



Session V-FX95

Was ist neu in VFX 9.5?

Uwe Habermann, Venelina Jordanova

Einführung

Wieder ist zur Konferenz eine neue Version des RAD Frameworks Visual Extend fertig. Und wieder gibt es über endlos viele neue Features zu berichten. VFX 9.5 bringt Lösungen für sehr viele kleine Wünsche von Programmierern und Endkunden. In dieser Session wird gezeigt wie man ohne Programmierung, durch ein einfaches Update auf VFX 9.5, seinen Kunden viele neue Features zur Verfügung stellen kann. Aber auch die neuen Produktivitäts-Tools bei der Entwicklung kommen nicht zu kurz.

Neue Eigenschaften für Entwickler

Vererbungsarchitektur

Vfxobjbase.vcx

Die Vererbungsarchitektur der VFX-Klassen wurde erweitert. In bisherigen VFX-Versionen konnte nur mithilfe von Hooks in die Funktion und in das Layout von VFX-Basisklassen eingegriffen werden. Wenn ein Entwickler zum Beispiel in seiner gesamten Anwendung eine bestimmte Schriftart verwenden wollte, konnte dies nur über Hooks im *Init* Ereignis erreicht werden. Dies hatte den Nachteil, dass die Anwendung in den VFP-Designern stets in der VFX-Standardschriftart Arial angezeigt wurde und nur zur Laufzeit die über einen Hook eingestellte Schriftart angezeigt wurde.

In VFX 9.5 ist nun eine zusätzliche Vererbungsschicht vorhanden. Die in bisherigen VFX-Versionen vorhandenen VFX-Basisklassen aus der Klassenbibliothek *Vfxobj.vcx* befinden sich nun in der Klassenbibliothek *Vfxobjbase.vcx*. An die bisherigen Namen der Klassen wurde *base* als Suffix angehängt.

Zu jeder Klasse aus dieser Klassenbibliothek gibt es eine 1:1-Ableitung in der Klassenbibliothek *Vfxobj.vcx*. Die Klassenbibliothek *Vfxobj.vcx* steht nunmehr dem Entwickler für eigene Anpassungen und Erweiterungen zur Verfügung. Hier ist es beispielsweise möglich die Schriftart einer Klasse zu ändern. Diese Einstellung wirkt sich dann auf alle Steuerelemente basierend auf dieser Klasse in der gesamten Anwendung aus.

Bei einer Aktualisierung des Projekts mit dem VFX – Update Project Wizard wird die Klassenbibliothek *Vfxobj.vcx* nicht aktualisiert.

VFX-Formularklassen

Die VFX-Formulare verwenden jetzt keine Datenumgebung mehr. Das Design der VFX-Formularklassen wurde so erneuert, dass alle benötigten Datenquellen programmatisch geöffnet werden. Alle VFX-Formulare sind 1:1-Ableitungen aus den entsprechenden Formularklassen. Künftige Änderungen in VFX erfolgen also nur noch in den Formularklassen. Eine Änderung der VFX-Formulare ist nicht mehr erforderlich. Dadurch hat der Entwickler die Möglichkeit die VFX-Formulare nach eigenem Bedarf anzupassen, ohne dass bei einer Aktualisierung von VFX Änderungen verloren gehen bzw. noch mal gemacht werden müssen.

Datenzugriff

Einstellungen für die Datenumgebung

Viele SET-Einstellungen werden schon vor dem Öffnen von Cursors in der Datenumgebung benötigt. Zusätzlich zu der Methode *SetDataEnvironment()* des Objekts *goEnvironment* können jetzt SET-Einstellungen für CursorAdapter-Objekte gemacht werden. Standardmäßig gelten jetzt für alle CursorAdapter-Objekte die gleichen SET-Einstellungen, die auch für Formulare gelten.

Die neue Methode *OnSetEnv* aus der Cursoradapter-Klasse *cBaseDataAccess* überprüft, ob das Formular, das den CursorAdapter instanziiert, eine Methode *OnSetEnv* besitzt. Wenn diese Methode existiert und wenn diese Methode nicht bereits von einem anderen CursorAdapter aufgerufen wurde, wird sie ausgeführt. Wenn das Formular keine *OnSetEnv*-Methode hat, wird überprüft, ob das Objekt *goEnvironment* existiert. Wenn dieses Objekt existiert, wird dessen Methode *SetDataEnvironment* aufgerufen. In allen anderen Fällen werden fest codierte SET-Einstellungen ausgeführt.

In der Klasse *cAppDataAccess* in der Klassenbibliothek *Appl.vcx* können eigene SET-Einstellungen hinzugefügt werden.

Das neue Objekt goPath

Zur Laufzeit einer Anwendung wird das neue Objekt *goPath* instanziiert. Dieses Objekt hat Eigenschaften, deren Werte auf die aktuell verwendeten Pfade zeigen:

CDataDir – Pfad zur aktuell verwendeten Datenbank.

ClientName – Name der aktuellen Datenbank. Dies ist der für den Benutzer im Mandantenauswahldialog sichtbare Name. Dies ist nicht unbedingt der physikalische Name der Datenbank.

VfxPath – Pfad zu den VFX Systemtabellen.

ReportPath – Pfad zu den Berichtsdateien.

UpdatePath – Pfad zu dem Ordner mit einer aktualisierten Datenbankstruktur.

ImportPath – Diese Eigenschaft wird von VFX nicht verwendet und steht zur freien Verfügung.

ExportPath – Diese Eigenschaft wird von VFX nicht verwendet und steht zur freien Verfügung.

Wenn die Tabelle *Vfxpath.dbf* zur Mandantenauswahl verwendet wird, wird jedes Feld dem Objekt *goPath* als Eigenschaft hinzugefügt.

Wenn die Datei *Config.vfx* zur Mandantenauswahl verwendet wird, wird jedes Feld dem Objekt *goPath* als Eigenschaft hinzugefügt.

So kann zur Laufzeit auf einfachem Weg auf die aktuellen Pfadeinstellungen zugegriffen werden.

Erweiterungen in der Klasse **cBaseDataAccess**

Neue Eigenschaften

cForeignKeyName – Diese Eigenschaft wird nur bei Child-CursorAdapttern auf OneToMany-Formularen verwendet. Hier wird der Name des Feldes im Parent-Arbeitsbereich gespeichert, das den Primärschlüssel enthält.

cForeignKeyValue – Diese Eigenschaft wird nur bei Child-CursorAdapttern auf OneToMany-Formularen verwendet. Mit diesem Ausdruck wird der Wert des Fremdschlüssels des neu gespeicherten Parent-Datensatzes ermittelt. Normalerweise ist dies das Feld mit dem Primärschlüssel aus dem Parent-Arbeitsbereich. Das Ergebnis der Evaluierung dieses Ausdrucks wird in dem Feld gespeichert, das in der Eigenschaft *cForeignKeyName* angegeben ist.

cWhereClause – Hier kann eine Where-Klausel angegeben werden. Zur Laufzeit wird dieser Wert an den Wert aus der Eigenschaft *SelectCmd* angefügt. Anschließend wird die *CursorFill*-Methode aufgerufen. Da dieser Wert auf Formularebene eingegeben werden kann, bleibt die CursorAdapter-Klasse unverändert. Änderungen an der CursorAdapter-Klasse, insbesondere wegen Strukturänderungen der Basistabellen, können unabhängig von der Where-Klausel durchgeführt werden.

Neue Methode

OnSetEnv- Diese Methode wird während der Initialisierung aufgerufen. Die neue Methode *OnSetEnv* überprüft, ob das Formular, das den CursorAdapter instanziiert, eine Methode *OnSetEnv* besitzt. Wenn diese Methode existiert und wenn diese Methode nicht bereits von einem anderen CursorAdapter aufgerufen wurde, wird sie ausgeführt. Wenn das Formular keine *OnSetEnv*-Methode hat, wird überprüft, ob das Objekt *goEnvironment* existiert. Wenn dieses Objekt existiert, wird dessen Methode *SetDataEnvironment* aufgerufen. In allen anderen Fällen werden fest codierte SET-Einstellungen ausgeführt.

Builder und Wizards

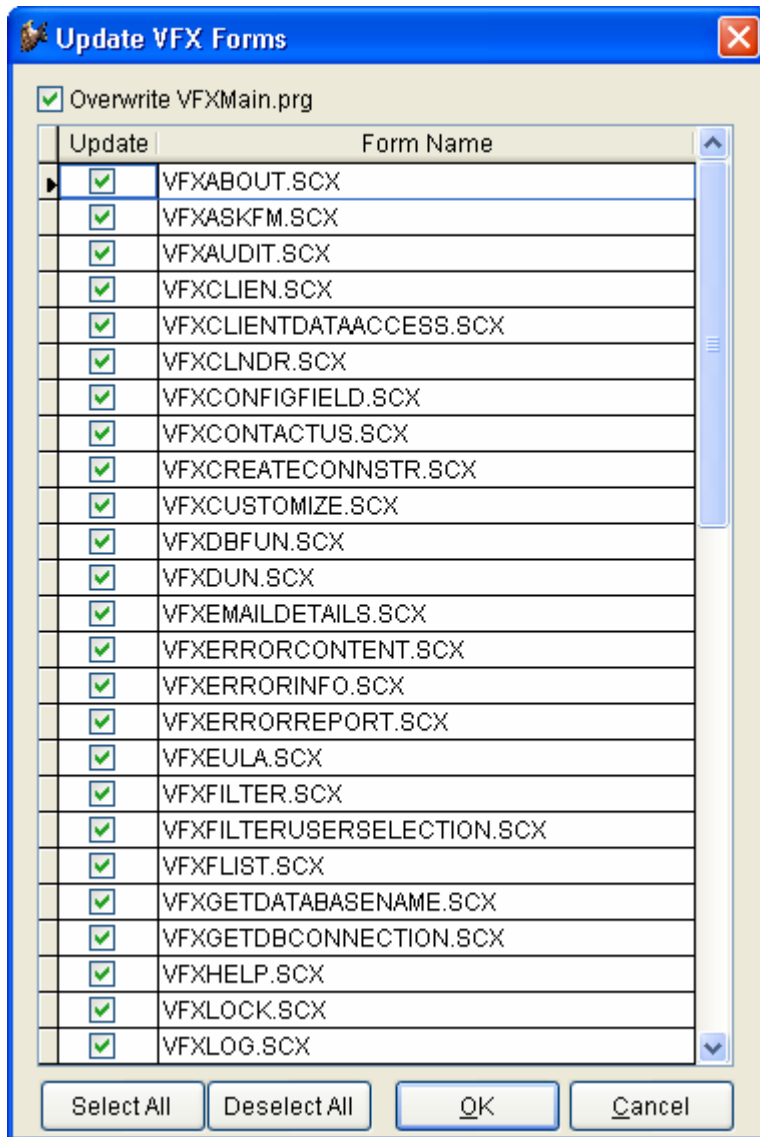
VFX – Task Pane

In der VFX – Task Pane befindet sich in der Symbolleiste eine neue Schaltfläche. Hierüber kann der VFX – Update Project Wizard gestartet werden.

VFX – Update Project Wizard

Vor der Aktualisierung wird eine Archivdatei vom Projektordner mit allen Unterordnern angelegt. Der Dateiname besteht aus dem Projektnamen, dem aktuellen Datum im ANSI-Format sowie der VFX-Version, die das Projekt vor der Aktualisierung hat.

In einem anschließend erscheinenden Dialog kann eingestellt werden, welche der VFX-Formulare bei der Aktualisierung überschrieben werden sollen. Entwickler, die in VFX-Formularen Änderungen gemacht haben, dürfen diese Formulare nicht überschreiben. Die Einstellungen in diesem Dialog werden für spätere Aktualisierungen gespeichert und bleiben so erhalten.



VFX – Upsizing Wizard

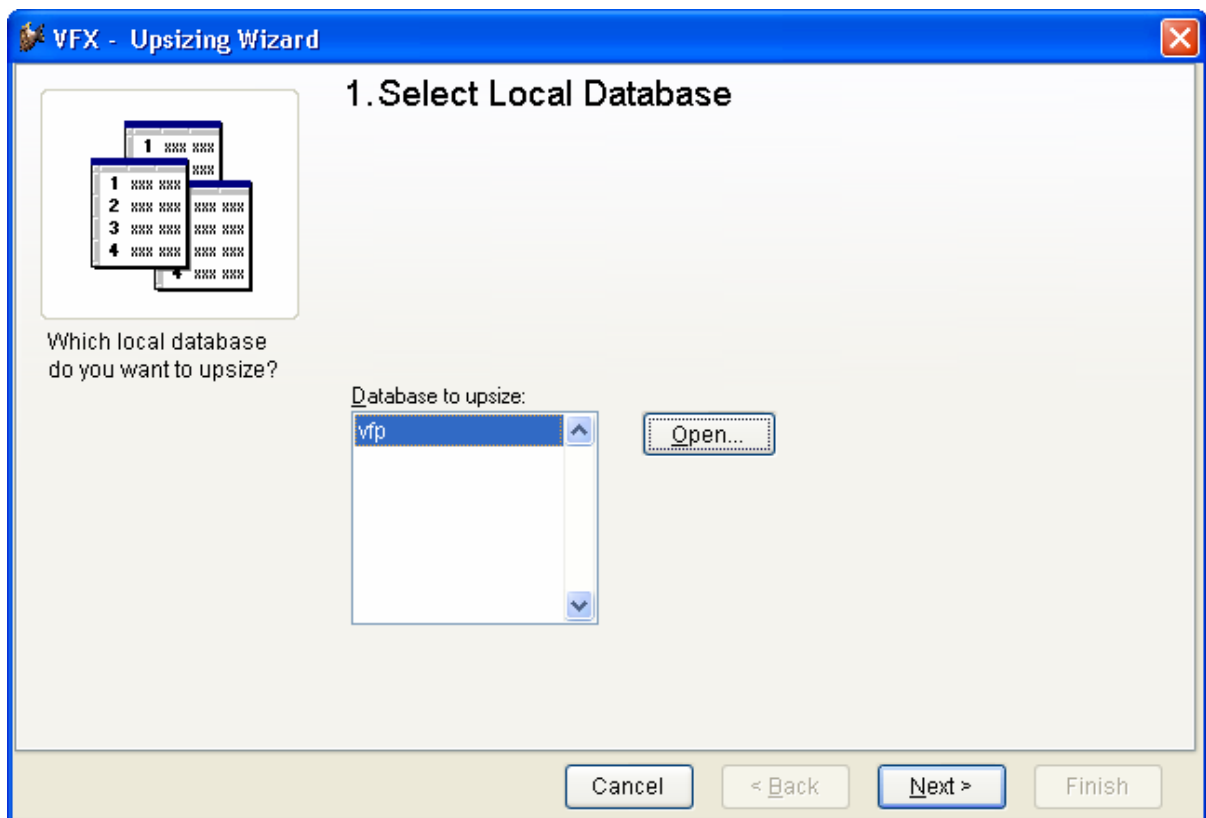
Der VFX – Upsizing Wizard entspricht in seinem Funktionsumfang etwa dem VFP – Upsizing Wizard. Mit dem VFX – Upsizing Wizard können jedoch auch ID-Werte korrekt in eine Remote Datenbank portiert werden. Die Portierung von Daten ist ohne Codepage-Konflikte möglich. Auch Feldnamen, die im SQL Server geschützte Befehlsbegriffe sind, können portiert werden. Standardmäßig sind in allen Feldern Nullwerte erlaubt. Im Gegensatz zum VFP – Upsizing Wizard wird standardmäßig eine neue Datenbank angelegt und nicht vorgeschlagen eine vorhandene Datenbank zu überschreiben. Und schließlich erfordert der VFX – Upsizing Wizard keine DSN-Verbindung, sondern erlaubt beliebige Verbindungswege zur Remote Datenbank.

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 Use Trusted Connection

User Name

Password

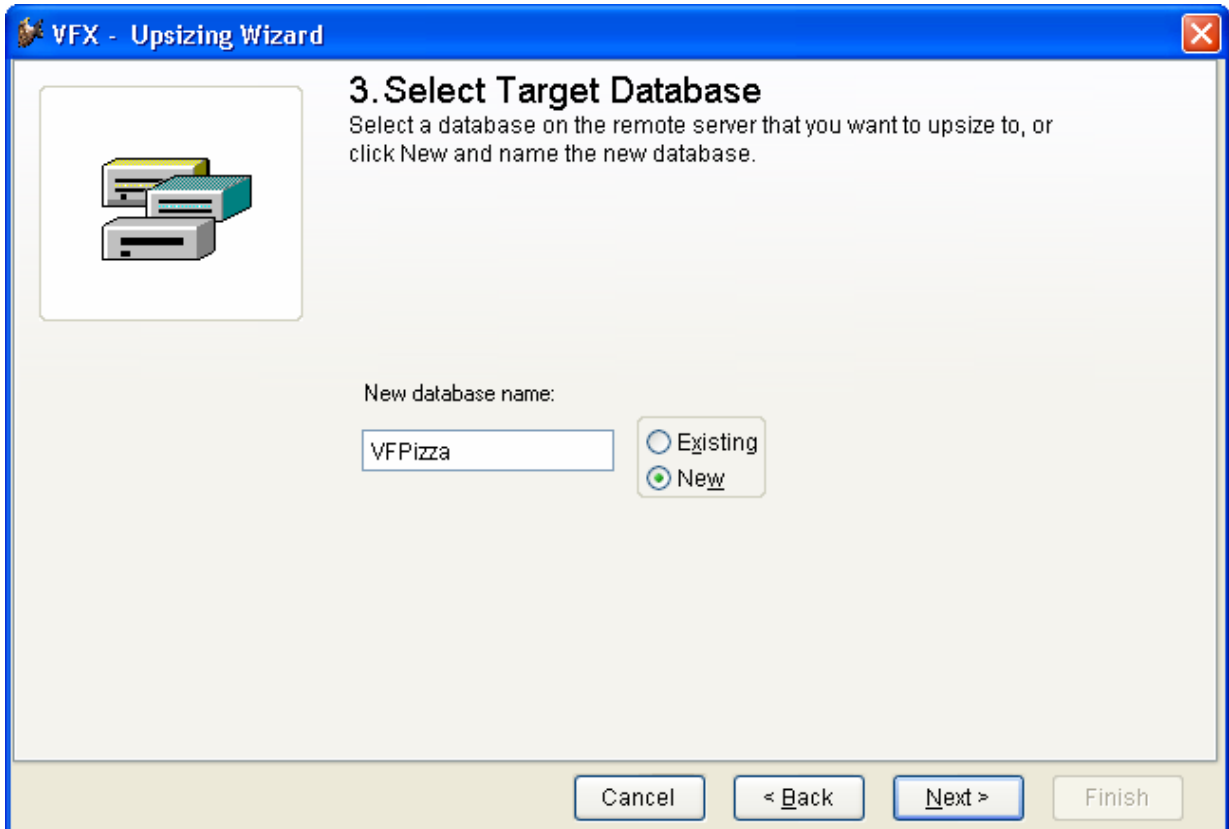
Use connection string

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 vertrauenswürdig Verbindung mit den Windows-Anmeldedaten herzustellen. In der Regel sind auf dieser Seite des Assistenten also keine Eingaben erforderlich.

3. Eingabe des Datenbanknamens

Hier wird der Name der neu zu erstellenden Datenbank eingegeben. Auf Wunsch kann auch eine vorhandene Datenbank überschrieben werden. Wenn die Option *New* ausgewählt ist, kann der Name der neuen Datenbank eingegeben werden. Der Name kann maximal 30 Zeichen lang sein und aus Buchstaben, Ziffern sowie den Sonderzeichen #, \$ und _ bestehen. Leerzeichen sind nicht erlaubt.



VFX - Upsizing Wizard

3. Select Target Database

Select a database on the remote server that you want to upsize to, or click *New* and name the new database.

New database name:

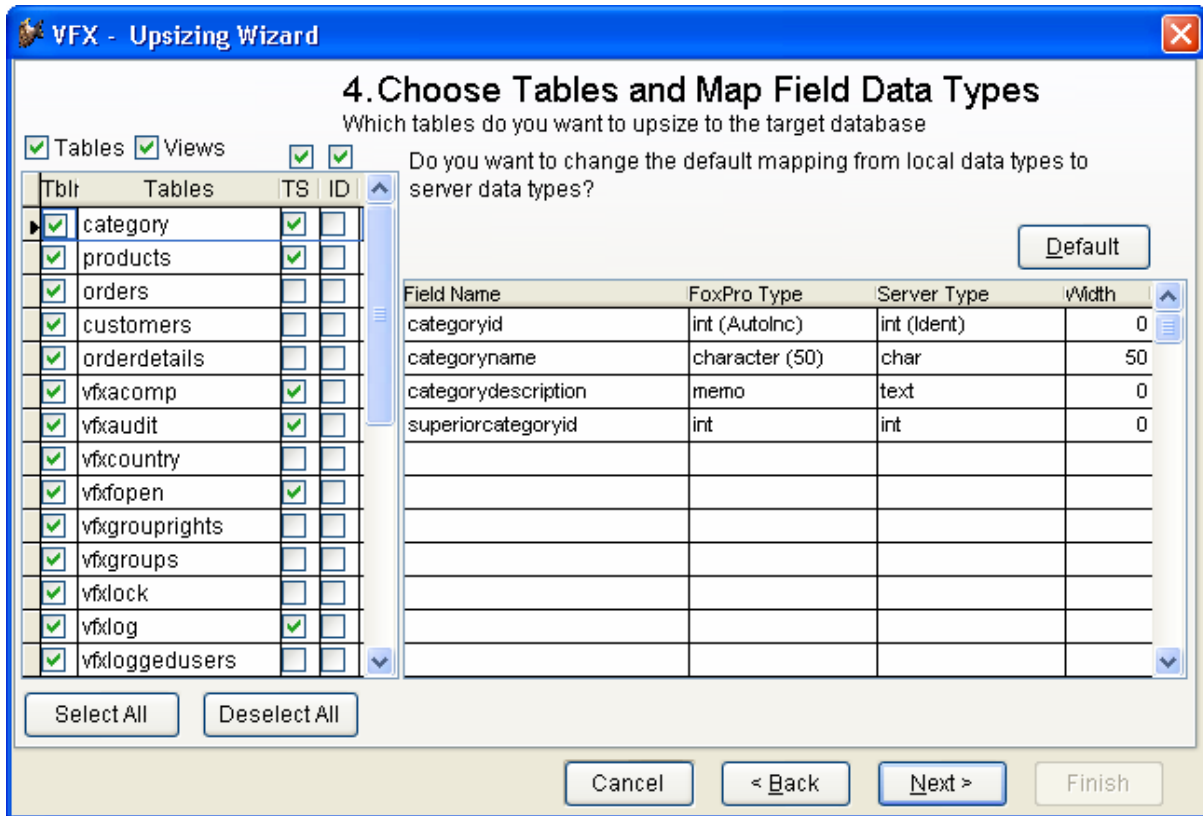
Existing
 New

Cancel < Back Next > Finish

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 überschrieben werden soll.

4. Auswahl der Tabellen und Zuordnung der Datentypen

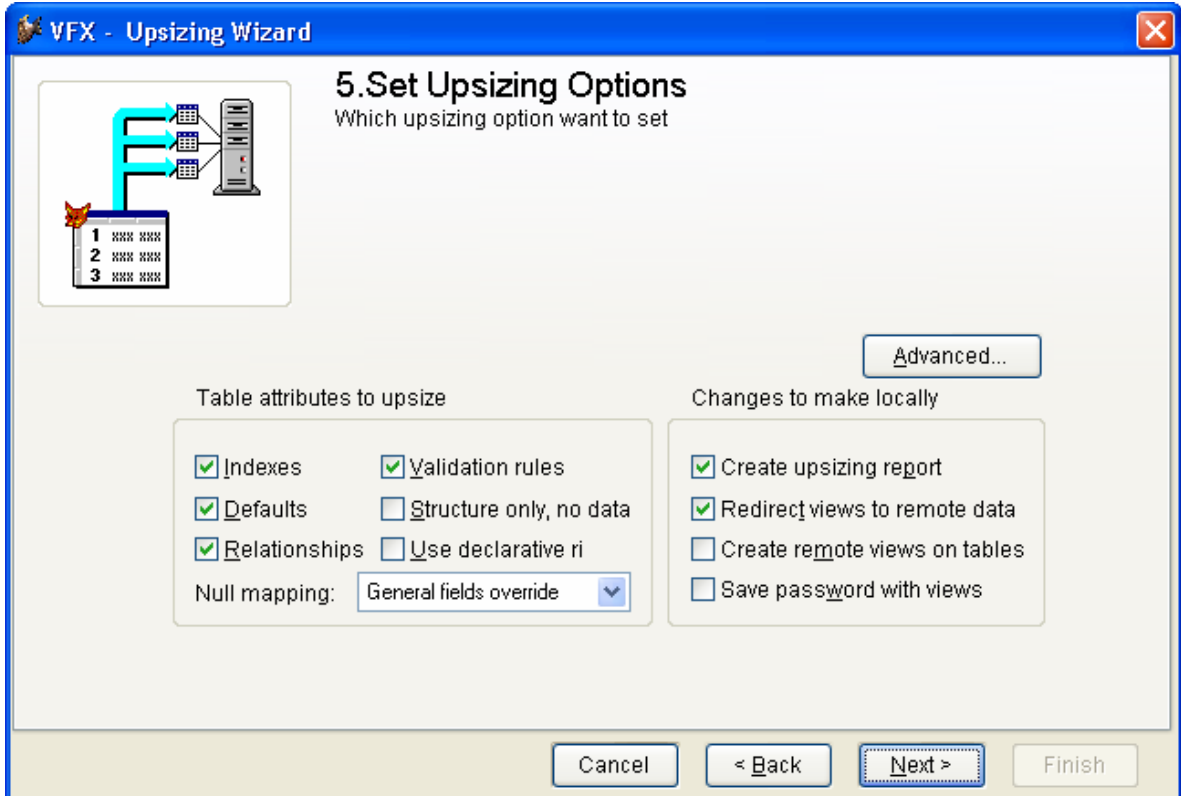
Hier kann ausgewählt werden welche Tabellen aus der VFP-Datenbank auf den SQL Server portiert werden sollen. Standardmäßig sind alle Tabellen zur Portierung markiert. Jeder Tabelle kann ein Timestamp-Feld sowie ein ID-Feld hinzugefügt werden. Der VFX – Upsizing Wizard fügt Tabellen mit Memo-Feldern standardmäßig 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

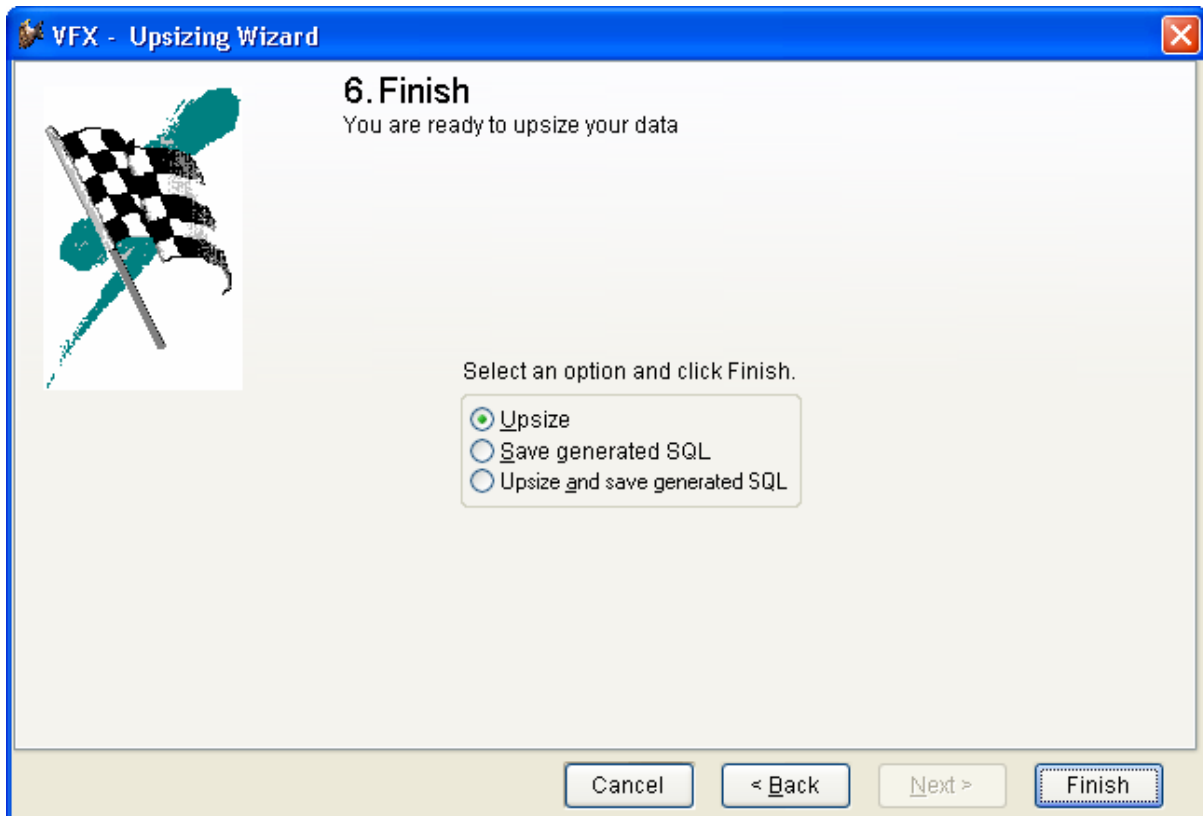
Standardmäßig werden die Strukturen von Tabellen sowie die Daten portiert. Es können auch Indexschlüssel, Standardwerte, Beziehungen (RI Constraints) und Validierungsregeln portiert werden. In der Combobox *Null mapping* kann eingestellt werden, ob Nullwerte erlaubt sind. Diese Option hilft sicherzustellen, 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 Ansichten aus der VFP-Datenbank umbenannt, um Tabellen und Ansichten mit den gleichen Namen in der SQL Server-Datenbank erstellen zu können.

VFX – CursorAdapter Wizard

CursorAdapter-Klassen können jetzt auch basierend auf Ansichten, gespeichert in einem DBC oder einer SQL Server Datenbank, erstellt werden.

Der Wizard kann jetzt auch auf einzelne Tabellen oder Ansichten angewendet werden.

Für jedes Feld aus jeder Tabelle kann eine Typkonvertierung durchgeführt werden. Es stehen alle VFP-Feldtypen zur Verfügung.

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

Der VFX – CursorAdapter Wizard fügt der Datei *Config.vfx* automatisch einen OLE-DB Verbindungseintrag zur ausgewählten Remote Datenbank hinzu.

VFX – Data Environment Builder

Der VFX – Data Environment Builder kann jetzt als eigenständiger Builder auf *Dataenvironment*-Klassen eingesetzt werden. Bereits in VFX 9.0 wurden *Dataenvironment*-Klassen für Formulare unterstützt. In VFX 9.5 kann nun der VFX – Data Environment Builder zur Bearbeitung dieser eigenständigen Klassen eingesetzt werden.

Im VFX – Data Environment Builder gibt es einige neue Einstellmöglichkeiten:

Where Clause – Dieser Wert wird nur bei Verwendung von CursorAdaptoren berücksichtigt. In dieser Spalte kann eine Where-Klausel eingetragen werden. Der eingegebene Wert wird in der Eigenschaft *cWhereClause* des CursorAdapters gespeichert. Zur Laufzeit wird diese Where-Klausel automatisch dem Select-Befehl aus der Eigenschaft *SelectCMD* eines CursorAdapters hinzugefügt, bevor der CursorAdapter mit Daten gefüllt wird. Dies hat den Vorteil, dass die Eigenschaft *SelectCMD* nicht auf Formularebene verändert werden braucht und der Wert in jedem Fall aus der basierenden CursorAdapter-Klasse vererbt wird. Bei einer Veränderung der Struktur der zugrunde liegenden Tabellen kann die CursorAdapter-Klasse mit dem VFX – CursorAdapter Wizard aktualisiert werden. Änderungen an Formularen sind nicht mehr erforderlich.

Wenn mit CursorAdaptoren gearbeitet wird und Primärschlüssel von der Datenbank vergeben werden, müssen in einem 1:n Szenario die vergebenen Primärschlüssel für den Parent-Datensatz als Fremdschlüssel in den Child-Datensätzen gespeichert werden. Dazu ist es erforderlich, dass zuerst der Parent-Datensatz gespeichert wird und nach dem Speichervorgang der Primärschlüssel aus der Datenbank gelesen wird. Im CursorAdapter für die Child-Daten muss der Fremdschlüssel spezifiziert werden. VFX speichert den Fremdschlüssel automatisch in allen Datensätzen eines Child-CursorAdapters. Zu diesem Zweck gab es bereits in VFX 9.0 in der VFX-CursorAdapter-Klasse die Eigenschaften *cForeignKeyName* und *cForeignKeyValue*. In VFX 9.5 werden die Werte dieser Eigenschaften automatisch ermittelt und vorbelegt und können im VFX – Data Environment Builder bearbeitet werden.

In VFP-Datenbanken kann zur Erzeugung von Primärschlüsseln der Datentyp *Integer (Autoinc)* und bei SQL Server-Datenbanken der Datentyp *Integer Identity* verwendet werden.

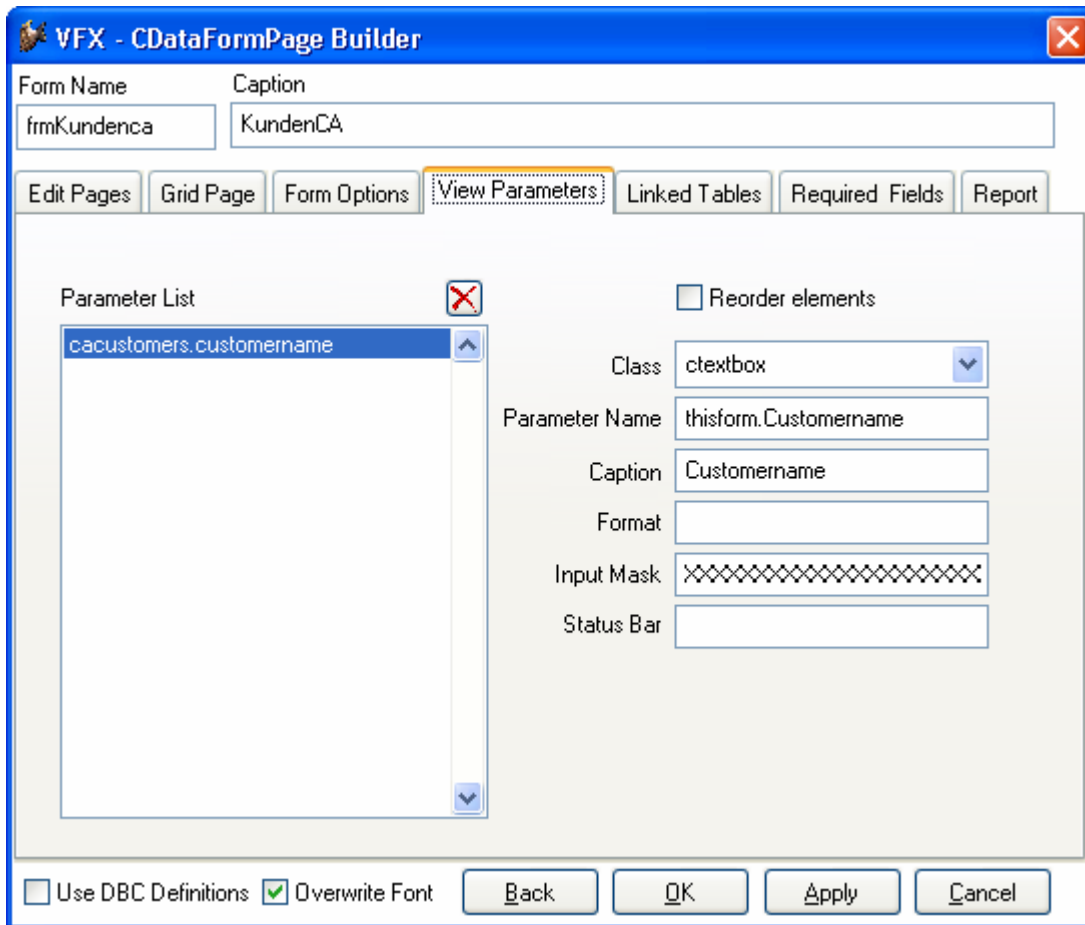
Foreign Key Name – Hier wird der Name des Feldes angegeben, in dem der Fremdschlüssel gespeichert werden soll. Der Name des Feldes wird in der Eigenschaft *cForeignKeyName* gespeichert.

Foreign Key Value – Hier wird der Name des Feldes aus dem Parent-CursorAdapter angegeben, das den neuen Primärschlüssel nach dem Speichern enthält. Der hier eingegebene Wert wird in der Eigenschaft *cForeignKeyValue* gespeichert. Hier kann auch ein Ausdruck eingegeben werden. Dieser Ausdruck wird evaluiert und dem in der Eigenschaft *cForeignKeyName* eingetragenen Feld zugewiesen.

Wenn CursorAdapter verwendet werden, die nicht auf der VFX-Klasse *cAppDataAccess* basieren, aber die Eigenschaften des VFX – Data Environment Builder trotzdem genutzt werden sollen, kann das Kontrollkästchen *Add Methods and Properties* markiert werden. Hierdurch werden dem CursorAdapter die benötigten Eigenschaften automatisch hinzugefügt.

VFX – Form Builder

Neu in VFX 9.5 kann in allen Formular Buildern die basierende Klasse für Steuerelemente zur Eingabe von Ansichtsparametern ausgewählt werden.



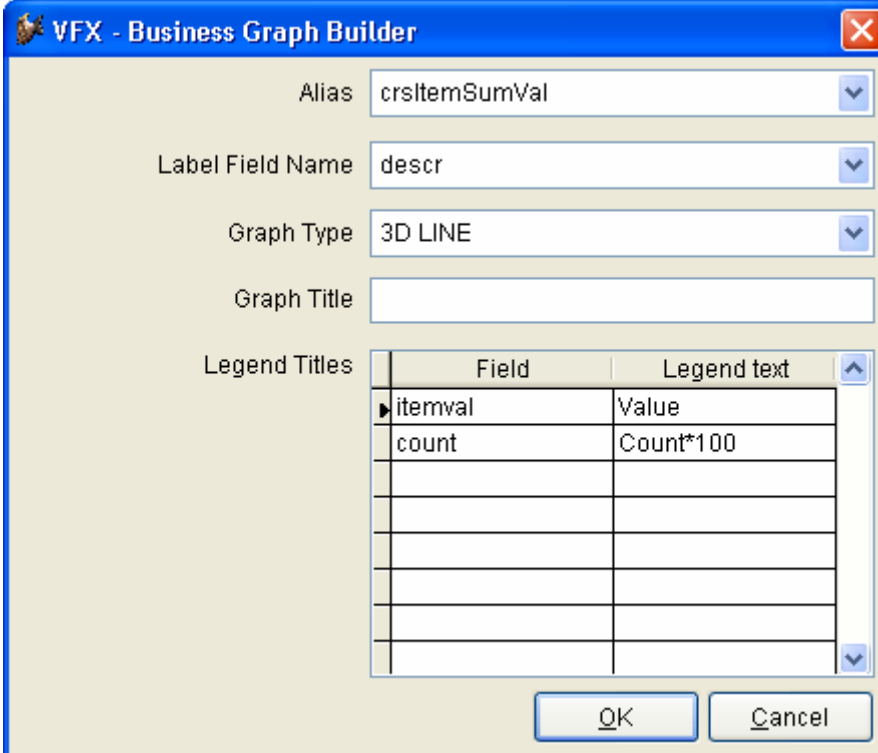
Die Einstellungen in den VFX – Form Buildern *Is Child Form* und *Has linked child form* werden nicht mehr benötigt und sind redundant. In VFX 9.5 kann jedes Formular ohne besondere Einstellung als Parent- und als Child-Formular dienen.

Wenn Formulare basierend auf CursorAdaptoren als Datenquellen erstellt werden, lesen die VFX – Form Builder jetzt die Eigenschaften aus den zugrunde liegenden Tabellen aus, wenn die Datenbanken zur Verfügung stehen.

Die VFX – Form Builder erstellen jetzt bei Seitenrahmen lokalisierte Überschriften für alle Seiten.

VFX – Business Graph Builder

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



Field	Legend text
itemval	Value
count	Count*100

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 – Parent/Child Builder

In früheren VFX-Versionen war es möglich in der *OnMore*-Methode Child-Formulare oder Methoden des Parent-Formulars aufzurufen oder auch ein *Wait Window* anzuzeigen. Alle diese drei Aufrufmöglichkeiten können in VFX 9.5 auch im VFX – Parent/Child Builder verwaltet werden. Dafür kann im Grid in der Spalte *Command Type* eine der drei Aufrufmöglichkeiten ausgewählt werden.

Zusätzlich zur Bearbeitungsmöglichkeit im Grid können jetzt alle Werte wahlweise auch in Textboxen unterhalb des Grids eingegeben werden.

Command Type	Child Form	...	Parent field (Fix Field Value)	Child field (F...
Child Form	AUFTRAGCA.SCX	📁	cacustomers.CUSTOMERID	caorders.CU
Child Form				
Wait Window				
Method				

Wenn das Kontrollkästchen *Auto Sync. Child Form* markiert ist, werden beim Bewegen des Satzzeigers im Parent-Formular automatisch alle Child-Formulare synchronisiert.

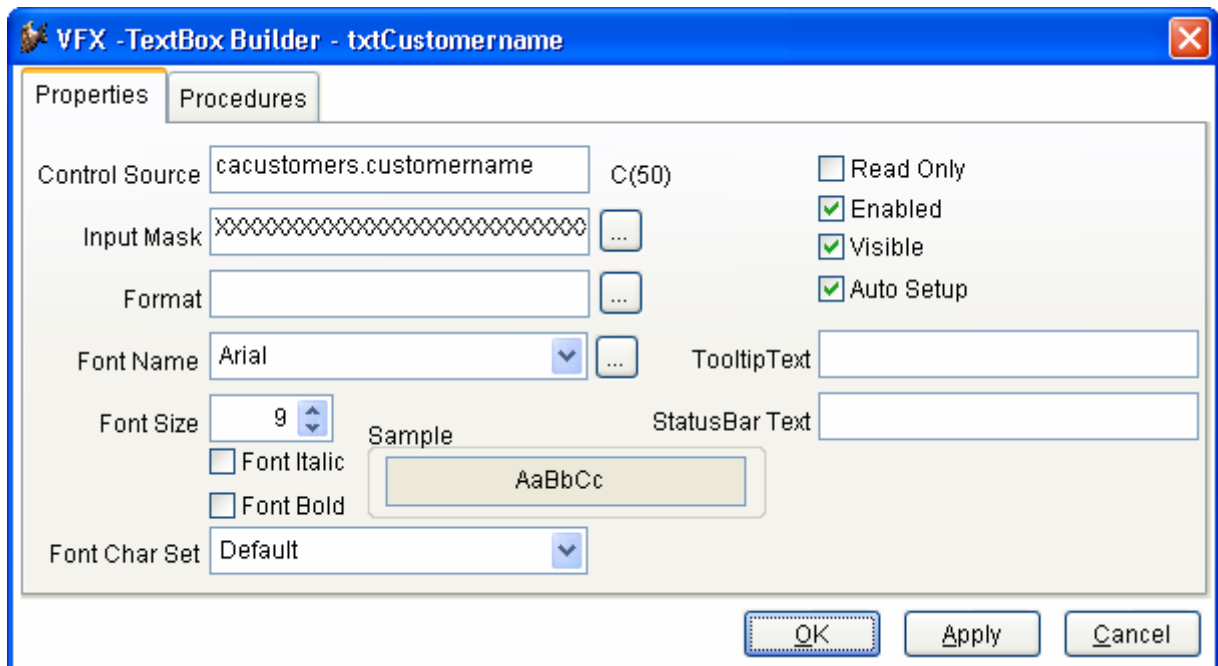
Bei der Auswahl eines Child-Formulars versucht der VFX – Parent/Child Builder möglichst die Informationen aus dem Parent- und dem Child-Formular zu ermitteln, um die folgenden Spalten im Grid richtig auszufüllen. In VFX 9.5 werden dabei auch Parent/Child-Szenarien basierend auf *CursorAdaptern* unterstützt. Als *Parent Field* wird dabei stets das Feld mit dem Primärschlüssel aus dem Parent-Formular vorgeschlagen.

VFX – TextBox Builder

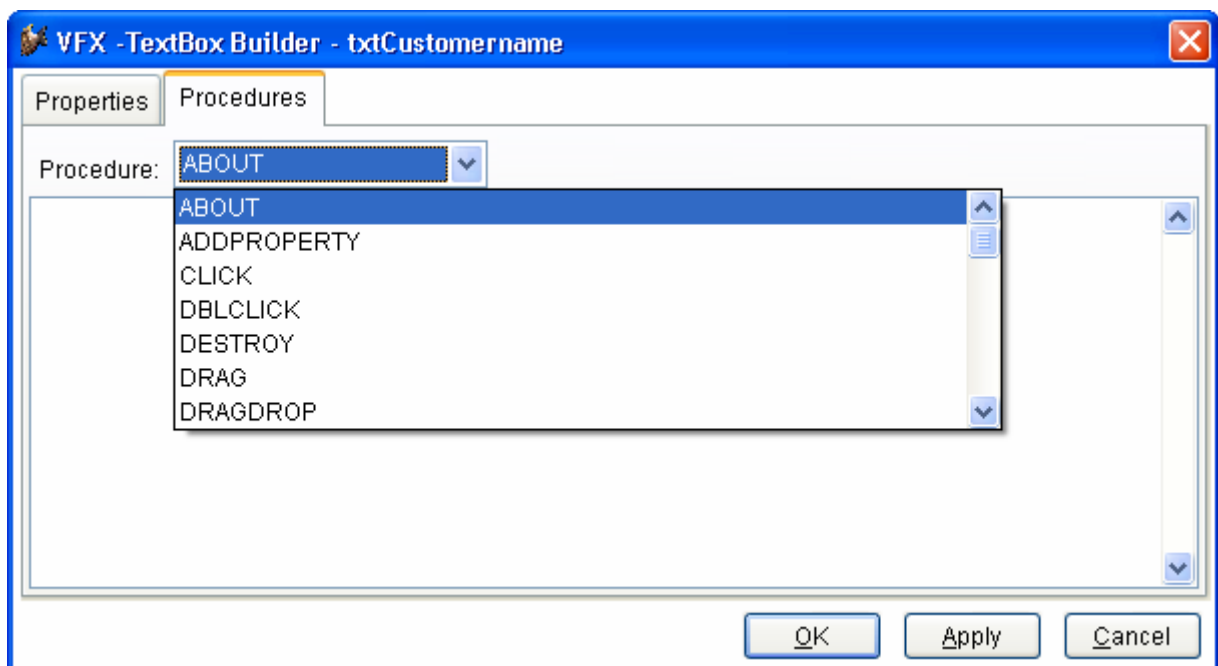
Im neuen VFX – TextBox Builder können die wichtigsten Eigenschaften von allen Steuerelementen basierend auf der VFP Basisklasse Textbox eingestellt werden. Auch die Bearbeitung sämtlicher Methoden ist hier möglich.

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.

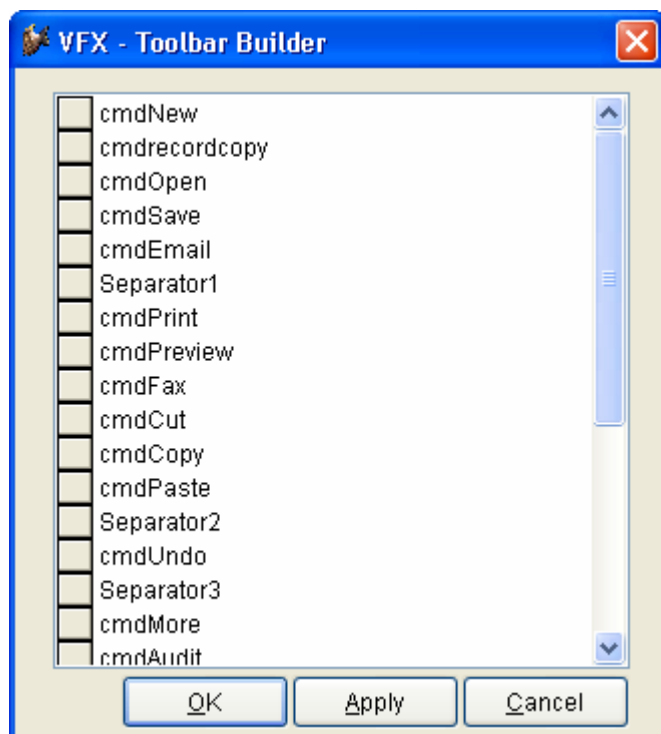


VFX – Toolbar Builder

Ein weiterer neuer Builder in VFX ist der Builder für Symbolleisten.

Der VFX – Toolbar Builder erleichtert es Steuerelemente und Separatoren innerhalb von Symbolleisten in eine neue Reihenfolge zu bringen. Dazu können die Elemente in einer Listbox nach oben oder unten bewegt werden.

Der VFX – Toolbar Builder kann im VFP Klassen-Designer mit der Auswahl Builder aus dem Kontextmenü gestartet werden, wenn eine Symbolleisten-Klasse geöffnet ist.



Die Anordnung der Elemente in der Listbox entspricht der Reihenfolge der Elemente in der Symbolleiste. Um die Reihenfolge zu ändern, können die Einträge mit den Mover-Bars verschoben werden.

Lokalisierung

VFX - Language Management Builder

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-Symboleiste zur Verfügung.

Active	Language	Abbreviation	LCID	IconFile	Compliance List	Font Char Set	Lang Message ID
<input checked="" type="checkbox"/>	English	ENG	09	bitmap\languages\FLGUK.ICO	09,0c,07,10,0a,02,08,13,16,19,05	0	LNG_ENGLISH
<input checked="" type="checkbox"/>	French	FRE	0c	bitmap\languages\FLGFRAN.ICO	09,0c,07,10,0a,13,16	0	LNG_FRENCH
<input checked="" type="checkbox"/>	German	GER	07	bitmap\languages\FLGGERM.ICO	09,0c,07,10,0a,13,16,05	0	LNG_GERMAN
<input checked="" type="checkbox"/>	Italian	ITA	10	bitmap\languages\FLGITALY.ICO	09,0c,07,10,0a,13,16	0	LNG_ITALIAN
<input type="checkbox"/>	Spanish	ESP	0a	bitmap\languages\FLGSPAIN.ICO	09,0c,07,10,0a,13,16	0	LNG_SPANISH
<input type="checkbox"/>	Bulgarian	BUL	02	bitmap\languages\FLGBUL.ICO	02	1	LNG_BULGARIAN
<input type="checkbox"/>	Greek	GRE	08	bitmap\languages\FLGGRE.ICO	08	161	LNG_GREEK
<input checked="" type="checkbox"/>	Dutch	NL	13	bitmap\languages\FLGNETH.ICO	09,0c,07,10,0a,02,13,16,19,05	0	LNG_DUTCH
<input checked="" type="checkbox"/>	Portuguese	POR	16	bitmap\languages\FLGPORT.ICO	09,0c,07,10,0a,13,16,05	0	LNG_PORTUGUESE
<input checked="" type="checkbox"/>	Russian	RU	19	bitmap\languages\FLGRUS.ICO	19	1	LNG_RUSSIAN
<input type="checkbox"/>	Czech	CZE	05	bitmap\languages\FLGCZE.ICO	05	238	LNG_CZECH
<input checked="" type="checkbox"/>	Finnish	FIN	0b	bitmap\languages\FLGFIN.ICO	0b	0	LNG_FINNISH
<input checked="" type="checkbox"/>	Polish	PL	15	bitmap\languages\FLGPOL.ICO	15	0	LNG_POLISH
<input checked="" type="checkbox"/>	ENU	ENU	33		33	0	LNG_ENU
<input checked="" type="checkbox"/>	Turkish	TR	1f	bitmap\languages\FLGTR.ICO	1f	0	LNG_TURKISH

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.

Ausführen der Methode LangSetup

Bei Aufruf einer *LangSetup*-Methode auf Formularebene werden automatisch alle auf dem Formular befindlichen Objekte nach Vorhandensein einer *LangSetup*-Methode durchsucht. Container-Objekte werden rekursiv durchsucht. Die *LangSetup*-Methode wird so in allen Objekten ausgeführt.

VFX - LangSetup Builder

Der VFX – LangSetup Builder unterstützt in VFX 9.5 nicht nur Formulare, sondern auch Berichte sowie den Öffnen-Dialog.

Um den VFX – LangSetup Builder auf einer Berichtsdatei anzuwenden, muss die Berichtsdatei zunächst im VFP Berichts-Designer geöffnet werden. Jetzt kann der VFX – LangSetup Builder aus dem VFX-Menü gestartet werden. Es erscheint folgende Messagebox:



Wenn mit „ja“ geantwortet wird, fügt der VFX – LangSetup Builder für jede Bezeichnung in dem Bericht einen Datensatz an die Tabelle *Vfxmsg.dbf* an. Die Bezeichnungen im Bericht werden durch „^“ gefolgt von der *Message Id* des neuen Datensatzes in *Vfxmsg.dbf* ersetzt. Zur Laufzeit ersetzt ein ReportListener die Bezeichnungen im Bericht durch die lokalisierten Texte aus der Tabelle *Vfxmsg.dbf*.

Wenn nur ein Projekt, aber kein Formular oder Bericht im Designer geöffnet ist und der VFX – LangSetup Builder gestartet wird, wird die Tabelle *Vfxopen.dbf* lokalisiert.

Project Hook

Bereits in VFX 9.0 wurden zur Laufzeit Objekte angelegt für alle Felder aus *Vfxsys.dbf* und aus dem aktuellen Datensatz aus *Vfxusr.dbf*. Aus Gründen der Kompatibilität mit früheren VFX-Versionen gibt es die Datei *Vfxglobal.h*, die Ersetzungen für alle Variablennamen enthält, die in früheren VFX-Versionen verwendet wurden. In VFX 9.5 wird die Datei *Vfxglobal.h* automatisch bei jeder Neuerstellung des Projekts oder beim Erstellen einer App- oder Exe-Datei neu angelegt. Dabei werden alle Felder, also auch selbst hinzugefügte, aus den Dateien *Vfxsys.dbf* und *Vfxusr.dbf* berücksichtigt.

Produktaktivierung

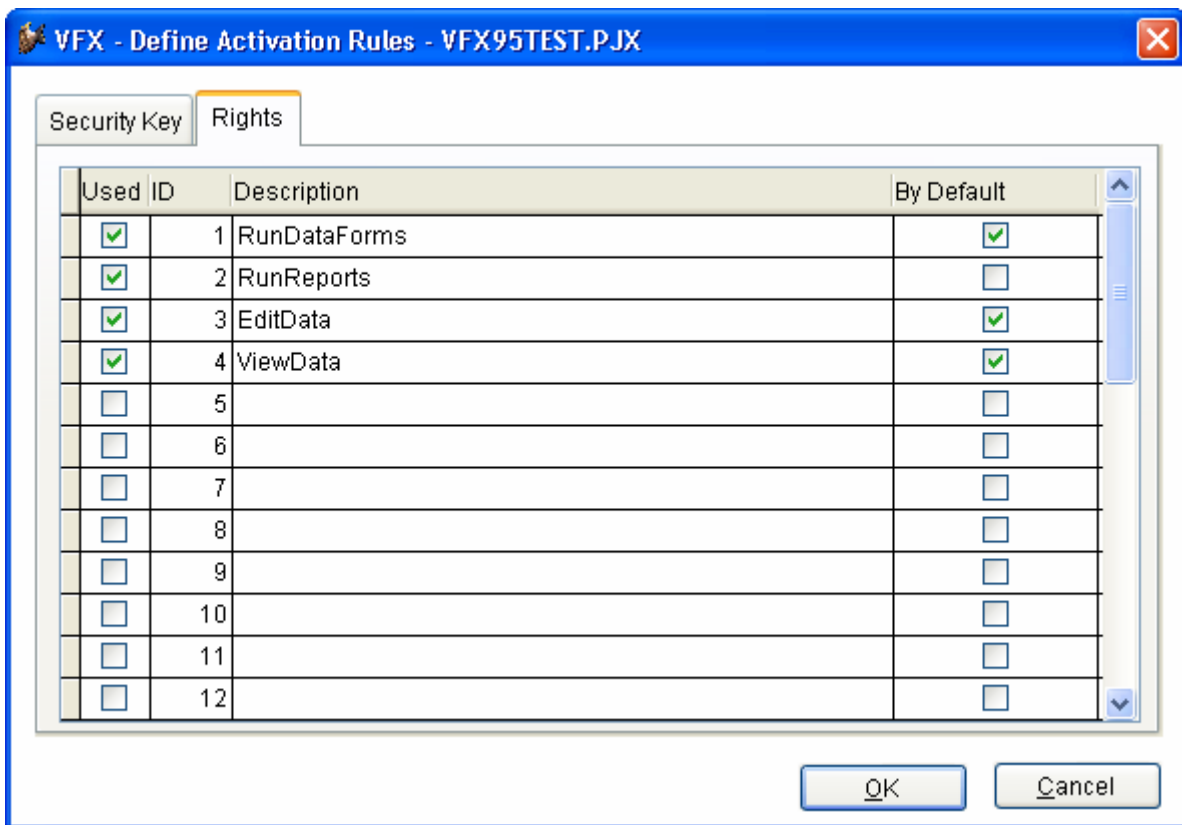
VFX 9.5-Anwendungen können vor unberechtigter Nutzung durch eine Produktaktivierung geschützt werden. Die Produktaktivierung funktioniert so, wie es im Benutzerhandbuch beschrieben ist. Hier werden nur die Erweiterungen in VFX 9.5 erläutert.

Für die in VFX integrierte Produktaktivierung steht jetzt ein Web Service zur Verfügung. Ähnlich wie bei der Aktivierung von VFX können sich Benutzer jetzt über einen Web Service einen Aktivierungsschlüssel für eine Anwendung holen.

Zur Verwaltung der Aktivierungsschlüssel und Kundendaten steht die neue Anwendung VFX – Kundenverwaltung zur Verfügung.

Definieren der Aktivierungsregeln

Zunächst müssen jedoch in der Anwendung die Aktivierungsregeln festgelegt werden. Dies geschieht im Dialog *VFX – Define Activation Rules*. Die Definition der Regeln geschieht genau so, wie in VFX 9.0. Die Aktivierungsregeln werden in der Klassenbibliothek *Appl.vcx* in der Klasse *cVfxActivation* in der Eigenschaft *cActPattern* verschlüsselt gespeichert.



Um ein Recht zu aktivieren muss zunächst die Checkbox in der ersten Grid-Spalte markiert werden. In der Spalte *Description* muss dem Recht ein Name gegeben werden. Zur Laufzeit wird dem Objekt *SecurityRights* eine Eigenschaft mit dem gewählten Namen hinzugefügt.

In der Spalte *By Default* kann eingestellt werden, ob dieses Recht standardmäßig aktiviert werden soll. Die Standardwerte gelten für neu angelegte Benutzer, können in der Kundenverwaltung je Benutzer geändert werden.

Die Registrierungsnummer ist ein numerischer Wert mit 10 Stellen Länge. Der Benutzer muss die Registrierungsnummer dem Entwickler mitteilen oder per E-Mail senden. Der Entwickler erfasst in der VFX – Kundenverwaltung einen neuen Datensatz für diesen Benutzer und gibt hier die Registrierungsnummer ein.

Aktivierungsschlüssel erstellen

Der Aktivierungsschlüssel enthält die Berechtigungen, für die einzelnen Module der Anwendung. Aktivierungsschlüssel für VFX 9.5-Anwendungen werden mit der VFX – Kundenverwaltung erstellt. Die VFX – Kundenverwaltung ist ein eigenständiges Projekt und wird mit VFX geliefert. Aus dem Projekt VFX – Kundenverwaltung kann eine Exe-Datei erstellt werden. Die Verwaltung der Kundendaten und die Generierung von Aktivierungsschlüsseln kann so auf einem PC unabhängig vom Entwicklungsrechner durchgeführt werden.

Damit die VFX – Kundenverwaltung Aktivierungsschlüssel erstellen kann, müssen ihr die Aktivierungsregeln bekannt sein. Die Aktivierungsregeln sind aber in der eigentlichen Anwendung in der Klassenbibliothek *Appl.vcx* gespeichert.

Bei der Generierung eines Aktivierungsschlüssels benutzt die VFX – Kundenverwaltung die Registrierungs-Dll. Das Projekt zur Erstellung der Registrierungs-Dll befindet sich unterhalb des Projektordners der Anwendung und wird vom VFX – Application Wizard in jedes neue Projekt kopiert. Der Name des Registrierungsprojekts ist: „Register“ + <Name der Anwendung>

Bei der Erstellung der Registrierungs-Dll werden über einen Projekt Hook die Aktivierungsregeln aus der Klassenbibliothek *Appl.vcx* der Anwendung gelesen und in der Klasse *cregdll* in der Eigenschaft *cActPatternName* gespeichert. Die Registrierungs-Dll enthält also die Aktivierungsregeln. Dadurch ist die VFX – Kundenverwaltung unabhängig von einer bestimmten Anwendung. Die VFX – Kundenverwaltung kann Registrierungs-Dlls für verschiedene Anwendungen benutzen. Die Registrierungs-Dll kann aus dem VFX-Menü über den Menüpunkt *Activation, Build register DLL* erstellt werden.

In der Registrierungs-Dll befindet sich die Methode *generateactkey*. Die Parameter dieser Methode sind die Registrierungsnummer, eine Zeichenkette mit den zu erteilenden Rechten sowie der Pfad zur Datei *VFXGenActKey.app*. Der Rückgabewert ist der generierte Aktivierungsschlüssel.

Die Registrierungs-Dll ruft eine Funktion der Anwendung *VFXGenActKey.app* auf. *VFXGenActKey.app* wird mit VFX geliefert und befindet sich im Projektordner der VFX – Kundenverwaltung. *VFXGenActKey.app* enthält den Algorithmus, der anhand der Aktivierungsregeln einen Aktivierungsschlüssel erstellt. Der Quell-Code von *VFXGenActKey.app* wird nicht mit VFX geliefert. Entwickler, die den Quell-Code zur Verfügung hätten, könnten Aktivierungsschlüssel für VFX-Anwendungen anderer Entwickler erstellen.

Die VFX – Kundenverwaltung greift auf die Kundendatenbank über die Datei *Config.vfx* zu. Die Kundendatenbank kann sich in einer VFP-Datenbank oder in einer SQL Server-Datenbank befinden. Der Datenzugriff erfolgt genauso, wie bei anderen VFX-Anwendungen auch. Für die VFX – Kundenverwaltung wurde der Datei *Config.vfx* eine Spalte hinzugefügt. Die Spalte *RegDllName* enthält den Namen der zu verwendenden Registrierungs-Dll.

VFX – Kundenverwaltung

Diese VFX-Anwendung enthält zwei Formulare: Kundenverwaltung und Versionsverwaltung.

The screenshot shows a Windows-style application window titled "Customer management". It has a "Customers" tab and a "List" button. The form is divided into several sections:

- User account information:** E-mail: Uwe.Habermann@dFPUG.de
- Customer information:** Customer number: 1, First name: Uwe, Last name: Habermann, Company: ISYS GmbH, Street: (empty), Zip Code: (empty), City: Kronberg, State: (empty), Country: Germany (dropdown). Checkboxes: registered (checked), e-mail notification (unchecked).
- Bank account information:** Phone, Fax, Tax ID number, Bank name, Bank code, Bank account (all empty).
- Registration information:** Registration number: 1234567890, Activation key: (empty), Registration date: 10/30/2005 12:00:00 AM, Last updated: // : : AM. Buttons: "Generate Activation key" and "Save Activation key as xak file".
- Permissions:** Allow update application download.

At the bottom, there is a table with columns "ID", "Description", and "User has this right".

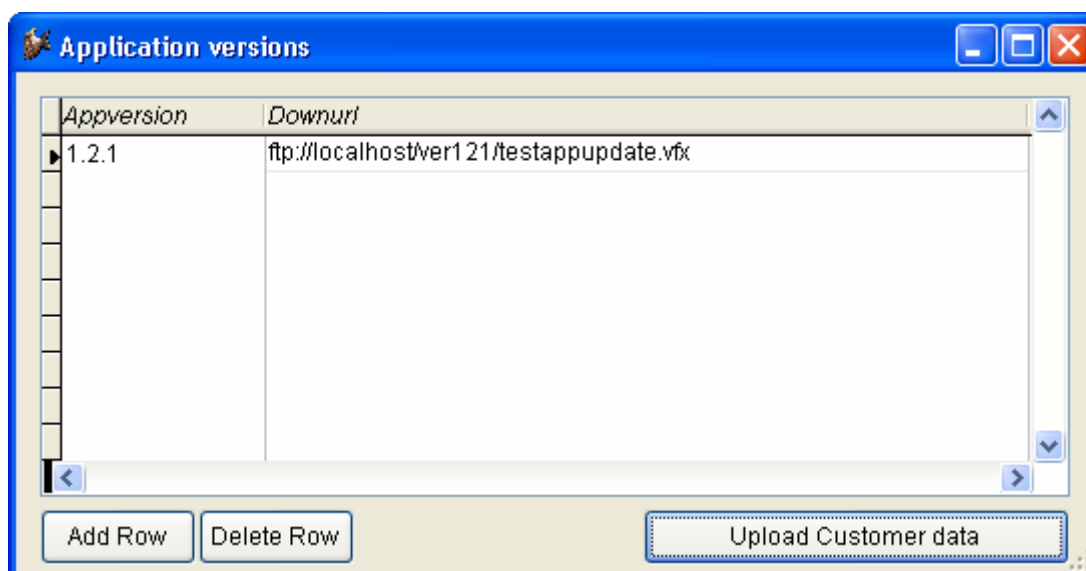
ID	Description	User has this right
1	Rule 1	<input checked="" type="checkbox"/>
2	Rule 2	<input checked="" type="checkbox"/>
3	Rule 3	<input checked="" type="checkbox"/>
4	Rule 4	<input type="checkbox"/>
5	Rule 5	<input type="checkbox"/>
6	Rule 6	<input checked="" type="checkbox"/>
7	Rule 7	<input type="checkbox"/>

Im Formular Kundenverwaltung können die Kundendaten bearbeitet werden. Zu jedem Kunden werden die Registrierungsnummer und die vergebenen Rechte gespeichert. Die Rechte können bei Bedarf verändert werden und es kann aus diesem Formular ein neuer Aktivierungsschlüssel generiert werden.

Über die Schaltfläche *Generate Activation Key* kann ein Aktivierungsschlüssel entsprechend der eingegebenen Registrierungsnummer und Benutzerrechte erstellt werden. Der generierte Aktivierungsschlüssel wird in der Kundenverwaltung gespeichert.

Über die Schaltfläche *Save Activation key as xak file* kann der angezeigte Aktivierungsschlüssel in einer Datei mit dem Namen *<Projektname>.xak* im aktuellen Ordner gespeichert werden. Diese Datei kann an den Kunde geschickt und im Ordner der Anwendung gespeichert werden. Beim Start der Anwendung wird der Aktivierungsschlüssel automatisch aus der Datei gelesen und die Anwendung wird damit aktiviert.

Mit dem Formular Versionsverwaltung werden die Versionen und Download-Links der Anwendung verwaltet. Neue Versionen der Anwendung können auf einem Internet Server bereitgestellt werden. Wenn ein Kunde seine Anwendung aktualisieren will, lädt die Anwendung zunächst die Datei *UpdateCustomers.vfx* herunter. In dieser Datei befinden sich die Registrierungsnummern, der zur Aktualisierung berechtigten Kunden. Wenn die Aktualisierungsberechtigung besteht, wird die Datei *Updateversions.vfx* heruntergeladen. In dieser Datei befinden sich die Download-Links zu den verfügbaren Anwendungsversionen. Die Download-Links zu den Dateien *UpdateCustomers.vfx* und *UpdateVersions.vfx* sind in der Anwendung gespeichert und können mit dem VFX – Application Builder eingestellt werden.



Durch einen Mausklick auf die Schaltfläche *Upload Customer Data* werden die Dateien *UpdateCustomers.vfx* und *UpdateVersions.vfx* erstellt.

Damit diese Dateien auf den Internet Server hochgeladen werden können, müssen die Anmeldeinformationen im Projekt der Registrierungs-Dll in der Klassenbibliothek *Regdll.vcx* in der Klasse *cregdll* gespeichert werden.

Eigenschaften

cFtpUrl – Domainname eines FTP-Servers.

cFtpDir – Verzeichnis auf dem FTP-Server, in dem die Dateien *UpdateCustomers.vfx* und *UpdateVersions.vfx* gespeichert werden sollen.

cPort – Zu verwendender Übertragungsport. Standardmäßig wird für FTP-Uploads der Port 21 verwendet.

cUserName – Benutzername für den FTP-Zugang.

cPassword – Kennwort für den FTP-Zugang.

Web Service für die Registrierung

Mit der VFX – Kundenverwaltung wird das Projekt *RegistrationWebService.pjx* im Ordner *RegistrationWebService* unterhalb der VFX – Kundenverwaltung geliefert. Aus diesem Projekt kann ein COM Server erstellt werden, der auf einem Internet Information Server als Web Service installiert werden kann. An diesen Web Service können Anwendungen die Registrierungsdaten von Benutzern senden. Der Web Service kann an die Anwendung einen Aktivierungsschlüssel senden.

Der Web Service benutzt eine *Config.vfx* für den Datenzugriff. Hier muss auf die gleiche Datenbank gezeigt werden, die auch von der VFX – Kundenverwaltung verwendet wird. Im einfachsten Fall kann der Web Service im gleichen Ordner wie die VFX – Kundenverwaltung installiert werden und so die gleiche *Config.vfx* benutzen, wie die Kundenverwaltung.

Wenn der Web Service auf einem entfernten Internet Server laufen soll, so ist die Kundendatenbank mit dem VFX – Upsizing Wizard auf einen SQL Server zu portieren. Auf die so erzeugte SQL Server Datenbank können sowohl der Web Service, als auch die VFX – Kundenverwaltung über das Internet zugreifen.

Damit eine Anwendung über den Web Service aktiviert werden kann, müssen in der Anwendung ein paar Einstellungen im Formular *VfxRegister.scx* gemacht werden:

cWSDL – Enthält die URL der WSDL Datei. Diese Datei wird bei der Registrierung des Web Service mit dem SOAP Toolkit auf dem Internet Server generiert.

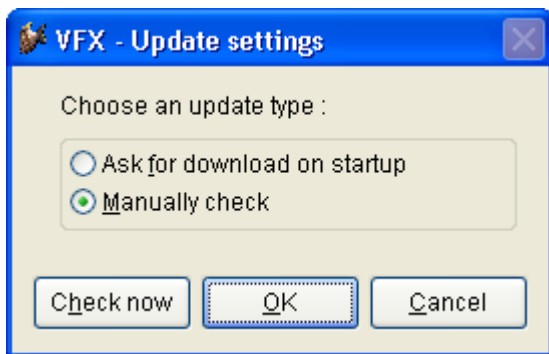
cServiceName – Enthält den Namen des Web Service.

cServiceMethodName – Enthält den Namen der verwendeten Web Service Methode. Standardmäßig ist dies die Methode *GenerateActKey*.

Wenn sich ein Kunde über den Web Service registriert, werden die Registrierungsdaten an den Web Service im XML Format übertragen. Der Web Service sucht in der Kundendatenbank nach einem Datensatz mit der gleichen E-Mailadresse sowie dem gleichen Anwendungsnamen und der gleichen Version. Wenn dieser Datensatz gefunden wird, wird der dort eingetragene Aktivierungsschlüssel an den Kunden übertragen. Die Anwendung wird dabei automatisch aktiviert. Wenn kein solcher Datensatz gefunden wird, werden die Registrierungsinformationen in der Kundendatenbank gespeichert.

Aktualisierung von VFX

Es ist sinnvoll VFX regelmäßig zu aktualisieren, damit immer der aktuelle Stand zur Verfügung steht. VFX 9.5 kann automatisch auf Aktualisierungen prüfen. Dies kann im Dialog *Update Settings* eingestellt werden. Wenn die Option *Ask for download on startup* gewählt ist, überprüft VFX bei jedem ersten Start an jedem Tag, ob ein aktualisierter Build zur Verfügung steht. Falls ja, wird gefragt, ob der neue Build heruntergeladen und installiert werden soll. Die Überprüfung wird nicht durchgeführt, wenn keine Verbindung mit dem Internet besteht. Über die Schaltfläche *Check for updates now* kann die Überprüfung nach aktualisierten Builds jederzeit manuell gestartet werden.



Die Klasse *cUpdate*

Diese Klasse stellt den Dialog zur Bearbeitung der Aktualisierungseinstellungen bereit. Über eine Schaltfläche kann sofort geprüft werden ob Aktualisierungen zur Verfügung stehen. Die Klasse wird sowohl von VFX intern als auch in Endanwendungen verwendet.

tlUpdateApp - Initialisierungsparameter.

- .T. – Aktualisierung für Endanwendungen.
- .F. – Aktualisierung für VFX.

Methode

WriteData() – Speichert die gewählte Methode zur Aktualisierung in der Tabelle *Vfxsys*.

Die Klasse *cUpdateEngine*

Mit dieser Klasse wird die Aktualisierung der Anwendung beim Kunden durchgeführt. Abhängig von den Einstellungen im Dialog für die Aktualisierungseinstellungen wird eine Datei heruntergeladen, die die aktuell verfügbare Versionsnummer enthält. Um eine neue Programmversion herunterzuladen wird ein Objekt der Klasse *cDownload* instanziiert.

Eigenschaften

cINI_URL – URL der Datei, die die aktuelle Versionsnummer enthält.

cVersion – Versionsnummer der aktuell laufenden Anwendung.

dLastChecked – Datum der letzten Überprüfung auf Aktualisierungen. Dieses Datum wird aus der Tabelle *Vfxsys* gelesen.

nManualCheck – Wie wurde die Überprüfung auf Aktualisierungen gestartet?

- 0 – Automatische Überprüfung auf Aktualisierungen bei Anwendungsstart.
- 1 – Manuelle Überprüfung auf Aktualisierungen aus dem Menü.

nUpdateType – Aktualisierungstyp. Dieser Wert wird aus der Tabelle *Vfxsys* gelesen.

- 1 – Automatisches herunterladen und installieren, wenn Aktualisierungen vorhanden sind. Die Überprüfung wird beim ersten Anwendungsstart täglich durchgeführt.
- 2 – Automatisches herunterladen von Aktualisierungen mit anschließender Frage, ob die Aktualisierung installiert werden soll. Die Überprüfung wird beim ersten Anwendungsstart täglich durchgeführt.
- 3 – Überprüfung nach verfügbaren Aktualisierungen täglich beim ersten Anwendungsstart. Bei vorhandener aktualisierter Programmversion wird der Benutzer gefragt, ob er die Aktualisierung herunterladen und installieren will.
- 4 – Manuelle Überprüfung. Die Überprüfung auf Aktualisierungen kann aus dem Menü aufgerufen werden.

Methoden

CheckUpdType()

Ermittelt die Versionsnummer der laufenden Anwendung sowie den Aktualisierungstyp und das Datum der letzten Überprüfung auf Aktualisierung. Die ermittelten Werte werden in den Eigenschaften *cVersion*, *nUpdateType* und *dLastChecked* gespeichert. Wenn alle Werte erfolgreich ermittelt werden können, wird .T. zurückgegeben. Bei Auftreten eines Fehlers, wird .F. zurückgegeben.

DoUpdate(tnUpdateType)

tnUpdateType = *nUpdateType*

Aufruf der Methode *UpdateApp(tnUpdateType)* (in *Vfxfunc.prg*).

NewVersion()

Herunterladen der Datei aus *cINI_URL*. Rückgabewert:

- 0 – Es ist keine neue Version verfügbar.
- 1 – Eine neue Version ist verfügbar.
- 2 – Es ist ein Fehler aufgetreten.

SetNewCheckDate(tdDate)

tdDate – Datum, an dem die nächste Überprüfung auf Aktualisierungen stattfindet. Das als Parameter übergebene Datum wird in der Tabelle *Vfxsys* im Feld *lastcheckd* gespeichert.

StartUpdate(nManualCheck)

nManualCheck – Einstellen des Aktualisierungstyps.

0/.F. – Automatische Aktualisierung.

1/.T. – Manuelle Aktualisierung.

Dies ist die wichtigste Methode dieser Klasse. Hiermit wird die Aktualisierung gestartet. Bevor diese Methode aufgerufen wird, muss der Eigenschaft *cINI_URL* die URL mit der Versionsdatei zugewiesen werden. Aus dieser Methode werden die Methoden *CheckUpdType()* und *NewVersion()* aufgerufen.

Wenn eine neue Version verfügbar ist, werden die Methoden *SetNewCheckDate(tdDate)* und *DoUpdate(tnUpdateType)* aufgerufen. Bei manueller Überprüfung auf Aktualisierungen wird nur die Methode *DoUpdate(tnUpdateType)* aufgerufen.

AFX Unterstützung

In der Klasse *cFoxAppl* befinden sich zwei neue Methoden, die die Generierung von Web-Anwendungen basierend auf VFX-Projekten vereinfachen. Es sind die Methoden *VFXMessageBox* und *VFXWaitWindow*. Diese Methoden akzeptieren die gleichen Parameter wie die entsprechende VFP-Funktion *MessageBox* bzw. der Befehl *Wait Window*.

Methoden

VFXMessageBox()

LPARAMETERS *emessageText, ndialogboxtype, ctitlebartext, ntimeout*

emessageText – Anzuzeigender Text in der Messagebox. Wenn dieser Wert fehlt oder von falschem Typ ist, wird eine leere Zeichenkette angezeigt.

ndialogboxtype – Typ der Messagebox. Hierüber können Schaltflächen und das Icon eingestellt werden. Der Standardwert ist 0.

ctitlebartext – Titel der Messagebox. Wenn dieser Wert fehlt oder von falschem Typ ist, wird eine leere Zeichenkette angezeigt.

ntimeout – Zeitspanne zur Anzeige der Messagebox. Wenn dieser Wert fehlt oder von falschem Typ ist, wird eine leere Zeichenkette angezeigt.

Diese Methode führt die VFP-Funktion MESSAGEBOX() aus. Wenn die Anwendung als Web-Anwendung läuft, werden Web-Seiten angezeigt. Wenn die Anwendung ohne sichtbare Ausgaben läuft, wird der Wert der Standardschaltfläche der Messagebox zurückgegeben.

VFXWaitWindow()

LPARAMETERS *tcMessageText, tnRow, tnColumn, tlNowait, tlClear, tlNoclear, tnTimeout*

tcMessageText – Anzuzeigender Text. Der Standardwert ist eine leere Zeichenkette.

tnRow – Zeilennummer des Wait Window. Nur in Zusammenhang mit *tnColumn* verwendbar.

tnColumn – Spaltennummer des Wait Window. Nur in Zusammenhang mit *tnRow* verwendbar.

tlNowait - .T. um die Programmausführung nach Anzeige des Wait Window fortzusetzen, .F. um die Programmausführung anzuhalten bis der Benutzer eine Taste oder Maustaste drückt. Der Standardwert ist .F.

tlClear - .T. um aktuell angezeigte Wait Windows zu löschen. Der Standardwert ist .F.

tlNoclear - .T. und das Wait Window angezeigt zu lassen, bis der Befehl WAIT CLEAR ausgeführt wird. Der Standardwert ist .F.

tnTimeout – Anzeigedauer des Wait Window. Wenn 0 angegeben wird, wird das Wait Window nicht gelöscht, bis der Befehl WAIT CLEAR ausgeführt wird. Der Standardwert ist 0.

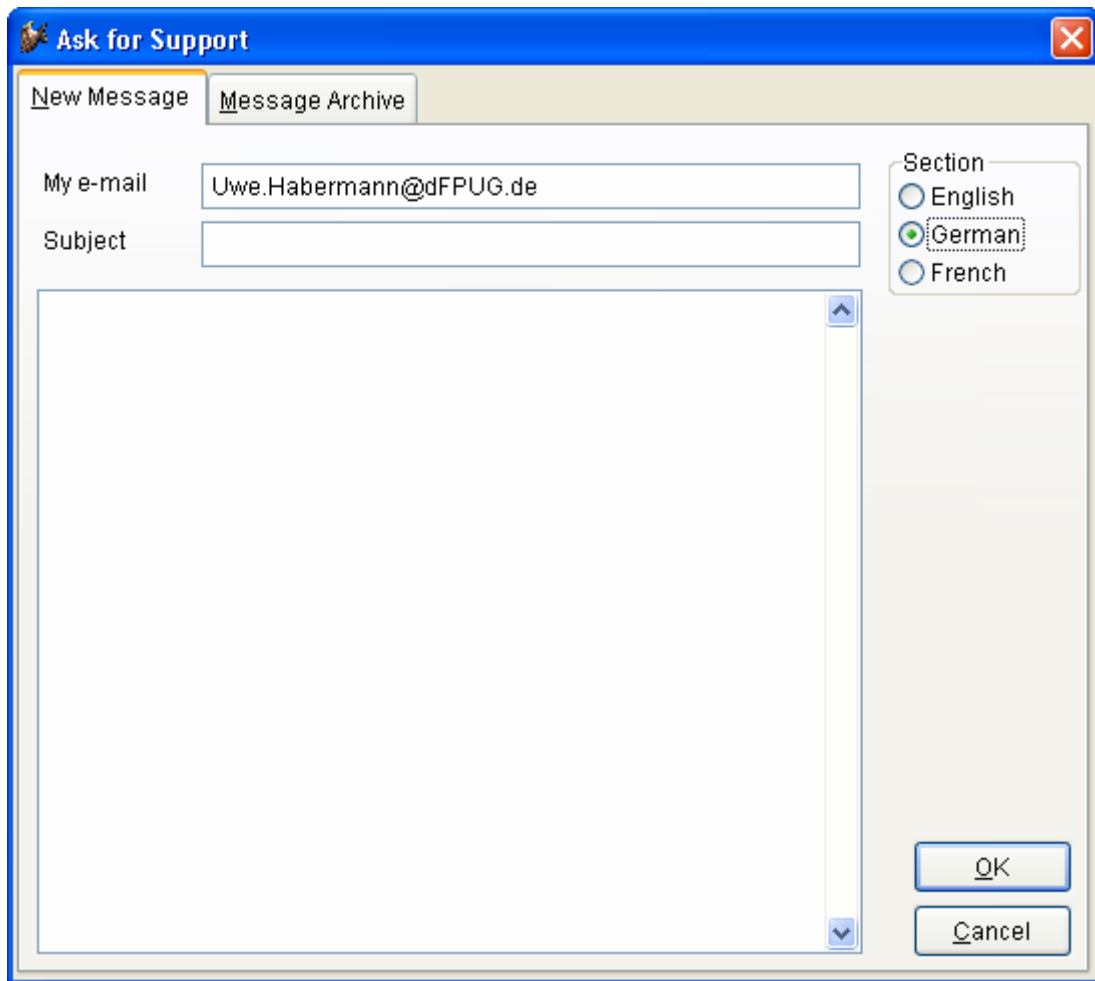
Diese Methode führt den VFP-Befehl WAIT WINDOW aus. Wenn die Anwendung ohne sichtbare Ausgaben läuft, erfolgt keine Anzeige.

Hilfe

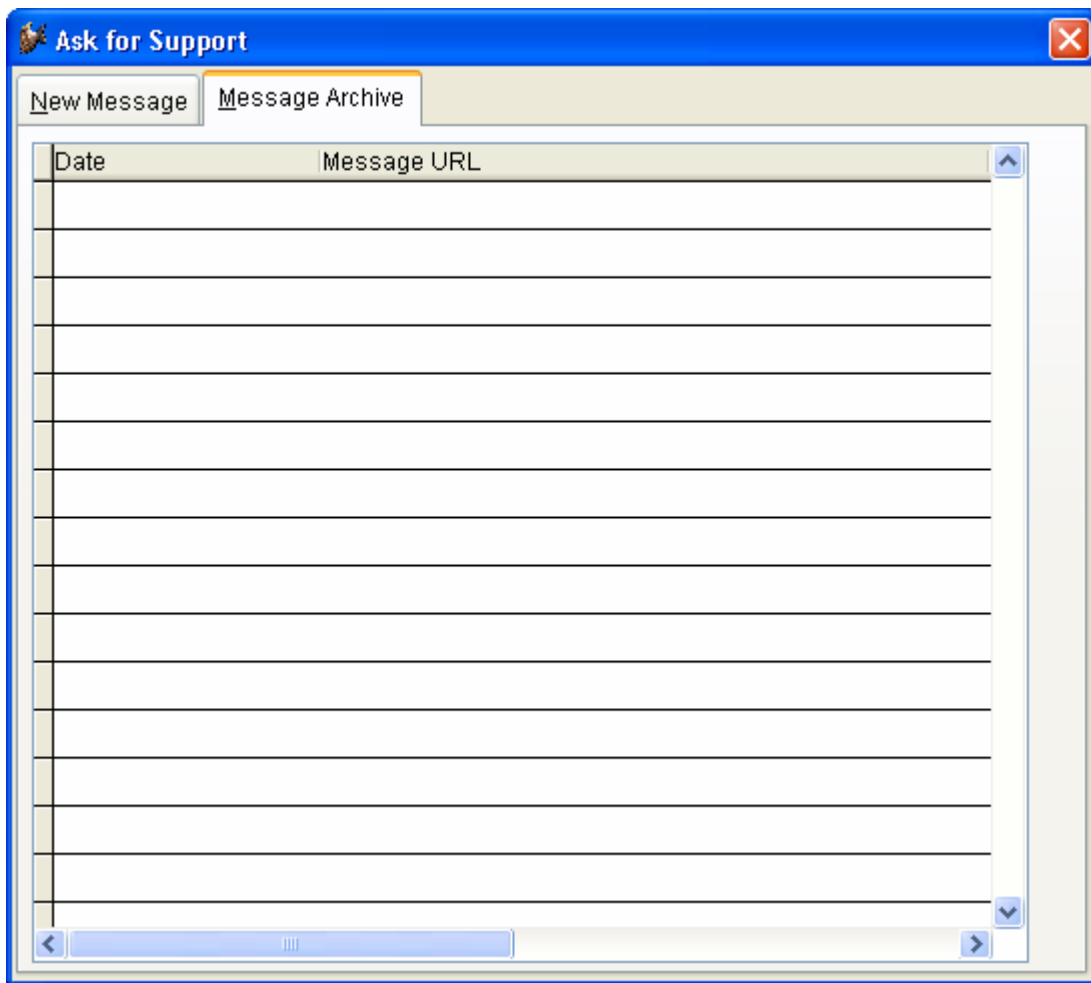
In der neuen Hilfe-Sektion des VFX-Menüs kann das Benutzerhandbuch geöffnet werden und es werden nützliche Informationen rund um VFX sowie Links zu Online Ressourcen angeboten.

Support-Anfragen an das Forum richten

Es gibt zwei Möglichkeiten um direkt aus VFX eine Support-Anfrage im Forum zu publizieren.



Zum einen kann aus dem VFX-Menü über den Menüpunkt *VFX Help, Visual Extend Online, Forum* der Internet Browser gestartet werden. Es öffnet sich die Startseite des Forums. Hier können Nachrichten online gelesen und auch neue Nachrichten erstellt werden. Zum anderen kann über den Menüpunkt *VFX Help, Ask for Support* ein Dialog geöffnet werden, in dem offline eine Forumsnachricht verfasst werden kann. Zum Versenden einer Nachricht ist eine Internetverbindung erforderlich. Die auf diesem Weg verfassten Nachrichten bleiben gespeichert und können später auf der Seite *Message Archive* angesehen werden.



Benutzerhandbuch und Dokumentation der Neuheiten

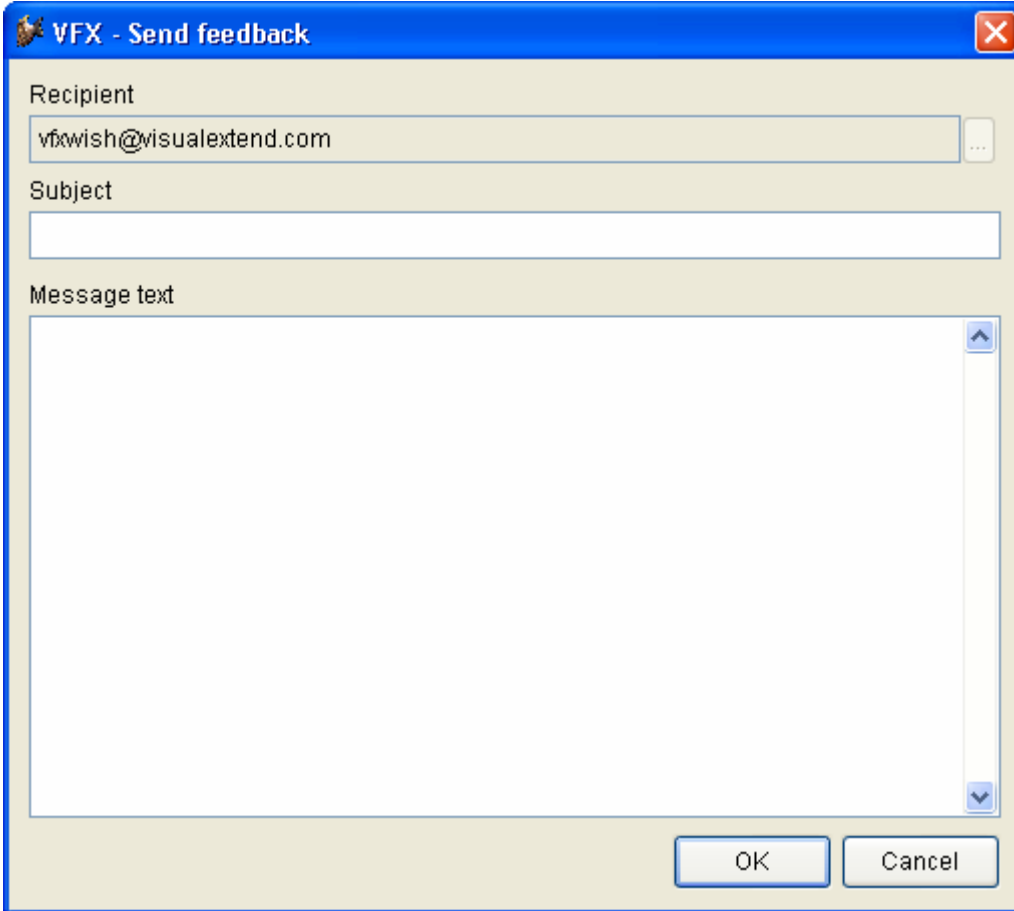
Über den Menüpunkt *VFX Help, User Manuals and What's New* können das Benutzerhandbuch und die Neuheitendokumentation in deutscher und englischer Sprache geöffnet werden. Alle diese Dokumente liegen als PDF-Dateien vor. Zum Öffnen ist der Adobe Reader erforderlich.

Über den Menüpunkt *Update Notes* kann ein Dokument geöffnet werden, das alle Änderungen in der VFX 9.5-Template-Anwendung seit dem Erscheinen beschreibt.

Wenn eins der anzuzeigenden Dokumente nicht installiert ist, wird das entsprechende Dokument automatisch heruntergeladen.

Senden Sie uns eine E-Mail!

Machen Sie Vorschläge, sagen Sie uns Ihre Meinung oder äußern Sie Erweiterungswünsche. Über den Menüpunkt *VFX Help* -> *Send user feedback* können Sie dem VFX-Team eine E-Mail senden.



The image shows a Windows-style dialog box titled "VFX - Send feedback". It has a blue title bar with a close button (X) in the top right corner. The dialog contains three input fields: "Recipient" with the email address "vfxwish@visualextend.com", "Subject" which is currently empty, and "Message text" which is a large empty text area with a vertical scrollbar on the right. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Visual Extend Online

Über den Menüpunkt *VFX Help* -> *Visual Extend Online* erreichen Sie direkt die folgenden Seiten der VFX Website:

- VFX Startseite (zurzeit in vier Sprachen verfügbar - deutsch, englisch, französisch und spanisch)
- VFX Directory (Startseite des Visual Extend Portals)
- Forum (lesen und schreiben Sie im Online-Forum)
- Newsletters (abonnieren Sie aktuelle Informationen über VFX)
- DevCons (besuchen Sie die Entwicklerkonferenz!)
- Online shop (kaufen Sie VFX, Bücher, VFP und anderes)

So erreichen Sie uns

Über den Menüpunkt *VFX Help*, *How to reach us* wird Ihnen angezeigt, wie Sie uns erreichen können.

Info

Informationen über die installierte Visual Extend Version sowie Registrierungsinformationen erhalten Sie über den Menüpunkt *VFX Help*, *About Visual Extend*.

Weitere Verbesserungen für Entwickler

Das Formular für die Aktivierung von VFX ist jetzt lokalisiert.

Im VFX – Messagebox Builder und im VFX – Message Editor können jetzt die Texte aller Sprachen bearbeitet werden, unabhängig von den Unicode-Einstellungen des Betriebssystems.

In den VFX – Form Buildern und im VFX – Class Switcher steht jetzt eine größere Auswahl von VFX-Klassen zur Verfügung.

Im VFX-Menü gibt es unter *Project* jetzt eine Möglichkeit vom aktuellen Projekt eine Archivdatei anzulegen. Dateiname ist der Projektname gefolgt von einem Timestamp.

In der Tabelle *Vfxfopen.dbf* gibt es jetzt ein Feld *Iconfile*. In diesem Feld kann ein Icon zu dem aktuellen Formular gespeichert werden. Dieses Icon wird im XP-Öffnen-Dialog angezeigt.

Das Debug-Menü für die Entwicklungsumgebung kann jetzt im VFX – Application Builder eingeschaltet werden. Wahlweise kann manuell die Eigenschaft *IDebugMode* der Klasse *cFoxAppl* in der Klassenbibliothek *Appl.vcx* auf *.T.* eingestellt werden.

In *Vfxfunc.prg* wurde die neue Funktion *GetNewGUID* hinzugefügt, die einen global und für immer eindeutigen ID-Wert zurückgibt. Die GUID wird mithilfe der API-Funktion *CoCreateGuid* ermittelt. Die Länge eines GUID ist 36 Zeichen. GUIDs können insbesondere dann als ID in Tabellen verwendet werden, wenn ein Datenabgleich mit anderen Tabellen vorgesehen ist.

Bei der Aktualisierung von Child-Daten in OneToMany-Formularen werden jetzt auch die Felder mit dem Benutzernamen (*ins_usr*, *edt_usr*) und dem Timestamp (*ins_date*, *edt_date*) von VFX automatisch mit Daten gefüllt.

In der Klasse *cFormbase* gibt es im Ereignis *Destroy* einen neuen Hook. Damit wird eine Eingriffsmöglichkeit für eigenen Code beim Schließen eines Formulars gegeben.

OnChildRequery

In früheren VFX-Versionen musste Code in die Methode *OnChildRequery* von OneToMany-Formularen eingetragen werden, wenn die Child-Daten auf Ansichten oder CursorAdaptern basierten.

In VFX 9.5 ist dies nicht mehr erforderlich. Beim Bewegen des Satzeigers im Parent-Teil des Formulars werden automatisch alle Child-Arbeitsbereiche überprüft. Wenn ein Child-Arbeitsbereich auf einer Ansicht oder einem CursorAdapter basiert, werden die Daten aktualisiert. Bei Ansichten wird dazu *REFRESH()* aufgerufen. Bei CursorAdaptern wird die Methode *CursorRefresh()* ausgeführt.

Neue Eigenschaften für Endbenutzer

Neue Icons

Viele neue Icons wurden für Endbenutzer erstellt und geben den Anwendungen ein deutlich verbessertes Erscheinungsbild. Auch in die Builder von VFX wurden zahlreiche neue Icons integriert und verbessern die intuitive Bedienung für Entwickler.

Datenzugriff

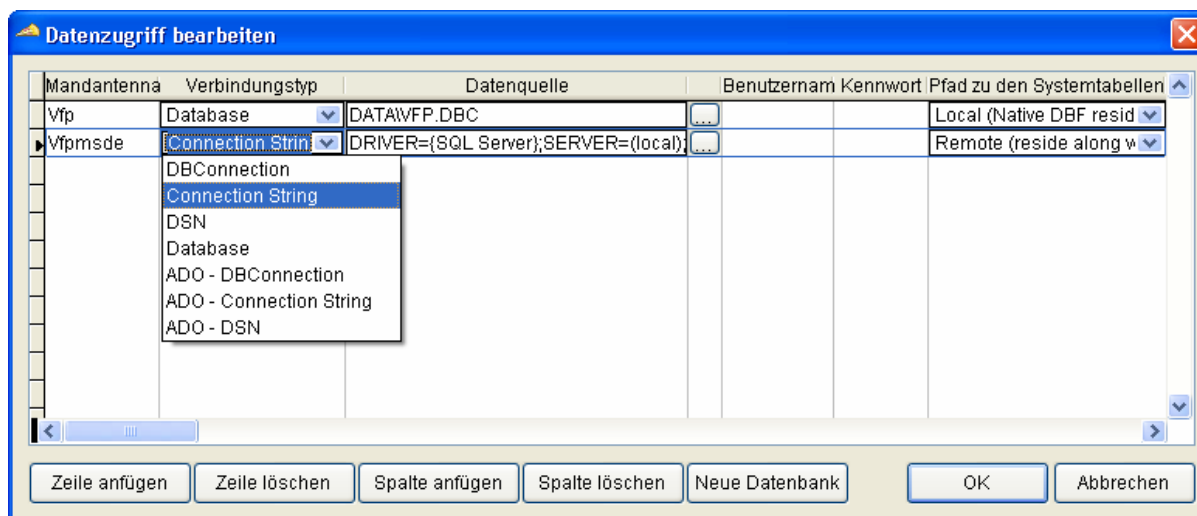
Der Dialog Datenzugriff bearbeiten

Zusätzlich zu den VFX 9.0 bekannten Möglichkeiten des Datenzugriffs kann in VFX 9.5 ein OLE-DB Provider zum Zugriff auf die Daten verwendet werden. Die Funktionalität der Klasse *cConnectionMgr* wurde erweitert um OLE-DB Verbindungen zu ermöglichen.

Im Dialog *Manage Config.vfx* kann zwischen drei OLE-DB Verbindungsmöglichkeiten gewählt werden:

- ADO – DBConnection: Das ist eine Verbindung, die in einem Datenbank-Container gespeichert ist.
- ADO – Connection String: Eine Verbindungszeichenfolge für einen OLE-DB Provider.
- ADO – DSN: Eine auf dem Rechner gespeicherte ADO-DSN-Verbindung.

Diese drei OLE-DB Verbindungstypen entsprechen etwa den drei ODBC-Verbindungstypen DBConnection, Connection String und DSN.



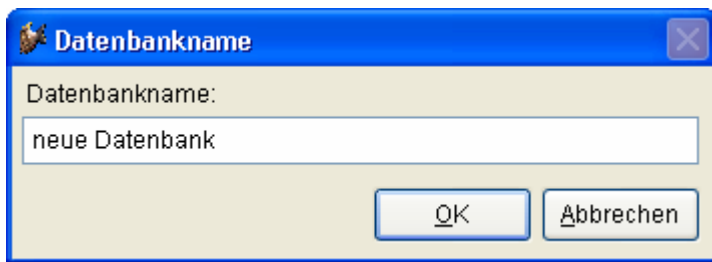
Die VFX-Tabellen können jetzt wahlweise auch in einer eigenen Remote Datenbank gespeichert werden und stehen so auf Wunsch Datenbank-Übergreifend zur Verfügung. Alle Verbindungstypen stehen auch für die Datenbank mit den VFX-Tabellen zur Verfügung.

Anlegen einer neuen Datenbank beim Kunden

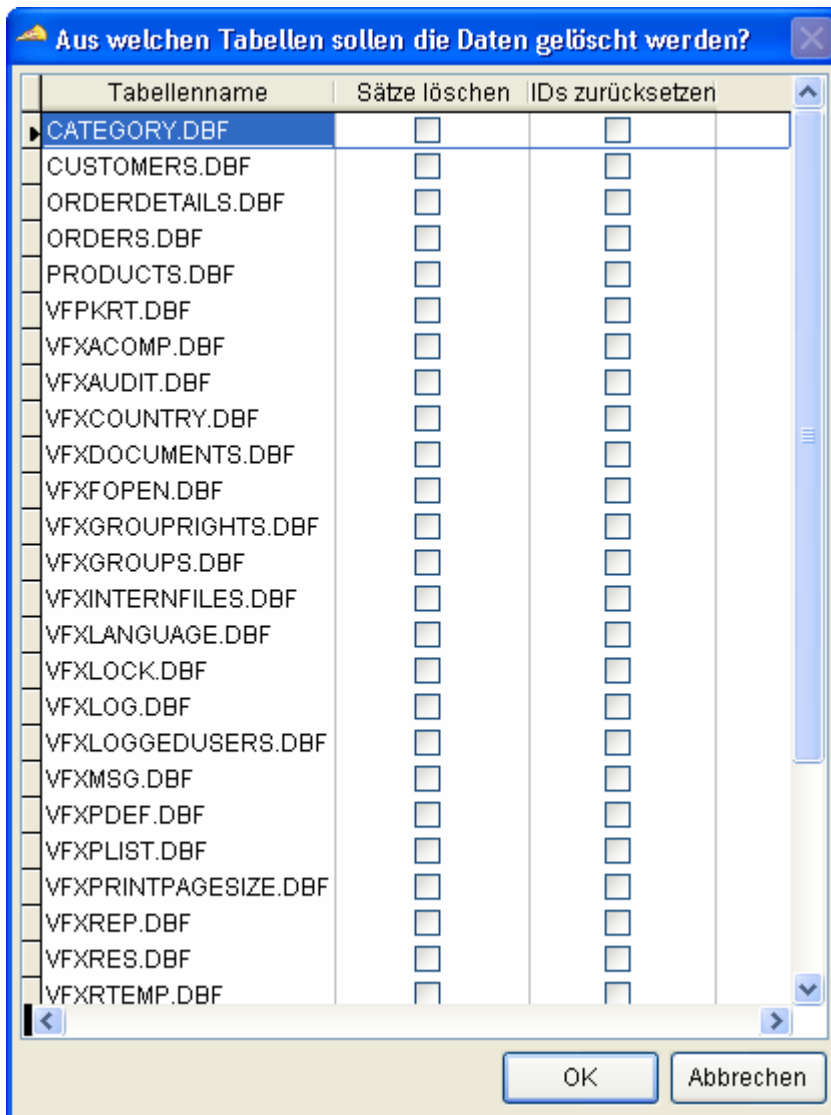
In VFX 9.5-Anwendungen kann auf einfachem Weg eine neue Datenbank angelegt werden. Zur Laufzeit ist im Dialog *Datenzugriff bearbeiten* die Schaltfläche *Neue Datenbank* sichtbar. Die neue Datenbank wird mit der Struktur der aktuell selektierten Datenbank angelegt.

Wenn die aktuelle Datenbank eine VFP-Datenbank ist, erscheint ein Ordnerauswahldialog und der Benutzer kann einen neuen Ordner für die neue Datenbank anlegen.

Wenn die aktuelle Datenbank eine Remote Datenbank ist, kann der Benutzer der neuen Datenbank in einem Dialog einen Namen geben.



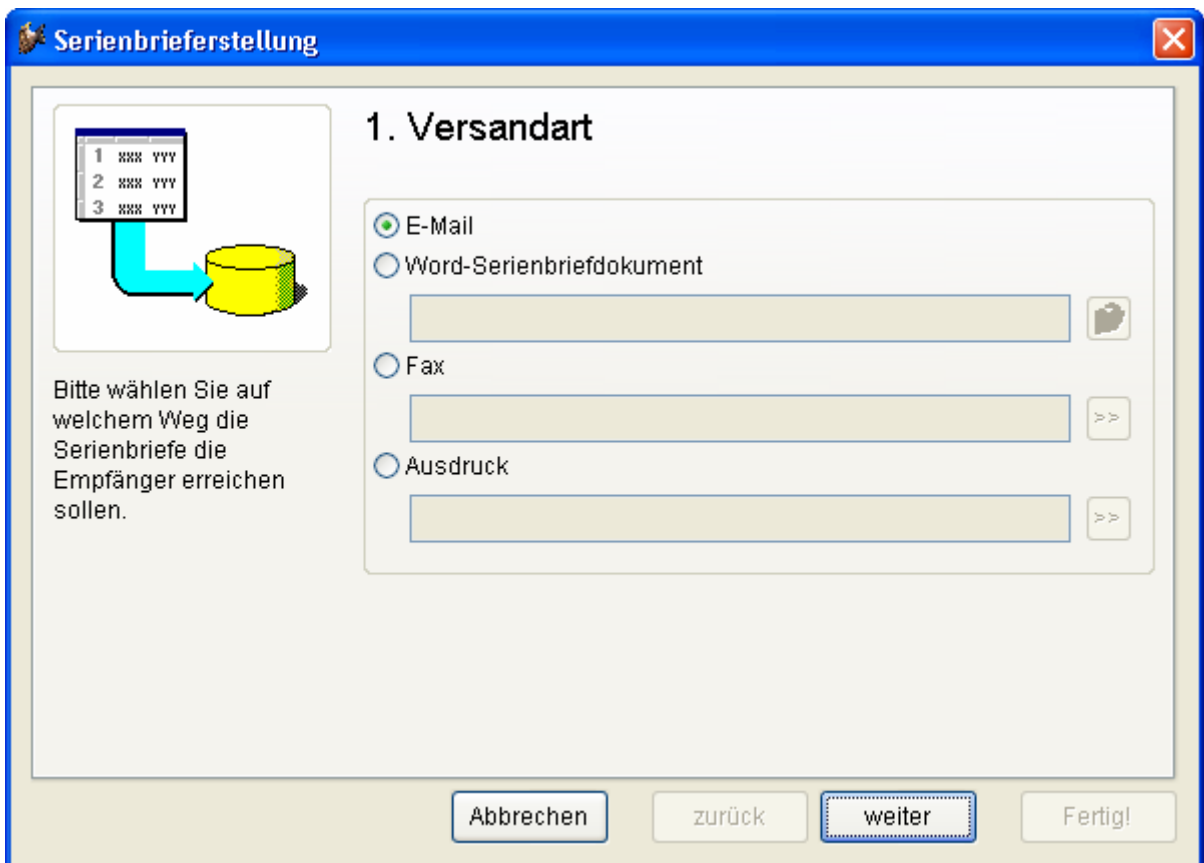
Im nächsten Dialog wird gefragt, aus welchen Tabellen die Daten in der neu angelegten Datenbank gelöscht werden sollen. Auf Wunsch können auch die ID-Werte für einzelne Tabellen zurückgesetzt werden.



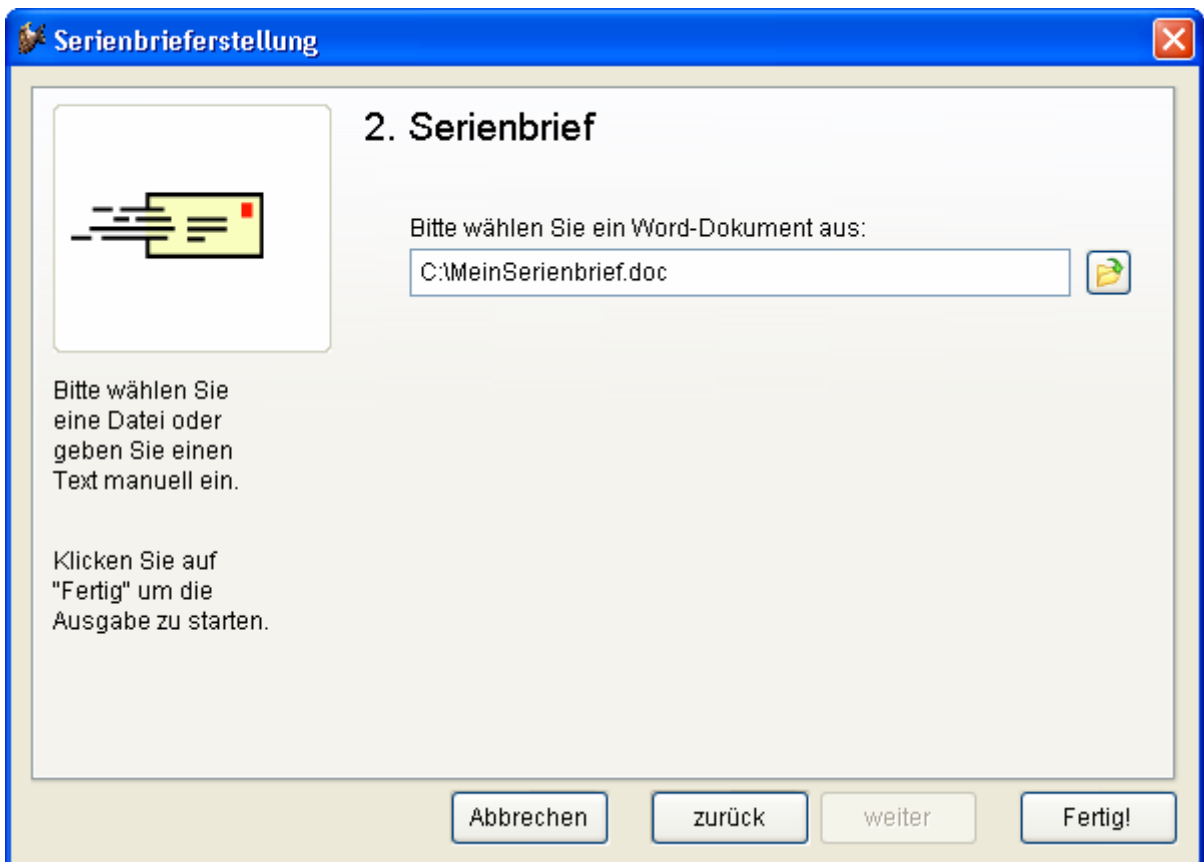
Seriendokumenterstellung

Mit dem Assistenten zur Seriendokumenterstellung kann dem Benutzer die Möglichkeit gegeben werden Seriendokumente basierend auf den Daten der Anwendung zu erstellen. Als Text für die generierten Dokumente kann ein Word-Serienbriefdokument oder eine Textdatei verwendet werden oder es kann manuell im Assistenten ein Text eingegeben werden. Das Ergebnis der Seriendokumentaussgabe kann wahlweise als Word-Dokument gespeichert werden, gedruckt werden, als Fax gesendet werden oder als E-Mail gesendet werden. Der Benutzer wird durch den Assistenten in wenigen intuitiven Schritten geführt.

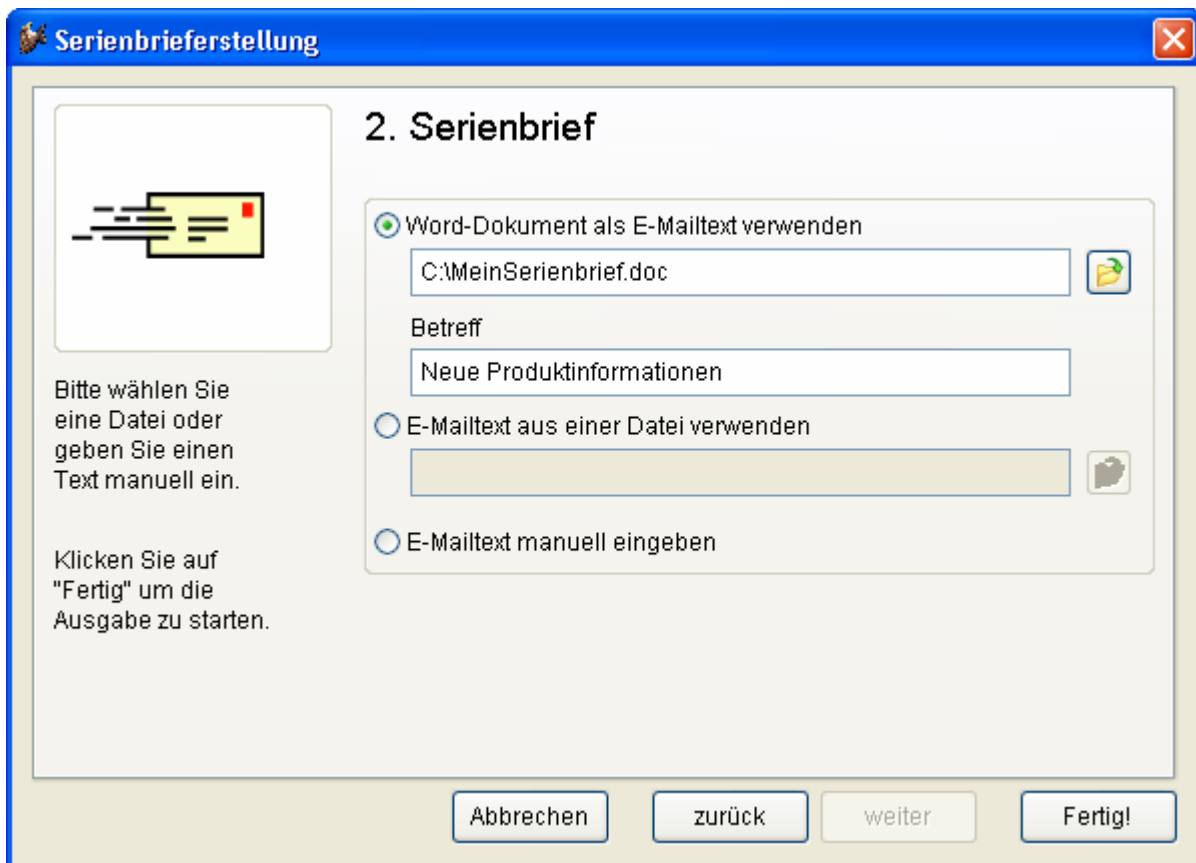
Im ersten Schritt wählt der Benutzer die Versandart.



Im zweiten Schritt wird der Text für die zu erstellenden Dokumente ausgewählt.



Wenn im ersten Schritt Dokument, Fax oder Drucker ausgewählt wurde, kann im zweiten Schritt ein Word Serienbriefdokument ausgewählt werden.



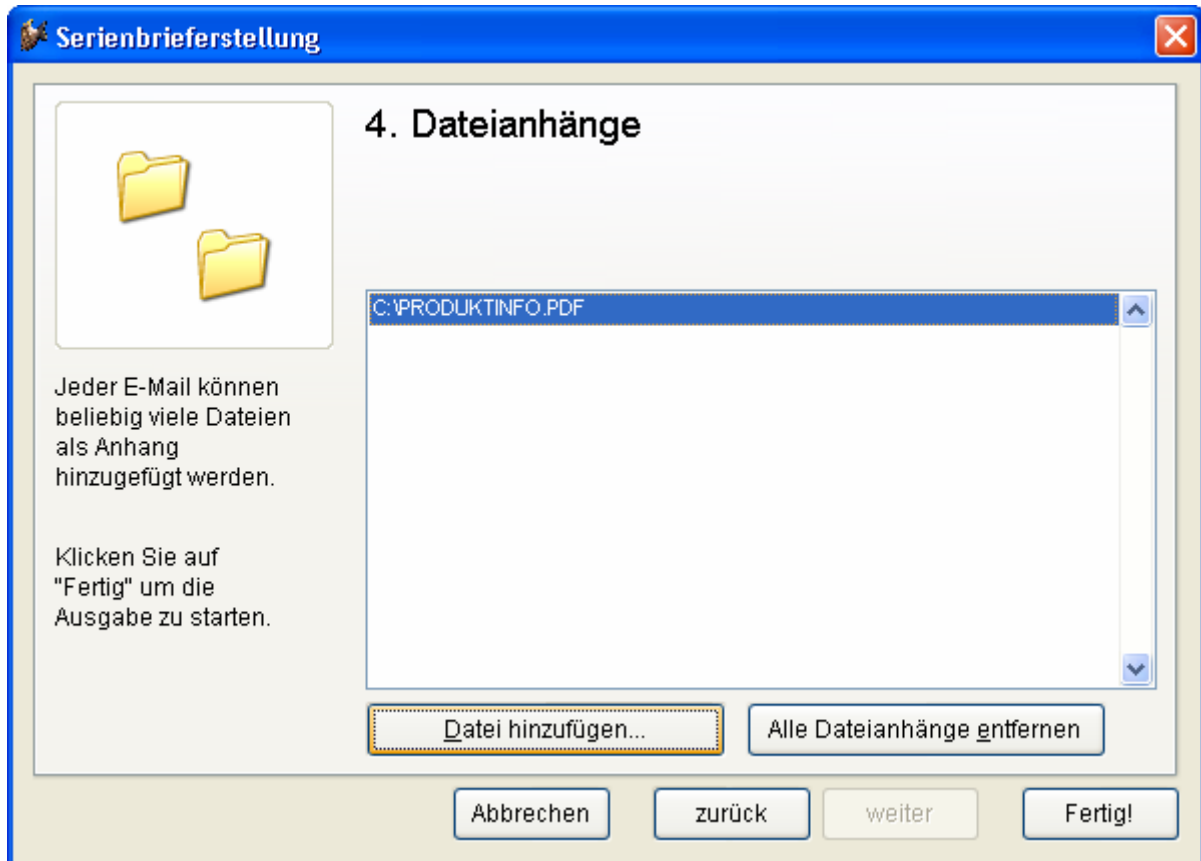
Wenn im ersten Schritt E-Mail ausgewählt wurde, kann der Benutzer im zweiten Schritt zwischen drei möglichen Textquellen wählen.



Wenn im zweiten Schritt ausgewählt wurde, dass ein Text manuell eingegeben werden soll, kann der Text im dritten Schritt erfasst werden.

Wenn im zweiten Schritt ein Textdokument ausgewählt wurde, kann dieser Text im dritten Schritt bei Bedarf geändert werden.

In Serienbriefdokumenten kann, ähnlich wie in Word, ein variabler Text verwendet werden. Dieser variable Text muss in speziellen Zeichen eingeschlossen sein. Standardmäßig sind diese Zeichen doppelte spitze Klammern, also „<<“ und „>>“. Die Begrenzungszeichen können vom Entwickler in den Eigenschaften *cLeftDelim* und *cRightDelim* der Klasse *cMailMerge* eingestellt werden.



Wenn E-Mails versendet werden sollen, können im vierten Schritt Anhänge hinzugefügt werden.

Durch einen Mausklick auf die Schaltfläche *Fertig* werden die Seriendokumente erstellt. Im letzten Schritt wird dem Benutzer die Anzahl der erstellten Seriendokumente angezeigt.

Die Klasse *cMailMerge*

Diese Formalklasse ist in der Klassenbibliothek *Vfxform.vcx* gespeichert.

Mit dieser Klasse können Endanwender anspruchsvolle Seriendokumente erstellen. Folgende Optionen stehen zur Verfügung:

1. E-Mail
Erstellen von Serien-E-Mails. Der E-Mailtext kann aus einem Word-Dokument oder einer Textdatei stammen oder auch manuell in einer Editbox eingegeben werden. Wenn eine Text-E-Mail erstellt wird, können zusätzlich beliebig viele Dateien als Anhang versendet werden.
2. Word-Dokument
Erstellen einer Word-Serienbriefausgabe basierend auf einem Word-Serienbriefdokument. Die Word-Serienbriefausgabe kann dann in Word beliebig weiterbearbeitet werden.
3. Fax
Versenden von Serienfaxen basierend auf einem Word-Serienbriefdokument.
4. Drucken
Drucken von Serienbriefen basierend auf einem Word-Serienbriefdokument.

Zur Seriendokumenterstellung muss ein Cursor vorhanden sein, der die erforderlichen Felder für jede mögliche Benutzerauswahl enthält.

Eigenschaften

cDataSource – Enthält den Namen der Datenquelle für die Seriidokumenterstellung. Diese Datenquelle wird von Word oder vom Assistenten direkt verwendet. Alle variablen Felder müssen in dieser Datenquelle enthalten sein.

cMailAddressFieldName – Enthält den Namen des Feldes für die E-Mailadresse. Dieser Feldname muss in dem in *cDataSource* angegebenen Cursor enthalten sein. Diese Eigenschaft wird nur beim Versand von E-Mails verwendet.

cCCFieldName – Enthält den Feldnamen für eine CC E-Mailadresse. Dieser Feldname muss in dem in *cDataSource* angegebenen Cursor enthalten sein. Diese Eigenschaft wird nur beim Versand von E-Mails verwendet.

cBCCFieldName – Enthält den Feldnamen für eine BCC E-Mailadresse. Dieser Feldname muss in dem in *cDataSource* angegebenen Cursor enthalten sein. Diese Eigenschaft wird nur beim Versand von E-Mails verwendet.

cFaxNumberFieldName – Enthält den Namen des Feldes mit der Faxnummer. Dieser Feldname muss in dem in *cDataSource* angegebenen Cursor enthalten sein. Diese Eigenschaft wird nur beim Faxversand verwendet.

cLeftDelim – Linkes Begrenzungszeichen für die Ersetzung von Text. Standardwert ist „<<“. Die hier angegebene Zeichenkette ist die linke Begrenzung eines variablen Wertes.

cRightDelim – Rechtes Begrenzungszeichen für die Ersetzung von Text. Standardwert ist „>>“. Die hier angegebene Zeichenkette ist die rechte Begrenzung eines variablen Wertes.

cMergeText – Nur zur internen Verwendung. Hier ist der Serientext gespeichert.

nEmailsSent – Nur zur internen Verwendung. Zähler für die Anzahl der zu erstellenden Dokumente. Die Anzahl wird auf der letzten Seite des Wizard angezeigt.

nPreviousPageNum – Nur zur internen Verwendung. Enthält die Nummer der vorhergehenden Seite im Wizard.

Methoden

LoadFileContent – Lädt die im zweiten Schritt angegebene Textdatei. Der Text kann im dritten Schritt in einer Editbox bearbeitet werden.

SendMails – Erstellen der Seriidokumente entsprechend der gewählten Optionen. Von hier wird eine der Methoden *SendThroughMapi* oder *SendThroughOleWord* aufgerufen.

SendThroughMapi – Erstellt Serien-E-Mails unter Verwendung der VFX-Klasse *cEmail*.

SendThroughOleWord – Diese Methode erstellt Serienbriefe per OLE-Automatisierung von Word. Auf diesem Weg kann die Serienbriefausgabe in ein Word-Dokument gespeichert, per Fax gesendet, gedruckt oder per E-Mail versendet werden.

Berichte

In VFX 9.5-Anwendungen ist die Standardeinstellung für Berichtsausgaben *Set Reportbehavior 90*. Damit können alle neuen Eigenschaften des Berichts-Designers und der Berichts-Engine von VFP 9 genutzt werden. Mit der Eigenschaft *nReportBehavior* des Anwendungsobjekts kann auf Wunsch das Reportbehavior auf 80 eingestellt werden. Diese Eigenschaft kann auch im VFX – Application Builder eingestellt werden.

Alle Berichte, die auf Grids basieren und zur Laufzeit erzeugt werden, können jetzt mit mehrzeiligen Detailbändern ausgegeben werden. Mit der Eigenschaft *nMultiLineReport* des Anwendungsobjekts kann das Verhalten eingestellt werden. Ein Wert von 0 bedeutet, dass mit der Eigenschaft *lMultiLineReport* auf Formularebene mehrzeilige Berichte aktiviert werden können. Bei einem Wert von 1 werden mehrzeilige Berichte in allen Formularen aktiviert. Mit dem Wert 2 wird das Verhalten von früheren VFX-Versionen eingestellt. Berichte werden einzeilig, bis zum rechten Papierrand bedruckt.

ReportOutput und ReportPreview

Der Quellcode der VFP-Berichts Anwendungen *ReportOutput.app* und *ReportPreview.app* ist in VFX integriert worden. Selbstverständlich wurde der Code so angepasst, dass sowohl die Lokalisierung zur Entwicklungszeit als auch zur Laufzeit unterstützt werden.

Die Anwendung ReportOutput enthält alle verfügbaren ReportListener. Zusätzlich gibt es einen neuen ReportListener zur Erstellung von PDF-Dateien.

In der Seitenansicht von Berichten steht ein Rechtsklickmenü zur Verfügung. Hierüber kann der Benutzer die Berichtsausgabe drucken, in eine Datei speichern oder als E-Mail versenden.

Für das Drucken von Berichten steht ein erweiterter Druckdialog zur Verfügung. In diesem Dialog können die zu druckenden Seiten gewählt werden. Es ist auch möglich mehrere Seiten eines Berichts auf einer Seite zu drucken. Die Sortierfolge der Seiten kann eingestellt werden.

PDF-ReportListener

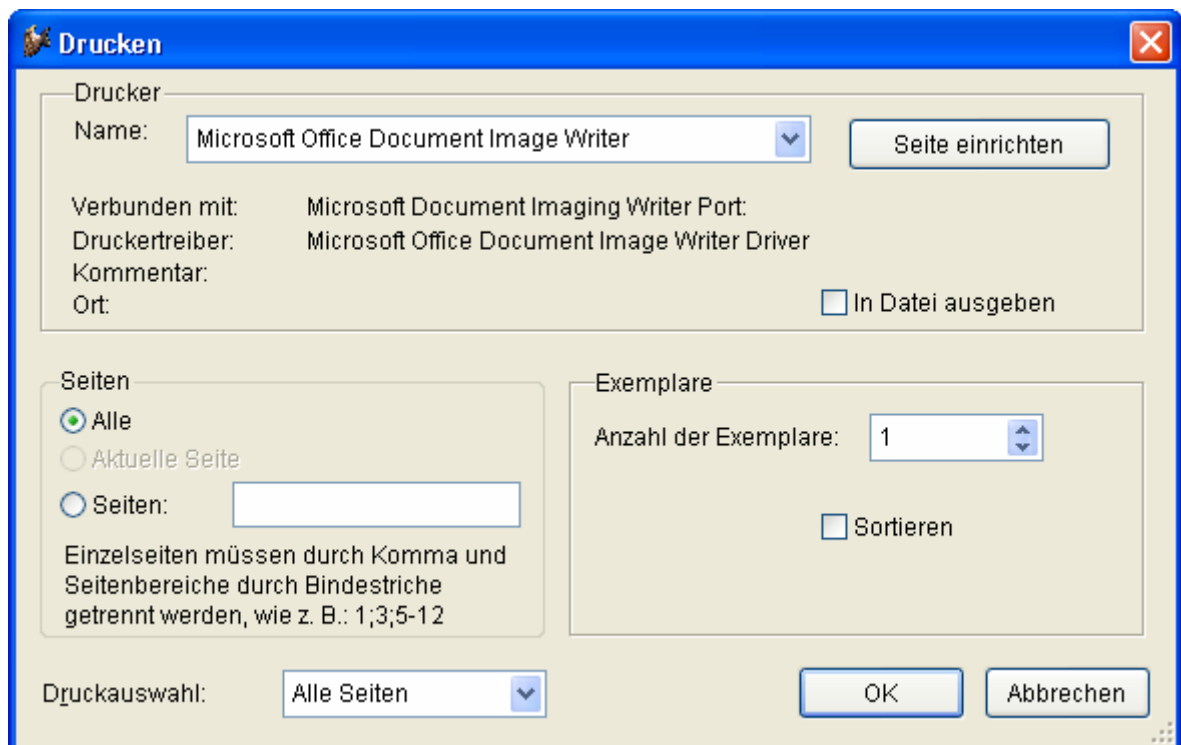
Dieser neue ReportListener wurde zusätzlich zu den bereits in VFX 9.0 vorhandenen ReportListnern hinzugefügt. Mit diesem ReportListener ist die Berichtsausgabe in PDF-Dateien möglich. Die Installation von Ghostscript ist nicht erforderlich.

Erweiterter Druckdialog

In VFX 9.5-Anwendungen kann ein erweiterter Druckdialog verwendet werden, der es den Benutzern erlaubt die Druckausgaben genauer einzustellen. Der neue Druckdialog basiert auf der Klasse *cPrintDialog* und benutzt die neue Klasse *cPrintEngine*.

Die Klasse cPrintDialog

Die Klasse *cPrintDialog* ist in der Klassenbibliothek *Vfxform.vcx* gespeichert. Diese Klasse zeigt den erweiterten Druckdialog an.



In diesem Dialog kann der Drucker ausgewählt werden, die Seiteneinstellungen können verändert werden, die Ausgabe kann in eine Datei umgelenkt werden, die Anzahl der Exemplare kann eingestellt werden und die zu druckenden Seiten können ausgewählt oder eingegeben werden.

Eigenschaften

cPageRange – In dieser Eigenschaft steht die Auswahl der zu druckenden Seiten. Dieser Wert wird nur berücksichtigt, wenn ein manuell eingegebener Seitenbereich gedruckt werden soll (*nPagesSelectionType* = 3).

nAllOddEven - Auswahl zu druckender Seiten.

- 1 – Alle
- 2 – Ungerade Seiten
- 3 – Gerade Seiten

nCollate – Sortierfolge der Seiten.

- 0 – Exemplare werden nacheinander gedruckt.
- 1 – Zu allen Exemplaren wird zunächst die erste Seite gedruckt, dann werden alle zweiten Seiten gedruckt usw.

nNumberOfCopies – Anzahl zu druckender Exemplare.

nPagesSelectionType - Seitenauswahl

- 1 – Alle Seiten
- 2 – Aktuelle Seite
- 3 – Seitenbereich

nPrintToFile – Numerischer Wert mit dem Ausgabeziel.

- 0 – Drucken
- 1 – Ausgabe in eine Datei

oUnderlyingObject – Referenz auf ein Objekt der Klasse *cPrintEngine*.

Printers – Array mit den Informationen aller installierten Druckertreiber.

Die Klasse cPrintEngine

Diese Klasse dient zum drucken von Berichten. Als Parameter wird eine Referenz auf ein ReportListener-Objekt übergeben. Diese Klasse hat Methoden um einen Druckauftrag zu starten, mit Seiten zu füllen und zu schließen. Abhängig vom Wert der Eigenschaft *goProgram.nCustomPrintDialog* kann ein erweiterter Druckdialog angezeigt werden, der es ermöglicht den Drucker, die Druckereigenschaften, die Anzahl zu druckender Exemplare, eine Auswahl von Seiten und die Sortierung der Seiten zu wählen. Der erweiterte Druckdialog ist eine Ableitung von der Klasse *cPrintDialog*.

Eigenschaften

aPagesToPrint – In diesem Array werden die zum Drucken ausgewählten Seiten gespeichert. Das erste Element jeder Zeile enthält die Nummer der Startseite. Das zweite Element jeder Zeile enthält die Nummer der Endseite. Eine einzeln zu druckende Seite wird in beide Elemente einer Zeile eingetragen. Wenn alle Seiten gedruckt werden sollen, enthält dieses Array nur eine Zeile.

cPageRange - In dieser Eigenschaft steht die Auswahl der zu druckenden Seiten. Dieser Wert wird nur berücksichtigt, wenn ein manuell eingegebener Seitenbereich gedruckt werden soll (*nPagesSelectionType* = 3).

cPrinterDriver - Name des aktuell ausgewählten Druckertreibers.

cPrinterName - Name des aktuell ausgewählten Druckers.

cPrintFileName - Name der Datei, wenn die Ausgabe in eine Datei umgelenkt werden soll.

lCustomPrintDialog – Soll der erweiterte Druckdialog angezeigt werden?

- .T. – Anzeige des erweiterten Druckdialogs.
- .F. – Anzeige des Standarddruckdialogs.

nAOE – Auswahl der zu druckenden Seiten.

- 1 – Alle Seiten
- 2 – Nur ungerade Seiten
- 3 – Nur gerade Seiten

nHeight – Höhe des bedruckbaren Bereichs des aktuellen Druckers.

nLeft – Linker Rand des bedruckbaren Bereichs des aktuellen Druckers.

nPagesSelectionType – Seitenauswahl.

- 1 – Alle Seiten
- 2 – Aktuelle Seite
- 3 – Seitenbereich

nPrinterHandle – Handle des verwendeten Druckertreibers.

nPrintToFile – Numerischer Wert mit dem Ausgabeziel.

- 1 – Berichtsausgabe in eine Datei.
- 0 – Drucken.

nShowDialog – Nur zur internen Verwendung. Gibt an, ob der Druckdialog angezeigt wurde.

nTop – Oberer Rand des bedruckbaren Bereichs des aktuellen Druckers.

nWidth – Breite des bedruckbaren Bereichs des aktuellen Druckers.

oRepListener – Referenz zum aufrufenden ReportListener.

Methoden

EndPrinterJob()

Beendet einen Druckauftrag. Bei erfolgreicher Ausführung wird .T. zurückgegeben. Im Fehlerfall wird .F. zurückgegeben.

GetPageProp()

Ermittelt die Eigenschaften Breite, Höhe, oberer Rand und linker Rand des bedruckbaren Bereichs des ausgewählten Druckers und speichert die Werte in den Eigenschaften *nWidth*, *nHeight*, *nTop*, *nLeft*. Bei erfolgreicher Ausführung wird .T. zurückgegeben. Im Fehlerfall wird .F. zurückgegeben.

GetPagesToPrint()

Füllt das Array *aPagesToPrint* entsprechend den Werten der Eigenschaften *nPagesSelectionType* und *cPageRange*. Wenn keine gültigen Seiten ermittelt werden können, wird das Array verworfen und dem Benutzer wird eine Meldung angezeigt. Bei erfolgreicher Ausführung wird .T. zurückgegeben. Im Fehlerfall wird .F. zurückgegeben.

OpenPrintDialog()

Zeigt den erweiterten Druckdialog an. Wenn der Benutzer auf die Schaltfläche *OK* klickt, werden die Methoden *GetPagesToPrint()* und *StartPrinterJob()* aufgerufen. Bei erfolgreicher Ausführung wird .T. zurückgegeben. Im Fehlerfall wird .F. zurückgegeben.

Output_Page(nPageNo)

nPageNo – Gibt die Nummer der zu druckenden Seite an. Es wird die ReportListener-Methode *OutputPage()* verwendet um die Seite an den Druckertreiber zu senden.

Print_Pages()

Liest das Array *aPagesToPrint* und ruft für jede Seite die Methode *Output_Page(nPageNo)* auf. Abschließend wird die Methode *EndPrinterJob()* aufgerufen. Bei erfolgreicher Ausführung wird .T. zurückgegeben. Im Fehlerfall wird .F. zurückgegeben.

StartPrinterJob()

Stellt den im erweiterten Druckdialog selektierten Drucker als aktuellen Drucker ein und startet den Druckauftrag. Rückgabewert ist ein Handle zu dem Druckauftrag.

Datenexport

Im Menü für Endanwender gibt es im Menü unter *Datei* den neuen Menüpunkt *Export als*. Darunter gibt es die Auswahlmöglichkeiten *CSV*, *Excel*, *XML* und *DBF*. Diese Menüpunkte sind aktiviert, wenn ein Formular mit Daten geöffnet und aktiv ist. Die Auswahl einer dieser Optionen öffnet einen *Speichern unter*-Dialog. Nach Eingabe eines Dateinamens werden die Daten aus dem *Initialselectedalias* des Formulars in einer Datei mit dem gewählten Dateiformat gespeichert. Die aktuelle Sortierung sowie ein eventuell gesetzter Filter werden berücksichtigt. Es werden alle Felder exportiert.

Suchdialog

Wenn ein eingegebener Filter zu einer leeren Ergebnismenge führt, wird dies dem Benutzer in einer Messagebox angezeigt und der Suchdialog bleibt geöffnet. Wenn alle Filtereinstellungen gelöscht werden, bleibt der Suchdialog ebenfalls geöffnet. Einzelne Zeilen im Suchdialog können jetzt über eine Schaltfläche gelöscht werden.

Über die Eigenschaft *nFilterBehavior* in der Klasse *cFoxAppl* kann eingestellt werden, ob die neuen Eigenschaften des Suchdialogs von VFX 9.5 genutzt werden sollen oder ob der Suchdialog die gleichen Funktionen wie in VFX 9.0 haben soll. Es ist auch möglich den gewünschten Suchdialog für jedes Formular einzustellen. Hierfür ist *cFoxAppl.nFilterBehavior=0* einzustellen. Die Formulareigenschaft *nFilterBehavior* kann auf 1 eingestellt werden, um das zu VFX 9.0 kompatible Verhalten einzustellen. Mit dem Wert 2 wird der neue Suchdialog aktiviert.

Feld	Operator	Wert	A/a
Date	Gleich	30.09.2005	

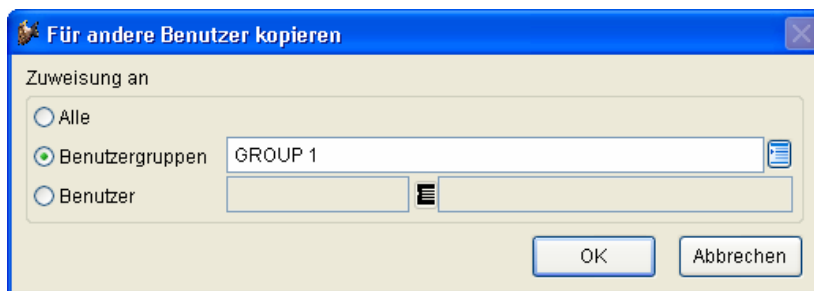
Filtereinstellungen können jetzt je Formular und je Benutzer oder Benutzergruppe gespeichert werden. Der Suchdialog wurde um Steuerelemente zur Verwaltung der Filterdefinitionen erweitert. Filterdefinitionen können für andere Benutzer kopiert werden.

Im neuen Suchdialog stehen die gleichen Filteroptionen wie in VFX 9.0 zur Verfügung. Zusätzlich können Benutzer die Einstellungen speichern und anderen Benutzern oder Benutzergruppen zur Verfügung stellen.

Eine Filtereinstellung kann allen Benutzern, einer Benutzergruppe oder einem Benutzer zugänglich gemacht werden. Jeder Filtereinstellung kann ein Name und eine Beschreibung gegeben werden. Die Filtereinstellungen werden zum aufrufenden Formular gespeichert und können später wieder verwendet werden.

Ein Benutzer kann seine eigenen Filtereinstellungen sehen sowie die Filtereinstellungen, die für alle Benutzer oder für Benutzergruppen freigegeben sind, in denen der Benutzer Mitglied ist. Für jedes Formular kann im Rechtestdialog bzw. in der Verwaltung der Benutzergruppen eingestellt werden, welche Benutzerstufe erforderlich ist, um Filtereinstellungen bearbeiten zu können.

Benutzer können Filtereinstellungen neu anlegen, kopieren, für andere Benutzer kopieren, bearbeiten und löschen. Wenn eine Filtereinstellung für andere Benutzer kopiert werden soll, erscheint ein Dialog zur Auswahl des Benutzers bzw. der Benutzergruppe.



Anpassen-Dialog

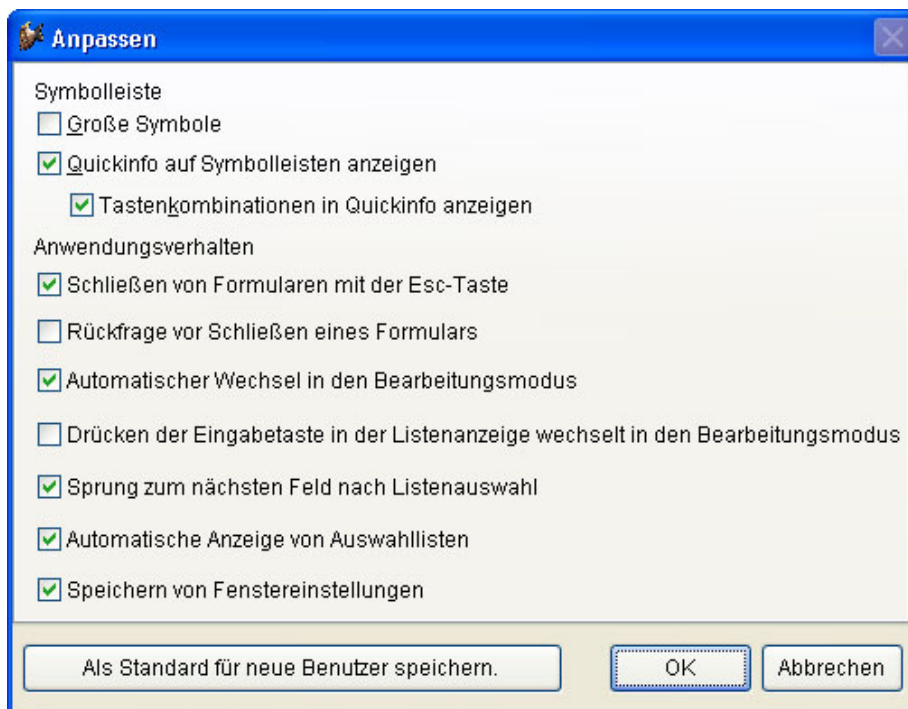
Viele Eigenschaften der Anwendung kann sich jeder Benutzer individuell selbst anpassen. Die Anpassbarkeit dieser Einstellungen kann über die Eigenschaft *AllowUserCustomization* des Anwendungsobjekts für die Anwendung gesteuert werden. Wenn der Wert dieser Eigenschaft *.F.* ist, ist der Anpassen-Dialog in der Anwendung nicht sichtbar. Wenn der Wert dieser Eigenschaft *.T.* ist, kann der Administrator für jeden Benutzer individuell erlauben den Anpassen-Dialog zu verwenden.

Der Anpassen-Dialog kann aus dem *Optionen*-Menü aufgerufen werden.

Der Anpassen-Dialog kann nur von Benutzern angezeigt werden, die das Recht *Anpassungen je Benutzer ermöglichen* haben. Dieses Recht kann nur von Administratoren vergeben werden, wenn die Eigenschaft *goProgram.AllowUserCustomization* auf *.T.* eingestellt ist.

Über den Anpassen-Dialog kann jeder Benutzer seine individuellen Einstellungen zu Symbolleisten und zum Anwendungsverhalten machen.

Die Schaltfläche *Als Standard für neue Benutzer speichern* ist nur für Administratoren sichtbar. Wenn ein Administrator auf diese Schaltfläche klickt, werden die aktuell sichtbaren Einstellungen als Standardwerte für neu anzulegende Benutzer gespeichert. Wenn sich ein neuer Benutzer erstmalig anmeldet, gelten diese Standardwerte.



Mit der Schaltfläche *OK* werden die aktuell sichtbaren Einstellungen für den angemeldeten Benutzer übernommen. Mit der Schaltfläche *Abbrechen* werden die Einstellungen verworfen.

Die Klasse CArchive

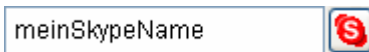
Dieser Klasse wurde die neue Eigenschaft *IQuietMode* hinzugefügt. Über diese Eigenschaft kann gesteuert werden, ob alle Anzeigen während eines Archivierungsvorgangs unterdrückt werden sollen.

IQuietMode – Gibt an, ob die Archivierung ohne Meldungen ausgeführt werden soll. Diese Eigenschaft muss eingestellt werden, bevor eine der Methoden *CreateArchive()* oder *ExtractFromArchive()* ausgeführt wird.

- .T. – Meldungen unterdrücken (unterdrückt auch die Fortschrittsanzeige)
- .F. – Meldungen anzeigen

Neue Klasse CTextSkype

Die neue Klasse *CTextSkype* befindet sich in der Klassenbibliothek *Vfxctrl.vcx*. Diese Klasse besteht aus einem Container mit einer Textbox und einer Schaltfläche.



Wenn der Benutzer zur Laufzeit auf die Schaltfläche klickt, wird der in der Textbox befindliche Wert als Skype-Name an das Programm Skype übergeben. Mit Skype ist es möglich Telefongespräche über das Internet zu führen und Sofortnachrichten zu senden. Mehr Informationen zu Skype finden Sie im Internet auf <http://www.skype.com>.

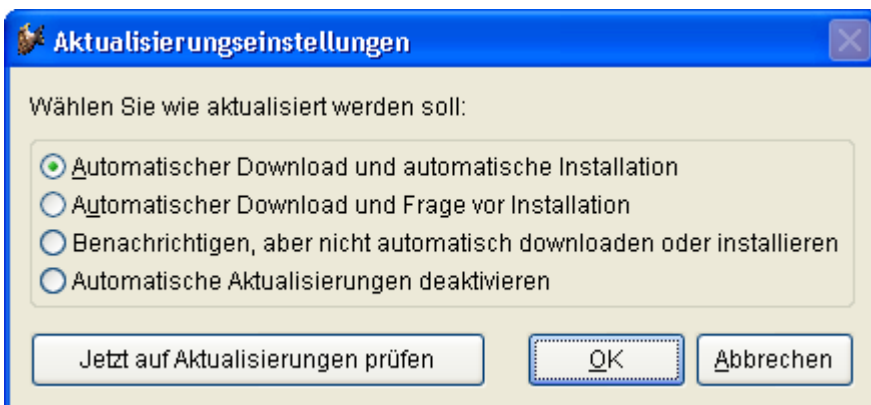
Behandlung von Laufzeitfehlern

Wenn ein Laufzeitfehler auftritt, prüft die Funktion *OnError* ob eine Verbindung zur Remote Datenbank besteht. Wenn möglich, werden die Fehlerinformationen in der Tabelle *Vfxlog* in der Remote Datenbank gespeichert. Wenn eine Verbindung zur Remote Datenbank nicht möglich ist, werden die Fehlerinformationen in der lokalen Tabelle *Vfxlog.dbf* gespeichert.

Wenn eine VFP-Datenbank verwendet wird, werden die Fehlerinformationen immer in der lokalen Tabelle *Vfxlog.dbf* gespeichert.

Aktualisierung der Anwendung

Mit VFX erstellte Anwendungen können über das Internet aktualisiert werden. Wenn die Eigenschaft *AllowUpdates* des Anwendungsobjekts auf .T. eingestellt ist, ist die Aktualisierungsfunktion in der Anwendung aktiviert und die Menüpunkte *Aktualisierung der Anwendung* und *Aktualisierungseinstellungen* stehen zur Verfügung.



Im Dialog *Aktualisierungseinstellungen* können Benutzer zwischen vier Optionen wählen. Wenn automatische Aktualisierungen nicht deaktiviert sind, prüft die Anwendung täglich bei jedem ersten Start ob Aktualisierungen vorliegen.

Die Aktualisierungsfunktionen sind in den Klassen *cUpdate* und *cUpdateEngine* implementiert.

Aktualisierung der Datenbank beim Kunden

Die Aktualisierung der Datenbank beim Kunden wurde erweitert. Das neue Verfahren ist kompatibel zu bisherigen Versionen von VFX. Die Aktualisierung kann also weiterhin dadurch erfolgen, dass eine neue Datenbankstruktur im *Update*-Ordner an den Kunden ausgeliefert wird.

Um das neue Verfahren aus VFX 9.5 zu nutzen, muss die Exe-Datei mit einer Versionsnummer versehen werden. Es empfiehlt sich die Versionsnummer automatisch bei jedem Build von VFP erhöhen zu lassen. Bei jedem Erstellen einer Exe-Datei werden über einen Project Hook für alle Datenbanktypen aus der Datei *Config.vfx* im Projektordner Metadaten erstellt. Die Metadaten werden in die Exe-Datei eingeschlossen und stehen so beim Kunden zur Verfügung.

Wenn eine Exe-Datei gestartet wird, wird die Versionsnummer mit der Nummer verglichen, die im Feld *AppVersion* in der Tabelle *Vfxsys* gespeichert ist. Wenn die Version der aktuellen Exe-Datei größer als die gespeicherte Versionsnummer ist, wird die Aktualisierung der Datenbankstruktur gestartet. Die Aktualisierung wird für jede Datenbank durchgeführt, die in *Config.vfx* oder in *Vfxpath.dbf* eingetragen ist. Es werden dabei VFP-Datenbanken und Remote Datenbanken aktualisiert. Anschließend wird die neue Versionsnummer im Feld *AppVersion* in der Tabelle *Vfxsys* gespeichert.

Weitere Verbesserungen für Endbenutzer

Die Archivierungsfunktion aus dem Menü erstellt jetzt Dateinamen, die aus dem Ordernamen, dem Datenbanknamen sowie dem aktuellen Datum im ANSI-Format bestehen.

Alle benutzerspezifischen Einstellungen, wie Formularegröße, Position auf dem Bildschirm und Grid-Einstellungen können jetzt wahlweise für jede verwendete Bildschirmauflösung separat gespeichert werden. Dafür ist die Eigenschaft *ISaveFormLayoutResolutionDependent* in der Klasse *cFoxAppl* in der Klassenbibliothek *Appl.vcx* auf *.T.* zu stellen. Der Standardwert ist *.F.* Diese Einstellung kann auch mit dem VFX – Application Builder gemacht werden.

Die Daten aus Child-Grids auf OneToMany-Formularen können jetzt per Drag & Drop in andere Anwendungen kopiert werden. Diese Option kann im VFX – COneToMany Builder und im VFX – CChildgrid Builder ein- bzw. ausgeschaltet werden.

Zu jeder Spalte in einem Child-Grid auf einem OneToMany-Formular kann eine Summe gebildet werden. Diese Option kann im VFX – COneToMany Builder und im VFX – CChildgrid Builder ein- bzw. ausgeschaltet werden. Wenn eine Summe gebildet werden soll, wird am unteren Formularrand ein Label mit der Bezeichnung der Spalte sowie eine Textbox mit der Summe hinzugefügt.

In VFX 9.0 konnten die Daten aus allen Grids per Drag & Drop in andere Anwendungen gezogen werden. In VFX 9.5 ist dieses Verhalten global und je Grid einstellbar. Wenn die Eigenschaft *nOLEDragGrid* der Klasse *cFoxAppl* in der Klassenbibliothek *Appl.vcx* auf 1 eingestellt wird, können die Daten aus allen Grids der Anwendung per Drag & Drop in andere Anwendungen gezogen werden. Dies war das Standardverhalten in VFX 9.0. Wenn der Wert dieser Eigenschaft auf 0 eingestellt wird, kann dies für jedes Grid individuell mit der Eigenschaft *IOLEDragGrid* eingestellt werden. 1 ist der Standardwert. Wenn der Wert der Eigenschaft *nOLEDragGrid* auf 2 eingestellt wird, ist OLE Drag & Drop in allen Grids der Anwendung ausgeschaltet.

Bereits in VFX 9.0 konnte mit der Eigenschaft *IKeepLoggedUsers* des Anwendungsobjekts eingestellt werden, ob Benutzeranmeldungen protokolliert werden sollen. Für eine Anwendung konnte mit dem VFX – Application Builder eingestellt werden, ob sich Benutzer mehrmals gleichzeitig anmelden dürfen. In VFX 9.5 kann der Administrator für jeden Benutzer individuell einstellen, ob eine mehrfache Anmeldung erlaubt ist. Zu diesem Zweck gibt es in der Benutzerverwaltung das Kontrollkästchen *Mehrfache Anmeldung erlauben*.

Das Menü *Hilfe* in Anwendungen wurde erweitert. Über den Eintrag *Besuchen Sie unsere Website* kann der Benutzer die Website besuchen, deren URL in der Eigenschaft *cCompanyWebSiteURL* des Anwendungsobjekts hinterlegt ist. Über den Menüpunkt *So erreichen Sie uns* können dem Benutzer Kontaktinformationen angezeigt werden. Als Kontaktinformation dient der Text, der in der Tabelle *Vfxinternfiles.dbf* im Datensatz mit *type="contactus"* gespeichert ist.

Favoriten erscheinen jetzt auch im XP-Öffnen-Dialog in eigenen Gruppen je Formular.

Anhang

Die folgenden User-Defined Transact-SQL Zeichenfolgenfunktionen wurden freundlicherweise von Igor Nikiforov zur Verfügung gestellt und werden mit VFX geliefert.

AT()

Gibt die numerische Anfangsposition zurück, an der ein Zeichenausdruck zum ersten Mal in einem anderen Zeichenausdruck vorkommt, und zwar vom äußersten linken Zeichen aus gerechnet.

Syntax

AT(@cSearchExpression, @cExpressionSearched [, @nOccurrence])

Parameter

- @cSearchExpression - Gibt den Zeichenausdruck an, nach dem AT() in @cExpressionSearched sucht.
- @cExpressionSearched - Gibt den Zeichenausdruck an, in dem mit @cSearchExpression gesucht wird. Sowohl @cSearchExpression als auch @cExpressionSearched können von beliebiger Größe sein.
- @nOccurrence - Gibt an, nach dem wie vielen Vorkommen (ersten, zweiten, dritten usw.) von @cSearchExpression in @cExpressionSearched gesucht werden soll. Standardmäßig sucht AT() nach dem ersten Vorkommen von @cSearchExpression (@nOccurrence = 1). Durch Angabe von @nOccurrence können Sie weitere Vorkommen von @cSearchExpression in @cExpressionSearched suchen. Wenn @nOccurrence größer ist als die Anzahl der Vorkommen von @cSearchExpression in @cExpressionSearched, gibt AT() den Wert 0 zurück.

Rückgabewert

Smallint

Hinweise

AT() sucht im zweiten Zeichenausdruck nach dem ersten Vorkommen des ersten Zeichenausdrucks. Ist die Suche erfolgreich, gibt AT() eine ganze Zahl zurück, die die Position des ersten Zeichens des gefundenen Zeichenausdrucks angibt. Ist die Suche nicht erfolgreich, gibt AT() den Wert 0 zurück.

Die mit AT() ausgeführte Suche berücksichtigt Groß- und Kleinschreibung. Wenn Sie einen Suchvorgang ausführen möchten, bei dem die Groß-/Kleinschreibung nicht berücksichtigt wird, verwenden Sie die ATC()-Funktion.

Ähnlich zu der bekannten Oracle-Funktion INSTR.

Siehe auch RAT().

Beispiel

```
declare @gcString nvarchar(4000), @gcFindString nvarchar(4000)
select @gcString = N'Johann Wolfgang von Goethe (1749-1832)', @gcFindString = 'von'
select dbo.AT(@gcFindString, @gcString, default) -- Anzeige 17
set @gcFindString = 'VON'
select dbo.AT(@gcFindString, @gcString, default) -- Anzeige 0, case-sensitive
```

ATC()

Gibt die numerische Anfangsposition des ersten Auftretens eines Zeichenausdrucks innerhalb eines anderen Zeichenausdrucks zurück, ohne die Groß-/Kleinschreibung dieser beiden Ausdrücke zu berücksichtigen.

Syntax

ATC(@cSearchExpression, @cExpressionSearched [, @nOccurrence])

Parameter

- @cSearchExpression** - Gibt den Zeichenausdruck an, nach dem ATC() in @cExpressionSearched sucht. Der Ausdruck kann von beliebiger Größe sein.
- @cExpressionSearched** - Gibt den Zeichenausdruck an, in dem mit @cSearchExpression gesucht wird. Der Ausdruck kann von beliebiger Größe sein.
- @nOccurrence** - Gibt an, nach dem wie vielen Vorkommen (ersten, zweiten, dritten usw.) von @cSearchExpression in @cExpressionSearched gesucht werden soll. Standardmäßig sucht ATC() nach dem ersten Vorkommen von @cSearchExpression (@nOccurrence = 1). Durch Angabe von @nOccurrence können Sie weitere Vorkommen von @cSearchExpression in @cExpressionSearched suchen.

Rückgabewert

Smallint

Hinweise

ATC() sucht im zweiten Zeichenausdruck nach dem ersten Zeichenausdruck, ohne dabei für die beiden Ausdrücke die Groß-/Kleinschreibung (Groß- oder Kleinbuchstaben) zu berücksichtigen. Soll bei einem Suchvorgang die Groß-/Kleinschreibung berücksichtigt werden, verwenden Sie die AT()-Funktion.

ATC() gibt eine ganze Zahl zurück, die die Position angibt, an der das erste Zeichen des gesuchten Zeichenausdrucks gefunden wurde. Wird der jeweilige Zeichenausdruck nicht gefunden, gibt ATC() den Wert 0 zurück.

Siehe auch AT(), RAT().

Beispiel

```
declare @gcString nvarchar(4000), @gcFindString nvarchar(4000)
select @gcString = N'Johann Wolfgang von Goethe (1749-1832)', @gcFindString = 'VON'
select dbo.ATC(@gcFindString, @gcString, default) -- Anzeige 17, case-insensitive
```

RAT()

Gibt für eine Zeichenfolge die numerische Position zurück, ab der der Ausdruck das letzte Mal (äußerst rechts) in einer anderen Zeichenfolge vorkommt.

Syntax

RAT(@cSearchExpression, @cExpressionSearched [, @nOccurrence])

Parameter

- @cSearchExpression** - Gibt den Zeichenausdruck an, nach dem RAT() in @cExpressionSearched sucht. Der Ausdruck kann von beliebiger Größe sein.
- @cExpressionSearched** - Gibt den Zeichenausdruck an, den RAT() durchsucht. Der Ausdruck kann von beliebiger Größe sein.
- @nOccurrence** - Gibt an, nach welchem Vorkommen (von links nach rechts) von @cSearchExpression RAT() in @cExpressionSearched sucht. Standardmäßig sucht RAT() nach dem letzten Vorkommen von @cSearchExpression (@nOccurrence = 1). Wenn @nOccurrence gleich 2 ist, sucht RAT() nach dem vorletzten Vorkommen usw.

Rückgabewert

Smallint

Hinweise

RAT(), die Umkehrfunktion zu AT(), durchsucht den Zeichenausdruck in @cExpressionSearched von rechts nach links nach dem letzten Auftreten der in @cSearchExpression angegebenen Zeichenfolge.

RAT() gibt eine ganze Zahl zurück, die die Position des ersten Zeichens von @cSearchExpression in @cExpressionSearched angibt. RAT() gibt 0 zurück, wenn @cSearchExpression nicht in

@cExpressionSearched gefunden wird oder wenn @nOccurrence größer ist als die Anzahl des Auftretens von @cSearchExpression in @cExpressionSearched.

Die mit RAT() ausgeführte Suche berücksichtigt Groß- und Kleinschreibung.

Siehe auch AT(), ATC().

Beispiel

```
declare @gcString nvarchar(4000), @gcFindString nvarchar(4000)
select @gcString = N'"Alles Vergängliche / Ist nur ein Gleichnis // Das Unzulängliche, // Hier wirds Ereignis; //
Das Unbeschreibliche, // Hier ist es getan; // Das Ewig- Weibliche // Zieht uns hinan. "' - Faust II, Vers 12104ff,
Chorus mysticus ', @gcFindString = 'Das'
select dbo.RAT(@gcFindString, @gcString, 2) -- Anzeige 94, case-sensitive
```

OCCURS(), OCCURS2()

Gibt den Wert zurück, wie oft ein Zeichenausdruck in einem anderen Zeichenausdruck vorkommt.

Syntax

```
OCCURS(@cSearchExpression, @cExpressionSearched)
OCCURS2(@cSearchExpression, @cExpressionSearched)
```

Parameter

@cSearchExpression - Gibt einen Zeichenausdruck an, den OCCURS() in @cExpressionSearched sucht.

@cExpressionSearched - Gibt den Zeichenausdruck an, in dem OCCURS() nach @cSearchExpression sucht.

Rückgabewert

Smallint

Hinweise

OCCURS() gibt 0 (Null) zurück, wenn @cSearchExpression nicht in @cExpressionSearched gefunden wird.

OCCURS(): einschließlich Deckungen.

```
select dbo.OCCURS('ABCA', 'ABCABCABCA') -- Anzeige 3
```

1 Auftreten 'ABCA .. BCABCA'

2 Auftreten 'ABC...ABCA...BCA'

3 Auftreten 'ABCABC...ABCA'

OCCURS2(): ausschließlich der Deckungen.

```
select dbo.OCCURS2('ABCA', 'ABCABCABCA') -- Anzeige 2
```

1 Auftreten 'ABCA .. BCABCA'

2 Auftreten 'ABCABC... ABCA'

Siehe auch AT(), RAT(), OCCURS2()

Beispiel 1

```
declare @gcString nvarchar(4000)
select @gcString = '"Blut ist ein ganz besonderer Saft."' - Faust I, Vers 1740, Mephistopheles '
select dbo.OCCURS('a', @gcString ) -- Anzeige 3
select dbo.OCCURS('b', @gcString ) -- Anzeige 1
```

Beispiel 2

Zählt das Auftreten verschiedener Buchstaben aus der Zeichenkette @gcCharacters in der Zeichenkette @gcString.

```
declare @gcString nvarchar(4000), @gcCharacters nvarchar(256), @i smallint, @counter smallint
select @i = 1, @counter = 0
```

```

select @gcString = N'Den Teufel spürt das Völkchen nie, und wenn er sie beim Kragen hätte.', @gcCharacters =
N'abccaü'
while @i <= datalength(@gcCharacters)/2
begin
if charindex(substring(@gcCharacters,@i,1), left(@gcCharacters, @i - 1)) = 0
select @counter = @counter + dbo.OCCURS2(substring(@gcCharacters,@i,1), @gcString)
select @i = @i + 1
end
select @counter -- Anzeige 5

```

PADL(), PADR(), PADC()

Gibt aus einem Ausdruck eine Zeichenfolge zurück, die links, rechts oder auf beiden Seiten bis zu einer angegebenen Länge mit Leerzeichen oder Zeichen aufgefüllt ist.

Syntax

```

PADL(@eExpression, @nResultSize [, @cPadCharacter])
PADR(@eExpression, @nResultSize [, @cPadCharacter])
PADC(@eExpression, @nResultSize [, @cPadCharacter])

```

Parameter

- @eExpression - Gibt den aufzufüllenden Ausdruck an. Bei diesem Ausdruck kann es sich um jeden Ausdruckstyp mit Ausnahme eines logischen Ausdrucks bzw. einer Währung, eines Objekt- oder eines Bildfeldes handeln.
- @nResultSize - Gibt die Gesamtzahl der Zeichen im Ausdruck nach dem Auffüllen an.
- @cPadCharacter - Gibt den Wert an, der zum Auffüllen verwendet werden soll. Dieser Wert wird so oft wiederholt, bis der Ausdruck auf die angegebene Anzahl an Zeichen aufgefüllt ist. Wenn Sie @cPadCharacter nicht angeben, werden zum Auffüllen Leerzeichen (ASCII-Zeichen 32) verwendet.

Rückgabewert

Nvarchar(4000)

Hinweise

Mit PADL() wird ein Ausdruck links, mit PADR() rechts und mit PADC() auf beiden Seiten aufgefüllt.

Beispiel

```

declare @gcString nvarchar(4000)
select @gcString = 'Mephistopheles'
select dbo.PADL(@gcString, 40, default) -- Anzeige 'Mephistopheles'
select dbo.PADL(@gcString, 40, '+='+) -- Anzeigen'++++++++=+=+=+=+=+=+=+=+=+=+=Mephistopheles'
select dbo.PADR(@gcString, 40, '!!!=') -- Anzeige 'Mephistopheles=!!!=!!!=!!!=!!!=!!!=!!!='
select dbo.PADC(@gcString, 40, '*=*') -- Anzeige '*====*====*====*====*====*====*====*'

```

CHRTRAN()

Jedes Zeichen in einem Zeichenausdruck, das einem Zeichen in einem zweiten Zeichenausdruck entspricht, wird durch das entsprechende Zeichen eines dritten Zeichenausdrucks ersetzt.

Syntax

```
CHRTRAN(cSearchedExpression, @cSearchExpression, @cReplacementExpression)
```

Parameter

- cSearchedExpression - Gibt den Ausdruck an, in dem CHRTRAN() Zeichen ersetzt.
- @cSearchExpression - Gibt den Ausdruck mit den Zeichen an, nach denen CHRTRAN() in cSearchedExpression sucht.

@cReplacementExpression - Gibt den Ausdruck mit den Ersetzungszeichen an.

Rückgabewert

Nvarchar(4000)

Hinweise

Wird ein Zeichen aus @cSearchExpression in cSearchedExpression gefunden, wird es in cSearchedExpression durch das Zeichen in @cReplacementExpression ersetzt, dessen Position in @cReplacementExpression seiner Position in @cSearchExpression entspricht. Hat @cReplacementExpression weniger Zeichen als @cSearchExpression, werden die übrigen Zeichen aus @cSearchExpression in cSearchedExpression gelöscht. Im umgekehrten Fall werden die überschüssigen Zeichen in @cReplacementExpression ignoriert.

CHRTRAN() übersetzt mit Hilfe der Übersetzungsausdrücke @cSearchExpression und @cReplacementExpression den Zeichenausdruck cSearchedExpression und gibt die sich ergebende Zeichenfolge zurück.

Siehe auch STRFILTER()

Beispiel

```
select dbo.CHRTRAN('ABCDEF', 'ACE', 'XYZ') -- Anzeige 'XBYDZF'  
select dbo.CHRTRAN('ABCDEF', 'ACE', 'XYZQRST') -- Anzeige 'XBYDZF'
```

STRTRAN()

Durchsucht einen Zeichenausdruck nach dem Auftreten eines zweiten Zeichenausdrucks und ersetzt diesen jeweils durch einen dritten Zeichenausdruck.

Syntax

```
STRTRAN(@cSearched, @cExpressionSought [, @cReplacement]  
[, @nStartOccurrence] [, @nNumberOfOccurrences] [, @nFlags])
```

Parameter

@cSearched - Gibt den Zeichenausdruck an, der durchsucht wird.

@cExpressionSought - Gibt den Zeichenausdruck an, nach dem in @cSearched gesucht wird. Bei der Suche wird die Groß- und Kleinschreibung berücksichtigt.

@cReplacement - Gibt den Zeichenausdruck an, der cSearchFor bei jedem Auftreten in @cSearched ersetzt. Wenn Sie @cReplacement nicht angeben, wird @cExpressionSought bei jedem Auftreten durch eine leere Zeichenfolge ersetzt.

@nStartOccurrence - Gibt an, bei welchem Auftreten von @cExpressionSought die Ersetzung beginnen soll. Wenn Sie beispielsweise für @nStartOccurrence den Wert 4 angeben, beginnt das Ersetzen beim vierten Auftreten von @cExpressionSought in @cSearched. Die ersten drei aufgetretenen Ausdrücke werden nicht geändert. Ohne Angabe von @nStartOccurrence beginnt das Ersetzen standardmäßig beim ersten Auftreten von @cExpressionSought.

@nNumberOfOccurrences - Gibt an, wie oft @cExpressionSought ersetzt werden soll. Wenn Sie @nNumberOfOccurrences nicht angeben, wird @cExpressionSought bei jedem Auftreten ersetzt, beginnend mit dem in @nStartOccurrence angegebenen Auftreten.

@nFlags - Gibt an, ob bei der Suche die Groß-/Kleinschreibung berücksichtigt werden soll, und zwar entsprechend den Werten in der folgenden Liste: Wert für @nFlags.

0 (Standardwert) - Beim Suchen wird die Groß-/Kleinschreibung berücksichtigt, das Ersetzen findet mit dem exakten @cReplacement-Text statt.

1 - Beim Suchen wird die Groß-/Kleinschreibung nicht berücksichtigt, das Ersetzen findet mit dem exakten @cReplacement-Text statt.

2 - Beim Suchen wird die Groß-/Kleinschreibung berücksichtigt. Die Groß-/Kleinschreibung beim Parameter @cReplacement wird an die Groß-/Kleinschreibung beim Parameter @cExpressionSought angepasst, der ersetzt wird.

3 - Beim Suchen wird die Groß-/Kleinschreibung nicht berücksichtigt. Die Groß-/Kleinschreibung beim Parameter @cReplacement wird an die Groß-/Kleinschreibung beim Parameter @cExpressionSought angepasst, der ersetzt wird.

Rückgabewert

Nvarchar(4000)

Hinweise

Sie können angeben, wo die Ersetzung beginnen und wie oft diese durchgeführt werden soll. STRTRAN() gibt die Ergebniszeichenfolge zurück. Geben Sie den Wert -1 für optionale Parameter ein, die übersprungen werden sollen. Gleiches gilt, wenn Sie nur die Einstellung für @nFlags angeben wollen.

Siehe auch replace(), CHRTRAN()

Beispiel

```
select dbo.STRTRAN('ABCDEF', 'ABC', 'XYZ',-1,-1,0) -- Anzeige XYZDEF
select dbo.STRTRAN('ABCDEF', 'ABC', default,-1,-1,0) -- Anzeige DEF
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', default,2,-1,0) -- Anzeige ABCDEFGHJabcQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', default,2,-1,1) -- Anzeige ABCDEFGHJQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'XYZ', 2, 1, 1) -- Anzeige
ABCDEFXYZGHJabcQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'XYZ', 2, 3, 1) -- Anzeige
ABCDEFXYZGHJXYZQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'XYZ', 2, 1, 2) -- Anzeige
ABCDEFXYZGHJabcQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'XYZ', 2, 3, 2) -- Anzeige
ABCDEFXYZGHJabcQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'xyZ', 2, 1, 2) -- Anzeige
ABCDEFXYZGHJabcQWE
select dbo.STRTRAN('ABCDEFABCGHJabcQWE', 'ABC', 'xYz', 2, 3, 2) -- Anzeige
ABCDEFXYZGHJabcQWE
select dbo.STRTRAN('ABCDEFabcCGHJabcQWE', 'Aab', 'xyZ', 2, 1, 2) -- Anzeige
ABCDEFabcCGHJabcQWE
select dbo.STRTRAN('abcDEFabcGHJabcQWE', 'abc', 'xYz', 2, 3, 2) -- Anzeige abcDEFxyzGHJxyzQWE
select dbo.STRTRAN('ABCDEFabcCGHJabcQWE', 'Aab', 'xyZ', 2, 1, 3) -- Anzeige
ABCDEFabcCGHJabcQWE
select dbo.STRTRAN('ABCDEFabcGHJabcQWE', 'abc', 'xYz', 1, 3, 3) -- Anzeige XYZDEFxyzGHJxyzQWE
```

STRFILTER()

Entfernt alle Buchstaben aus einer Zeichenkette, ausgenommen den spezifizierten Zeichen.

Syntax

STRFILTER(@cExpressionSearched, @cSearchExpression)

Rückgabewert

Nvarchar(4000)

Parameter

@cExpressionSearched - Spezifiziert die Zeichenfolge, die durchsucht werden soll.

@cSearchExpression - Spezifiziert die Buchstaben, die in @cExpressionSearched erhalten bleiben sollen.

Hinweise

STRFILTER() entfernt alle Buchstaben von @cExpressionSearched, die nicht in @cSearchExpression enthalten sind.

Siehe auch CHRTRAN().

Beispiel

```
select dbo.STRFILTER('asdfghh5hh1jk6f3b7mn8m3m0m6','0123456789') -- Anzeige 516378306
select dbo.STRFILTER('ABCDABCDABCD', 'AB') -- Anzeige ABABAB
```

GETWORDCOUNT()

Zählt die Anzahl der Wörter in einer Zeichenfolge.

Syntax

```
GETWORDCOUNT(@cString[, @cDelimiters])
```

Parameter

@cString - Gibt die Zeichenfolge an, deren Wörter gezählt werden sollen.

@cDelimiters - Gibt ein oder mehrere Zeichen an, durch die Zeichengruppen in @cString getrennt werden sollen. Die Standardtrennzeichen sind Leerzeichen, Tabulator- und Wagenrücklaufzeichen. Beachten Sie, dass GETWORDCOUNT() jedes der Zeichen in @cDelimiters als Trennzeichen verwendet und nicht die ganze Zeichenkette als einzelnes Trennzeichen.

Rückgabewert

Smallint

Hinweise

GETWORDCOUNT() geht standardmäßig davon aus, dass Wörter durch Leerzeichen oder Tabstopps getrennt werden. Wenn Sie als Trennzeichen andere Zeichen angeben, ignoriert diese Funktion Leerzeichen und Tabstopps und verwendet nur die angegebenen Zeichen.

Siehe auch GETWORDNUM(), GETALLWORDS()

Beispiel

```
declare @cString nvarchar(4000)
set @cString = N'Werd ich zum Augenblicke sagen: Verweile doch! Du bist so schön! Dann magst du mich in Fesseln schlagen, dann will ich gern zugrunde gehn!'
-- Wenn Sie als Zielzeichenfolge für GETWORDCOUNT() @cString verwenden, erhalten Sie folgende Ergebnisse:
select dbo.GETWORDCOUNT(@cString, default) -- Anzeige 34 Wörter, getrennt durch Leerzeichen.
select dbo.GETWORDCOUNT(@cString, ',') -- Anzeige 2 Zeichenfolgen abgegrenzt mit ','.
```

GETWORDNUM()

Gibt ein angegebenes Wort aus einer Zeichenfolge zurück.

Syntax

```
GETWORDNUM(@cString, @nIndex[, @cDelimiters])
```

Parameter

@cString - Gibt die Zeichenfolge zurück, die ausgewertet werden soll.

@nIndex - Gibt die Indexposition des zurückzugebenden Worts an. Wenn beispielsweise @nIndex auf 3 gesetzt ist, gibt GETWORDNUM() das dritte Wort zurück (wenn @cString drei oder mehr Wörter enthält).

@cDelimiters - Gibt ein oder mehrere optionale Zeichen an, die verwendet werden, um die Wörter in @cString zu trennen. Die Standardtrennzeichen sind Leerzeichen, Tabulator- und Wagenrücklaufzeichen. Beachten Sie, dass GETWORDCOUNT() jedes der Zeichen in @cDelimiters als Trennzeichen verwendet und nicht die ganze Zeichenkette als einzelnes Trennzeichen.

Rückgabewert

Nvarchar(4000)

Hinweise

Gibt das Wort an der Position zurück, die von @nIndex in der Zielzeichenfolge @cString angegeben wurde. Wenn @cString weniger Wörter als die in @nIndex angegebene Anzahl enthält, gibt GETWORDNUM() eine leere Zeichenfolge zurück.

Siehe auch GETWORDCOUNT(), GETALLWORDS()

Beispiel

```
declare @cString nvarchar(4000)
set @cString = N'Wer immer strebend sich bemüht, Den können wir erlösen.'
select dbo.GETWORDNUM(@cString, 7, default) -- Anzeige 'können'
```

GETALLWORDS()

Fügt die Wörter aus einer Zeichenkette in eine Tabelle ein.

Syntax

GETALLWORDS(@cString[, @cDelimiters])

Parameter

@cString nvarchar(4000) - Spezifiziert die Zeichenkette, deren Wörter in die Tabelle @GETALLWORDS eingesetzt werden.

@cDelimiters - Gibt ein oder mehrere Zeichen an, durch die Zeichengruppen in @cString getrennt werden sollen. Die Standardtrennzeichen sind Leerzeichen, Tabulator- und Wagenrücklaufzeichen. Beachten Sie, dass GETWORDCOUNT() jedes der Zeichen in @cDelimiters als Trennzeichen verwendet und nicht die ganze Zeichenkette als einzelnes Trennzeichen.

Rückgabewert

Tabelle @GETALLWORDS (WORDNUM smallint, WORD nvarchar(4000), STARTOFWORD smallint, LENGTHOFWORD smallint)

Hinweise

GETWORDCOUNT() geht standardmäßig davon aus, dass Wörter durch Leerzeichen oder Tabstopps getrennt werden. Wenn Sie als Trennzeichen andere Zeichen angeben, ignoriert diese Funktion Leerzeichen und Tabstopps und verwendet nur die angegebenen Zeichen.

Siehe auch GETWORDNUM(), GETWORDCOUNT().

Beispiel

```
declare @cString nvarchar(4000)
set @cString = N'Wo fass ich dich, unendliche Natur? Euch Brüste, wo? Ihr Quellen alles Lebens'
select * from dbo.GETALLWORDS(@cString, default)
select * from dbo.GETALLWORDS(@cString, ' ,.?)'
```

PROPER()

Gibt für einen Zeichenausdruck eine Zeichenfolge zurück, deren Wörter kleingeschrieben sind, aber jeweils mit einem Großbuchstaben beginnen.

Syntax

PROPER(@cExpression)

Parameter

@cExpression - Gibt den Zeichenausdruck an, von dem PROPER() eine Zeichenfolge zurückgibt, deren Wörter kleingeschrieben sind, aber jeweils mit einem Großbuchstaben beginnen.

Rückgabewert

Nvarchar(4000)

Hinweise

Siehe auch lower(), upper().

Beispiel

```
select dbo.PROPER(N'JOHANN CARL FRIEDRICH GAUß') -- Anzeige 'Johann Carl Friedrich Gauß'
```

ARABTOROMAN()

Wandelt einen numerischen Ausdruck (von 1 bis 3999) in römische Ziffern um.

Syntax

ARABTOROMAN(@tNum)

Parameter

@tNum Zahl

Rückgabewert

Varchar(15)

Beispiel

```
select dbo.ARABTOROMAN(1777) -- Anzeige MDCCLXXVII
```

ROMANTOARAB()

Wandelt römische Ziffern (von I bis MMMCMXCIX) in einen numerischen Ausdruck um.

Syntax

ROMANTOARAB(@tcRomanNumber)

Parameter

@tcRomanNumber - varchar(15) römische Ziffern.

Rückgabewert

Smallint

Beispiel

```
select dbo.ROMANTOARAB('MDCCCLV') -- Anzeige 1855
```

Mehr als 5000 Entwickler haben bereits meine Funktionen gedownloadet. Ich hoffe, dass sie auch für Sie nützlich sind.

Um mehr Informationen über die Zeichenketten UDFs in Transact-SQL zu erhalten, besuchen Sie bitte:

<http://www.universalthread.com/wconnect/wc.dll?LevelExtreme~2,54,33,27115>