



VFX 9.0 – Was ist neu?

November 2004

Inhaltsverzeichnis

Neue Eigenschaften für Entwickler.....	4
Das Anwendungsobjekt goProgram.....	4
Datenzugriff mit CursorAdapttern	5
Die Klasse cBaseDataAccess	5
Die Verwaltung der Datei Config.vfx	6
Der VFX CursorAdapter Wizard	8
Auswahl der Datenquelle	8
Auswahl der Klassen und Klassenbibliotheken	9
Auswahl der Tabellen	10
Datenbankreparatur.....	11
VFX - Application Wizard und VFX - Application Builder	11
VFX Form Wizard.....	17
VFX Form Builder	17
Bearbeitung der Datenumgebung mit dem VFX Form Builder	17
Weitere neue Eigenschaften der VFX Form Builder.....	19
VFX Parent/Child Builder	27
Die Klasse cComboPicklist.....	28
VFX – Combo Pick List Builder	29
Das Formular zur Bearbeitung von Auswahllisten.....	30
Erweiterungen in OneToMany-Formularen.....	30
Die Formulkasse COneToManyPageFrame	30
VFX - COneToManyPageFrame Builder	31
Fehlerbehandlung	31
Erstellen von Aktivierungsschlüsseln.....	31
Fernwartung	32
Wie funktioniert die Fernwartung?.....	32
Voraussetzungen.....	33
Registrierung einer Subdomain	33
Das Fernwartungsprogramm Radmin	34
Die Fernwartung aus der Sicht des Supporters	34
VFX Project Update Wizard	35
VFX Task Pane	36
VFP Toolbox.....	36
Visual Extend Menü-Designer	36
Registrierung und Aktivierung von VFX 9.0.....	36
PDM.....	36
Weitere Verbesserungen für Entwickler.....	37
Neue Eigenschaften für Endbenutzer	38
Benutzerverwaltung	38
Das Formular Benutzerverwaltung.....	38
Zurzeit angemeldete Benutzer.....	39
Benutzergruppen	40
Layout	41
Gedockte Formulare	43
Berichte.....	44
E-Mail Erweiterungen	45
Fax	46
Geschäftsgrafiken	47
Beispiel	48
Lokalisierung zur Laufzeit	49

Integration der VFP Toolbox für Endanwender.....	50
Aktualisierung der Anwendung	52
Verbesserte Suchfunktionen	52
Treeview	53
Klasse cDocumentManagement zur Dokumentverwaltung.....	54
Info-Dialog.....	55
Weitere Verbesserungen für Endbenutzer	55

Neue Eigenschaften für Entwickler

Das Anwendungsobjekt *goProgram*

Neue Eigenschaften:

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-, PDF-, 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-Mailadressen eine Textbox zur Eingabe von BCC-Empfängern angezeigt. Der Standardwert ist *.T.* 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.

nIndexInsteadFilter – 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.

Neue Methoden:

OnPreStart – In dieser Methode kann Code eingetragen werden, der vor Ausführung der *Start*-Methode ausgeführt werden soll.

OnPostStart – In dieser Methode kann Code eingetragen werden, der nach Ausführung der *Start*-Methode ausgeführt werden soll.

Datenzugriff mit CursorAdapttern

Die Builder von VFX 9.0 unterstützen jetzt die Verwendung von CursorAdapttern in der Datenumgebung. CursorAdapter können in der Datenumgebung genauso wie lokale und remote Ansichten verwendet werden.

CursorAdapter werden auch als Datenquelle für Auswahllisten unterstützt. CursorAdapter können in allen VFX Buildern und Wizards als Datenquelle angegeben werden.

VFX 9.0 enthält eine CursorAdapter-Klasse, die die Grundfunktionalität zum Zugriff auf die Anwendungsdaten enthält. Dies ist die Klasse *cBaseDataAccess* in der Klassenbibliothek *Vfxctrl.vcx* und sollte als Basis für alle CursorAdapter verwendet werden. Diese Klasse stellt sicher, dass die gesamte Anwendung eine gemeinsame Verbindung verwendet und keine überflüssigen Verbindungen geöffnet werden.

Die Klasse cBaseDataAccess

In VFX 9.0 gibt es neue Möglichkeiten des Datenzugriffs. Die neue Klasse *cBaseDataAccess* ermöglicht es basierend auf der VFP-Klasse *CursorAdapter* auf verschiedene Datenquellen zuzugreifen. Wenn in einer Anwendung der Datenzugriff ausschließlich über die Klasse *cBaseDataAccess* erfolgt, ist es leicht die Anwendung später auf andere Datenquellen zu portieren. So ist es zum Beispiel einfach möglich zwischen einer VFP-Datenbank und einer SQL Server-Datenbank zu wechseln. Die Datenzugriffseinstellungen für die Klasse *cBaseDataAccess* sind in der Datei *Config.vfx* gespeichert.

Wenn ein Objekt der Klasse *cBaseDataAccess* instanziiert wird, wird aus der Eigenschaft *goProgram.cDataSourceType* der zu verwendende Datenbanktyp gelesen. Wenn der Datenbanktyp *NATIVE* ist, wird eine VFP-Datenbank verwendet. Aus den Eigenschaften *goProgram.cDatadir* und *goProgram.cMainDatabase* werden der Pfad zur Datenbank und der Name der Datenbank gelesen. Bei anderen Datenbanktypen werden die Verbindungsinformationen aus der Methode *GetConnection* des Verbindungs-Managers bezogen.

In der Klassenbibliothek *Appl.vcx* befindet sich die Klasse *cAppDataAccess*, die eine 1:1-Ableitung der Klasse *cBaseDataAccess* ist. Entwickler sollten eigene Erweiterungen oder Änderungen des Datenzugriffs in der Klasse *cAppDataAccess* machen.

Eigenschaften:

cConnMgrName – Name der Eigenschaft des Anwendungsobjekts *goProgram*, das den Namen des Verbindungs-Manager-Objekts enthält. Dieses Verbindungs-Manager-Objekt verwaltet den Datenzugriff der Klasse *cBaseDataAccess*.

cExecuteAfterCursorFill – Der hier eingetragene Befehl wird nach Ausführung der Methode *CursorFill* des *CursorAdapters* ausgeführt. Hier kann Code eingetragen werden, der die Daten des erstellten Cursors verarbeitet. Mithilfe dieser Eigenschaft kann einem *Cursoradapter* zur Laufzeit Code hinzugefügt werden.

Filter – Ein logischer Ausdruck mit dem die Daten des erstellten Cursors gefiltert werden.

Order – Der hier angegebene Indexschlüssel wird zur Sortierung des erstellten Cursors verwendet. Indexschlüssel für *Cursoradapter* können im VFX – Data Environment Builder angelegt werden.

Methoden:

CreateIndexes – Der Code dieser Methode wird vom VFX - Data Environment Builder erstellt. Hier werden Befehle zur Erstellung von temporären Indexdateien für den *Cursor* eingetragen. Diese Methode wird nach Ausführung der Methode *CursorFill* aufgerufen.

Die Verwaltung der Datei *Config.vfx*

Während der Entwicklung einer Anwendung wird für alle *CursorAdapter* eine Datenquelle verwendet, die auf dem Entwicklungsrechner zur Verfügung steht. Die Datenquelle auf den Kundenrechnern muss nicht identisch sein. Zum Beispiel kann auf dem Entwicklungsrechner eine SQL Server-Datenbank verwendet werden, während bei den Kunden eine VFP-Datenbank zum Einsatz kommt.

Auch wenn auf dem Entwicklungsrechner und auf dem Kundenrechner eine SQL Server-Datenbank verwendet werden soll, so wird der Name des SQL Servers auf beiden Rechnern unterschiedlich sein. Daher ist in der Regel auf dem Entwicklungsrechner und dem Kundenrechner eine andere Verbindungszeichenfolge erforderlich.

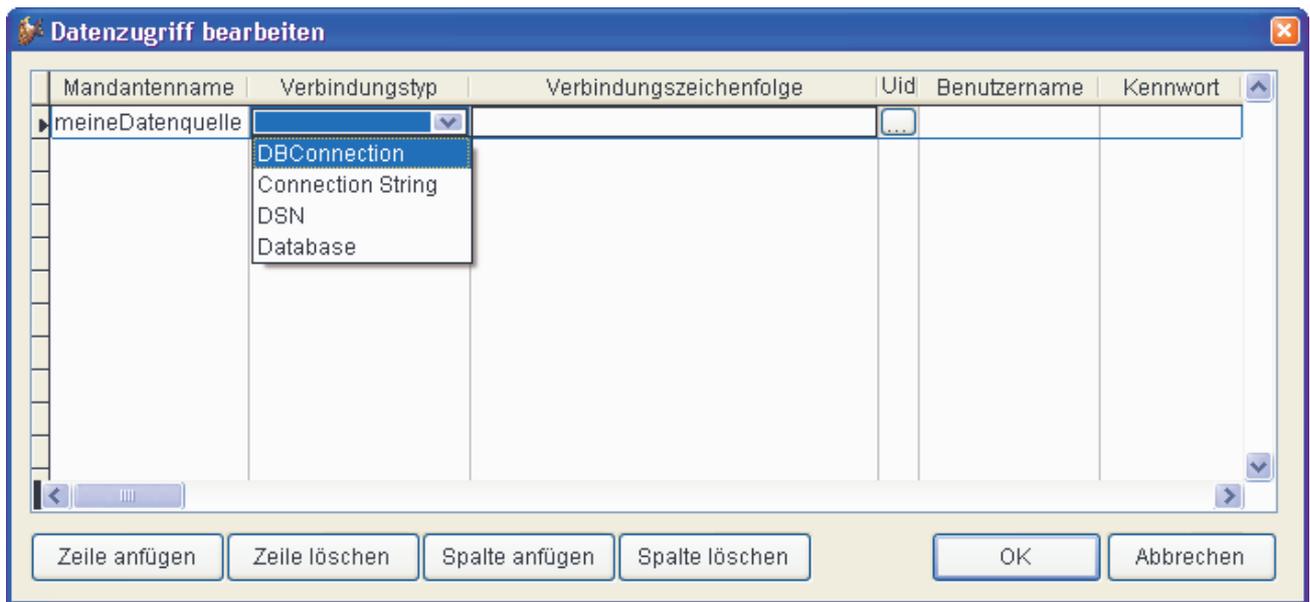
VFX verwendet einen eigenen Verbindungs-Manager um eine Verbindung zur Datenquelle herzustellen. Dieser Verbindungs-Manager wird als Child-Objekt des Anwendungsobjekts instanziiert und steht über die Referenz *goProgram.oConnMgr* zur Laufzeit zur Verfügung.

CursorAdapter-Objekte basierend auf der Klasse *cBaseDataAccess* verwenden das Objekt *goProgram.oConnMgr*, eine Instanz der Klasse *cConnectionMgr*, um eine Verbindung zur Datenquelle herzustellen. Die Einstellungen für das *goProgram.oConnMgr* Objekt werden aus der Datei *Config.vfx* gelesen. In dieser Datei befinden sich die Informationen über die von der Anwendung verwendete Datenquelle.

Die Datei *Config.vfx* enthält aus Sicherheitsgründen verschlüsselte Daten, die zur Verbindung mit der Kundendatenbank verwendet werden, zum Beispiel Typ der Datenquelle, Verbindungszeichenfolge und andere. Das Kennwort zur Verschlüsselung ist in der Eigenschaft *goProgram.cConfigPassword* gespeichert. VFX-Entwickler sollten dieses Kennwort selbst zuweisen.

Die Datei *Config.vfx* kann vom Entwickler erstellt und zusammen mit der Anwendung ausgeliefert werden. Wenn beim Start der Anwendung keine Datei *Config.vfx* gefunden wird, erscheint der Dialog *Datenquellen verwalten* und der Anwender muss die Datenquelle eintragen.

Benutzer mit Administratorrechten können die Datei *Config.vfx* später über den Menüpunkt *Extras, Datenquellen verwalten* bearbeiten.



Für jeden Kunden kann gewählt werden, ob mit einer VFP-Datenbank oder einer Remote-Datenbank gearbeitet werden soll. Die Datei *Config.vfx* kann auch mehrere Datensätze enthalten. Wenn mehr als ein Datensatz vorhanden ist, erscheint beim Start der Anwendung ein Datenbankauswahldialog.

Es kann eine Verbindung aus einer VFP-Datenbank verwendet werden. Zur Laufzeit wird der Name der Verbindung in der Eigenschaft *cDBCConn* des Objekts *goProgram* gespeichert. In der Datei *Config.vfx* wird der Name der zu verwendenden Datenbank gespeichert. Beim Start der Anwendung werden die Informationen zur Datenbank aus dieser Datei gelesen.

Um eine ODBC-Verbindung zu benutzen, kann eine Verbindungszeichenfolge oder eine existierende DSN verwendet werden. Wenn eine Verbindungszeichenfolge als Datenquelle gewählt wird, kann über die Schaltfläche ein Dialog angezeigt werden, der hilft eine gültige Verbindungszeichenfolge zu erstellen.

Wenn eine DSN als Datenquelle gewählt wird, können ein Benutzername und ein Kennwort eingegeben werden, die zur Anmeldung bei der Datenquelle zur Laufzeit verwendet werden. Wenn hier kein Benutzername und Kennwort eingegeben werden und die Datenquelle eine Anmeldung erfordert, erscheint zur Laufzeit ein Anmeldedialog, der den Anwender zur Eingabe von Benutzername und Kennwort auffordert.

Wenn die Datei *Config.vfx* anstelle von *Vfxpath.dbf* verwendet wird, müssen alle anderen Spalten entsprechend der Tabelle *Vfxpath.dbf* ausgefüllt werden.

Genau wie bei der Tabelle *Vfxpath.dbf* können der Datei *Config.vfx* eigene Felder hinzugefügt werden, deren Werte dann zur Laufzeit der Anwendung zur Verfügung stehen. Die Schaltfläche *Add Column* zeigt einen Dialog an, in dem Name und Typ von neuen Feldern eingegeben werden können.



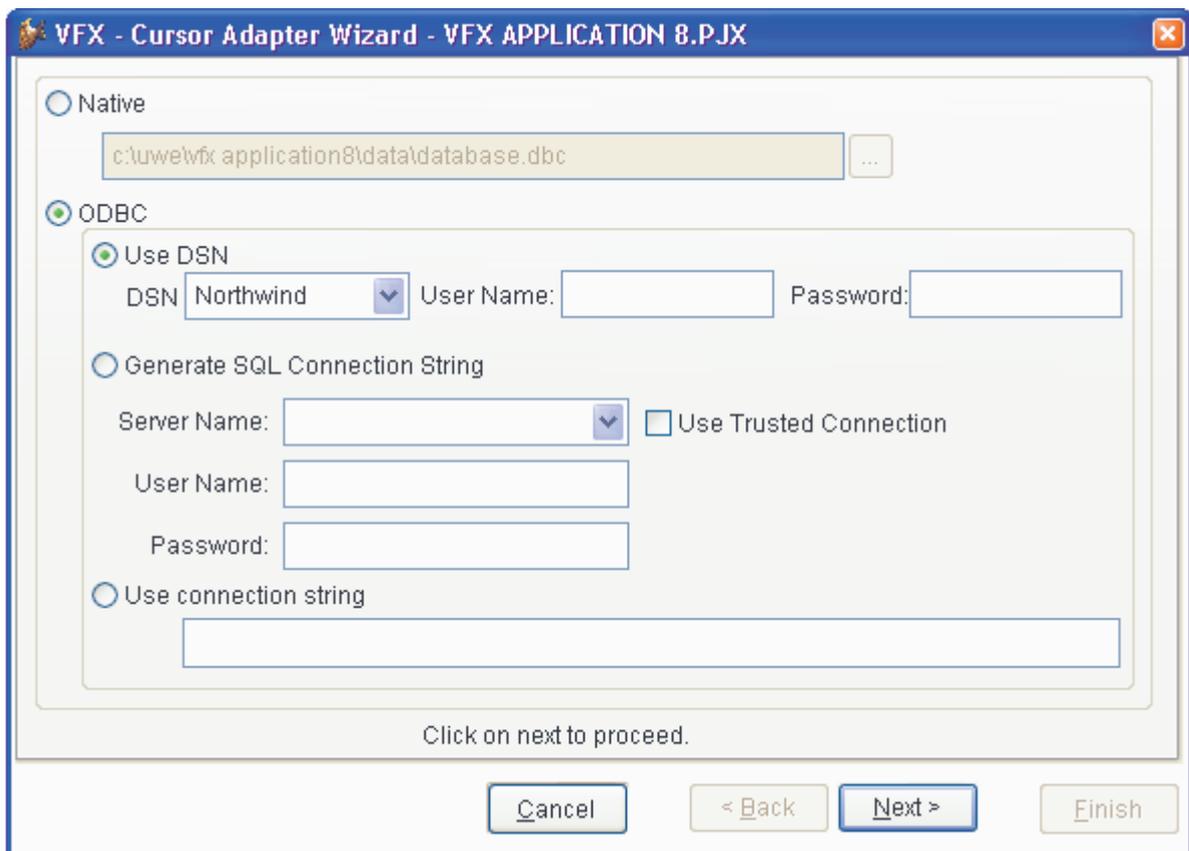
Der VFX CursorAdapter Wizard

Um die Erstellung von CursorAdaptoren zu erleichtern enthält VFX 9.0 einen CursorAdapter Wizard. Dieser Wizard erstellt für jede Tabelle aus einer Datenbank eine CursorAdapter-Klasse.

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.

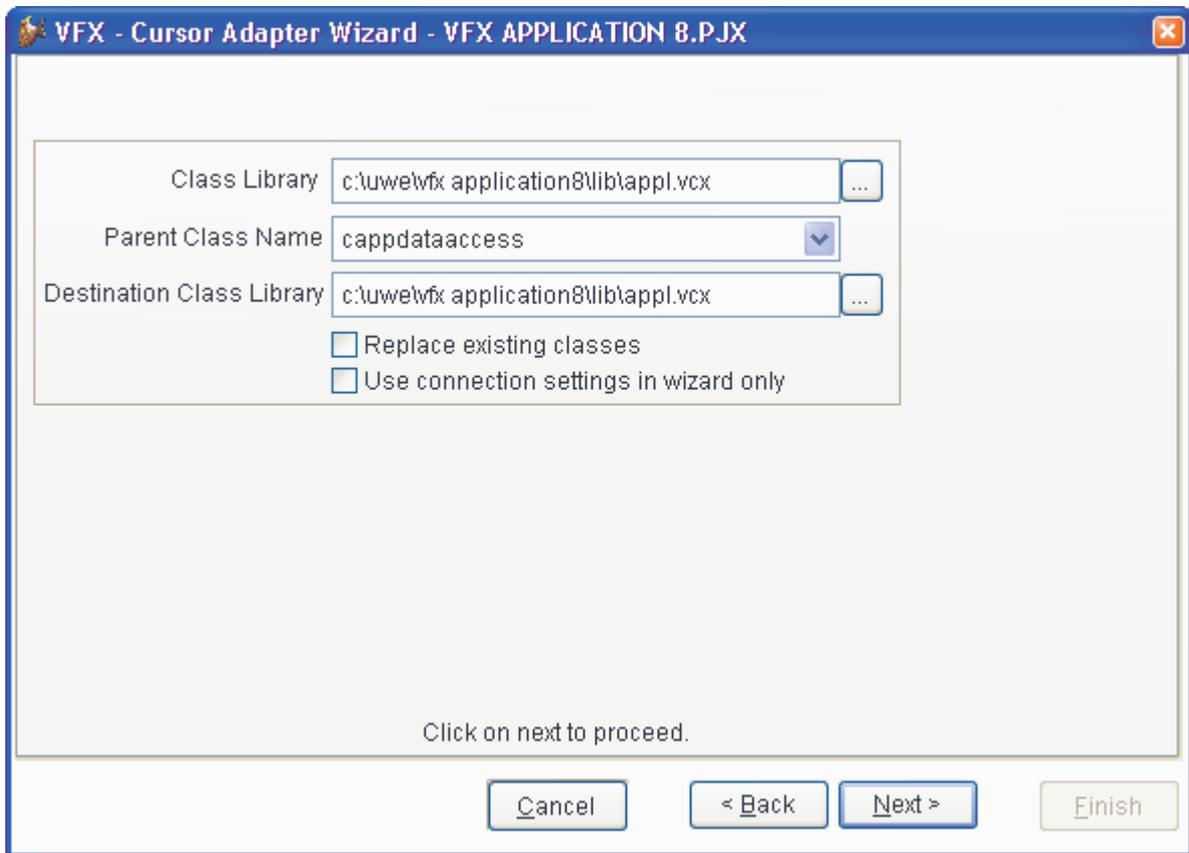
Auswahl der Datenquelle



Diese Datenquelle wird die Datenquelle der kompilierten 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.

Auswahl der Klassen und Klassenbibliotheken



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:

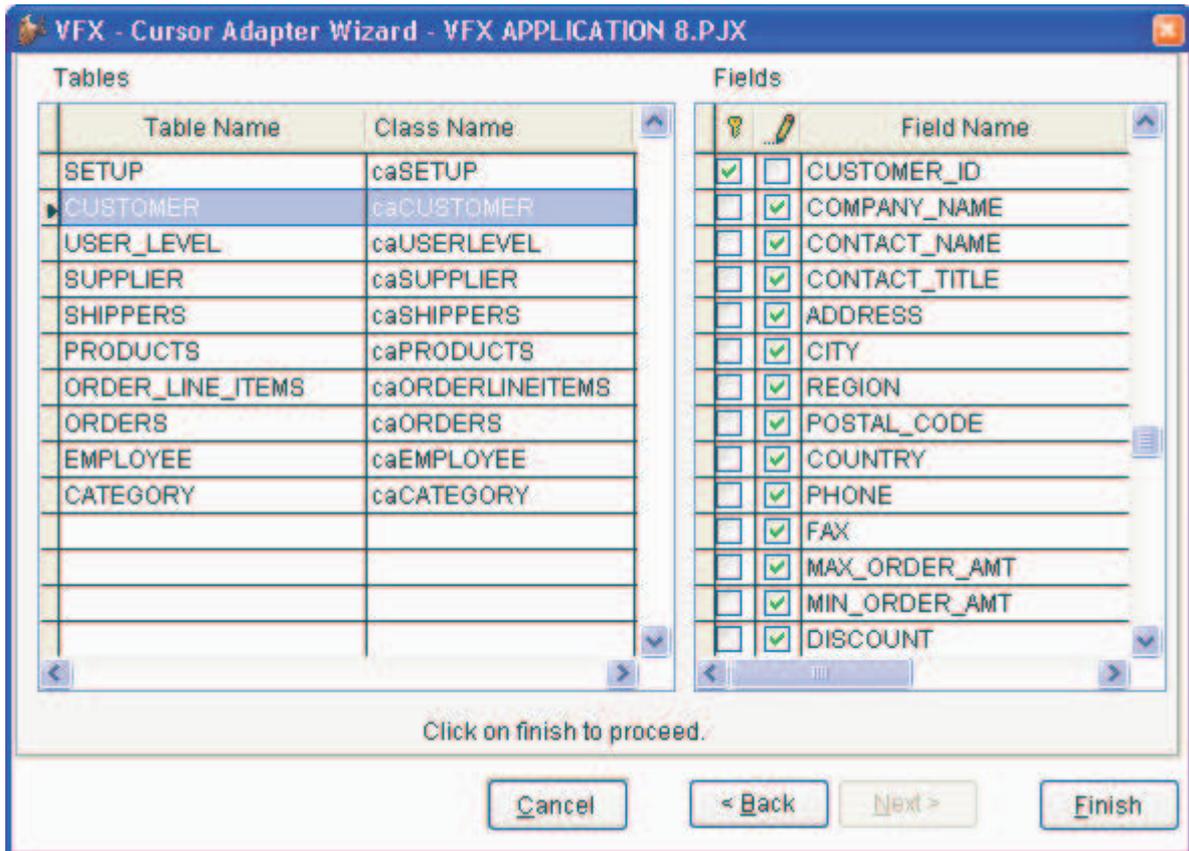
Class Library: *Vfxctrl.vcx*

Parent Class Name: *cBaseDataAccess*

Destination Class Library: *Appl.vcx*

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

Auswahl der Tabellen



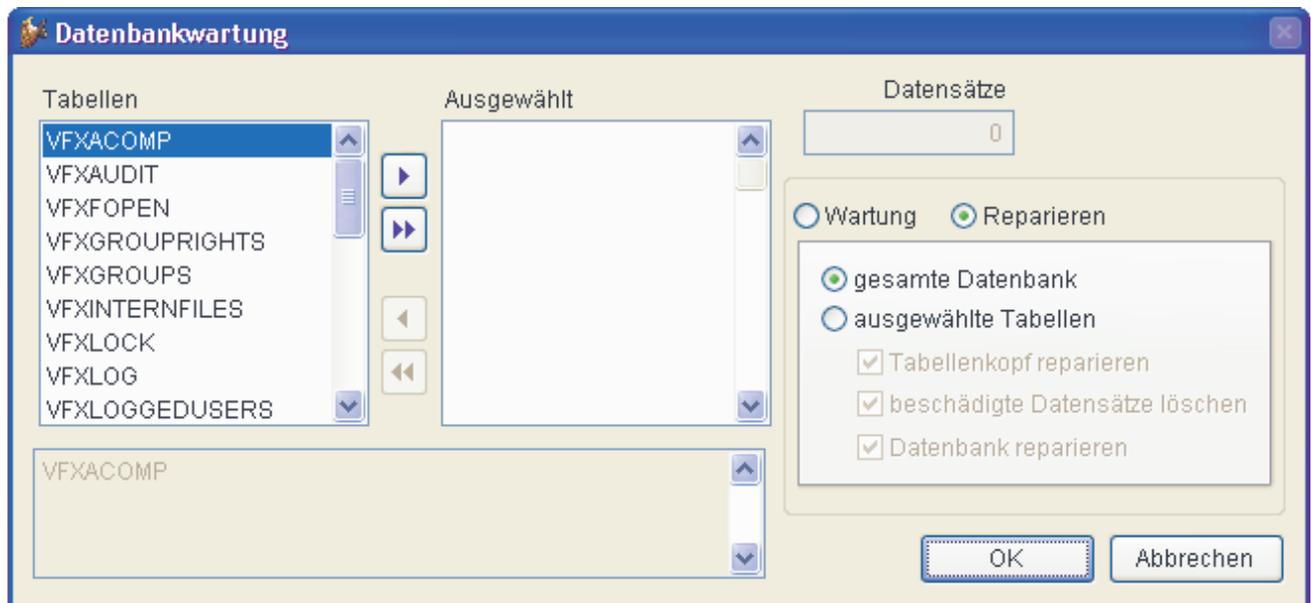
Der letzte Schritt zeigt Listen aller Tabellen und Felder für die CursorAdapter erstellt werden sollen. Beim Bewegen des Satzzeigers in der Tabellenliste auf der linken Seite werden auf der rechten Seite die dazugehörigen Felder angezeigt.

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.

Datenbankreparatur

Zusätzlich zu den Datenbank-Wartungsmöglichkeiten aus bisherigen enthält VFX 9.0 ein neues Werkzeug zur Reparatur von defekten Datenbanken. Die Reparaturmöglichkeit von Datenbanken ist den Dialog *Datenbankwartung* integriert.



Bei Bedarf können wahlweise ausgewählte Tabellen oder die gesamte Datenbank repariert werden. Wenn nur ausgewählte Tabellen repariert werden sollen, kann nur der Tabellenkopf repariert werden oder es werden defekte Datensätze gelöscht.

Zur Datenbankreparatur wird eine leere Datenbank benötigt, die die gleiche Struktur wie die beschädigte Datenbank hat. Vor der Erstellung einer ausführbaren Datei wird mithilfe von *Gendbc.prg* ein Programm erstellt, das diese Struktur zur Laufzeit herstellen kann. Das generierte Programm wird dem Projekt automatisch hinzugefügt.

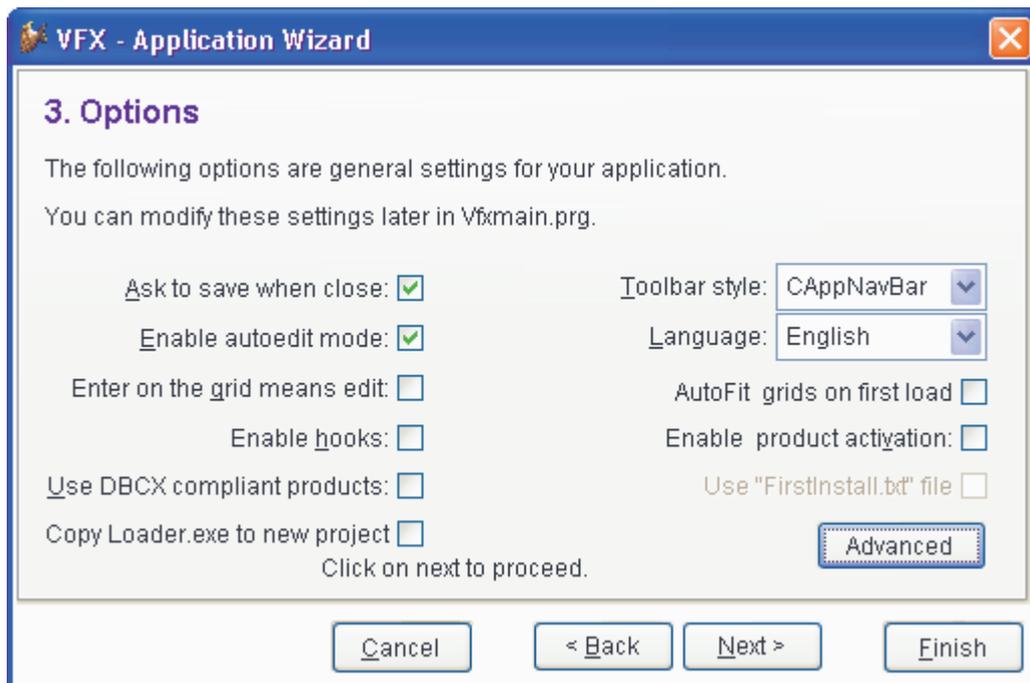
Wenn *beschädigte Datensätze löschen* ausgewählt wird, werden alle Datensätze ohne Primärschlüssel oder mit doppeltem Primärschlüssel gelöscht.

VFX - Application Wizard und VFX - Application Builder

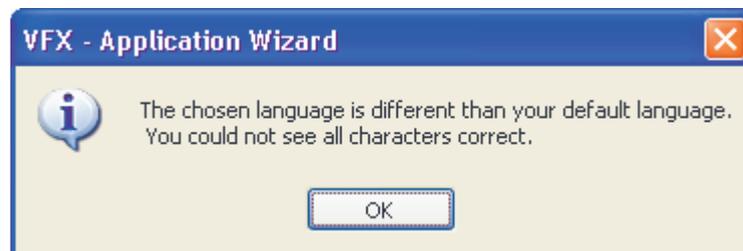
Das Erstellen neuer Projekte geschieht mit dem VFX - Application Wizard. Dieser Wizard kann über eine Schaltfläche in der VFX Task Pane oder über das VFX 9.0-Menü gestartet werden.

Auf den Seiten 1 und 2 werden, wie in bisherigen VFX-Versionen auch, der Name des Projekts, der Pfad sowie Angaben zum Info-Dialog gespeichert. 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.

Viele der Einstellungen, die im VFX - Application Wizard gemacht werden, werden zur Verwendung in späteren Projekten gespeichert.



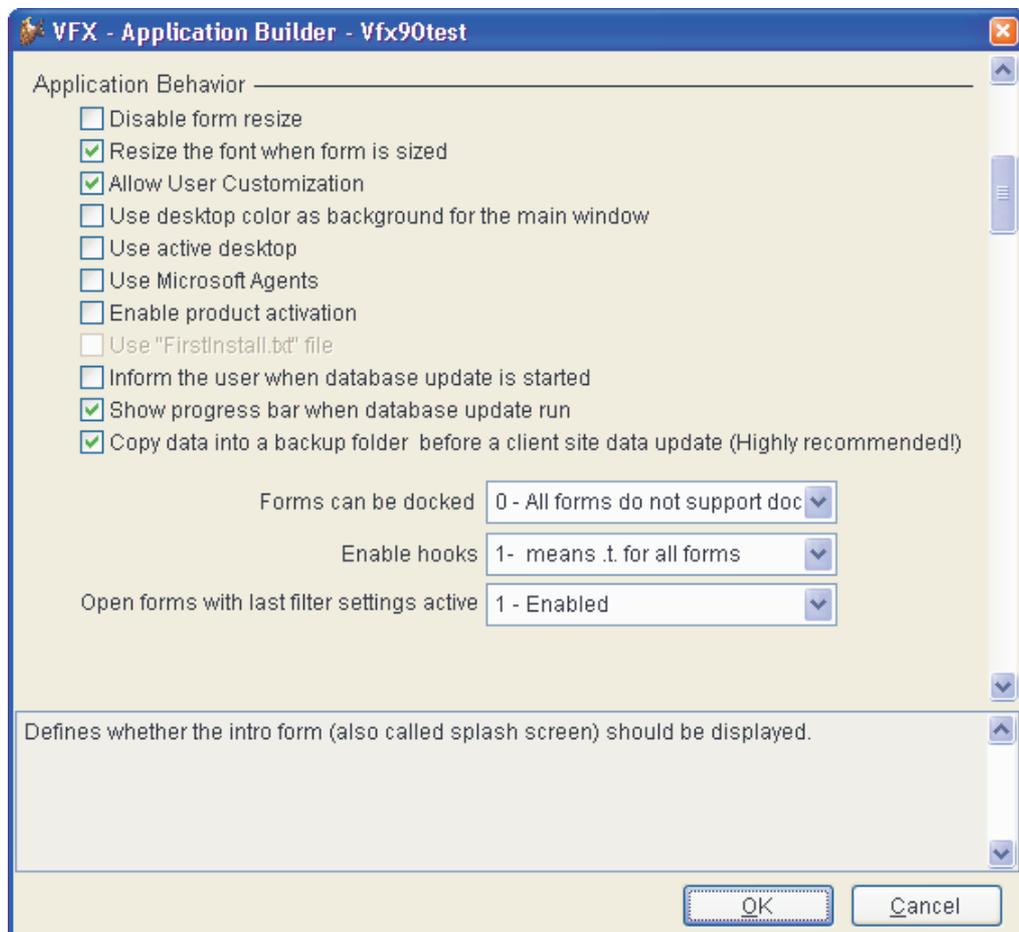
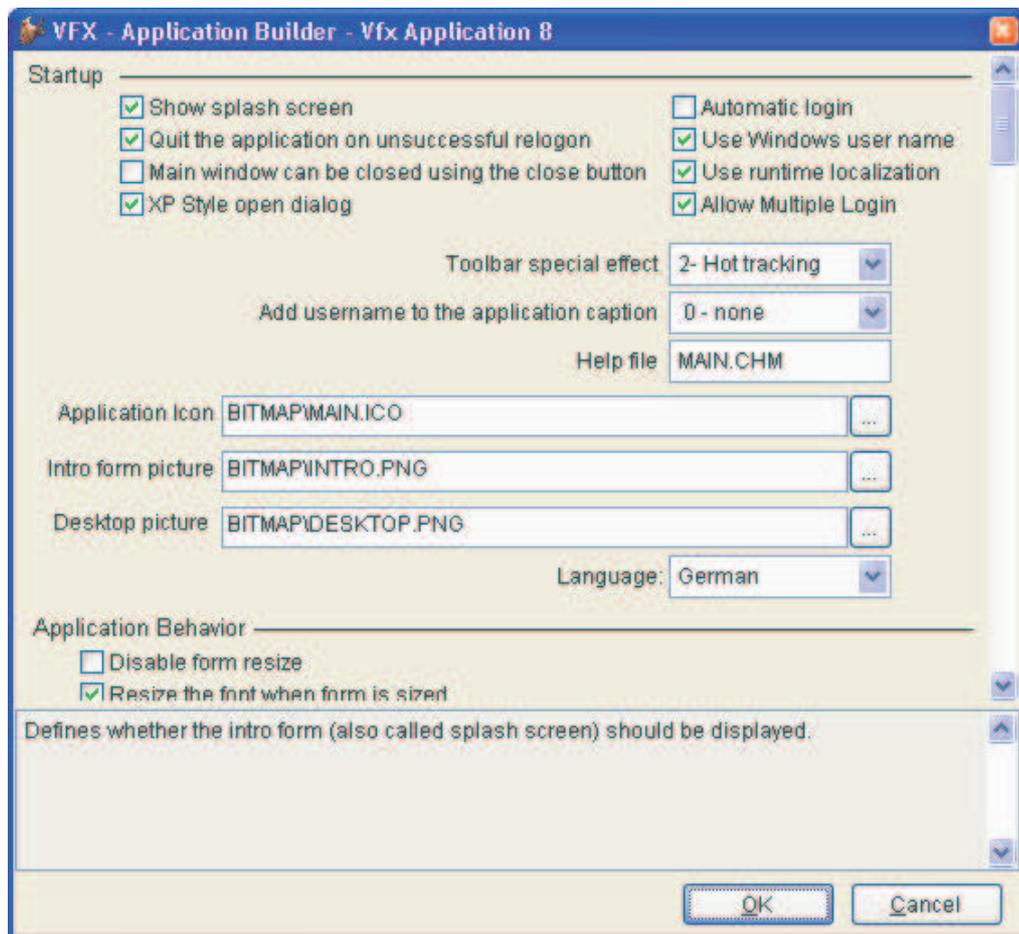
Bei der Auswahl einer Sprache für die generierte Anwendung prüft VFX die aktuellen Unicode-Einstellungen des Betriebssystems. Wenn die Zeichen der gewählten Sprache mit den aktuellen Einstellungen nicht angezeigt werden können, erscheint eine Warnung.

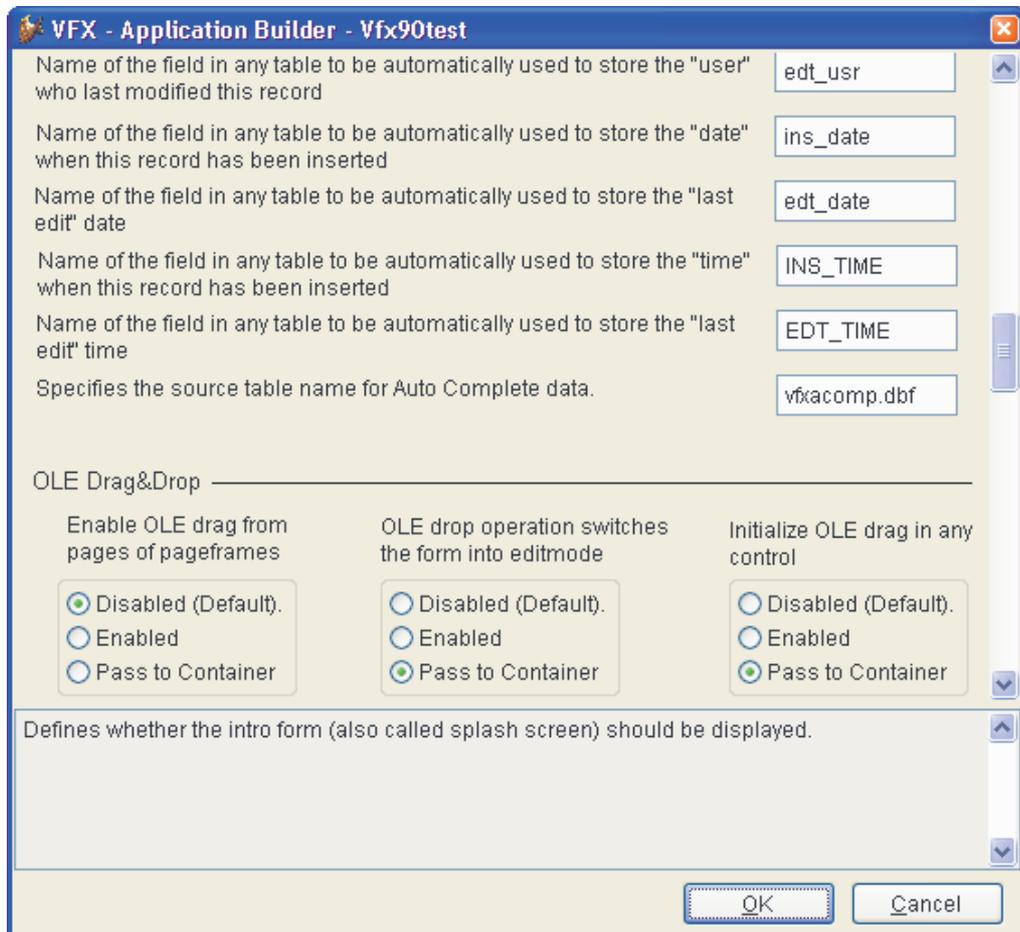
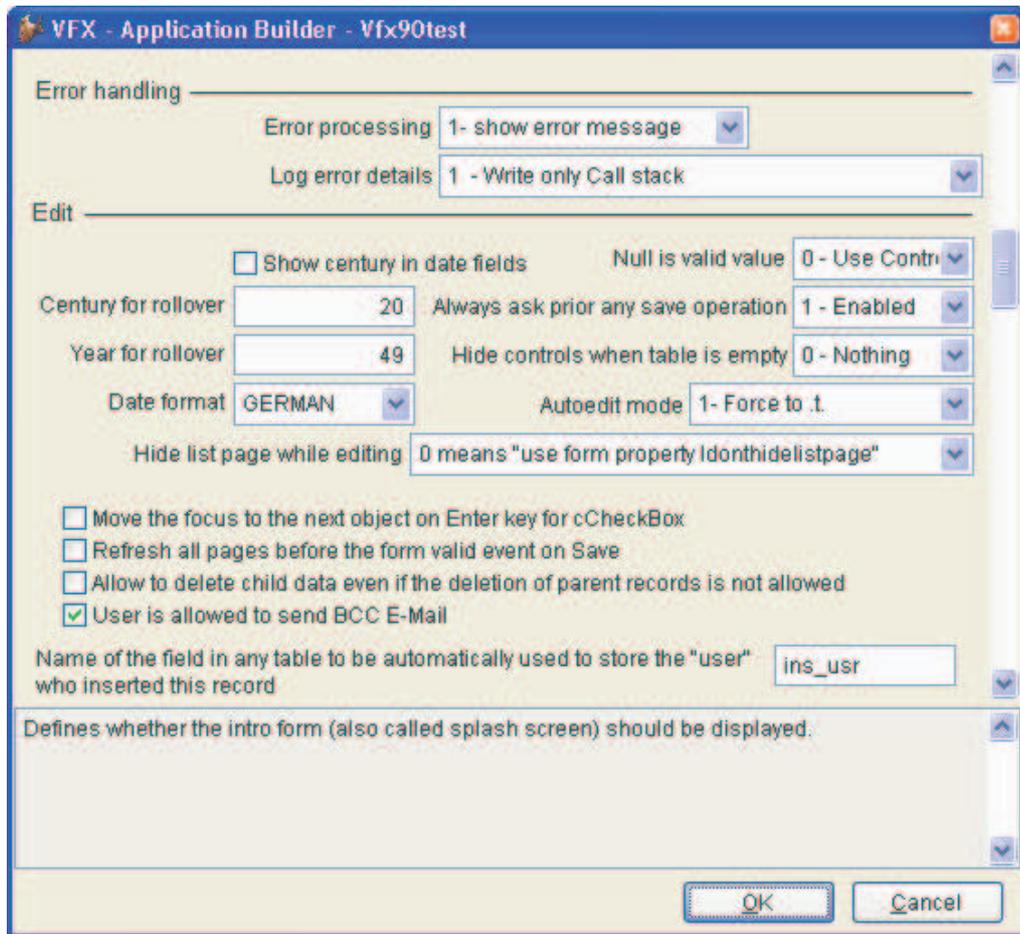


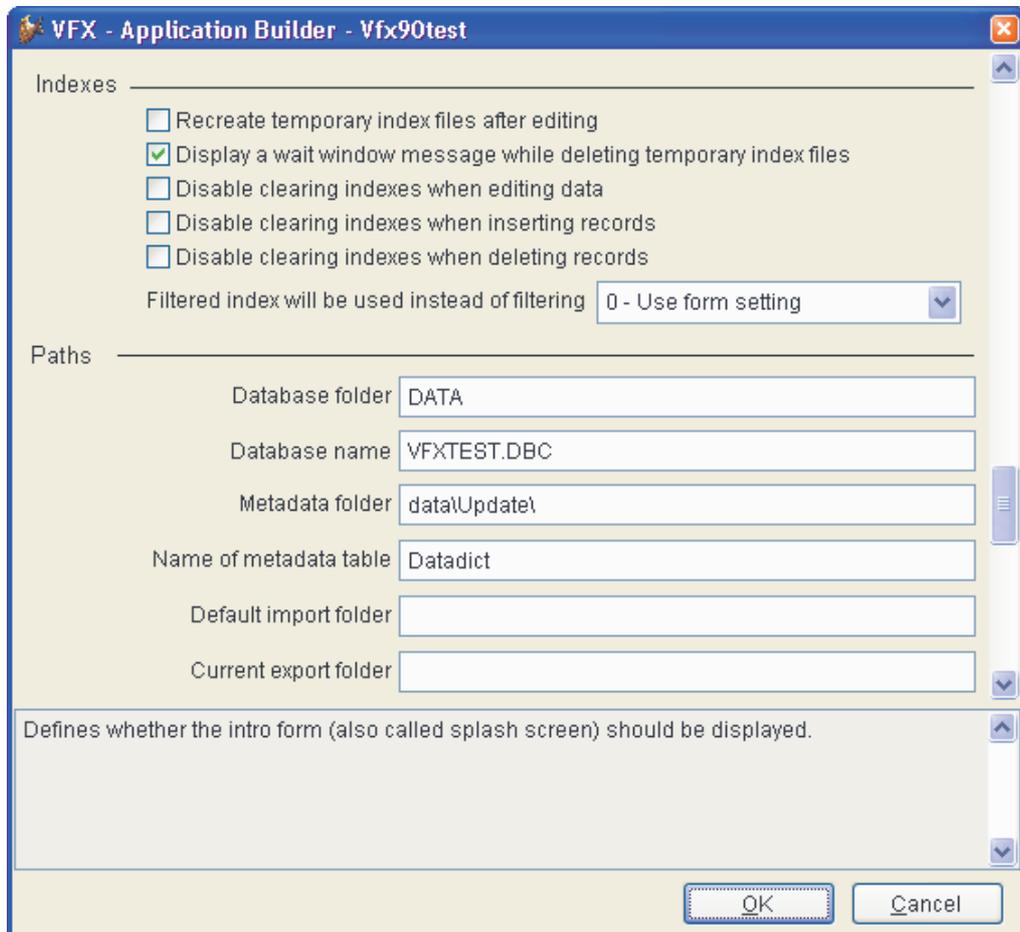
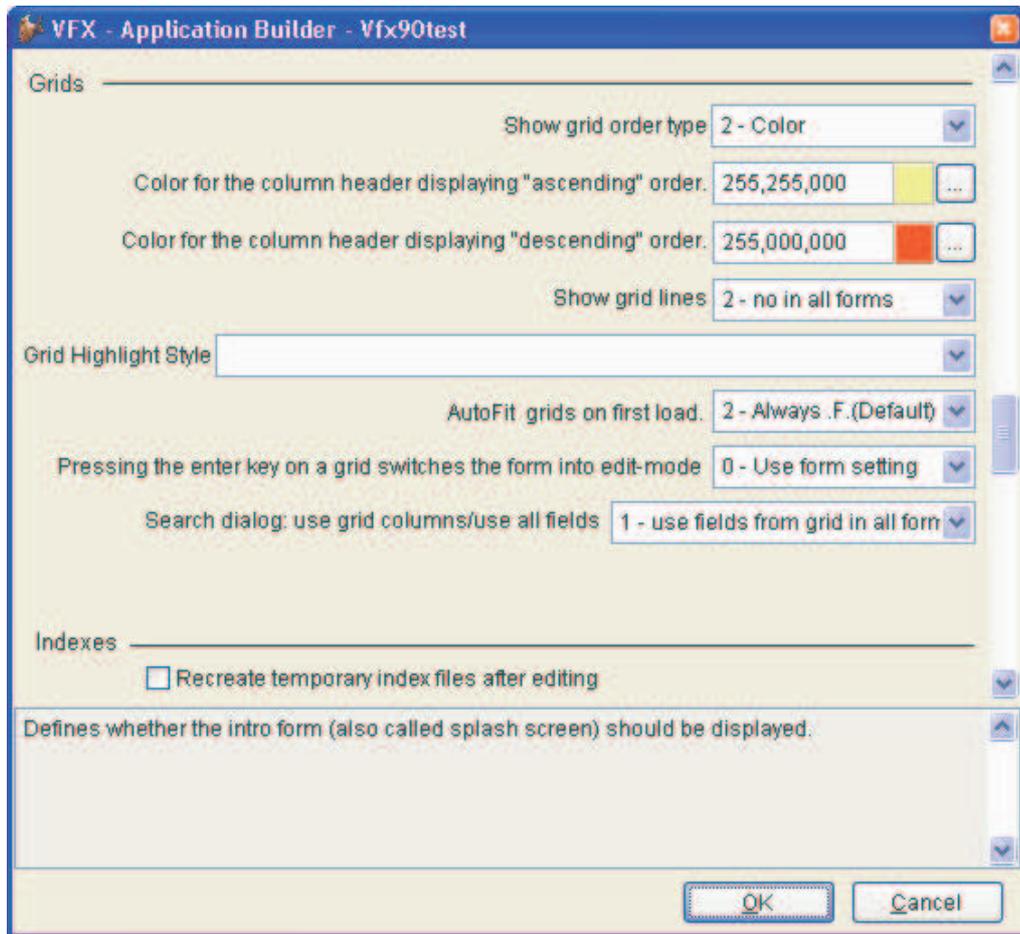
Auf der Seite *Options* des VFX - Application Wizard befindet sich eine neue Schaltfläche *Advanced*. Durch einen Klick auf diese Schaltfläche erscheint ein Dialog, indem alle Eigenschaften des Anwendungsobjekts eingestellt werden können. Im unteren Teil dieses Dialogs wird ein Hilfetext mit einer Erklärung zur aktuellen Eigenschaft angezeigt.

Dieser Dialog kann später jederzeit über den Menüpunkt *Application Builder* aufgerufen werden, um Einstellungen des Anwendungsobjekts zu ändern.

Anmerkung: Die mit dem VFX - Application Builder gemachten Einstellungen werden für das nächste neue Projekt übernommen.







VFX - Application Builder - Vfx90test

Default import folder

Current export folder

Path to the external report files (*.fx)

Save Export files folder per user

Misc _____

Name of Postscript printer to be installed when necessary

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

URL to be used when check for internet connection existence

Password to be used for encrypting config.vfx file

Support URL

Support e-mail

Defines whether the intro form (also called splash screen) should be displayed.

VFX - Application Builder - Vfx90test

Password to be used for encrypting config.vfx file

Support URL

Support e-mail

Author _____

Author:

Company:

Address:

City:

State:

PostalCode:

Country

Support E-mail Address

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 zum Beispiel mit dem VFX CursorAdapter Wizard erstellt werden.

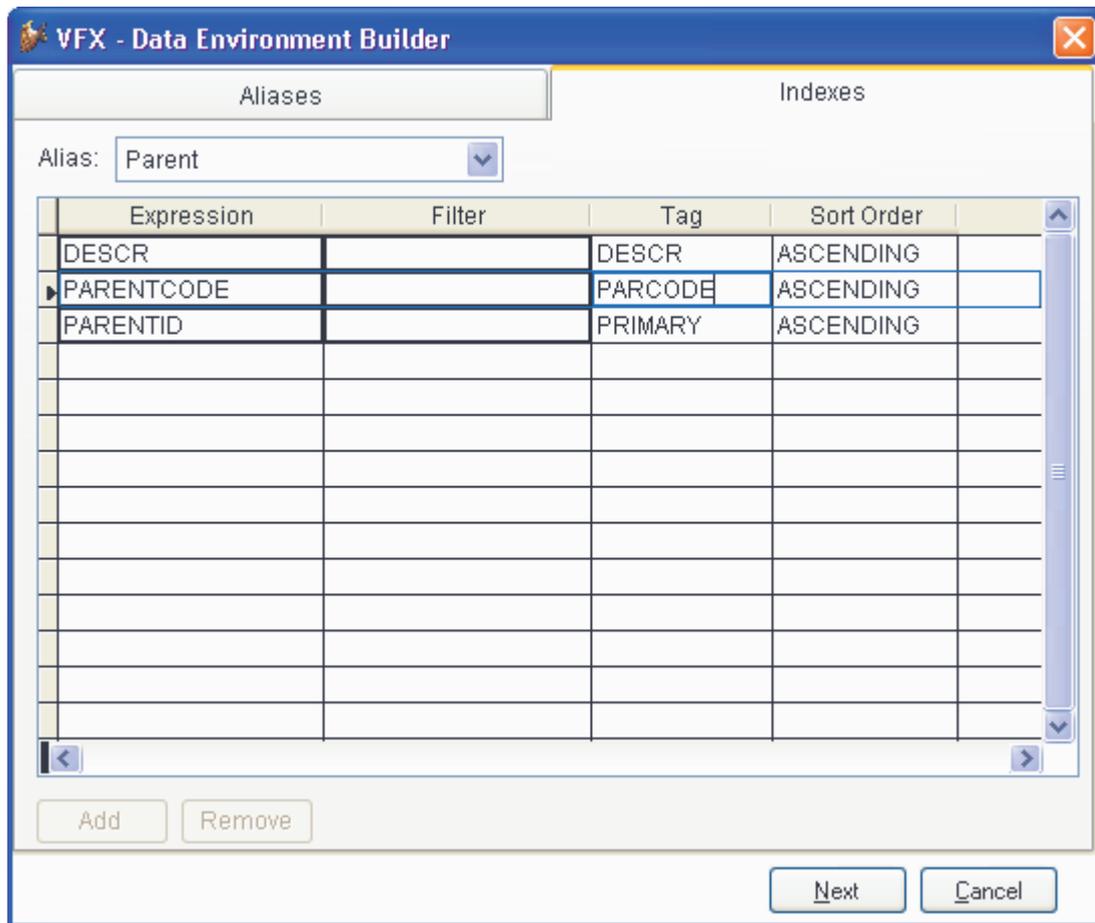
Über die Schaltfläche *New CA* kann ein neues Objekt basierend auf der Klasse *cAppDataAccess* mithilfe des VFP CursorAdapter Builder erstellt werden.

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. Der Indexschlüssel wird zur Laufzeit erstellt, nachdem der CursorAdapter 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.

In der Spalte *Filter* kann ein logischer Ausdruck eingegeben werden, der zur Laufzeit als Filterausdruck verwendet wird. Dieser Ausdruck wird der Eigenschaft *Filter* des Cursor-Objekts zugewiesen.

Die Spalten *Parent Alias* und *Rel Expression* geben die Möglichkeit Relationen zwischen Cursors in der Datenumgebung aufzubauen. Nach Auswahl eines Aliasnamen in der Spalte *Parent Alias*, kann in der Spalte *Rel.Expression* ein Feld für den Relationsausdruck ausgewählt werden oder es kann ein eigener Relationsausdruck eingegeben werden. Zur Laufzeit werden die Beziehungen vom *oRelationMgr*-Objekt verwaltet.



Um zwischen CursorAdapter-Objekten Beziehungen herstellen zu können, müssen temporäre Indexschlüssel zur Laufzeit erstellt werden. Auf der Seite *Indexes* kann der Entwickler die erforderlichen Indexschlüssel erstellen. VFX erstellt die entsprechenden Indexdateien temporär zur Laufzeit und erstellt die Beziehungen, die auf der Seite *Aliases* eingegeben wurden.

Für Cursor-Objekte, die auf Tabellen basieren, werden die zur Verfügung stehenden Indexschlüssel angezeigt. Für CursorAdapter-Objekte können die Indexschlüssel bearbeitet und neue Indexschlüssel hinzugefügt werden.

Für jeden zu erstellenden Indexschlüssel müssen der Tag-Name, der Indexausdruck und die Sortierfolge eingegeben werden. Wenn ein gefilterter Indexschlüssel gewünscht wird, kann der Filterausdruck in der Spalte Filter eingegeben werden.

Weitere neue Eigenschaften der VFX Form Builder

Die VFX Form Builder unterstützen alle neuen Formulareigenschaften von VFX 9.0. Zusätzlich können jetzt viele Funktionen über die Form Builder eingestellt werden, die bisher nur manuell in VFP bearbeitbar waren. Auf neuen Seiten der Form Builder können Ansichtsparemeter, in Beziehung stehende Tabellen, erforderlich Eingabefelder und Felder für Berichte bearbeitet werden.

Edit Pages

The screenshot shows the 'VFX - CDataFormPage Builder' dialog box with the 'Edit Pages' tab selected. The 'Form Name' is 'frmParent' and the 'Caption' is 'Parent'. The 'Page Count' is set to 1, 'Page Title' is 'Page1', and 'Page BackColor' is a light yellow color. The 'Fields List' on the left contains: Parent.parentid (selected), Parent.descr, Parent.checked, Parent.value, Parent.overid, and Parent.parentcode. The 'Control Type' is 'ckeyfield', 'Caption' is 'Parent ID', 'Format' is empty, 'Input Mask' is '999999999', 'Status Bar' is 'Parent ID', 'AutoCompSource' is empty, and 'AutoComplete' is '0 - Does not support AutoComplete'. The 'Read Only' checkbox is unchecked. At the bottom, 'Use DBC Definitions' is unchecked and 'Overwrite Font' is checked. Buttons for 'Back', 'OK', 'Apply', and 'Cancel' are visible.

Im VFX Form Builder können auf der Seite *Edit Pages* alle neuen Formulareigenschaften von VFX 9.0 wie Hintergrundbild oder Hintergrundfarbe für Seiten eines Seitenrahmens, verknüpfte Tabellen und erforderliche Felder sowie AutoComplete-Eigenschaften eingestellt werden.

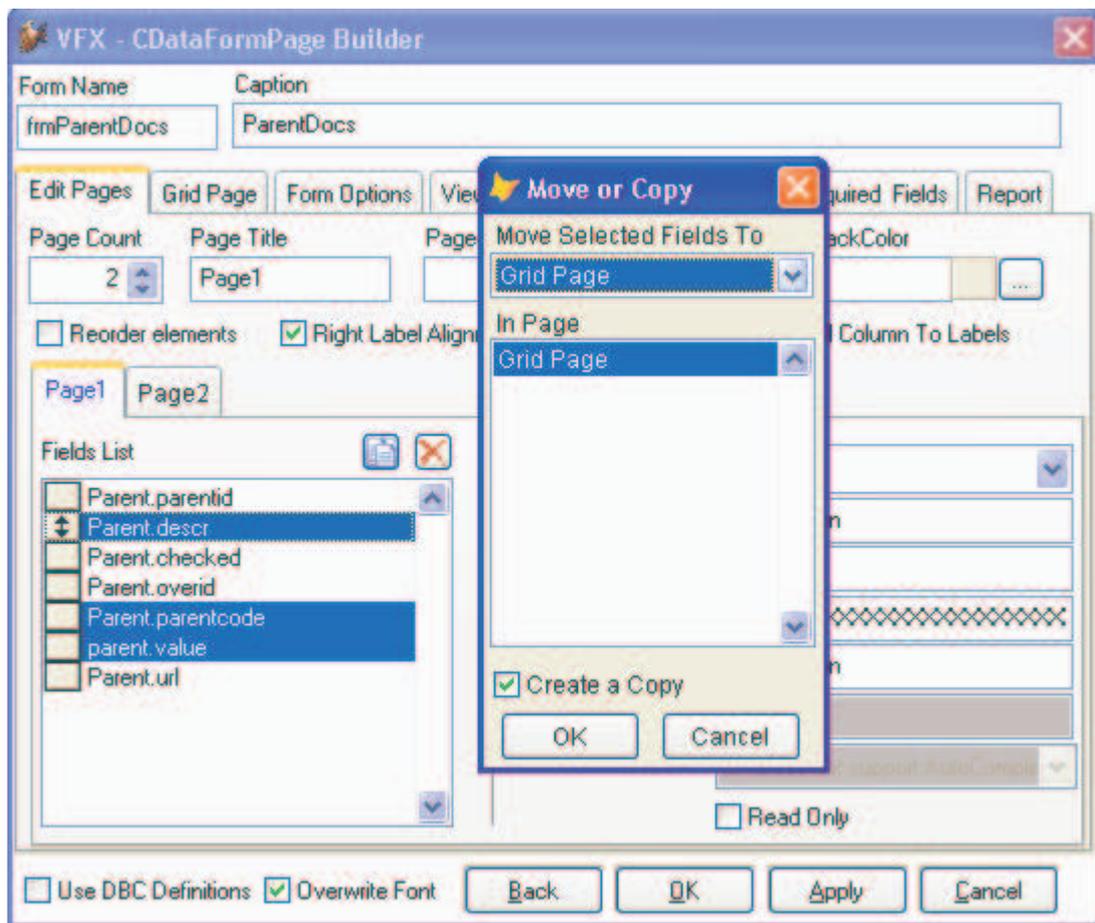
Für alle Textboxen kann die *AutoComplete*-Eigenschaft eingestellt werden und es kann eine *AutoCompSource*-Tabelle zugewiesen werden. AutoComplete-Tabellen müssen mit der Anwendung nicht ausgeliefert werden. Diese Tabellen werden von VFP bei Bedarf automatisch erstellt.

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

Grid Page

The screenshot shows the 'VFX - CDataFormPage Builder' dialog box with the 'Grid Page' tab selected. The 'Form Name' is 'frmParentDocs' and the 'Caption' is 'ParentDocs'. The 'Grid Page Title' is 'List' and the 'Grid Class' is 'cgrid'. The 'Grid Page Picture' field is empty, and the 'Use Grid Page' checkbox is checked. The 'Grid Page BackColor' is set to a light yellow color. The 'Fields Selected' list contains the following items: parent.parentid (selected), parent.descr, parent.date, parent.checked, parent.value, parent.ins_date, parent.ins_usr, parent.edt_date, parent.edt_usr, parent.overid, and parent.parentcode. The 'Control Type' is 'textbox', the 'Header' is 'Parent ID', the 'Control Source' is 'parent.parentid', and the 'Output Mask' is '999999999'. The 'Read Only' and 'Incremental Search' checkboxes are checked. At the bottom, there are checkboxes for 'Use DBC Definitions' (unchecked) and 'Overwrite Font' (checked), and buttons for 'Back', 'OK', 'Apply', and 'Cancel'.

Bei der Bearbeitung vorhandener Formulare ist die neue 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.

Zusätzlich gibt es auf dem VFX Form Builder vier neue Seiten um die neuen Eigenschaften der VFX Formularklassen bearbeiten zu können.

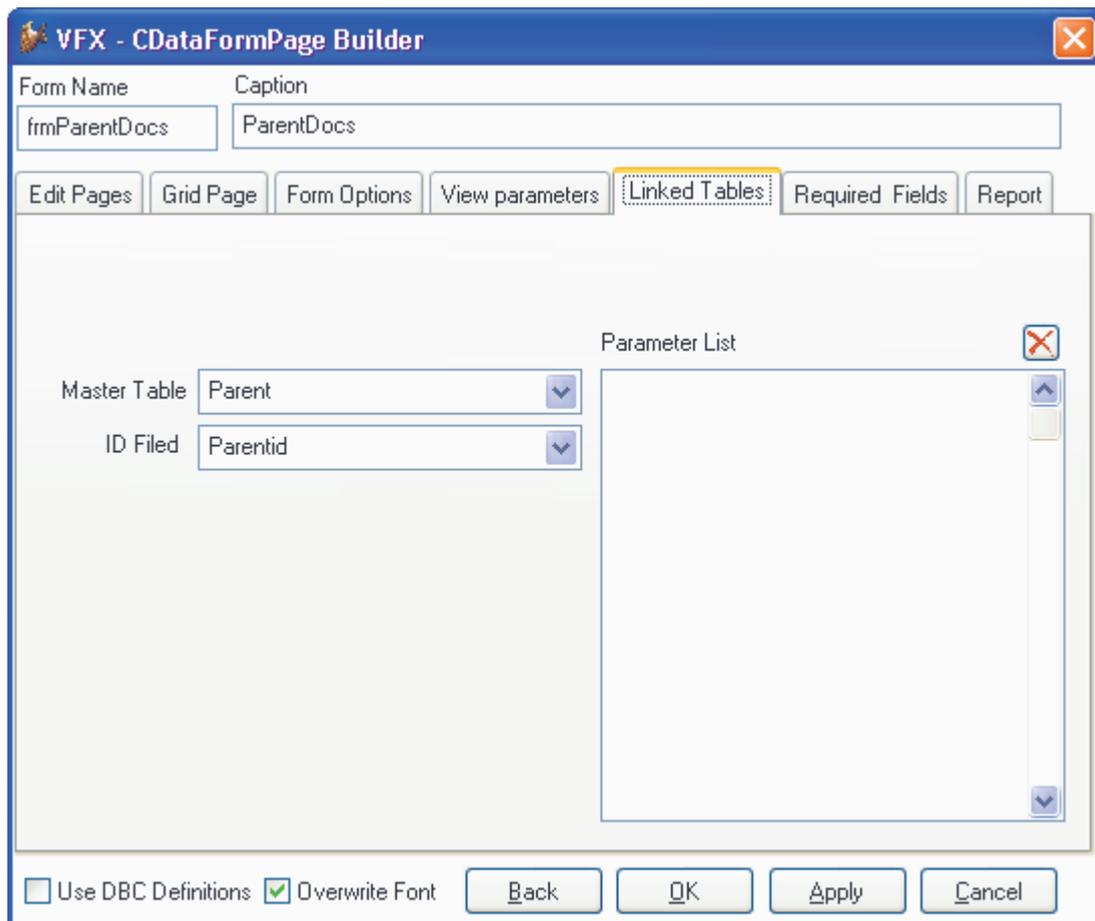
View Parameters

The screenshot shows the 'View Parameters' dialog box in the VFX - CDataFormPage Builder. The dialog has a title bar with a close button. Below the title bar, there are two text boxes: 'Form Name' containing 'frmParentDocs' and 'Caption' containing 'ParentDocs'. A row of buttons includes 'Edit Pages', 'Grid Page', 'Form Options', 'View parameters' (which is highlighted), 'Linked Tables', 'Required Fields', and 'Report'. The main area is divided into two sections. On the left, a 'Parameter List' contains a single entry 'Parent.overid'. On the right, there are several input fields: 'Parameter Name' (with a checked 'Reorder elements' checkbox above it), 'Caption' (containing 'Overid'), 'Format', 'Input Mask' (containing '999999999'), and 'Status Bar'. At the bottom, there are two checkboxes: 'Use DBC Definitions' (unchecked) and 'Overwrite Font' (checked), followed by 'Back', 'OK', 'Apply', and 'Cancel' buttons.

Auf der Seite *View Parameters* können Steuerelemente zur Eingabe von Ansichtsparametern angelegt werden. Ähnlich wie auf Formularen basierend auf der Klasse *cAskViewArg* kann der Benutzer hier zur Laufzeit Werte eingeben. Über eine *Requery*-Schaltfläche in der Standardsymbolleiste kann die Ansicht aktualisiert werden. Auf diesem Weg entfällt die Instanziierung eines weiteren Formulars.

Die Steuerelemente zur Eingabe von Ansichtsparametern werden am oberen Rand des Formulars, oberhalb des Seitenrahmens, platziert. Diese Steuerelemente sind immer sichtbar. Für diese Steuerelemente muss der Name eines Ansichtsparameters anstelle einer *ControlSource* angegeben werden.

Linked Tables

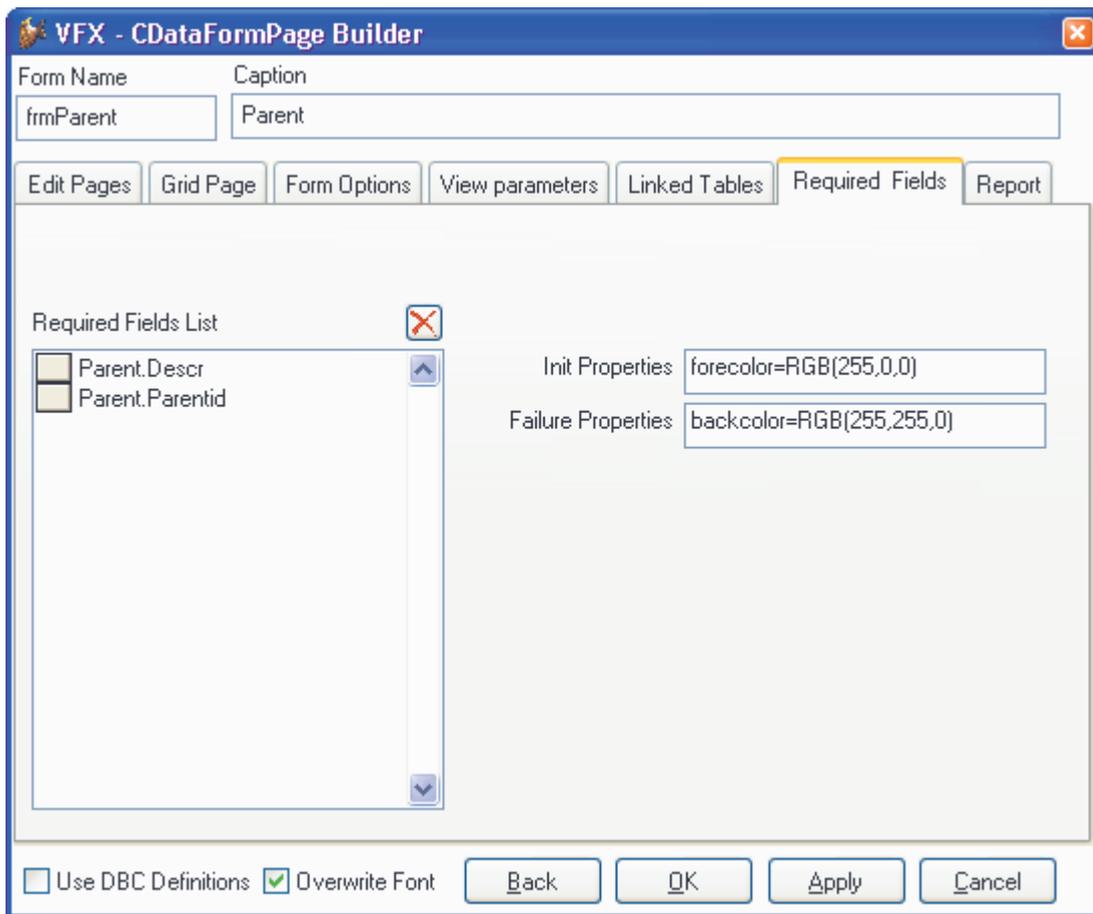


VFX 9.0-Anwendungen unterstützen 1:1-Beziehungen zwischen der Hauptbearbeitungstabelle und weiteren Tabellen. Hierdurch bekommt der Entwickler eine größere Flexibilität bei der Entwicklung komplexer Datenbanken ohne zusätzlichen Code zur Gewährung der Integrität der Datenbank schreiben zu müssen. VFX 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.

Required Fields



Mithilfe der neuen 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 ]
```

eingegeben 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:

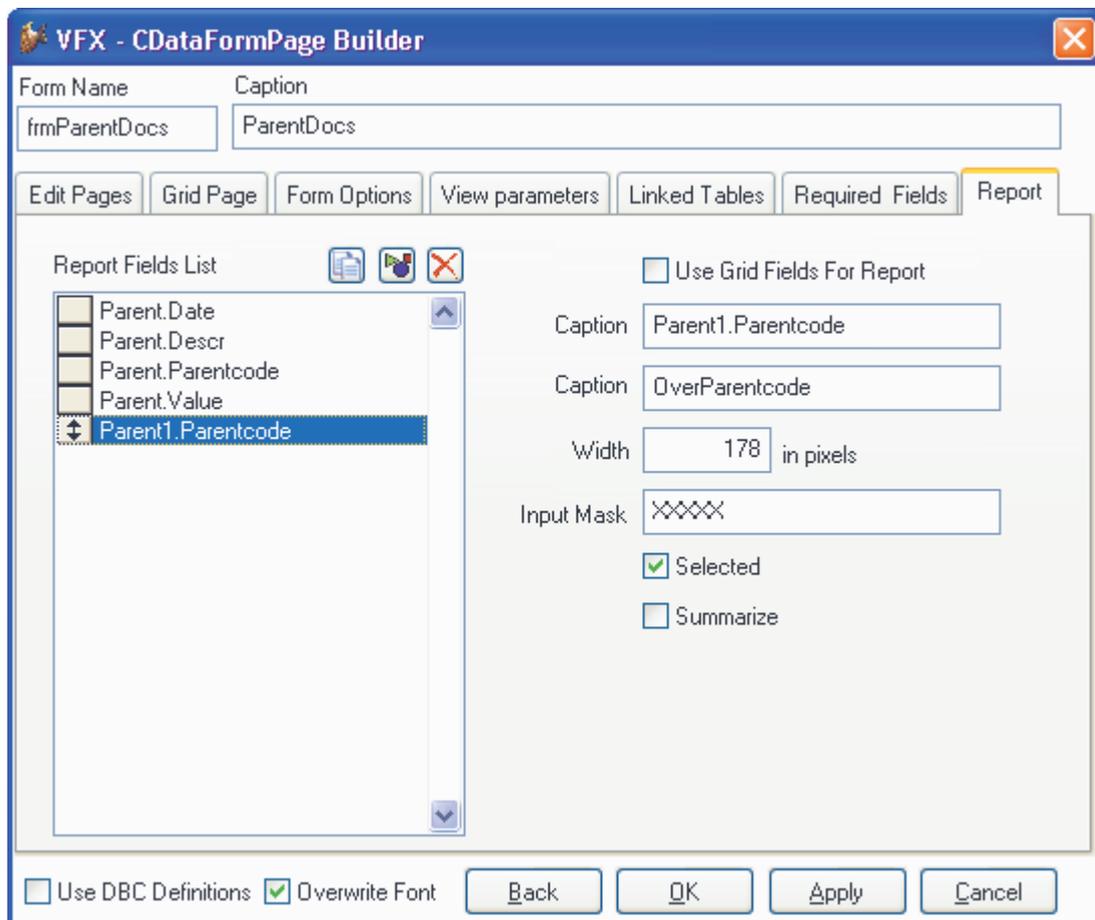
```
FontBold = .T.; ForeColor = RGB(255,0,0); BackColor = RGB(255,255,196)
```

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.

Report



Häufig ist es erforderlich auf Berichten Felder zu drucken, die auf der Listenseite eines Formulars nicht zur Verfügung stehen. Genau 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.

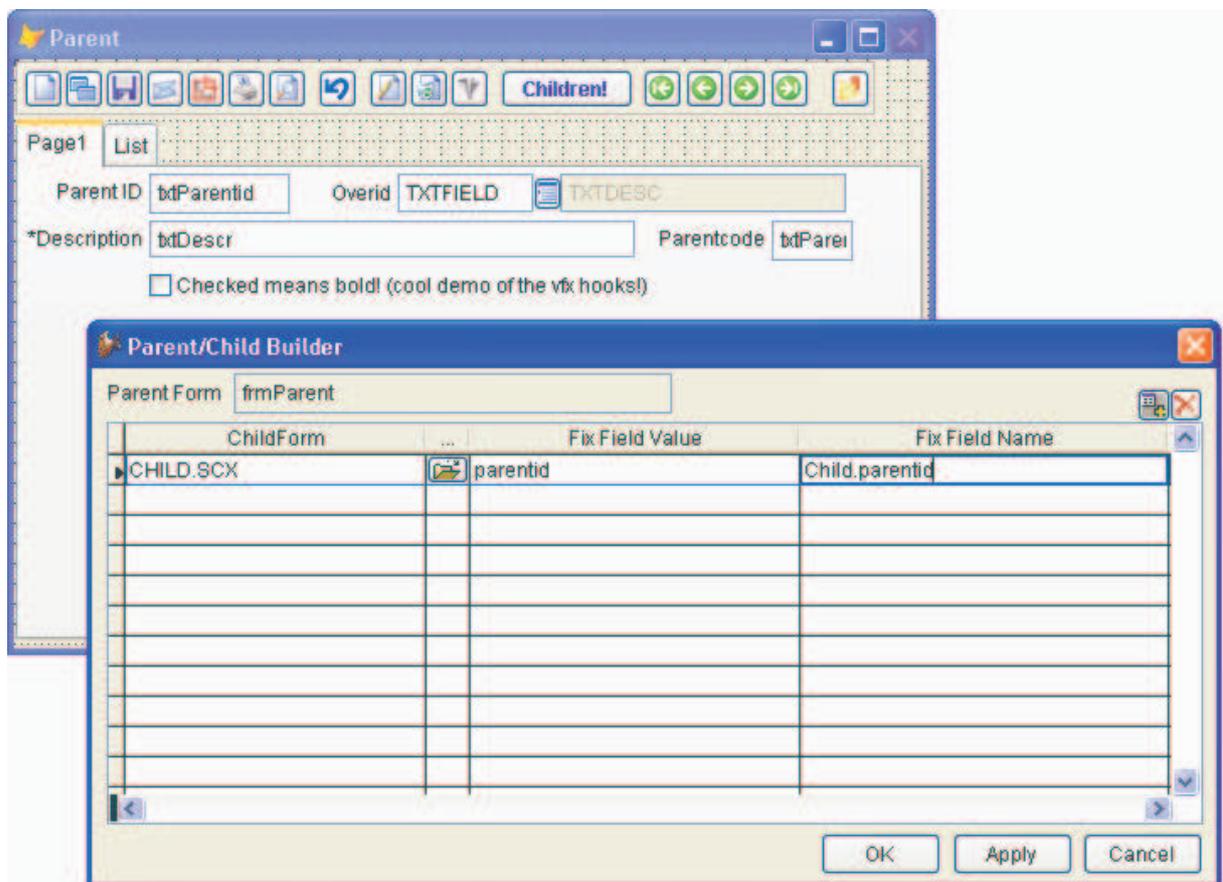
Für jedes Feld können die Breite des Feldes, eine Eingabemaske und eine Überschrift vorgegeben werden.

Wenn, wie in früheren VFX-Versionen, alle Felder des Grids im Suchdialog verwendet werden sollen, muss das Kontrollkästchen *Use Grid Fields For Report* markiert werden.

VFX Parent/Child Builder

In VFX 9.0 gibt es einen Builder zur Bearbeitung von Parent/Child-Beziehungen. Zur einfacheren Verwaltung von Parent/Child-Beziehungen gibt es die neue Klasse *cChildManager*. Zur Verwendung des VFX - Parent/Child Builder muss zunächst das Parent-Formular im VFP Formular-Designer geöffnet werden. Dann kann der VFX - Parent/Child Builder aus dem VFX 9.0 Menü gestartet werden.

Im Builder können beliebig viele Child-Formulare verwaltet werden.



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.

Die Klasse *cComboPicklist*

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 *Descript* 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 *Descript*-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.

Eigenschaft:

nParentID – ID Schlüsselwert der Tabelle *Vfxpdef.dbf*.

Methode:

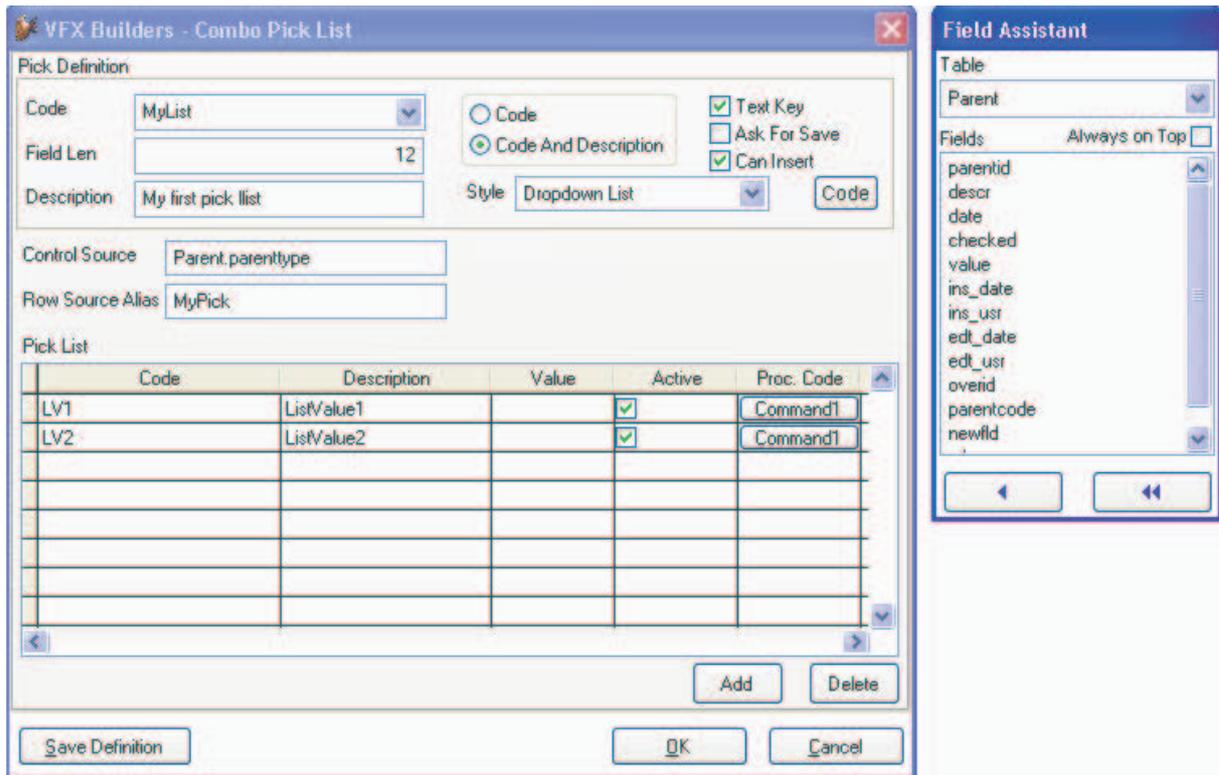
Addnewcode – Diese Methode wird ausgeführt wenn der Benutzer einen neuen Wert in die Combobox einträgt. Wenn beim Hinzufügen von Werten weiterer Code ausgeführt werden soll, muss er in dieser Methode eingetragen werden.

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.

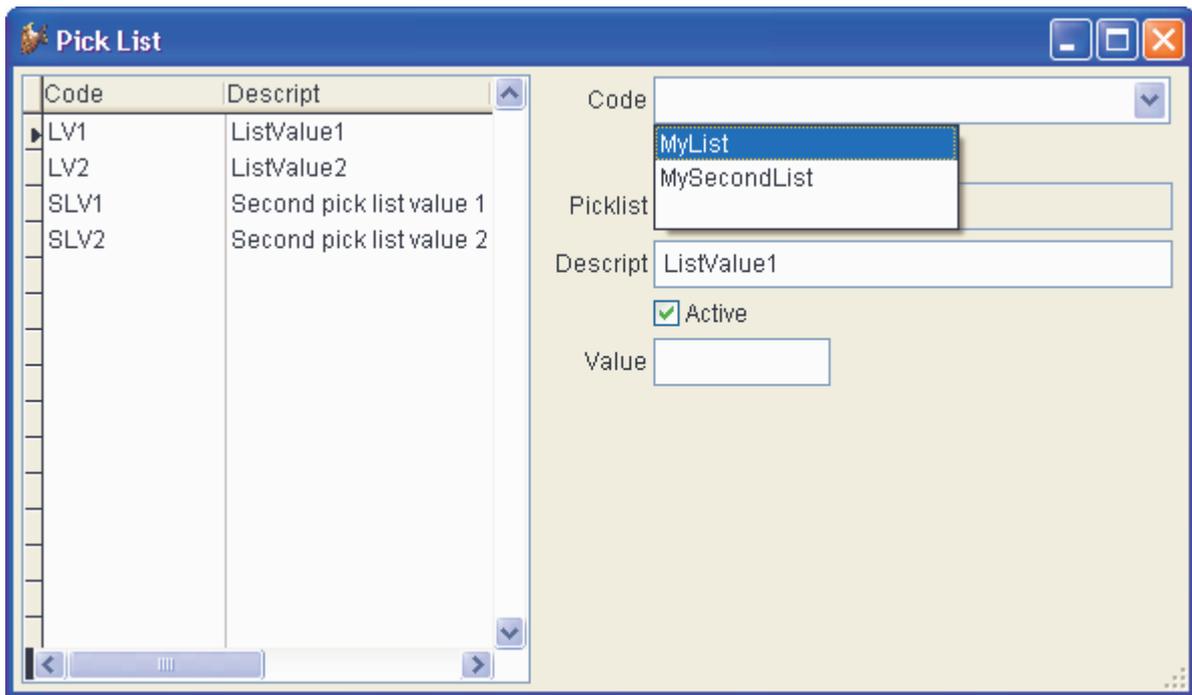
VFX – Combo Pick List Builder



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 Controlsourc 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.

Das Formular zur Bearbeitung von Auswahllisten



Dieses Formular kann Anwendern zur Bearbeitung von Auswahllisten zur Verfügung gestellt werden. Das Formular befindet sich in jedem VFX 9.0-Projekt und hat den Namen *VFXPlist.scx*.

Der Benutzer kann zwischen der Bearbeitung aller Datensätze wählen oder den sichtbaren Bereich durch einen Filter auf das Feld *Code* einschränken. Es ist möglich Datensätze zu löschen, aber Datensätze können auch als nicht aktiv markiert werden.

Erweiterungen in OneToMany-Formularen

In Formularen, basierend auf den Klassen *cOnetomany* und *cTreeviewonetomany* gibt es einige Verbesserungen.

- Die Schaltflächen zum Einfügen und Löschen von Child-Daten sind nur dann enabled, wenn sich das Formular im Bearbeitungsmodus oder im Einfügemodus befindet.
- Der Child-Teil kann jetzt auch andere Steuerelemente als nur ein Childgrid enthalten.
- Bearbeitungsseiten im Child-Teil von Onetomany-Formularen können mit dem Form Builder genauso erstellt werden, wie Bearbeitungsseiten im Parent-Teil.

Die Klasse *cChildgrid*, die auf allen Onetomany-Formularen zur Bearbeitung der Child-Daten verwendet wird, wurde um einige Funktionen erweitert.

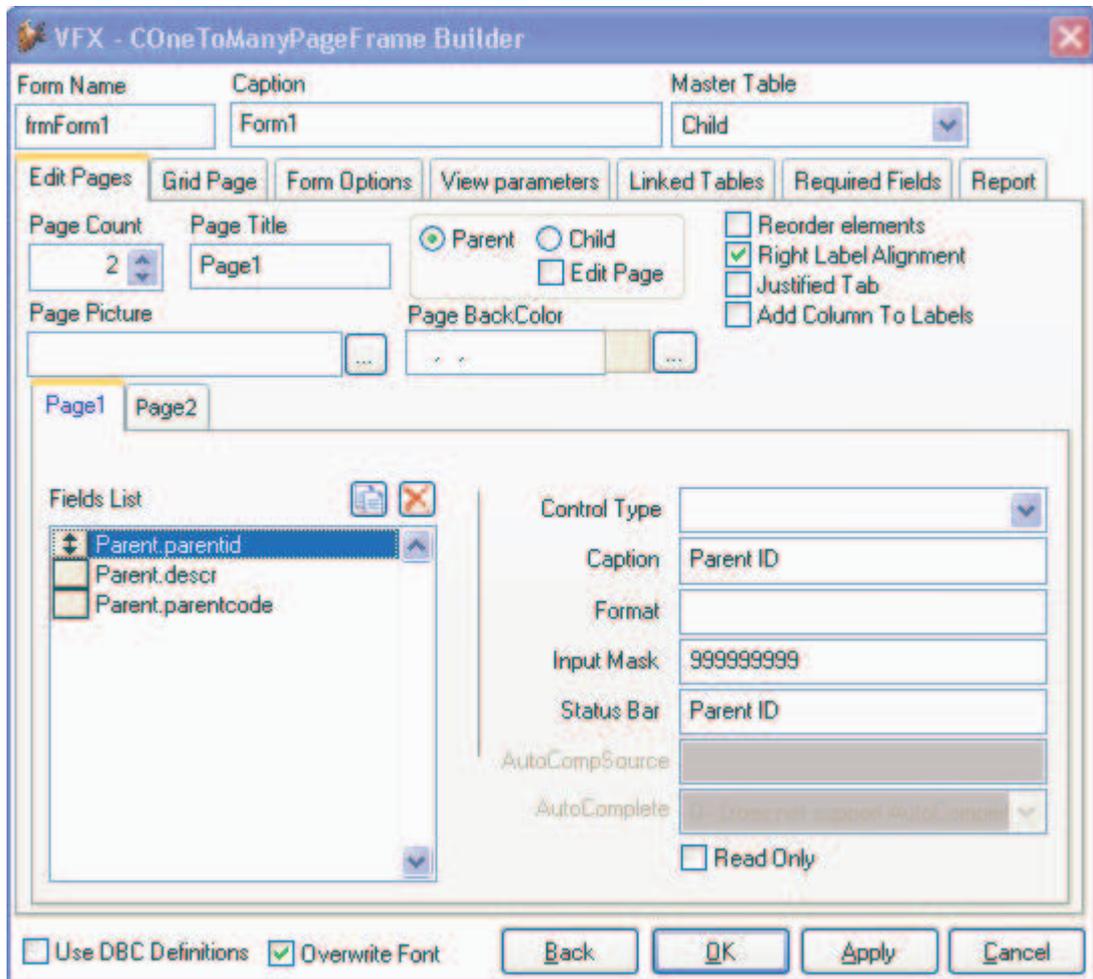
- Wenn die Child-Daten auf einer Ansicht oder auf einem Cursoradapter basieren, kann jetzt in den Child-Daten inkrementell gesucht werden.
- Ein Klick in den leeren Bereich eines Child-Grids fügt einen neuen Child-Datensatz an.

Die Formulkasse COneToManyPageFrame

Die neue 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.

VFX - COneToManyPageFrame Builder



Zusätzlich zu den Einstellungen, die der Entwickler im Form Builder für andere Formulklassen 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.

Fehlerbehandlung

In VFX 9.0 ist eine erweiterte Behandlung von Laufzeitfehlern implementiert. Das Laufzeitfehlerprotokoll kann vom Kunden jetzt per E-Mail an den Entwickler gesendet werden. Der Kunde wird über den Inhalt des Fehlerberichts informiert. Der Versand des Fehlerberichts als E-Mail an den Entwickler ist der schnellste Weg Probleme in einer Anwendung zu lokalisieren und zu beseitigen. Die E-Mailadresse des Entwicklers wird der Eigenschaft *goProgram.csupportemail* zugewiesen. Der Wert dieser Eigenschaft kann mit dem VFX Application Builder bearbeitet werden.

Erstellen von Aktivierungsschlüsseln

VFX 9.0-Anwendungen können vor unbefugter Nutzung mit einem Aktivierungsschlüssel geschützt werden. Die Daten der Kunden, die einen Aktivierungsschlüssel erhalten haben, können mit umfangreichen Benutzerdaten und Benutzerrechten verwaltet werden.

dung beim Kunden registriert daher die aktuelle IP-Adresse des Kunden-PCs als Subdomain bei DynDNS. So kann der Supporter den Kunden-PC über einen Subdomain-Namen im Internet finden.

Voraussetzungen

Der Entwickler muss die VFX-Anwendung zunächst für die Fernwartung vorbereiten. Dafür muss zunächst eine Subdomain bei DynDNS für den Support der eigenen Anwendung angemeldet werden. Die Anmeldung ist kostenlos.

Die Registrierungsinformationen werden in der VFX-Anwendung in der Tabelle *Vfxsys.dbf* im Memofeld *dyndns* verschlüsselt gespeichert, damit die Registrierungsinformationen auf dem Kunden-PC nicht einsehbar sind. Die Verschlüsselung erfolgt mit dem Kennwort *cconfigpassword*. Dieses Kennwort muss in *Appl.vcx – cfoxappl* in der Eigenschaft *cconfigpassword* eingetragen werden.

Die Bearbeitung der DynDNS-Registrierungsinformationen erfolgt über den Menüpunkt *Edit Vfxsys* im *VFX 9.0*-Menü.

Der Inhalt des Memofeldes *dyndns* besteht aus vier Zeilen.

1. Benutzername bei DynDNS
2. Kennwort bei DynDNS
3. Subdomain-Name
4. Kennwort für den Radmin-Zugriff auf den Kunden-PC

Registrierung einer Subdomain

Über die Organisation Dynamic DNS Network Services ist es möglich kostenlos Subdomains zu registrieren. Jeder Entwickler sollte bei <http://www.dyndns.org/services/dyndns> eine dynamische DNS registrieren.

Für die Erstellung eines Kontos bei DynDNS sind ein Benutzername, ein Kennwort und eine E-Mailadresse erforderlich. Der Subdomain-Name kann beliebig gewählt werden. Es kann aus einer Vielzahl von Domain-Namen ausgewählt werden.

Beispiel: *meineFirma.dnsalias.com*

In diesem Beispiel ist *meineFirma* der selbst gewählte Subdomain-Name. *Dnsalias.com* ist der von DynDNS bereitgestellte Domain-Name.

Bei der Registrierung der Subdomain muss ein Benutzerkonto mit Benutzernamen und Kennwort angelegt werden. Mit den Anmeldedaten kann das Konto konfiguriert werden. Die Anmeldedaten sind auch in die obige URL einzusetzen.

Die dem Domain-Namen zugehörige IP-Adresse kann man beliebig oft und mit verschiedenen Methoden ändern. Ausführliche Beschreibungen zu allen Methoden finden sich auf der Website www.dyndns.org.

Die VFX-Anwendung ruft eine URL auf, um die aktuelle IP-Adresse des Kunden-PCs zu registrieren. Die URL hat das folgende Format:

```
http://benutzername:kennwort@members.dyndns.org/nic/update?hostname=meineFirma.dnsalias.com
```

Wenn man diese URL im Internet-Explorer eingibt, erhält man als Antwort eine HTML-Seite mit dem Wort „Good“.

Da der Internet-Browser die eigene IP-Adresse an den Server übermittelt, muss die IP-Adresse nicht gesondert angegeben werden. Der Internet-Server muss ja wissen an welche Adresse er die Antwort zurückschicken muss. DynDNS benutzt also automatisch diese IP-Adresse für die Registrierung der Subdomain.

Das Fernwartungsprogramm Radmin

Das Fernwartungsprogramm Radmin kann von der Website www.radmin.com herunter geladen werden. Auf dieser Website befindet sich auch die Dokumentation.

Radmin ist Shareware und kann kostengünstig registriert werden. Die Vollversion, die für den Supporter-Arbeitsplatz notwendig ist, kostet zurzeit 35 US\$. Eine Lizenz für einen Kunden kostet 15 US\$. Kundenlizenzen können nur in Paketen ab 50 Lizenzen erworben werden.

Ähnlich wie VFX ist auch Radmin über einen Aktivierungsschlüssel geschützt.

Wenn der Kunde die Fernwartung benutzen will, kann Radmin sofort verwendet werden. Wenn nach der 30-tägigen Testphase versucht wird eine Verbindung aufzubauen, wird der Supporter aufgefordert einen Registrierungsschlüssel zum Kundenrechner zu übertragen.

Der Registrierungsschlüssel kann während der Radmin-Verbindung vom Supporter an den Kundenrechner übertragen werden.

Neben der Fernwartung bietet Radmin die Möglichkeit zur Dateiübertragung.

Die Fernwartung aus der Sicht des Supporters

Der Kunde sollte die Fernwartung nur nach Rücksprache mit dem Supporter starten. Das Fernwartungsprogramm ermöglicht den uneingeschränkten Zugriff auf den Kunden-PC und stellt für den Kunden damit ein erhebliches Sicherheitsrisiko dar!

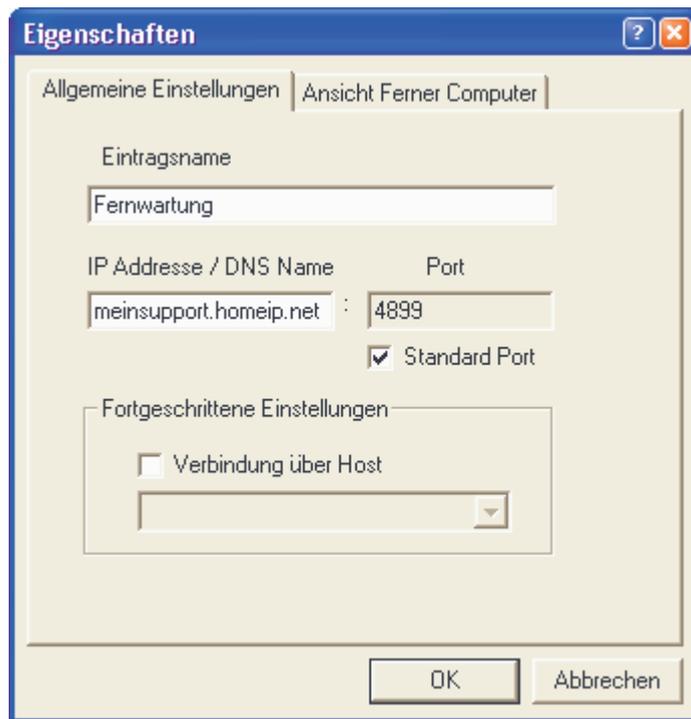
Der Zugriff auf den Kunden-PC sollte daher durch ein Kennwort geschützt werden.

Es ist nicht sehr wahrscheinlich, dass ein wartender Radmin-Server an einer dynamisch zugeteilten IP-Adresse im Internet von Hackern schnell gefunden wird. Zusätzlich ist der Zugriff auf den Kunden-PC durch ein Kennwort geschützt, das beim Verbindungsaufbau vom Supporter zum Kunden-PC eingegeben werden muss.

Im Remote Administrator Viewer wird ein Eintrag für den Support der Anwendung gemacht.



In den Eigenschaften des Remote-Eintrags wird im Feld IP-Adresse der Subdomain-Name eingetragen.

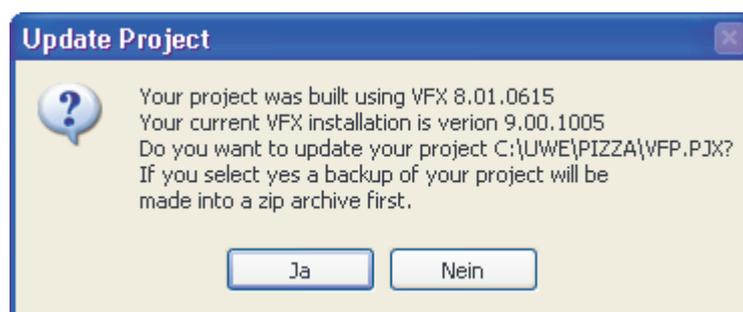


Der Kunden-PC kann jetzt über den Subdomain-Namen im Internet gefunden werden. Der Supporter braucht also nur einen einzigen Eintrag zur Fernwartung aller Kundenrechner.

Nach erfolgreicher Verbindungsherstellung kann der Kunden-PC im Fenster des Radmin-Viewers genau wie der eigene PC bedient werden.

VFX Project Update Wizard

Projekte, die mit älteren Versionen von VFX oder mit älteren Builds von VFX 9.0 erstellt wurden, können jetzt automatisch auf die neueste Version aktualisiert werden.



Der VFX Project Update Wizard kann direkt aus dem VFX 9.0-Menü über den Menüpunkt Update Project gestartet werden. Der VFX Project Update Wizard vergleicht die Version des geöffneten Projekts mit der installierten VFX 9.0-Version. Wenn das Projekt mit einer älteren VFX-Version erstellt wurde, wird der Entwickler gefragt, ob das Projekt aktualisiert werden soll.

Der VFX Project Update 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. Eventuelle neue Felder in den freien VFX-Tabellen werden automatisch ergänzt und mit Werten gefüllt.

Vor der Aktualisierung wird automatisch eine Sicherungskopie des Projekts gemacht.

VFX Task Pane

In der VFX 9.0 Task Pane gibt es zwei neue Eigenschaften.

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.

Die zweite Neuerung ist der in die Task Pane integrierte VFX Project Update Wizard. Dieser Wizard wird automatisch gestartet, wenn ein Projekt geöffnet wird, das mit einer älteren VFX-Version erstellt wurde. Es er scheint eine Messagebox in der der Entwickler gefragt wird, ob das Projekt aktualisiert werden soll.

VFP 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.

Visual Extend Menü-Designer

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

Registrierung und Aktivierung von VFX 9.0

Wie bisherige Versionen von VFX ist auch VFX 9.0 über eine Produktaktivierung geschützt. Die Aktivierung von VFX 9.0 erfolgt mit einem Web Service. Der Vorteil ist, dass der Aktivierungsschlüssel unmittelbar an den Entwickler-PC gesendet wird und manuelle Tätigkeiten bei zur Eingabe des Schlüssels entfallen.

In einem Registrierungsformular kann der Entwickler seine persönlichen Daten eingeben. Diese Daten werden nach einem Klick auf die Schaltfläche *Register online* an den Web Service übermittelt und in der VFX-Kundendatenbank gespeichert.

Wenn eine Aktivierung mit dem Web Service nicht möglich sein sollte, kann der Aktivierungsschlüssel manuell eingegeben werden.

Wenn VFX 9.0 mit einem 30-Tage-Testschlüssel betrieben wird, wird in einem Dialog die Restlaufzeit in Tagen angezeigt. Über die Schaltfläche *Buy VFX* wird die Website von Visual Extend angezeigt und es kann online eine Lizenz erworben werden.

PDM

Eine speziell für VFX entwickelte Version des Projekt- und Datenbank-Dokumentations-Tools PDM wird mit VFX geliefert. Das PDM kann über das VFX 9.0-Menü gestartet werden und fertigt zu einem Projekt vollautomatisch eine vollständige technische Dokumentation an. Die Dokumentation wird im HTML-Format erstellt und enthält zahlreiche Querverweise.

Weitere Verbesserungen für Entwickler

- Aufruf aller VFX Form Builder auch vom Pageframe ausgehend möglich.
- Unterstützung von Ansichten und Cursoradapter bei der Anzeige des Audit Trails.
- Unterstützung von allen Steuerelementklassen in Buildern.
- Als Trennzeichen in allen VFX-Eigenschaften können jetzt wahlweise Komma oder Semikolon verwendet werden.
- Zusätzliche Felder *cins_time* und *cedt_time* zur Speicherung der letzten Bearbeitungszeit.
- Wenn *readonly=.T.* eingestellt ist, wird automatisch *tabstop=.F.* eingestellt.
- VFX Cpickfield Builder: die Eigenschaften *cfieldlist* und *cfieldtitle* sind auf dem Builder mit einer einfachen Textbox direkt erreichbar.
- VFX-Tabellen können wahlweise in einer SQL-Datenbank gespeichert werden.
- Neuer Builder zur Generierung von Audit-Trail-Triggern im DBC.

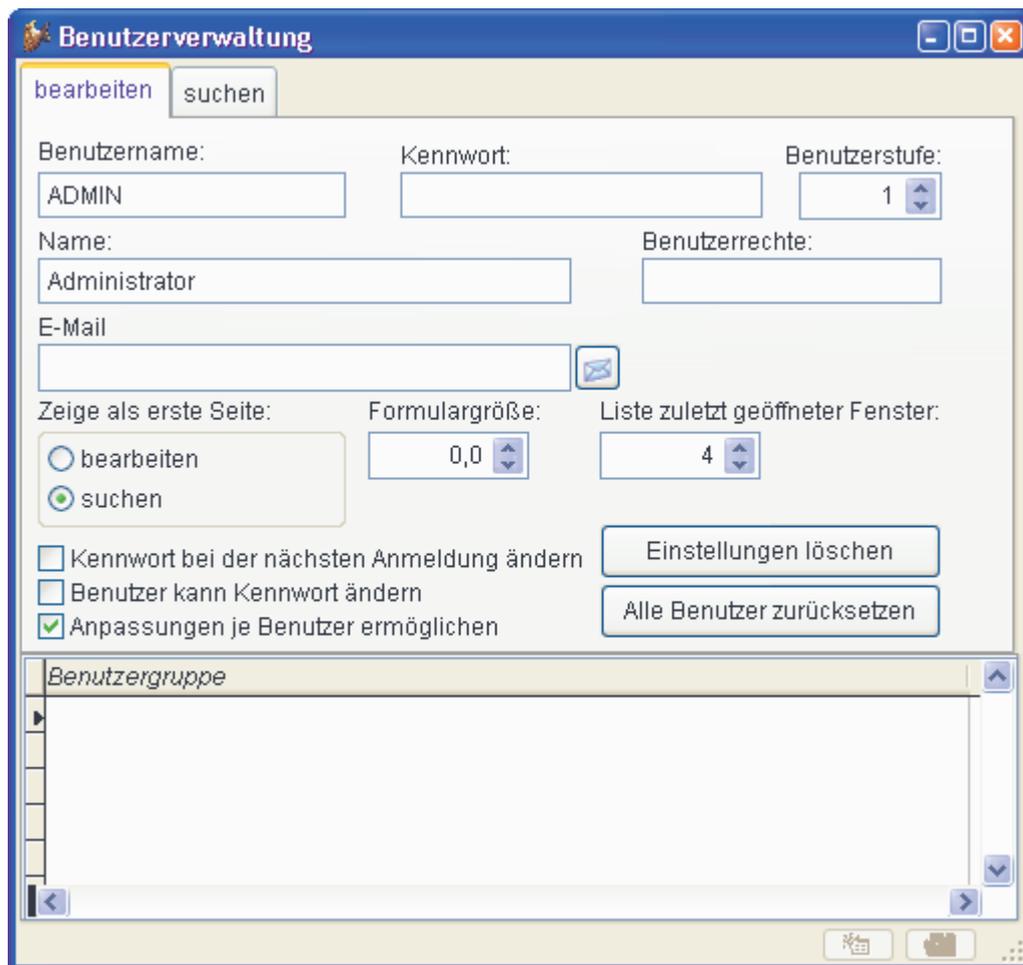
Neue Eigenschaften für Endbenutzer

Benutzerverwaltung

Das Formular Benutzerverwaltung

Die Benutzerverwaltung wurde in VFX 9.0 stark erweitert. Der Administrator kann jetzt mit der Schaltfläche „Alle Benutzer zurücksetzen“ die Ressourcen für alle Benutzer zurücksetzen.

Für jeden Benutzer kann der Administrator einstellen, dass das Kennwort bei der nächsten Anmeldung geändert werden muss. Der Administrator kann auch einstellen, dass ein Benutzer sein Kennwort nicht ändern kann.



Benutzer haben erweiterte Möglichkeiten ihre Umgebung anzupassen. Der Entwickler kann es Benutzern erlauben ihre Umgebungseinstellungen zu ändern, indem die Eigenschaft *lAllowUserCustomization* des Anwendungsobjekts auf *.T.* eingestellt wird.

```
goProgram.lAllowUserCustomization=.T.
```

Wenn diese Eigenschaft auf *.T.* eingestellt ist, kann der Administrator allen Benutzern erlauben die Umgebungseinstellungen zu ändern. Wenn diese Eigenschaft auf *.F.* eingestellt ist, ist das Kontrollkästchen *Anpassungen je Benutzer ermöglichen* für den Administrator nicht sichtbar und die Umgebungseinstellungen können in der Anwendung grundsätzlich nicht eingestellt werden.

Wenn der Administrator anderen Benutzern nicht erlaubt Umgebungseinstellungen anzupassen, gelten die Einstellungen des Administrators für alle Benutzer der Anwendung.

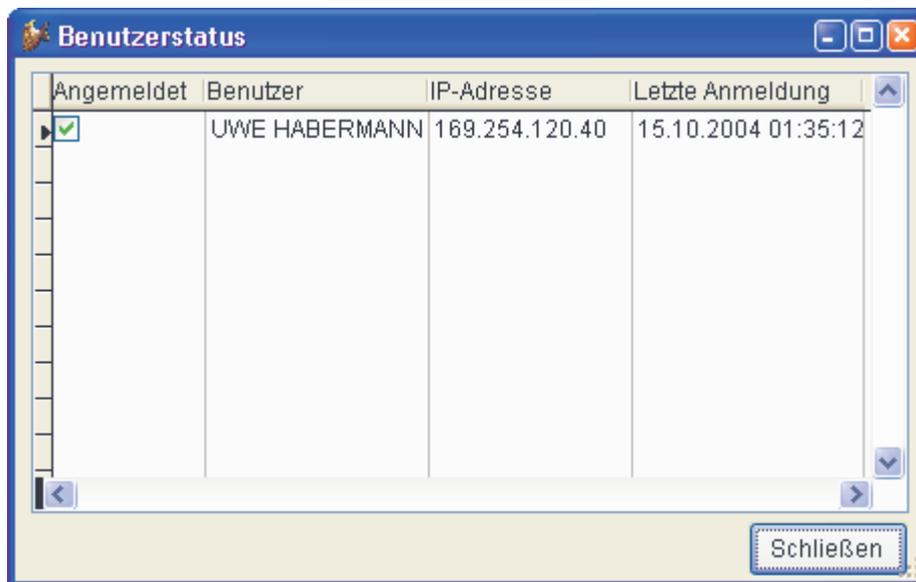
Zurzeit angemeldete Benutzer

VFX verwaltet zurzeit angemeldete Benutzer in einer Tabelle. Mit der Eigenschaft *lAllowMultipleLogin* des Anwendungsobjekts kann eingestellt werden, ob sich Benutzer mehrmals gleichzeitig an der Anwendung anmelden können. Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt wird, können sich Benutzer mehrmals anmelden. Der Standardwert ist *.T.*

```
goProgram.lAllowMultipleLogin=.T.
```

Für jeden Benutzer wird die IP-Adresse des Arbeitsplatzes gespeichert, von dem aus er sich angemeldet hat. Wenn sich ein Benutzer abmeldet, wird die IP-Adresse gelöscht.

Benutzer mit Administratorrechten können über den Menüpunkt *Extras, Benutzerstatus* sehen, welche Benutzer zurzeit angemeldet sind. Es werden die IP-Adresse und die Anmeldezeit angezeigt. Die Spalte Anmeldezeit zeigt in jedem Fall das Datum und die Zeit der letzten Anmeldung, auch wenn der Benutzer zurzeit nicht angemeldet ist.



Angemeldet	Benutzer	IP-Adresse	Letzte Anmeldung
<input checked="" type="checkbox"/>	UWE HABERMANN	169.254.120.40	15.10.2004 01:35:12

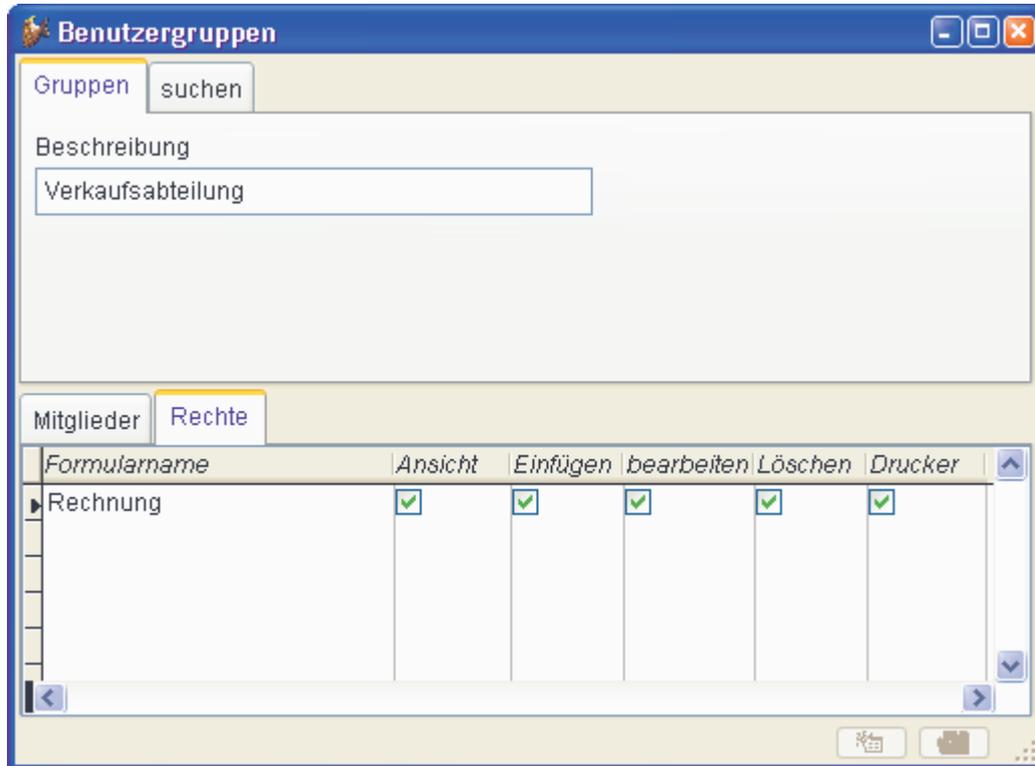
Wenn mit einer VFP-Datenbank gearbeitet wird, ist der Datensatz für den angemeldeten Benutzer ständig gesperrt. Im Falle einer Verbindungsunterbrechung oder eines Programmabbruchs, wird die Satzsperrung automatisch aufgehoben. Der Benutzer kann sich erneut anmelden, ohne dass eine Mehrfachanmeldung festgestellt wird.

Wenn die VFX-Tabellen in einer SQL Server-Datenbank gespeichert sind, wird die System Prozess ID verwendet, um den an den SQL Server angemeldeten Benutzer zu identifizieren. Die aktuelle *SPID* wird in der *Vfxusr*-Tabelle gespeichert. Bei einer versuchten zweiten Anmeldung kann so festgestellt werden, ob der Benutzer bereits angemeldet ist. Wenn eine mehrfache Anmeldung nicht erlaubt ist, wird der Benutzer zurückgewiesen.

Benutzergruppen

Zusätzlich zu den Benutzerstufen, die es auch schon in früheren VFX-Versionen gab, können in VFX 9.0 Benutzergruppen verwaltet werden.

Benutzer mit Administratorrechten (Benutzerstufe 1) können Benutzergruppen anlegen und jeder Gruppe für jedes Formular individuelle Rechte zuweisen. Benutzerrechte können für alle Formulare eingestellt werden, die in der Tabelle *Vfxopen.dbf* eingetragen sind.



Zusätzlich zu den bisherigen Möglichkeiten zur Vergabe von Benutzerrechten, können jetzt Benutzergruppen angelegt werden. Benutzer können Mitglied von einer oder mehreren Benutzergruppen sein. Benutzergruppen können Rechte zugewiesen werden. Wenn ein Benutzer Mitglied von mehreren Benutzergruppen ist, erhält er die Rechte von allen Benutzergruppen.

Wenn einem Benutzer keiner Benutzergruppe zugeordnet ist, gilt die Benutzerstufe wie in früheren VFX-Versionen.

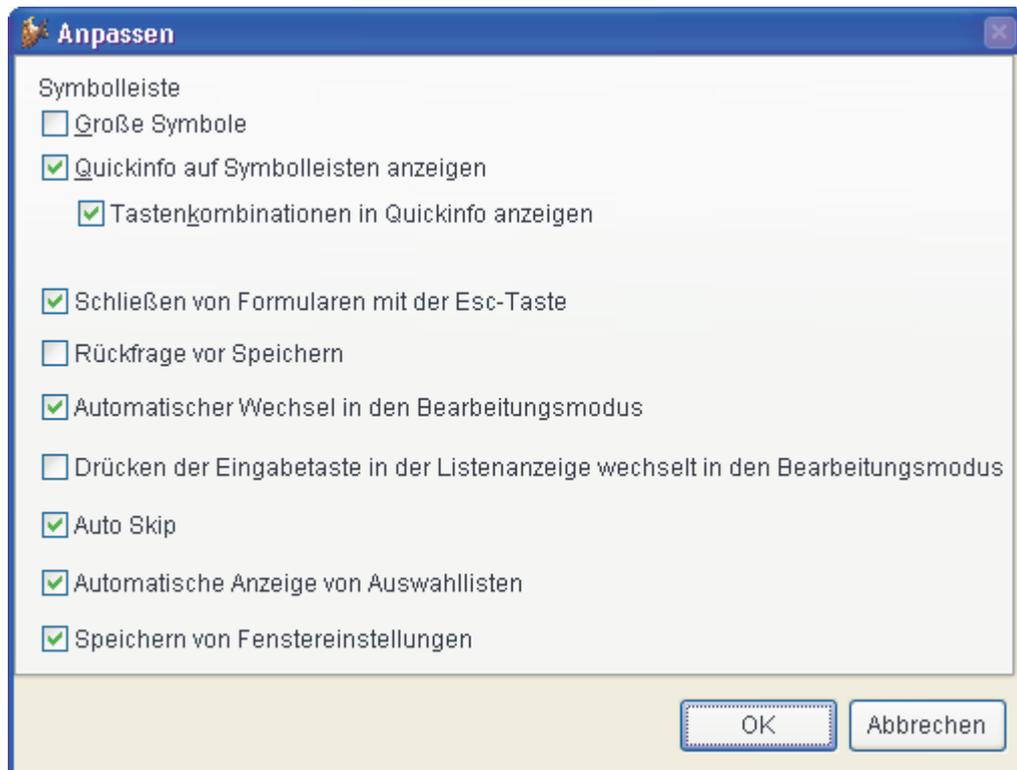
Zur Laufzeit wird ein globales Objekt *goUserRights* instanziiert. Dieses Objekt enthält Child-Objekte für jedes Formular der Anwendung. Die Namen dieser Objekte entsprechen den Namen der Formulare. Jedes dieser Objekte besitzt die Eigenschaften *deletepermit*, *editpermit*, *newpermit*, *printpermit* und *viewpermit*.

Wenn ein Benutzer Mitglied der Gruppe „Sales department“ aus der Abbildung von oben ist, sehen die Eigenschaften des Objekts *goUserRights* zur Laufzeit so aus:

Name	Value	Type
goUserRights	(Object)	O
frminvoices	(Object)	O
deletepermit	.F.	L
editpermit	.T.	L
newpermit	.T.	L
printpermit	.T.	L
viewpermit	.T.	L
frmorders	(Object)	O
deletepermit	.T.	L
editpermit	.T.	L
newpermit	.T.	L
printpermit	.T.	L
viewpermit	.T.	L

Layout

Das Erscheinungsbild von VFX 9.0-Anwendungen wurde durch neue Symbole im Windows-XP-Stil verbessert. Neue Symbole wurden für die Symbolleiste, Menüeinträge und andere Dialoge entwickelt.



Endbenutzer können das Layout der Anwendung über den Menüpunkt *Extras, Anpassen* selbst entsprechend den eigenen Wünschen einstellen. Es kann zwischen kleinen und großen Symbolen in Symbolleisten gewählt werden. Wahlweise können Quickinfos angezeigt werden. Wenn das Kontrollkästchen *Tastenkombinationen in Quickinfo anzeigen* markiert ist, werden an die Quickinfo die Hotkeys angefügt. Beispielsweise ist der Hotkey für die Schaltfläche *Neu* die Tastenkombination *Strg+N*.

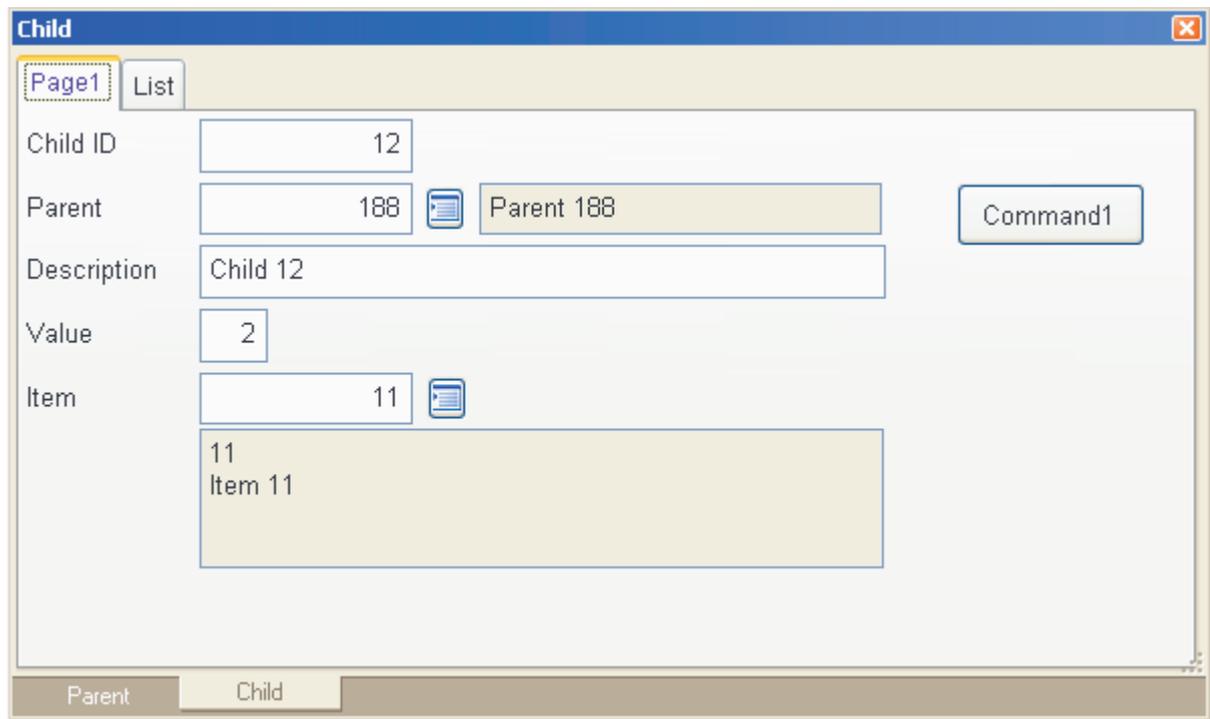


Neu für das Layout von VFX-Formularen ist die Möglichkeit Hintergrundbilder für Seiten auf Seitenrahmen in Formularen auszuwählen. Das Hintergrundbild kann in den VFX Form Buildern eingestellt werden.

Anstelle eines Hintergrundbildes kann mit den VFX – Form Buildern auch eine Hintergrundfarbe für Seiten eines Seitenrahmens eingestellt werden.

Gedockte Formulare

VFX 9.0 unterstützt ineinander gedockte Formulare. Formulare können jedoch nicht an den Rand der Anwendung gedockt werden.



Das Dock-Verhalten von Formularen wird durch die Eigenschaft *goProgram.nDockable* des Anwendungsobjekts gesteuert. Wenn der Wert dieser Eigenschaft auf -1 eingestellt ist, wird die Einstellung des Formulars verwendet. Wenn *goProgram.nDockable* einen Wert größer als 1 enthält, wird dieser Wert in der Eigenschaft *Dockable* des Formulars gespeichert.

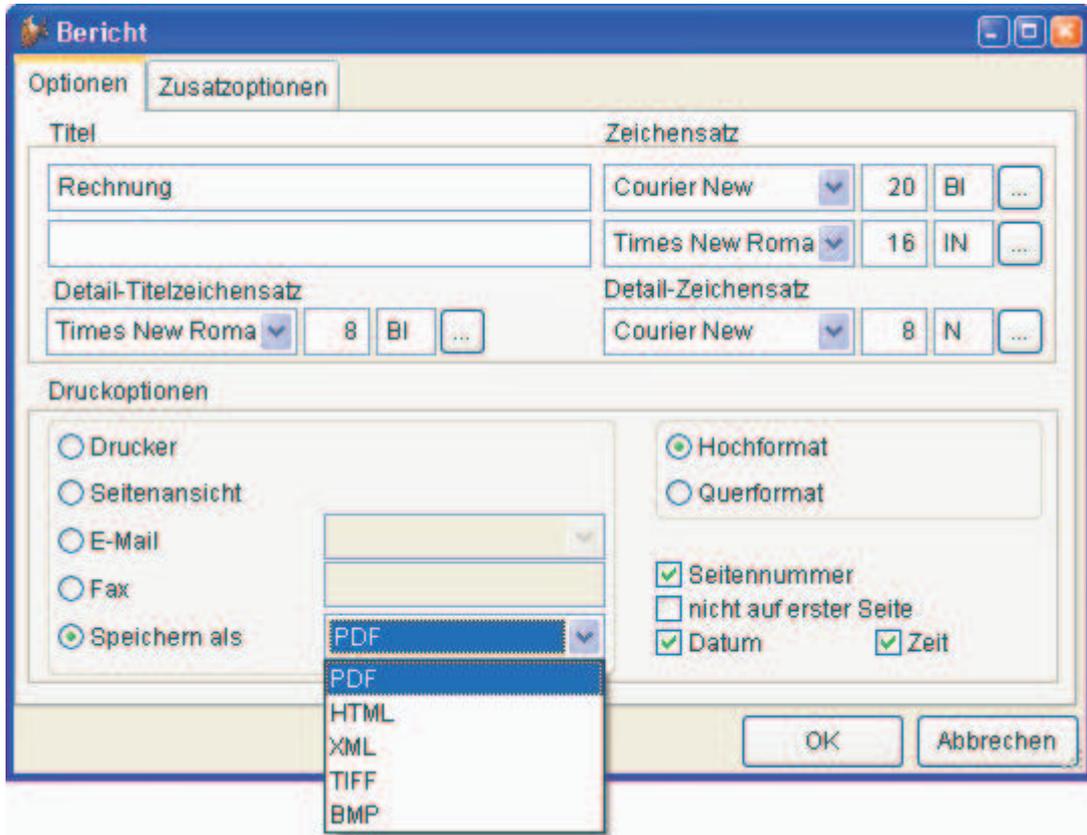
Anmerkung: Wenn die Eigenschaft *WindowType* des Formulars auf *Modal* eingestellt ist, wird die Eigenschaft *goProgram.nDockable* nicht ausgewertet. Modale Formulare können grundsätzlich nicht gedockt werden.

Der Dockstatus und die Dockposition eines Formulars werden für jeden Benutzer in der Ressourcentabelle *Vfxres.dbf* gespeichert.

Berichte

VFX 9.0 unterstützt alle Möglichkeiten von VFP 9 um Berichtsausgaben in verschiedenen Dateiformaten speichern zu können. Die unterstützten Dateiformate sind PDF, HTML, XML, TIFF and BMP. Alle diese Dateiformate können auch als E-Mailanhang versendet werden.

Im Berichtsdialog kann das Dateiformat in einer Combobox ausgewählt werden, wenn eine der Optionen *E-Mail* oder *Speichern als* gewählt wird.

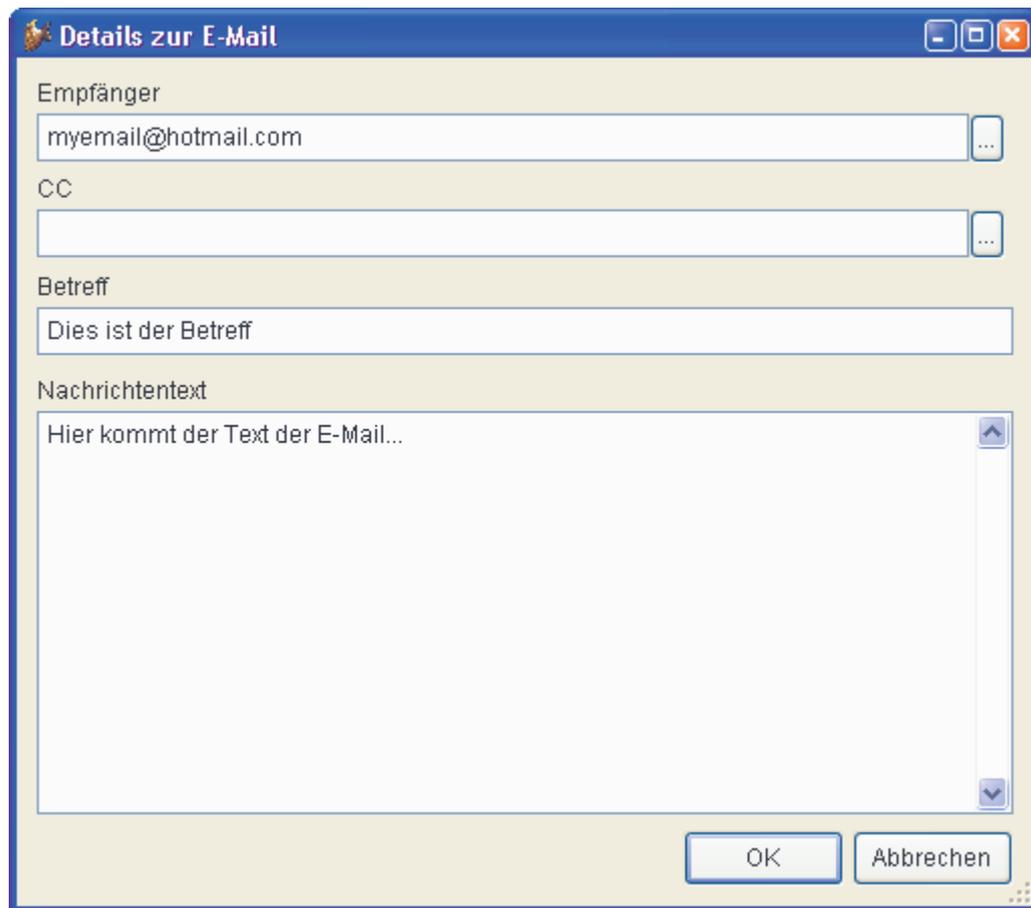


Wenn als Dateiformat TIFF oder BMP gewählt wird, wird für jede Seite des Berichts eine eigene Datei angelegt. Dem vom Anwender eingegebenen Dateinamen wird ein numerischer Wert mit der jeweiligen Seitennummer angehängt.

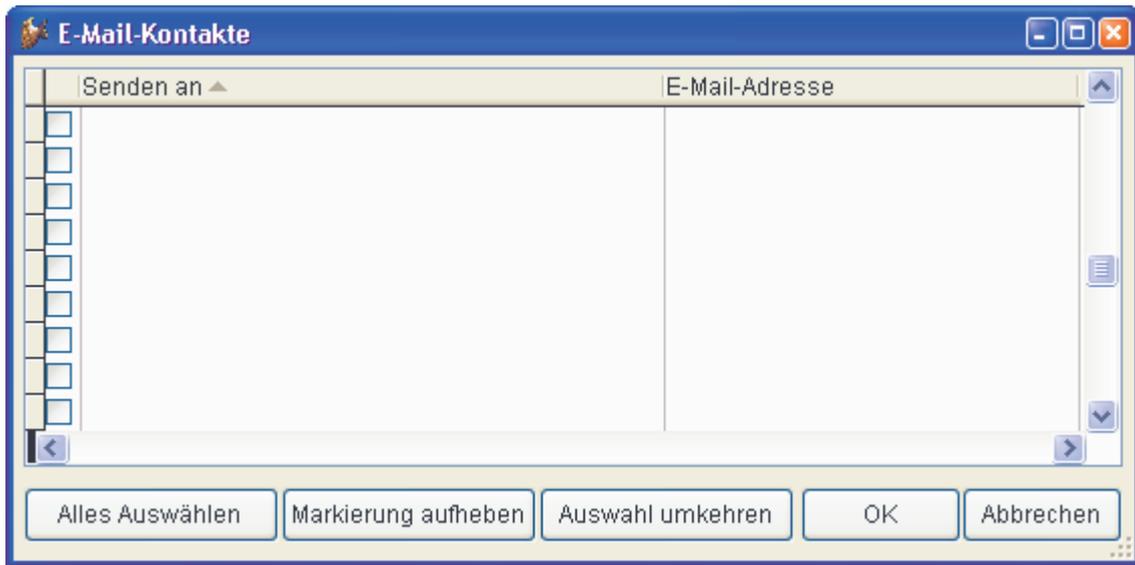
E-Mail Erweiterungen

Alle Dateiformate, in denen Berichtsausgaben gespeichert werden können, können als E-Mailanhang versendet werden.

Im Dialog *Details zur E-Mail* können ein oder mehrere E-Mailempfänger, CC-Empfänger, der Betreff und ein Text eingegeben werden. Wenn der Wert der Eigenschaft *goProgram.UseBCCRecipients* auf *.T.* eingestellt ist, können auch BCC-Empfänger eingegeben werden.



Für jede Art von Empfängerliste kann über eine Schaltfläche eine Auswahlliste mit allen Adressen aus dem Outlook-Adressbuch angezeigt werden.

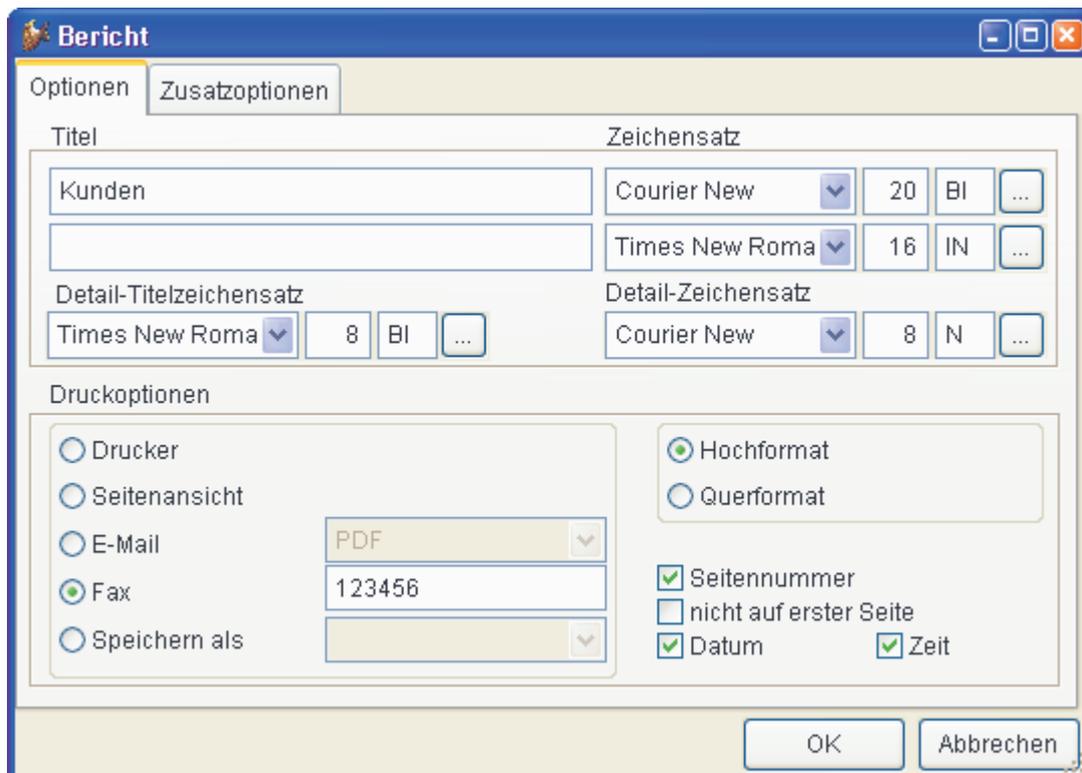


Die ausgewählten E-Mailadressen werden durch einen Klick auf die Schaltfläche *OK* in das Feld mit der Empfängerliste übernommen.

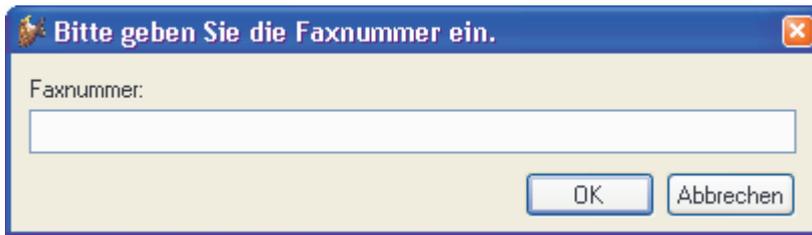
Fax

Eine weitere Möglichkeit Berichtsausgaben zu erzeugen ist der Versand als Fax. Wenn der Anwender die Fax-Option wählt, muss die Faxnummer eingegeben werden.

VFX 9.0 unterstützt die Fax-Programme FRITZ!fax von AVM und Winfax von Symantec. VFX 9.0 erkennt automatisch, ob eins dieser beiden Fax-Programme installiert ist. Wenn ein Fax-Programm erkannt wird, wird die Berichtsausgabe an den entsprechenden Fax-Druckertreiber übergeben.



Die Faxnummer wird von der VFX-Anwendung direkt an das VFX-Programm übergeben. Der Endanwender wird nicht mit Dialogen des Fax-Programms konfrontiert.



Geschäftsgrafiken

Statistische Auswertungen in endlosen Listen sind schwer zu lesen und zu analysieren. Der bessere Weg zur Veranschaulichung von Geschäftsdaten sind grafische, farbige Präsentationen. Die neue Klasse *cBusinessGraph* gibt dem VFX-Entwickler die Möglichkeit Anwendungsdaten mit nur wenigen Minuten Programmierarbeit in Grafiken anzuzeigen und zu drucken.

Zur Anzeige der Grafiken wird das ActiveX-Steuerelement *MSChart* eingesetzt. Die anzuzeigenden Daten können aus einem beliebigen Cursor kommen. Jede Spalte des Cursors entspricht einer Koordinate in der Grafik. Eins der Felder kann Bezeichnungstexte enthalten. Wenn kein Feld mit Bezeichnungstexten angegeben wird, werden alle Felder des Cursors zur Datenanzeige verwendet. Felder zur Datenanzeige müssen einen numerischen Datentyp haben. Zusätzlich können Texte für die Legende der Grafik angegeben werden.

Eigenschaften:

cAliasName – Aliasname des Cursors, der die Daten enthält.

cGraphTitle – Titel der Grafik.

cLabelField – Name des Feldes, das die Bezeichnungstexte enthält.

cLegendTitles – Eine komma-separierte Liste mit der Legende.

lShowLegend – Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt wird, wird neben der Grafik eine Legende angezeigt.

nGraphType – Anzeigetyp der Grafik:

- 0 - 3D Balken (Säule)
- 1 - 2D Balken/Piktogramm
- 2 - 3D Linie (Band)
- 3 - 2D Linie
- 4 - 3D Fläche
- 5 - 2D Fläche
- 6 - 3D Schritt
- 7 - 2D Schritt
- 8 - 3D Kombination
- 14 - 2D Kreis
- 16 - 2D X Y (Punkt)

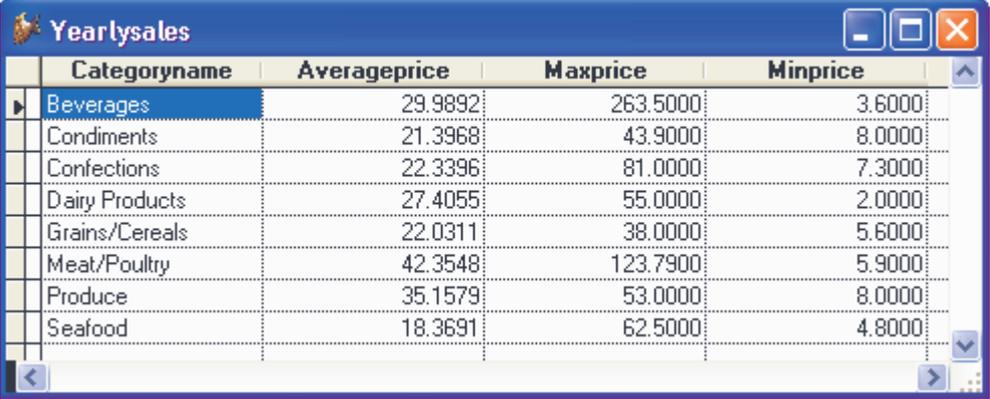
Methoden:

DrawGraph – Erstellen der Grafik anhand der zur Verfügung stehenden Daten und der zuvor eingestellten Eigenschaften. Alle Bezeichnungen und die Legende werden aktualisiert.

OnPrint – Druckt die aktuelle Grafik mit der Berichtsvorlage *Hardcopy.frx*.

Beispiel

Ein Programmteil einer Anwendung erstellt den folgenden Cursor. Daraus soll eine Geschäftsgrafik erstellt werden.



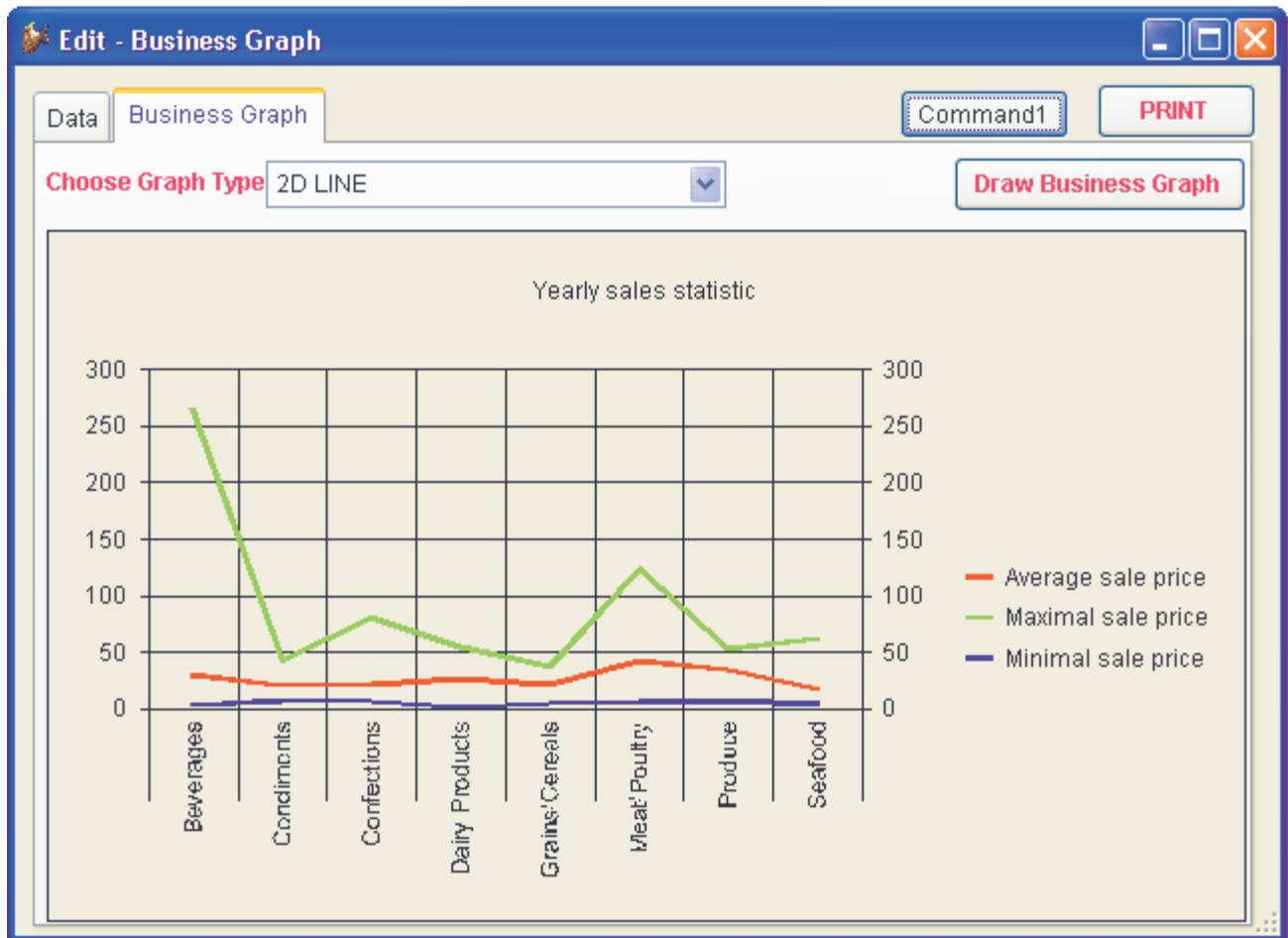
Categoryname	Averageprice	Maxprice	Minprice
Beverages	29.9892	263.5000	3.6000
Condiments	21.3968	43.9000	8.0000
Confections	22.3396	81.0000	7.3000
Dairy Products	27.4055	55.0000	2.0000
Grains/Cereals	22.0311	38.0000	5.6000
Meat/Poultry	42.3548	123.7900	5.9000
Produce	35.1579	53.0000	8.0000
Seafood	18.3691	62.5000	4.8000

Die Klasse *cBusinessGraph* kann auf ein beliebiges Formular gezogen werden. Die folgenden Einstellungen werden bei dem Objekt gemacht:

```
.cAliasName = "YearlySales"  
.cGraphTitle = "Yearly sales statistic"  
.cLabelField = "CategoryName"  
.cLegendTitles = "Average sale price, Maximal sale price, Minimal sale price"
```

Der Eigenschaft *cLabelField* wird der Name der Spalte für die Bezeichnungen zugewiesen. Der Eigenschaft *cLegendTitles* wird eine Aufzählung der Texte für die Legende zugewiesen. Die Reihenfolge der Texte muss der Reihenfolge der Spalten im Cursor entsprechen.

Wenn nun die Methode *DrawGraph* ausgeführt wird, erscheint die folgende Grafik.

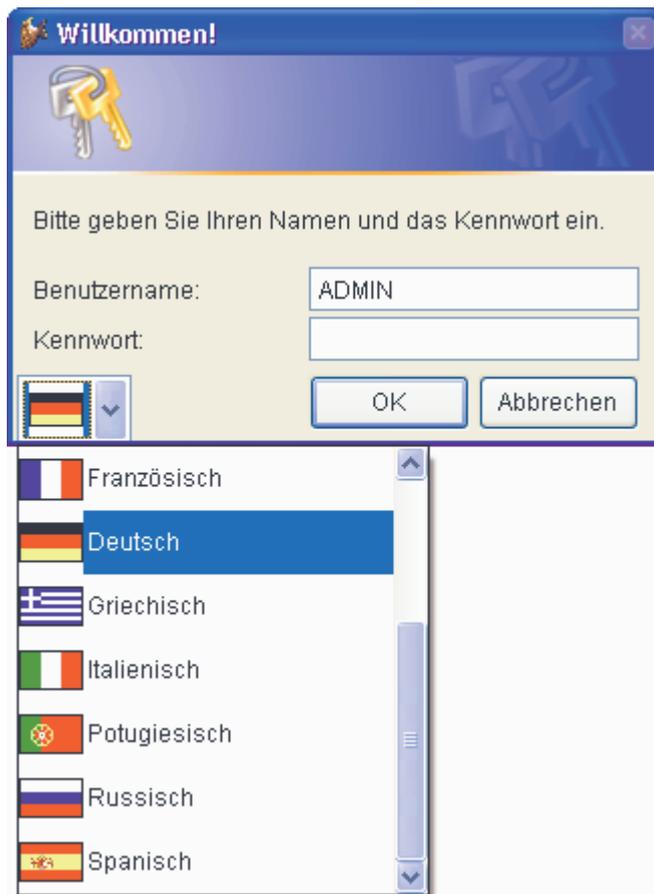


Lokalisierung zur Laufzeit

Mit VFX 9.0 können nicht nur Anwendungen für verschiedene Sprachen lokalisiert erstellt werden, es ist jetzt auch möglich die Sprache einer Anwendung zur Laufzeit umzustellen.

Die Möglichkeit zur Umstellung der Sprache zur Laufzeit wird über die Eigenschaft *goProgram.lRuntimeLocalization* des Anwendungsobjekts gesteuert. Wenn dieser Eigenschaft der Wert *.T.* zugewiesen wird, kann die Sprache der Anwendung im Anmeldedialog ausgewählt werden. Zusätzlich kann, während die Anwendung läuft, die Sprache über eine Combobox in der Standard-Symbolleiste umgeschaltet werden.

Die Eigenschaft `goProgram.lRuntimeLocalization` kann mit dem VFX Application Builder eingestellt werden.



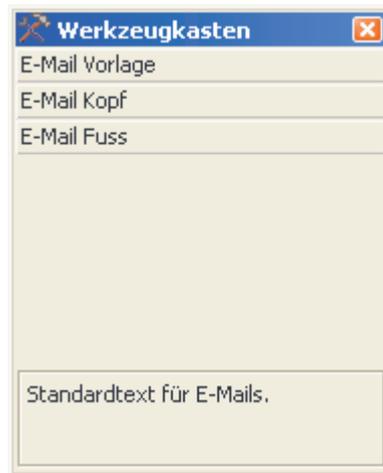
Wenn die Lokalisierung zur Laufzeit aktiviert wird ist, wird ein global sichtbares Objekt mit dem Namen `goLocalize` beim Anwendungsstart instanziiert. Dieses Objekt hat Eigenschaften entsprechend den Texten in der Tabelle `Vfxmsg.dbf`. Für jeden Datensatz in der Tabelle `Vfxmsg.dbf` wird dem Objekt `goLocalize` zur Laufzeit eine Eigenschaft hinzugefügt. Der Name der Eigenschaft entspricht der `Message_ID` mit dem Präfix `c`. Wenn sich beispielsweise in der Tabelle `Vfxmsg.dbf` ein Datensatz mit der `Message_ID` `CAP_APPLICATION_TITLE` befindet, heißt die entsprechende Eigenschaft des Lokalisierungsobjekts `goLocalize.CCAP_APPLICATION_TITLE`. Auf das Lokalisierungsobjekt und seine Eigenschaften kann jederzeit zugegriffen werden.

Die von jedem Benutzer zuletzt verwendete Sprache wird in der Ressourcentabelle `Vfxres.dbf` gespeichert. Wenn sich ein Benutzer erneut anmeldet, erscheint die Anwendung in der zuletzt benutzten Sprache.

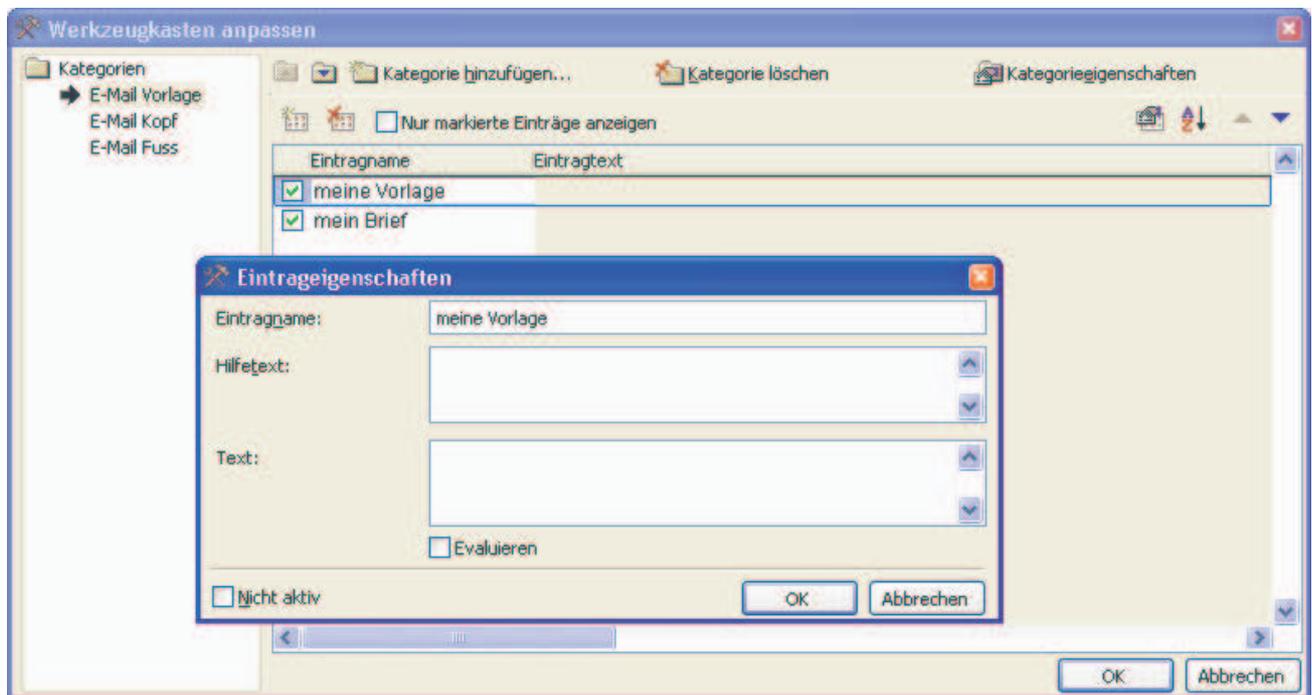
Integration der VFP Toolbox für Endanwender

Die VFP Toolbox ist in VFP 9 auch für Endanwender nutzbar. In VFX 9.0 wurde die Toolbox vollständig integriert und an VFX angepasst. Ähnlich wie die Toolbox für Entwickler, dient der Werkzeugkasten für Endanwender als universelle Drag & Drop Quelle bzw. auch als Ziel. Einträge aus dem Werkzeugkasten können in Textboxen, Editboxen und andere Drop-Ziele gezogen werden.

Die Einträge im Werkzeugkasten sind in Kategorien gruppiert.



Mit einem Rechtsklick auf dem Werkzeugkasten und über den Kontextmenüpunkt *Werkzeugkasten anpassen* können Kategorien und Einträge hinzugefügt, bearbeitet und gelöscht werden.



Für jede Kategorie können der Kategorienname und ein Hilfetext gespeichert werden. Für Einträge können ein Eintragsname, ein Hilfetext und ein Eintragstext gespeichert werden. Kategorienamen und Eintragsnamen werden im Werkzeugkasten angezeigt. Der jeweilige Hilfetext wird am unteren Rand des Werkzeugkastens in einer Editbox als Beschreibung zum aktuellen Eintrag angezeigt.

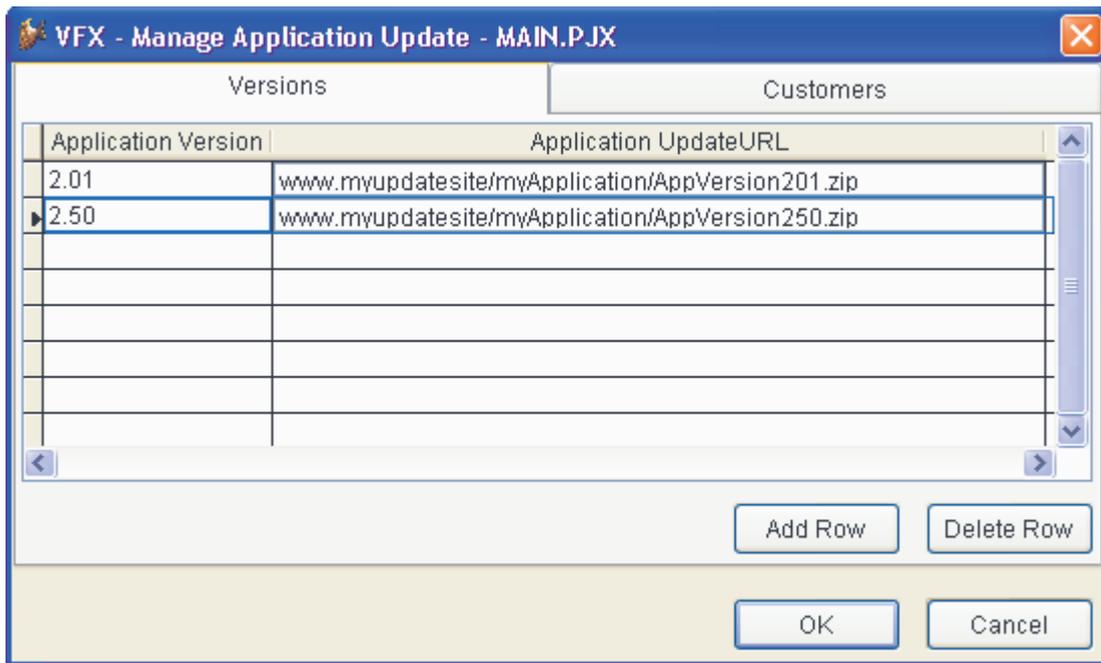
Der Eintragstext wird auf dem jeweiligen Drop-Ziel eingefügt.

Mit den Schaltflächen  und  können Anwender die Reihenfolge der Kategorienanzeige im Werkzeugkasten ändern. Einträge können mit den Schaltflächen  und  innerhalb einer Kategorie verschoben werden

Aktualisierung der Anwendung

Die Möglichkeiten zur Aktualisierung der Anwendung beim Kunden über das Internet wurden erweitert. Der Entwickler kann eine Liste der Kunden anlegen, die berechtigt ist, aktualisierte Programmversionen herunter zu laden und zu installieren. Diese Kundenliste wird in einer verschlüsselten Datei auf dem Web-Server gespeichert und vor der eigentlichen Aktualisierung auf den Kunden-PC heruntergeladen und geprüft. Zusammen mit der Kundenliste wird eine Versionsliste heruntergeladen. Mithilfe dieser Versionsliste können abhängig von der beim Kunden installierten Programmversion unterschiedliche Aktualisierungen durchgeführt werden.

Beide Listen können aus dem VFX 9.0-Menü über den Menüpunkt *Manage Application Update* bearbeitet werden.



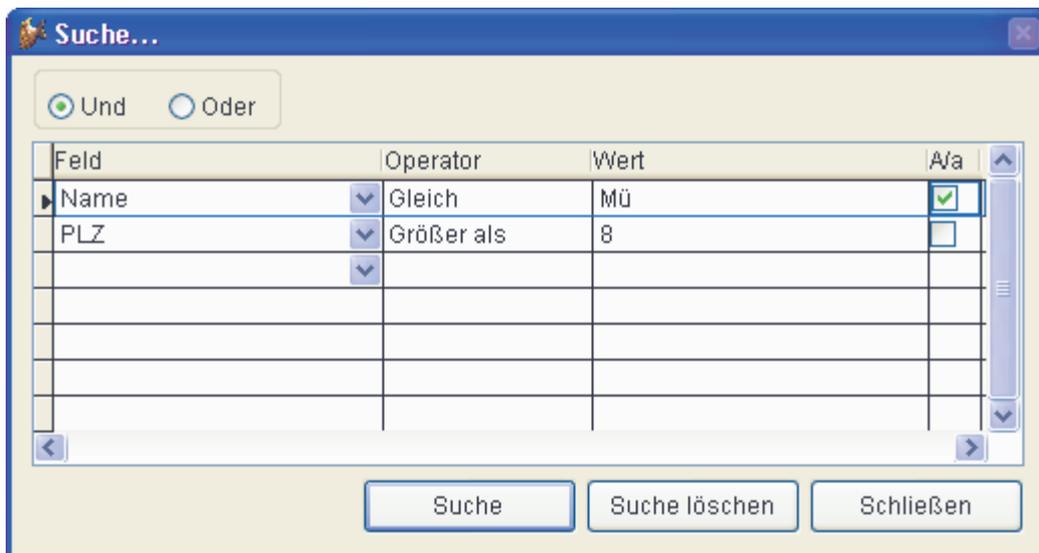
In der Spalte *Application Version* wird die Nummer einer Anwendungsversion eingetragen. In der Spalte *Application Update URL* befindet sich der dazugehörige Download-Link.

Die Durchführung der Aktualisierung geschieht beim Kunden in zwei Schritten. Im ersten Schritt wird ein Download-Skript ausgeführt, das die Kundenliste und die Versionsliste herunterlädt. Der Name der Datei mit der Kundenliste ist standardmäßig *UpdateCustomer.vfx*. Die Versionsliste heißt standardmäßig *UpdateVersion.vfx*. Das Download-Skript für diese beiden Dateien befindet sich in der Tabelle *Vfxsys.dbf* im Feld *UpdateApp*.

Nachdem die beiden Dateien heruntergeladen und entschlüsselt wurden, wird im zweiten Schritt geprüft, ob der Benutzer zur Aktualisierung berechtigt ist. Der Download-Link der für seine Anwendung geeigneten aktualisierten Version befindet sich in der Datei *UpdateVersion.vfx*.

Verbesserte Suchfunktionen

Im Suchdialog können Anwender nur noch gültige Ausdrücke eingeben. Je nach gewähltem Datentyp stehen nur die geeigneten Vergleichsoperatoren zur Verfügung. Es können nur Werte vom gleichen Datentyp eingegeben werden.



In der Spalte *Value* befinden sich mehrere Steuerelemente. Die Eigenschaft *CurrentControl* dieser Spalte wird abhängig vom Datentyp des in der Spalte *Feld* gewählten Feldes umgeschaltet.

Wenn ein Feld vom Typ *Zeichen* gewählt wird, wird in der Spalte *Wert* eine Textbox angezeigt, in die beliebige Werte eingegeben werden können. Es steht zusätzlich der Vergleichsoperator *Enthält* zur Verfügung. In diesem Fall wird der Filterausdruck mit dem Operator *\$* aufgebaut. Zusätzlich kann in der rechten Spalte im Grid für jede Zeile eingestellt werden, ob die Groß-/Kleinschreibung berücksichtigt werden soll.

Wenn ein numerisches Feld in der ersten Spalte gewählt wird, wird in der Spalte *Wert* eine Textbox angezeigt, die es dem Benutzer erlaubt nur Zahlenwerte einzugeben.

Wenn ein Feld vom Typ *Date* oder *Datetime* gewählt wird, wird die *Inputmask* in der Spalte *Wert* entsprechend eingestellt.

Wenn ein logisches Feld ausgewählt wird, kann in der Spalte *Wert* in einer Combobox *Wahr* oder *Falsch* ausgewählt werden. Die manuelle Eingabe eines Wertes durch den Anwender ist nicht erforderlich.

Auf diesem Weg ist es dem Benutzer nicht möglich unzulässige Werte in der Spalte *Wert* einzugeben.

Treeview

Die Klasse *cTreeView* wurde so verbessert, dass eine wesentliche verkürzte Ladezeit erreicht werden konnte. Der aktuelle Zustand aller Knoten, geöffnet oder geschlossen, wird in der Ressourcentabelle *Vfxres.dbf* für jeden Benutzer gespeichert. Beim erneuten Öffnen eines Formulars erscheinen alle Knoten in dem Zustand, in dem das Formular geschlossen wurde.

Es ist jetzt möglich aus Formularen basierend auf einer der Formulklassen *cTreeviewForm* oder *cTreeviewOneToMany* Berichte zu drucken, die die Struktur des Treeview beinhalten.

Dem Treeview-Steuerelement wurde ein Kontextmenü mit den Einträgen *Neu*, *Umbenennen* und *Löschen* hinzugefügt.

Neue Eigenschaften:

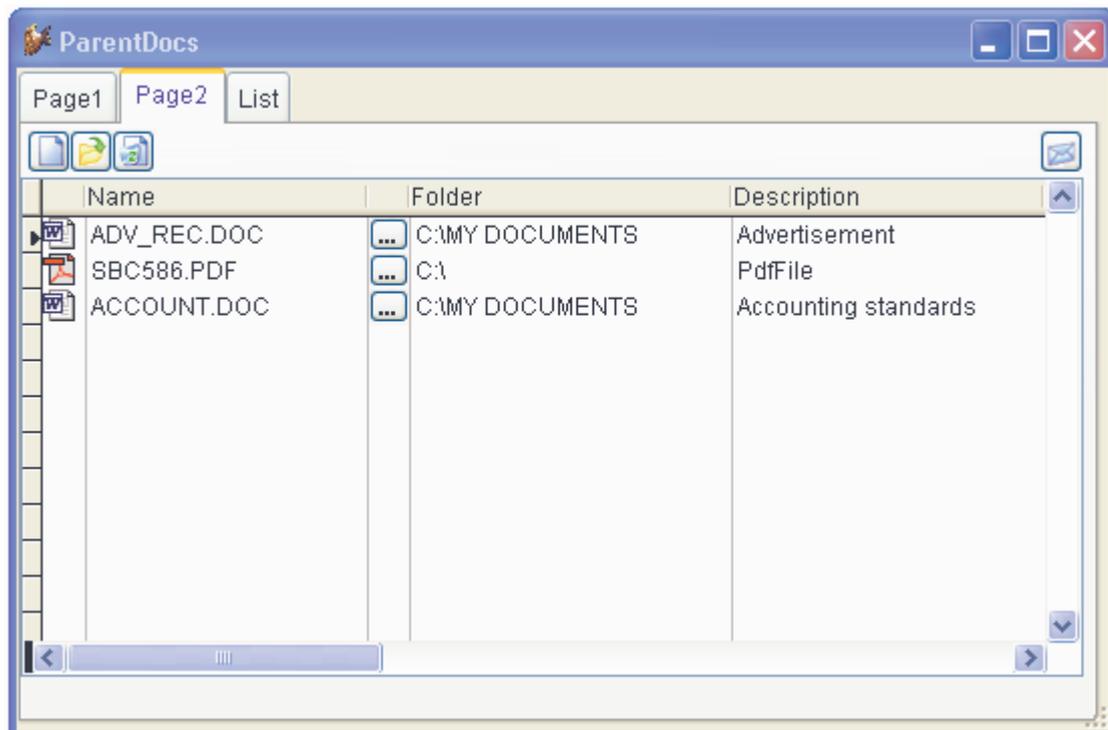
lLoadAllTreeViewNodes – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, werden alle Knoten des Treeview beim Laden des Formulars geladen. Wenn der Wert dieser Eigenschaft auf .F. eingestellt ist, werden beim Laden des Formulars nur die sichtbaren Knoten geladen. In diesem Fall werden beim Öffnen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

lRestoreTreeViewStatus – 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.

Klasse *cDocumentManagement* zur Dokumentverwaltung

Die neue 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.



Eigenschaften:

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.

Info-Dialog

Dem Info-Dialog wurde ein Link-Label zur Anzeige des Endbenutzer-Lizenzvertrags (EULA) hinzugefügt. Über dieses Link-Label wird ein Dialog angezeigt, indem der Benutzer den Lizenzvertrag lesen und drucken kann.

Der Endbenutzer-Lizenzvertrag ist in der Tabelle *Vfxmsg.dbf* im Datensatz *Message_ID=“MSG_EULA“* gespeichert. So ist es einfach möglich für jede Sprache einen lokalisierten Endbenutzer-Lizenzvertrag zur Verfügung zu stellen.

Weitere Verbesserungen für Endbenutzer

- Unterstützung der inkrementellen Suche auch wenn der aktuelle Zelleninhalt *.NULL.* ist.
- Lokalisierte Hotkeys für die Klasse *cpickdate* und ein mehrzeiliger Tooltip als Hilfe.
- Neue Klassen: E-Mail mit Outlook-Aufruf, Hyperlink mit Internet Explorer-Aufruf, numerische Textbox mit Taschenrechneraufruf, TAPI, Dateiauswahl mit Fileselectbox.
- Unterstützung von *visible=.F.* in Grid-Columns für den Suchdialog und den Druckdialog.
- Restzeitanzeige bei der Aktualisierung der Kundendatenbank.
- Skript für Download und Installation von Adobe Reader (für PDF-Dokumente).
- Tastaturbedienung des XP-Öffnen-Dialogs.
- Unterstützung von Drag & Drop in Mover-Dialogen.
- Beim erneuten Öffnen eines Formulars wird der Satzzeiger auf den zuletzt angezeigten Datensatz positioniert.
- Unterstützung der Eigenschaft *HighLightStyle* in Grids.
- Verbesserte Anzeige von Memo-Feldern in Grids.
- Wenn alle Favoriten gelöscht werden, wird das dazugehörige, leere Menü gelöscht.