



Session D-VFX

Visual Extend 8.0

Uwe Habermann

Einführung

Visual Extend 8.0 (VFX) ist eine vollständige Entwicklungsumgebung zur Erstellung von Anwendungen mit VFP 8.0. Visual Extend gehört seit vielen Jahren zu den leistungsfähigsten Zusatzprodukten von Visual FoxPro. Mit Visual Extend (im folgenden Text mit VFX abgekürzt) ist es möglich in wenigen Minuten den Rahmen für eine Visual FoxPro-Applikation voll funktionsfähig zu erstellen. Wenn vor der Applikationsentwicklung bereits eine Datenbank oder ein Datenmodell zur Verfügung stehen, ist es ein Leichtes mit den Buildern von VFX innerhalb kürzester Zeit Bearbeitungsformulare zu erstellen.

Lernen wir die wichtigsten Eigenschaften von VFX kennen in dem wir die Arbeitsschritte zur Erstellung einer Applikation durchgehen. Der Entwickler wird durch zahlreiche reentrante Builder bei der Arbeit unterstützt. Der Kunde erhält Applikationen, die in der Bedienung ähnlich zu den bekannten Office-Anwendungen sind. VFX 8.0 ist dabei sowohl für Einzelplatzlösungen als auch für den Fileserver und den Client/Server-Bereich geeignet. Diese Vendor-Session soll dem Einsteiger in VFX vermitteln wie innerhalb kürzester Zeit eine kleine Anwendung mit VFX komplett lauffähig erstellt werden kann.

Überblick

Visual Extend stellt eine umfassende Entwicklungsumgebung für Softwareentwickler dar, die mit Microsoft Visual FoxPro 8.0 oder einer neueren Version arbeiten. Visual Extend beinhaltet Builder, die den Softwareentwickler bei seiner täglichen Arbeit unterstützen und so die Entwicklerproduktivität drastisch steigern. Dies, ohne jegliche Einbußen bezüglich Flexibilität oder Leistungsfähigkeit in Kauf nehmen zu müssen. Visual Extend macht aus Visual FoxPro ein echtes Rapid Application Development Tool, dies sowohl für Desktop- als auch für Client/Server-Datenbank-Applikationsentwicklungen.

Visual FoxPro ist ein exzellentes Entwicklungswerkzeug. Dank der Objektorientierung und der OLE-Technologie wird der Traum eines jeden Softwareentwicklers nach einfachster Wiederverwendung von eigenen oder fremden Softwaremodulen Wirklichkeit. Das Erstellen einer eigenen Entwicklungsumgebung stellt jedoch ein größeres Unterfangen dar, welches sich heutzutage immer weniger Softwareentwickler wirklich leisten können. Es ist nicht nur schwierig, eine stabile Klassenbibliothek für alle Anwendungen zu entwickeln, es wäre auch sehr zeitaufwendig, die Klassen manuell einzusetzen und alle Eigenschaften und Methoden über das Eigenschaftsfenster während der Entwicklung einer neuen Anwendung zu bearbeiten.

Visual Extend für Visual FoxPro füllt exakt diese Lücke und stellt eine vollständige Anwendungsentwicklungsumgebung für Visual FoxPro Softwareentwickler dar. Dank des durchdachten, modularen Designs von Visual Extend kann der Softwareentwickler jederzeit selbst entscheiden, ob er die gesamte Entwicklungsphilosophie von Visual Extend verwenden oder nur ausgewählte Teile daraus für die Erstellung seiner eigenen Anwendungen übernehmen will. Die Objektorientierung von Visual Extend erlaubt dem Entwickler Unter-

klassen aller Visual Extend Klassen zu erstellen, um so die Entwicklungsumgebung noch besser seinen speziellen Bedürfnissen anzupassen.

Visual Extend ist weit mehr als nur eine Klassenbibliothek. Vielmehr beinhaltet Visual Extend neben einer leistungsfähigen Klassenbibliothek ebenso leistungsfähige Builder, um einen maximalen Produktivitätsgewinn zu erzielen. Visual Extend besteht aus den folgenden Hauptkomponenten:

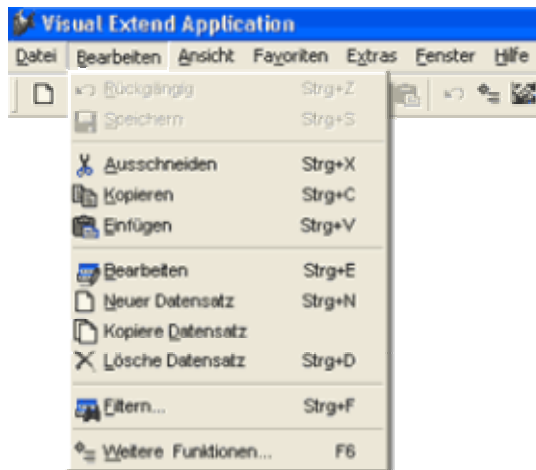
- Modulare den Microsoft Standards entsprechende Klassenbibliothek zur umfassenden Unterstützung bei der Anwendungsentwicklung.
- Visual Extend Assistenten und voll wieder verwendbare Builder für Anwendung, Formular, Grid, Child-Grid, Auswahlliste, Auswahltextfeld und 1:n-Formulare.
- Weitere Visual Extend Entwickler-Produktivitätswerkzeuge wie das Entwicklermenü, die VFX Task Pane, der VFX – Base Class Switcher und der Visual Object Name Picker.

Leistungsmerkmale

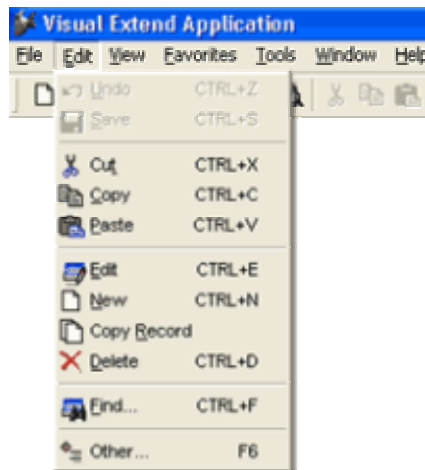
Softwareentwickler werden die folgenden Visual Extend-Merkmale besonders zu schätzen wissen:

- Anwendungs-Assistent für das automatische Erstellen von neuen Anwendungen in der Sprache Ihrer Wahl. Nach nur wenigen Sekunden ist Ihre lauffähige Visual FoxPro Anwendung vorbereitet!
- Volle Wiederverwendbarkeit von allen VFX-Buildern (Formular-Builder, 1:n-Formular-Builder, Table Form-Builder, Grid-Builder, Child-Grid-Builder, Auswahltextbox-Builder), die es vereinfachen, Änderungen an mit den VFX-Buildern erstellten Formularen durchzuführen!
- Benutzen Sie die Visual FoxPro-Entwicklungsumgebung wann immer Sie wollen, ohne die Wiederverwendbarkeit der VFX-Builder zu verlieren, solange Sie alle Steuerelemente mit Hilfe der VFX-Builder hinzufügen bzw. entfernen!
- Builder für Standardformulare inklusive Parent/Child-Technik (aufrufen und auferufen von).
- Builder für leistungsfähige Grids.
- Builder für jeden Bedarf an Auswahllisten.
- Builder für klassische sowie fortgeschrittene 1:n-Formulare mit mehrseitiger Bearbeitung der Haupttabelle sowie mehrseitiger Bearbeitung für mehrere Child-Tabellen in einem Formular.
- Alle Builder lesen die vorhandenen Feldbeschreibungen und andere Eigenschaften aus der Datenumgebung.
- Die Formular-Builder passen die Längen der Textfeld-Steuerelemente den Größen der zugrunde liegenden Felder an.
- Die VFX-Formular-Builder sind auf eigenen, von den VFX-Klassen abgeleiteten Klassen einsetzbar.
- Testen von Formularen direkt aus dem Formular-Designer.
- Navigieren mit der Symbolleiste oder mit Navigations-Schaltflächen auf dem Formular oder mit Schaltflächenleisten innerhalb eines Formulars.
- Messagebox-Assistent.
- Task Pane Anwendungs-Manager.
- Einfaches Bilden einer Unterklasse der Anwendungsklasse und Anpassen der Umgebungsklasse.
- Einfaches Erstellen der anwendungsspezifischen Standard-Symbolleisten.
- Technik verbundener Child-Formulare.
- Die Entwicklungsumgebung stellt bereits alle Elemente der Benutzeroberfläche in den Sprachen deutsch, englisch, französisch, spanisch, italienisch, bulgarisch, tschechisch und griechisch zur Verfügung. Starten Sie eine neue Anwendung in der Sprache Ihrer Wahl, ohne ein Wort der Visual Extend Software-Entwicklungsumgebung übersetzen zu müssen.

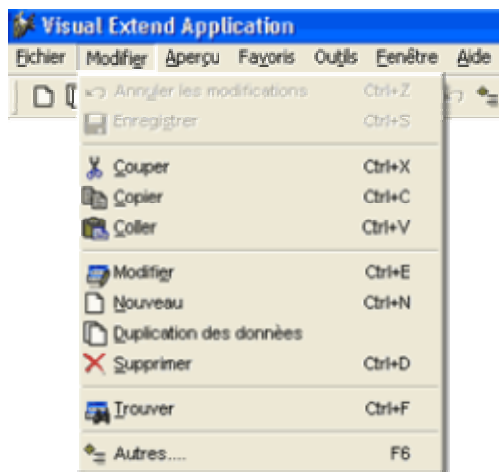
Deutsch



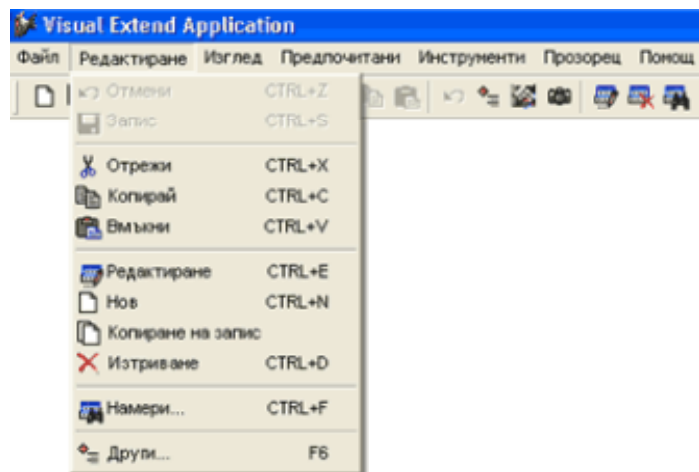
English



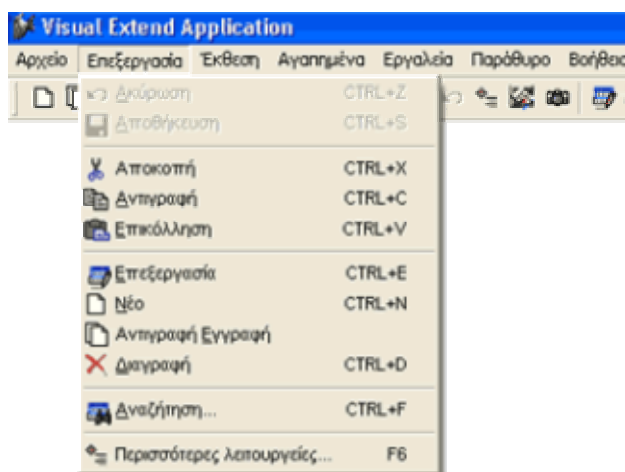
Französisch



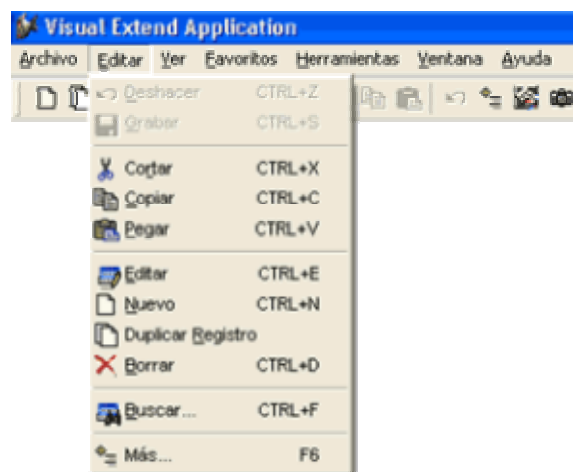
Bulgarisch



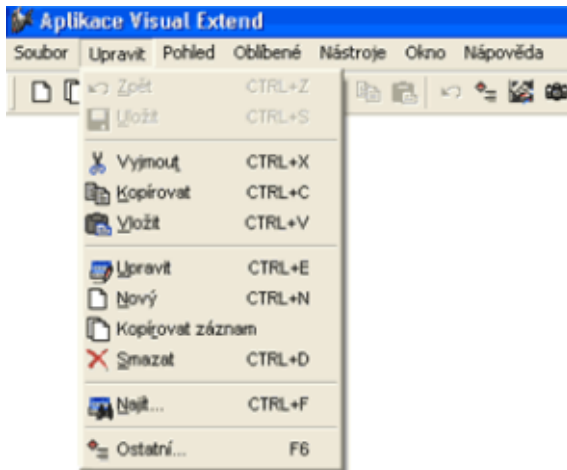
Griechisch



Espanisch



Tschechisch



VFX hilft Ihnen, Ihre Visual FoxPro-Anwendungen in einer höheren Qualität und wesentlich schneller als bisher zu entwickeln. Ihre Entwickler-Produktivität steigert sich dramatisch. Und das alles, ohne irgendwelche Einbußen bezüglich der Flexibilität von Visual FoxPro in Kauf nehmen zu müssen. Produktiver als je zuvor mit Visual Extend für Visual FoxPro!

Eigenschaften von mit Visual Extend erstellten Anwendungen

Anwendungen, die mit Visual FoxPro und der Software-Entwicklungsumgebung Visual Extend entwickelt wurden, haben die folgenden Eigenschaften:

- Bereit zur Office-Compatible-Zertifizierung.
- Standard-Symbolleiste und optionale individuelle Symbolleiste für jedes Formular.
- Unterstützung von XP-Themes in allen Steuerelementen.
- Hot Tracking von Schaltflächen in Symbolleisten.
- Icons in Menüs.
- Navigieren, Suchen, Neu, Kopieren, Bearbeiten, Löschen als Optionen im Formular oder in der Symbolleiste.
- Multiinstanzfähige Formulare.
- Zuletzt aufgerufene Formulare im Menü Datei sowie aktuell geöffnete Formulare im Menü Fenster.
- Inkrementelle Suche inklusive automatischer Sortierung in allen VFX-Grids.
- Wechsel der Sortierung durch Doppelklick auf die Spaltenüberschrift in jedem VFX-Grid.
- Anzeige der aktuellen Sortierung in der Spaltenüberschrift, wahlweise auch farbliche Anzeige.
- Automatisches Speichern und Wiederherstellen der Größe und der Position von allen Formularen.
- Automatisches Speichern und Wiederherstellen aller Layoutänderungen und der aktuellen Sortierfolge im Grid.
- Auswahllisten-Steuerelement mit automatischer Validierung.
- Auswahllisten-Formular mit inkrementeller Suche, automatischer Sortierung, Wechsel der Sortierung durch Doppelklick auf eine Spaltenüberschrift und Start des Bearbeitungsformulars mit der Möglichkeit neue Datensätze einzugeben.
- Automatisches Speichern und Wiederherstellen der Größe und Position von allen Auswahllisten-Formularen inklusive aller Layoutänderungen im Auswahllisten-Grid.
- Leistungsfähige Auswahllisten in Child-Grids.
- Benutzerverwaltung mit Kennwort-Verschlüsselung.
- Automatische Übernahme des Netzwerk-Anmeldenamens und Möglichkeit der automatischen Benutzeranmeldung.
- Verwaltung der Benutzerrechte mit Ansichts-, Bearbeitungs-, Neuanlage- und Löschrecht auf Formular-ebene.
- Datenbankwartung für das Komprimieren und neu Indizieren von lokalen Tabellen.
- Automatisches protokollieren aller Laufzeitfehler.
- System Lock-Tabelle für optionale Semaphore Locking Schemata.

- Infodialog.
- Benutzerfreundliche Mover-Dialoge für die einfache Auswahl mehrerer Elemente.
- Automatische Übernahme der Windows-Systemfarben.
- Favoriten-Menü.
- Öffnen-Formular im XP-Stil.
- Optionale „Active Desktop“ Einzelklick-Benutzeroberfläche.
- Automatisches Erstellen von gedruckten Berichten basierend auf der Datenanzeige in einem Grid.
- Berichtsauswahl und Berichtsoptionendialog.
- Unterstützung mehrerer Datenbanken mit der Möglichkeit die Datenbank zur Laufzeit zu wechseln.
- Automatische Aktualisierung der Strukturen der Kundendatenbank für VFP- und SQL Server-Datenbanken.
- Optionales Bearbeitungsprotokoll zur Verfolgung der Datenbearbeitung.
- Die Microsoft Agenten können zur Gestaltung der Benutzeroberfläche verwendet werden.
- Automatischer Ausdruck des Bildschirminhalts.
- Es können mehrsprachige Anwendungen erstellt werden.

Installation

Hardware- und Software-Anforderungen

Da es sich bei Visual Extend um eine Erweiterung zu Microsoft Visual FoxPro 8.0 handelt, benötigen Sie eine Hard- und Softwareumgebung, auf der Visual FoxPro 8.0 eingesetzt werden kann. Sehen Sie bitte bei den Systemanforderungen zu Microsoft Visual FoxPro nach.

Die Installation von VFX

Starten Sie das Installationsprogramm mit dem Namen *VFX80Setup.exe* und folgen Sie den Anweisungen auf dem Bildschirm.

Installieren Sie VFX 8.0 mit dem Installationsprogramm in einen neuen Ordner. Installieren Sie VFX 8.0 nicht in den Ordner, in dem sich eine frühere Version von VFX befindet!

VFX 8.0 hat einen Software-Kopierschutz. Nach der Installation, beim ersten Start eines VFX-Builders, zeigt Ihnen ein Dialog Ihre persönliche Registrierungsnummer. Sie brauchen Ihre Lizenz nur online auf unserer VFX-Registrierungswebseite zu registrieren und wir werden Ihnen den Aktivierungsschlüssel, den Sie in dem Dialog eingeben müssen, per E-Mail senden. Wir bieten zwei verschiedene Typen von Aktivierungsschlüsseln: Einer ist auf eine Laufzeit von 30 Tagen beschränkt, der andere ist unbefristet gültig.

Beachten Sie, dass Sie die Installation von VFX nicht von einem PC auf einen anderen PC kopieren können ohne einen neuen Aktivierungsschlüssel anfordern zu müssen. Ihre Registrierungsnummer wird aus den Daten Ihres PCs ermittelt und ist einmalig. Jeder VFX-Benutzer hat eine andere, einmalige Registrierungsnummer und muss sich daher online registrieren, um den Aktivierungsschlüssel zu bekommen. Erst dann ist die Arbeit mit den VFX-Buildern möglich. Der einzige Weg einen Aktivierungsschlüssel zu erhalten, ist die Registrierung auf unserer Webseite:

<http://www.visualextend.de>

Einstellen der Visual FoxPro Umgebung für VFX

Sie müssen Microsoft Visual FoxPro 8.0 funktionsfähig installiert haben, bevor Sie die Arbeit mit VFX 8.0 beginnen können.

Als nächstes sollten Sie sicherstellen, dass das VFX 8.0-Menü jedes Mal automatisch erscheint, wenn Sie Ihr Visual FoxPro 8.0 starten. Wir schlagen folgenden Weg vor:

Fügen Sie diese Zeile der Datei **CONFIG.FPW** in Ihrem VFP 8.0-Ordner hinzu:

ANMERKUNG: Wenn Sie keine Datei mit dem Namen **CONFIG.FPW** haben, können Sie diese Datei mit dem Editor anlegen.

```
command = do (HOME() + "vfx.prg")
```

Diese Zeile teilt VFP mit, dass das Programm VFP.PRG ausgeführt werden soll, wenn VFP gestartet wird. In der Datei VFX.PRG (erstellen Sie diese Datei ebenfalls mit dem Editor und speichern Sie diese auch im VFP-Ordner) fügen Sie folgende Zeile hinzu:

```
do c:\vfx80\builder\vfxmnu.app
```

Wir gehen hier davon aus, dass VFX im Ordner c:\vfx80 installiert ein. Passen Sie den Pfad ggf. an.

Beim Start des VFX-Menüs werden automatisch die folgenden Einstellungen in VFP gemacht:

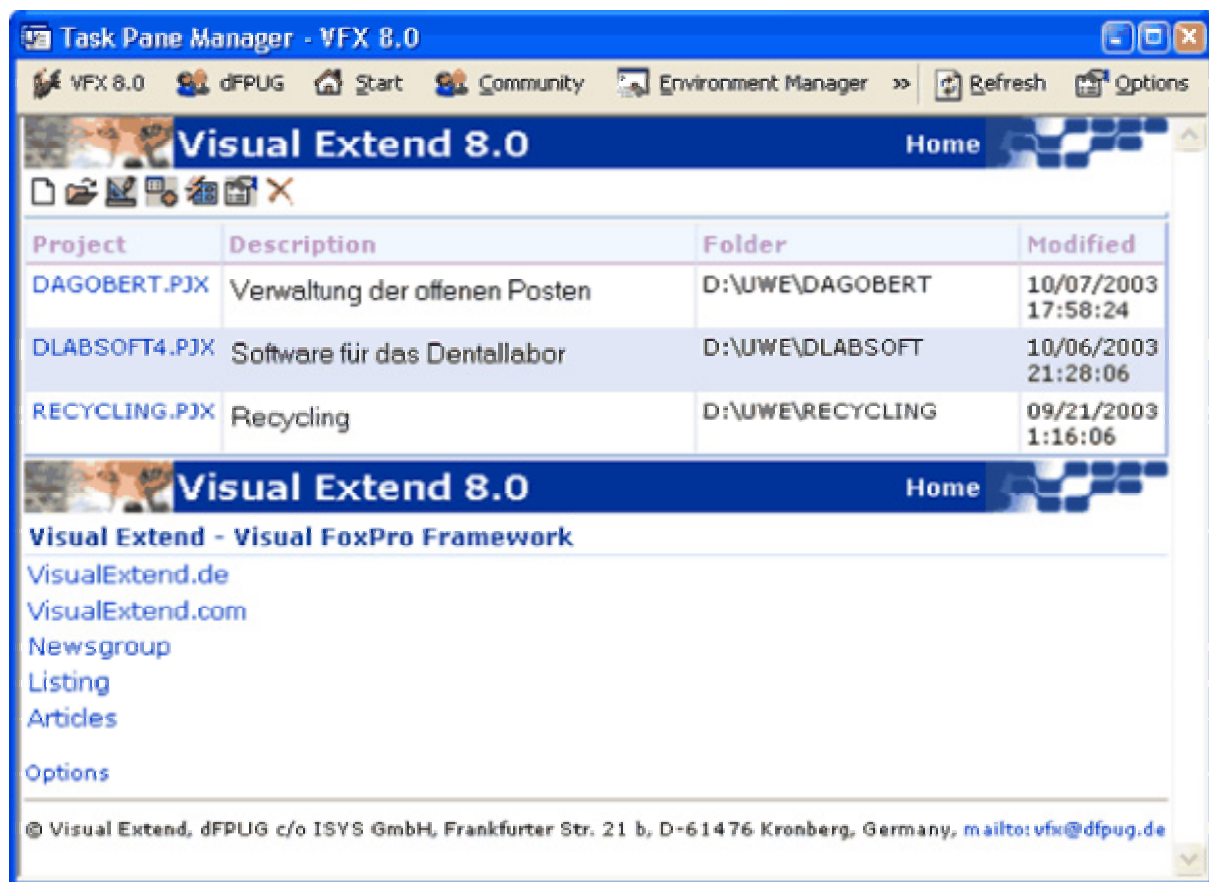
- **Builder:** zeigt Sie auf den VFX-Anwendungs-Assistenten mit dem Namen *VFXBLDR.APP* im Ordner *\VFX80\BUILDER*.
- **Suchpfad:** *\VFX80\BUILDER* wird dem Suchpfad hinzugefügt.

Wenn die Task Pane während des Starts des VFX-Menüs geöffnet ist, wird die VFX Task Pane automatisch in die VFP Task Pane integriert.

Das beste Werkzeug um zwischen Projekten zu wechseln, ist die VFX 8.0 Task Pane. Sie können die Task Pane über den VFP-Menüpunkt Extras, Task Pane öffnen. Wir empfehlen die VFP Task Pane beim Start von VFP automatisch öffnen zu lassen. Wählen Sie hierzu im Task Pane Manager die Option „Open the Task Pane Manager when Visual FoxPro starts“.

VFX 8.0 Task Pane

Öffnen Sie die VFP Task Pane. Beim nächsten Start von VFP wird automatisch die VFX 8.0 Task Pane integriert. Die VFX Task Pane enthält unter anderem alle Funktionen des aus früheren VFX-Versionen bekannten VFX-Application Manager.



Ein nützliches Tool befindet sich in der VFX 8.0 Task Pane, der Application Manager. In einer Tabelle werden Informationen über alle VFP-Projekte verwaltet. Über den VFX-Application Manager kann ein Projekt geöffnet werden. Dabei wird automatisch der Pfad in das Projektverzeichnis gesetzt. Außerdem kann über den VFX-Application Manager ein „Rebuild all“ durchgeführt werden. Dabei wird das Projekt komplett kompiliert. Änderungen in Include-Dateien werden berücksichtigt.

Erstellen einer Anwendung mit dem VFX - Application Wizard

Ziel

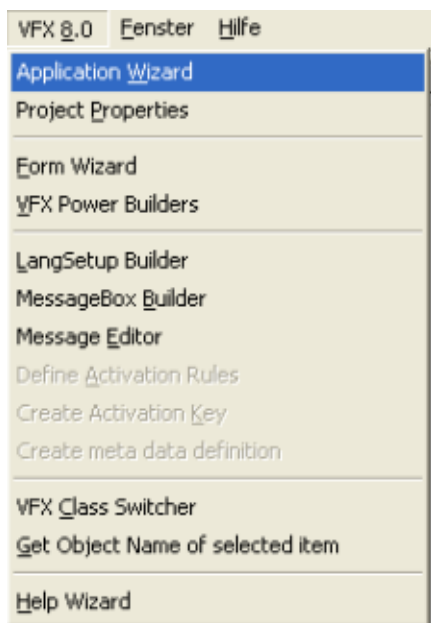
Wenn Sie ein neues Projekt beginnen, könnten Sie die ganze Ordnerstruktur von Hand erstellen, alle benötigten Dateien kopieren, wie etwa die Klassenbibliotheken, die Standardformulare, die Konfigurationsdateien, die Bilddateien usw. Hier greift der VFX-Anwendungs-Assistent ein: er erstellt das gesamte Projekt in der Sprache Ihrer Wahl. Er stellt außerdem die wichtigsten Eigenschaften der Anwendungsklasse ein und erstellt Include-Dateien mit den wichtigsten Konstanten, um die manuelle Arbeit so weit wie möglich zu reduzieren.

Vorbereitung

Schließen Sie alle Formulare und stellen Sie sicher, dass keine Klassenbibliotheken eines VFX-Projekts geöffnet sind. Am Besten beenden Sie Visual FoxPro und starten Sie erneut, bevor Sie den VFX-Anwendungs-Assistenten benutzen.

Der VFX - Application Wizard

Wählen Sie den Menüpunkt *Application Wizard* im VFX 8.0-Menü.



Oder starten Sie den *Application Wizard* aus der VFX Task Pane durch einen Klick auf das linke Symbol.

Der VFX-Application Wizard erscheint:

The screenshot shows the 'VFX - Application Wizard' window. The title bar says 'VFX - Application Wizard'. The main heading is '1. With this wizard you create a new VFX project'. Below this, there are four input fields: 'Master VFX home folder:' with the value 'c:\programme\wfx80\', 'Enter the name of the new project file:' with the value 'MAIN', 'Enter the name of the new project's folder:' (empty), and 'Database name:' with the value 'DATABASE.DBC'. There are '...' buttons next to the first three fields. Below the fields is the text 'Click on next to proceed.' At the bottom are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

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

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

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

Enter the name of the new project's folder. Geben Sie den Ordner für Ihr neues Projekt ein. Wenn der Ordner noch nicht existiert, so wird er von dem VFX-Application Wizard erstellt.

Database Name. Geben Sie den Namen Ihres Datenbank-Containers an (DBC). Geben Sie nur den Namen des Datenbank-Containers ohne Pfad und ohne Namenserweiterung ein.

Auf der Seite mit dem Titel *2. About* machen Sie die folgenden Eingaben:

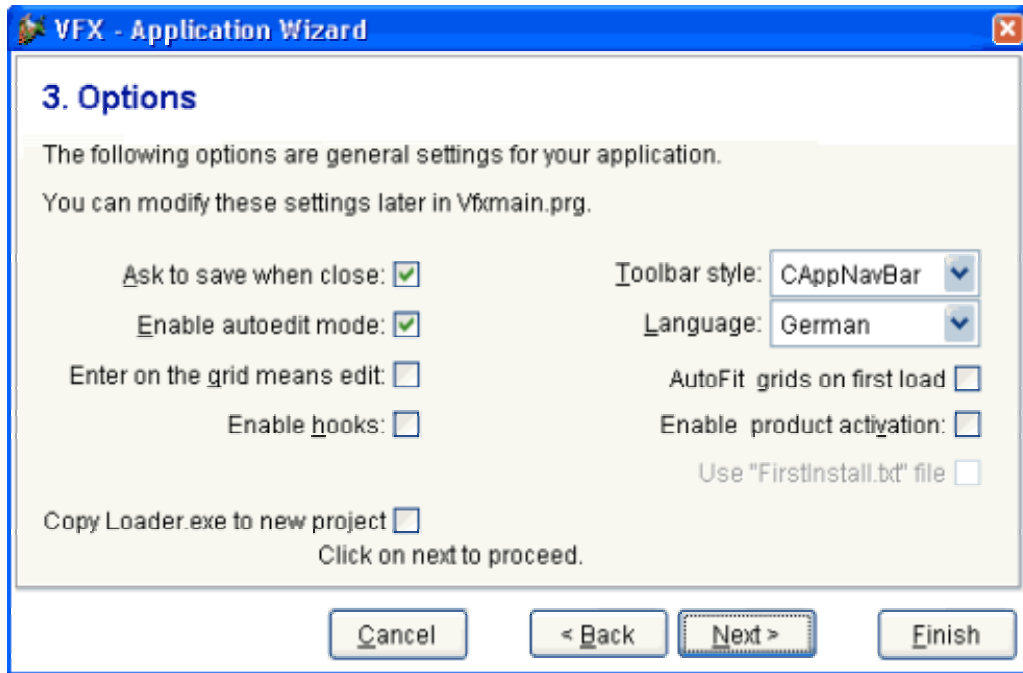
The screenshot shows the 'VFX - Application Wizard' window, Step 2: '2. About'. The title bar says 'VFX - Application Wizard'. The main heading is '2. About'. Below this is the text 'Enter the title of your application and the data which will be displayed in the About... dialog of your application.' There are three input fields: 'Application title:' with the value 'Visual Extend Application', 'Version:' with the value 'Version #', and 'Copyright:' with the value 'Copyright© dFPUG c/o ISYS GmbH'. At the bottom is the text 'Click on next to proceed.' and four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

Application title: Geben Sie die Überschrift für das Hauptfenster Ihrer Anwendung an. Diese Überschrift wird als Konstante CAP_APPLICATION_TITLE in der Include-Datei USERTXT.H gespeichert.

Version: Geben Sie die Versionsnummer für den Infodialog Ihrer Anwendung ein. Die Nummer wird in der Konstante CAP_LBLVERSION in der Include-Datei USERTXT.H gespeichert.

Copyright: Geben Sie Ihre Copyright-Information für den Infodialog Ihrer Anwendung ein. Diese Information wird in der Konstante CAP_LBLCOPYRIGHTINFORMATION in der Include-Datei USERTXT.H gespeichert.

Auf der Seite mit dem Namen 3. *Options* können Sie folgenden Optionen einstellen:



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

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

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

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

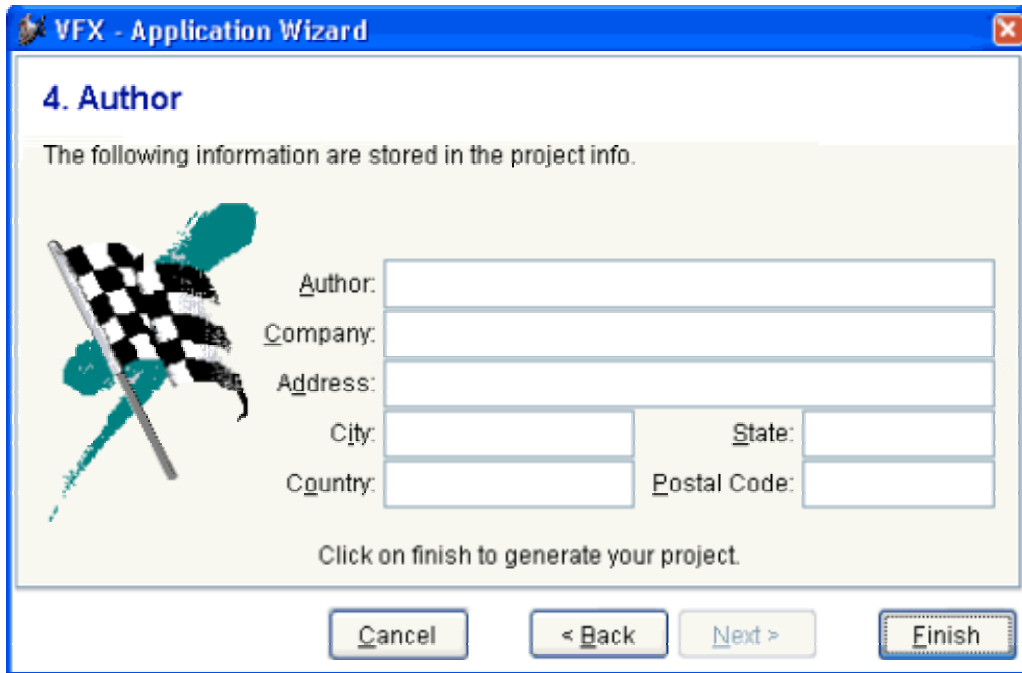
Toolbar style: Wählen Sie hier die Symbolleiste Klasse, die Sie verwenden wollen. *CAppNavBar* enthält Schaltflächen zur Bewegung des Datensatzzeigers und andere Schaltflächen zur Bearbeitung in der Standard-Symbolleiste. *CAppToolBar* enthält keine Schaltflächen zur Bewegung des Datensatzzeigers und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

Language: Wählen Sie die gewünschte Sprache für Ihr neues Projekt. Zurzeit können Sie aus folgenden Sprachen auswählen: deutsch, englisch, französisch, italienisch, spanisch, bulgarisch, tschechisch und griechisch.

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

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

Auf der Seite 4. *Author* können Sie Ihre persönlichen Daten eingeben, um Ihr Projekt zu dokumentieren.



VFX - Application Wizard

4. Author

The following information are stored in the project info.



Author:

Company:

Address:

City: State:

Country: Postal Code:

Click on finish to generate your project.

Diese Informationen werden in der Projektdatei gespeichert.

Erstellen des Projekts

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

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

Funktionsumfang der neuen Applikation

Die mit dem Application Wizard erstellte Applikation kann sofort getestet werden. Dazu kann direkt aus dem Projekt-Manager das Hauptprogramm *Vfxmain.prg* gestartet werden. Wahlweise kann auch eine App- oder Exe-Datei erstellt und getestet werden. Dies ist während der Entwicklung normalerweise aber nicht erforderlich.

Die Applikation startet mit einem Splashscreen. Als Bild für den Splashscreen wird eine Png-Datei verwendet, die der Entwickler leicht bearbeiten oder austauschen kann. Es ist möglich den Anmeldebildschirm zu umgehen. Nach Anzeige des Splashscreens baut sich der Hauptbildschirm auf und es erscheint der Anmeldebildschirm. Standardmäßig muss sich jeder Benutzer einer VFX-Applikation mit einem Namen und einem Kennwort anmelden. Es ist möglich den Anmeldebildschirm zu umgehen und den Benutzer automatisch mit dem Windows-Anmeldenamen anzumelden. Wahlweise kann die Benutzerverwaltung ganz abgeschaltet werden.

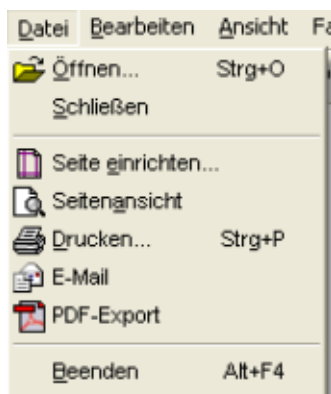
Diskussion der generierten VFX-Applikation

Nach einer erfolgreichen Anwendungsgenerierung mit dem VFX-Anwendungs-Assistenten, haben Sie eine lauffähige Anwendung mit allem was eine neue Anwendung benötigt vom Menü, über die Standard-Symbolleiste, die Benutzerverwaltung, die Systemeinstellungen, Datenbankwartung, ein Laufzeitfehlerprotokoll, ein Protokoll der Systemsperren bis hin zum Infodialog.

Office-kompatible Benutzeroberfläche

VFX erstellt Anwendungen, die nach dem *Office-Compatible*-Standard zertifiziert werden können.

Menü: Datei

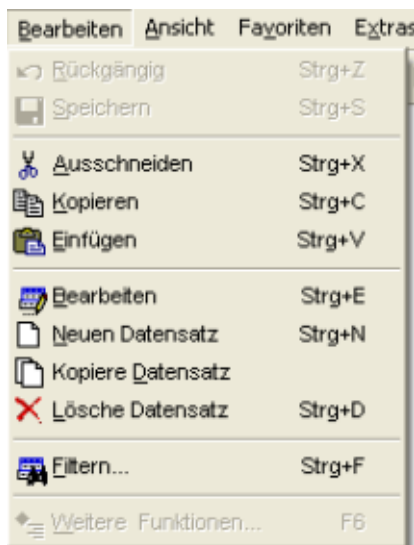


Mit einem Standard-*Datei/Öffnen*-Dialog wird die Komplexität von Menüs wesentlich reduziert. Der Benutzer öffnet Formulare immer durch einen einheitlichen Öffnen-Dialog. Standardmäßig wird der Öffnen-Dialog im Windows-XP-Stil am linken Bildschirmrand angezeigt.

VFX-Anwendungen bieten, dem *Office-Compatible*-Standard folgend, im Menü *Datei* die zuletzt geöffneten Dateien an. Wie viele Dateien angezeigt werden, ist für jeden Benutzer in der Benutzerverwaltung individuell einstellbar.

Auch die *Datei/Beenden* Option entspricht dem *Office-Compatible*-Standard.

Menü: Bearbeiten

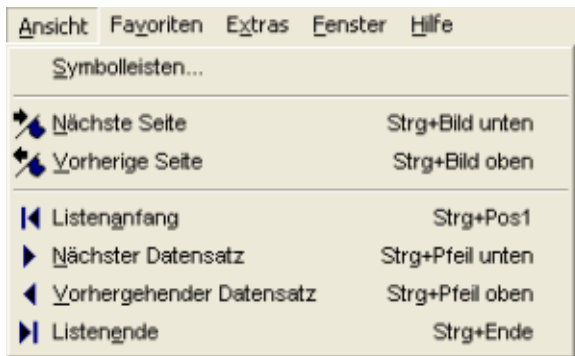


Hier befinden sich alle Funktionen zur Datenbearbeitung, die sich auf den aktuellen Datensatz beziehen sowie die Möglichkeit, die Formulare für Filtern und weitere Funktionen aufzurufen. Je nach Status des Formulars

Bearbeitungs-/Einfügemodus (`oForm.nFormStatus = 1` oder `2`) oder Anzeigemodus (`oForm.nFormStatus = 0`) sind einige der Optionen nicht verfügbar.

Um weitere Informationen zu erhalten, sehen Sie bitte im Kapitel *Das VFX Datenbearbeitungsformular* nach.

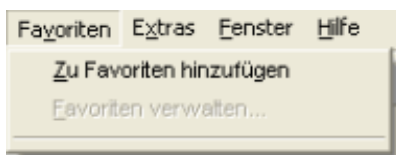
Menü: Ansicht



Hier können Sie den Symbolleisten-Dialog aufrufen, die Seite bei mehrseitigen Eingabefeldern wechseln, sowie den Datensatzzeiger bewegen.

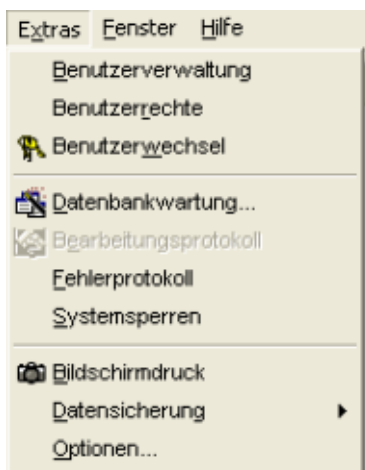
Um weitere Informationen zu erhalten, sehen Sie bitte im Kapitel *Das VFX Datenbearbeitungsformular* nach.

Menü: Favoriten



Dies ist das VFX-Favoriten-Menü. Mit der ersten Option wird der aktuelle Datensatz dem Favoriten-Menü hinzugefügt. Mit dem zweiten Eintrag werden die Favoriten verwaltet. Für alle verfügbaren Favoriten, gruppiert nach Formularen, werden Menüeinträge zur Laufzeit hinzugefügt.

Menü: Extras



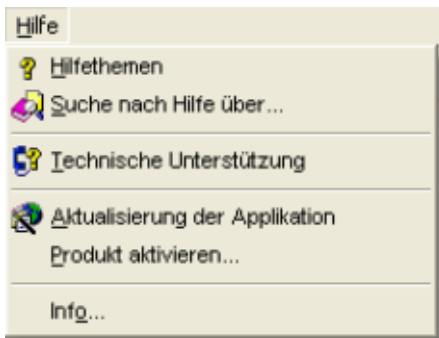
Um weitere Informationen zu den einzelnen Optionen zu erhalten, lesen Sie bitte in den Kapiteln Benutzerverwaltung, Benutzerrechte, Benutzerwechsel, Datenbankwartung, Bearbeitungsprotokoll und Fehlerprotokoll in diesem Handbuches nach.

Menü: Fenster



Falls Sie mehrere Fenster geöffnet haben, können Sie diese im Menü *Fenster* auswählen.

Menü: Hilfe



Das Hilfemenü bietet direkten Zugriff auf die Hilfedatei.

Standard-Symbolleiste

VFX-Anwendungen haben eine Standard-Symbolleiste, die Sie einfach um Ihre anwendungsspezifischen Schaltflächen erweitern können. Dadurch haben Benutzer einfachen Zugriff auf die Funktionen, die Ihre Anwendung bietet. Die VFX-Symbolleisten erscheinen im „Hot Tracking“ Layout.



<i>Neu (Strg+N)</i>	Anlegen eines neuen Datensatzes.
<i>Kopiere Datensatz</i>	Der angezeigte Datensatz wird in einen neuen Datensatz kopiert.
<i>Öffnen (Strg+O)</i>	Öffnet den Öffnen-Dialog am linken Bildschirmrand.
<i>Speichern (Strg+S)</i>	Speichern der Änderungen im aktiven Formular.
<i>E-Mail</i>	Versenden einer E-Mail aus der Berichtsausgabe aus dem aktiven Formular.
<i>PDF</i>	Erstellen einer PDF-Datei aus der Berichtsausgabe aus dem aktiven Formular.
<i>Drucken (Strg+P)</i>	Drucken eines Berichts oder einer Liste aus dem aktiven Formular.
<i>Seitenansicht</i>	Anzeige der Druckvorschau eines Berichts oder einer Liste aus dem aktiven Formular.
<i>Ausschneiden (Strg+X)</i>	Entfernt die Markierung und überträgt sie in die Zwischenablage.
<i>Kopieren (Strg+C)</i>	Kopiert die Markierung in die Zwischenablage.
<i>Einfügen (Strg+V)</i>	Fügt den Inhalt der Zwischenablage ein.
<i>Rückgängig (Strg+Z)</i>	Macht die Änderungen in aktuellen Formular rückgängig.
<i>Weitere Funktionen (F6)</i>	Öffnet das Fenster mit weiteren Funktionen zum aktuellen Formular.

<i>Bearbeitungsprotokoll</i>	Öffnet das Formular mit dem Bearbeitungsprotokoll zum aktuellen Datensatz im aktiven Formular.
<i>Bildschirminhalt drucken</i>	Die aktuelle Bildschirmansicht wird gedruckt.
<i>Bearbeiten (Strg+E)</i>	Schaltet das aktive Formular in den Bearbeitungsmodus.
<i>Löschen (Strg+D)</i>	Löscht den aktuellen Datensatz im aktiven Formular.
<i>Filtern (Strg+F)</i>	Filtern der Daten im aktiven Formular nach einzugebenden Kriterien.
<i>Anfang (Strg+Pos1)</i>	Bewegt den Datensatzzeiger auf den Anfang der Tabelle oder Ansicht.
<i>Rückwärts blättern (Strg+Pfeil oben)</i>	Bewegt den Datensatzzeiger auf den vorherigen Datensatz der Tabelle oder Ansicht.
<i>Vorwärts blättern (Strg+Pfeil unten)</i>	Bewegt den Datensatzzeiger auf den nächsten Datensatz der Tabelle oder Ansicht.
<i>Ende (Strg+Ende)</i>	Bewegt den Datensatzzeiger auf das Ende der Tabelle oder Ansicht.
<i>User</i>	Beispiel für eine individuell zu verwendende Schaltfläche.
<i>Hilfe (F1)</i>	Aufruf der kontextsensitiven Hilfe.
<i>Benutzerwechsel</i>	Ermöglicht die Anmeldung eines anderen Benutzers während das Programm läuft.
<i>Schließen (ESC)</i>	Das aktive Formular wird geschlossen.

Neben dieser Standard-Symbolleiste bietet Ihnen VFX an, eine formularspezifische Symbolleiste zu definieren. Alles was Sie tun müssen, ist eine Symbolleisten-Klasse zu definieren und den Namen dieser Symbolleiste in der Formular-Eigenschaft *CToolbarClass* einzutragen. VFX erledigt alles Weitere für Sie automatisch.

HINWEIS: Für eine ausführliche technische Beschreibung zur Benutzung von formularspezifischen Symbolleisten lesen Sie bitte in der VFX Technischen Referenz nach.

Abschließende Bemerkung zur Office-Kompatibilität

Je nach Art Ihrer Anwendung kann es erforderlich sein, vom Office-Compatible-Standard abzuweichen. Das VFX-Menü zeigt eine Alternative, die die meisten Bedürfnisse, aber nicht alle, von möglichen Anwendungen abdeckt. Es lohnt sich einige Zeit in den Aufbau des Menüs und der Symbolleisten zu investieren, die Sie in Ihren Anwendungen verwenden wollen.

Öffnen-Dialog

Formulare werden standardmäßig über den Öffnen-Dialog gestartet. Der Öffnen-Dialog erscheint im Windows XP-Layout. Die Daten der Formulare stehen in der Tabelle *Vfxfopen.dbf*.

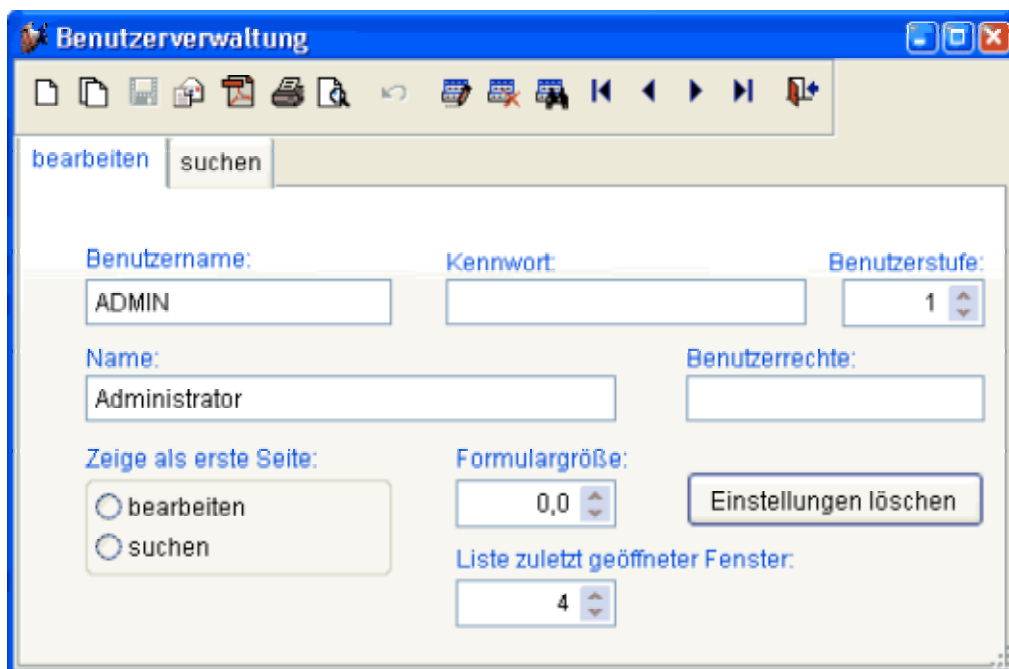


Benutzerverwaltung

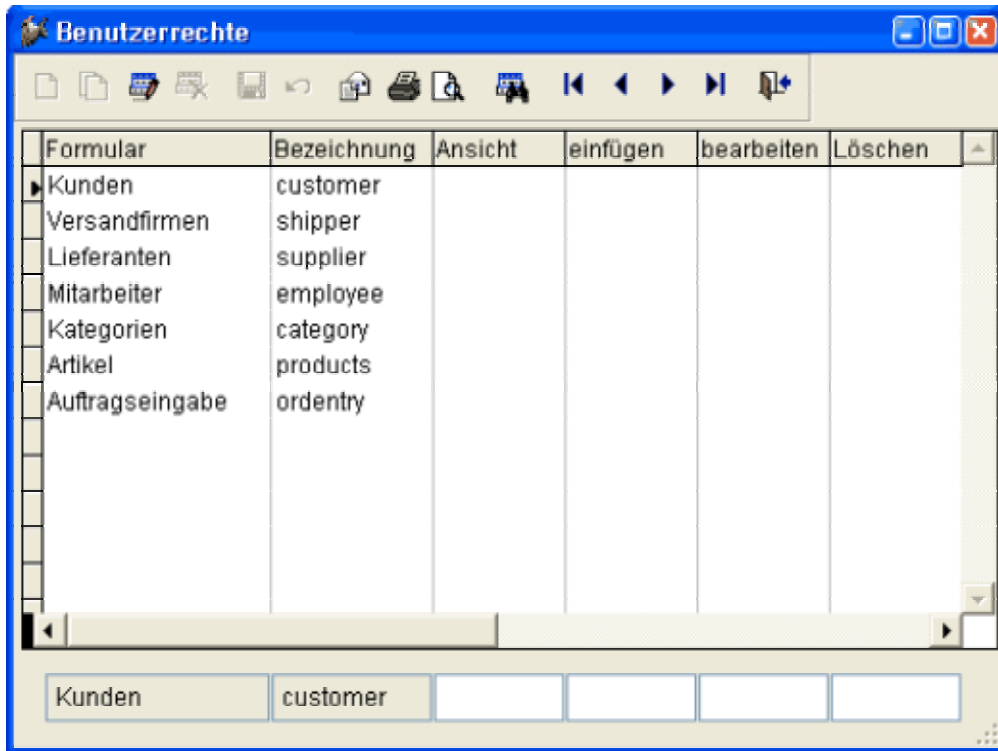
In jeder Mehrbenutzerapplikation sollte eine Benutzerverwaltung vorhanden sein. Als erstes muss festgelegt werden, wer zu Ihrer Anwendung Zugang hat. Dazu werden der Benutzername, das Kennwort und die Zugriffsrechte je Benutzer gespeichert. Eine weitere wichtige Funktion der Benutzerverwaltung ist die Speicherung der benutzerspezifischen Einstellungen.

Die Tabelle, in der die benutzerspezifischen Daten gespeichert sind, ist die freie Tabelle *VFXUSR.DBF/CDX*. Wenn Sie den Vorteil der langen Feldnamen nutzen möchten, können Sie diese Tabelle in Ihren Datenbank-Container einfügen.

Das Bearbeitungsformular, basierend auf der Klasse *CDataFormPage* wird automatisch vom VFX - Application Wizard vorbereitet.



Benutzer können ihre eigenen Daten in der VFX-Ressourcendatei löschen wenn sie mit neuen Einstellungen weitermachen wollen oder wenn sie von einer großen Bildschirmauflösung zu einer kleineren wechseln wollen oder wenn sie mit ihren bisherigen Einstellungen nicht mehr zufrieden sind. In der Ressourcendatei werden die Einstellungen für Formulargröße, Spaltenbreiten in Grids und Sortierfolgen in Grids und Auswahl-Grids sowie die Sortierfolgen gespeichert. Um die Daten in der VFX-Ressourcendatei zu löschen, drücken Sie auf die Schaltfläche *Einstellungen löschen*.



Die Zugriffsrechte werden über die Benutzerstufe gesteuert. Der Administrator hat die Benutzerstufe 1 und damit alle Rechte. Ein Benutzer, der die Benutzerstufe 99 hat, hat die wenigsten Rechte. Im Formular Benutzerrechte kann für jedes Formular festgelegt werden welche Benutzerstufe erforderlich ist um das Formular anzeigen zu können, um neue Datensätze erfassen zu können, um vorhandenen Datensätze bearbeiten zu können und um Datensätze löschen zu können.

ANMERKUNG: Benutzer können nicht die Daten von anderen Benutzern ändern, wenn diese eine höhere Sicherheitsstufe haben. Sicherheitsstufen starten mit 1 (Administrator) und enden mit 99 (niedrigste Sicherheitsstufe). Zusätzlich können Sie eine Zugriffszeichenfolge für die weitere Anpassung an Ihre Bedürfnisse festlegen. Für weitere Sicherheitsaspekte, besonders für alle VFX Formular-Sicherheitseigenschaften, lesen Sie bitte in der VFX Technischen Referenz nach.

Wenn ein Benutzer nicht das Recht hat ein Formular anzuzeigen, wird das betreffende Formular nicht instanziiert. Solange im Dialog Benutzerrechte keine Benutzerstufen eingetragen sind, gelten die Einstellungen, die mit dem VFX - Form Wizard in den Formular-Eigenschaften *lcaninsert*, *lcancopy*, *lcanedit* und *lcandelete* hinterlegt sind.

Datenbankwartung

Durch Auswahl des Menüpunktes *Extras, Datenbankwartung* erscheint der folgende Dialog:



In diesem Dialog sehen Sie eine Liste mit allen in Ihrer Anwendung verfügbaren Tabellen. In einem einfach zu bedienenden VFX-Mover-Dialog können die Tabellen ausgewählt werden, die bearbeitet werden sollen.

Es kann aus einer der folgenden Optionen ausgewählt werden:

- Komprimieren (pack)
- Memos packen (pack memo)
- Neu indizieren (reindex)

Drücken Sie nach der Auswahl auf *OK*, um die gewünschte Datenbankwartung durchzuführen.

HINWEIS: Der hier verwendete Mover-Dialog ist ebenfalls eine VFX-Klasse und steht auch für Ihre eigenen Anwendungen zur Verfügung.

Fehlerprotokoll

VFX protokolliert alle Laufzeitfehler automatisch. Die Tabelle mit den Fehlermeldungen ist die freie Tabelle *VFXLOG.DBF/CDX*.

Das Bearbeitungsformular, basierend auf der Klasse *CDataFormPage*, wird automatisch vom VFX Anwendungs-Assistenten vorbereitet.

The screenshot shows the 'Fehlerprotokoll' window with the following details:

- Title Bar:** Fehlerprotokoll
- Toolbar:** Contains icons for file operations (open, save, print, etc.) and navigation (back, forward, etc.).
- Tabs:** 'Daten' (active) and 'suchen'.
- Form Fields:**
 - Typ:** ERROR
 - Benutzer:** Uwe Habermann
 - Fehler:** 7
 - Datum:** 11.10.2003
 - Zeit:** 16:42:56
- Text Area:** Contains the message 'File already exists.' with a vertical scrollbar.
- Buttons:** 'Alles löschen' (Delete all) at the bottom right.

Der Administrator kann das Fehlerprotokoll mit der Schaltfläche *Alles löschen* löschen.

Systemsperrn

In viel benutzten Mehrbenutzerumgebungen kann eine Meldung wie *“Datensatz durch anderen Benutzer gesperrt”* unter Umständen nicht ausreichen. Für solche Fälle stellt VFX eine System-Sperrentabelle zur Verfügung. In dieser Tabelle wird gespeichert, welcher Benutzer seit wann welchen oder welche Datensätze in Benutzung hat. (Siehe die Funktionen *XLock()* sowie *XUnlock()* in der Technischen Referenz unter *Funktionen*).

Die Systemsperrtabelle in der alle Sperren mit VFX-Funktionsaufrufen gespeichert werden, ist die freie Tabelle *VFXLOCK.DBF/CDX*.

Das Bearbeitungsformular basiert auf der VFX-Klasse *CDataFormPage* und wird automatisch durch den VFX Anwendungs-Assistenten vorbereitet.

Der Administrator kann die Systemsperrn mit der Schaltfläche *Alles löschen* löschen.

Optionen

Im Gegensatz zu den benutzerspezifischen Einstellungen werden in der Tabelle *VFXSYS.DBF* die systemspezifischen Einstellungen gespeichert.

Das oben abgebildete Formular ist eine Vorlage, die für die eigenen Optionen verwendet werden kann.

Der VFX Anwendungs-Assistent erstellt das Formular *VFXSYS* für Sie in einer gebrauchsfertigen Form. Dieses Formular basiert auf der Klasse *CSystemDialog*. Alles was Sie noch tun müssen ist, die gewünschten Felder in der *VFXSYS* Tabelle anzulegen und die entsprechenden Steuerelemente direkt mit der jeweiligen Public Variablen *gs_* als Datenquelle einzufügen.

Auch hier, wie bei den benutzerspezifischen Einstellungen, die weiter oben beschrieben sind, wird für jedes Feld aus der Tabelle *VFXSYS* eine Public Variable mit dem Präfix *gs_* angelegt. VFX übernimmt auch hier vollautomatisch das Speichern und Wiederherstellen dieser Werte falls diese aus dem Optionen-Dialog heraus verändert werden.

Wenn Sie ein Feld mit dem Namen *TEST* in der Tabelle *VFXSYS* haben, wird eine Public Variable *gs_test* den Wert aus dem Feld *TEST* der *VFXSYS*-Tabelle beinhalten. Falls diese Variable verändert wird, wird beim Verlassen des Optionen-Dialogs dieser Wert wieder zurück in das Feld *Test* der Tabelle *VFXSYS* geschrieben.

Auf diese Weise ist es sehr einfach, systemspezifische Einstellungen zu speichern und wiederherzustellen. Es genügt hierzu, in der Tabelle *VFXSYS* die entsprechenden Felder als Variable hinzuzufügen. Das ist sehr einfach. Probieren Sie es!

Infodialog

Der VFX-Anwendungs-Assistent erstellt einen Infodialog, der auf der Klasse *CAboutDialog* basiert.

Sie finden den Infodialog im Menü Hilfe.



Um diesen Dialog Ihren Bedürfnissen anzupassen, steht Ihnen die Include-Datei *USERTXT.H* zur Verfügung:

```
#define CAP_APPLICATION_TITLE           "VFX 8.00 Build 0000 Test Application"
#define CAP_LBLCOPYRIGHTINFORMATION    "Copyright © dFPUG c/o ISYS GmbH"
#define CAP_LBLTHISPRODUCTISLICENSED  "This product is licensed to:"
#define CAP_LBLTRADEMARKINFORMATION   "Trademark Information"
#define CAP_LBLVERSION                 "Version "
#define CAP_LBLYOURAPPLICATIONNAME     "VFX Test Application"
```

Die VFX-BUILDER

Ziel

Formulare manuell zu erstellen kann viel Zeit beanspruchen, insbesondere dann, wenn Sie viele Formulare mit vielen Feldern anzeigen möchten. Stellt man sich zum Beispiel ein Formular mit 20 Feldern vor, so hat man bereits 40 Steuerelemente, allein für die Dateneingabefelder: 20 Textfelder oder andere Steuerelemente und 20 Bezeichnungen. Wenn Sie Klassenbibliotheken verwenden, müssen Sie Ihre Symbolleiste anpassen oder die gewünschten Steuerelemente per Drag & Drop auf das Formular ziehen. Mit den VFX-Formular-Buildern wird diese Aufgabe sehr schnell und einfach durchführbar.

Ein weiterer großer Vorteil der VFX-Formular-BUILDER ist die Wiederverwendbarkeit. Das bedeutet, dass Sie Änderungen, die Sie in Ihrer Datenbank gemacht haben, einfach in das bestehende Formular übernehmen können, indem Sie den VFX-Formular-BUILDER aufrufen und das Kontrollkästchen *Use DBC Definitions* auswählen. Auch Seiten zu einem Seitenrahmen hinzuzufügen oder Änderungen an den Spalten eines Grids sind sehr einfach dank der Wiederverwendbarkeit der VFX-Formular-BUILDER.

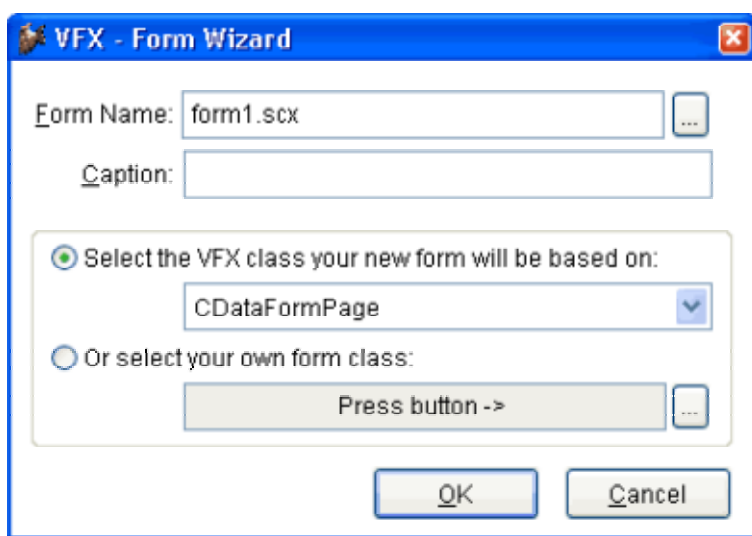
Vorbereitung

Zuerst müssen Sie die Datenbank für Ihre Anwendung erstellen. Legen Sie Ihre Tabellen, Felder und Indexschlüssel an.

ANMERKUNG: Wenn Sie die Daten für Überschriften, Formate, Eingabeformulare und Bibliothek für Anzeige im Datenbank-Container speichern, werden diese automatisch von den VFX-Formular-Buildern und vom VFX-Grid-BUILDER verwendet.

Erstellen eines Formulars mit dem VFX-Form Wizard

Mit Hilfe des VFX-Form Wizard wird ein neues Formular auf der Basis einer VFX-Formularklasse angelegt, in das Projekt eingetragen und zum Bearbeiten geöffnet.



Datenumgebung

Die von dem Formular zu verwendenden Tabellen oder Ansichten sind in der Datenumgebung einzutragen. Der VFX-Form Builder liest die Datenumgebung aus und stellt die Felder der Tabellen zur Auswahl um Steuerelemente zu erstellen. Zur Laufzeit wird die Datenumgebung ebenfalls ausgelesen um die Tabellen zu ermitteln, für die ein Tableupdate bzw. Tablerevert durchgeführt werden muss.

Der VFX-CDataFormPage Builder

Aufruf eines VFX-Formular-Builders

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

ANMERKUNG: Wenn Sie eine Meldung erhalten, dass es keinen Builder für das ausgewählte Objekt gibt oder wenn ein Standard-Visual FoxPro-BUILDER erscheint, prüfen Sie, ob Sie das Formularobjekt ausgewählt haben. Ein verbreiteter Fehler ist, den Seitenrahmen anstelle des Formularobjektes auszuwählen. Prüfen Sie das angezeigte Objekt im Eigenschaftsfenster, wenn Sie nicht sicher sind, ob das Formularobjekt ausgewählt ist.

Der VFX-CDataFormPage Builder wird geladen und zeigt einen benutzerfreundlichen Dialog.

Die Bedienung des VFX-Formular-Builders

Die VFX-Formular-BUILDER haben eine intuitive Bedienung.

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

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

Abhängigkeit vom Aufruf des Formulars haben soll, brauchen Sie sich um diese Überschrift keine Gedanken zu machen. Geben Sie in diesem Fall einfach eine mehr oder weniger zutreffende Überschrift ein.

Der VFX – CDataFormPage Builder hat einen Seitenrahmen mit drei Seiten mit den Namen *Edit Pages*, *Grid Page* und *Form Options*. Auf der Seite *Edit Pages* definieren Sie den Seitenrahmen, den Sie zur Bearbeitung der ausgewählten Felder benutzen. Auf der *Grid Page* definieren Sie das *Grid* für die Suche und auf der Seite *Form Options* setzen Sie verschiedene Optionen für das Formular.

Die folgenden Optionen sind auf der Seite *Edit Pages* verfügbar:

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

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

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

Für jede Bearbeitungsseite stehen die folgenden Optionen zur Verfügung:

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

Control Type. Geben Sie für alle ausgewählten Felder den zu benutzenden Steuerungstyp an. Die folgenden Klassen stehen hierfür zur Verfügung:

Steuerelement	Beschreibung	VFX Klassenbibliothek
<Standardwert>	Die Klasse, die Sie als Klasse für die Anzeige im Datenbank-Container angegeben haben (Standardwert).	
CTextBox	Normales Textfeld.	VFXOBJ.VCX
CKeyField	Textfeld für das Bearbeiten von Identifikationsfeldern, die nach Anlegen des Datensatzes nicht mehr verändert werden dürfen.	VFXOBJ.VCX
CFixField	Textfeld für das Bearbeiten von Feldern, die in einer Child-Tabelle mit einer Haupttabelle verbunden sind. Dieses Steuerelement wird verwendet, wo ein Child-Formular von einem Parent-Formular aufgerufen wird und einen festen Wert aus dem Parent-Formular übergeben erhält, z. B. bei Aufträgen von einem Kunden. In diesem Fall würde das Kundenfeld im Formular Aufträge ein <i>CfixField</i> sein, denn im Falle des Aufrufes der Aufträge von einem Kunden muss das Kundenfeld vorgelegt werden und darf nicht änderbar sein.	VFXOBJ.VCX
CPickField	Eingabefeld, bei dem die Eingabe mit Hilfe einer Tabelle oder Ansicht überprüft wird und auf Wunsch eine Auswahlliste zur Verfügung gestellt wird. Es wird nicht nur die Eingabe überprüft, sondern zusätzlich ein beliebiger Ausdruck aus der Prüftabelle geholt.	VFXOBJ.VCX
CEditBox	Bearbeitungsfeld für das Bearbeiten von Memofeldern und längeren Zeichenfeldern.	VFXOBJ.VCX
CComboBox	Kombinationsfeld.	VFXOBJ.VCX
CListBox	Listenfeld.	VFXOBJ.VCX
CCheckBox	Kontrollkästchen für logische Felder.	VFXOBJ.VCX
COptionGroup	Optionsgruppe.	VFXOBJ.VCX
CSpinner	Drehfeld für numerische Felder.	VFXOBJ.VCX

ANMERKUNG: Um Ihre eigenen Klassen zu verwenden, tragen Sie diese im Datenbank-Container bei jedem Feld bei „Bibliothek für Anzeige“ ein!

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

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

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

Status Bar. Meldung für die Statuszeile für dieses Feld. Der Standardwert wird aus dem Datenbank-Container übernommen. (Eigenschaft Feldkommentar, wenn dieser Wert leer ist, wird die Feldüberschrift genommen).

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

Die folgenden Optionen stehen auf der Seite *Grid Page* zur Verfügung:

The screenshot shows the 'VFX - CDataFormPage Builder' dialog box with the 'Grid Page' tab selected. The 'Form Name' is 'frmCustomer' and the 'Caption' is 'Kunden'. The 'Grid Page Title' is '\<Liste' and the 'Grid Class' is 'cgrid'. The 'Fields Selected' list includes 'customer.customer_id', 'customer.company_name', 'customer.contact_name', 'customer.contact_title', 'customer.address', 'customer.city', 'customer.region', and 'customer.postal_code'. The 'Control Type' is 'textbox', the 'Header' is 'Nummer', the 'Control Source' is 'customer.customer_id', and the 'Output Mask' is 'XXXXXX'. The 'Read Only' and 'Incremental Search' checkboxes are checked. The 'Use DBC Definitions' checkbox is unchecked, and the 'Overwrite Font' checkbox is checked. The 'OK', 'Apply', and 'Cancel' buttons are at the bottom.

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

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

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

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

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

Control Type. Geben Sie für alle ausgewählten Felder den gewünschten Kontrolltyp an. Die folgenden Kontrolltypen sind verfügbar (aus Gründen der Geschwindigkeitsoptimierung bieten wir nur VFP-Basisklassen für das Grid an):

Kontrolltyp	Beschreibung	VFP-Basisklasse
Textbox	Textfeld (Standard)	TEXTBOX
Editbox	Bearbeitungsfeld	EDITBOX
Combobox	Kombinationsfeld	COMBOBOX
Checkbox	Kontrollkästchen	CHECKBOX

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

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

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

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

Die folgenden Optionen sind auf der Seite *Form Options* verfügbar:

The screenshot shows the 'VFX - CDataFormPage Builder' dialog box with the 'Form Options' tab selected. The 'Form Name' is 'frmCustomer' and the 'Caption' is 'Kunden'. The 'Report Name' field is empty. The font is set to 'Arial,9,N'. The following options are checked: 'Is Child Form', 'Has More Functions', 'Has Linked Child Form', 'Auto Sync. Child Form', 'Put In Last File Menu', 'Put In Window Menu', 'Can Edit', 'Can Insert', 'Can Copy', 'Can Delete', 'Multi Instance', 'Close with ESC Key', 'Save/Restore Positions', and 'Overwrite Font'. The 'Add SpeedBar Control' and 'Work on View!' options are unchecked. At the bottom, 'Use DBC Definitions' is unchecked and 'Overwrite Font' is checked. The 'OK', 'Apply', and 'Cancel' buttons are at the bottom right.

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

Is Child Form. Wenn das Formular, das Sie gerade erstellen, von einem anderen Formular aufgerufen wird, ist dieses Formular ein Child-Formular.

ANMERKUNG: Bitte verwechseln Sie dies nicht mit dem später beschriebenen 1:n-Formular, wo Sie die Haupttabelle und die Child-Tabelle **auf dem gleichen Formular** bearbeiten können. Hier sprechen wir über folgendes Verhalten: Formular 1 ruft Formular 2 auf, wobei Formular 1 das Hauptformular und Formular 2 das Child-Formular ist. Im Formular 2 sehen Sie nur die Datensätze, die ein bestimmtes Kriterium erfüllen, das die Verbindung zur Haupttabelle im Formular 1 herstellt.

Wenn Sie beispielsweise in einem Formular die Aufträge eines Kunden anzeigen wollen, markieren Sie dieses Kontrollkästchen und der VFX-Formular-Builder wird das Formular automatisch als Child-Formular erstellen. Dabei werden automatisch die erforderlichen Codezeilen in die *Init()*-Methode des Formulars eingetragen. Sie müssen nur noch den Code der *Init()*-Methode prüfen und an Ihre Bedürfnisse anpassen.

Für weitere Details lesen Sie bitte im Abschnitt *Erweiterte Formulareigenschaften mit dem VFX-Formular-Builder* weiter unten in diesem Handbuch nach.

ANMERKUNG: Wenn Sie ein Formular haben, das sowohl als Child-Formular als auch als normales Formular dienen soll, markieren Sie die Option *Is Child Form*. Sie brauchen hierfür nicht zwei Formulare zu erstellen. Ein Formular kann sowohl alle Aufträge darstellen als auch nur die Aufträge eines bestimmten Kunden.

Has More Functions. Wenn das Formular, das Sie gerade erstellen, andere Formulare aufrufen oder Aktionen ausführen soll, müssen Sie dieses Kontrollkästchen markieren. Dadurch wird automatisch der erforderliche Code für die *OnMore()*-Methode Ihres Formulars erstellt. Sie müssen nur noch den Code in der *OnMore()*-Methode an Ihre Bedürfnisse anpassen. Normalerweise werden Sie eine Anzahl von Aktionen haben, die zur Auswahl in einem Formular angeboten werden. Der Benutzer kann dann die gewünschte Aktion auswählen.

Für weitere Details lesen Sie bitte im Abschnitt *Erweiterte Formulareigenschaften mit dem VFX-Formular-Builder* weiter unten in diesem Handbuch nach.

Has Linked Child Form. Wenn das Formular, das Sie gerade erstellen, Child-Formulare aufrufen soll, die dynamisch mit diesem Hauptformular verbunden bleiben, markieren Sie dieses Kontrollkästchen. Dadurch wird automatisch der Code für die Formularymethode *OnSetChilddata()* erstellt. Diese Methode wird automatisch für jedes vorhandene Child-Formular aufgerufen.

Autosynch Child Form. Hiermit wird die Formulareigenschaft *IAutosynchChildform* festgelegt. Dadurch wird angegeben, ob die Child-Formulare automatisch mit diesem Hauptformular synchronisiert werden, wenn Sie den Datensatzzeiger im Hauptformular bewegen.

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

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

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

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

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

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

Multi Instance. Hiermit wird die Formulareigenschaft *IMultiInstance* eingestellt. Standardmäßig können alle Formulare, die Sie mit VFX erstellen, mehrmals geöffnet werden (das nennt man multiinstanzfähig). Dies ist eine großartige Eigenschaft. Alles was Sie dabei beachten müssen ist, dass das Formular mit einer privaten Datensitzung arbeiten muss. Das ist der Standardwert in allen VFX-Formularen.

Trotzdem ist es manchmal günstig, die Eigenschaft multiinstanzfähig ausschalten zu können. Daher haben wir die Eigenschaft *IMultiInstance* eingeführt. Setzen Sie diese Eigenschaft auf *.F.* und das Formular kann nur einmal geöffnet werden.

Close with ESC key. Hier wird die Formulareigenschaft *ICloseonEsc* eingestellt, die angibt, ob der Benutzer ein Formular mit der Escape-Taste schließen kann.

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

Add Speedbar Control. Dieses Kontrollkästchen fügt dem Formular eine Schaltflächenleiste hinzu. Hier ein Beispiel:



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

Der Formularerstellungsprozess kann mehrmals gestartet werden. Diese Eigenschaft nennt man wieder verwendbar.

Anmerkung: Die Eigenschaft wieder verwendbar ist zu 100% nur für Formulare verfügbar, die mit dem VFX-Formular-Builder erzeugt wurden. Um das wieder verwendbare Verhalten des Builders sicherzustellen sollten Sie immer den VFX-Formular-Builder verwenden, wenn Sie Ihrem Formular Felder hinzufügen wollen.

Ein weiterer großer Vorteil der wieder verwendbaren VFX-Formular-Builder ist die Tatsache, dass Sie Änderungen, die Sie in der Datenbank (z. B. Überschrift, Format oder Eingabemaske) durchgeführt haben, durch Aufrufen des VFX-Formular-Builders und auswählen des Kontrollkästchens *Use DBC Definitions* in das Formular übernehmen können.

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

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

Test

Das Formular kann direkt aus dem Formular-Designer oder aus dem Projekt-Manager gestartet und getestet werden. In der Init-Methode aller VFX-Formulare wird geprüft, ob das Applikationsobjekt existiert. Falls dieses nicht vorhanden ist, wurde das Formular direkt aus dem Projekt-Manager gestartet und VFX stellt selbständig die Umgebung her, um das Formular voll funktionsfähig laufen zu lassen. Dabei wird auch die Hauptsymbolleiste instanziiert und kann für die Bedienung des Formulars verwendet werden.

Natürlich ist es auch möglich das Projekt über das Hauptprogramm *Vfxmain.prg* zu starten. Das Formular kann dann über den Öffnen-Dialog gestartet werden.

Nächste Schritte

Um mit dem VFX-Formular-Builder besser vertraut zu werden, lohnt es sich, einige Formulare zu generieren. Beginnend mit einfachen Formularen, später auch Formularen, die andere Formulare aufrufen.

Nachdem Sie mit dem Erstellen von Standard-VFX-Datenbearbeitungs-Formularen vertraut sind, können Sie sich den *1:n-Datenbearbeitungs-Formularen* zuwenden.

Der VFX – Cgrid Builder

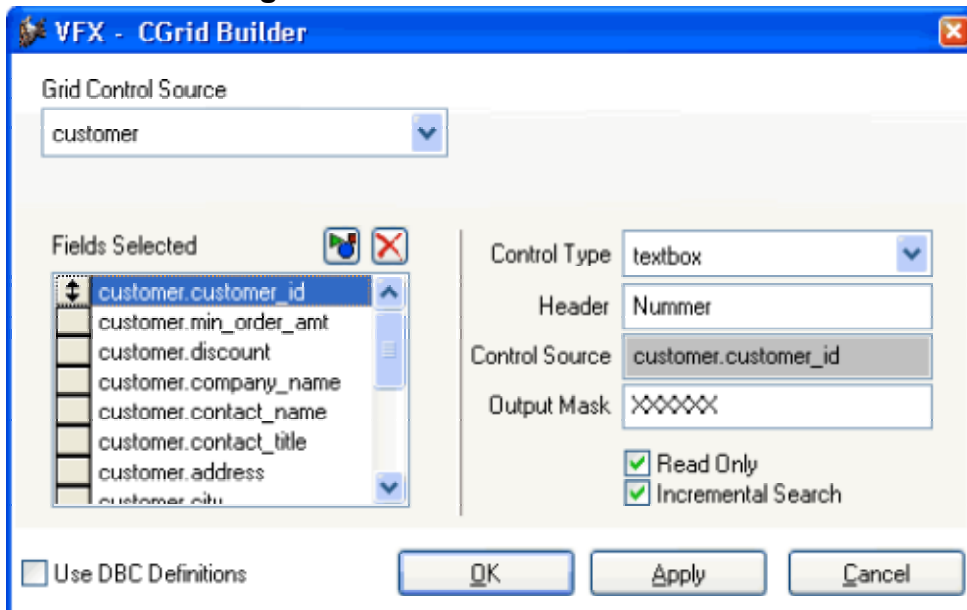
Aufruf des VFX – Cgrid Builder

Obwohl der VFX-Formular-Builder bereits eine Seite mit einem Grid anlegt, kann es sein, dass Sie nur in diesem Grid Änderungen durchführen wollen. Der VFX - CGrid Builder automatisiert die Erstellung von leistungsfähigen Grids. Die resultierenden VFX Power Grids sind einfach zu bedienen und bringen keine Geschwindigkeitseinbußen mit sich. Sie werden die Eigenschaften der VFX Power Grids sehr nützlich finden. Die inkrementelle Suche sowie die benutzerspezifische Speicherung der Spaltenreihenfolge, Spaltenbreiten und Sortierfolge des Grids werden von den Benutzern Ihrer Anwendung geschätzt werden.

Um den VFX – CGrid Builder aufzurufen, wählen Sie die letzte Seite Ihres Formulars und wählen Sie das Grid-Steuerelement aus. Um den Builder aufzurufen, drücken Sie die rechte Maustaste und wählen Sie Builder.

Der VFX – CGrid Builder wird geladen und zeigt den folgenden Dialog:

Die Bedienung des VFX-CGrid Builder



Die Bedienung ist die gleiche wie auf der Grid-Seite des VFX-Formular-Builders. Für eine detaillierte Beschreibung aller Optionen lesen Sie bitte die Beschreibungen im Abschnitt *Der VFX-Formular-Builder* nach.

Der VFX-Pickfield Builder

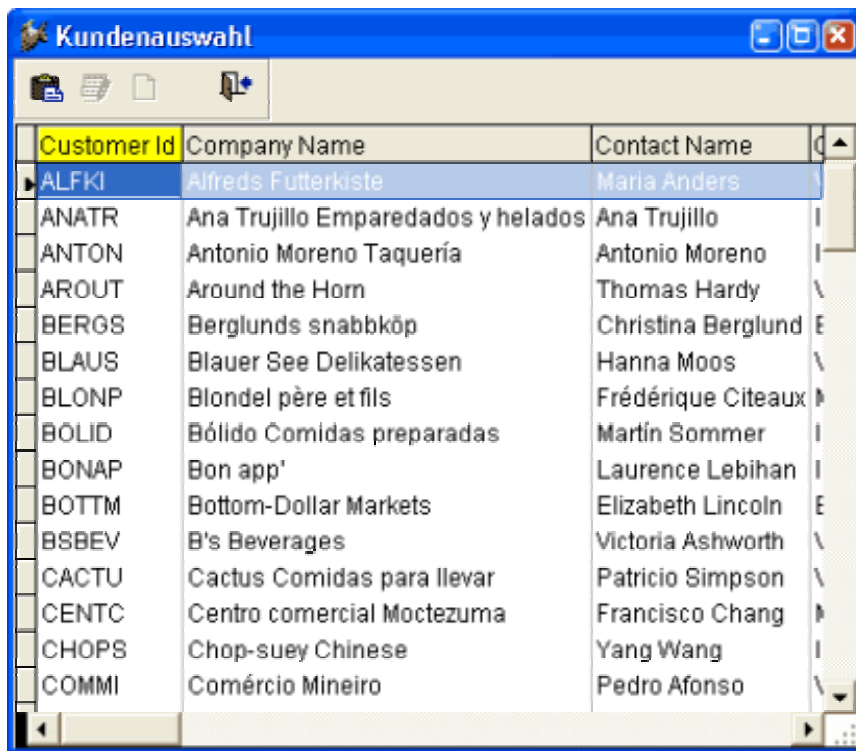
Ergebnis

Wenn Sie ein Auswahllisten-Steuerelement auf einem Formular einsetzen, sieht das etwa so aus:

Kunde:

Der Benutzer kann die Auswahlliste auf folgende Weise aufrufen:

- Drücken der Schaltfläche neben dem Auswahllisten-Eingabefeld (normalerweise mit drei Punkten beschriftet).
- Doppelklick auf das Auswahllisten-Eingabefeld oder auf den Beschreibungstext.
- Drücken der Funktionstaste F9.



Der Dialog der Auswahlliste hat folgende Eigenschaften (wie jedes VFX Power Grid):

- Inkrementelle Suche mit automatischer Einstellung der Sortierfolge.
- Einstellen der Sortierfolge durch Doppelklick auf die Spaltenüberschrift.
- Die Breite der Spalten kann verändert werden.
- Position und Gestaltung des Grids werden automatisch gespeichert.

Der Benutzer kann den gewünschten Datensatz auf folgende Weise auswählen:

- Doppelklick.
- Drücken der Taste *Eingabetaste*.
- Drücken der Schaltfläche *Übernehmen*.

Wenn der Benutzer die Tabelle bearbeiten möchte, die der Auswahlliste zugrunde liegt, kann er auf die Schaltfläche *Bearbeiten...* drücken. Daraufhin erscheint das Bearbeitungsformular für diese Tabelle. Wenn der Benutzer neue Datensätze hinzufügen will, drückt er auf die Schaltfläche *neu*.

Aufruf des VFX – CPickField Builder

Ein Steuerelement, das Sie häufig benutzen werden, ist die Auswahlliste. Dieses Container-Steuerelement bietet Ihnen eine einfache Möglichkeit Textfelder hinzuzufügen, die die Benutzereingabe mit Werten aus einer Tabelle oder Ansicht überprüfen. Außerdem kann der Benutzer ein Auswahllisten-Formular öffnen, um den gewünschten Datensatz auszuwählen. Da die Auswahllisten-Klasse einige Eigenschaften hat, die definiert werden müssen, hilft Ihnen der VFX – CPickField Builder bei der einfachen Erstellung der Auswahllisten-Steuerelemente. Und das ohne eine einzige Zeile Code oder Text im Eigenschaftsfenster des Auswahllisten-Containers manuell eintragen zu müssen!

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

ANMERKUNG: Um ein Steuerelement auszuwählen, das sich auf einer Seite in einem Seitenrahmen auf einem Formular befindet, müssen Sie den Visual FoxPro-Weg benutzen, um Steuerelemente innerhalb der Containerhierarchie auszuwählen (Klick, Rechtsklick, bearbeiten). Eine gute Möglichkeit, um festzustellen ob Sie das richtige Steuerelement ausgewählt haben, ist ein Blick in das Eigenschaftsfenster.

Die Bedienung des VFX – CPickField Builder

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Auf der Seite *Update* stehen die folgenden Optionen zur Verfügung:

The screenshot shows the 'VFX - CPickField Builder' dialog box with the 'Update' tab selected. The 'Update Source Fields' text box contains 'company_name;address;city;region;postal_code;country'. The 'Target Table Name' dropdown menu is set to 'orders'. The 'Update Target Fields' text box contains 'ship_to_name;ship_to_address;ship_to_city;ship_to_region;ship_to_postal_code;ship_to_country'. At the bottom are 'OK', 'Apply', and 'Cancel' buttons.

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

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

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

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

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

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

Auf der Seite *Work on View* stehen die folgenden Optionen zur Verfügung:

The screenshot shows the 'VFX - CPickField Builder' dialog box with the 'Work on View' tab selected. The 'Work on View' checkbox is checked. Under 'Validation Mode', 'Use Select Command' is selected with a radio button, and there is an empty text box below it. 'Use View' is also an option with a radio button and an empty text box below it. The 'Use SQL Pass Through' checkbox is unchecked. At the bottom, the 'Pick Dialog Class' text box contains 'VFXPICK'. 'OK', 'Apply', and 'Cancel' buttons are at the bottom.

Work on View. Wenn die Daten, aus denen Sie auswählen aus einer Ansicht stammen, markieren Sie dieses Kontrollkästchen.

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

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

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

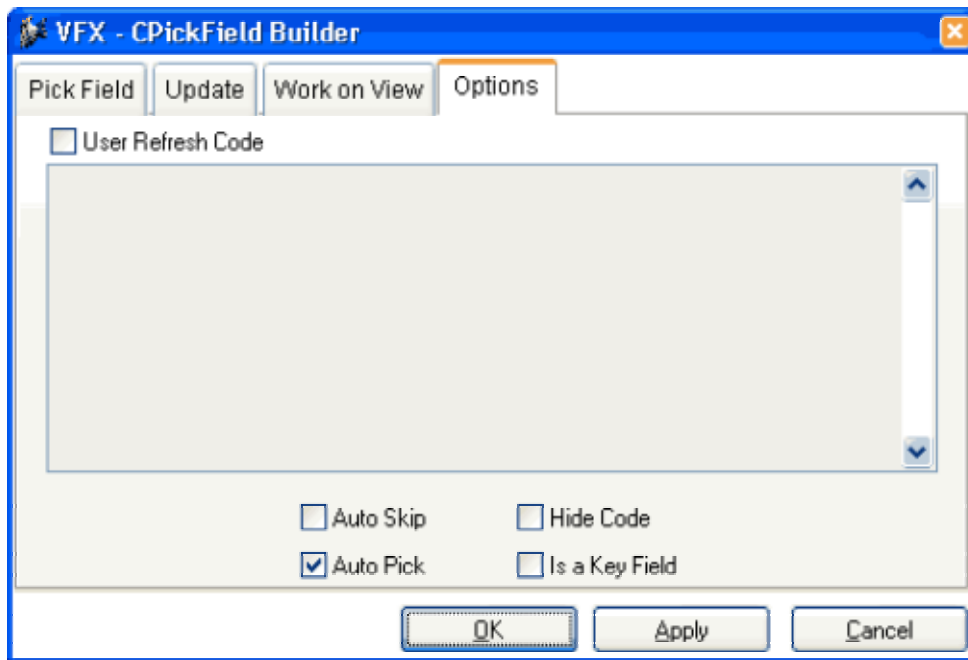
Pick Dialog Class: Hier kann eine eigene Klasse für das Auswahllisten-Steuerelement verwendet werden. Beachten Sie, dass die Klasse von der Klasse CPickField abgeleitet sein muss.

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

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

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

Auf der Seite *Options* stehen die folgenden Optionen zur Verfügung:



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

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

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

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

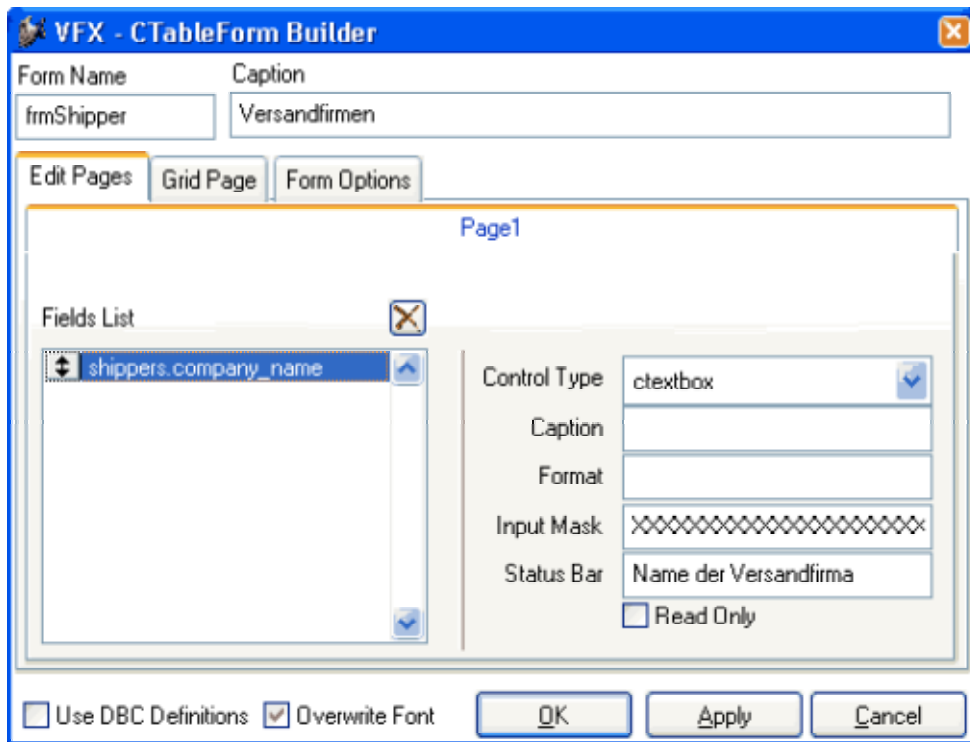
Is a Key Field. Markieren Sie diese Option, wenn Sie dieses Auswahllistenfeld als Schlüsselfeld definieren wollen. Ein Schlüsselfeld ist nur zugänglich während Sie einen neuen Datensatz anlegen (so wie die Textfeld-Klasse *ckeyfield*). Dadurch wird die *CPickField*-Eigenschaft *lKeyField* auf *.T.* gesetzt.

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

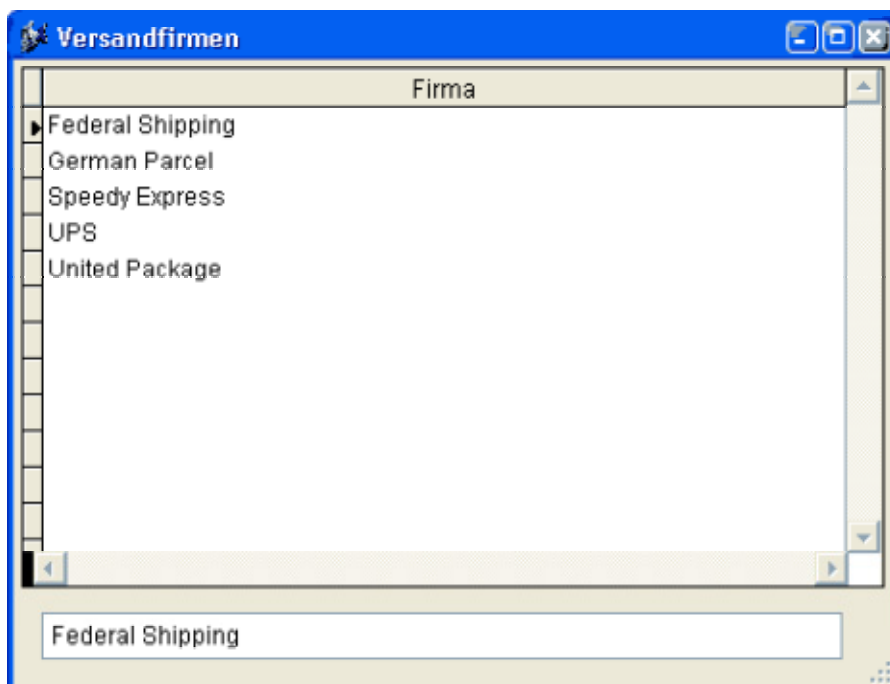
Apply. Macht das gleiche wie *OK*, jedoch wird der VFX – CPickField Builder nicht beendet.

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

Der VFX – CTableForm Builder



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



1:n Formulare

Das 1:n-Formular ist eine Weiterentwicklung des Standard-VFX-Datenbearbeitungs-Formulars. Das bedeutet, dass Sie auf einem einzigen Formular die normalen Datenbearbeitungsfunktionen haben können und ein Grid mit den Child-Datensätzen zu dem aktuell angezeigten Hauptdatensatz haben. VFX erlaubt es Ihnen, auch mehrere Child-Tabellen zu einer Haupttabelle auf mehreren Seiten eines Seitenrahmens zu bearbeiten. Wenn Sie viele Eingabefelder in Ihrer Child-Tabelle haben, können Sie die Felder auf mehrere Seiten eines Seitenrahmens verteilen. Das erlaubt Ihnen, eine große Anzahl verschiedenster Anwendungen abzudecken ohne wirklich programmieren zu müssen. Alles was Sie wissen müssen ist, wie man ein 1:n-Formular erstellt, die zugehörige Datenbank einrichtet und durch welche Felder die Haupttabelle und die Child-Tabelle miteinander verbunden sind. Lassen Sie uns ein einfaches Beispiel betrachten:

Ergebnis

Bitte lesen Sie im Kapitel *Diskussion des VFX-1:n-Datenbearbeitungs-Formulars* weiter unten in diesem Handbuch nach, um eine Vorstellung über die Bedienung von 1:n-Formularen zu bekommen, die mit VFX erstellt wurden.

Erstellen eines neuen Formulars

Starten Sie aus dem VFX-Menü den VFX – Form Wizard und erstellen Sie ein Formular basierend auf der Klasse cOneToMany.

Einrichten der Datenumgebung

Wie schon weiter oben in diesem Handbuch beschrieben, müssen Sie die Datenbank Ihrer Anwendung einrichten. Definieren Sie Ihre Tabellen, Felder und Indextypen sowie die Feldüberschriften. Die VFX-Builder benutzen diese Informationen, sodass Sie die Überschriften nicht nochmals eingeben müssen.

Bevor Sie ein 1:n-Formular erstellen, sollten Sie die Grundlagen des Datenbank-Designs und insbesondere 1:n-Beziehungen beherrschen. In 1:n-Beziehungen stellen Sie die Verbindung von einem Hauptdatensatz zu den Child-Datensätzen her. Ein gutes Beispiel für eine 1:n-Beziehung ist die Verbindung zwischen Aufträgen (Haupttabelle) und Auftragspositionen (Child-Tabelle) in jedem Auftragsbearbeitungssystem.

ANMERKUNG: Wenn Sie die referentielle Integrität (RI) nicht manuell mit Hilfe der VFX-Methoden wie OnPostDelete() herstellen wollen, ist es sinnvoll, den RI-Code im Datenbank-Designer anzulegen, bevor Sie mit der Erstellung von 1:n-Formularen beginnen. Wenn Sie diese Arbeit manuell erledigen wollen, müssen Sie den Code für das Löschen von Hauptdatensätzen und den zugehörigen Child-Datensätzen von Hand schreiben. Wenn Sie außerdem die Änderung des Schlüsselfeldes in der Haupttabelle erlauben, müssen Sie auch den Code schreiben, um die Child-Datensätze zu aktualisieren.

Richten Sie die Datenumgebung des Formulars ein, das Sie erstellen wollen. Der VFX – COneToMany Builder verwendet diese Informationen automatisch beim Erstellen des 1:n-Formulars.

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

WICHTIG: Denken Sie daran, den *InitialSelectedAlias* in der Datenumgebung anzugeben. Außerdem müssen Sie die 1:n-Beziehung zwischen der Haupttabelle und der Child-Tabelle herstellen. Ansonsten wird Ihr Formular nicht so funktionieren, wie Sie es erwarten!

Der VFX - COneToMany Builder

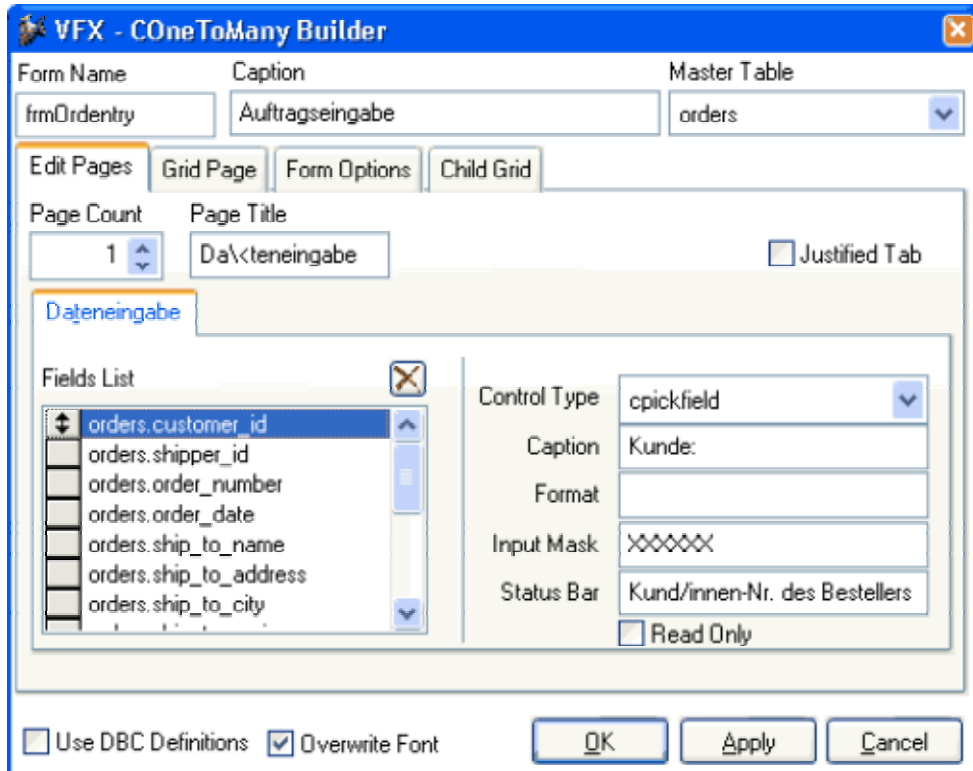
Aufruf des VFX – COneToMany Builder

Um den VFX-1:n-Formular-Builder aufzurufen, bewegen Sie die Maus auf den weißen Hintergrund des Formular-Designers, drücken Sie die rechte Maustaste und wählen Sie Builder.

ANMERKUNG: Wenn Sie eine Meldung erhalten, dass es keinen Builder für das ausgewählte Objekt gibt oder wenn ein Standard-Visual FoxPro-Builder erscheint, prüfen Sie, ob Sie das Formularobjekt ausgewählt haben. Ein verbreiteter Fehler ist, den Seitenrahmen anstelle des Formularobjektes auszuwählen. Prüfen Sie das angezeigte Objekt im Eigenschaftsfenster, wenn Sie nicht sicher sind, ob das Formularobjekt ausgewählt ist.

Die Bedienung des VFX - COneToMany Builder

Der VFX - COneToMany Builder hat eine intuitive Bedienung.



Bearbeiten Sie zunächst die folgenden Optionen:

Form Name. Siehe Beschreibung im Kapitel *Der VFX-Formular-Builder*.

Caption. Siehe Beschreibung im Kapitel *Der VFX-Formular-Builder*.

Master Table. Name der Haupttabelle oder Ansicht.

Als nächstes bearbeiten Sie den Seitenrahmen mit den Seiten *Edit Pages*, *Grid Page*, *Form Options* und *Child Grid*:

Auf der Seite mit dem Namen *Edit Pages* sehen Sie die gleichen Bedienungselemente wie im VFX-Formular-Builder, der weiter oben in diesem Handbuch beschrieben wurde. Hier legen Sie die Eigenschaften der Bearbeitungsseiten für die Haupttabelle fest:

Auf der Seite mit dem Namen *Grid Page* sehen Sie die gleichen Bedienungselemente wie im VFX-Formular-Builder, der weiter oben in diesem Handbuch beschrieben wurde. Hier beschreiben Sie die Eigenschaften des Grids für die Haupttabelle:

VFX - COneToMany Builder

Form Name: frmOrdentry Caption: Auftragseingabe Master Table: orders

Edit Pages **Grid Page** Form Options Child Grid

☒ Use Grid Page

Grid Page Title: \<Liste Grid Class: cgrid

Fields Selected:

- ☒ orders.order_id
- ☐ orders.order_date
- ☐ orders.deliver_by
- ☐ Calculated Field
- ☐ orders.paid

Control Type: textbox

Header: Auftragsnummer

Control Source: orders.order_id

Output Mask: \XXXXXX

☒ Read Only ☒ Incremental Search

☐ Use DBC Definitions ☒ Overwrite Font

OK Apply Cancel

Auf der Seite mit dem Namen *Form Options* sehen Sie die gleichen Bedienungselemente wie im VFX-Formular-Builders, der weiter oben in diesem Handbuch beschrieben wurde. Hier wählen Sie die Optionen für das 1:n-Formular:

VFX - COneToMany Builder

Form Name: frmOrdentry Caption: Auftragseingabe Master Table: orders

Edit Pages Grid Page **Form Options** Child Grid

Report Name: Arial,9,N

☒ Is Child Form ☒ Can Edit ☒ Save/Restore Positions

☐ Has More Functions ☒ Can Insert ☐ Add SpeedBar Control

☐ Has Linked Child Form ☐ Can Copy

☐ Auto Sync. Child Form ☒ Can Delete

☒ Put In Last File Menu ☒ Multi Instance

☒ Put In Window Menu ☒ Close with ESC Key ☐ **Work on View!**

☐ Use DBC Definitions ☒ Overwrite Font

OK Apply Cancel

Auf der Seite mit dem Namen *Child Grid* geben Sie an, wie das oder die Grids mit den Child-Daten aussehen sollen:

The screenshot shows the 'VFX - COneToMany Builder' dialog box with the 'Child Grid' tab selected. The 'Form Name' is 'frmOrdentry', 'Caption' is 'Auftragseingabe', and 'Master Table' is 'orders'. Under 'Page Count', '1' is selected. 'Page Title' is 'Page1'. 'Child Table' is 'order_line_items'. Checkboxes for 'Justified Tab', 'Inplace Editing', and '^Ins + ^Canc' are present, with the last two checked. The 'Fields Selected' list on the left includes 'products.product_name', 'order_line_items.quantity', 'order_line_items.unit_price', and 'Calculated Field 2'. On the right, 'Grid Class' is 'cchildgrid', 'Control Type' is 'combobox', 'Header' is 'Artikel', and 'Control Source' is 'products.product_name'. 'Output Mask' is empty and 'Read Only' is unchecked. At the bottom, 'Use DBC Definitions' is unchecked and 'Overwrite Font' is checked. 'OK', 'Apply', and 'Cancel' buttons are at the bottom right.

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

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

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

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

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

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

Die anderen Optionen sind mit denen auf der Grid-Seite des VFX-Formular-Builders identisch.

Der VFX – CChildGrid Builder

Der VFX – CChildGrid Builder erlaubt Ihnen, die Funktionalität der Child-Grids zu erweitern. Benutzen Sie diesen Builder, um die Felder für das Grid zusammenzustellen oder um den Code der Methode *OnPostInsert()* zu bearbeiten. Diese Methode wird immer dann ausgeführt, wenn dem Child-Grid ein neuer Datensatz hinzugefügt wurde. Ähnlich wie im Standard-VFX-Datenbearbeitungsformular stehen Ihnen hier die folgenden Ereignisse zur Verfügung:

- OnPreInsert()
- OnInsert()
- OnPostInsert()

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

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

Der VFX – CChildGrid Builder ist wie folgt zu bedienen. Auf der ersten Seite mit dem Namen *Grid* können Sie das Child-Grid, wie weiter oben in diesem Abschnitt beschrieben, anpassen:

The screenshot shows the 'VFX - CChildGrid Builder' dialog box with the 'Grid' tab selected. The 'Parent' dropdown is set to 'orders' and the 'Child' dropdown is set to 'order_line_items'. The 'Fields Selected' list on the left contains 'products.product_name', 'order_line_items.quantity', 'order_line_items.unit_price', and 'Calculated Field 2'. The 'Grid Class' is 'cchildgrid', 'Control Type' is 'combobox', 'Header' is 'Artikel', and 'Control Source' is 'products.product_name'. The 'Output Mask' is empty, and the 'Read Only' checkbox is unchecked. The 'Inplace Editing' and '^Ins + ^Canc' checkboxes are checked. At the bottom, there is a 'Use DBC Definitions' checkbox and 'OK', 'Apply', and 'Cancel' buttons.

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

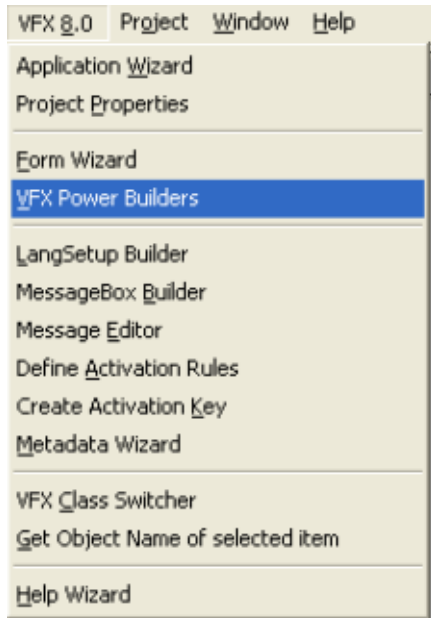
The screenshot shows the 'VFX - CChildGrid Builder' dialog box with the 'Options' tab selected. The 'CChildGrid::OnPostInsert()' method code is displayed in a text area. The code includes a wait statement for the window 'CChildGrid::OnPostInsert() is empty!', followed by a comment and a replace statement: '* replace order_id with orders.order_id in order_line_items'. The 'Use DBC Definitions' checkbox is unchecked, and 'OK', 'Apply', and 'Cancel' buttons are at the bottom.

ANMERKUNG: Der Grund, aus dem der VFX-Builder den Code der *OnPostInsert()*-Methode nicht automatisch generieren kann, ist, dass Sie zusammengesetzte Schlüssel verwenden könnten oder mehreren Feldern in der Child-Tabelle Werte zuweisen möchten. Wenn einfache Schlüssel verwendet werden, ist der generierte Code in der Regel richtig und Sie brauchen nur das Kommentarzeichen am Zeilenanfang zu entfernen.

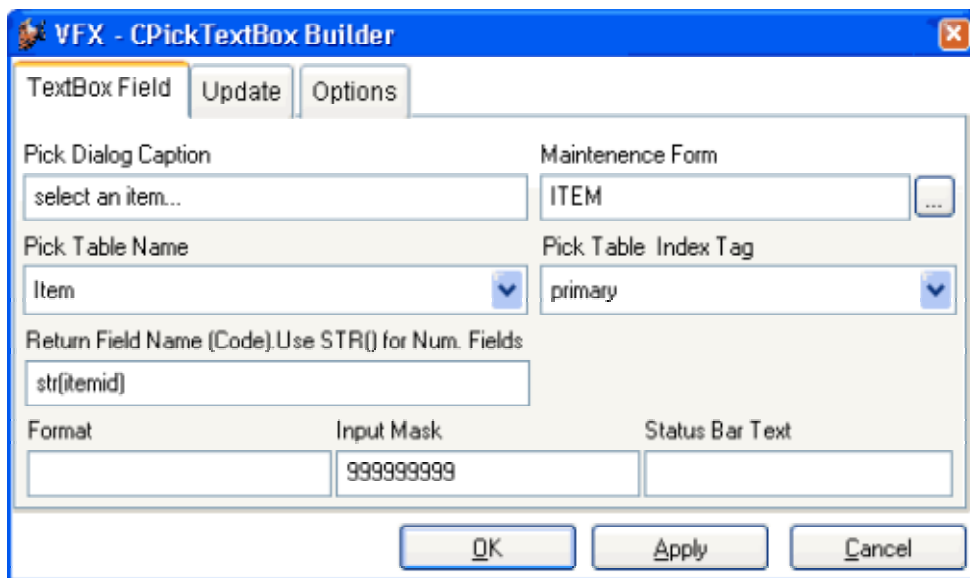
Der VFX – CPickTextBox Builder

Visual Extend bietet einen Builder, um leistungsfähige Auswahltextfelder zu erstellen. Die Auswahltextfelder können in Child-Grids verwendet werden.

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



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



The screenshot shows the 'VFX - CPickTextBox Builder' dialog box with the 'Update' tab selected. The 'Update Source Fields' section contains an empty text box. The 'Target Table Name' section has a dropdown menu currently showing 'Parent'. The 'Update Target Fields' section contains an empty text box. At the bottom are 'OK', 'Apply', and 'Cancel' buttons.

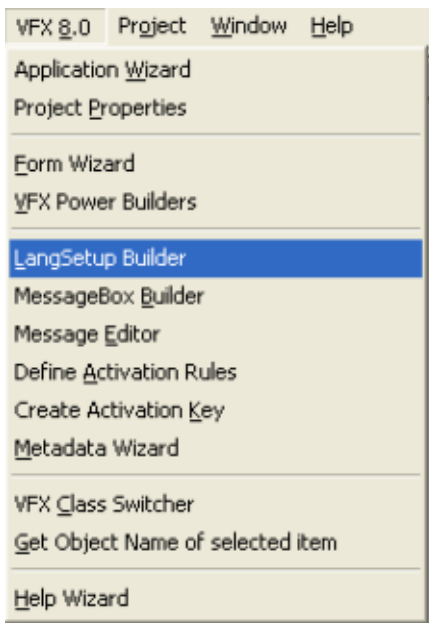
The screenshot shows the 'VFX - CPickTextBox Builder' dialog box with the 'Options' tab selected. There are two checkboxes: 'Work on View' and 'Is a Key Field', both of which are unchecked. Below them is a text box labeled 'Pick Dialog Class' containing the text 'VFXPICK'. At the bottom are 'OK', 'Apply', and 'Cancel' buttons.

Der VFX – LangSetup Builder

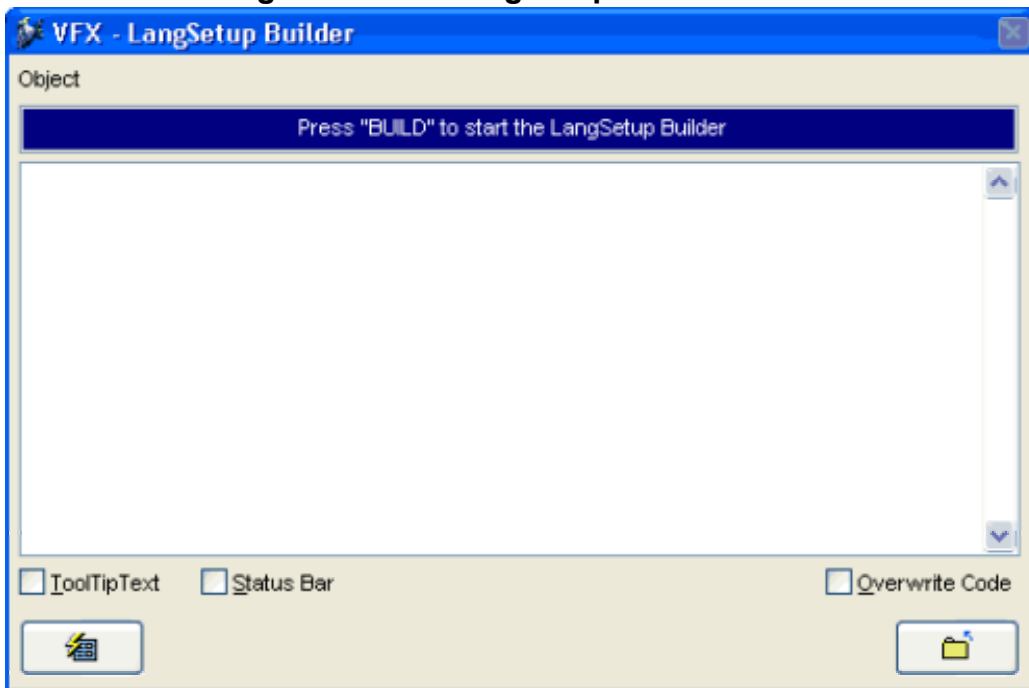
Der VFX – LangSetup Builder automatisiert die Erstellung des in der *LangSetup()*-Methode benötigten Codes. Sie brauchen diesen Code, wenn Sie Ihre Anwendung in mehr als einer Sprache erstellen wollen. Das Ziel dieses Builders ist es, aus dem Formular für alle Bezeichnungen, Tooltip-Texte und Statuszeilenmeldungen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

Aufruf des VFX – LangSetup Builders

Um den VFX – LangSetup Builder aufzurufen, öffnen Sie zunächst das Formular dessen Bezeichnungen, Tooltip-Texte und Statuszeilenmeldungen Sie analysieren lassen möchten. Wir könnten sagen alle für die Übersetzung in Frage kommenden Texte. Wählen Sie den folgenden Menüpunkt aus dem VFX-Menü:

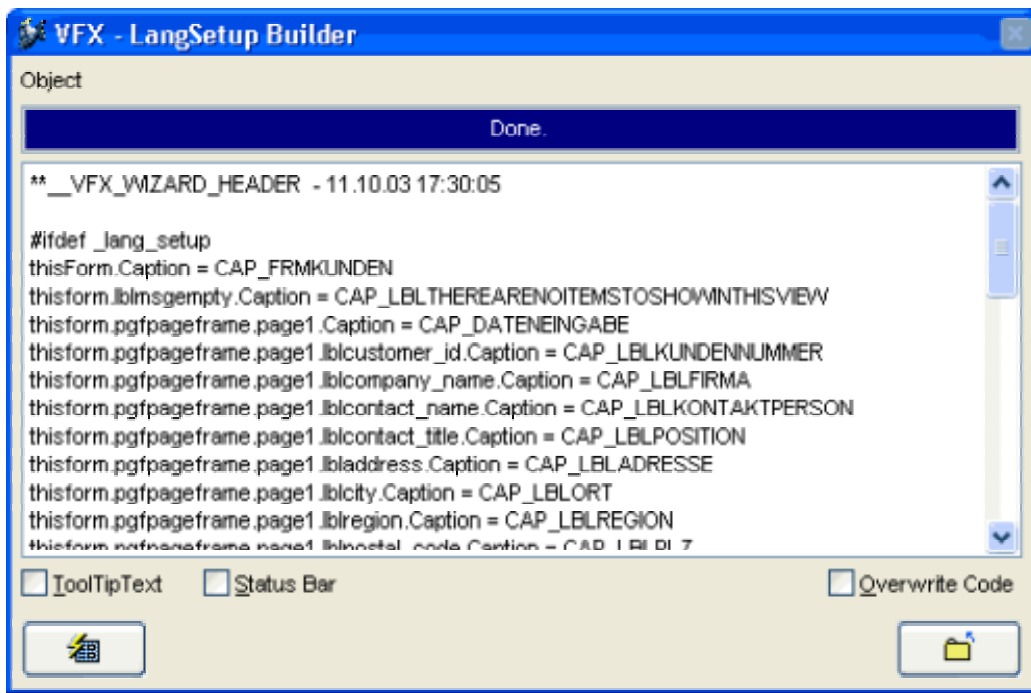


Die Bedienung des VFX – LangSetup Builders



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

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



Beachten Sie, dass die Konstanten automatisch in die Tabelle *VFXMSG.DBF* eingefügt werden, wenn Sie das Kontrollkästchen *Overwrite Code* markieren.

Define _Lang_Setup

In der Include-Datei *VFX.H* gibt die *_LANG_SETUP*-Konstante an, ob die *LangSetup()*-Methode ausgeführt wird. In der *LangSetup()*-Methode wird überprüft, ob diese Konstante existiert und falls ja, wird der Code der Methode ausgeführt. Dieses Verfahren dient der Geschwindigkeitsoptimierung für die Formulare.

```
#DEFINE _LANG_SETUP .T.
```

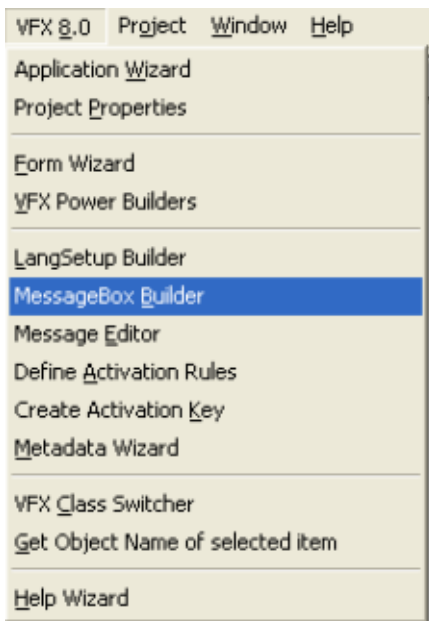
Der VFX – MessageBox Builder

Ziel

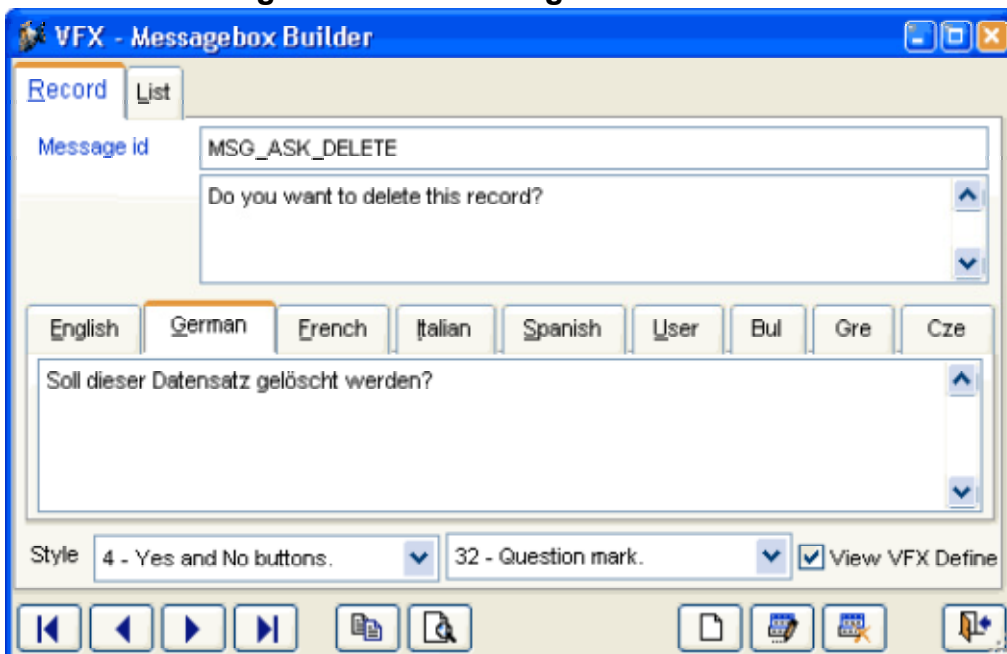
Der VFX – MessageBox Builder ist ein praktisches Werkzeug, um Meldungsdialoge (und WAIT WINDOWS) während der Entwicklung zu erstellen. Der VFX – MessageBox Builder hilft Ihnen nicht nur bei der Erstellung der Codezeile für den Meldungsdialog, sondern trägt die Texte auch gleich in die Tabelle *VFXMSG.DBF* ein, wo Sie diese weiter bearbeiten und in andere Sprachen übersetzen können. Der VFX – Message Editor wird weiter unten in diesem Handbuch beschrieben.

Aufruf des VFX – MessageBox Builder

Um den VFX – MessageBox Builder aufzurufen, wählen Sie den folgenden Menüpunkt aus dem VFX-Menü:



Die Bedienung des VFX – MessageBox Builder



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

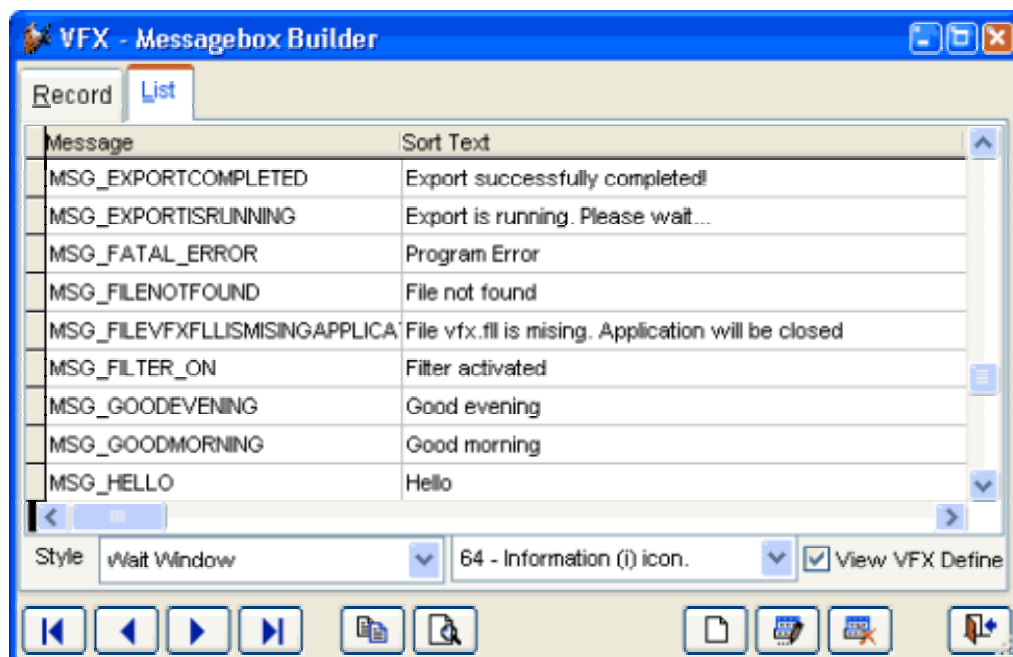
In der Zeile *Style* wählen Sie gewünschten Typ der MessageBox aus. Es kann zwischen verschiedenen Symbolen und Schaltflächen auf der MessageBox ausgewählt werden.

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

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

Der VFX – MessageBox Builder legt für jeden Eintrag einen Datensatz in der Tabelle *VFXMSG.DBF* an.

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



Tipp: Auch wenn Sie keine mehrsprachigen Anwendungen erstellen, können Sie den VFX – MessageBox Builder einsetzen.

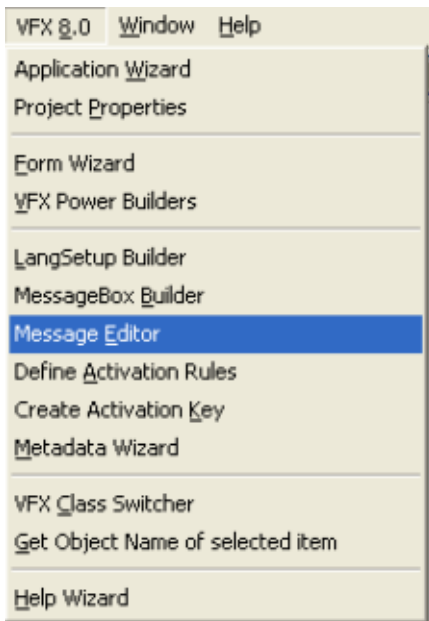
Der VFX – Message Editor

Ziel

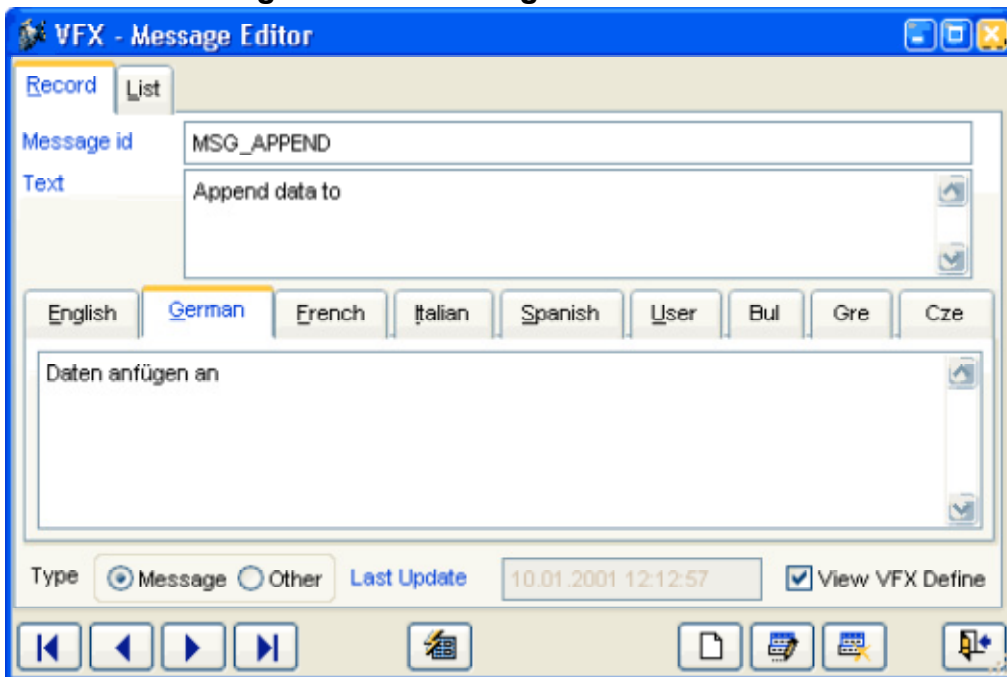
Der VFX – Message Editor ist der Zentrale Ort um, alle Bezeichnungen, Meldungen, Tooltip-Texte und Statuszeilenmeldungen zu verwalten und in andere Sprachen zu übersetzen. Aus dem VFX – Message Editor heraus können Sie die benötigten Include-Dateien (*USERTEXT.H* und *USERMSG.H*) erstellen.

Aufruf des VFX – Message Editor

Um den VFX – Message Editor aufzurufen, wählen Sie den folgenden Menüpunkt aus dem VFX-Menü:



Die Bedienung des VFX-Message-Editor



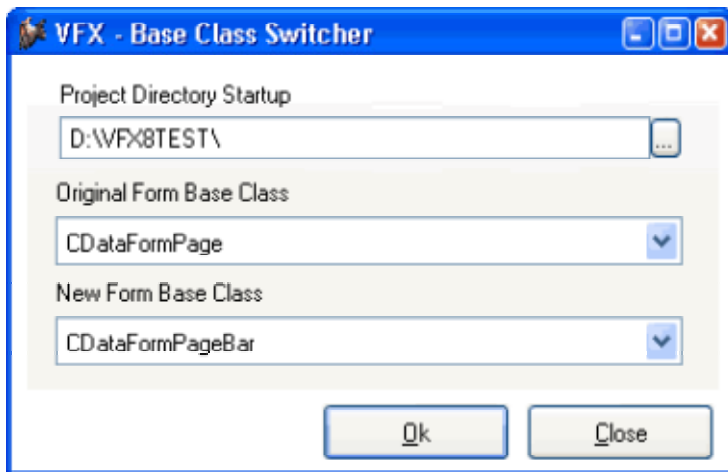
Klicken Sie auf die Schaltfläche *Make Include File* um eine Include-Datei in der im Seitenrahmen angezeigten Sprache zu erstellen. Die Include-Dateien werden in einem Ordner mit der Bezeichnung der jeweiligen Sprache unterhalb des Include-Ordners Ihres Projektes gespeichert.

Nach der Erstellung Ihrer Include-Dateien müssen Sie diese nur noch in den *INCLUDE*-Ordner Ihres Projektes kopieren, wie im Kapitel *Erstellen mehrsprachiger Anwendungen* beschrieben ist. **Tipp:** Sie können Ihre eigenen Konstanten mit den erzeugten Konstanten aus der Tabelle *VFXMSG.DBF* mischen. Schreiben Sie Ihre Konstanten vor oder nach dem VFX-Header bzw. -Footer.

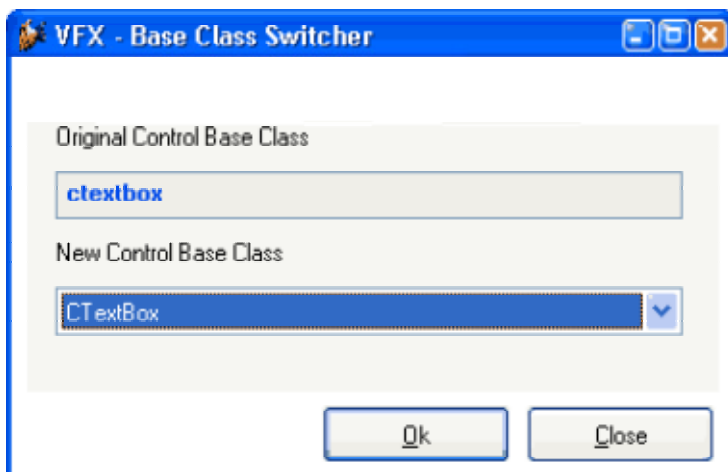
Der VFX – Class Switcher

Der Class Switcher hat zwei Funktionen.

Wenn beim Aufruf kein Formular geöffnet ist, ändert der Class Switcher die Klassen von Formularen in einem ganzen Projekt. Zum Beispiel kann die Formulkasse CDataFormPageBar durch CDataFormPage ersetzt werden. Dadurch ist es möglich alle Formulare mit Schaltflächen auszustatten bzw. diese wieder zu entfernen. Als besonders hilfreich erweist sich dieses Werkzeug bei der Aktualisierung vorhandener VFX 3-Projekte. In VFX 3 hatte jedes Formular am unteren Rand eine Leiste mit Schaltflächen. In VFX 8.0 kann man stattdessen eine richtige Symbolleiste verwenden.



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



Eigenschaften der erstellten Formulare

Die mit den VFX – Form Buildern erstellten Formulare haben standardmäßig viele gute Eigenschaften. Die Position des Formulars auf dem Bildschirm, die Größe des Formulars (die Größe eines Formulars kann mithilfe eines Resizers vom Benutzer zur Laufzeit eingestellt werden), die zuletzt aktive Seite des Seitenrahmens sowie die Einstellungen des Datenrasters, Sortierfolge, Spaltenbreiten, werden für jeden Benutzer individuell gespeichert. Schließt ein Benutzer ein Formular und öffnet er es wieder, erscheint es genauso, wie er es verlassen hat.

Formularbedienung

Die Standardbedienung für ein Standard-Datenbearbeitungsformular sieht wie folgt aus, wenn Sie sich nicht im Bearbeitungsmodus oder im Einfügemodus befinden:

Wenn Sie sich im Einfüge- oder Bearbeitungsmodus befinden, ändert sich die Überschrift des Formulars und die Schaltflächen der Symbolleiste werden entsprechend aktualisiert.

ANMERKUNG: Um große Datenmengen einzugeben, können Sie die Tastenkombination *Strg+N* drücken, auch wenn Sie sich bereits im Einfügemodus befinden. Dadurch ist es sehr schnell, mehrere Datensätze nacheinander zu erfassen. Aus den gleichen Optimierungsgründen bleiben die Navigations-Schaltflächen auch während der Bearbeitung aktiv.

Entsprechend der Einstellung der Eigenschaft *nAutoEdit* im Anwendungsobjekt bzw. der Formulareigenschaft *AutoEdit* kann der Benutzer einfach mit der Bearbeitung beginnen und das Formular wechselt automatisch in den Bearbeitungsmodus, wie hier gezeigt wird:

Bearbeiten - Mitarbeiter

Tab: **Dateneingabe** | Zusatzinformation | Liste

Nachname:

Vorname:

Position:

Geburtstag:

Eingestellt am:

Adresse:

Ort: Telefon privat:

Region: Durchwahl:

PLZ: Gruppe:

Land:

Die Schaltflächen der Symbolleiste sowie die Menüeinträge werden entsprechend dem Formularstatus aktiviert.

Das VFX Power Grid

In allen Spalten eines Grids ist standardmäßig eine inkrementelle Suche möglich. Durch einen Doppelklick auf eine Überschrift in einem Datenraster kann die entsprechende Spalte sortiert werden. Wenn für die Spalte kein geeigneter Index vorhanden ist, wird von VFX automatisch ein temporärer Index angelegt. Soll die Suche um eine zusätzliche Spalte erweitert werden, drückt man die Taste „Strg“ und klickt gleichzeitig auf eine weitere Überschrift. Die Rangfolge der Sortierung wird in den Überschriften durch Zahlen in Klammern dargestellt.

Mitarbeiter

Tab: Dateneingabe | Zusatzinformation | **Liste**

Nachname	Vorname	Position	Adresse	Ort
Fuller	Andrew	Geschäftsführer	908 W. Capital W	Tacoma
Hellstern	Albert	Geschäftsführer	13920 S.E. 40th S	Bellevue
Martin	Xavier	Marketingassistent	9 place de la Libe	Schiltigheim
Brid	Justin	Marketingdirektor	2 impasse du Sol	Haguenau
Patterson	Caroline	Sekretärin	16 Maple Lane	Auburn
Callahan	Laura	Verkaufskoordinator	4726 - 11th Ave. N	Seattle
Buchanan	Steven	Verkaufsleiter	14 Garrett Hill	London
Davolio	Nancy	Verkaufsrepräsentant	507 - 20th Ave. E.,	Seattle
Dodsworth	Anne	Verkaufsrepräsentant	7 Houndstooth Rd	London
King	Robert	Verkaufsrepräsentant	Edgeham Hollow,	London
Leverling	Janet	Verkaufsrepräsentant	722 Moss Bay Blv	Kirkland
Peacock	Margaret	Verkaufsrepräsentant	4110 Old Redmor	Redmond
Suyama	Michael	Verkaufsrepräsentant	Coventry House, M	London
Smith	Tim	Versandgehilfe	30301 - 166th Ave	Kent

Ein Doppelklick auf eine Überschrift sortiert eine Spalte. Ein weiterer Doppelklick kehrt die Sortierfolge um.

Nach einem Klick in eine Spalte kann mit der Eingabe eines Suchbegriffs begonnen werden. Die Sortierfolge wird auf diese Spalte umgestellt und der eingegebene Begriff wird inkrementell gesucht. Der eingegebene Begriff wird in der Statuszeile angezeigt:

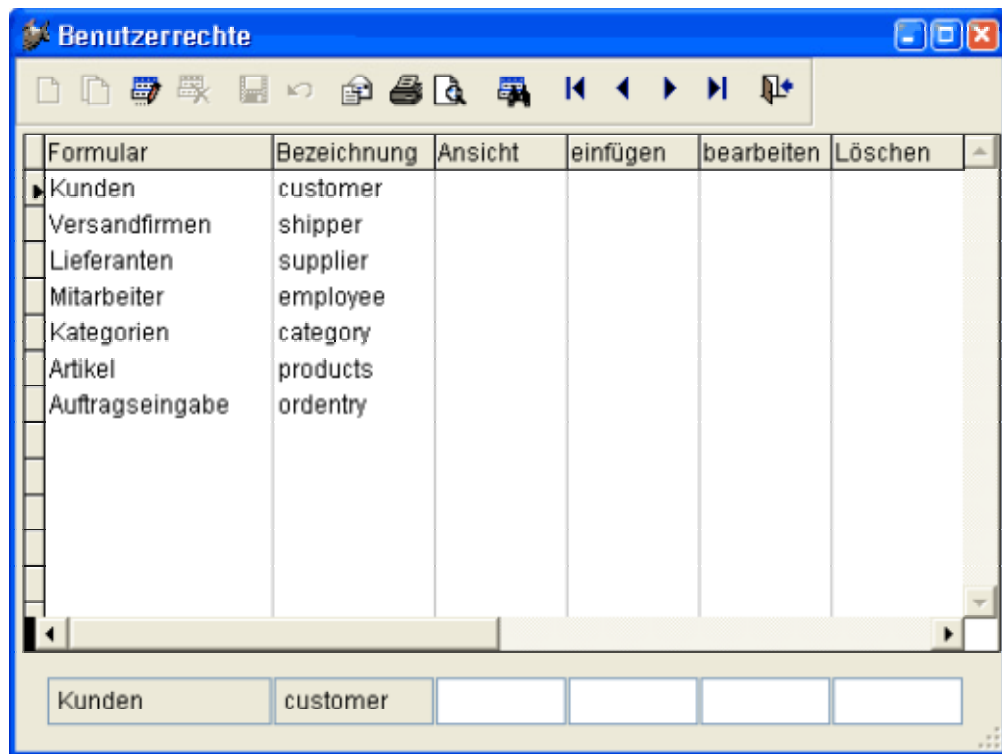
Suche : Martin

Inkrementelle Suche

Benutzen Sie den VFX CGrid Builder, um einzustellen für welche Spalten die inkrementelle Suche verwendet werden soll. Dadurch erhält der Benutzer die Möglichkeit, durch einfaches Eingeben eines Zeichens, einer Zahl oder auch eines Datums die inkrementelle Suche einzuleiten. Dabei wird die Sortierfolge automatisch umgestellt und es wird auf den der Eingabe entsprechenden Eintrag gesprungen. Während der inkrementellen Suche, wird der Suchbegriff in der Statuszeile angezeigt. Korrekturen können mit der Rückschritttaste durchgeführt werden.

Formulare basierend auf der Klasse CTableForm

Bei Formularen basierend auf der Klasse CTableForm sind das Such-Grid und andere Steuerelemente nebeneinander oder untereinander auf einem Container angeordnet. Ein typisches CTableForm-Formular ist die Verwaltung der Benutzerrechte.



Diskussion des VFX-1:n-Datenbearbeitungs-Formulars

Auftragseingabe

Dateneingabe | Liste

Kunde: CACTU ... Cactus Comidas para llevar

Auftragsnummer: 2

Name: Mère Paillarde

Auftragsdatum: 12.05.1992

Adresse: 43 rue St. Laurent

Ort: Montréal

PLZ: H1J 1C3

Region: Québec

Land: Kanada

Lieferinformationen

Speedy Express

Fällig: 09.06.1997

Notizen:

Kreditrahmen: 10 % Rabatt: 1.962,09

-12.228,3 Bezahl: ☐ Versandkosten: 79,45

Zwischensumme: 19.620,90

Rechnungsbetrag: 17.738,26

Artikel	Menge	Einzelpreis	Gesamtpreis
Boston Crab Meat	998,000	18,4000	18363,2000
Raclette Courdavault	24,000	38,5500	925,2000
Wimmers gute Semmelknödel	10,000	33,2500	332,5000

Bearbeiten der Haupttabelle

Die Bearbeitung der Daten der Haupttabelle ist identisch mit der im Standard-Datenbearbeitungs-Formular. Die Symbolleiste und das Menü *Bearbeiten* beziehen sich auf die Haupttabelle.

Bearbeiten der Child-Tabelle

Die Child-Datensätze werden im unteren Grid bearbeitet. Nur wenn Sie sich im Bearbeitungs- oder Einfügemodus der Haupttabelle befinden, können Sie auch das Child-Grid bearbeiten, Child-Datensätze einfügen und löschen. Alle Bearbeitungen der Child-Datensätze werden mit optimistischer Tabellenpufferung durchgeführt. Wenn Sie sich entscheiden, Ihre Änderungen rückgängig zu machen, werden die Änderungen in allen Child-Datensätzen rückgängig gemacht. Wenn Sie sich entscheiden, die Änderungen zu speichern, werden alle Änderungen an der Haupttabelle und in allen Child-Datensätzen gespeichert.

Auswahlliste innerhalb eines Child-Grids

Eine der interessantesten Funktionen von VFX ist die besondere Auswahlliste, die Sie Ihrem Child-Grid mit dem VFX – CPickTextBox Builder hinzufügen können. Die Auswahllisten können im Bearbeitungs- und im Einfügemodus erreicht werden.

Durch einen Doppelklick in die CPickTextBox oder durch drücken der Funktionstaste F9 wird die Auswahlliste angezeigt.

Drucken

Aus allen Formularen kann standardmäßig eine Liste gedruckt werden, ohne dass dafür Berichte angelegt werden müssen. VFX legt zur Laufzeit der Anwendung temporäre Berichtsdateien an, die auf der Ansicht der Suchseite eines Formulars basieren.

Bericht

Optionen | **Zusatzoptionen**

Titel: Kunden

Zeichensatz: Courier New, 20, BI

Detail-Titelzeichensatz: Times New Roma, 8, BI

Detail-Zeichensatz: Courier New, 8, N

Druckoptionen

☒ Seitenansicht
☐ Drucker
☐ E-Mail
☐ PDF-Export

☒ Hochformat
☐ Querformat

☒ Seitennummer
☐ nicht auf erster Seite
☒ Datum
☒ Zeit

OK | Abbrechen

Vor dem Druck bzw. der Seitenansicht kann der Benutzer nicht gewünschte Spalten aus der Liste entfernen. Die Breite der Spalten entspricht ungefähr der Breite der Spalte im Grid.

Bericht

Optionen | **Zusatzoptionen**

Markierung aufheben | Alles Auswählen

Feld	Auswahl	Summieren
Nummer	☒	☐
Name	☒	☐
Kontaktperson	☒	☐
Titel der Kontaktperson	☒	☐
Adresse	☒	☐
Ort	☒	☐
Region	☒	☐
PLZ	☒	☐
Land	☒	☐

OK | Abbrechen

Filtern

Der sichtbare Datenbereich in einem Formular kann durch Setzen eines Filters eingeschränkt werden. VFX stellt dafür einen fertigen Dialog zur Verfügung. Beliebig viele Felder können dabei mit „und“ oder „oder“ verknüpft werden.

Feld	Operator	Wert
Name	Gleich	Alfred
Max. Auftragswert	Größer als	5000

Weitere Entwicklungstechniken

Verändern von Eigenschaften des Applikationsobjektes

Im Hauptprogramm Vfxmain.prg wird programmatisch eine Ableitung der Klasse des Applikationsobjektes erstellt. Hier ist es also möglich den Code der VFX-Methoden zu ändern und Eigenschaften einzustellen ohne an den Klassenbibliotheken Veränderungen vornehmen zu müssen.

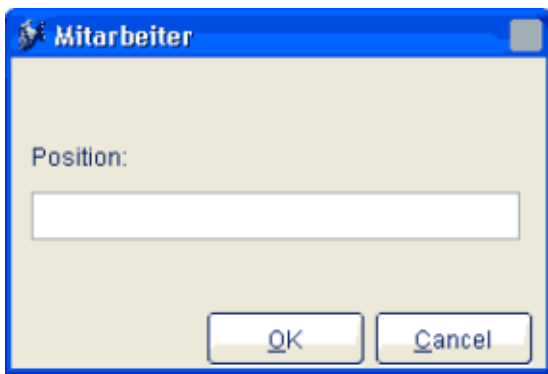
Formulare basierend auf Ansichten

Bei der Entwicklung von VFX wurde großer Wert darauf gelegt, dass sowohl direkt mit VFP-Tabellen, als auch mit lokalen Ansichten und mit Remote Ansichten gearbeitet werden kann. Wenn die Datenquelle eines Formulars eine Ansicht sein soll, muss auf der Seite Optionen des VFX - Form Builder das Häkchen bei „Work On View“ gesetzt werden. Damit weiß VFX, dass es sich bei der Datenquelle um eine Ansicht handelt. Ansichten können insbesondere keine Indexschlüssel haben. VFX muss also in jedem Fall, in dem eine Sortierung benötigt wird, eine temporäre Indexdatei erstellen.

Eingabe der Ansichtsparameter – CAskViewArg

In den meisten Fällen sind Ansichten parametrisiert. Die Parameter müssen vor Abfrage der Daten der Ansicht bekannt sein. Zur Eingabe der Ansichtsparameter stellt VFX die Formulkasse *CAskViewArg* zur Verfügung. Das Datenbearbeitungsformular wird wie gewohnt mit dem VFX - Form Builder erstellt. Die Eigenschaft *lworkonview* wird auf .T. gesetzt. Bei der Ansicht in der Datenumgebung wird die Eigenschaft *nodataonload* auf .T. gesetzt. Das bedeutet, dass die Ansicht beim Laden des Formulars geöffnet wird, ohne dass Daten abgefragt werden.

Jetzt wird ein neues Formular basierend auf der Klasse *CAskViewArg* erstellt. Die Steuerelemente, die als Controlsource Felder enthalten, die auch als Ansichtsparameter verwendet werden, können über die Zwischenablage vom Bearbeitungsformular auf das Formular basierend auf der Klasse *CAskViewArg* kopiert werden. In der Eigenschaft *cviewparameter* ist der Name des Ansichtsparameters einzutragen. Den Steuerelementen können geeignete Bezeichnungen hinzugefügt werden. Das Formular ist damit fertig und kann gespeichert werden.



Aus dem Bearbeitungsformular muss nun noch das Formular basierend auf der Klasse *CAskViewArg* aufgerufen werden. Dies geschieht am Ende des Init-Events:

```
do form <Formular zur Eingabe der Ansichtsparameter> with this
```

Es ist auch möglich zur Laufzeit des Formulars das Formular zur Eingabe der Ansichtsparameter erneut aufzurufen. Wenn der Aufruf aus einem Steuerelement, zum Beispiel aus dem Click-Event einer Schaltfläche erfolgt, muss der Aufruf so aussehen:

```
do form <Formular zur Eingabe der Ansichtsparameter> with thisform
```

Mehr ist bei der Arbeit mit Ansichten nicht zu beachten. Alles Weitere erledigt VFX.

Linked Child-Formulare

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

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

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

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

Der VFX-Entwickler muss dazu im Child-Formular mit dem Form Builder auf der Seite Optionen „Is Child Form“ auswählen oder manuell die Eigenschaft *lchildform* auf *.T.* zu setzen.

Beim Parent-Formular müssen mit dem Form Builder die Optionen „Has More Options“ (setzt die Eigenschaft *lmore* auf *.T.*), „Has Child Form“ und „Auto Sync Child Form“ (setzt die Eigenschaft *lautosynchildform* auf *.T.*) ausgewählt werden. Der Form Builder trägt automatisch Template-Code in die Methoden *onmore* und *onsetchilddata* ein. Der Code dieser Methoden muss anschließend manuell bearbeitet werden. In der Methode *onmore* wird das Child-Formular aufgerufen.

Erstellen eines Formulars, das ein Child-Formular aufruft

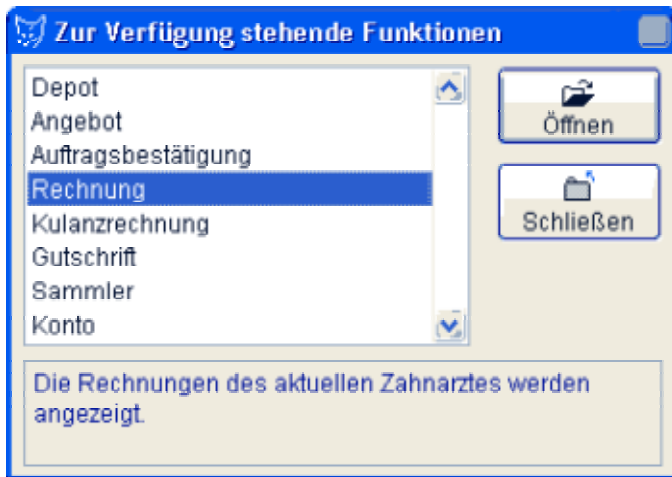
Obwohl es einen speziellen VFX-Builder zur Erstellung von 1:n-Formularen gibt, ist es manchmal besser, Child-Daten in einem eigenen Formular zu bearbeiten. Das ist insbesondere dann der Fall, wenn Sie das Child-Formular auch für die direkte Bearbeitung einsetzen und nicht nur durch das Hauptformular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

Im Abschnitt über den VFX-Formular-Builder haben wir bereits das Kontrollkästchen mit dem Namen *Has More Functions* betrachtet. Wenn Sie dieses Kontrollkästchen markieren, generiert der VFX-Formular-Builder einen Template-Code in der *OnMore()*-Methode des Formulars.

Wenn der Benutzer die verfügbaren Optionen zum aktuellen Datensatz sehen will, hat er verschiedene Möglichkeiten:

- Er kann die Funktionstaste *F6* drücken.
- Er wählt *Weitere Funktionen...* im *Bearbeiten*-Menü.
- Er drückt auf die *Weitere Funktionen*-Schaltfläche in der Standard-Symbolleiste.

Abhängig von dem Code in der Methode *OnMore()* wird der Benutzer einen Dialog sehen, der so ähnlich aussieht wie der folgende:



Der Aufruf der *OnMore()*-Methode mit dem Parameter *tnPassThrough* ist sehr nützlich, wenn Sie ein Formular direkt über die zugeordnete Zahl starten wollen. Sie können diese Technik benutzen, um ein Formular aus der *OnMore()*-Methode über eine Schaltfläche aus einer Symbolleiste zu starten.

Wenn es nur eine Option in der *OnMore()*-Methode gibt, wird das zugeordnete Formular geöffnet, ohne dass dieser Dialog erscheint.

Erstellen eines Child-Formulars

Das Gegenstück eines Formulars, das ein anderes Formular aufruft, ist das aufgerufene Formular. Wie in einem vorangegangenen Kapitel beschrieben, kann es verschiedene Gründe geben, aus denen ein Formular von einem anderen Formular aufgerufen wird.

Wenn Sie ein Formular aufrufen, übergeben Sie die benötigten Parameter an die *Init()*-Methode dieses Formulars. Da die übergebenen Parameter nicht automatisch für andere Methoden des gleichen Formulars sichtbar sind, speichern VFX-Formulare die benötigten Parameter in speziellen Eigenschaften.

Delayed Instantiation

Die Ladezeit eines Formulars hängt im Wesentlichen von der Anzahl der Steuerelemente ab, die mit dem Formular geladen werden müssen. Nun sind aber in der Regel nicht alle Steuerelemente eines Formulars sofort sichtbar, wenn ein Formular gestartet wird. Wenn mit einem Seitenrahmen gearbeitet wird, sind zunächst nur die Steuerelemente einer Seite sichtbar. Die Steuerelemente der anderen, zunächst nicht sichtbaren Seiten, brauchen also gar nicht geladen werden. Erst wenn der Benutzer erstmals eine andere Seite aktiviert, müssen die auf dieser Seite befindlichen Steuerelemente nachgeladen werden.

Die Delayed Instantiation wird von VFX mit der sehr praktischen Funktion *addpagedelay()* unterstützt.

Um das Ziel zu erreichen müssen zunächst alle Steuerelemente einer Seite eines Pageframes in einem Container als Klasse gespeichert werden. Dafür markiert man im VFP Formular-Designer alle Steuerelemente der aktuellen Seite und wählt im Menü *File* den Punkt *„Save As Class“*. Die Klasse sollte in der Klassenbibliothek *Appl.vcx* gespeichert werden. Diese Klassenbibliothek steht dem Entwickler für eigene Klassen zur Verfügung. Beim Speichern als Klasse ergänzt VFP automatisch einen Container um die ausgewählten Steuerelemente. Der Name der Klasse sollte so gewählt werden, dass der Bezug zu dem Formular und der Seite des Pageframes leicht

ersichtlich sind. Die als Klasse gespeicherten Steuerelemente können nun von dem Seitenrahmen gelöscht werden.

Um den Container zur Laufzeit des Formulars nachzuladen wird die Funktion `addpagedelay()` verwendet. Der Aufruf muss in das Activate Event der jeweiligen Seite eingefügt werden und sieht so aus:

```
AddPageDelay(thisform, this, 'x', '<Name der Klasse>')
```

Es empfiehlt sich ein Formular zunächst ohne Delayed Instantiation zu entwickeln und zu testen. Wenn das Formular fast fertig ist, kann es auf Delayed Instantiation umgestellt werden. Zu beachten ist dabei, dass Referenzen auf einzelne Steuerelemente geändert werden müssen. Während vor der Umstellung auf Delayed Instantiation auf eine Textbox zum Beispiel so referenziert werden konnte:

```
Thisform.pgfPageframe.Page1.txtMeinetextbox
```

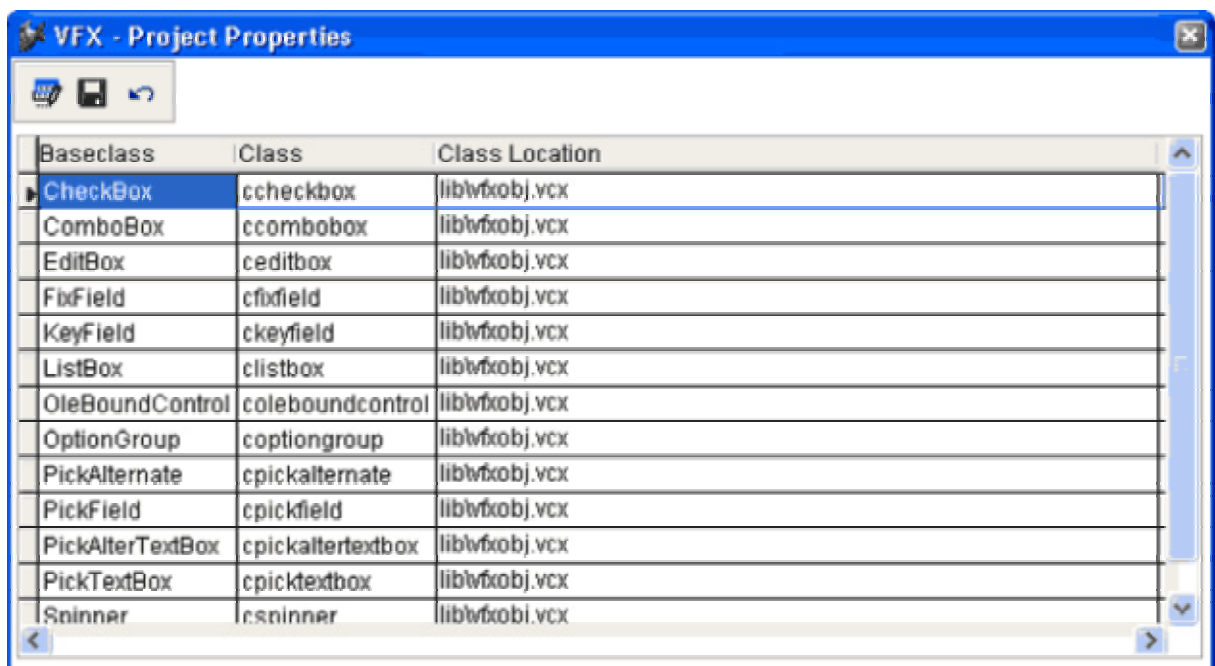
Sieht die Referenz nach Umstellung auf Delayed Instantiation so aus:

```
Thisform.pgfPageframe.Page1.x.txtMeinetextbox
```

Das x ist hierbei der Name des Containers, in dem sich die Steuerelemente der Seite befinden.

VFX – Project Properties

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



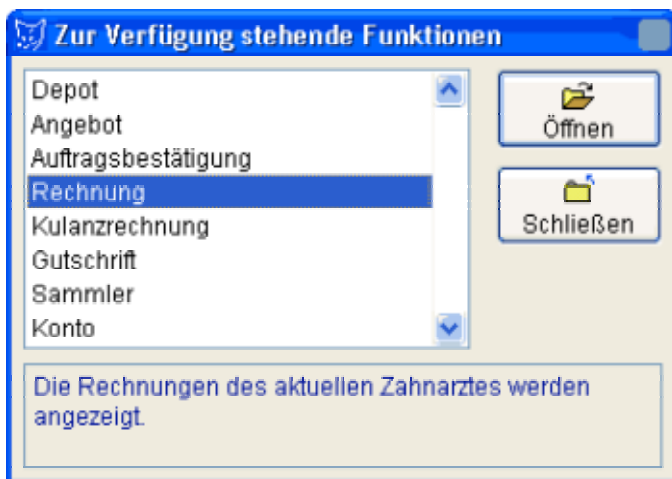
Wichtige VFX-Methoden

Valid

VFX bietet eine Valid-Methode auf Formularebene. Diese Methode wird immer aufgerufen, wenn die Daten des Formulars gespeichert werden sollen. Hier sollten also alle Validierungen untergebracht werden. Wenn aus dieser Methode der Wert `.F.` zurückgegeben wird, wird der Speichervorgang nicht fortgesetzt und das Formular bleibt im Bearbeitungsmodus. Durch Rückgabe von `.T.` werden die Daten gespeichert.

Onmore

Über eine Formulareigenschaft (`lMore`) kann die Schaltfläche „weitere Funktionen“ in der Standard-Symbolleiste aktiviert werden. In der Click-Methode dieser Schaltfläche wird die `OnMore`-Methode des aktiven Formulars aufgerufen. In dieser Methode steht bereits ein Template-Code, der leicht verändert werden kann. Hier werden in einem Array die Parameter für das VFXMore-Formular aufgerufen in dem in einem Dialog zwischen den zur Verfügung stehenden Funktionen ausgewählt werden kann. Z. B. können Child-Formulare gestartet werden.



Onpostinsert

Diese Methode wird unmittelbar nach dem Anfügen eines neuen Datensatzes aufgerufen, noch bevor der Benutzer die Möglichkeit zur Bearbeitung der Daten erhält.

Hier können also Standardvorgaben in den Feldern eingetragen werden. Diese Methode bietet sich auch an, um Primärschlüssel zu vergeben.

Onrecordmove

Jedes Mal, wenn der Satzzeiger bewegt wird, wird diese Methode aufgerufen. Hier können Werte angezeigt oder aktualisiert werden, die nicht aus der Datenbank stammen.

VFX Primärschlüssel-Generierung

Es kann Tabellen geben, aus denen Sie den Primärschlüssel nicht den Benutzern zeigen wollen. Aber für ein korrektes Datenbankdesign wollen Sie einen Primärschlüssel verwenden. Für diese und ähnliche Situationen bietet VFX eine Funktion, die die Erstellung von Primärschlüsseln ermöglicht und in einer Mehrbenutzerumgebung genauso funktioniert, wie in einer Client/Server-Umgebung.

Durch das modulare Design der VFX-Klassenhierarchie, haben Sie die Möglichkeit, nach dem Einfügen eines neuen Datensatzes einzugreifen. VFX bietet, neben vielen anderen Funktionen, eine Methode mit dem Namen `OnPostInsert()`, die in dem Moment ausgeführt wird, wenn ein neuer Datensatz gerade hinzugefügt wurde. Normalerweise bietet VFX für alle wichtigen Ereignisse Methoden, die automatisch vor, während und nach dem

Ereignis ausgeführt werden. In diesem Fall, in dem ein neuer Datensatz hinzugefügt wird, gibt es die folgenden Methoden:

- OnPreInsert()
- OnInsert()
- OnPostInsert()

Außerdem gibt es eine Eigenschaft die angibt, ob der Benutzer einen neuen Datensatz aufnehmen kann. Diese Eigenschaft trägt den Namen *ICanInsert*.

Um einen Primärschlüssel zu erzeugen, könnten Sie in die *OnPostInsert()*-Methode Ihres Formulars etwa folgenden Code einfügen. Hierdurch wird die Funktion *GetNewId()* aufgerufen. Der Parameter gibt die Tabelle an, für die der Schlüssel generiert wird.

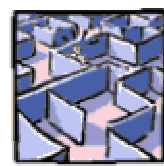
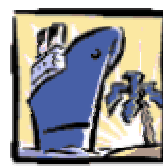
```
DODEFAULT()  
REPLACE comp_id WITH GetNewId('COMP_ID') IN customer
```

Der Zähler für den generierten Schlüssel wird in der Tabelle *VFXSYSID* gespeichert.

Active Desktop

Der Active Desktop gibt den Anwendungen ein professionelles Startbild. Auf dem sonst leeren Bildschirm werden Bilder und Auswahlmöglichkeiten angeboten. Durch das Bewegen der Maus über die Bilder wird das zugehörige Menü unterhalb der Bilder angezeigt. In den Menüs befinden sich unterstrichene Menüpunkte, die ähnlich Hyperlinks im Internet Explorer, einfach angeklickt werden können und eine Aktion ausführen. In den meisten Fällen wird als Aktion ein Formular gestartet werden.

Die Klasse des Active Desktop befindet sich in der Klassenbibliothek *Appl.vcx* und kann nach den Wünschen des Entwicklers um beliebige Steuerelemente erweitert werden.



Simple

Parent	Parent form which acts as parent form in a linked child scenario plus more...
Child	The same child form, just called directly, why not...
Item	Item table, shows the cTableForm class, very handy...
OneToMany	OneToMany form with parent -> child, almost a classic...
OneToMany2	OneToMany form item -> child, you are flexible, aren't you...
ParentTree	Parent Tree form shows the cTreeView class
OneToTree	Shows the cTreeViewOneToMany class

Der Active Desktop kann zusätzlich oder anstelle des Öffnen-Dialogs verwendet werden.

Benutzung des VFX-Moverdialogs

Der VFX-Moverdialog ist ein leistungsfähiges Bedienungselement, das Sie in Ihren Anwendungen benutzen können. Der VFX-Moverdialog bekommt als Parameter zwei Arrays übergeben. Das erste Array enthält zur Auswahl stehende Elemente. Diese Elemente werden in der linken Listbox angezeigt. Das zweite Array enthält die ausgewählten Elemente. Das zweite Array kann bei Aufruf des Moverdialogs leer sein. Der Anwender kann eine beliebige Anzahl von Elementen auswählen.



Hier ein Beispielcode für die praktische Anwendung des VFX-Moverdialog-Steuerelements:

```
LOCAL laSource[1,1], loMover

*--prepare the array of all available items
SELECT keygrp_id, keygrp_name FROM keygrp INTO ARRAY laSource

*--create the mover object based on the VFX Class CMoverDialog
loMover = CREATEOBJECT("CMoverDialog")
*--set the caption
loMover.Caption = CAP_KEYFIELDGEN
*--set the property which defines which column from the array get's displayed
loMover.cntMover.nColToView = 2
*--enable multiple selections
loMover.cntMover.lstSource.MultiSelect = .T.
*--pass the array of all available items
* here you can also pass a second parameter if you want to define, which
* elements from the array must appear as already selected
loMover.cntMover.SetData(@laSource)
*--show the mover dialog
loMover.Show()

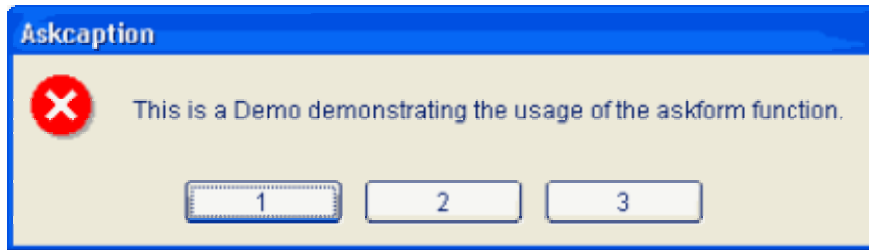
*--Result: The Public Array _gaMoverList contains the selected items, use it
* and release this Public Array after you have done.
```

Nach der Erstellung des Objektes *loMover* haben Sie die vollständige Kontrolle darüber und können alle gewünschten Eigenschaften und Methoden verändern.

ANMERKUNG: Um eine detaillierte technische Beschreibung der VFX-Klassenbibliotheken inklusive aller Eigenschaften und Methoden zu erhalten, lesen Sie bitte in der VFX Technischen Referenz nach.

Askform

Die Askform entspricht in etwa einer MessageBox, hat jedoch eine erweiterte Funktionalität. Die Beschriftungen der (maximal) drei Schaltflächen können als Parameter übergeben werden. Außerdem ist es möglich ein Timeout für die MessageBox festzulegen. Bei Erreichen des Timeouts ohne Benutzeraktion wird ein Rückgabewert geliefert, der dem Drücken der Standard-Schaltfläche entspricht.



Ein Beispiel zur Verwendung der Funktion *Askform()* befindet sich im Formular *Parent.scx* aus der Demoapplikation VFX80Test.

IDX Know How

VFX macht von vorhandenen Indexschlüsseln bestmöglichen Gebrauch. Für die inkrementelle in VFX-Power Grids durchsucht VFX automatisch alle vorhandenen Indexschlüssel der verwendeten Tabelle. Für Zeichenfelder wird ein Indexschlüssel mit *UPPER()*-Klausel erwartet. Für Datumsfelder wird ein Indexschlüssel mit *DTOS()*-Klausel erwartet.

Wenn VFX keinen passenden Indexschlüssel findet, wird eine temporäre Indexdatei angelegt. Diese Indexdatei wird gelöscht, sobald das Formular geschlossen wird. Ferner wird die Indexdatei gelöscht, wenn das Formular in den Bearbeitungsmodus oder in den Einfügemodus wechselt sowie beim Löschen von Datensätzen. Das ist sinnvoll, weil laufende Transaktionen, wie sie zum Beispiel im RI-Code verwendet werden, zu VFP-Laufzeitfehlern führen würden, wenn temporäre Indexdateien geöffnet sind. VFP erlaubt keine temporären Indexdateien, wenn mit Transaktionen gearbeitet wird.

Wenn in einem Formular Transaktionen verwendet werden, kann auf Wunsch nach der Datenbearbeitung der zuvor gültige Indexschlüssel wieder erstellt werden. Dem Anwender wird vorgetäuscht, dass die gewählte Sortierfolge ständig erhalten bleibt. Stellen Sie dafür in *Vfxmain.prg* ein:

```
lremakeidxafterclear = .T. && Index nach der Bearbeitung wieder erstellen.
```

Wenn in einem Formular und jeglichem daraus aufgerufenen Code keine Transaktionen ausgeführt werden, also in den beteiligten Tabellen auch kein RI-Code hinterlegt ist, können Sie in *Vfxmain.prg* einstellen, dass temporäre Indexdateien bei der Datenbearbeitung nicht gelöscht werden:

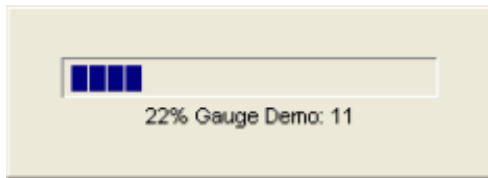
```
lnoclearidxonedit = .t. && Index zum bearbeiten nicht löschen.  
lnoclearidxoninsert = .t. && Index zum einfügen nicht löschen.  
lnoclearidxondelete = .t. && Index zum löschen von Datensätzen nicht löschen.
```

Temporäre Indexdateien werden in jedem Fall beim Schließen eines Formulars gelöscht.

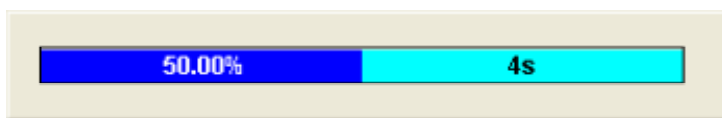
Fortschrittsanzeige

VFX bietet 2 Möglichkeiten den Fortschritt von lange andauernden Vorgängen zu verdeutlichen.

Die einfache Variante, realisiert mit der Formalklasse *cGaugeWin*, zeigt einen Balken zur Anzeige des Fortschritts an.



Mit dem Formular *Vfxmtr.scx* kann eine Fortschrittsanzeige mit Anzeige der Restzeit dargestellt werden.



Beispiele für die Verwendung beider Fortschrittsanzeigen befinden sich im Formular *Parent.scx* der Demoapplikation VFX80Test.

Datumsauswahl

Die Klasse *cPickDate* stellt ein Steuerelement zur Auswahl eines Datums zur Verfügung. Es wird das ActiveX-Control Microsoft MonthView verwendet. Bei der Erstellung eines Setups muss dieses ActiveX-Control (Mscomct2.ocx) mit in das Setup einbezogen werden.



Auswahl von Berichten

Wenn zu einem Formular verschiedene Berichte gedruckt werden sollen, bietet die Klasse *cRSelection* einen geeigneten Auswahldialog. Die zur Verfügung stehenden Berichte werden aus Tabellen gelesen. Es kann zwischen Berichten unterschieden werden, die für alle Benutzer sichtbar sind und Berichten, die nur für einzelne Benutzer sichtbar sind.

Ein Beispiel zur Anwendung findet sich im Formular *Reports.scx* in der Demoapplikation VFX80Test.

Die Microsoft Agents

Die Agents sind nette Charaktere, die die Benutzung von VFX-Anwendungen auflockern.

In VFX80Test zeigt das Formular *Agent.scx* einfache Beispiele für die Verwendungsmöglichkeiten.

Benutzerspezifische Einstellungen

VFX erstellt für jedes Feld aus der Tabelle *VFXUSR.DBF* eine Public Variable mit dem Präfix *gu_* und erledigt vollautomatisch das Speichern und Lesen dieser Werte.

Nehmen wir an, dass Sie ein Feld mit dem Namen *TEST* in der Tabelle *VFXUSR* haben. Nach der Benutzeranmeldung wird eine Public Variable *gu_test* den Wert aus dem Feld *Test* der *VFXUSR*-Tabelle beinhalten. Falls diese Variable verändert wird, wird beim Verlassen der Anwendung dieser Wert wieder zurück in das Feld *Test* der Tabelle *VFXUSR* geschrieben.

Auf diese Weise ist es sehr einfach, benutzerspezifische Einstellungen zu speichern. Es reicht aus, in der Tabelle *VFXUSR* ein entsprechendes Feld anzulegen.

Include-Dateien

Die Include-Dateien spielen bei VFX eine wichtige Rolle. Es lohnt sich deshalb, die vorhandenen Include-Dateien etwas näher anzusehen:

Include-Datei	Verwendung	Ordner	Sprachabhängig?	Inhalt/Beschreibung
VFX.H	VFXMAIN.PRG	\VFX\INCLUDE\	Nein	Definiert die Konstanten <i>_DEBUG_MODE</i> , <i>LANGSETUP</i> , <i>_DBCX</i> und schließt andere Include-Dateien ein.
FOXPRO.H	VFX.H	Visual FoxPro Ordner	Nein	Standard-FoxPro-Definitionen.
VFXDEF.H	VFX.H	\VFX\INCLUDE\	Ja	Definiert die <i>ID_LANGUAGE</i> -Konstante und andere Konstanten.
VFXTXT.H	VFX.H	\VFX\INCLUDE\	Ja	Sprachabhängige Texte und Tooltip-Texte, die in der VFX-Entwicklungsumgebung verwendet werden.
VFXMSG.H	VFX.H	\VFX\INCLUDE\	Ja	Sprachabhängige Meldungstexte, die in der VFX-Entwicklungsumgebung verwendet werden.
VFXOFFCE.H	VFX.H	\VFX\INCLUDE\	Nein	In den Office-Klassen Word, Excel und Outlook verwendet.
USERTXT.H	VFX.H	\VFX\INCLUDE\	Ja	Sprachabhängige Texte und Tooltip-Texte, die Sie in Ihrer eigenen Anwendung verwenden. Die Datei wird von dem VFX – Message Editor erzeugt, wenn Sie die Option OTHER wählen.
USERMSG.H	VFX.H	\VFX\INCLUDE\	Ja	Sprachabhängige Meldungstexte, die Sie in Ihrer eigenen Anwendung verwenden. Die Datei wird von dem VFX - Message Editor erzeugt, wenn Sie die Option MESSAGE wählen.
USERDEF.H	VFX.H	\VFX\INCLUDE\	Nein	Sprachunabhängige Konstanten, die in Ihrer Anwendung verwendet werden.

Der VFX Anwendungs-Assistent generiert die meisten Konstanten automatisch, wenn Sie ein neues Projekt generieren. Wenn Sie den Debug-Modus oder die aktuelle Sprache wechseln wollen, müssen Sie Änderungen in einigen der Include-Dateien machen.

Define _Debug_Mode

VFX benutzt eine Konstante in der Include-Datei *VFX.H*, die angibt ob die Anwendung im Debug-Modus ablaufen soll oder nicht. Standardmäßig sind die folgenden Codezeilen in der Datei *VFXMAIN.PRG*, um den Debug-Modus in Abhängigkeit von der Konstanten *_DEBUG_MODE* einzustellen:

```
#ifdef _DEBUG_MODE
    goProgram.DebugMode(.t.)
#endif
```

Wenn Sie nicht wollen, dass Ihre Anwendung im Debug-Modus ausgeführt wird, kommentieren Sie Zeile mit der *_DEBUG_MODE*-Konstanten aus. Die Konstante befindet sich in der Include-Datei **VFX.H**:

```
* #DEFINE _DEBUG_MODE .T.
```

Im Debug-Modus ist ein zusätzliches Menü sichtbar, mit dessen Hilfe jederzeit der Debugger gestartet werden kann. Außerdem kann durch einen Rechtsklick mit der Maus auf einem Formular der Debugger gestartet werden. Dabei wird auch das Set-Fenster geöffnet.

Define ID_Language

In der Include-Datei *VFXDEF.H* ist die *ID_Language*-Konstante definiert, die die aktuelle Sprache Ihrer Anwendung angibt.

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

Wenn Sie Ihre Anwendung mit dem VFX-Anwendungs-Assistenten anlegen, wird die Anwendung in der Sprache angelegt, die im VFX-Anwendungs-Assistenten angegeben ist. Wenn Ihre Anwendung in eine andere Sprache übersetzt werden soll, ändern Sie die Konstante *ID_Language*. Lesen Sie im Kapitel *Mehrsprachige Anwendungen mit VFX* nach, um nähere Informationen zu erhalten.

Define _Lang_Setup

In der Include-Datei *VFX.H* gibt die Konstante *_LANG_SETUP* an, ob die *LangSetup()*-Methode ausgeführt wird oder nicht. Innerhalb der *LangSetup()*-Methode wird überprüft, ob die Konstante existiert. Nur wenn die Konstante existiert, wird der Code der *LangSetup()*-Methode ausgeführt. Dies dient der Geschwindigkeitsoptimierung in allen Formularen.

```
#DEFINE _LANG_SETUP .T.
```

Kompilieren Ihrer Anwendung nach Änderungen in Include-Dateien

Um Visual FoxPro zu einem Neukompilieren zu veranlassen, müssen Sie eine Änderung in der oder den Datei(en) vornehmen, die die Include-Dateien einschließen. Der Befehl *clear program* im Befehlsfenster löscht alle kompilierten Programme im Hauptspeicher. Sie sollten die Datei *VFX.H* in Ihre Formulare einschließen, wenn Sie Konstanten in Ihren Formularen verwenden.

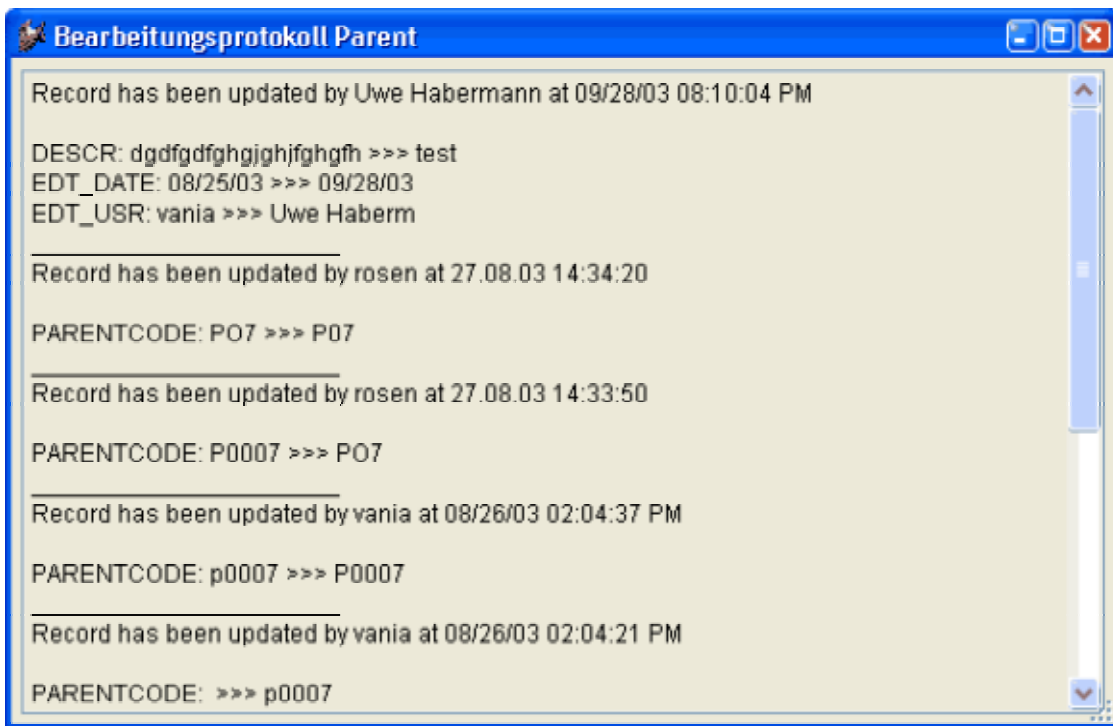
Bearbeitungsprotokoll

Das Bearbeitungsprotokoll (Audit-Trail) protokolliert Änderungen von Daten. VFX verwendet Trigger um die Änderung von Daten zu ermitteln. Die Trigger-Funktionen werden bei allen zu überwachenden Tabellen eingetragen.

- *_audit_insert()* protokolliert die Erfassung neuer Datensätze
- *_audit_update()* protokolliert alle Änderungen
- *_audit_delete()* protokolliert das Löschen von Datensätzen

Ein Audit-Trigger kann mit einem RI-Trigger mit einem logischen „und“ verknüpft werden:

```
__ri_delete_parent() AND _audit_delete()
```



Über eine Schaltfläche in der Standard-Symbolleiste kann zum aktuell angezeigten Datensatz das Änderungsprotokoll angesehen werden.

OLE drag & drop

In VFX-Anwendungen steht OLE drag & drop auf drei verschiedene Arten zur Verfügung. Standardmäßig ist OLE drag & drop in Datenrastern eingeschaltet. Der gesamte Inhalt eines Datenrasters kann mit einem Mausklick zum Beispiel nach Excel kopiert werden.

Auf Wunsch können auch die Inhalte einzelner Steuerelemente per OLE drag & drop verschoben werden. Diese Eigenschaft ist standardmäßig ausgeschaltet und kann durch die Zeile

```
nOLEenableDrag=1 && 0 use form setting (default), 1 enable, 2 disable
```

in Vfxmain.prg eingeschaltet werden.

Weiterhin ist es möglich die Daten aller Steuerelemente einer Seite eines Seitenrahmens in eine andere OLE drag & drop-fähige Anwendung zu kopieren. Auch diese Eigenschaft ist standardmäßig ausgeschaltet und kann bei Bedarf durch die Zeile

```
nPageOLEdragdrop=1 && 0 use form setting (default), 1 enable, 2 disable
```

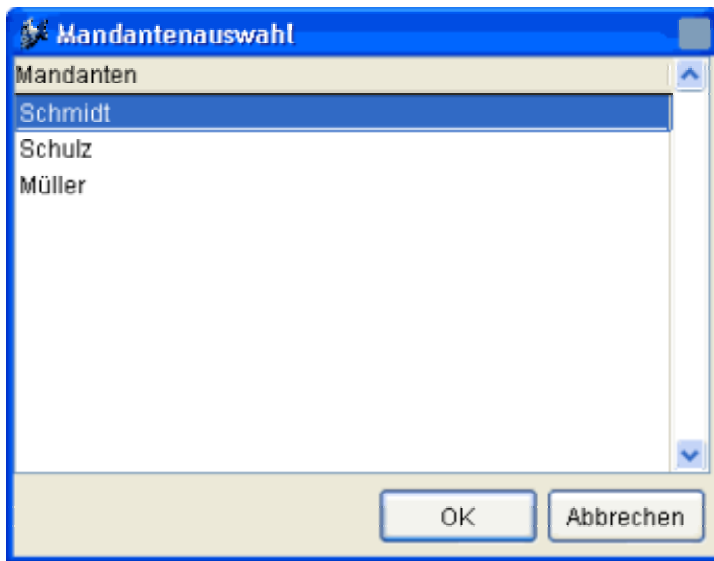
in Vfxmain.prg eingeschaltet werden.

Multi-Client-Support

Standardmäßig arbeitet eine VFX-Anwendung mit genau einer Datenbank, so wie es im VFX – Application Wizard eingetragen wurde. Auf Wunsch kann eine Mandantenfähigkeit eingebaut werden. Dazu ist in Vfxmain.prg der Datenpfad auf einen Leerstring zu setzen.

```
cdataidir = ''
```

Wenn der Datenpfad leer ist, sucht die VFX-Anwendung zur Laufzeit nach der Tabelle *Vfxpath.dbf*. Diese Tabelle muss sich im gleichen Ordner wie die ausführbare Programmdatei befinden. Wenn in dieser Tabelle genau ein Datensatz enthalten ist, wird der dort eingetragene Datenpfad verwendet. Enthält die Tabelle mehr als einen Datensatz erscheint beim Start der Anwendung ein Dialog zur Auswahl der gewünschten Datenbank.



Aktualisierung der Kundendatenbank

Verwendung von VFP-Datenbanken

VFX bietet die Möglichkeit, die Datenbank beim Kunden automatisch zu aktualisieren. Die zu aktualisierenden Tabellen werden hierbei ohne Daten in den Update-Ordner unterhalb des Datenordners kopiert. Beim ersten Programmstart erkennt die VFX-Anwendung das Vorhandensein der Tabellen im Update-Ordner und aktualisiert die Datenbank. Es können auch freie Tabellen aktualisiert werden.

Verwendung von SQL Server-Datenbanken

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

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

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

Hooks

Eine elegante Möglichkeit in den Funktionsablauf von VFX-Methoden einzugreifen, ohne die Klassen verändern zu müssen, ist der Einsatz von Hooks.

Das Konzept der Hooks wurde in VFX 8.0 erweitert. Bisher war es möglich durch einen Hook innerhalb einer VFX-Methode einen eigenen Codeblock auszuführen. Über den Rückgabewert des Hooks konnte man steuern, ob der noch folgende VFX-Code in der Methode weiter ausgeführt werden sollte oder nicht. Der Rückgabewert, den die VFX-Methode dabei lieferte, konnte nicht beeinflusst werden und war in VFX fest vorgegeben.

Mit den erweiterten Hooks in VFX 8.0 kann nun zusätzlich der Rückgabewert der Methode vom Hook gesteuert werden.

Hooks sind in der Datei Vfxhook.prg gespeichert. Die Verwendung von Hooks kann im VFX – Application Manager oder in Vfxmain.prg mit der Zeile

```
nenablehook = 1
```

eingeschaltet werden. Nenablehook ist eine Eigenschaft des Applikationsobjekts.

Im folgenden Beispiel wird bei allen Steuerelementen, die disabled sind, die Schriftfarbe schwarz eingestellt.

```
function EventHookHandler(tcEvent, toObject, toForm)
    local lContinue
    lContinue = .T.
    DO CASE
    CASE UPPER(tcEvent)=="INIT"
        IF PEMSTATUS(toObject,"disabledforecolor",5)
            toObject.disabledforecolor=;
            eval(left(rgbscheme(1,2),at(", ",rgbscheme(1,2),3)-1)+") ")
        IF PEMSTATUS(toObject,"disabledbackcolor",5)
            toObject.disabledbackcolor=;
            eval("rgb("+substr(rgbscheme(1,2),;
            at(", ",rgbscheme(1,2),3)+1))
        ENDIF
    ENDIF
    ENDCASE
    return lContinue
endfunc
```

Benutzen Sie die gewünschte Standard-Symboleiste

Es ist vernünftig, für die Bedürfnisse Ihrer Anwendung (oder Ihrer Firma) eine eigene Klassenbibliothek anzulegen. Wir haben eine Klassenbibliothek mit dem Namen *APPL.VCX* für Sie vorbereitet. Um Ihnen die Arbeit so einfach wie möglich zu machen, haben wir in dieser Klassenbibliothek bereits zwei Klassen für Symbolleisten angelegt:

CAppBar und *CAppNavBar*.

Die Erste ist die Standard-Symboleiste und die Zweite ist eine Symbolleiste, die Sie verwenden können, wenn Sie Navigations- und andere Schaltflächen nicht auf Ihren Formularen haben wollen.

***CAppBar*:**



CAppBar wird benutzt, wenn die Schaltflächen zur Navigation und zur Bearbeitung auf Ihren Formularen sind.

CAppNavBar:



CAppNavBar wird benutzt, wenn die Schaltflächen zur Navigation und zur Bearbeitung nicht auf Ihren Formularen sind.

Um zwischen diesen beiden Symbolleisten zu wechseln, brauchen Sie nur eine Eigenschaft der Anwendungs-klasse in *VFXMAIN.PRG* zu ändern:

```
define class CApplication as CFoxApp
...
...
*****
** CAppToolBar - Toolbar without Navigation Buttons
** CAppNavBar - Toolbar with Navigation Buttons
cMainToolBar = "CAppNavBar"
...
...
```

Symbolleisten zu Formularen

Es hat sich als sehr praktisch erwiesen Formularen eigene Symbolleisten zuordnen zu können. Die Symbolleisten sollten auf der Klasse *ctoolbar* basieren und in der Klassenbibliothek *Appl.vcx* gespeichert werden. Der Name der Symbolleiste wird dem Formular in der Eigenschaft *ctoolbarclass* bekannt gemacht.

VFX instanziiert die Symbolleiste zusammen mit dem Formular. Die Symbolleiste ist sichtbar, solange das Formular das aktive Formular ist.

Um zum Beispiel ein Child-Formular über eine Schaltfläche in einer Symbolleiste zu öffnen, fügen wir der Symbolleiste eine Schaltfläche basierend auf der Klasse *ctoolbarclass* hinzu. In das Click-Event der Schaltfläche schreiben wir

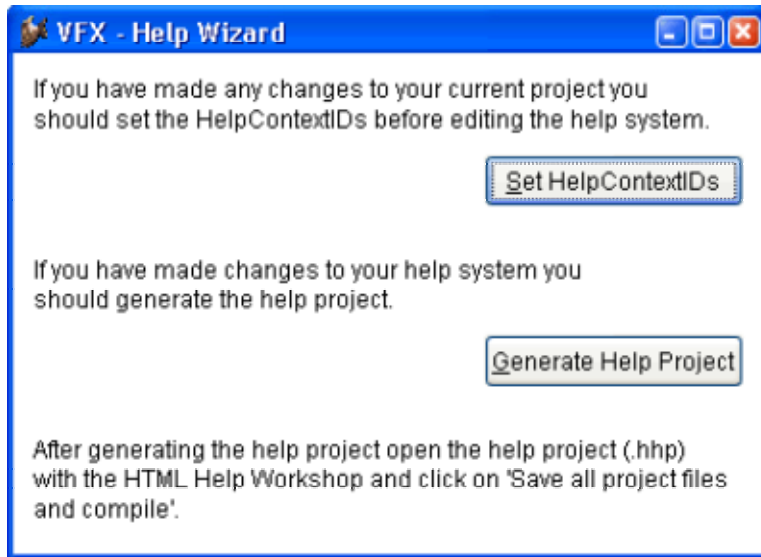
```
_screen.activeform.onmore(1)
```

Das ist alles. Da VFX sicherstellt, dass die Symbolleiste nur dann sichtbar ist, wenn das dazugehörige Formular aktiv ist, können wir sicher sein, dass *_screen.activeform* existiert. Von diesem Formular wird die *onmore* Methode aufgerufen und bekommt als Parameter eine 1 übergeben. Damit wird das Formular aufgerufen, das im ersten Array-Element der *onmore* Methode angegeben ist, ohne dass der OnMore-Dialog angezeigt wird.

VFX – Help Wizard

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

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



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

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

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

OLE-Klassen

Es ist möglich Word, Excel, Outlook und Powerpoint per OLE aus VFX-Applikationen anzusteuern. Die wichtigsten Funktionen stehen in Klassen zur Verfügung.

Die Weiterentwicklung mit VFP

Das gesamte VFX 8.0-Projekt liegt in normalen VFP Quelldateien vor. Die erstellte Anwendung kann also jederzeit mit VFP weiterentwickelt werden, auch wenn auf dem Entwicklungsrechner VFX nicht installiert ist.

Dokumentation

Neben dem Benutzerhandbuch gibt es zu VFX eine Menge an Online-Dokumentation. Dazu gehört insbesondere die Technische Referenz, die als Windows-Hilfedatei vorliegt. In ihr ist zu jeder Klassenbibliothek, zu jeder Klasse jede Methode und jede Eigenschaft beschrieben. In einem Tutorial werden anhand von typischen Anwenderfragen die Lösungen mit VFX erläutert. Direkt aus der Technischen Referenz können Videos (Avi-Dateien) gestartet werden. Es gibt 10 Videos mit insgesamt ca. 45 Minuten Dauer. In den Videos wird die Erstellung von Formularen für Fileserver- und Client-/Server-Datenbanken beschrieben und gezeigt. Für den VFX-Anfänger eine große Hilfe bei der Einarbeitung.

Support

Support für VFX ist im dFPUG-Forum (<http://forum.dfpug.de>) zu finden. Dort gibt es sowohl eine deutsche als auch eine englische Sektion zu VFX. Diese Sektionen können auch alternativ als Newsgroup (<news://news.dfpug.de>) gelesen und bearbeitet werden.

Im Internet findet man auf der Website von Visual Extend (<http://www.visualextend.de>) weitere Informationen zum Produkt. Auch ist hier der Download der Demoapplikation, der gesamten Dokumentation und der aktuellen Vollversion von VFX möglich. Eine umfangreiche Sammlung weiterer Dokumente rund um VFX findet sich im Dokumentenportal der dFPUG (<http://portal.dfpug.de>). Aktuelle Informationen erhalten Sie über den kostenlosen dFPUG-eNewsletter im Abschnitt zu VFX (<http://newsletter.dfpug.de>).

Zusammenfassung

Wie wir gesehen haben stellt VFX eine vollständige Entwicklungsumgebung bereit, die keine Wünsche offen lässt. Alle wesentlichen Einstellungen an VFX-Klassen, insbesondere an den Formularklassen, können mit reentranten Buildern durchgeführt werden. Alle in diesem Artikel beschriebenen Eigenschaften und Funktionen lassen sich praktisch ohne Programmierung nur durch den Einsatz der Builder erreichen.

Trotzdem ist es an praktisch jeder Stelle über Hooks möglich in den Programmablauf einzugreifen.

Da VFX mit Quellen geliefert wird und selbst mit VFP programmiert ist, hat der Entwickler unbegrenzte Freiheit eigene Erweiterungen oder Anpassungen an eigene Bedürfnisse vorzunehmen.

Die Performance von VFX-Anwendungen ist so gut, wie sie mit VFP-Anwendungen nur sein kann. Die Vererbungstiefe ist gering. Die meisten Klassen haben nur 1 bis 2, maximal jedoch 4 Vererbungsebenen hinter sich. Um das Laden von umfangreichen Formularen weiter zu beschleunigen kann Delayed Instantiation verwendet werden. Auch dies wird von VFX mit einfach zu handhabenden Funktionen unterstützt.

Die mit VFX erstellten Applikationen vermitteln dem Anwender einen sehr professionellen Eindruck und eine Office-kompatible Bedienung.

VFX bietet mit all dem ein unschlagbares Preis-/Leistungsverhältnis. Es bietet jedem Programmierer eine Fundgrube an Ideen und eine Vielzahl von fertigen Problemlösungen.

VFX 8.0 - Produktiver als je zuvor!