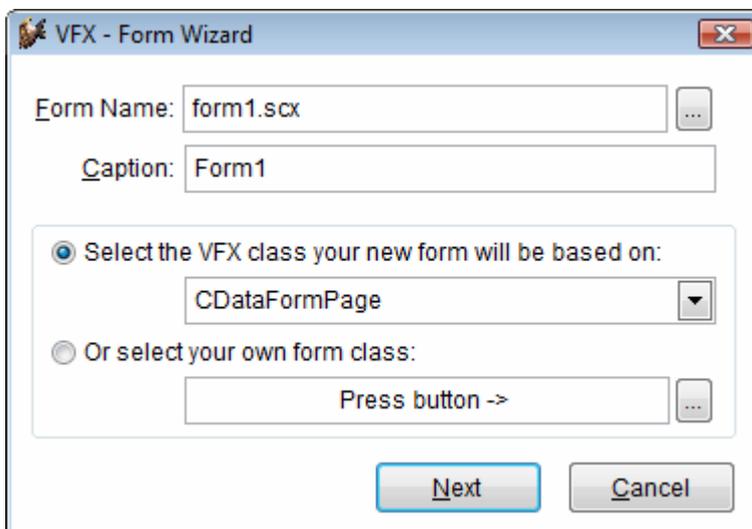


---

## Formulare

### VFX – Form Wizard

Der VFX – Form Wizard dient zum Anlegen neuer Formulare, die auf einer der VFX Formularklassen basieren. Wenn eigene Formularklassen vorhanden sind, die auf einer der VFX Formularklassen basieren, können diese ebenfalls ausgewählt werden.



Nach einem Klick auf Next erscheint der VFX – DataEnvironment Builder.

## VFX – DataEnvironment Builder

Mit dem VFX – DataEnvironment Builder können der Datenumgebung Tabellen, Ansichten oder bestehende CursorAdapter-Klassen hinzugefügt werden oder auch neue CursorAdapter-Klassen erstellt werden. Es können Indexschlüssel für CursorAdapter erstellt werden und es können Beziehungen zwischen Cursor-Objekten eingerichtet werden.

Auf der Seite *Aliases* können Cursor-Objekte hinzugefügt oder erstellt werden.

The screenshot shows the 'Aliases' tab in the VFX - Data Environment Builder. The 'Initial Selected Alias' is 'caorders'. Below is a table with columns: Name, Cursor Source, Alias, Order, Filter, Parent Alias, Rel Expression, and Where Clause.

Name	Cursor Source	Alias	Order	Filter	Parent Alias	Rel Expression	Where Clause
CaORDERS	ORDERS	caorders					customerid=
CaORDERDE	ORDERDETAILS	caorderdetails	orderid				orderid=?ca
CaCUSTOME	CUSTOMERS	cacustomers	customerid		caorders	customerid	
CaPRODUCT	PRODUCTS	caproducts	productid		caorderdetail	productid	

Below the table is a configuration form for the selected alias:

- Cursor Source: ORDERS
- Alias: caorders
- Order: (dropdown)
- Parent Alias: (dropdown)
- Rel Expression: (dropdown)
- Foreign Key Name: (text)
- Foreign Key Value: (text)
- Filter: (text)
- Where Clause: customerid=?thisform.tcustomerid
- NoData  Send Updates
- Security Table (ST): (text)
- ST User Field Name: (text)
- ST ParentID Field Name: (text)
- Main/Parent table alias: (text)
- Security Join Expression: (text)

Buttons at the bottom: Add, Add CA, New CA, Builder, Remove, Add Methods and Properties (checked), OK, Cancel.

Mit einem Klick auf die Schaltfläche *Add* können bestehende Tabellen oder Ansichten der Datenumgebung hinzugefügt werden. Der VFP Dialog zur Auswahl von Tabellen und Ansichten wird geöffnet. Wenn ein Cursor in der Datenumgebung auf einer Tabelle basiert, kann in der Spalte *Order* ein Index der Tabelle gewählt werden.

Über die Schaltfläche *Add CA* kann ein CursorAdapter, basierend auf einer CursorAdapter-Klasse, hinzugefügt werden. Eine solche CursorAdapter Klasse kann mit dem VFX – CursorAdapter Wizard erstellt werden.

Über die Schaltfläche *New CA* kann ein neues Objekt basierend auf der Klasse *CAppDataAccess* mithilfe des VFP – CursorAdapter Builder erstellt werden.

Wenn der Cursor in der Datenumgebung auf einer Tabelle basiert, kann in der Spalte *Order* eine Sortierfolge aus den existierenden Indexschlüsseln ausgewählt werden. Wenn der Cursor auf einem CursorAdapter basiert, kann ein Indexausdruck aus einer Liste der für diesen CursorAdapter definierten Indexausdrücke ausgewählt werden. Die Indexschlüssel werden zur Laufzeit erstellt, nachdem der CursorAdapter mit dem Ereignis *Cursor-Fill()* mit Daten gefüllt wurde. Indexschlüssel für CursorAdapter können auf der Seite *Indexes* angelegt werden.

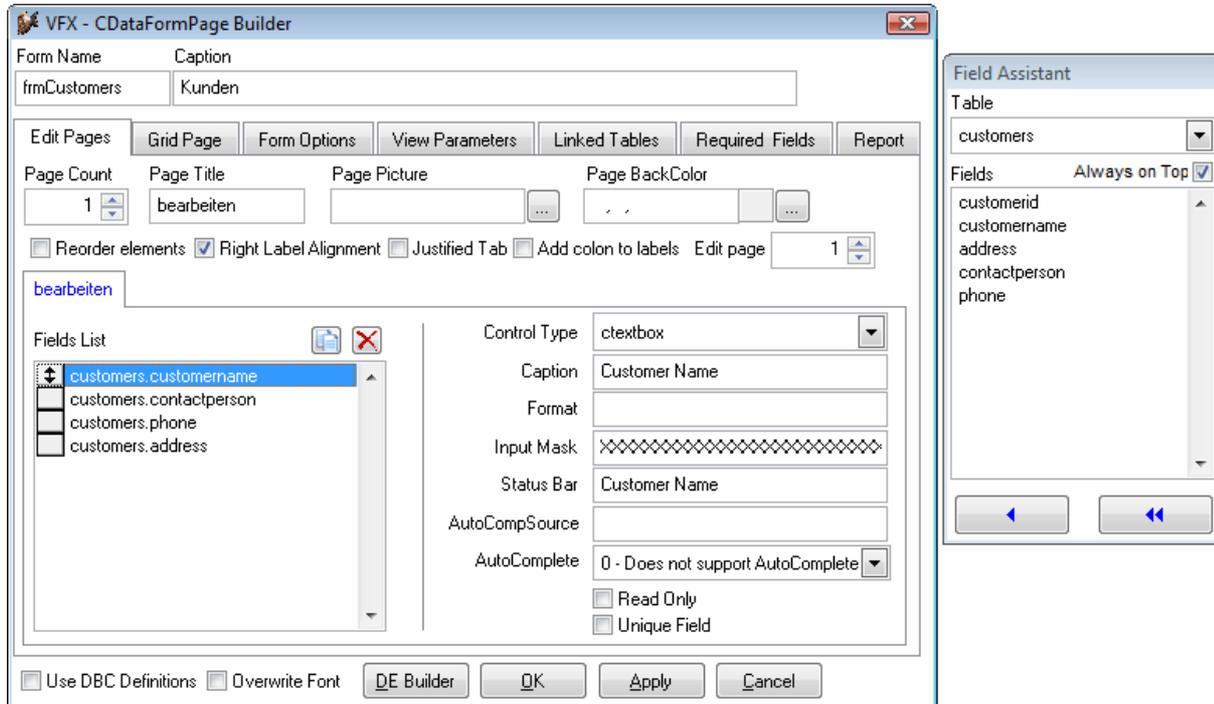
Die Namen und Aliasnamen der Cursor in der Datenumgebung können beliebig geändert werden.



## VFX – CDataFormPage Builder

Mit dem VFX – CDataFormPage Builder werden die für das Formular benötigten Steuerelemente dem Formular hinzugefügt. Für jedes Steuerelement können dabei die zugrunde liegende VFX-Klasse gewählt sowie viele Eigenschaften eingestellt werden.

Alle VFX – Formular Builder sind voll reentrant. Das heißt, man kann die Builder beliebig oft aufrufen um Einstellungen an einem Formular zu verändern. Es ist auch möglich das Formular von Hand mit VFP zu bearbeiten und anschließend wieder mit dem VFX – Formular Builder zu arbeiten, ohne dass Einstellungen verloren gehen oder überschrieben werden.



Auf der Seite *Edit Pages* werden die Steuerelemente für die Bearbeitungsseiten des Formulars zusammengestellt.

Wenn das Kontrollkästchen *Add colon to labels* markiert wird, wird an alle Labels ein Doppelpunkt angefügt.

**Form Name.** Geben Sie den Namen des neuen Formulars ein. Der VFX – Form Wizard hat bereits einen Standardnamen entsprechend den Namenskonventionen zugewiesen. Der Name beginnt mit *frm*. Selbstverständlich können Sie Ihrem Formular einen beliebigen Namen geben, aber wir empfehlen Ihnen, sich an die allgemeinen Namenskonventionen zu halten.

**Caption.** Geben Sie die Überschrift für Ihr Formular ein. Während Sie die Überschrift eingeben, wird diese bereits in der Überschrift des Formular-Builders angezeigt.

**Page Count.** Geben Sie ein, wie viele Bearbeitungsseiten Sie benötigen. Für einige Formulare wird eine Bearbeitungsseite ausreichend sein. Wenn Sie mehr Felder haben, werden Sie diese auf mehrere Seiten verteilen wollen. In Abhängigkeit von der Anzahl der gewählten Seiten, sehen Sie im Seitenrahmen des Formular-Builders einen Seitenrahmen, der diese Seiten anzeigt. Wenn Sie zwei Bearbeitungsseiten eingeben, sehen Sie zwei Seiten auf dem Seitenrahmen, wenn Sie drei Bearbeitungsseiten eingeben, sehen Sie drei Seiten auf dem Seitenrahmen usw.

**Page Title.** Geben Sie die Überschrift der aktuellen Bearbeitungsseite ein. Wenn Sie die Überschrift für die zweite Seite eingeben wollen, drücken Sie auf die zweite Seite und Sie können die Überschrift auch für diese Seite eingeben. Der VFX-Formular-Builders zeigt während der Eingabe die sich ergebende Überschrift für die einzelnen Seiten an.

**Justified Tab.** Markieren Sie dieses Kontrollkästchen, wenn die Seitenüberschriften justiert sein sollen. Ansonsten haben die Überschriften eine variable Länge und füllen nicht die Breite des Seitenrahmens.

Für jede Bearbeitungsseite stehen die folgenden Optionen zur Verfügung:

**Fields Selected.** Hier sehen Sie alle Felder, die Sie für die aktuelle Bearbeitungsseite ausgewählt haben. Um Felder hinzuzufügen benutzen Sie das *Field Assistant*-Fenster, das in einem eigenen Formular angezeigt wird und alle aus der Datenumgebung zur Verfügung stehenden Felder anzeigt.

**Control Type.** Geben Sie für alle ausgewählten Felder den zu benutzenden Steuerungstyp an.

**Caption.** Überschrift für das ausgewählte Feld. Der Standardwert wird aus dem Datenbank-Container übernommen.

**Format.** Format-Eigenschaft für das selektierte Feld. Der Standardwert wird aus dem Datenbank-Container übernommen.

**Input Mask.** Eingabemasken-Eigenschaft für das selektierte Feld. Der Standardwert wird aus dem Datenbank-Container übernommen.

**Status Bar.** Meldung für die Statuszeile für dieses Feld. Der Standardwert wird aus dem Datenbank-Container aus der Eigenschaft Feldkommentar übernommen.

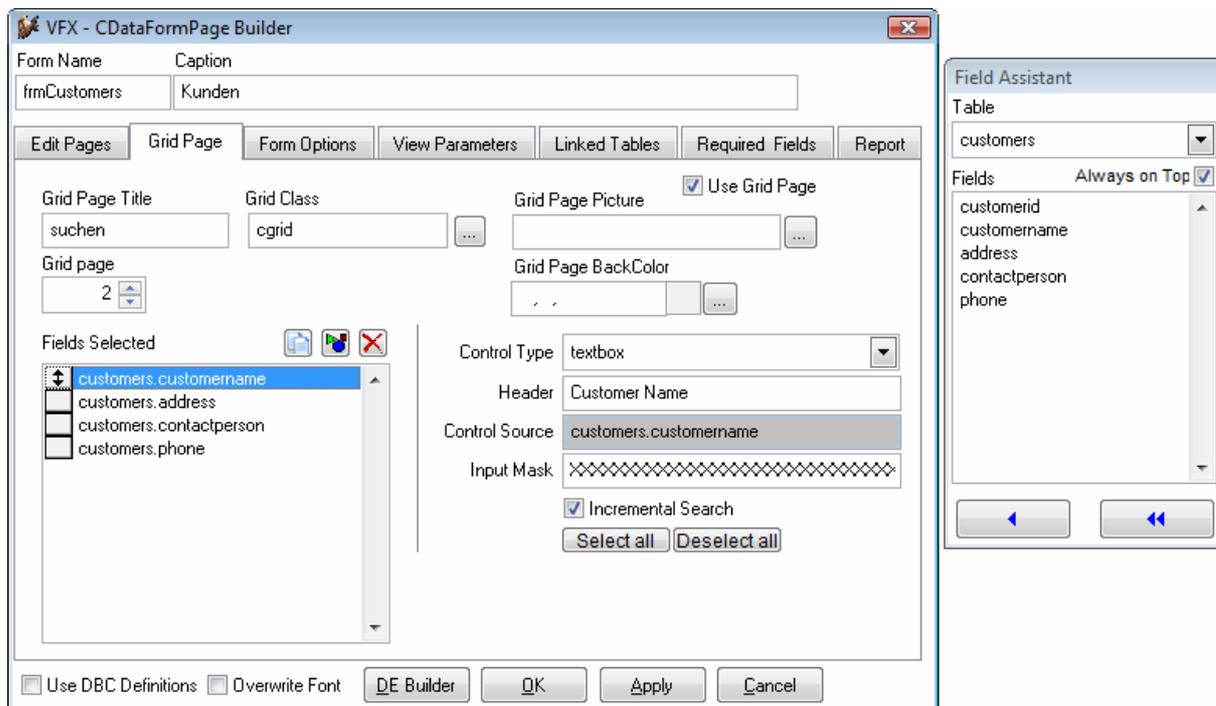
**AutoCompSource.** Name der Tabelle, die für die AutoComplete Funktion in diesem Feld verwendet werden soll. AutoComplete-Tabellen müssen mit der Anwendung nicht ausgeliefert werden. Diese Tabellen werden von VFP bei Bedarf automatisch erstellt.

**AutoComplete.** Wert der Eigenschaft AutoComplete. Die AutoComplete Funktion steht nur bei Textboxen zur Verfügung.

**Read only.** Wenn ein Steuerelement nur zur Anzeige von Informationen verwendet wird, markieren Sie dieses Kontrollkästchen.

Bei der Bearbeitung vorhandener Formulare ist die Schaltfläche  *Move or Copy Fields* sehr nützlich. In der Feldliste können beliebig viele Felder markiert werden. Mithilfe des Dialogs *Move or Copy* können die markierten Felder auf eine andere Seite des Seitenrahmens kopiert oder verschoben werden. Die Zielseite kann eine Bearbeitungsseite, die Listenseite oder die Berichtseite sein.

Wenn die ausgewählten Steuerelemente kopiert und nicht verschoben werden sollen, wird eine Kopie der Steuerelemente mit allen Eigenschaften auf der gewählten Seite angelegt.



Auf der Seite *Grid Page* werden die Spalten für das Such-Grid zusammengestellt.

**Use Grid Page.** Markieren Sie dieses Kontrollkästchen, wenn Sie eine Listenseite auf Ihrem Formular haben wollen.

**Grid Page Title.** Geben Sie die Überschrift für die letzte Seite Ihres Formulars ein, die normalerweise ein Grid mit allen Datensätzen Ihrer Tabelle oder Ansicht enthält.

**Grid Class.** Geben Sie die Klasse für das Grid ein oder benutzen Sie den Standardwert, die *CGrid*-Klasse.

**Fields Selected.** Hier sehen Sie alle für das Grid ausgewählten Felder. Um Felder auszuwählen, benutzen Sie das *Field Assistant*-Fenster, in dem alle Felder aus der Datenumgebung zur Auswahl stehen.

**Calculated Fields.**  Drücken Sie auf diese Schaltfläche um ein beliebiges berechnetes Feld hinzuzufügen.

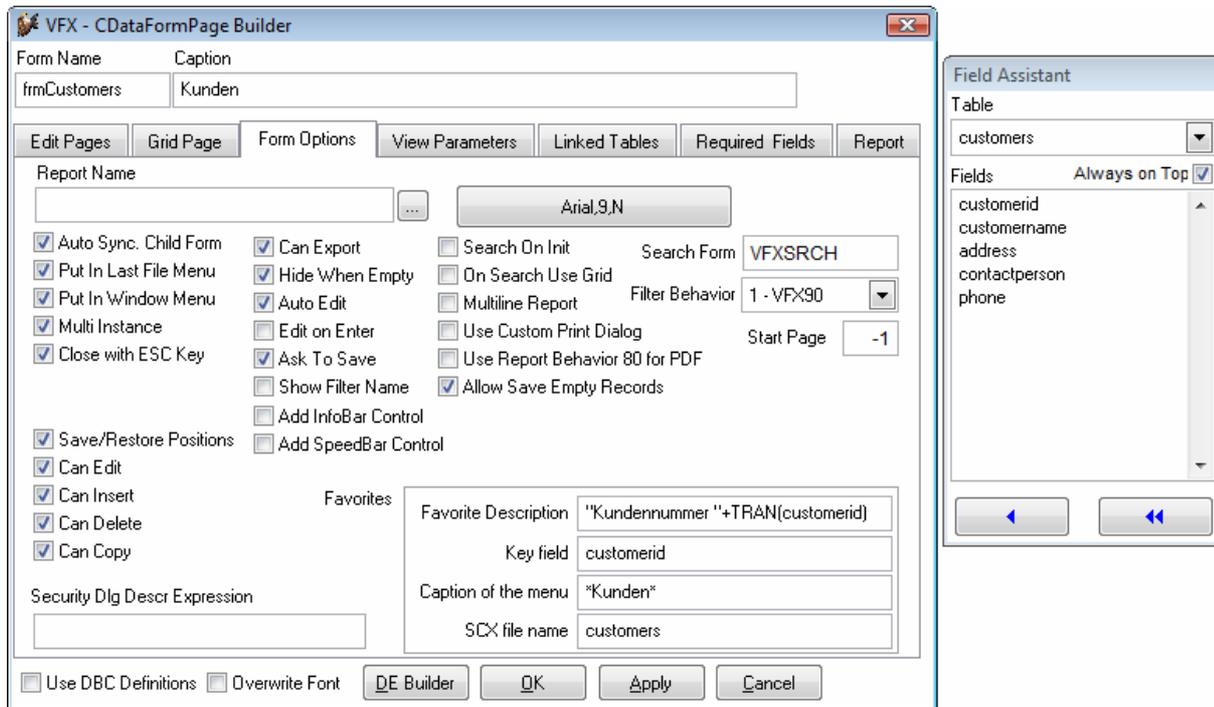
**Control Type.** Geben Sie für alle ausgewählten Felder den gewünschten Kontrolltyp an.

**Header.** Überschriften für die Spalten Ihres Grids. Die VFX – Formular Builder fügen automatisch die Überschriften aus dem Datenbank-Container ein.

**Output Mask.** Die VFX – Formular Builder erstellen die Ausgabemaske anhand der Feldlänge. Sie können die Ausgabemaske ändern, um sie an Ihre Bedürfnisse anzupassen.

**Read only.** Wenn ein Steuerelement nur zur Anzeige von Informationen verwendet wird, markieren Sie dieses Kontrollkästchen.

**Incremental Search.** Markieren Sie dieses Kontrollkästchen, wenn Sie die inkrementelle Suche für die ausgewählte Spalte aktivieren wollen. Beachten Sie, dass VFX eine temporäre Indexdatei erstellt, wenn kein Indexschlüssel für die Spalte vorhanden ist. (Mit der *CGrid*-Eigenschaft *nMaxRec* können Sie angeben ab welcher Anzahl Datensätze dem Benutzer eine Meldung angezeigt werden soll, bevor eine temporäre Indexdatei erstellt wird.)



Auf der Seite *Form Options* können Einstellungen gemacht werden, die für dieses Formular gelten.

**Report Name.** Hier können Sie den Namen eines Berichts eingeben. Wenn der Benutzer *drucken* oder *Seitenansicht* wählt, wird dieser Bericht gedruckt bzw. angezeigt. Sie brauchen für diese Funktionalität keinen Code in die Methode *OnPrint()* einzufügen. Wenn diese Eigenschaft leer gelassen wird, sucht VFX nach einem Bericht, der den gleichen Namen wie das Formular hat.

**Put in Last File Menu.** Hiermit wird die Formulareigenschaft *IPutinLastFile* festgelegt. Sie gibt an, ob die Formularüberschrift in die Liste der benutzen Dateien im Menü *Datei* eingetragen werden soll.

**Put in Window Menu.** Hiermit wird die Formulareigenschaft *IPutinWindowmenu* festgelegt. Sie gibt an ob das laufende Formular in das Menü *Fenster* eingetragen werden soll. Beachten Sie auch die Eigenschaft *nWinMnuCount* und die Methode *RefreshWindowMenu()* im Anwendungsobjekt.

**Can Edit.** Hiermit wird die Formulareigenschaft *ICanEdit* festgelegt. Sie gibt an, ob der Benutzer Datensätze im aktuellen Formular bearbeiten kann.

**Can Insert.** Hiermit wird die Formulareigenschaft *ICanInsert* festgelegt. Sie gibt an, ob der Benutzer Datensätze im aktuellen Formular einfügen kann.

**Can Copy.** Hiermit wird die Formulareigenschaft *ICanCopy* festgelegt. Sie gibt an, ob der Benutzer Datensätze im aktuellen Formular kopieren kann.

**Can Delete.** Hiermit wird die Formulareigenschaft *ICanDelete* festgelegt. Sie gibt an, ob der Benutzer Datensätze im aktuellen Formular löschen kann.

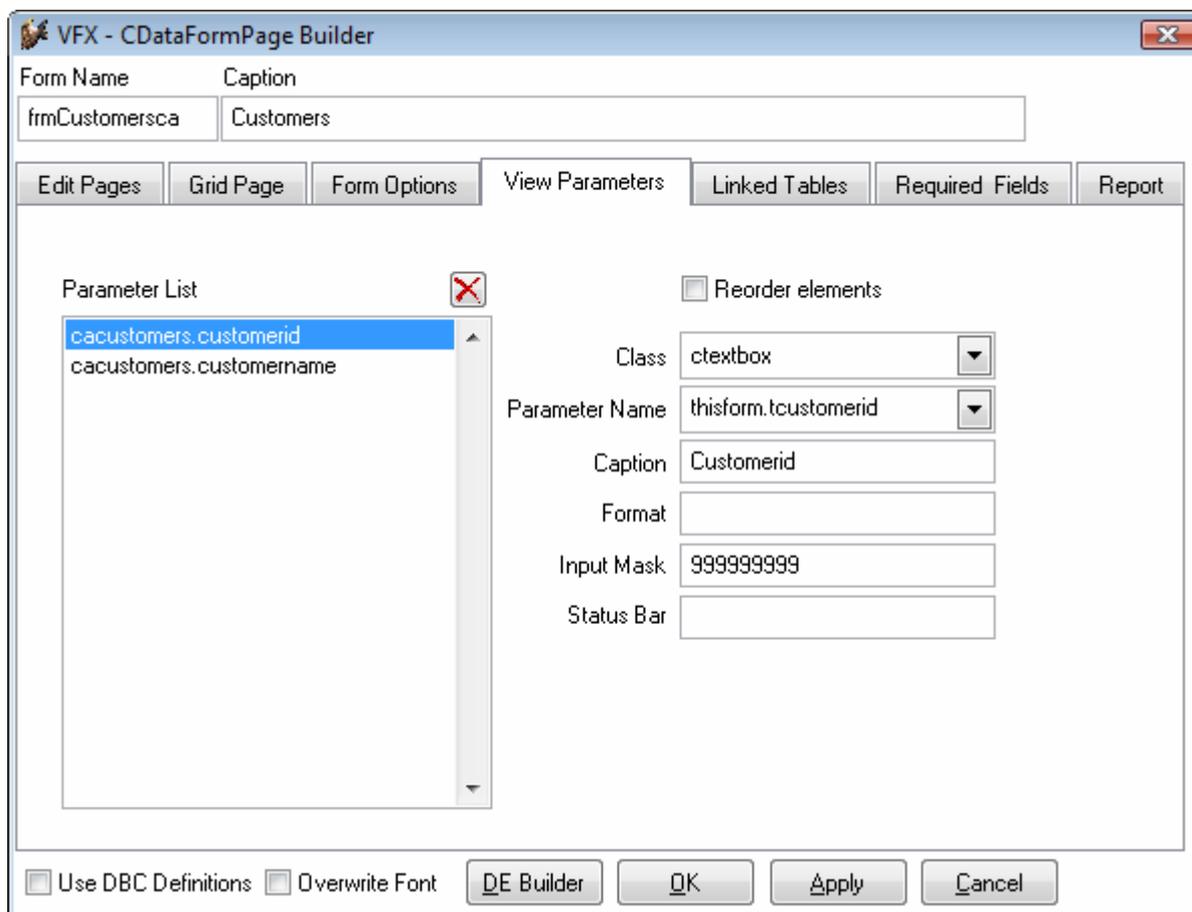
Die Einstellungen *Can Edit*, *Can Insert*, *Can Delete* und *Can Copy* werden nicht berücksichtigt, wenn zur Laufzeit individuelle Berechtigungen im Dialog Benutzerrechte gemacht werden.

**Multi Instance.** Hiermit wird die Formulareigenschaft *IMultiInstance* eingestellt. Standardmäßig können alle Formulare, die Sie mit VFX erstellen, mehrmals geöffnet werden.

**Close with ESC key.** Hier wird die Formulareigenschaft *ICloseonEsc* eingestellt, die angibt, ob der Benutzer ein Formular mit der Escape-Taste schließen kann. Wenn Anpassungen für den angemeldeten Benutzer erlaubt sind, kann der Benutzer dieses Verhalten selbst einstellen.

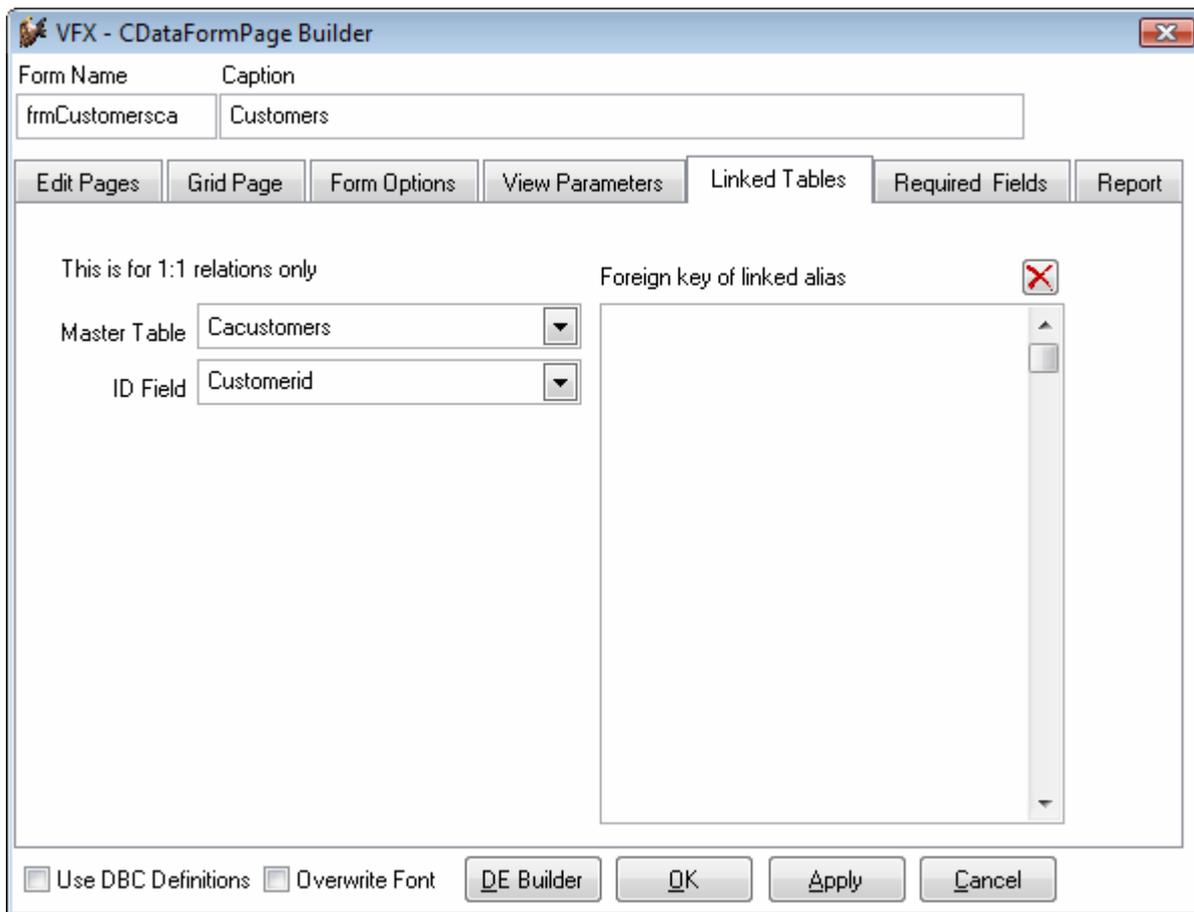
**Save/Restore positions.** Hier wird die Formulareigenschaft *ISavePosition* eingestellt, die angibt, ob die Positionen und andere Formulareinstellungen in der VFX-Ressourcentabelle gespeichert werden sollen.

**Add Speedbar Control.** Dieses Kontrollkästchen fügt dem Formular eine Schaltflächenleiste hinzu.



Auf der Seite *View Parameters* können Felder ausgewählt werden, für die der Builder automatisch Steuerelemente am oberen Rand des Formulars platziert. Für jedes Feld wird automatisch eine Formulareigenschaft angelegt, die dem gewählten Feldnamen entspricht.

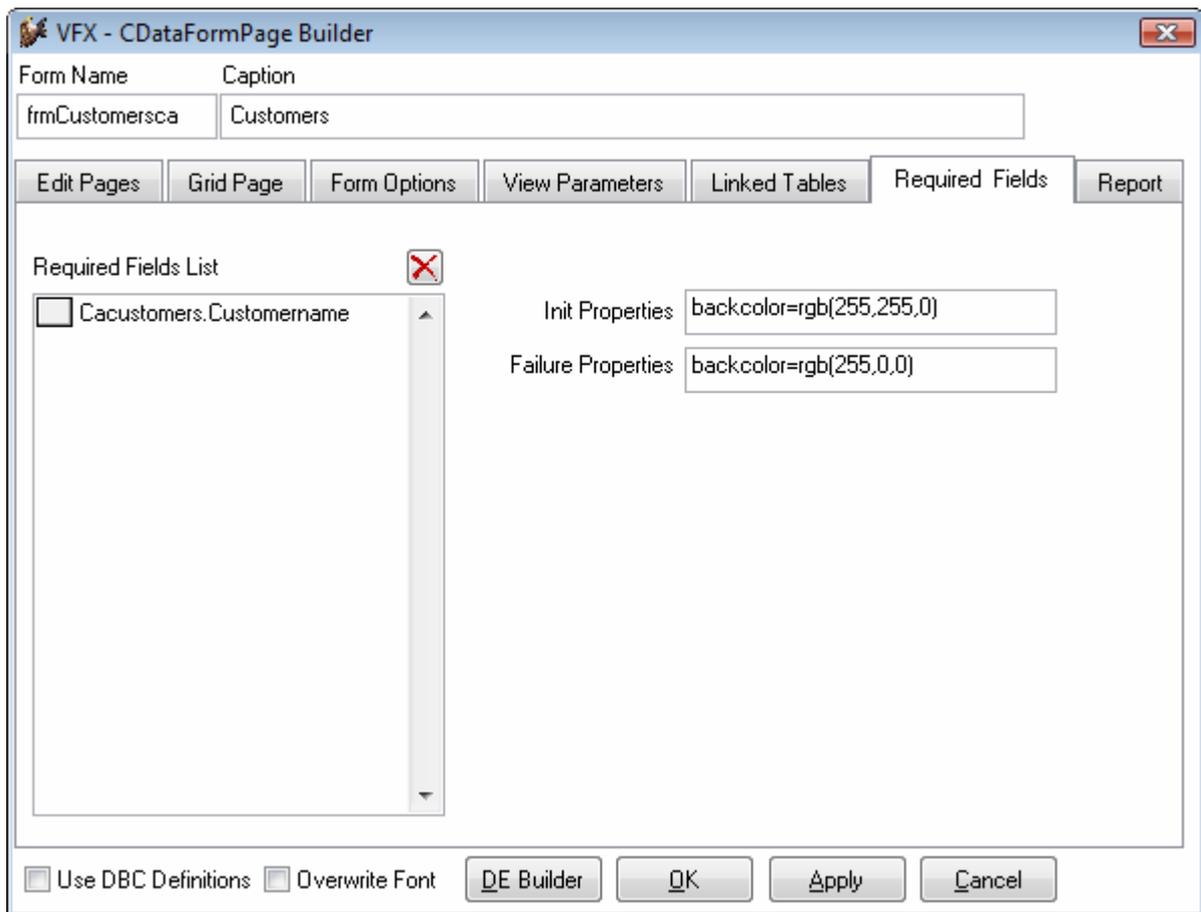
Das einzige, was manuell gemacht werden muss, ist die Where Klausel des Cursoradapters zu ergänzen, auf den die Parameter Steuerelemente wirken sollen.



VFX Anwendungen unterstützen 1:1-Beziehungen zwischen der Hauptbearbeitungstabelle und weiteren Tabellen. Hierdurch bekommt der Entwickler eine größere Flexibilität bei der Entwicklung komplexer Datenbanken ohne zusätzlichen Code zur Gewährung der Integrität der Datenbank schreiben zu müssen. VFX hält die Daten automatisch konsistent.

Es ist nicht notwendig, dass die Hauptbearbeitungstabelle und die in Beziehung stehenden Tabellen Primärschlüssel mit denselben Namen haben. Die Schlüsselfelder der in Beziehung stehenden Tabellen werden beim Einfügen neuer Datensätze mit dem Primärschlüssel der Haupttabelle gefüllt. Beim Löschen von Datensätzen in der Haupttabelle werden automatisch auch die in Beziehung stehenden Datensätze gelöscht.

Auf der Seite *Linked Tables* muss zunächst die Hauptbearbeitungstabelle mit dem Primärschlüssel ausgewählt werden. In der Parameterliste können Felder aus in Beziehung stehenden Tabellen gewählt werden. Es kann genau ein Feld je Tabelle ausgewählt werden. Über die ausgewählten Felder wird die Beziehung hergestellt und die referenzielle Integrität gewährleistet. Wenn versucht wird ein zweites Feld aus einer Tabelle auszuwählen, so wird das zuerst gewählte Feld überschrieben.



Auf der Seite *Required Fields* können Felder hinzugefügt werden, die beim Speichern der Daten des Formulars nicht leer bleiben dürfen.

Mithilfe der Formulareigenschaften *cRequiredFields*, *cRequiredFieldInitProps*, *cRequiredFieldFailureProps* und *cRequiredFieldFailureForm* kann verhindert werden, dass Feldinhalte mit Nullwerten oder ohne Inhalt gespeichert werden.

Der Listbox *Required Fields List* kann eine beliebige Anzahl von Datenfeldern aus dem Feldassistenten zugewiesen werden. Während der Initialisierung des Formulars werden alle Steuerelemente auf eine Controlsourc aus dieser Liste überprüft. Alle Steuerelemente mit einer entsprechenden Controlsourc werden als erforderliche Eingabefelder behandelt.

Die Liste der erforderlichen Eingabefelder wird vom Form Builder der Formulareigenschaft *cRequiredFields* zugewiesen.

In der Textbox *Init Properties* kann eine Semikolon-Separierte Liste mit Zuweisungen an Eigenschaften in der Form

```
PropertyName = cExpression [; PropertyName = cExpression ]
```

eingetragen werden. Für alle erforderlichen Eingabefelder werden während der Initialisierung diese Zuweisungen ausgeführt. Im Beispiel aus der Abbildung bekommen alle Steuerelemente, die ein erforderliches Datenfeld als Controlsourc haben, die Vordergrundfarbe rot.

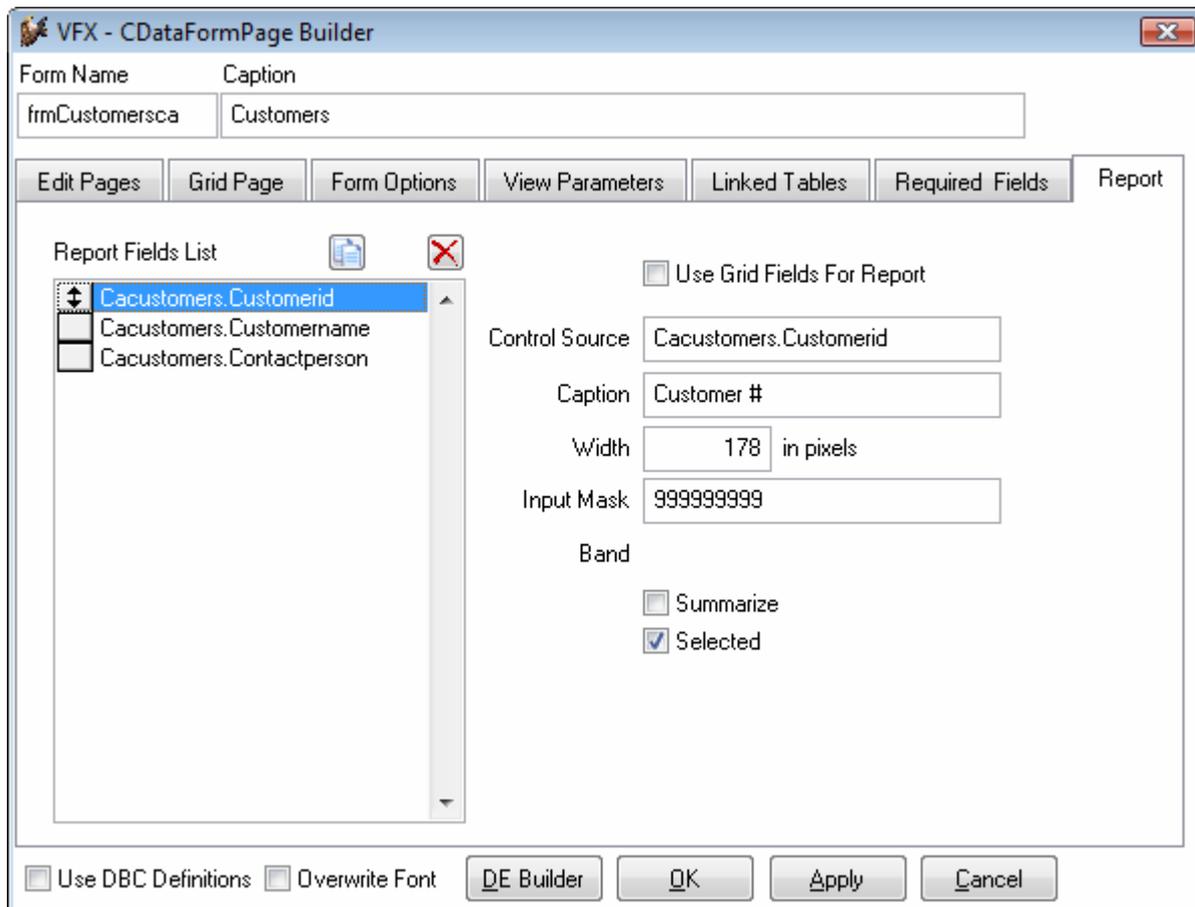
Wenn mehreren Eigenschaften Werte zugewiesen werden sollen, werden die Zuweisungen durch Semikolon getrennt. Wenn beispielsweise alle erforderlichen Eingabefelder mit einer fetten Schrift, roten Schriftfarbe und einem hellgelben Hintergrund angezeigt werden sollen, ist im Feld *Init Properties* folgender Wert einzutragen:

```
FontBold = .T.; ForeColor = RGB(255,0,0); BackColor = RGB(255,255,196)
```

Auf diesem Weg kann dem Benutzer auf einfachem Weg gezeigt werden, welche Felder ausgefüllt werden müssen. Der Wert des Feldes *Init Properties* wird der Formulareigenschaft *cRequiredFieldInitProps* zugewiesen.

Beim Speichern der Daten des Formulars werden alle erforderlichen Eingabefelder auf einen eingegebenen Wert überprüft. Wenn ein fehlender Wert festgestellt wird, werden dem entsprechenden Steuerelement die Eigenschaften aus dem Feld *Failure Properties* zugewiesen. Die Eingabe erfolgt nach den gleichen Regeln, wie beim Feld *Init Properties*. Der Wert des Feldes *Failure Properties* wird der Formulareigenschaft *cRequiredFieldFailureProps* zugewiesen.

Solange nicht alle erforderlichen Eingabefelder mit Werten gefüllt sind, werden die Daten des Formulars nicht gespeichert.



Häufig ist es erforderlich auf Berichten Felder zu drucken, die auf der Listenseite eines Formulars nicht zur Verfügung stehen. Genauso kann es möglich sein, dass Felder aus dem Grid nicht gedruckt werden sollen. Die Seite *Report* ermöglicht es Felder auszuwählen, die zur Laufzeit auf der Seite *Erweitert* des Druckdialogs zur Auswahl stehen sollen. Hier kann eine Vorauswahl der standardmäßig zu druckenden Felder und der Felder mit Summierung gemacht werden.

Für jedes Feld können die Breite des Feldes, eine Eingabemaske und eine Überschrift vorgegeben werden.

Wenn das Kontrollkästchen „Use Grid Fields For Report“ markiert ist, werden dem Anwender zur Laufzeit alle Spalten für den Bericht angeboten, die im Suchgrid zur Verfügung stehen.

**OK.** Wählen Sie diese Schaltfläche, um Ihr Formular generieren zu lassen. Dies dauert einige Sekunden und das Ergebnis ist ein Formular, auf dem Sie die gewünschte Anzahl von Bearbeitungsseiten mit den gewählten Feldern auf jeder Seite haben. Wenn Sie mehr Felder gewählt haben als untereinander auf eine Seite passen, werden zwei Spalten erzeugt.

**Apply.** Hat die gleiche Funktion wie die Schaltfläche *OK*, schließt den VFX Formular Builder jedoch nicht.

**Cancel.** Bricht die Ausführung des VFX-Formular-Builders ab. Jede Auswahl und Eingabe geht dabei verloren.

Beim ersten Erstellen des Formulars wird automatisch ein Eintrag in der Tabelle *Vfxfopen.dbf* angelegt, sodass das Formular über den Öffnen-Dialog gestartet werden kann.

Starten Sie Ihre Anwendung, wählen Sie im Öffnen-Dialog Ihr neu erstelltes Formular und starten Sie es mit einem Mausklick. Testen Sie es und prüfen Sie, wo Ihr Formular erweitert werden muss.

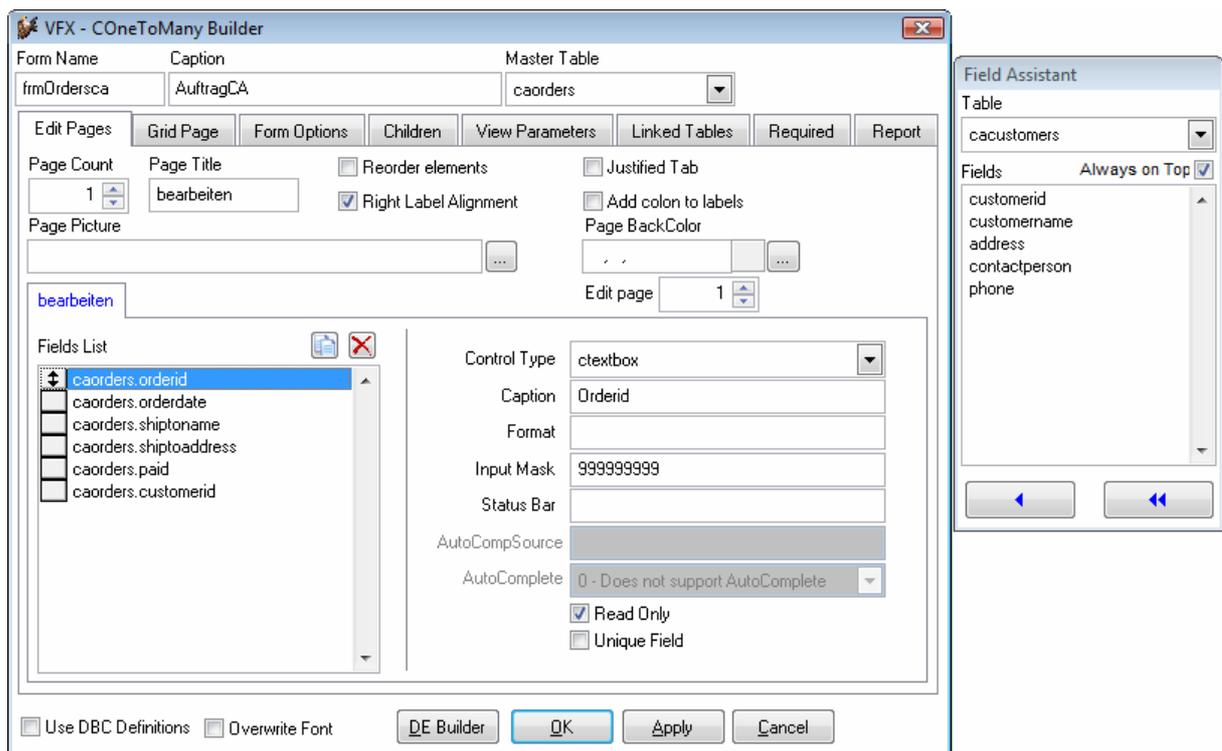
Um einen VFX-Formular-Builder für ein existierendes Formular aufzurufen, bewegen Sie die Maus auf den weißen Hintergrund des Formular-Designers, drücken Sie die rechte Maustaste und wählen Sie Builder.

## VFX – COneToMany Builder

Das 1:n-Formular ist eine Weiterentwicklung des Standard-VFX-Datenbearbeitungs-Formulars. Das bedeutet, dass Sie auf einem einzigen Formular die normalen Datenbearbeitungsfunktionen haben können und ein Grid mit den Child-Datensätzen zu dem aktuell angezeigten Hauptdatensatz haben. VFX erlaubt es Ihnen, auch mehrere Child-Tabellen zu einer Haupttabelle auf mehreren Seiten eines Seitenrahmens zu bearbeiten. Wenn Sie viele Eingabefelder in Ihrer Child-Tabelle haben, können Sie die Felder auf mehrere Seiten eines Seitenrahmens verteilen. Das erlaubt Ihnen, eine große Anzahl verschiedenster Anwendungen abzudecken ohne wirklich programmieren zu müssen.

In 1:n-Beziehungen stellen Sie die Verbindung von einem Hauptdatensatz zu den Child-Datensätzen her. Ein gutes Beispiel für eine 1:n-Beziehung ist die Verbindung zwischen Aufträgen (Haupttabelle) und Auftragspositionen (Child-Tabelle) in jedem Auftragsbearbeitungssystem.

Der VFX – COneToMany Builder hilft Ihnen bei der Erstellung von anspruchsvollen 1:n-Formularen, ohne zu programmieren. Wenn Sie die 1:n-Beziehung zwischen der Haupttabelle und der Child-Tabelle hergestellt haben, können Sie 1:n-Formulare genauso einfach erstellen wie Standard-VFX-Datenbearbeitungsformulare. Wenn Sie mehrere Child-Tabellen mit einer Haupttabelle verbinden wollen, müssen Sie von jeder Child-Tabelle eine Beziehung zu der Haupttabelle herstellen.

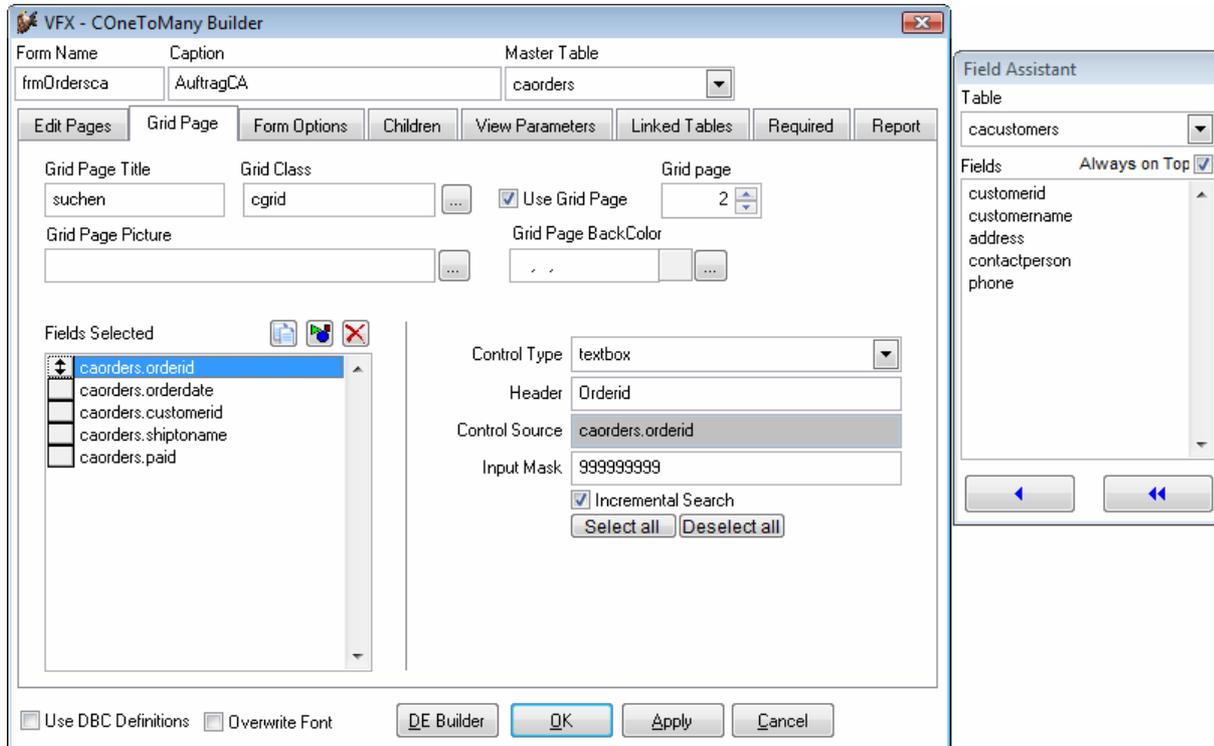


Genau wird im VFX – CDataFormPage Builder können auf der Seite *Edit Pages* Steuerelemente für die Bearbeitungsseiten hinzugefügt werden.

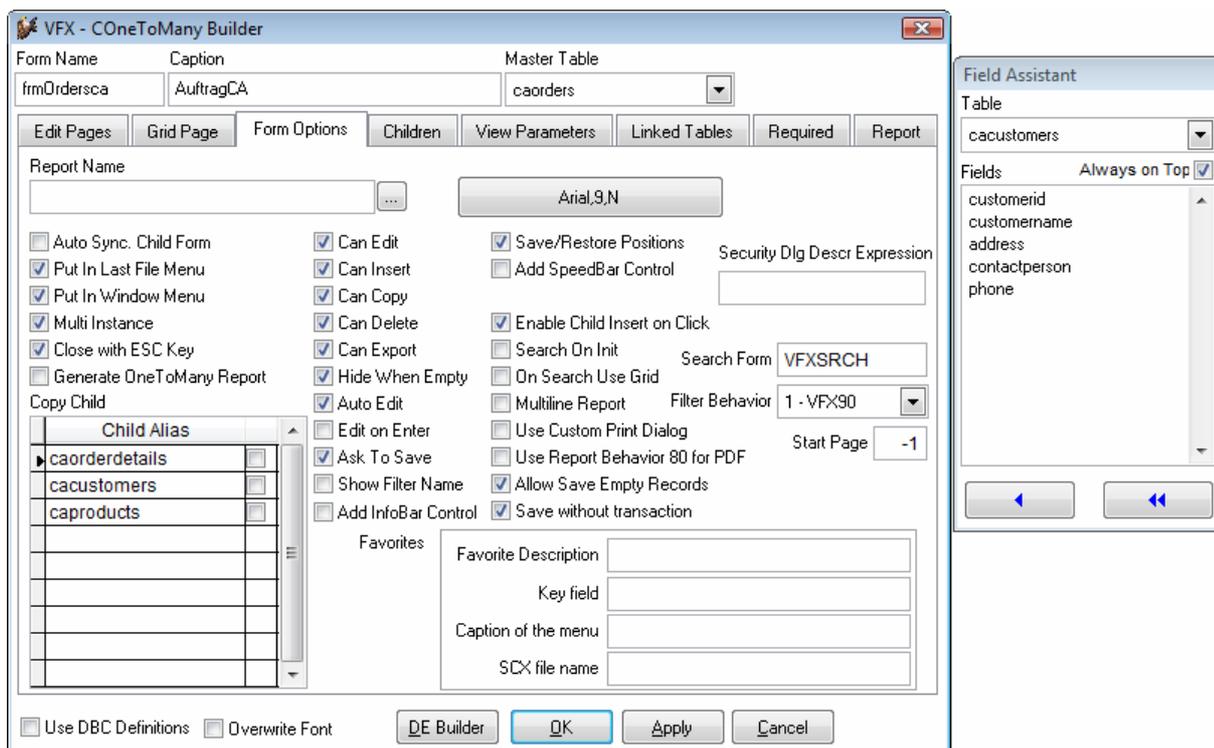
**Form Name.** Geben Sie den Namen des neuen Formulars ein. Der VFX – Form Wizard hat bereits einen Standardnamen entsprechend den Namenskonventionen zugewiesen. Der Name beginnt mit *frm*. Selbstverständlich können Sie Ihrem Formular einen beliebigen Namen geben, aber wir empfehlen Ihnen, sich an die allgemeinen Namenskonventionen zu halten.

**Caption.** Geben Sie die Überschrift für Ihr Formular ein. Während Sie die Überschrift eingeben, wird diese bereits in der Überschrift des Formular-Builders angezeigt.

**Master Table.** Name der Haupttabelle oder Ansicht.



Das Suchgrid kann ebenfalls genauso, wie im VFX – CDataFormPage Builder erstellt werden.



Zusätzlich zu den Optionen im VFX – CDataFormPage Builder kann auf der Seite Form Options ausgewählt werden, aus welchen Child Tabellen Datensätze kopiert werden sollen, wenn der Parent Datenstz kopiert wird.

Auf der Seite Children können die Steuerelemente für den Child Teil des Formulars zusammengestellt werden. In jedem Fall enthält die erste Seite mit Child Daten ein Grid für die Bearbeitung. Auf den Folgeseiten kann eingestellt werden, ob die Daten in einem Grid oder mit anderen Steuerelementen bearbeitet werden sollen.

**Page Count.** Geben Sie ein, wie viele Child-Grids Ihr Formular haben soll. Für die meisten 1:n-Formulare wird ein Grid ausreichen. Wenn Sie mehrere Child-Tabellen haben, werden Sie diese über mehrere Seiten verteilen wollen. Entsprechend der Anzahl der Seiten, die Sie gewählt haben, erscheint der Seitenrahmen des Formular-Builders mit der gewählten Anzahl von Seiten. Wenn Sie zwei Seiten einstellen, hat der Seitenrahmen zwei Seiten, wenn Sie drei Seiten einstellen, hat der Seitenrahmen drei Seiten usw.

**Page Title.** Geben Sie die Überschrift für das aktuell gewählte Child-Grid an. Wenn Sie die Überschrift für die zweite Seite eingeben wollen, drücken Sie auf die zweite Seite. Der VFX – COneToMany Builder zeigt sofort den eingegebenen Text als Überschrift der jeweiligen Seite an.

**Child Table.** Geben Sie die Datenquelle für Ihr Child-Grid an. Achtung: Es ist sehr wichtig, diese Einstellung zu machen. Wenn Sie diese Eigenschaft nicht einstellen, wird Ihr Formular nicht richtig funktionieren.

**Justified Tab.** Markieren Sie dieses Kontrollkästchen, wenn die Seitenüberschriften justiert sein sollen. Ansonsten haben die Überschriften eine variable Länge und füllen nicht die Breite des Seitenrahmens.

**Inplace Editing.** Markieren Sie diese Option, wenn Sie Daten in das Child-Grid eingeben wollen, was normalerweise der Fall ist.

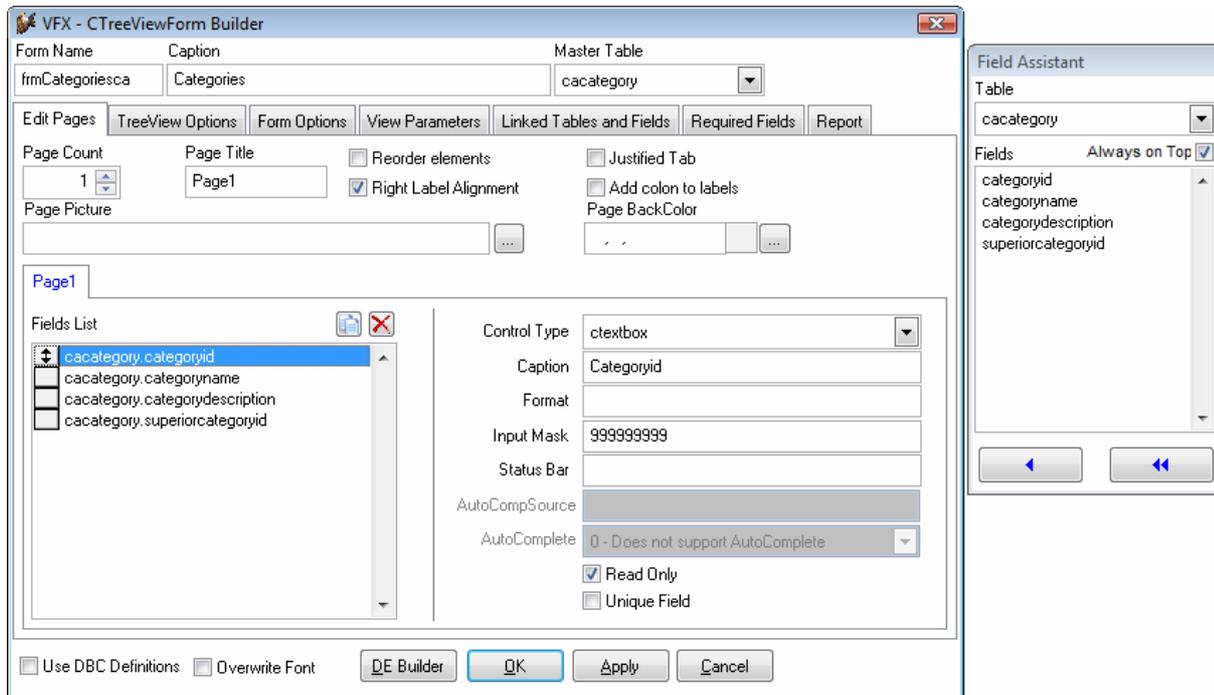
**^Ins+^Canc.** Markieren Sie diese Option, wenn Sie die Möglichkeit haben wollen, mit Strg+Einfg Datensätze einzufügen und mit Strg+Entf Datensätze im Child-Grid zu löschen.

Die weiteren Seiten des VFX – COneToMany Builder sind identisch mit den entsprechenden Seiten des VFX – CDataFormPage Builder.

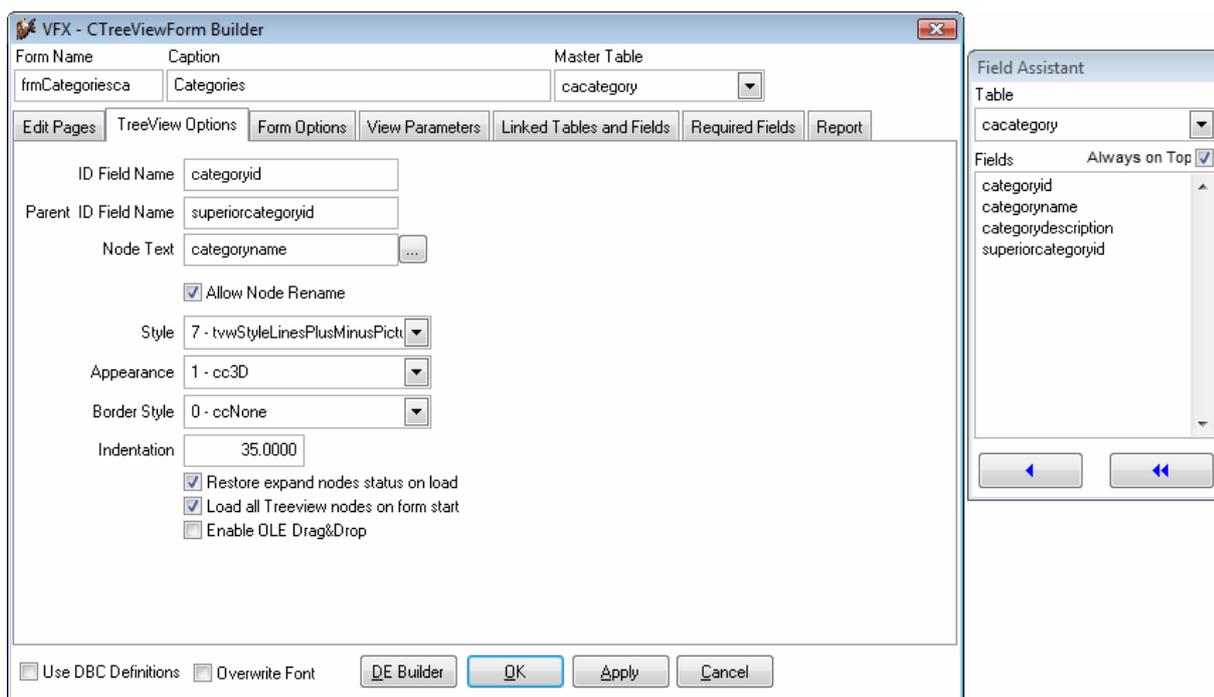
## VFX – CTreeViewForm Builder

Der Haupteinsatzzweck dieser Klasse ist die Darstellung von Daten aus einer Tabelle in einer Baumstruktur. Die Baumstruktur gibt dem Endanwender einen kompletten Überblick über die hierarchischen Beziehungen in einer Tabelle.

Diese Klasse basiert auf der Klasse *CDataForm* (*Vfxform.vcx*) und enthält ein Treeview-Steuerelement aus der Klasse *CTreeView* (*Vfxappl.vcx*). Die Klasse kombiniert die Funktionalität von *CDataFormPage* mit den Möglichkeiten der hierarchischen Datenpräsentation in einer Baumstruktur. Wenn ein Eintrag im Treeview-Steuerelement ausgewählt wird, wird der Datensatzzeiger in der zugrunde liegenden Tabelle mitgeführt und der Anwender kann die Daten im rechten Teil des Formulars bearbeiten.



Der Builder arbeitet ähnlich dem VFX – CDataFormPage Builder. Die Einstellungen können auf den Seiten *Edit Pages* und *Form Options* genauso gemacht werden, wie im VFX – CDataFormPage Builder. Zusätzlich müssen die Einstellungen für das Treeview-Steuerelement auf der Seite *TreeView Options* gemacht werden.



*IDFieldName* – Hier wird der Name des Feldes mit dem Primärschlüssel der Bearbeitungstabelle eingetragen.

*ParentIDFieldName* – Diese Eigenschaft enthält den Namen des Feldes, in dem der Primärschlüssel des Parent-Datensatzes gespeichert ist

*NodeText* – Hier kann entweder der Name eines Feldes, das einen Beschreibungstext enthält eingetragen werden oder es wird ein Ausdruck eingetragen, der zur Laufzeit evaluiert wird und dessen Rückgabewert als Bezeichnung in der Baumstruktur angezeigt wird. Wenn ein Feldname verwendet wird, kann dem Anwender erlaubt werden die Bezeichnung direkt im Treeview-Steuerelement zu ändern. Dies hängt vom Wert der Eigenschaft *AllowNodeRename* ab. Wenn *AllowNodeRename* auf .T. gesetzt ist, kann der Anwender die Bezeichnungen im Treeview-Steuerelement ändern. Dabei werden die Daten im zugrunde liegenden Tabellenfeld automatisch aktualisiert.

*AllowNodeRename* – Über diese Eigenschaft wird gesteuert, ob der Anwender die Bezeichnung im Treeview-Steuerelement ändern kann. Die Bearbeitung der Bezeichnung im Treeview-Steuerelement ist nur möglich, wenn die Bezeichnung auf einem einzelnen Tabellenfeld basiert. Dieses Tabellenfeld wird bei der Bearbeitung automatisch aktualisiert.

*LoadAllTreeViewNodes* – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, werden alle Knoten des Treeview beim Laden des Formulars geladen. Wenn der Wert dieser Eigenschaft auf .F. eingestellt ist, werden beim Laden des Formulars nur die sichtbaren Knoten geladen. In diesem Fall werden beim Öffnen eines Knotens dynamisch die Untereinträge geladen.

*RestoreTreeViewStatus* – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird die Liste der geöffneten Knoten beim Schließen des Formulars für den angemeldeten Benutzer in der Ressourcentabelle gespeichert. Beim nächsten Laden des Formulars wird das Treeview dementsprechend wiederhergestellt.

Diese Einstellungen entsprechen denen des TreeView-ActiveX-Steuerelements.

*Style:* 0 - tvwStyleText  
1 - tvwStylePictureText  
2 - tvwStylePlusMinusText  
3 - tvwStylePlusMinusPictureText  
4 - tvwStyleLinesText  
5 - tvwStyleLinesPictureText  
6 - tvwStyleLinesPlusMinusText  
7 - tvwStyleLinesPlusMinusPictureText

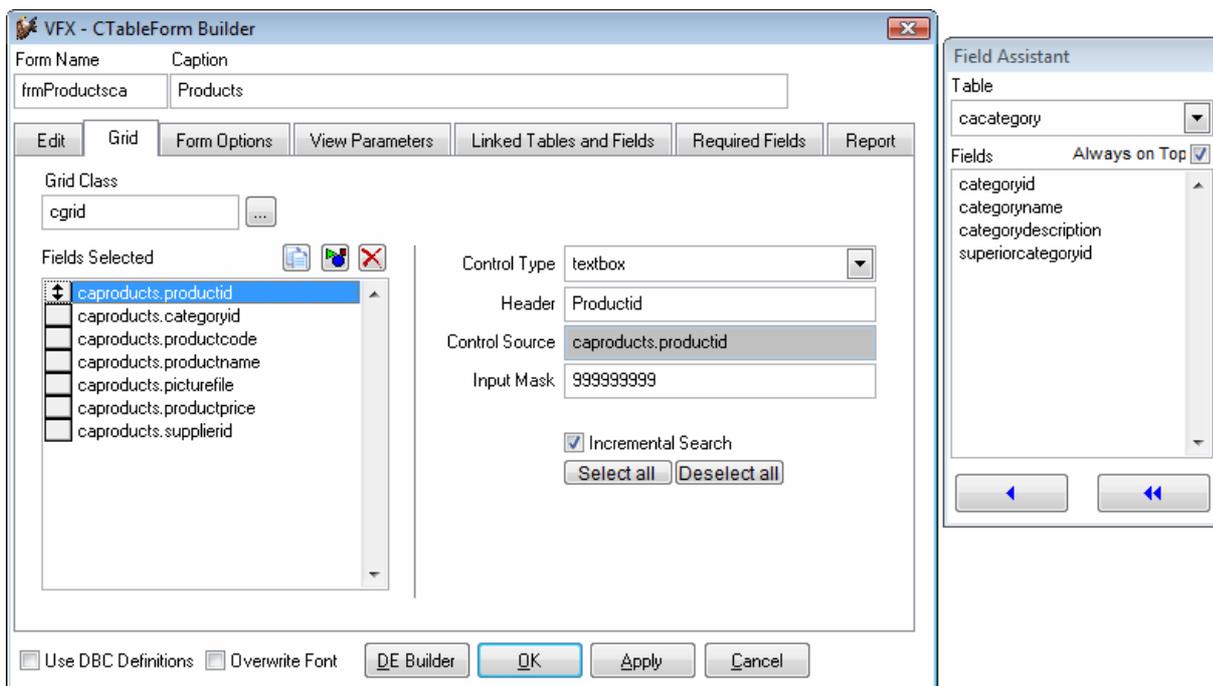
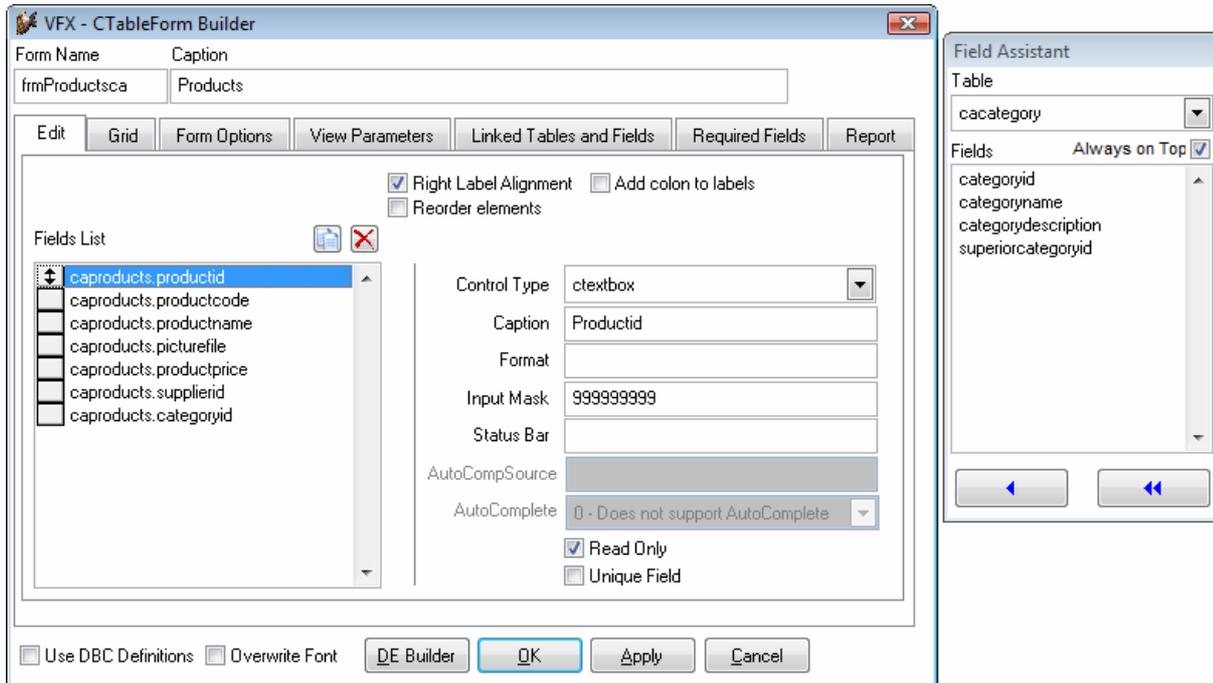
*Appearance:* 0 - ccFlat  
1 - cc3D

*BorderStyle:* 0 - ccNone  
1 - ccFixedSingle

*Indentation:* Diese Eigenschaft bestimmt die Breite des Einzugs der Knoten.

## VFX – CTableForm Builder

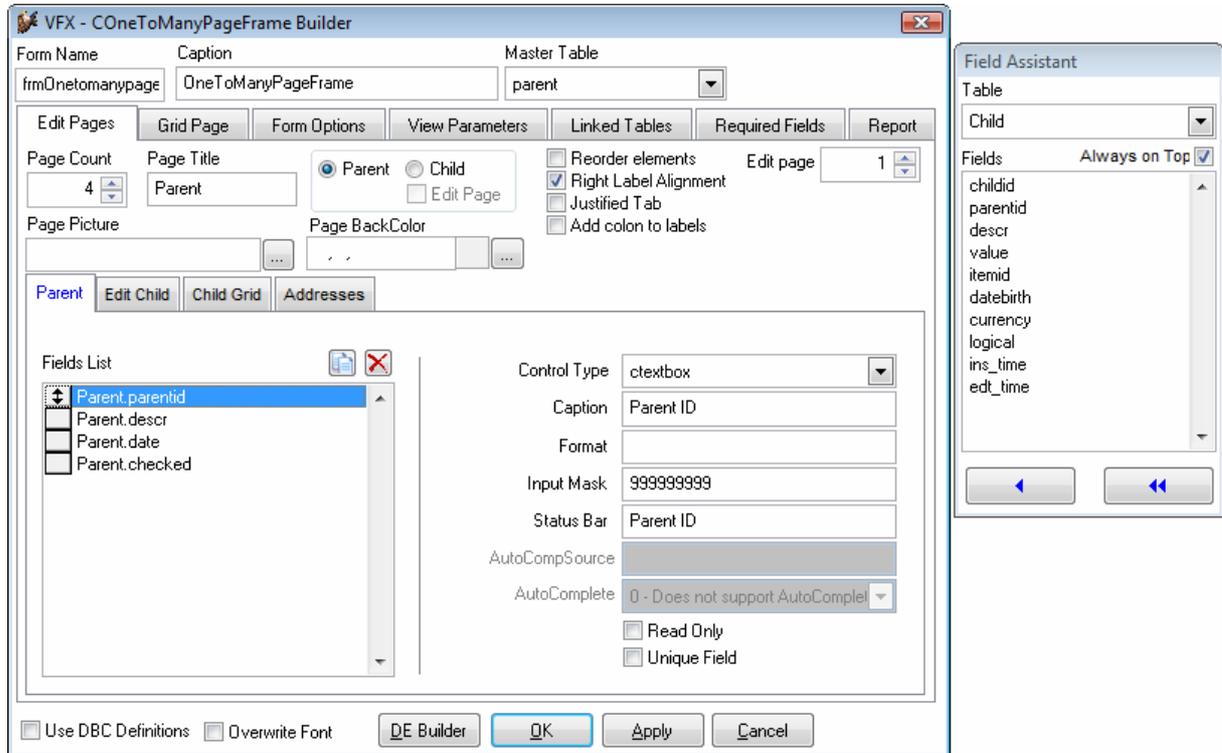
Eine weitere Formularart ist die CTableForm. Bei diesem Formular werden das Listen-Grid und die Steuerelemente nebeneinander oder untereinander dargestellt. Es eignet sich daher insbesondere für Formulare mit nur wenigen Eingabefeldern.



## VFX – COneToManyPageFrame Builder

Die Klasse *COneToManyPageframe* gibt dem Entwickler die Möglichkeit auf einem Seitenrahmen auf verschiedenen Seiten Parent Daten und Child Daten darzustellen. Die Klasse vereint die Vorteile der Klasse *CDataFormPage* mit der Möglichkeit Child Daten zu bearbeiten.

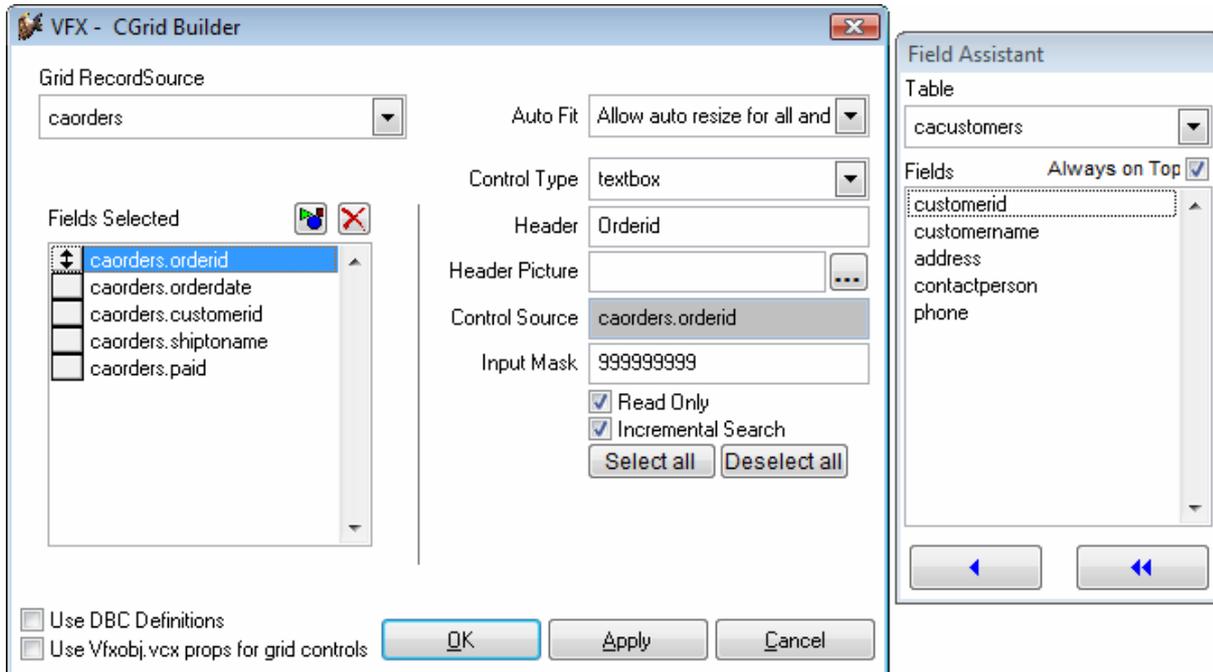
Wenn die aktive Seite des Seitenrahmens Steuerelemente vom Typ „Parent“ enthält, bezieht sich die Navigation auf die Parent-Daten. Wenn die aktive Seite des Seitenrahmens Steuerelemente vom Typ „Child“ enthält, bezieht sich die Navigation auf die Child-Daten. Auf Child-Seiten können wahlweise beliebige Steuerelemente oder ein Childgrid platziert werden.



Zusätzlich zu den Einstellungen, die der Entwickler im Form Builder für andere Formulareklassen machen kann, ist es hier erforderlich einzustellen, ob eine Seite Parent Daten oder Child Daten enthalten soll. Wenn eine Seite eine Child Seite sein soll, kann eingestellt werden, ob sich Steuerelemente oder ein Child Grid auf dieser Seite befinden soll.

## VFX – Grid Builder

Sollen nur Änderungen am Grid vorgenommen werden, braucht nicht der Form Builder verwendet zu werden. Mit dem VFX – Grid Builder können die Einstellungen des Grids verändert werden. Wie alle VFX Builder ist auch der Grid Builder reentrant.

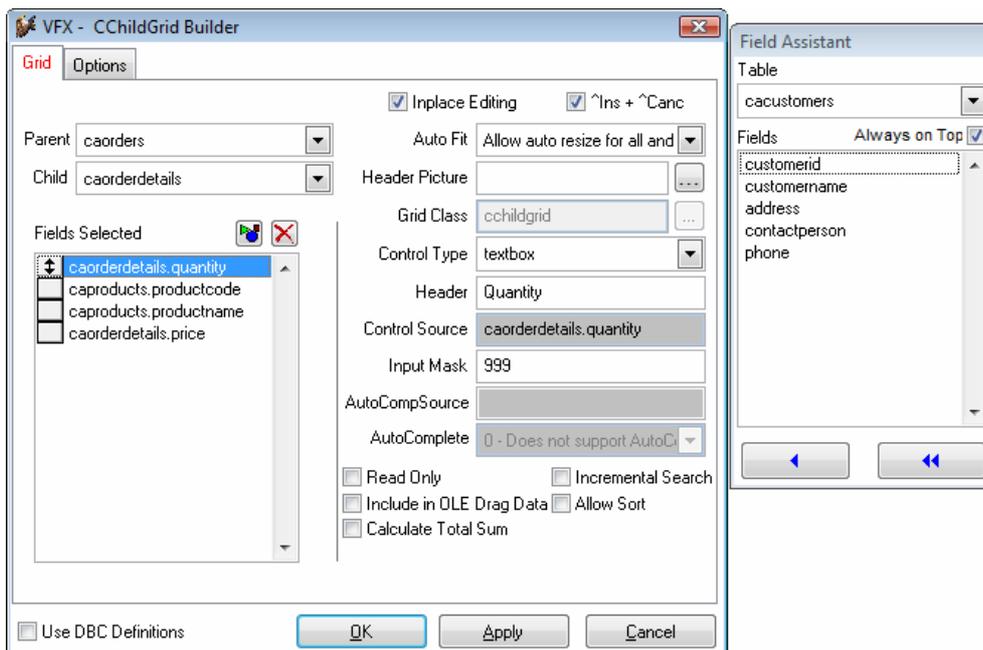


## VFX – CChildgrid Builder

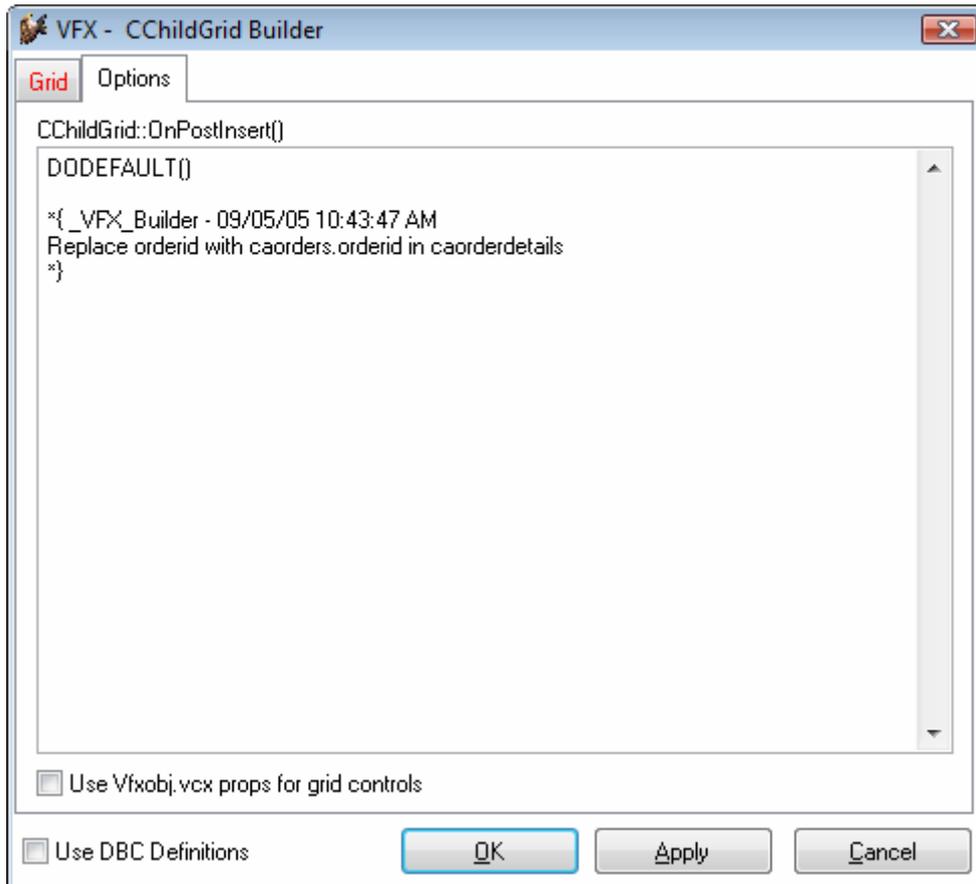
Der VFX – CChildGrid Builder erlaubt die Funktionalität der Child Grids zu erweitern. Benutzen Sie diesen Builder, um die Felder für das Grid zusammenzustellen oder um den Code der Methode *OnPostInsert()* zu bearbeiten. Diese Methode wird immer dann ausgeführt, wenn dem Child Grid ein neuer Datensatz hinzugefügt wurde.

In der *OnPostInsert()*-Methode des Child-Grids müssen Sie das Feld der Child-Tabelle ausfüllen, das die Verknüpfung zur Haupttabelle herstellt. Normalerweise benötigen Sie dafür folgenden Code:

```
REPLACE <ChildLinkField> WITH <Master.MasterField> IN <ChildTable>
```



Auf der zweiten Seite mit dem Namen *Options* können Sie den Code der *OnPostInsert()*-Methode bearbeiten, um das Feld der Child-Tabelle mit dem Wert der Haupttabelle zu füllen.



Wenn einfache Schlüssel verwendet werden, ist der von VFX generierte Code in dieser Methode in der Regel richtig.

## VFX – Parent/Child Builder

Obwohl es einen speziellen VFX-BUILDER zur Erstellung von 1:n-Formularen gibt, ist es manchmal besser, Child-Daten in einem eigenen Formular zu bearbeiten. Das ist insbesondere dann der Fall, wenn Sie das Child-Formular auch für die direkte Bearbeitung einsetzen und nicht nur durch das Hauptformular einsetzen wollen. Wenn Sie außerdem viele Felder auf dem Child-Formular haben, kann es schwierig werden, diese in einem 1:n-Formular zu bearbeiten.

Eine besondere Stärke von VFX ist die Verwendung der Linked Child-Technik. Dabei werden zwei Formulare logisch miteinander verbunden. Ein Formular dient dabei als Parent-Formular. Als Parent-Formular kann jede VFX-Formularklasse dienen. Auch das Child-Formular kann auf jeder VFX-Formularklasse basieren.

Beim Bewegen des Satzzeigers im Parent-Formular wird die Ansicht im Child-Formular automatisch aktualisiert und es werden die zum aktuellen Parent gehörenden Datensätze angezeigt.

Wenn das Child-Formular auf einer Tabelle basiert, wird ein Filter verwendet, um den sichtbaren Datenbereich einzuschränken. Wenn das Child-Formular auf einem Cursoradapter basiert, wird bei Bedarf ein Cursorrefresh() durchgeführt um die gewünschte Datenmenge anzuzeigen. Der zugrunde liegende Cursoradapter darf dabei genau einen variablen Ansichtsparameter haben, der dem Parent Schlüssel entsprechen muss.

Ein Parent Formular kann mehrere, verschiedene Child Formulare aufrufen. Ein Child-ormular kann wiederum als Parent für andere Child-Formulare dienen.

Dem Child Formular wird der Schlüssel des Parent Formulars übergeben. Im Child Formular sind nur die Daten sichtbar, die dem Schlüssel des Parent Datensatzes entsprechen.

Im VFX – Parent/Child Builder können beliebig viele Child-Formulare verwaltet werden.

VFX - Extended Parent/Child Builder

Parent Form: frmCustomersca  Auto Sync. Child Form  Close Child Form on Exit

The caption will be evaluated. Include constant text in quotation marks.

Command Type	Child Form	...	Parent field	Child field
Child Form	ORDERSCA.SCX		cacustomers.customerid	customerid

Onmore | **Advanced** | Help

Child Form: ORDERSCA.SCX

Parent Field (Fix Field Value): cacustomers.customerid

Child Field (Fix Field Name): customerid

Child Form Caption Expression: "Aufträge von "+tran(cacustomers.customerid)

Text for Open Form Expression:

Description for Open Form Expression:

Available on onMoreDialog

OK Apply Cancel

VFX - Extended Parent/Child Builder

Parent Form: frmCustomersca  Auto Sync. Child Form  Close Child Form on Exit

The caption will be evaluated. Include constant text in quotation marks.

Command Type	Child Form	...	Parent field	Child field
Child Form	ORDERSCA.SCX		cacustomers.customerid	customerid

Onmore | **Advanced** | Help

Auto Sync. Child Form  Close Child Form on Exit

Parent Form Behavior: None

Child Form Position: None

Child Form Mode: Default

Child Form Active Page: Default Child Form Active Page Number:

Unique Identifier: Code Identifier:

Child Form Filter:

Record Position Expression:

OK Apply Cancel

In der Spalte *Child Form* kann der Name eines Child-Formulars über die Öffnen-Schaltfläche ausgewählt werden. In der Spalte *Parent field (Fix Field Value)* wird der Name des ID-Feldes der Parent-Tabelle eingetragen. Der Wert dieses Feldes wird an das Child-Formular beim Start und bei jeder Bewegung des Satzzeigers im Parent-Formular übergeben.

In der Spalte *Child field (Fix Field Name)* wird der dazugehörige Fremdschlüssel aus der Child-Tabelle eingetragen.

# Auswahllisten

## VFX – CPickfield Builder

VFX enthält mehrere Klassen für Auswahlfelder. Ein Auswahlfeld besteht aus einem Textfeld, einer Schaltfläche und einem schreibgeschützten Textfeld. In dem Textfeld kann ein Wert eingetragen werden. Beim Verlassen des Feldes wird überprüft, ob der eingegebene Wert in der Tabelle mit den Auswahlwerten enthalten ist. Falls nein, wird ein Auswahlformular gestartet. Im Auswahlformular kann der Anwender den gewünschten Datensatz auswählen. In einem schreibgeschützten Textfeld können weitere Informationen aus der Auswahltabelle angezeigt werden. Auf Wunsch kann dem Benutzer erlaubt werden neue Datensätze in der Auswahltabelle zu erfassen. Alle Eigenschaften des Auswahlfeldes können mit dem VFX – CPickField Builder gemacht werden.

Um den VFX – CPickField Builder aufzurufen, wählen Sie das Auswahllisten-Container-Steuerelement auf dem Formular, drücken die rechte Maustaste und wählen Builder.

The screenshot shows the 'VFX - CPickField Builder' dialog box. It has a title bar with a close button. The dialog is divided into several sections with labels and input fields:

- Pick Dialog Caption:** A text field containing 'Auswahl eines Kunden' and a button labeled 'Maintenance Form'.
- Pick Table Name:** A dropdown menu showing 'customers'.
- Pick Table Index Tag:** A dropdown menu showing 'customerid'.
- CPickField::txtField.ControlSource:** A dropdown menu showing 'orders.customerid'.
- CPickField::txtDesc.ControlSource:** A dropdown menu showing 'customers.customername'.
- Return Field Name (Code).Use STR() for Num. Fields.:** A text field containing 'TRANSFORM(customerid)'.
- Return Field Name (Description):** A text field containing 'customername'.
- Field List:** A text field containing 'customerid;customername'.
- Field Title:** A text field containing 'Kundennummer;Kundenname'.
- Column Width:** An empty text field.
- Sort Columns:** An empty text field.
- Always Force Column Width:** An unchecked checkbox.
- Always Force Sort Columns:** An unchecked checkbox.
- Format:** An empty text field.
- Input Mask:** A text field containing '999999999'.
- Status Bar Text:** A text field containing 'Kundenauswahl'.

At the bottom of the dialog are three buttons: 'OK', 'Apply', and 'Cancel'.

Auf der Seite *Pick Field* stehen die folgenden Optionen zur Verfügung:

**Pick Dialog Caption.** Geben Sie die Überschrift für das Auswahllisten-Formular ein. In diesem Formular kann der Benutzer einen Wert auswählen.

**Maintenance Form.** Wenn der Benutzer den gewünschten Datensatz in dem Auswahllisten-Formular nicht findet, möchten Sie dem Benutzer vielleicht die Möglichkeit geben, das normale Bearbeitungsformular aufzurufen. Geben Sie hier den Namen für das Bearbeitungsformular ein. Es wird aufgerufen, wenn der Benutzer auf die Schaltfläche *Bearbeiten...* im Auswahllisten-Formular drückt.

**Pick Table Name.** Wählen Sie den Namen der Tabelle oder Ansicht aus der Sie den Wert auswählen oder überprüfen möchten. Hier können Sie zwischen allen Tabellen oder Ansichten aus der Datenumgebung wählen.

**Pick Table Index Tag.** Dieser Indexschlüssel wird zur Überprüfung der Benutzereingabe verwendet.

**CPickField::txtField.ControlSource.** Dies ist die Datenquelle für das Eingabetextfeld.

**CPickField::txtDesc.ControlSource.** Wählen Sie die Datenquelle für das Beschreibungsfeld des Auswahllisten-Steuerelementes. Stellen Sie sicher, dass Sie eine korrekte Beziehung zu der Tabelle herstellen aus der diese Datenquelle stammt. Andernfalls wird dieses Steuerelement nicht den gewünschten Wert anzeigen, wenn Sie den Datensatzzeiger in Ihrem Formular bewegen.

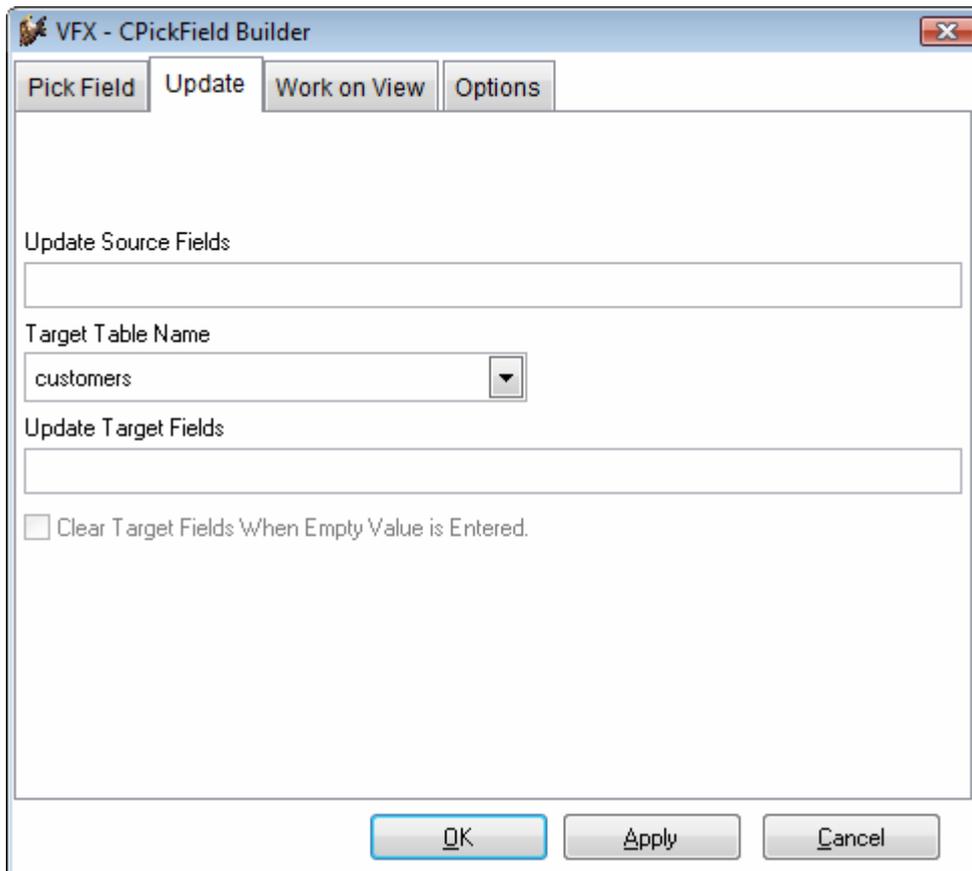
**Return Field Name (Code).** Geben Sie den Namen des Feldes (aus der Tabelle oder Ansicht der Auswahlliste) ein, das den ausgewählten Wert enthält. Geben Sie keinen Aliasnamen ein, weil Tabellen für Auswahllisten mit einem temporären Namen geöffnet werden.

**Return Field Name (Description).** Geben Sie den Namen des Feldes (aus der Tabelle oder Ansicht der Auswahlliste) ein, das den Wert mit der Beschreibung enthält. Geben Sie keinen Aliasnamen ein, weil Tabellen für Auswahllisten mit einem temporären Namen geöffnet werden.

**Format.** Der VFX – CPickField Builder übernimmt diese Eigenschaft aus dem Datenbank-Container.

**Input Mask.** Der VFX – CPickField Builder übernimmt diese Eigenschaft aus dem Datenbank-Container.

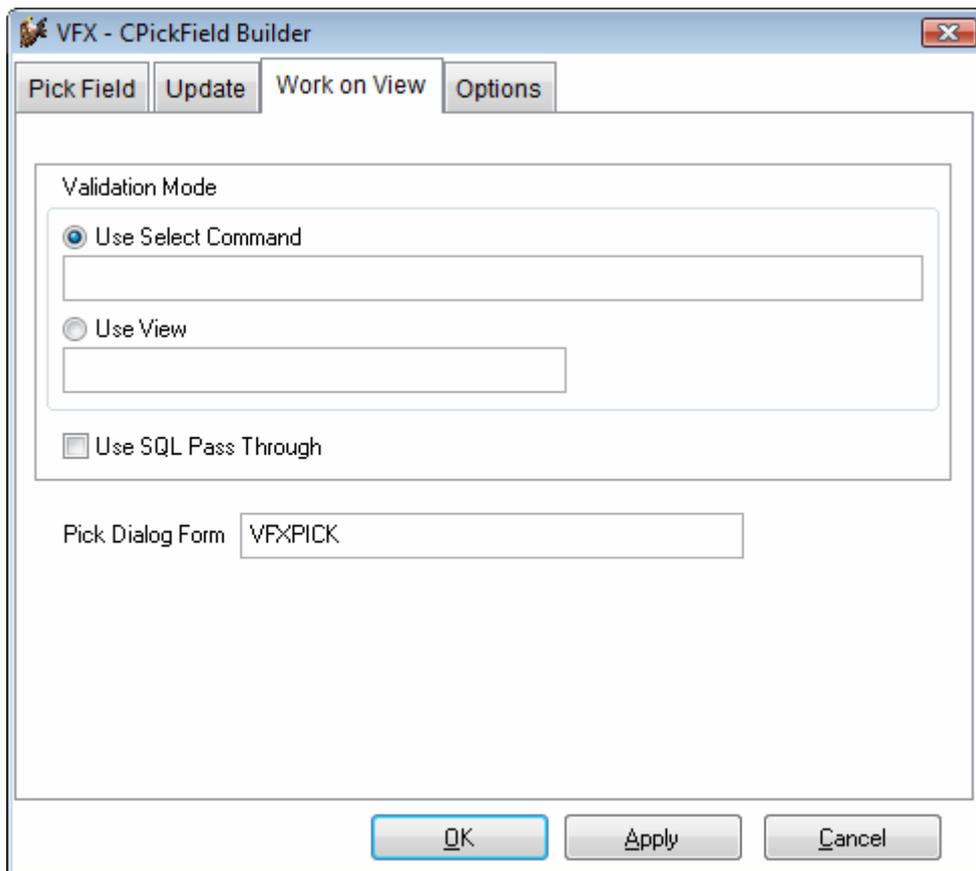
**Status Bar Text.** Der VFX – CPickField Builder übernimmt diese Eigenschaft aus dem Datenbank-Container.



**Update Source Fields.** Hier können sie Felder aus der Auswahlliste eingeben, deren Werte in die Bearbeitungstabelle übernommen werden sollen. Wenn Sie mehrere Werte eingeben, so müssen diese durch Semikolon getrennt werden.

**Target Table Name.** Wählen sie die Zieltabelle aus. Normalerweise ist dies die Bearbeitungstabelle des Formulars.

**Update Target Fields.** Weisen sie die Zielfelder zu. Wenn Sie mehrere Werte eingeben, so müssen diese durch Semikolon getrennt werden.

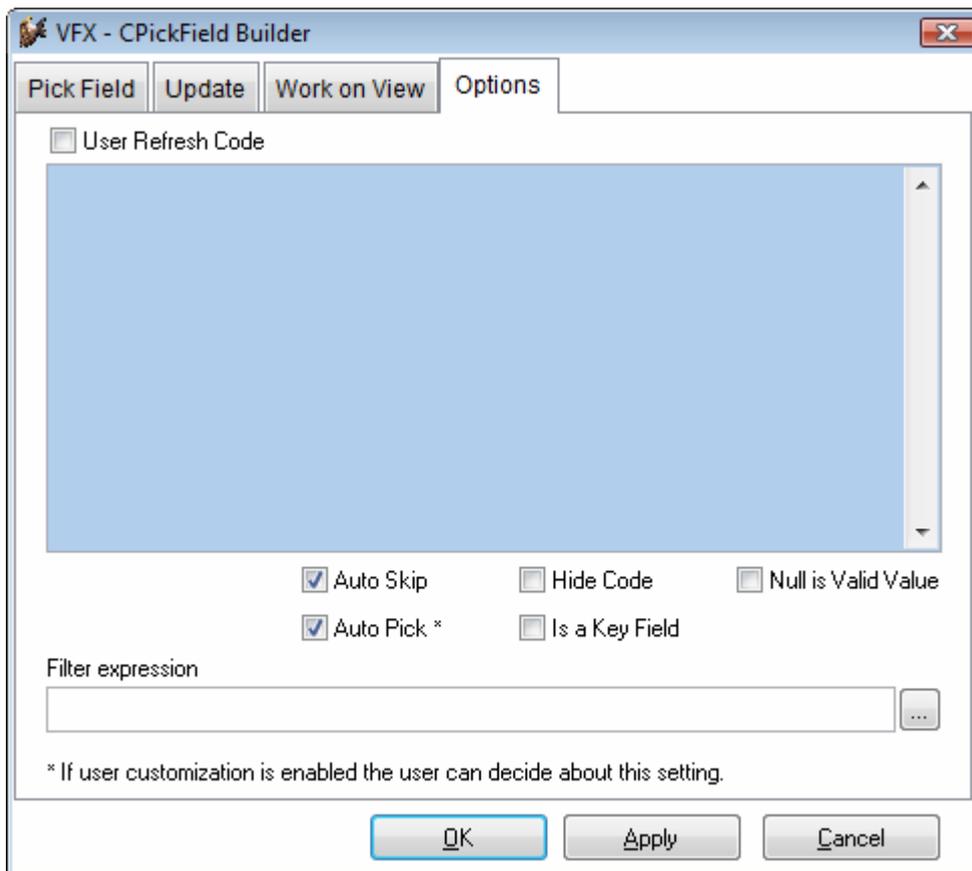


**Use Select Command:** Wahlweise kann ein Select-Befehl oder eine Ansicht zur Überprüfung der Benutzereingabe verwendet werden. Wenn Sie einen Select-Befehl verwenden, muss durch eine Where-Klausel sichergestellt sein, dass maximal ein Wert zurückgegeben wird. Beispiel: „*select customer\_id from lv\_customer where customer\_id = trim(this.txtField.Value)*”

**Use View:** Wahlweise kann ein Select-Befehl oder eine Ansicht zur Überprüfung der Benutzereingabe verwendet werden. Wenn Sie eine Ansicht verwenden, geben Sie hier den Namen der Ansicht ein. Die Where-Klausel der Ansicht muss sicherstellen, dass maximal ein Wert zurückgegeben wird.

**Use SQL Pass Through:** Wenn Sie dieses Kontrollkästchen markieren, wird der in der Ansicht enthaltene Select-Befehl von VFX ausgelesen und per SQL Pass Through an die Remote-Datenquelle gesendet.

**Pick Dialog Class:** Hier kann eine eigene Klasse für das Auswahllisten-Steuerelement verwendet werden. Beachten Sie, dass die Klasse von der Klasse *CPickField* abgeleitet sein muss.



**User Refresh Code.** Manchmal benötigen Sie speziellen Code in der *Refresh()*-Methode des Auswahllisten-Containers.

**Auto Skip.** Markieren Sie diese Option, wenn Sie automatisch zum nächsten Steuerelement springen wollen, nachdem Sie einen Wert aus der Auswahlliste ausgewählt haben. Dadurch wird die *CPickField*-Eigenschaft *UseTab* auf *.T.* gesetzt.

**Auto Pick.** Markieren Sie diese Option, wenn Sie automatisch die Auswahlliste aufrufen wollen, wenn der Benutzer einen falschen Wert eingegeben hat. Dadurch wird die *CPickField*-Eigenschaft *AutoPick* auf *.T.* gesetzt.

**Hide Code.** Markieren Sie diese Option, wenn Sie das Eingabefeld in der Auswahlliste verstecken wollen. Dadurch wird die *CPickField*-Eigenschaft *HideCode* auf *.T.* gesetzt. Der Benutzer kann keinen Wert eingeben, sondern nur aus der Auswahlliste auswählen.

**Is a Key Field.** Markieren Sie diese Option, wenn Sie dieses Auswahllistenfeld als Schlüsselfeld definieren wollen. Ein Schlüsselfeld ist nur zugänglich während Sie einen neuen Datensatz anlegen (so wie die Textfeld-Klasse *ckeyfield*). Dadurch wird die *CPickField*-Eigenschaft *IsKeyField* auf *.T.* gesetzt.

**OK.** Die eingestellten Optionen werden in das ausgewählte Auswahllisten-Objekt eingefügt.

**Apply.** Macht das gleiche wie *OK*, jedoch wird der VFX – CPickField Builder nicht beendet.

**Cancel.** Bricht die Arbeit mit dem VFX – CPickField Builder ab. Alle Eingaben werden verworfen.

Auch dieser Builder ist voll wieder verwendbar. Das bedeutet, dass Sie diesen Builder während des Entwicklungsprozesses beliebig oft verwenden können ohne die Eigenschaften zu verlieren, die Sie bereits eingestellt haben.

## VFX – CPickAlternate Builder

Ähnlich zum *CPickField*-Steuerelement kann die Klasse *CPickAlternate* verwendet werden um eine Benutzer-eingabe zu verifizieren. Es kann eine Auswahlliste aufgerufen werden, die dem Anwender erlaubt einen Wert aus einer Liste auszuwählen. Bei Verwendung der Klasse *CPickAlternate* wird der Primärschlüssel des ausgewählten Datensatzes in der Bearbeitungstabelle gespeichert während der Benutzer einen Wert aus einem anderen Feld aus der Auswahltabelle angezeigt bekommt.

Das *CPickAlternate*-Steuerelement ist einer Combobox zu bevorzugen, wenn aus einer Tabelle mit vielen Datensätzen ausgewählt werden soll. Der Einsatz ist auch sinnvoll, wenn der vom Anwender eingegebene Wert nicht dem Schlüssel der Auswahltabelle entspricht. Das Ziel dieser Klasse ist es dem Anwender eine einfach zu bedienende Schnittstelle zu geben, die es erlaubt ihm bekannte Werte einzugeben anstelle von vom Programm generierten Primärschlüsseln. Der vom Anwender eingegebene Wert wird verwendet um den dazugehörigen Datensatz in der Auswahltabelle zu finden. Wenn der gesuchte Datensatz gefunden ist, wird als Rückgabewert der Primärschlüssel an das *CPickAlternate*-Steuerelement zurückgegeben.

Diese Klasse basiert auf der Klasse *CPickField* und erbt alle ihre Eigenschaften und Methoden. Zusätzlich hat diese Klasse die neue Eigenschaft *cControlSourceInternalKey* in die der Name des Feldes der Bearbeitungstabelle mit dem Fremdschlüssel eingetragen wird. Dieser Fremdschlüssel entspricht dem Primärschlüssel aus der Auswahltabelle.

Mithilfe des VFX – CPickAlternate Builder können die Eigenschaften dieser Klasse einfach eingestellt werden.

The screenshot shows the 'VFX - CPickAlternate Builder' dialog box with the following settings:

- Pick Dialog Caption:** Kundenauswahl
- Maintenance Form:** CUSTOMERS.SCX
- Pick Table Name:** cacustomers
- Pick Table Index Tag:** customerna
- CPickAlternate::txtField.ControlSource:** cacustomers.customername
- CPickAlternate::txtDesc.ControlSource:** cacustomers.contactperson
- Return Field Name (Code).Use STR() for Num. Fields:** customername
- Return Field Name (Description):** contactperson
- Control Source Internal Key:** caorders.customerid
- Return Field Name (Internal Key):** TRANSFORM(customerid)
- Field List:** (empty)
- Field Title:** (empty)
- Column Width:** (empty)
- Sort Columns:** (empty)
- Always Force Column Width
- Always Force Sort Columns
- Format:** (empty)
- Input Mask:** (empty)
- Status Bar Text:** (empty)

Buttons at the bottom: OK, Apply, Cancel.

*Pick Table Name* – Hier kann der Name der Auswahltabelle aus einer der Datenquellen der Datenumgebung ausgewählt werden.

*Pick Table Index Tag* – Dies ist der Name des Indexschlüssels, der verwendet wird um in der Auswahltabelle zu suchen. Dieser Indexschlüssel entspricht dem Wert des Eingabefeldes.

*CPickAlternate.txtField.ControlSource* – Die Controlsource des Eingabefeldes. Dieses Feld muss aus der Auswahltabelle stammen.

*CpickAlternate.txtDesc.ControlSource* – Der Name des Beschreibungsfeldes. Der Wert wird nach der erfolgreichen Überprüfung der Benutzereingabe im Beschreibungsfeld angezeigt. Dieses Feld stammt ebenfalls aus der Auswahltabelle.

*Return Field Name (Code)* – Der Name des Feldes mit dem vom Anwender eingegebenen Wert aus der Auswahltabelle. In der Regel entspricht dieser Feldname dem Namen, der in *txtField.ControlSource* angegeben ist. Hier ist nur der Feldname ohne den Tabellennamen anzugeben. Der Wert dieses Feldes muss vom Typ „Zeichen“ sein. Gegebenenfalls ist der Wert mit TRANSFORM() in einen Zeichentyp umzuwandeln.

*Return Field Name (Description)* – Der Name des Feldes mit der Beschreibung, die aus der Auswahltabelle zurückgegeben wird. Es kann auch ein Ausdruck zurückgegeben werden. Der Wert wird im Beschreibungsfeld angezeigt. Der Wert muss vom Typ „Zeichen“ sein. Gegebenenfalls ist der Wert mit TRANSFORM() in einen Zeichentyp umzuwandeln.

*Return Field Name (Internal Key)* – Der Name des Feldes aus der Auswahltabelle, das den Primärschlüssel enthält. Über dieses Feld wird die Beziehung von der Bearbeitungstabelle zur Auswahltabelle in der Datenumgebung hergestellt.

*Control Source Internal Key* – Der Name des Feldes aus der Bearbeitungstabelle, das den Primärschlüssel enthält. Dieses Feld enthält den Fremdschlüssel aus der Auswahltabelle.

Die weiteren Seiten des VFX – CPickalternate Builder entsprechen den Seiten des VFX – CPickfield Builder.

## VFX – CPickTextBox Builder

Visual Extend bietet einen Builder, um leistungsfähige Auswahltextfelder für Childgrids zu erstellen.

Um den VFX – CPickTextBox Builder aufzurufen, wählen Sie die Spalte im Grid, die das Auswahltextfeld erhalten soll und wählen Sie den Menüpunkt VFX Power Builder aus dem VFX-Menü:

Der VFX – CPickTextBox Builder ist in der Bedienung dem normalen VFX – CPickField Builder ähnlich und ist ebenfalls voll wieder verwendbar:

The screenshot shows the 'VFX - CPickTextBox Builder' dialog box with the 'TextBox Field' tab selected. The 'Update' and 'Options' tabs are also visible. The dialog contains the following fields and options:

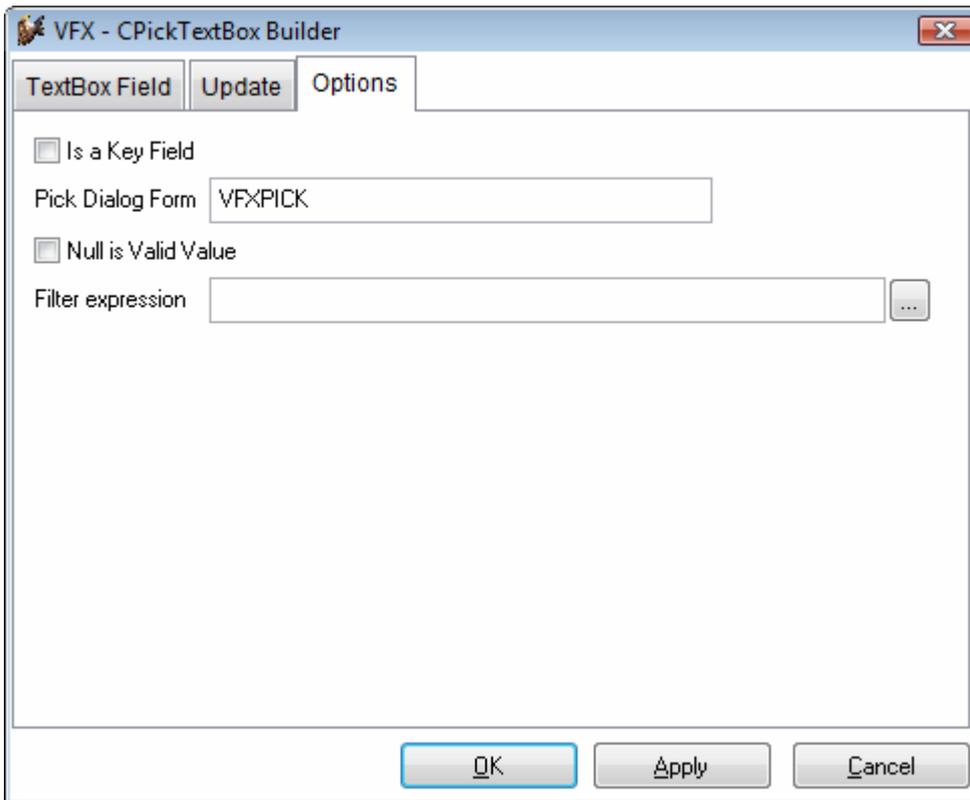
- Pick Dialog Caption:** Maintenance Form
- Pick Table Name:** products
- Pick Table Index Tag:** productcod
- Return Field Name (Code).Use STR() for Num. Fields:** productcode
- Field List:** productcode;productname
- Field Title:** Artikel;Bezeichnung
- Column Width:** (empty)
- Sort Columns:** (empty)
- Always Force Column Width
- Always Force Sort Columns
- Format:** (empty)
- Input Mask:** XXXXXX
- Status Bar Text:** (empty)

Buttons: OK, Apply, Cancel

The screenshot shows the 'VFX - CPickTextBox Builder' dialog box with the 'Update' tab selected. The dialog contains the following fields and options:

- Update Source Fields. Use ; as delimiter:** productprice;productid
- Target Table Name:** orderdetails
- Update Target Fields:** price;productid
- Clear Target Fields When Empty Value is Entered.

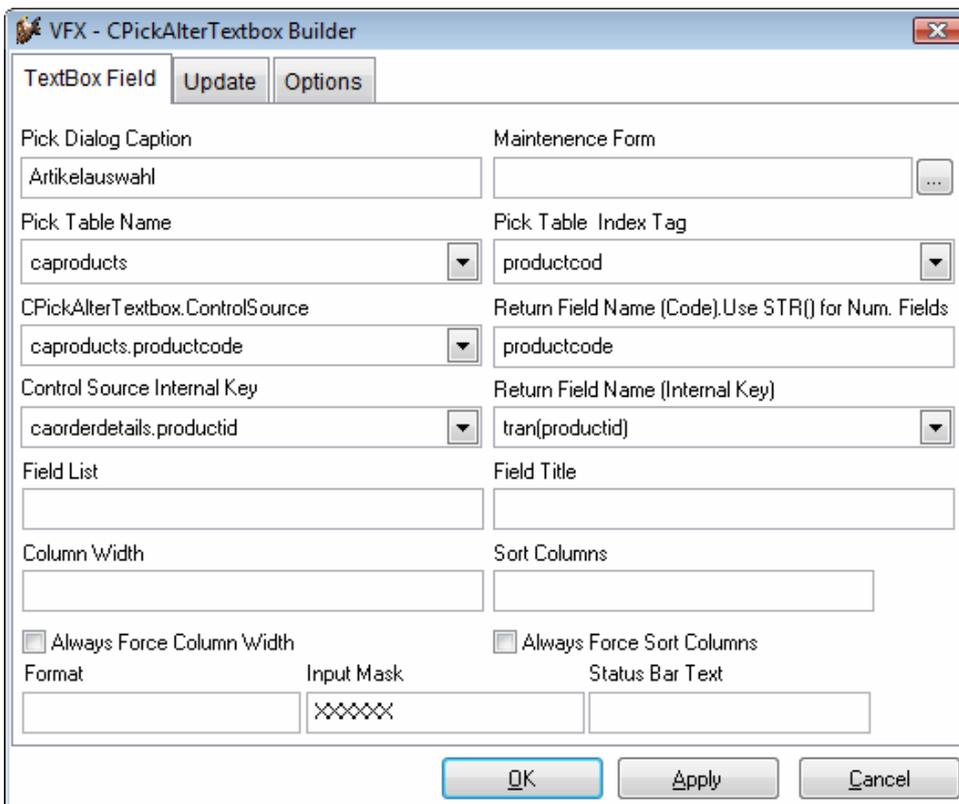
Buttons: OK, Apply, Cancel



## VFX – CPickAlterTextbox Builder

Die CPickalterTextbox bietet die Funktionalität der Klasse CPickTextbox, basiert jedoch auf Primärschlüsseln, ähnlich der Klasse CPickaltermate.

Dementsprechend vereint der VFX – CPickAlterTextbox Builder die Felder des VFX – CPickTextBox Builders mit denen des VFX – CPickaltermate Builders.



VFX - CPickAlterTextbox Builder

TextBox Field Update Options

Update Source Fields. Use ; as delimiter  
productprice

Target Table Name  
caorderdetails

Update Target Fields  
price

Clear Target Fields When Empty Value is Entered.

OK Apply Cancel

VFX - CPickAlterTextbox Builder

TextBox Field Update Options

Is a Key Field

Pick Dialog Form VFXPICK

Null is Valid Value

Filter expression

OK Apply Cancel

## VFX – Combo Pick List Builder

Diese Klasse dient zur einfachen Erstellung von Auswahllisten. Es können Auswahllisten erstellt werden, die nicht auf einer eigenen Tabelle basieren müssen.

Die Klasse *CComboPicklist* benutzt zwei VFX-Systemtabellen: *Vfxpdef.dbf* und *Vfxplist.dbf*.

Die Tabelle *Vfxpdef.dbf* enthält die Beschreibungen der Auswahllisten. Für jede Auswahlliste gibt es einen Datensatz. Zu jeder Auswahlliste kann es Code geben, der ausgeführt wird, wenn der Benutzer eine Auswahl trifft. Dieser Code wird bei jeder Auswahl ausgeführt. In der Tabelle *Vfxplist.dbf* kann zu jedem Eintrag ein Code zugeordnet werden.

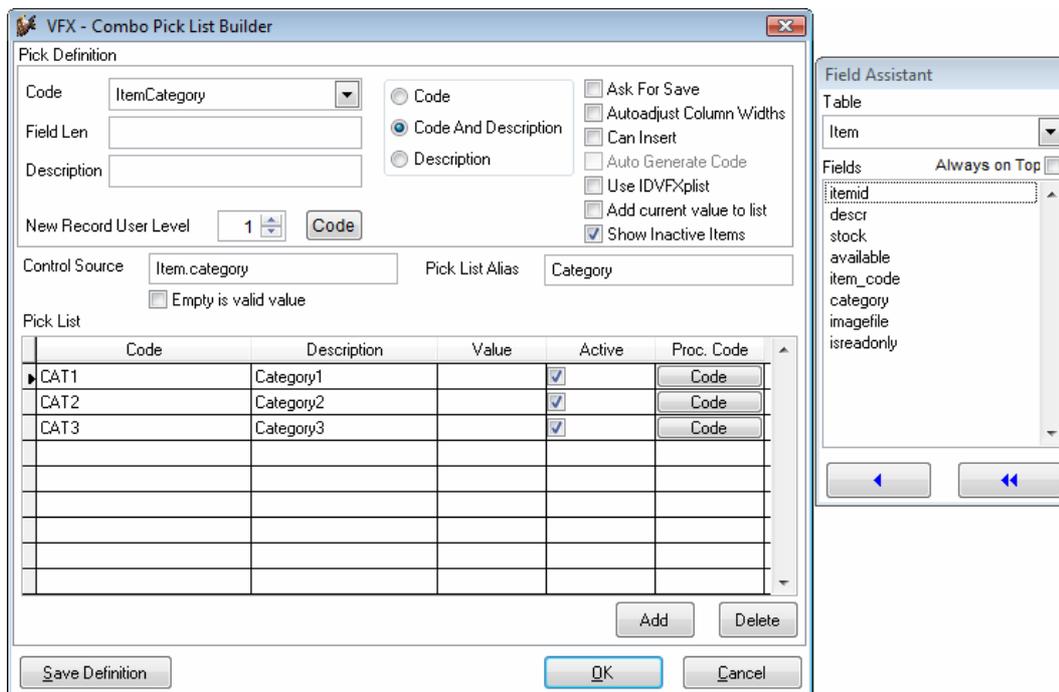
Die Tabelle *Vfxplist.dbf* enthält die auswählbaren Einträge. Das Feld *Picklist* enthält den Fremdschlüssel und zeigt auf einen korrespondierenden Datensatz in der Tabelle *Vfxpdef.dbf*. Die Felder *Code* und *Describe* enthalten Werte, die in der Auswahlliste angezeigt werden. Abhängig von der Einstellung der Auswahlliste in der Tabelle *Vfxpdef.dbf* kann nur die *Code*-Spalte oder die *Code*-Spalte und die *Describe*-Spalte angezeigt werden. Im Feld *Proccode* kann zu einem Eintrag Code eingetragen werden, der ausgeführt wird, wenn dieser Eintrag ausgewählt wird.

Für jede Verwendung der Klasse *CComboPicklist* kann eingestellt werden, ob neue Datensätze hinzugefügt werden dürfen, und welche Berechtigungsstufe Benutzer haben müssen, um neue Datensätze hinzufügen zu dürfen.

Für die Klasse *CComboPicklist* können zwei Code-Blöcke in Tabellenfeldern hinterlegt werden. In der Tabelle *Vfxpdef.dbf* ist es das Memofeld *ProcCode* und in der Tabelle *Vfxplist.dbf* ist es das Memofeld *ProcCode*.

Der Code aus dem Feld *Vfxpdef.ProcCode* wird zur Laufzeit immer dann ausgeführt, wenn der Wert in der Combobox geändert wird. Der Code aus dem Feld *Vfxplist.ProcCode* ist einem bestimmten Eintrag zugeordnet und wird immer dann ausgeführt, wenn dieser Eintrag ausgewählt wird.

Für jeden Eintrag in der Tabelle *Vfxplist.dbf* kann eingestellt werden, ob es sich um einen aktiven Eintrag handelt. Durch dieses Verfahren brauchen Einträge, die zeitweise nicht zur Auswahl stehen sollen, nicht aus der Tabelle gelöscht werden. Um einen Eintrag zu deaktivieren muss der Wert im Feld *Active* auf *.F.* gesetzt werden.



Die Klasse *CComboPicklist* sowie die Tabellen *Vfxpdef.dbf* und *Vfxplist.dbf* können mit dem VFX – Combo Pick List Builder bearbeitet werden.

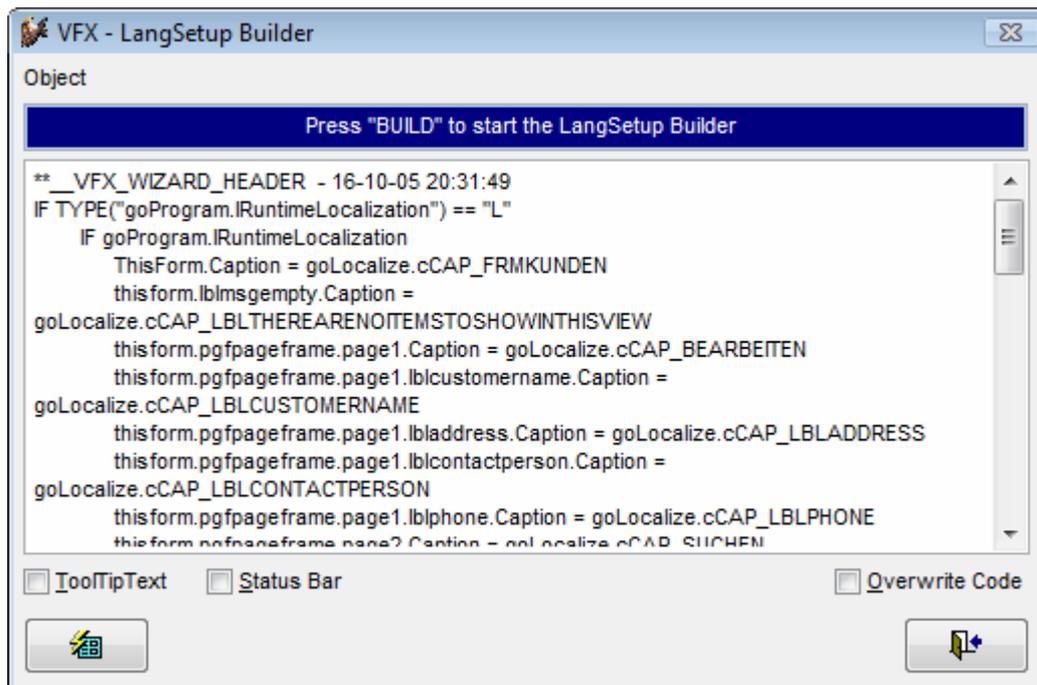
Für die *CComboPicklist* müssen die Controlsources und der Alias für die *Rowsource* angegeben werden. Wenn der Alias für die *Rowsource* bereits in der Datenumgebung vorhanden ist, fragt der Builder, ob dieser Alias verwendet werden soll, oder ob eine weitere Instanz dieses Cursors der Datenumgebung hinzugefügt werden soll. Wenn der Alias für die *Rowsource* nicht in der Datenumgebung gefunden werden kann, wird das entsprechende Cursor-Objekt vom Builder automatisch der Datenumgebung hinzugefügt und die Eigenschaften werden eingestellt.

# Lokalisierung

## VFX – LangSetup Builder

Der VFX – LangSetup Builder automatisiert die Erstellung des in der *LangSetup()*-Methode benötigten Codes. Sie brauchen diesen Code, wenn Sie Ihre Anwendung in mehr als einer Sprache erstellen wollen. Das Ziel dieses Builders ist es, aus dem Formular für alle Bezeichnungen, Tooltip-Texte und Statuszeilenmeldungen Datensätze anzulegen und diese in der Tabelle *Vfxmsg.dbf* zu speichern. Nach diesem Vorgang können Sie den VFX – Message Editor benutzen, um die Texte in verschiedene Sprachen zu übersetzen.

Um den VFX – LangSetup Builder aufzurufen, öffnen Sie zunächst das Formular dessen Bezeichnungen, Tooltip-Texte und Statuszeilenmeldungen Sie analysieren lassen möchten. Wählen Sie den Menüpunkt *Form, LangSetup Builder* aus dem VFX-Menü.



Markieren Sie die Kontrollkästchen entsprechend den gewünschten Optionen. Klicken Sie auf die Schaltfläche *Build* um den Code für die *LangSetup()*-Methode generieren zu lassen.

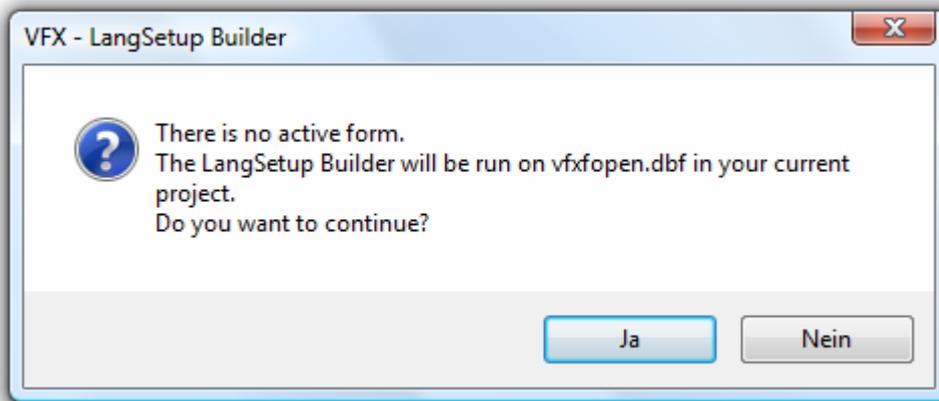
Nach der Generierung sehen Sie den Code, der für die *LangSetup()*-Methode erzeugt wurde. Wenn Sie das Kontrollkästchen *Overwrite Code* markieren, wird der erzeugte Code in die *LangSetup()*-Methode des aktuell in der Entwicklungsansicht geöffneten Formulars geschrieben. Der Bezeichnungscode wird in der VFX-Meldungstabelle *Vfxmsg.dbf* gespeichert. Hier können Sie die Texte bearbeiten und in andere Sprachen übersetzen.

In der Include-Datei *Vfxdef.h* ist die *ID\_Language*-Konstante definiert, die die aktuelle Sprache Ihrer Anwendung angibt.

```
...  
#define ID_LANGUAGE "ENG"  
...
```

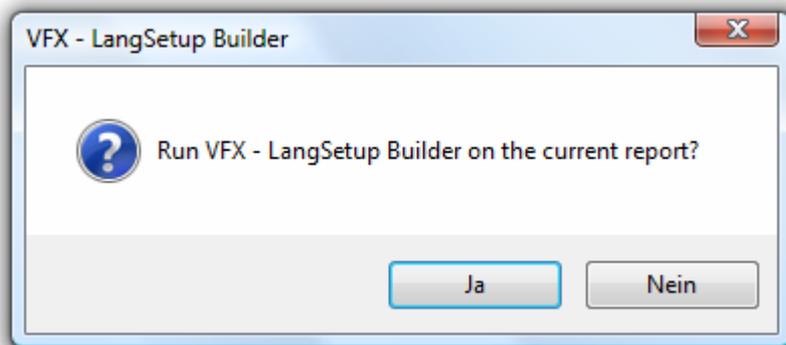
Wenn Sie Ihre Anwendung mit dem VFX-Anwendungs-Assistenten anlegen, wird die Anwendung in der Sprache angelegt, die im VFX-Anwendungs-Assistenten angegeben ist. Wenn Ihre Anwendung in eine andere Sprache übersetzt werden soll, ändern Sie die Konstante *ID\_Language*.

Wenn der VFX – LangSetup Builder gestartet wird während kein Formular im VFP Formular-Designer geöffnet ist, erscheint die folgende Messagebox.



Wenn die Frage mit Ja beantwortet wird, werden die Einträge in der Tabelle Vxfopen lokalisiert.

Wenn der VFX – LangSetup Builder gestartet wird während ein Bericht im VFP Berichts-Designer geöffnet ist, erscheint die folgende Messagebox.



Wenn die Frage mit Ja beantwortet wird, werden alle Bezeichnungen im Bericht lokalisiert.

## VFX – Language Management

Dieser Builder verwaltet die verfügbaren Sprachen für eine Anwendung, die mit Lokalisierung zur Laufzeit arbeitet. Die Informationen über die zur Verfügung stehenden Sprachen sind in *VFXLanguage.dbf* gespeichert.

Bei der Generierung einer neuen Anwendung werden alle Sprachen, die mit VFX geliefert werden, in das neue Projekt kopiert. Hier können Sprachen hinzugefügt, aber auch gelöscht werden. Die mit VFX gelieferten Sprachen können jedoch nicht gelöscht werden, sondern nur als nicht aktiv gekennzeichnet werden. Als nicht aktiv gekennzeichnete Sprachen stehen zur Laufzeit der Anwendung nicht in den Comboboxen zur Sprachauswahl im Anmeldedialog und in der Standard-Symbolleiste zur Verfügung.

Active	Blank	Default	Language	Abbreviation	LCID	IconFile	Compliance List	Font Char Set	Lang Message ID
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="radio"/>	English	ENG	09	bitmap\languages\FLGUK.ICO	09,0c,07,10,0a,02,08,13,16,19,05	0	LNG_ENGLISH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	French	FRE	0c	bitmap\languages\FLGFRAN.ICO	09,0c,07,10,0a,13,16	0	LNG_FRENCH
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	German	GER	07	bitmap\languages\FLGGERM.ICO	09,0c,07,10,0a,13,16,05	0	LNG_GERMAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Italian	ITA	10	bitmap\languages\FLGITALY.ICO	09,0c,07,10,0a,13,16	0	LNG_ITALIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Spanish	ESP	0a	bitmap\languages\FLGSPAIN.ICO	09,0c,07,10,0a,13,16	0	LNG_SPANISH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Bulgarian	BUL	02	bitmap\languages\FLGBUL.ICO	02	1	LNG_BULGARIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Greek	GRE	08	bitmap\languages\FLGGRE.ICO	08	161	LNG_GREEK
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Dutch	NL	13	bitmap\languages\FLGNETH.ICO	09,0c,07,10,0a,02,13,16,19,05	0	LNG_DUTCH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Portuguese	POR	16	bitmap\languages\FLGPORT.ICO	09,0c,07,10,0a,13,16,05	0	LNG_PORTUGUESE
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Russian	RU	19	bitmap\languages\FLGRUS.ICO	19	1	LNG_RUSSIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Czech	CZE	05	bitmap\languages\FLGCZE.ICO	05	238	LNG_CZECH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Finnish	FIN	0b	bitmap\languages\FLGFIN.ICO	0b	0	LNG_FINNISH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Polish	PL	15	bitmap\languages\FLGPOL.ICO	15	0	LNG_POLISH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Turkish	TR	1F	bitmap\languages\FLGTR.ICO	1F	162	LNG_TURKISH
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Albanian	ALB	1C	bitmap\languages\flgalb.ico	1C	0	LNG_ALBANIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Swiss	CHD	07	bitmap\languages\flgchd.ico	09,0c,07,10,0a,13,16,05	0	LNG_SWISS
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Romanian	RO	18	bitmap\languages\flgrom.ico	18	0	LNG_ROMANIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Slovak	SVK	1B	bitmap\languages\flgsvk.ico	1B	238	LNG_SVK
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Estonian	EST	25	BITMAP\LANGUAGES\FLGEST.ICO	07,0c,09,10,0a,0b,25	186	LNG_ESTONIAN
<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	ChineseSimp	CHS	04	BITMAP\LANGUAGES\FLGCHS.ICO	04,11	0	LNG_CHINESESIMPL

In der Spalte *Language* wird die Bezeichnung einer Sprache eingetragen, so wie sie in der Sprachauswahl-Combobox zur Laufzeit angezeigt werden soll.

In der Spalte *Abbreviation* wird der Name des Feldes in der Tabelle *Vfxmsg.dbf* eingetragen. Aus diesem Feld werden zur Laufzeit die Texte der gewählten Sprache gelesen.

Die Spalte *LCID* enthält den *Locale Identifier* der Sprache. Dies ist ein in Windows definierter Wert und wird für die Regionaleinstellungen verwendet.

In der Spalte *IconFile* kann der Name zu einer Icondatei ausgewählt werden. Das Icon sollte eine Flagge zur Veranschaulichung einer Sprache anzeigen. Das Icon wird in der Sprachauswahl-Combobox zur Laufzeit angezeigt.

Die Spalte *Compliance List* enthält eine durch Komma separierte Liste *Locale Identifiers* von Sprachen, die zur aktuellen Sprache kompatibel sind. Diese Liste enthält Werte von Regionaleinstellungen, die geeignet sind, die aktuelle Sprache korrekt anzuzeigen.

Die Spalte *Lang Message Id* enthält den Wert des Feldes *MessageID* aus der Tabelle *Vfxmsg.dbf*, die die lokalisierten Texte der aktuellen Sprache für die Sprachauswahl-Combobox enthält.

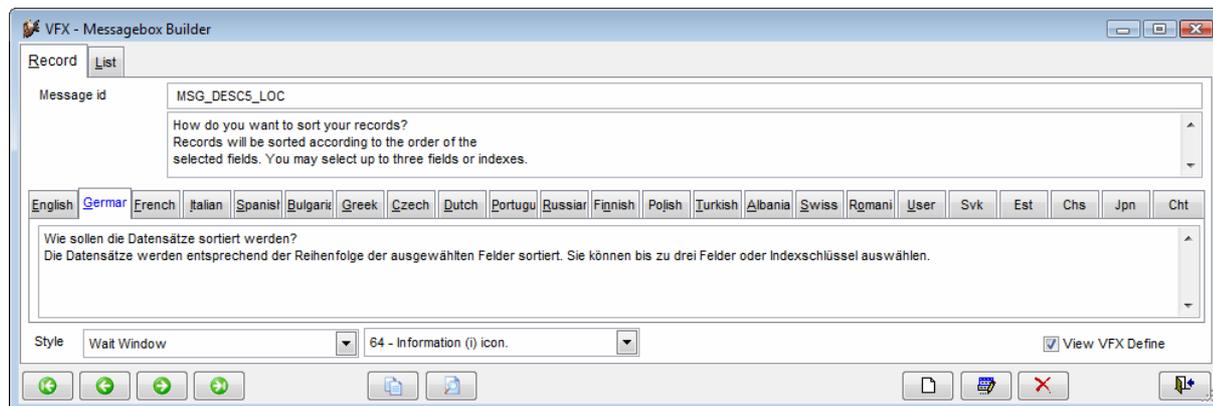
Die oben beschriebenen Einstellungen können für mit VFX gelieferte Sprachen nicht geändert werden. Diese Einstellungen können nur für neu hinzugefügte Sprachen bearbeitet werden. Eine neue Sprache kann über die Schaltfläche *Add Language* hinzugefügt werden. Eine so hinzugefügte Sprache wird in der Sprachauswahl-Combobox angezeigt. Für eine neu angelegte Sprache wird automatisch ein Feld in der Tabelle *Vfxmsg.dbf* mit Bezeichnung aus der Spalte *Abbreviation* angelegt.

Um eine neu hinzugefügte Sprache verwenden zu können, müssen alle Texte aus der Tabelle *Vfxmsg.dbf* in die neue Sprache übersetzt werden.

## VFX – Messagebox Builder

Ein nützliches Werkzeug zur Erstellung von Messageboxen in verschiedenen Sprachen ist der VFX – Messagebox Builder. Die Texte der Messagebox werden in der Tabelle *Vfxmsg.dbf* gespeichert. Der Befehl zur Anzeige der Messagebox wird in die Zwischenablage kopiert und kann von dort in den eigenen Programm Quelltext übernommen werden. Dabei wird nicht der Text selbst, sondern eine Konstante als Parameter übergeben. Die Include-Dateien mit den Werten der Konstanten in der gewünschten Sprache werden mit dem VFX – Message Editor erstellt.

Um den VFX – Messagebox Builder aufzurufen, wählen Sie den Menüpunkt *Form, MessageBox Builder* aus dem VFX-Menü.



Klicken Sie auf die Schaltfläche *neu* um eine neue Messagebox anzulegen. Tragen Sie dann im Feld *Message id* eine eindeutige Bezeichnung für die Messagebox ein. Im Seitenrahmen können Sie für jede benötigte Sprache den Text hinterlegen.

In der Zeile *Style* wählen Sie gewünschten Typ der Messagebox aus. Es kann zwischen verschiedenen Symbolen und Schaltflächen auf der Messagebox ausgewählt werden.

Durch einen Klick auf die Schaltfläche *Test it!* wird die Messagebox in der Vorschau angezeigt.

Kopieren Sie den vom VFX – Messagebox Builder erstellten Code mit der Schaltfläche *Copy code to clipboard* in die Zwischenablage. Aus der Zwischenablage kann der Code in einem beliebigen Programmteil eingefügt werden.

Der VFX – Messagebox Builder legt für jeden Eintrag einen Datensatz in der Tabelle *Vfxmsg.dbf* an.

Auf der Seite *List* erhalten Sie eine Übersicht über alle vorhandenen Datensätze:

---

**Tip:** Auch wenn Sie keine mehrsprachigen Anwendungen erstellen, können Sie den VFX – Messagebox Builder einsetzen.

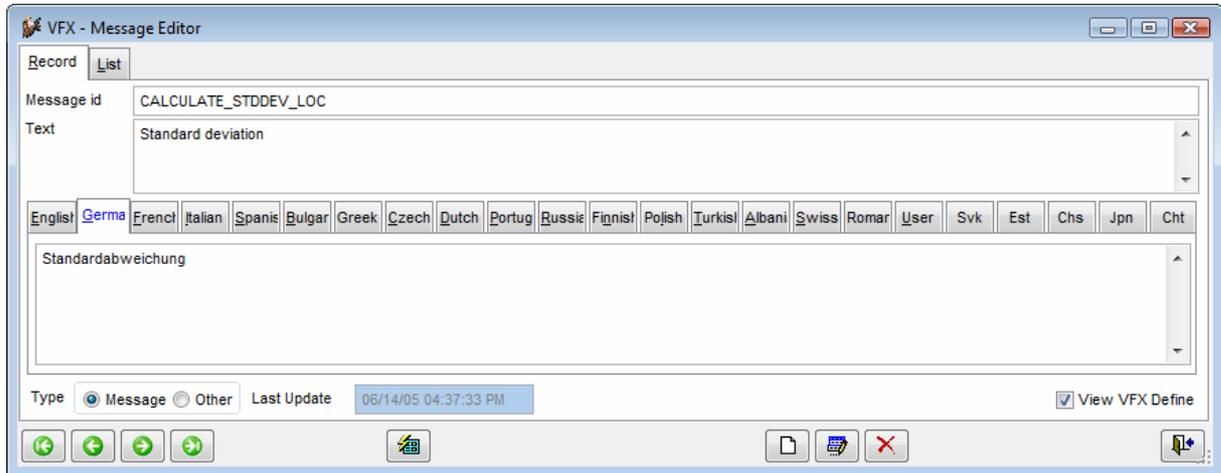
---

## VFX – Message Editor

Die Werte aller von VFX verwendeten Konstanten stehen in der freien Tabelle *Vfxmsg.dbf*. Für jede Sprache ist ein Memofeld mit dem Text vorhanden. Mit dem VFX – Message Editor können diese Texte bearbeitet werden.

Der VFX – Message Editor ist der Zentrale Ort um, alle Bezeichnungen, Meldungen, Tooltip-Texte und Statuszeilenmeldungen zu verwalten und in andere Sprachen zu übersetzen. Aus dem VFX – Message Editor heraus können Sie die benötigten Include-Dateien (*User.txt.h* und *Usermsg.h*) erstellen.

Um den VFX – Message Editor aufzurufen, wählen Sie den Menüpunkt *Form, Message Editor* aus dem VFX-Menü.



Klicken Sie auf die Schaltfläche *Make Include File* um eine Include-Datei in der im Seitenrahmen angezeigten Sprache zu erstellen. Die Include-Dateien werden in einem Ordner mit der Bezeichnung der jeweiligen Sprache unterhalb des Include-Ordners Ihres Projektes gespeichert.

Nach der Erstellung Ihrer Include-Dateien müssen Sie diese nur noch in den *INCLUDE*-Ordner Ihres Projektes kopieren, wie im Kapitel *Erstellen mehrsprachiger Anwendungen* beschrieben ist.

---

**Tipp:** Sie können Ihre eigenen Konstanten mit den erzeugten Konstanten aus der Tabelle *Vfxmsg.dbf* mischen. Schreiben Sie Ihre Konstanten vor oder nach dem VFX-Header bzw. -Footer.

---

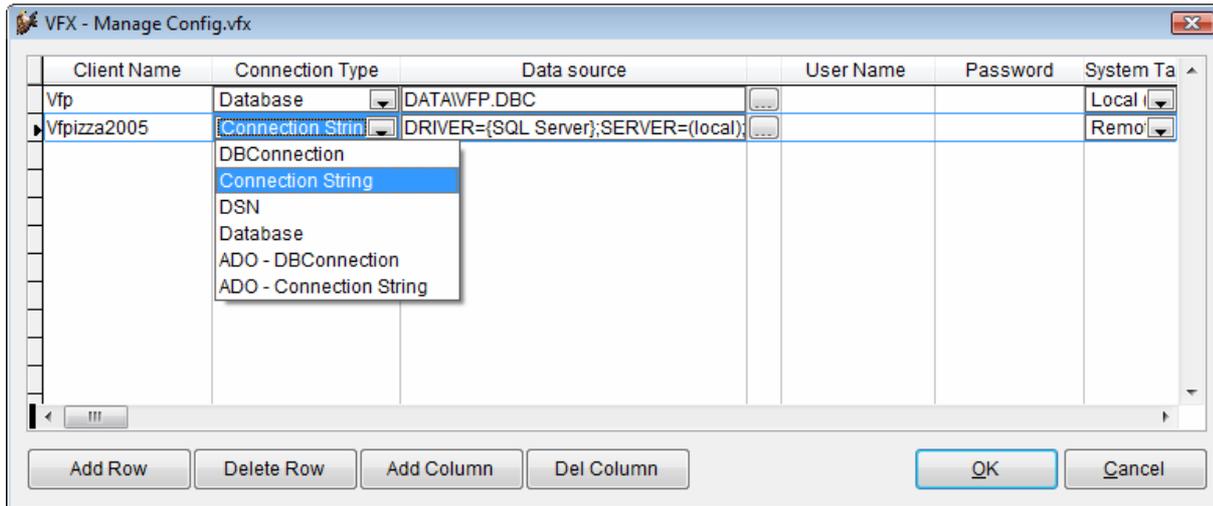
# Daten

## VFX – Manage Config.vfx

Der Dialog VFX – Manage Config.vfx dient zur Verwaltung der Datenzugriffsinformationen.

Wenn mit einer VFP Datenbank in der Entwicklungsumgebung gearbeitet wird, steht hier in der Regel nur der Verweis auf die Datenbank im Ordner Data.

Wenn mit einer Remote Datenbank gearbeitet wird, steht hier die Verbindungszeichenfolge, mit der die Verbindung zur Remote Datenbank hergestellt wird.



Die VFX-Tabellen können jetzt wahlweise auch in einer eigenen Remote Datenbank gespeichert werden und stehen so auf Wunsch Datenbank-Übergreifend zur Verfügung. Alle Verbindungstypen stehen auch für die Datenbank mit den VFX-Tabellen zur Verfügung.

Der Dialog VFX – Manage Config.vfx steht auch in Endanwendungen zur Verfügung.

## VFX – Cursor Adapter Wizard

Der VFX – CursorAdapter Wizard erstellt zu jeder Tabelle einer Datenbank eine CursorAdapter-Klasse. Mit Hilfe der so generierten CursorAdapter kann zum Beispiel aus Formularen auf die Daten zugegriffen werden. Der CursorAdapter Wizard kann eine beliebige von VFP unterstützte Datenquelle als Grundlage zur Generierung von CursorAdaptoren verwenden.

Die generierten CursorAdapter-Klassen können nach der Generierung durch den Wizard im VFP Klassen-Designer weiter bearbeitet werden. Es sollte insbesondere in Erwägung gezogen werden, welche Parameter für die CursorAdapter sinnvoll eingesetzt werden können.

Standardmäßig basieren diese CursorAdapter-Klassen auf der Klasse *CAppDataAccess* und werden in der Klassenbibliothek *Appl.vcx* gespeichert. Die Klassenbibliothek und die Basisklasse können bei Bedarf im Wizard geändert werden.

Der Wizard führt den Entwickler durch drei Schritte.

## 1. Auswahl der Datenquelle

VFX - Cursor Adapter Wizard - VFPIZZA.PJX

Native  
DATAWFP.DBC

ODBC

Use DSN  
DSN: Excel Files User Name: Password:

Generate SQL Connection String  
Server Name:  Use Trusted Connection  
User Name:  
Password:

Use connection string

Click on next to proceed.

Cancel < Back Next > Finish

Diese Datenquelle wird nicht die Datenquelle der Anwendung. Diese Datenquelle wird vom Wizard nur zur Erstellung der CursorAdapter verwendet. Die zur Laufzeit verwendete Datenquelle wird aus der Datei *Config.vfx* gelesen. Auf diesem Weg können für verschiedene Kunden unterschiedliche Datenquellen verwendet werden.

## 2. Auswahl der Klassen und Klassenbibliotheken

VFX - Cursor Adapter Wizard - VFPIZZA.PJX

Class Library c:\uwe\wfx100demos\wfpizza100\lib\appl.vcx

Parent Class Name cappdataaccess

Destination Class Library c:\uwe\wfx100demos\wfpizza100\lib\appl.vcx

Replace existing classes

Click on next to proceed.

Cancel < Back Next > Finish

Wenn die Option *Generate SQL Connection String* gewählt wird, muss im zweiten Schritt zunächst eine Datenbank vom gewählten SQL Server gewählt werden.

In diesem Schritt werden die verwendete CursorAdapter-Basisklasse und die Klassenbibliothek ausgewählt, in der die CursorAdapter gespeichert werden sollen.

Die Standardwerte sind:

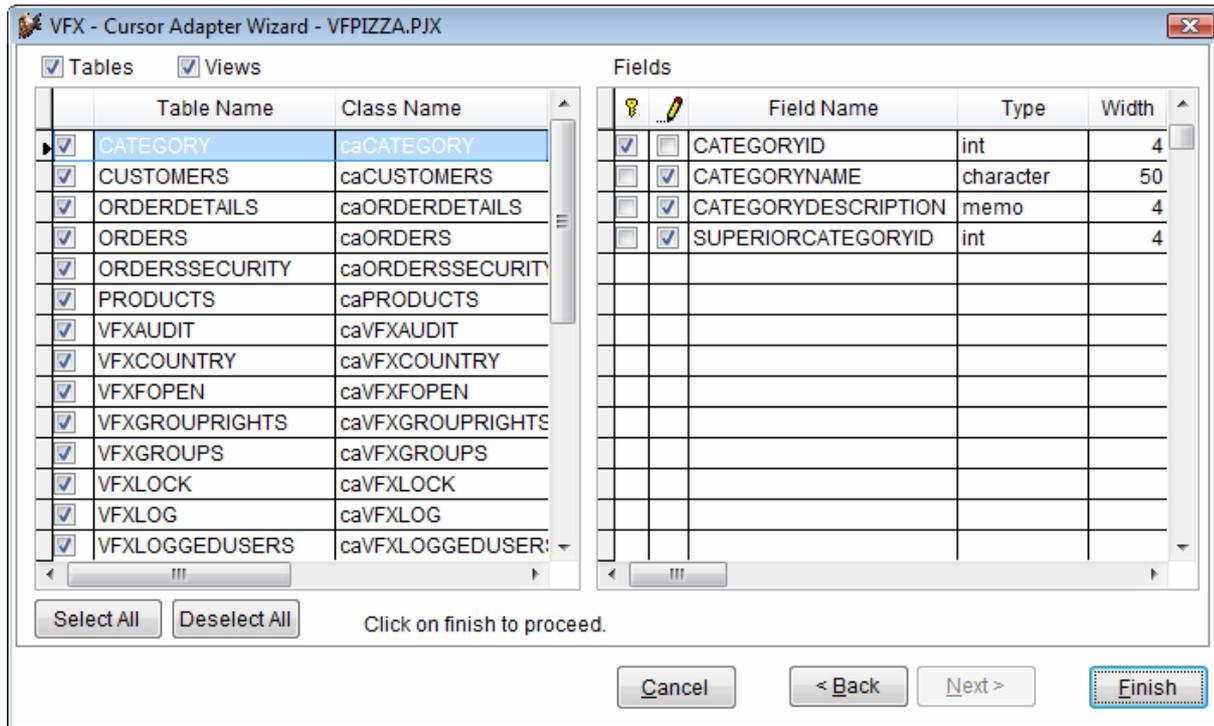
**Class Library:** *Appl.vcx*

**Parent Class Name:** *CAppDataAccess*

**Destination Class Library:** *Appl.vcx*

Wahlweise können existierende Klassen in der Zielklassenbibliothek überschrieben werden, wenn eine Markierung im Kontrollkästchen *Replace existing classes* gesetzt wird.

### 3. Auswahl der Tabellen



Der letzte Schritt zeigt Listen aller Tabellen und Felder für die CursorAdapter erstellt werden sollen. Beim Bewegen des Satzeigers in der Tabellenliste auf der linken Seite werden auf der rechten Seite die dazugehörigen Felder angezeigt.

Schlüsselfelder aus den Tabellen sind standardmäßig automatisch als Schlüsselfelder für die zu erstellenden CursorAdapter markiert. Alle anderen Felder sind standardmäßig als aktualisierbar markiert.

Als Ergebnis erstellt der VFX – CursorAdapter Wizard eine CursorAdapter-Klasse für jede Tabelle aus der ausgewählten Datenbank. Bei jedem CursorAdapter werden die Eigenschaften *CursorSchema*, *Tables*, *SelectCmd*, *KeyFieldList*, *UpdatableFieldList* und *UpdateNameList* vom Wizard eingestellt.

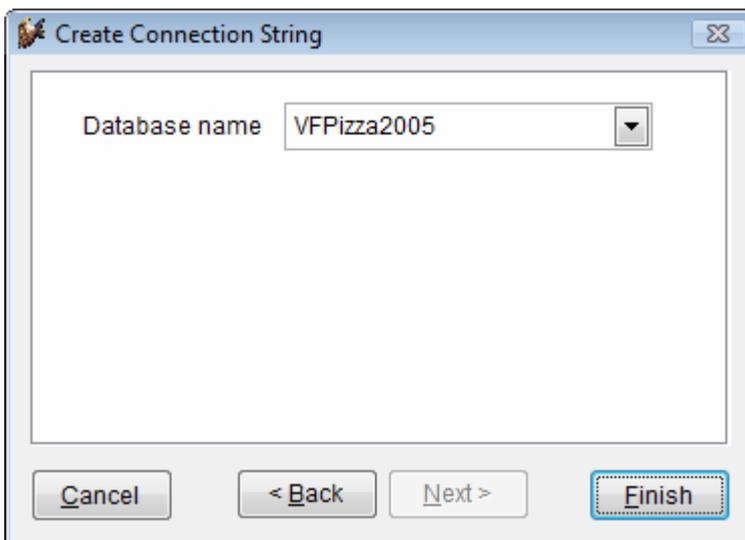
## VFX – Connectionstring Wizard

Der VFX – ConnectionString Wizard steht auch im VFX – CursorAdapter Wizard zur Verfügung.



Server Name: (local) ▼  
 Use Trusted Connection  
User Name: sa  
Password:   
Cancel < Back Next > Finish

Nach Eingabe der Verbindungsdaten und Auswahl der Datenbank wird ein Connectionstring generiert.



Database name VFPizza2005 ▼  
Cancel < Back Next > Finish

## VFX – Metadata Wizard

Der VFX – Metadata Wizard hilft Ihnen Metadaten aus Ihrer aktuell benutzten SQL Server-Datenbank zu erstellen. Die Metadaten können zur Aktualisierung der Datenbank beim Kunden verwendet werden.

Wahlweise kann die Verbindung aus einer VFP-Datenbank ausgelesen werden um die Verbindung zu einem SQL Server herzustellen oder der SQL Server kann manuell ausgewählt werden.

Der Metadata Wizard erstellt die Tabelle *Datadict.dbf*. Dies ist eine freie Tabelle, in der die Struktur der SQL Server Datenbank inklusiv Constraints, benutzerdefinierten Datentypen, Regeln, Ansichten und gespeicherten Prozeduren gespeichert wird. Der Metadata Wizard durchsucht das aktive Projekt nach Verbindungen und analysiert die Struktur der Datenbank. Wenn die Tabelle *Datadict.dbf* an die Kunden weitergegeben wird, wird die Struktur der dortigen Datenbank aktualisiert. Dabei wird wieder die bestehende Verbindung zum Zugriff auf die Datenbank verwendet.

Der VFX – Metadata Wizard befindet sich nur noch aus Kompatibilitätsgründen mit älteren VFX Versionen im VFX Menü der VFX – Metadata Wizard wird automatisch beim Erstellen einer Exe Datei ausgeführt, wenn sich in der Datei Config.vfx ein Verweis auf eine Remote Datenbank befindet. Es ist daher nicht mehr erforderlich den VFX – Metadata Wizard manuell auszuführen.

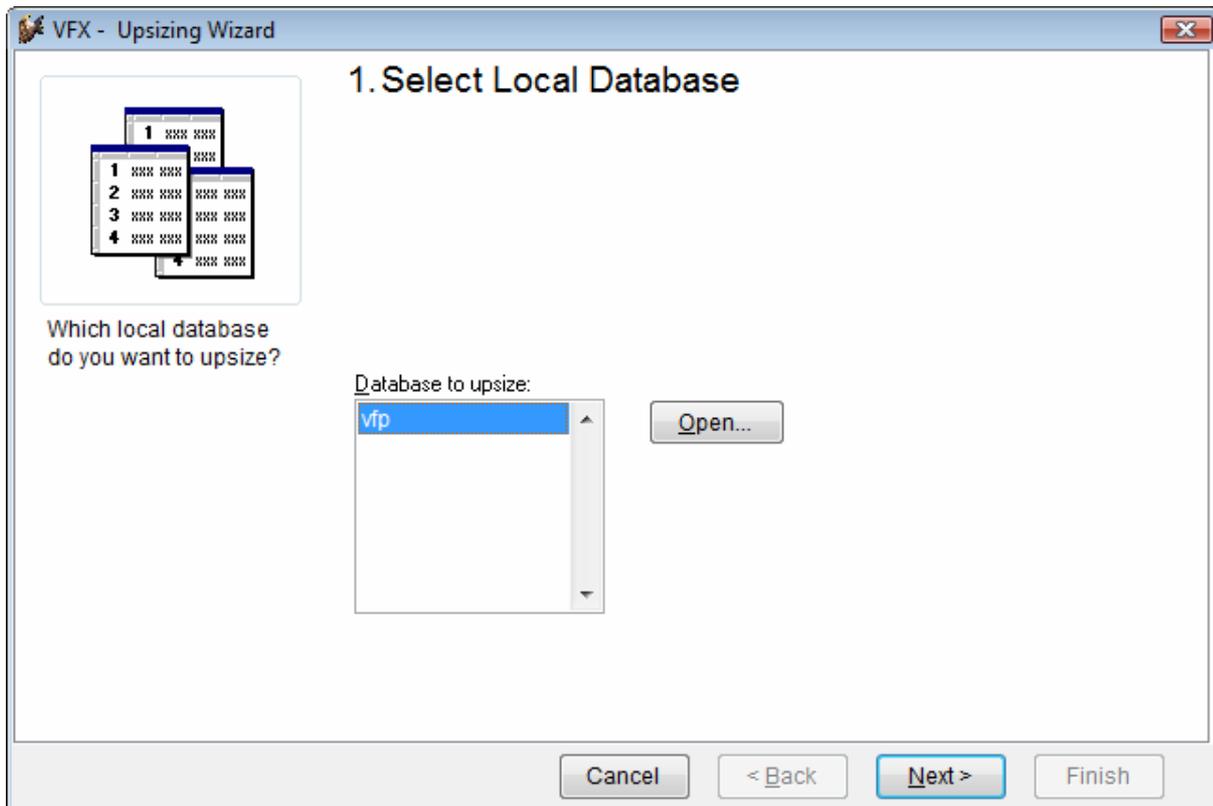
## VFX – Upsizing Wizard

Der VFX – Upsizing Wizard ermöglicht es eine vorhandene VFP-Datenbank auf einen SQL Server zu portieren. Dabei wird die Struktur so genau wie möglich auf dem SQL Server abgebildet. Die Daten werden in die neue SQL Server-Datenbank übernommen. Auch die Portierung von Ansichten wird unterstützt.

Der VFX – Upsizing Wizard führt den Entwickler durch sechs Schritte:

### 1. Auswahl der lokalen Datenbank

Im ersten Schritt wird die lokale Datenbank ausgewählt, die auf den SQL Server portiert werden soll.



Es wird eine Liste der zurzeit geöffneten Datenbanken angezeigt. Wenn die zu portierende Datenbank nicht geöffnet ist, kann sie hier über die Schaltfläche *Open* geöffnet werden.

## 2. Ziel

In diesem Schritt wird die Verbindung zum SQL Server angegeben. Es kann eine vorhandene Verbindung aus einem DBC verwendet werden. Es kann aber auch eine vorhandene DSN-Verbindung oder eine vorhandene Verbindungszeichenfolge verwendet werden.

VFX - Upsizing Wizard

**2. Destination** Which data source do you want to upsize your database to?

Use Database connections

Connections

ODBC

Use DSN

DSN  User Name  Password

Generate SQL Connection String

Server Name  (local)  Use Trusted Connection

User Name

Password

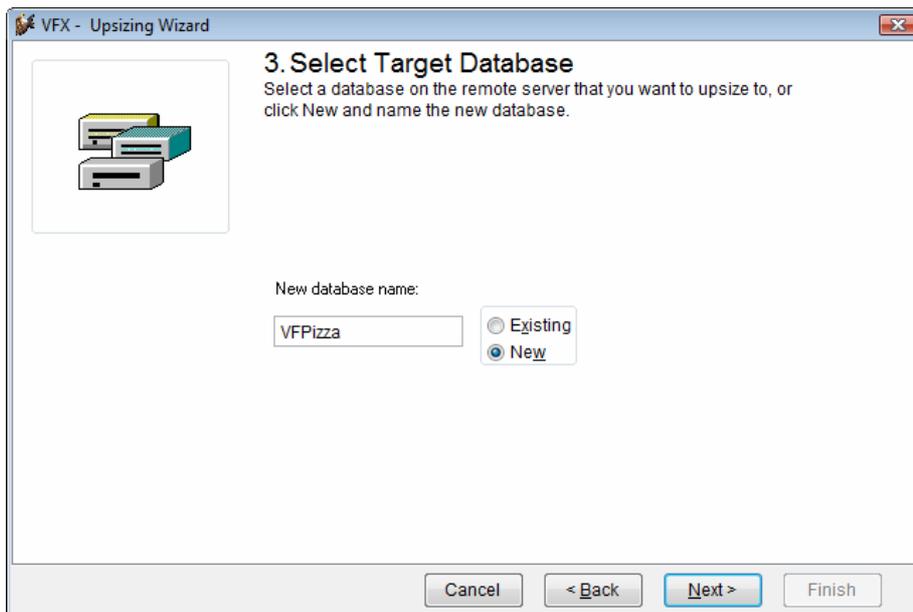
Use connection string

Cancel < Back Next > Finish

Mit dem Wizard für Verbindungszeichenfolgen kann ebenfalls eine Verbindung hergestellt werden. Hierbei wird standardmäßig der lokal installierte SQL Server vorgeschlagen. Wenn der eingegebene Benutzername mit dem Kennwort nicht zu einer erfolgreichen Anmeldung führt, wird automatisch versucht eine vertrauenswürdige Verbindung mit den Windows-Anmeldedaten herzustellen. In der Regel sind auf dieser Seite des Assistenten also keine Eingaben erforderlich.

### 3. Eingabe des Datenbanknamens

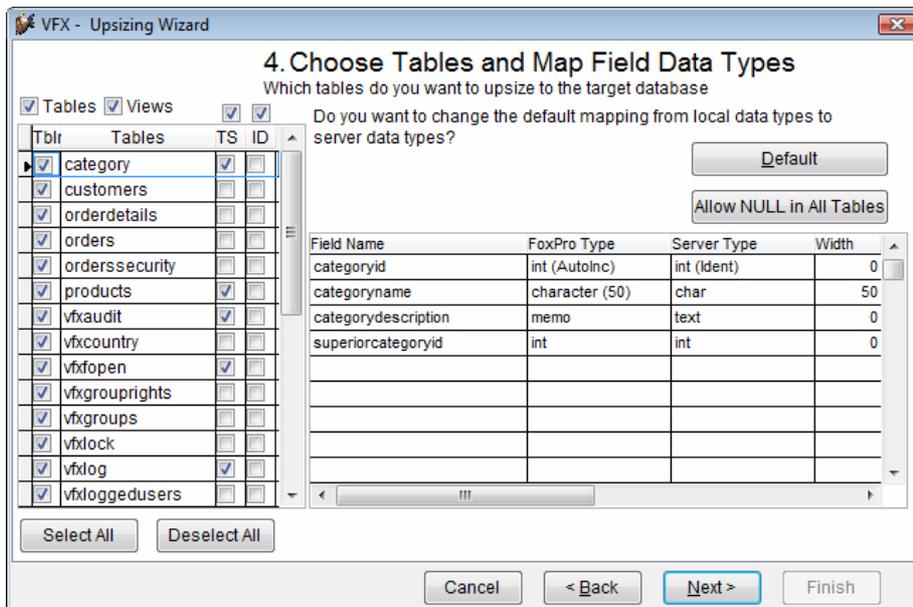
Wenn die Option *New* ausgewählt ist, kann der Name der neuen Datenbank eingegeben werden. Der Name muss ein gültiger Name für eine remote Datenbank sein.



Wenn die Option *Existing* ausgewählt wird, wird eine Liste der auf dem SQL Server vorhandenen Datenbanken angezeigt. Wählen Sie die Datenbank aus, die vom VFX – Upsizing Wizard aktualisiert werden soll.

### 4. Auswahl der Tabellen und Zuordnung der Datentypen

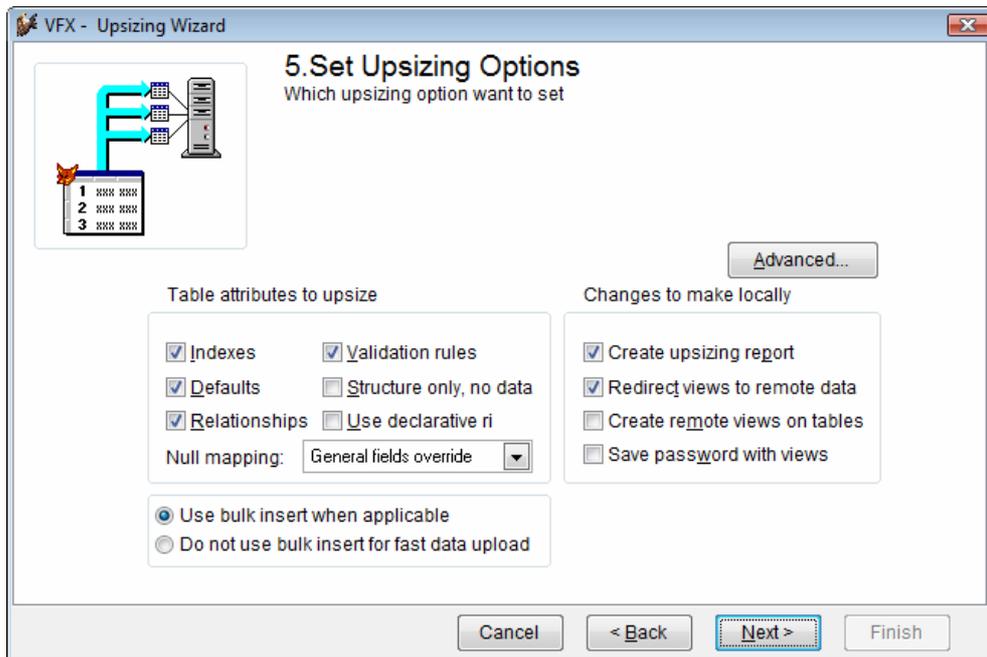
Hier kann ausgewählt werden welche Tabellen aus der VFP-Datenbank auf den SQL Server portiert werden sollen. Standardmäßig sind alle Tabellen zur Portierung markiert. Jeder Tabelle kann ein Timestamp-Feld sowie ein ID-Feld hinzugefügt werden. Der VFX – Upsizing Wizard fügt Tabellen mit Memo-Feldern standardmäßig ein Timestamp-Feld hinzu. Tabellen, die kein Primärschlüsselfeld enthalten, wird automatisch ein ID-Feld hinzugefügt. Diese Einstellungen können bei Bedarf je Tabelle geändert werden.



Zur selektierten Tabelle werden die Struktur in der VFP-Datenbank sowie die Struktur in der zu erstellenden SQL Server-Datenbank angezeigt. In der Regel wird für jeden Feldtyp eine sinnvolle Portierung vorgenommen. Bei Bedarf kann die Zuordnung des Datentyps hier geändert werden.

## 5. Portierungsoptionen

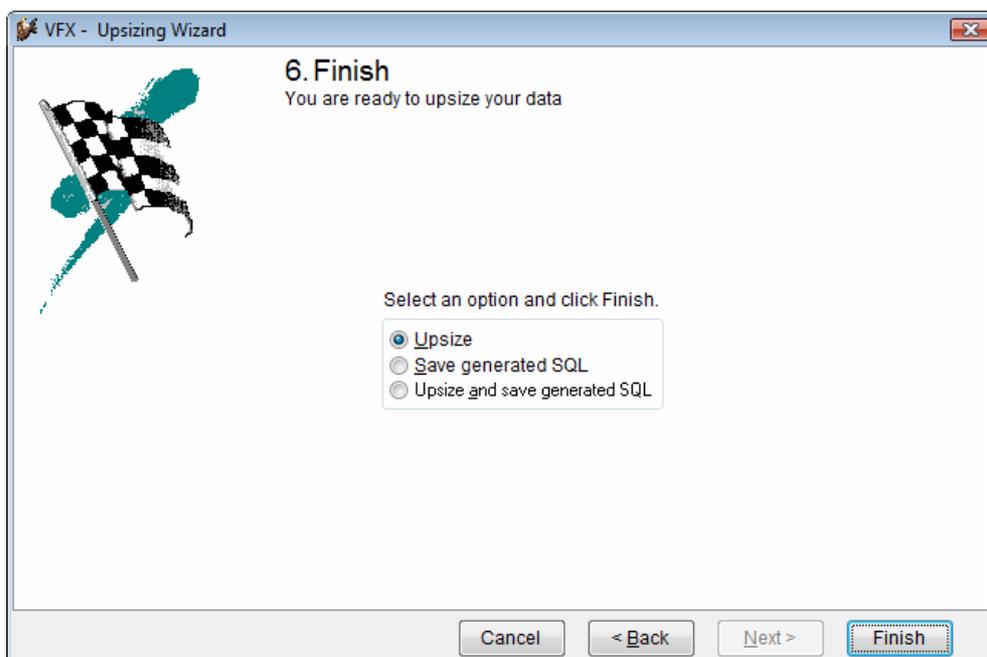
Standardmäßig werden die Strukturen von Tabellen sowie die Daten portiert. Es können auch Indexschlüssel, Standardwerte, Beziehungen (RI Constraints) und Validierungsregeln portiert werden. In der Combobox *Null mapping* kann eingestellt werden, ob Nullwerte erlaubt sind. Diese Option hilft sicherzustellen, dass Einfüge- und Aktualisierungsvorgänge erfolgreich durchgeführt werden können.



In diesem Schritt kann insbesondere auch eingestellt werden, ob ein Bericht über die Portierung erstellt werden soll. Der Bericht wird in ein neues VFP-Projekt eingefügt. Aus dem Bericht sind Probleme bei der Portierung ersichtlich.

## 6. Fertig

In diesem Schritt kann eingestellt werden, wie die Portierung durchgeführt werden soll.



Es kann eine der Optionen gewählt werden:

**Upsize**– Führt die Portierung wie oben beschrieben durch.

**Save generated SQL**– Generiert SQL Befehle, die für die Portierung erforderlich sind. Durch Ausführung dieser Befehle kann die eigentliche Portierung zu einem späteren Zeitpunkt durchgeführt werden.

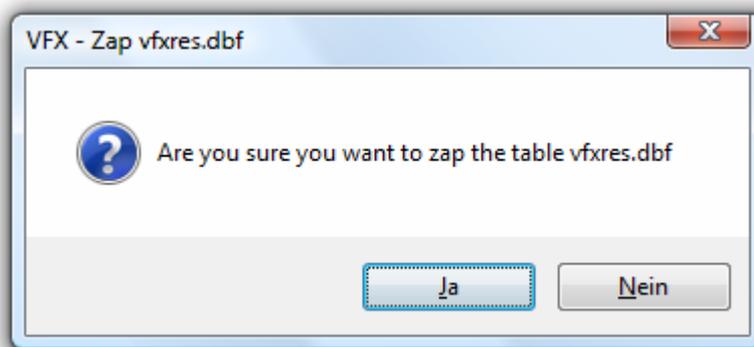
**Upsize and save generated SQL**– Führt die Portierung wie oben beschrieben durch und generiert zusätzlich SQL Befehle, um die Portierung zu einem späteren Zeitpunkt wiederholen zu können.

Es ist eine gute Idee vor der Portierung eine Datensicherung durchzuführen. Während der Portierung werden Tabellen und lokale Ansichten aus der VFP-Datenbank umbenannt, um Tabellen und remote Ansichten mit den gleichen Namen in der SQL Server-Datenbank erstellen zu können.

Der VFX – Upsizing Wizard erlaubt es Felder vom Typ *Date* und *Datetime* mit leeren Werten in eine SQL Datenbank zu portieren. Wenn diese Felder in der SQL Datenbank den Zustand *NULL* erlauben, wird *NULL* in die SQL Datenbank geschrieben. Wenn der Zustand *NULL* nicht erlaubt ist, wird *01.01.1900* in die SQL Datenbank geschrieben.

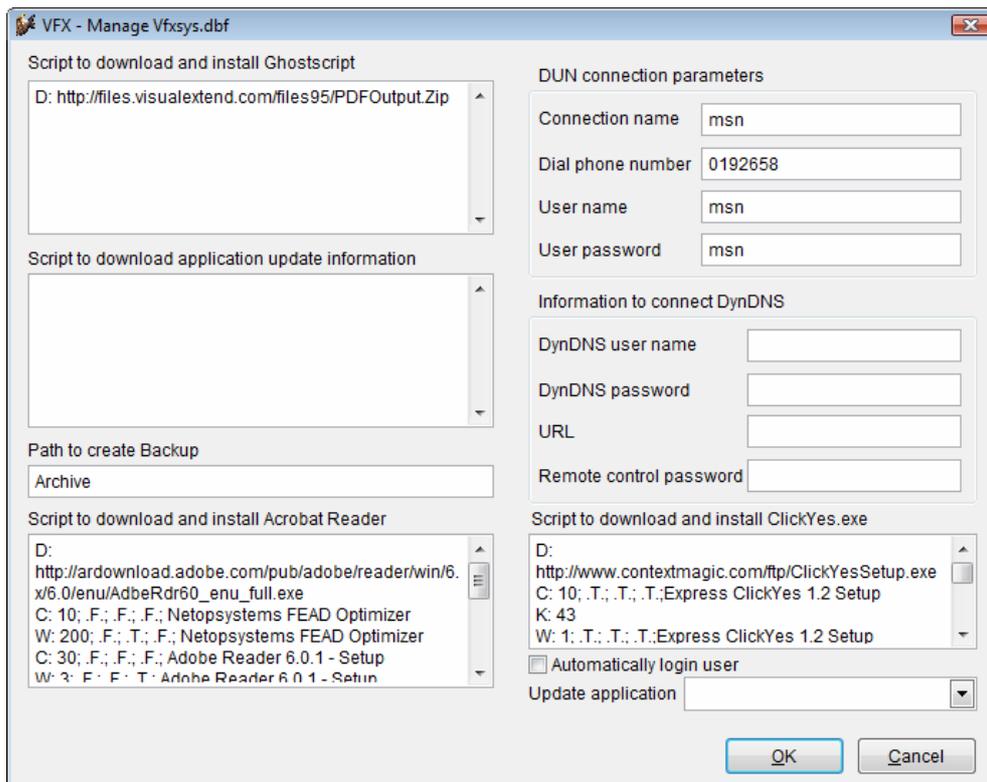
## Zap Vfxres

Mit dieser Funktion kann der Inhalt der Tabelle Vfxres.dbf in der Entwicklungsumgebung gelöscht werden. Damit werden alle Benutzereinstellungen zurückgesetzt, was bei Tests in der Entwicklungsumgebung sinnvoll sein kann.



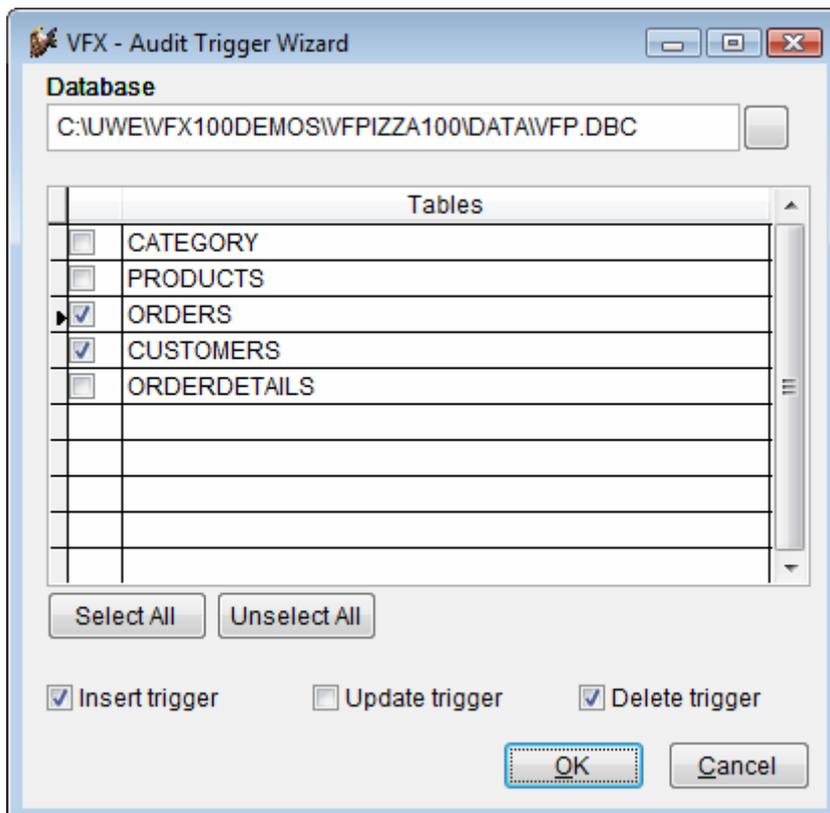
## Manage Vfxsys.dbf

Verwaltung der Tabelle Vfxsys.dbf mit teilweise verschlüsseltem Inhalt.



## VFX – Audit Trigger Wizard

Der AuditTrigger-Wizard erstellt Ihnen automatisch alle Trigger für den Audit-Trail für einzelne oder alle Tabellen eines Datenbankcontainers zwecks Nachverfolgung.





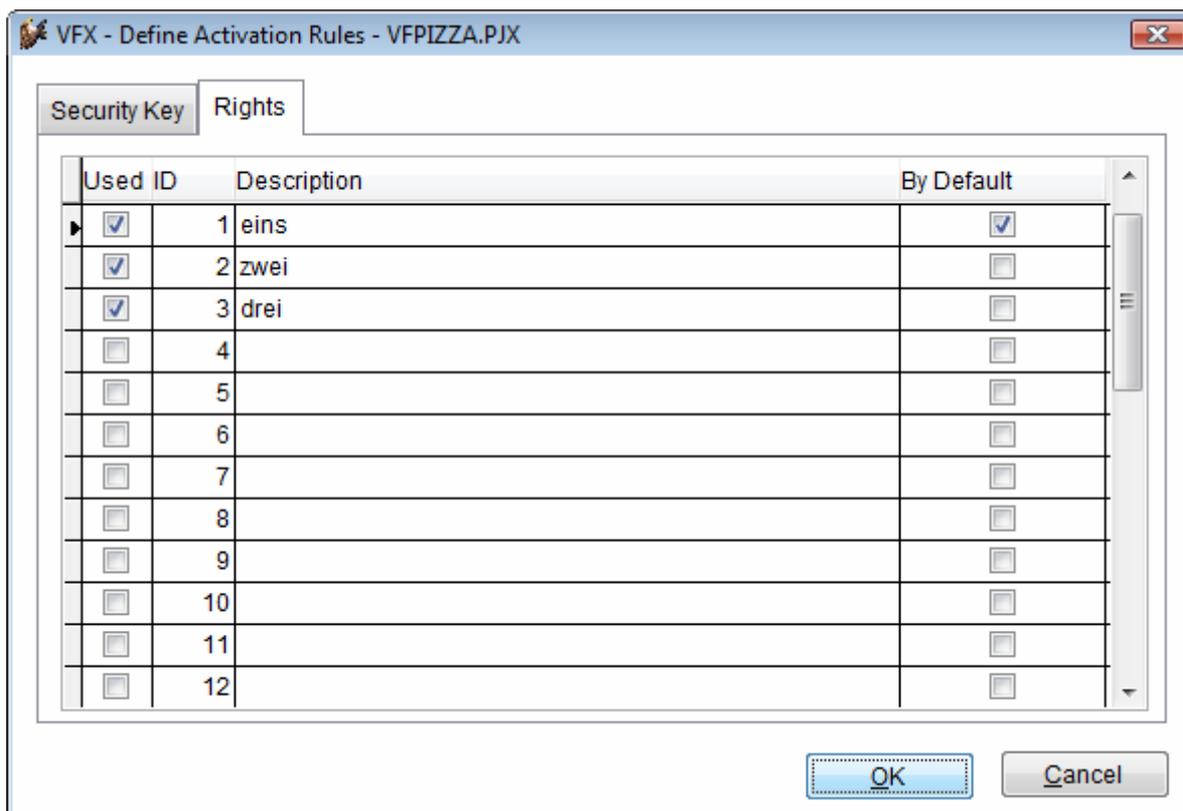
Es können so viele Zeilen dem Grid hinzugefügt werden, wie benötigt werden. Die Zeilen im Grid können mit den Pfeiltasten am rechten Rand des Assistenten in eine andere Reihenfolge gebracht werden. Durch verschieben der Zeilen im Grid ändern sich die Aktivierungsregeln.

---

**ACHTUNG:** Der Wert der Eigenschaft *cActPattern* darf niemals gelöscht werden! Ohne diesen Wert ist es nicht möglich Aktivierungsschlüssel zu erstellen!

---

Auf der Seite *Rights* können bis zu 32 verschiedene Benutzerrechte angelegt werden. Damit kann der Zugriff auf bis zu 32 Module einer Anwendung gesteuert werden. Beispielsweise können Rechte angelegt werden, die es dem Anwender erlauben Formulare zu starten, Berichte zu drucken, Daten zu bearbeiten, Daten anzusehen usw. Zur Laufzeit der Anwendung können die einzelnen Berechtigungen überprüft werden und ggf. wird die entsprechende Aktion ausgeführt.



Alle Benutzerrechte stehen zur Laufzeit als Eigenschaften des global sichtbaren Objekts *goProgram.SecurityRights* zur Verfügung, sodass an jeder Stelle der Anwendung darauf zugegriffen werden kann.

Wenn die Anwendung nicht aktiviert ist, haben alle Benutzerrechte den Wert -1. Wenn die Anwendung aktiviert ist, hat ein Benutzerrecht den Wert 1, wenn die Aktion erlaubt ist und 0, wenn die Aktion nicht erlaubt ist.

Um im Assistenten ein Recht einzutragen muss zuerst das Kontrollkästchen in der ersten Spalte markiert werden. Dann wird ein Name für das Recht eingetragen. Zur Laufzeit der Anwendung wird eine Eigenschaft des *SecurityRights*-Objekts mit diesem Namen angelegt. Daher müssen bei der Eingabe des Namens die Konventionen zur Namensgebung von VFP beachtet werden.

---

**ANMERKUNG:** Anwendungsrechte sind für jede Anwendung unterschiedlich. Die Rechte, die für eine andere Anwendung erstellt wurden, können nicht verwendet werden. Auch wenn ähnliche Rechte benötigt werden, müssen diese neu erstellt werden. Die Anwendungsrechte werden in der Tabelle *Vfxapprights.dbf* im Projektordner gespeichert.

---

---

# Dokumentation

## VFX – Project Documenting

Eine speziell für VFX entwickelte Version des Projekt- und Datenbank-Dokumentations-Tools PDM wird mit VFX geliefert. Das PDM kann über den VFX Menüpunkt Project, Project Documenting gestartet werden und fertigt zu einem Projekt vollautomatisch eine vollständige technische Dokumentation an. Die Dokumentation wird im HTML-Format erstellt und enthält zahlreiche Querverweise.

## VFX – Help Wizard

In VFX ist ein System zur Erstellung von CHM-Hilfedateien integriert.

Der VFX – Help Wizard trägt in alle Steuerelemente eines Projekts automatisch eindeutige *HelpContextIDs* ein.

Wenn zur Laufzeit der Anwendung die Tabelle *Vfxhelp.dbf* zur Verfügung steht, können Hilfetexte in diese Tabelle erfasst werden. Dafür wird das Formular *Vfxhelp.scx* geöffnet. Der eigentliche Hilfetext wird in einer Editbox erfasst und in der Tabelle *Vfxhelp.dbf* gespeichert.

Mittels des VFX – Help Wizard können aus den Daten der Tabelle *Vfxhelp.dbf* vollautomatisch HTM-Dateien sowie ein Hilfe-Projekt erstellt werden. Mit dem Help-Workshop muss dieses Projekt nur noch kompiliert werden und die CHM-Hilfedatei mit kontextsensitiver Hilfe zur gesamten Anwendung ist fertig.

Wenn die Tabelle *Vfxhelp.dbf* zur Laufzeit der Anwendung nicht zur Verfügung steht, wird das normale kontextsensitive Hilfesystem aktiviert. Die CHM-Hilfedatei wird geöffnet und als Parameter wird die *HelpContextID* des aktiven Steuerelements übergeben.

---

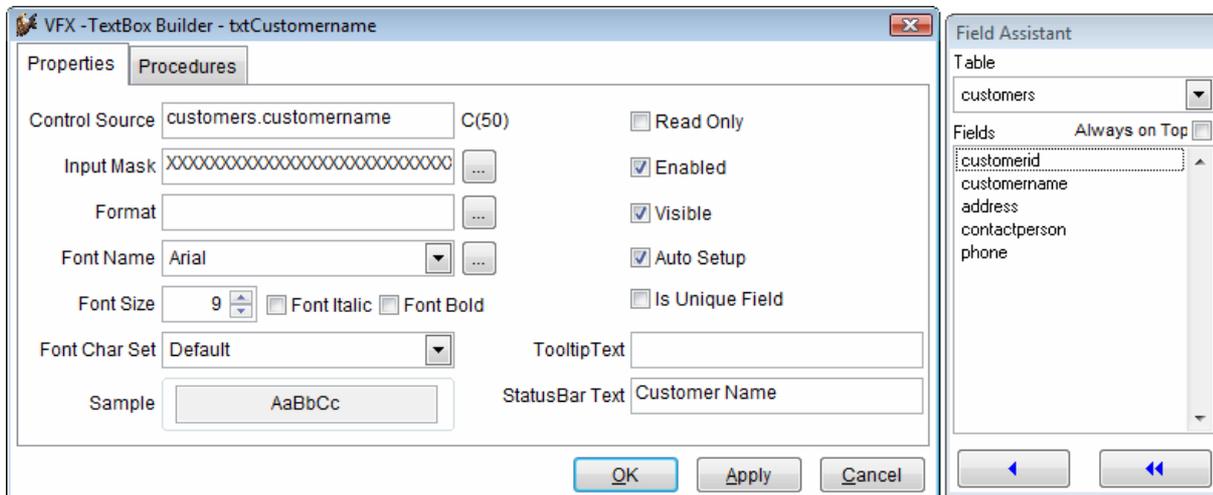
# Sonstiges

## VFX – Textbox Builder

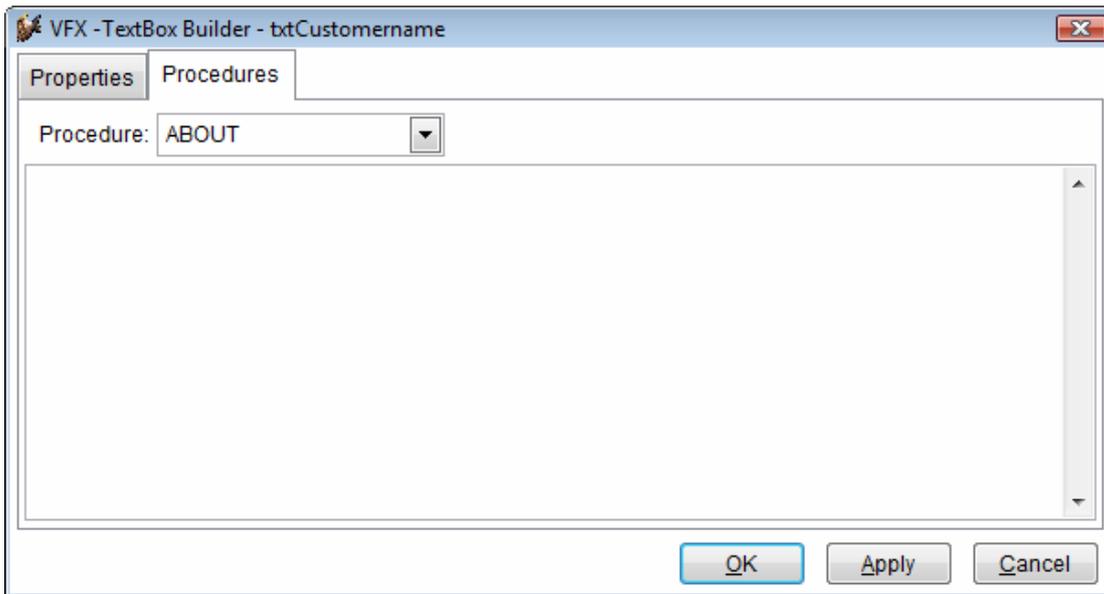
Im neuen VFX – TextBox Builder können die wichtigsten Eigenschaften von allen Steuerelementen basierend auf der VFP Basisklasse Textbox eingestellt werden. Auch die Bearbeitung sämtlicher Methoden ist hier möglich.

Der VFX – TextBox Builder kann aus dem VFX Menü über den Menüpunkt *VFX Power Builders* gestartet werden, wenn eine Textbox das ausgewählte Steuerelement im VFP Formular-Designer ist. Wahlweise kann der Builder auch aus dem Kontextmenü der Textbox gestartet werden.

Auf der Seite *Properties* können insbesondere die Schriftarteneinstellungen gemacht werden. Im *Sample* Feld wird eine Vorschau auf die eingestellten Werte gegeben.

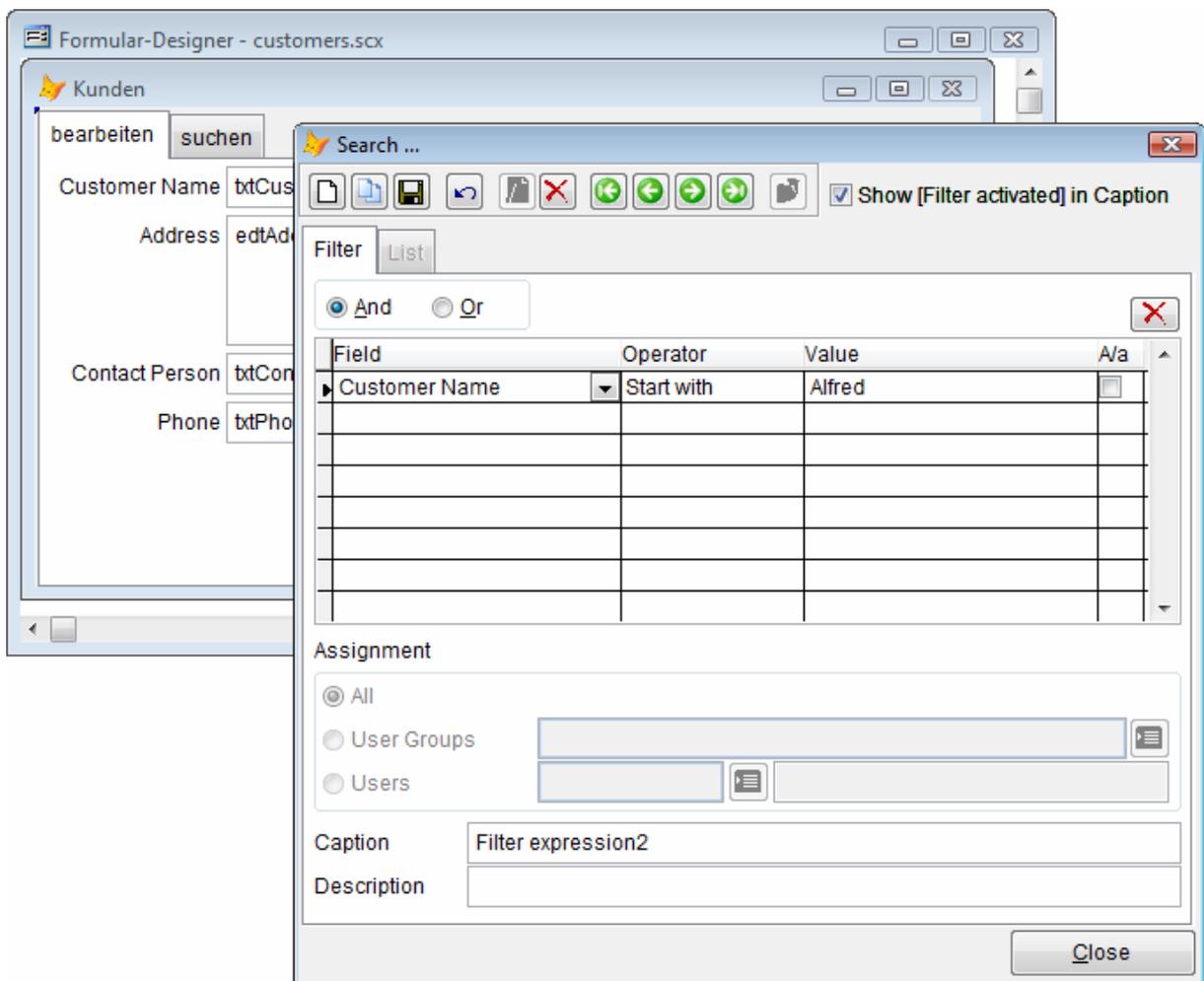


Auf der Seite *Procedures* kann der Code aller Methoden und Ereignisse bearbeitet werden.



## VFX – Filter Builder

Mit dem VFX - Filter Builder können zur Entwicklungszeit Systemfilter erstellt werden, die zur Laufzeit als schreibgeschützte Filter zur Verfügung stehen. Diese Systemfilter können durch Endbenutzer nicht verändert oder gelöscht werden. Um den VFX - Filter Builder starten zu können müssen ein Projekt und ein Formular im Formular-Designer geöffnet sein.

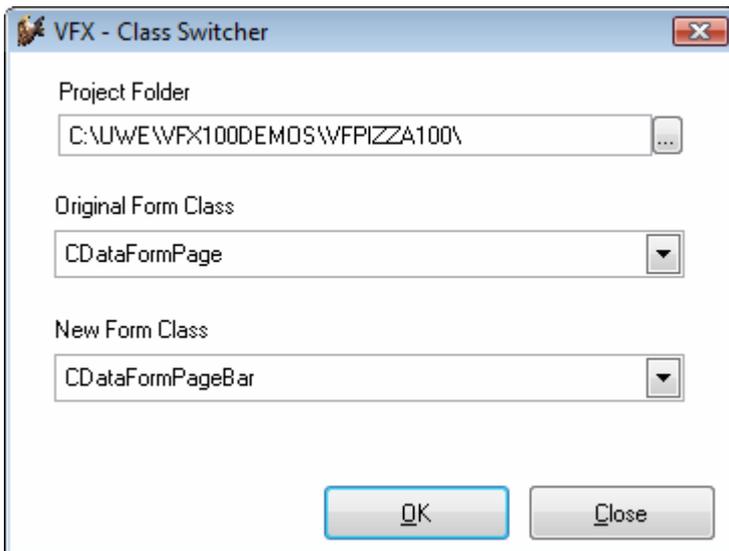


Die Filterbedingungen werden genauso eingegeben, wie es auch im Filterdialog zur Laufzeit der Anwendung möglich ist. Die Felder, die vom VFX – Filter Builder zur Konstruktion der Filterbedingung verwendet werden, werden aus dem geöffneten Formular genauso gelesen, wie es auch zur Laufzeit der Anwendung gemacht wird. Auf der Seite *List* werden alle Systemfilter angezeigt, die bereits für das aktuelle Formular definiert wurden. Systemfilter stehen immer allen Benutzern zur Verfügung. Alle Benutzer können Systemfilter anwenden, aber nicht verändern.

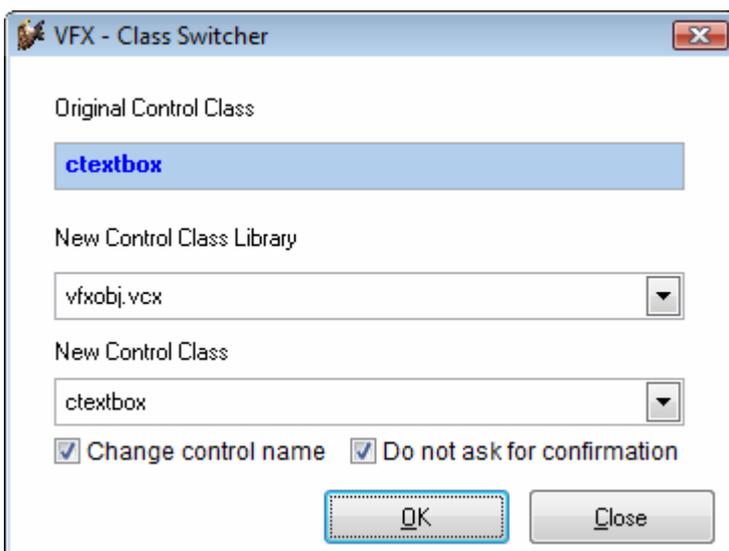
## VFX – Class Switcher

Der Class Switcher hat zwei Funktionen.

Wenn beim Aufruf kein Formular geöffnet ist, ändert der VFX – Class Switcher die Klassen von Formularen in einem ganzen Projekt. Zum Beispiel kann die Formulareklasse *CDataFormPageBar* durch *CDataFormPage* ersetzt werden. Dadurch ist es möglich alle Formulare mit Schaltflächen auszustatten bzw. diese wieder zu entfernen.

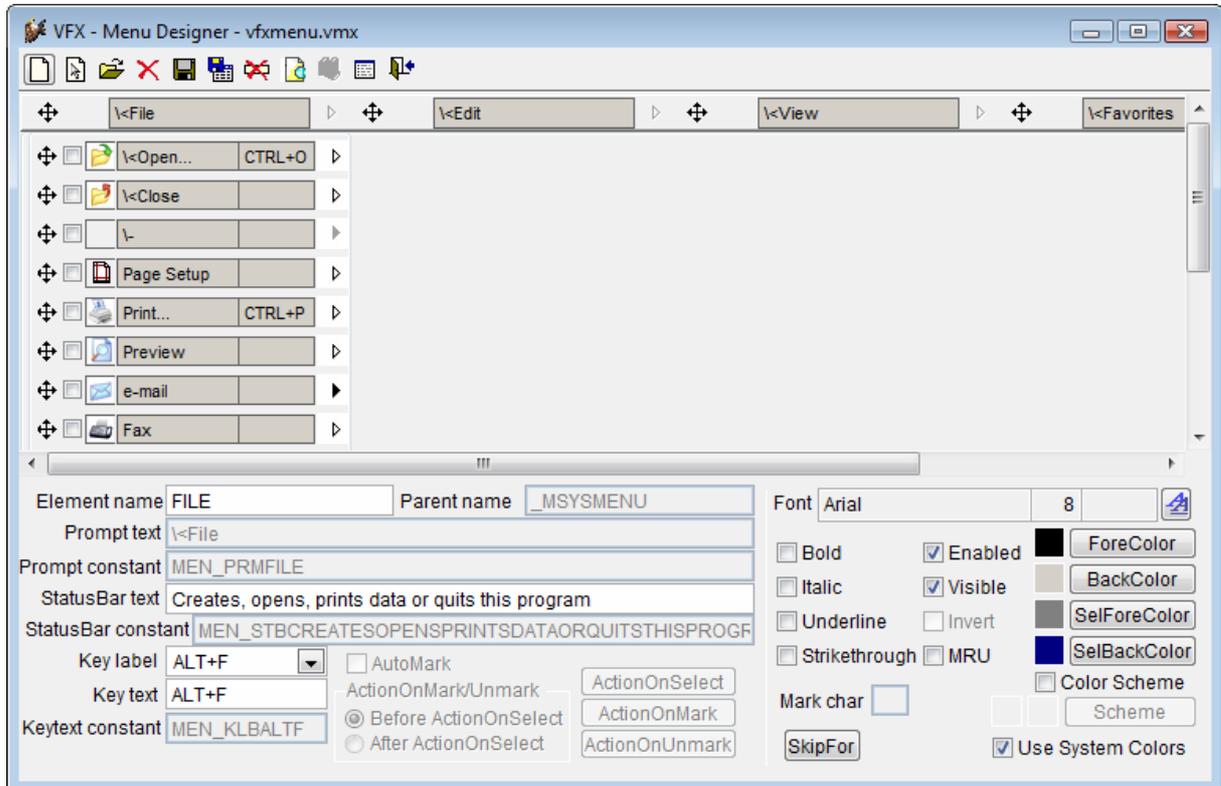


Wenn beim Aufruf des VFX – Class Switcher ein Formular zur Bearbeitung geöffnet ist, können die einzelnen Objekten zugrunde liegenden Klassen geändert werden. So ist es z. B. möglich, aus einer Textbox nachträglich ein Drehfeld zu machen.



## VFX – Menu Designer

Der VFX Menü-Designer (VMD) ist ein Werkzeug zur schnellen Entwicklung von Menüs. Der VMD ist ein visueller Designer, in dem das Menü schon während der Entwicklung so angezeigt wird, wie es zur Laufzeit aussehen wird. Der VMD macht die Entwicklung einfacher und ermöglicht die schnelle Einstellung aller Menü-Eigenschaften im Gegensatz vom VFP Menü-Designer, der nicht alle Eigenschaften von Menüs unterstützt. Es können mehrsprachige Menüs erstellt werden, indem auf die entsprechende Schaltfläche in der Symbolleiste geklickt wird.



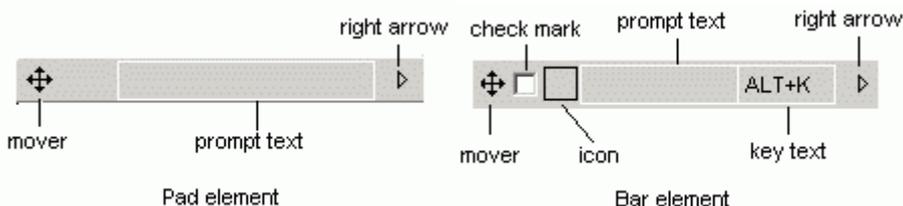
Ein in einem VFX-Projekt enthaltenes Menü kann direkt aus dem VFP-Projekt-Manager mit dem VMD geöffnet werden. Wahlweise können Menüs auch aus dem VMD heraus über das Öffnen-Symbol in der Symbolleiste oder über den entsprechenden Menüpunkt geöffnet werden. Im Öffnen-Dialog kann zwischen den Menütypen .mnx und .vmx gewechselt werden. Wenn ein Menü geöffnet wird, das noch nie mit dem VMD bearbeitet wurde, wird es automatisch in das .vmx-Format konvertiert.

Das geöffnete Menü kann visuell bearbeitet werden. Es können Einträge hinzugefügt und gelöscht werden und es können die Eigenschaften der einzelnen Einträge bearbeitet werden.

Neue Menü-Pads können durch einen Klick auf den Rechtspfeil, der sich rechts neben jedem Menüeintrag befindet, angelegt werden. Dadurch wird ein Untermenü angelegt. Einem Menü können neue Einträge hinzugefügt werden, indem auf den Pfeil nach unten unterhalb des Eintrags geklickt wird.

Ein Menüeintrag oder ein Menü-Pad können gelöscht werden, wenn sich der Fokus darauf befindet. Über den Menüpunkt löschen oder mit der Tastenkombination Strg+Entf wird der markierte Eintrag gelöscht.

Einige der Eigenschaften eines Menüeintrags können visuell eingestellt werden:



- Prompt text

Der angezeigte Text kann direkt eingetragen werden, wenn sich der Fokus auf dem jeweiligen Eintrag befindet. Die Textbox Prompt text im unteren Teil des VMD dient nur zur Anzeige des aktiven Eintrags im visuellen Teil des Designers.

- Key text

Die Bezeichnung des Tastenschlüssels zeigt dem Anwender die Zugriffstaste oder Tastenkombination an, mit der der Eintrag ausgewählt werden kann. Die Bezeichnung sollte dem im unteren Teil des VMD gewählten Tastenschlüssels entsprechen.

- Check mark

Damit sich ein Menüeintrag wie ein Kontrollkästchen verhält, muss bei AutoMark eine Markierung gesetzt werden. Wenn zusätzlich eine Markierung bei Check mark gesetzt wird, ist der Menüeintrag bereits beim Laden des Menüs markiert. Für Menüeinträge, die sich wie ein Kontrollkästchen verhalten, können zusätzliche Code-Teile ausgeführt werden, wenn das entsprechende Kontrollkästchen markiert wird (ActionOnMark) bzw. wenn die Markierung aufgehoben wird (ActionOnUnmark). Im Standard-VFP-Editorfenster kann der jeweilige Code bearbeitet werden.

Der Code, der bei ActionOnMark oder ActionOnUnmark eingegeben wird, kann wahlweise vor oder nach der ActionOnSelect ausgeführt werden. Um dieses Verhalten einzustellen, ist die entsprechende Option „Before ActionOnSelect“ oder „After ActionOnSelect“ auszuwählen.

- Icon

Jedem Eintrag in einem Menü kann ein Symbol zugeordnet werden. Dieses Symbol kann aus den in VFP integrierten Systemressourcen ausgewählt werden oder es kann eine Datei verwendet werden. Durch einen Klick auf das schwarzumrandete Kästchen kann ein Symbol mithilfe des Get a picture from-Dialogs ausgewählt werden. In diesem Dialog kann zwischen einer Datei und einem Symbol aus den VFP Systemressourcen gewählt werden.

Wenn einem Menüeintrag ein Symbol zugeordnet ist und sich dieser Menüeintrag wie ein Kontrollkästchen verhalten soll, dient das Symbol als Markierung. Wenn der Eintrag markiert wird, erscheint das Symbol eingedrückt. Wenn die Markierung aufgehoben wird, erscheint das Symbol normal.

Die Position der einzelnen Einträge innerhalb des Menüs kann per drag & drop verändert werden. Für diesen Vorgang ist der Vierwegepfeil , der sich links neben allen Einträgen befindet, festzuhalten. In einigen Fällen sind Verschiebeoperationen nicht möglich. Ein Menü-Pad kann nicht in einen Menüeintrag umgewandelt werden und umgekehrt. Außerdem ist es nicht möglich einen Menüeintrag in ein Untermenü zu verschieben.

Weitere Eigenschaften der Menüeinträge können im unteren Teil des Menü-Designers eingestellt werden. Dazu gehören der Zeichensatz, die Vordergrund- und Hintergrundfarbe, eine Meldung, die in der VFP-Statusbar angezeigt wird sowie der Name einer Konstanten, die verwendet wird, wenn ein mehrsprachiges Menü erstellt wird. Alle Änderungen werden unmittelbar im aktiven Element angezeigt.

Mit der Schaltfläche ActionOnSelect kann in einem Editor-Fenster die auszuführende Aktion eingegeben werden. Über die Schaltfläche SkipFor kann eine Bedingung eingegeben werden. Wenn diese Bedingung .T. liefert, kann der dazugehörige Menüeintrag nicht ausgewählt werden.

Die eingestellten Eigenschaften beziehen sich immer auf den aktiven Menüeintrag. Neue Menüeinträge erben die Eigenschaften des zuvor ausgewählten Eintrags.

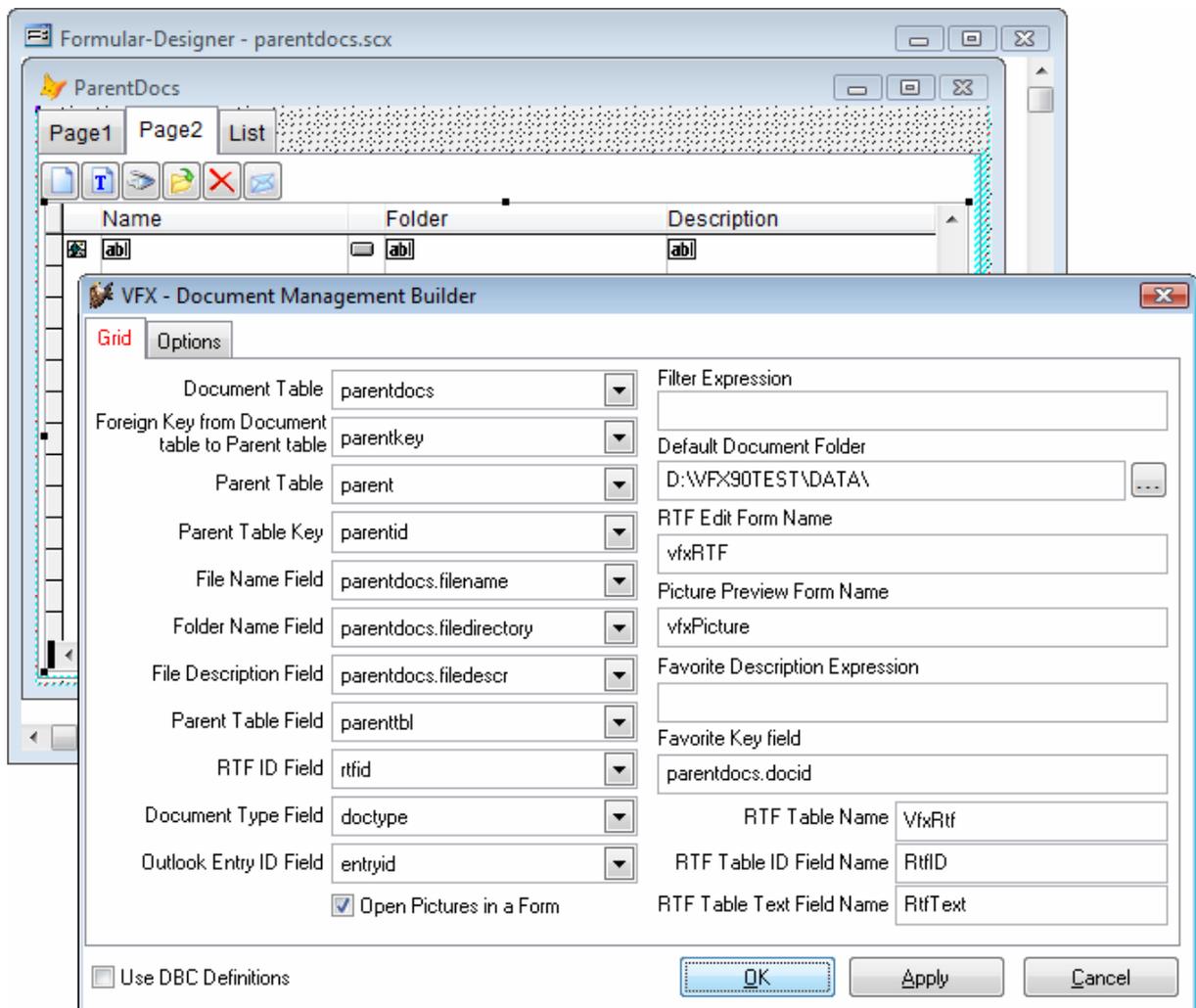
Der Zeichensatz kann über die Schaltfläche Font ausgewählt werden. Der Standard-Windows-Dialog zur Auswahl eines Zeichensatzes erscheint. In diesem Dialog können insbesondere die Schriftart und die Schriftgröße sowie der Schriftschnitt ausgewählt werden.

Zu jeder Zeit kann eine Vorschau des Menüs angezeigt werden, indem in der Symbolleiste oder im VMD-Menü Preview gewählt wird.

Der VMD erstellt nach der Bearbeitung eines Menüs automatisch die erforderlichen Include-Dateien für sprachunabhängige Menüs. Zusätzliche Arbeitsschritte nach der Bearbeitung von Menüs sind nicht erforderlich.

## VFX – Document Management Builder

Die Klasse *CDocumentManagement* dient zur Verwaltung von Dokumenten aller Art (z. B. Word, Excel, Powerpoint) innerhalb einer Anwendung. Die Klasse *CDocumentManagement* ist ein Container, der Child-Datensätze zum aktuellen Datensatz im Formular verwaltet. Die Dokumentenverwaltung ermöglicht dem Anwender Dokumente zu öffnen und als E-Mailanhang zu versenden.



Diese Klasse kann bestehenden Formularen einfach hinzugefügt werden.

*cDefaultDocumentFolder* – Standardordner für Dokumente.

*cFilterExpression* – Anzuwendender Filterausdruck.

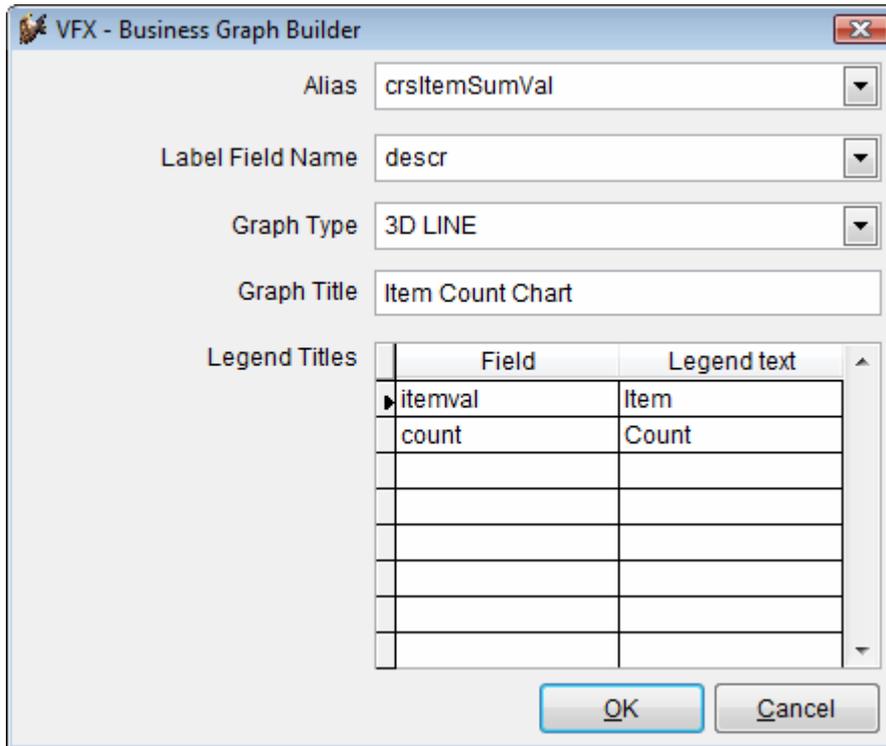
*lOpenPicturesInForm* – Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt ist, werden Bilddateien in einem VFX-Formular angezeigt. Der Name des Formulars kann in der Eigenschaft *cPicturePreviewFormname* angegeben werden. Wenn der Wert dieser Eigenschaft auf *.F.* eingestellt ist, werden Bilddateien mit der Anwendung geöffnet, die im Windows-Explorer mit der Namensweiterung der Datei verknüpft ist. Der Standardwert ist *.F.*

*cPicturePreviewFormname* – Name des Formulars zur Vorschau auf Bilddateien. Der Standardwert ist *VFXPicture*.

*cPicturePreviewCaption* – Der hier zugewiesene Text wird dem Formular zur Vorschau auf Bilddateien als Caption mitgegeben.

## VFX – Business Graph Builder

Mit dem VFX – Business Graph Builder können alle Eigenschaften von *cBusinessGraph* Objekten eingestellt werden.



Field	Legend text
itemval	Item
count	Count

Eine Geschäftsgrafik wird basierend auf einem Arbeitsbereich erstellt. Dieser Arbeitsbereich kann aus der Combobox im Builder aus den Datenquellen der Datenumgebung ausgewählt oder manuell eingegeben werden. Der Arbeitsbereich wird in der Eigenschaft *cAliasName* gespeichert.

In der Combobox *Label Field Name* werden alle Felder des ausgewählten Arbeitsbereichs aufgeführt. Hier kann ein Feld ausgewählt werden, das als Beschriftung für die Datenreihen verwendet wird. Der ausgewählte Feldname wird in der Eigenschaft *cLabelField* gespeichert.

Der Darstellungstyp wird in der Combobox *Graph Type* ausgewählt. Es kann aus den Typen 3D BAR, 2D BAR, 3D LINE, 2D LINE, 3D AREA, 2D AREA, 3D STEP, 2D STEP, 3D COMBINATION, 2D PIE und 2D XY gewählt werden. Der ausgewählte Wert wird in der Eigenschaft *nGraphType* gespeichert.

In der Textbox *Graph Title* kann eine Überschrift für die Geschäftsgrafik eingegeben werden. Die Überschrift wird in der Eigenschaft *cGraphTitle* gespeichert.

Im Grid wird für jedes Feld aus dem ausgewählten Arbeitsbereich eine Zeile angezeigt, ausgenommen ist das Feld, das als Beschriftung für die Datenreihen dient. Die Daten aus allen diesen Feldern werden in der Geschäftsgrafik angezeigt. In der ersten Spalte des Grids werden die Namen der Felder angezeigt. In der zweiten Spalte sollte für jedes Feld eine Bezeichnung angegeben werden. Diese Bezeichnungen werden in der Legende der Geschäftsgrafik angezeigt. Die Legendenbezeichnungen werden in einer Komma-Separierten Liste entsprechend der Reihenfolge der Felder in der Eigenschaft *cLegendTitles* gespeichert.

---

## VFX – Task Pane

Der VFX – Application Manager ist in die VFP Task Pane integriert.

Über die Symbolleiste stehen folgende Funktionen zur Verfügung:

*New Project* Startet den VFX – Application Wizard.

*Open Project* Öffnet ein VFP-Projekt und stellt den aktuellen Pfad auf den Projektordner.

*Modify Project* Öffnet das in der VFX – Task Pane selektierte Projekt und stellt den aktuellen Pfad auf den Projektordner.

*Add Project* Fügt ein vorhandenes VFP-Projekt der VFX – Task Pane hinzu.

*Rebuild* Neu kompilieren aller Dateien des in der VFX – Task Pane selektierten Projekts. Das Projekt wird nach dem Kompilieren zur Bearbeitung geöffnet.

*Properties* Start der VFX – Project Properties zum in der VFX – Task Pane selektierten Projekt.

*Project Backup* Erstellt eine Zip-Datei vom selektierten Projekt.

*Delete* Entfernt das selektierte Projekt aus der VFX – Task Pane.

Mit einem einfachen Mausklick kann von einem Projekt eine Sicherungskopie in eine Zip-Datei erstellt werden.

Mit einem Klick auf das Symbol  wird die Sicherung gestartet. Wenn das Projekt zu diesem Zeitpunkt geöffnet ist, wird es vor Beginn der Sicherung geschlossen.