



VFX 9.5 – Was ist neu? 3. Quartal 2006

Oktober 2006



Uwe Habermann, Venelina Jordanova

Inhaltsverzeichnis

Neuheiten für Entwickler.....	3
Suchdialog.....	3
Herunter laden und entpacken eines Archivs in den Exe-Ordner.....	3
Eindeutige Felder.....	3
VFX – Data Environment Builder.....	3
Produktaktivierung.....	4
Anwendungsschutz.....	4
Erstellen von Aktivierungsschlüsseln mit der VFX – Kundenverwaltung.....	6
VFX – Web Service für die Registrierung.....	6
Impersonation oder Personifizierung.....	7
VFX – Installation Wizard.....	7
Behandlung von Laufzeitfehlern.....	9
Datenbankreparatur.....	9
Datenbankwartung bei Anwendungsstart.....	9
Die Klasse cDatabaseTools.....	10
Neue Eigenschaften des Anwendungsobjekts goProgram.....	11
Neuheiten für Endbenutzer.....	12
Die Klasse cComboPickList.....	12
Verwaltung von Auswahllisten.....	13
VFX – Combo Pick List Builder.....	14
Die Klasse cDateTextBox.....	15
Die Klasse cDocumentManagement.....	16
Drag und Drop in cDocumentManagement.....	16
VFX – Document Management Builder.....	17
Die Klasse cChildGrid.....	18
Unterstützung von Favoriten.....	18
VFX – CChildGrid Builder.....	18
Positionierung des Satzzeigers in Onetomany Formularen.....	18
Berichte.....	19
Scannen von Bildern.....	20
Die Klasse cScannerControl.....	20
Die Klasse cScanForm.....	22
Zum Scannen benötigte Funktionen in der VFX.FLL.....	23
Liste der TWAIN Fehlermeldungen.....	24

Neuheiten für Entwickler

Suchdialog

Filterdefinitionen können wahlweise in einer der Tabellen *VfxFilter.dbf* oder *VfxRes.dbf* gespeichert werden. Diese Einstellung gilt für benutzerspezifische Filter und für Systemfilter. Die verwendete Tabelle kann mit der Eigenschaft *IUseFilterTable* des Anwendungsobjekts eingestellt werden. Wenn der Wert auf *.T.* eingestellt ist, wird die Tabelle *VfxFilter.dbf* für benutzerspezifische Filter und für Systemfilter verwendet.

Im Suchdialog können Werte in den Datentypen *Date* und *Datetime* in einem Steuerelement der Klasse *cDateTextBox* eingegeben werden. Ein Kalender steht zur Verfügung.

Herunter laden und entpacken eines Archivs in den Exe-Ordner

Mit dieser Funktion kann bei der Aktualisierung der Anwendung zusätzlich eine Zip-Datei heruntergeladen und im Ordner der Exe-Datei entpackt werden. Die Zip-Datei wird dabei entpackt, bevor die neue Exe-Datei erstmalig gestartet wird.

Für diesen Zweck wurde der Klasse *cFoxApp* die neue Eigenschaft *cAddFilesDownloadURL* hinzugefügt. Diese Eigenschaft enthält die URL der herunter zu ladenden und zu entpackenden Datei. Der Code hierzu befindet sich in der neuen Methode *AddFilesUpdate* des Anwendungsobjekts.

Eindeutige Felder

Der Klasse *cTextBoxBase* wurde die neue Eigenschaft *IUniqueField* hinzugefügt. Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt wird, wird beim Speichern die Eindeutigkeit dieses Wertes in der Datenquelle, die als *Controlsource* angegeben ist, geprüft.

Der Wert dieser Eigenschaft kann in allen VFX – Form Buildern mit der Checkbox *Unique field* eingestellt werden. Der Wert dieser Eigenschaft kann auch im VFX – TextBox Builder eingestellt werden.

Die Eindeutigkeit kann nur in Steuerelementen basierend auf der Klasse *cTextBox* geprüft werden. Die Eindeutigkeit kann nur für Felder aus der Tabelle geprüft werden, die der *Initialselectedalias* des Formulars ist.

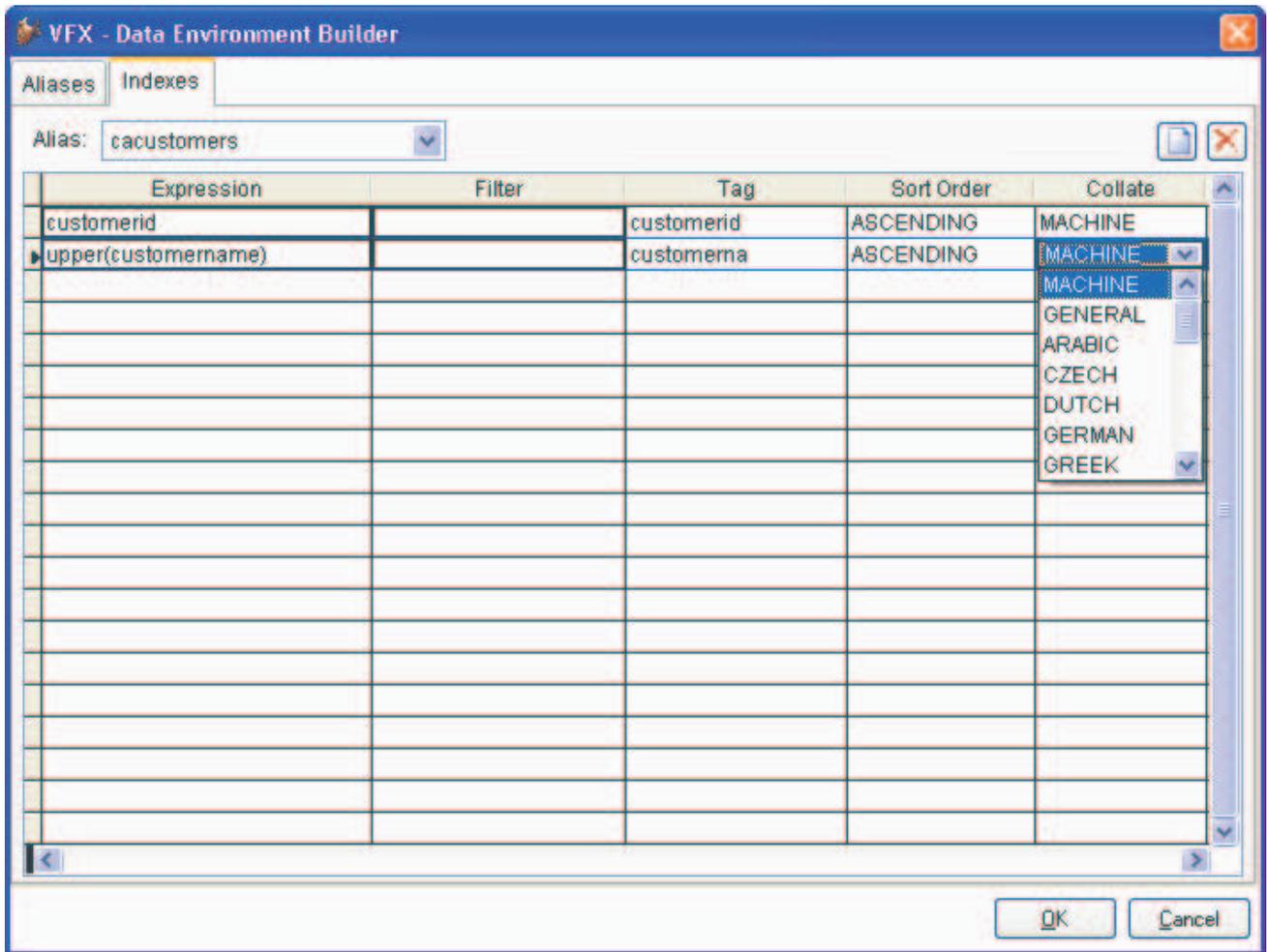
VFX – Data Environment Builder

Im VFX – Data Environment Builder wurde auf der Seite *Aliases* die Checkbox *NoData* hinzugefügt. Hiermit kann der Wert der Eigenschaft *NoData* bei Cursoradapter-Objekten eingestellt werden.

Wenn in der Eigenschaft *cWhereclause* ein Parameter eingetragen wird, wird der Wert der Eigenschaft *NoData* automatisch auf *.T.* eingestellt. Dadurch wird das doppelte Füllen des Cursors beim Formularstart vermieden.

Wenn der Wert aus der Eigenschaft *cWhereclause* gelöscht wird, behält die Checkbox *NoData* den Wert *.T.* Ggf. muss dieser Wert dann manuell geändert werden.

Auf der Seite *Indexes* kann in einer Combobox die Collate Sequenz für jeden Indexschlüssel individuell eingestellt werden. Der Standardwert ist *Machine*.



Produktaktivierung

Anwendungsschutz

Aktivierungsschlüssel können in zwei verschiedenen Formaten generiert werden. Standardmäßig wird das aus VFX 9.0 bekannte Format für Aktivierungsschlüssel verwendet. Wahlweise kann ein Microsoft kompatibles Format verwendet werden. Das heißt, dass der Aktivierungsschlüssel aus 25 alphanumerischen Zeichen besteht. Wenn das Microsoft kompatible Format für Aktivierungsschlüssel verwendet wird, können zeitlich befristete Aktivierungsschlüssel erstellt werden. Die hierfür benötigten Einstellungen können in Eigenschaften der Klasse *cVFXActivate* in der Klassenbibliothek *Appl.vcx* gemacht werden. Die Einstellungen können im VFX –Application Builder gemacht werden.

nProductActivationBehavior – 1 – (Standardwert) VFX 9.0 kompatibles Format des Aktivierungsschlüssels, 2 – Microsoft kompatible Format des Aktivierungsschlüssels.

UseTimeLimitedActivationKey – Um zeitlich befristete Aktivierungsschlüssel erstellen zu können, muss der Wert dieser Eigenschaft auf *.T.* eingestellt werden. Dies ist nur zulässig, wenn der Wert der Eigenschaft *nProductApplicationBehavior* auf 2 eingestellt ist. Der Standardwert ist *.F.*

dStartActivationDate – In dieser Eigenschaft wird das Startdatum zur Berechnung von zeitlich befristeten Aktivierungsschlüsseln gespeichert. Der Standardwert ist der 01. Januar 2000. Das Datum der letzten möglichen Benutzung wird aus diesem Startdatum und der Anzahl der Tage der Gültigkeit berechnet.

nDefaultValidityDays – Diese Eigenschaft enthält die Anzahl an Tagen, die standardmäßig verwendet wird, wenn ein zeitlich befristeter Aktivierungsschlüssel erstellt wird. Dieser Wert wird verwendet, wenn vom Web Service ein Aktivierungsschlüssel automatisch erstellt wird.

Außerdem ist einstellbar, ob ein Aktivierungsschlüssel tolerant gegenüber Änderungen in der Hardware-Konfiguration sein soll. Die Anzahl der möglichen Hardware-Veränderungen kann eingestellt werden. Die hierfür benötigten Eigenschaften befinden sich in der Klasse *cVFXActivation* in der Klassenbibliothek *Appl.vcx* und können im VFX – Application Builder eingestellt werden.

nHardwareParametersTolerance – Anzahl der möglichen Veränderungen an der Hardware. Wenn der Wert dieser Eigenschaft größer als 1 ist, ist die Toleranz aktiv. Der Standardwert ist 0.

cStoreHardwareParameters – Name der Textdatei, in der die Informationen über die bei der Aktivierung aktuellen Hardware-Parameter gespeichert werden. Der Standarddateiname ist *vfx.hrd*.

cEncryptPassword – In dieser Eigenschaft wird das Kennwort gespeichert, das zur Verschlüsselung der Datei mit den Hardware-Parametern verwendet wird.

Wenn die Hardware-tolerante Aktivierung verwendet wird, werden auf dem PC des Endanwenders die Informationen über die Hardwareparameter, die zum Zeitpunkt der Aktivierung verwendet wurden, in einer verschlüsselten Textdatei gespeichert. Das zur Verschlüsselung benötigte Kennwort ist in der Eigenschaft *cEncryptPassword* der Klasse *cVFXActivate* gespeichert. Der Name dieser Textdatei befindet sich in der Eigenschaft *cStoreHardwareParameters* der Klasse *cVFXActivate*. Die Datei wird im gleichen Ordner, wie die Ini-Datei gespeichert. Der Speicherordner entspricht den Richtlinien „Designed for Windows XP“.

Wenn eine Anwendung die Gültigkeit der Aktivierung überprüft, werden die Hardware Parameter ermittelt. Es wird dann zunächst verglichen, ob die aktuellen Werte zu dem Aktivierungsschlüssel passen. Wenn die Werte nicht übereinstimmen, werden die aktuellen Hardware Parameter mit den Werten verglichen, die in der Textdatei *vfx.hrd* gespeichert sind. Wenn die Anzahl der nicht übereinstimmenden Werte kleiner oder gleich der Einstellung in der Eigenschaft *nHardwareParametersTolerance* ist, wird angenommen, dass die Aktivierung gültig ist. Wenn ein Aktivierungsschlüssel vorhanden ist, aber die Datei mit den Hardware Parametern nicht gefunden wird, wird davon ausgegangen, dass keine Hardware Toleranz erlaubt ist.

Erstellen von Aktivierungsschlüsseln mit der VFX – Kundenverwaltung

Wenn die VFX – Kundenverwaltung für eine Anwendung gestartet wird, die Microsoft kompatible Aktivierungsschlüssel verwendet, wird der Aktivierungsschlüssel in einer Textbox mit fünf alpha-numerischen Gruppen dargestellt.

The screenshot shows a web form for creating activation keys. It includes the following fields and controls:

- Registrierungsnummer:** A text input field.
- gültig bis:** A date selection control with a calendar icon.
- Aktivierungsschlüssel:** A text input field containing five dashes (- - - -) to indicate the key format.
- Registrierungsdatum:** A date selection control.
- Geändert am:** A date selection control.
- Download von Aktualisierungen erlauben**
- Aktivierungsschlüssel erstellen** (button)
- Aktivierungsschlüssel in XAK-Datei speichern** (button)

Das Steuerelement zur Eingabe der Gültigkeitsdauer ist nur dann sichtbar, wenn die Anwendung zeitlich befristete Aktivierungsschlüssel unterstützt. Der Standardwert für die Gültigkeit des Aktivierungsschlüssels wird aus dem aktuellen Tagesdatum und dem Wert der Eigenschaft `nDefaultValidityDays` ermittelt.

VFX – Web Service für die Registrierung

Der VFX – Web Service für die Registrierung kann zeitlich befristete Aktivierungsschlüssel erstellen und bei der Registrierung an den Kunden zurückschicken. Dadurch können Kunden auf sehr einfachem Weg temporär gültige Aktivierungsschlüssel erhalten, um eine Anwendung testen zu können. Die Möglichkeit besteht natürlich nur dann, wenn für die Anwendung zeitlich befristete Aktivierungsschlüssel erlaubt sind. Der generierte Aktivierungsschlüssel wird als Rückgabewert bei der Registrierung geliefert. Die Gültigkeitsdauer wird durch die Einstellung in der Eigenschaft `nDefaultValidityDays` in der Klasse `cVFXActivation` bestimmt. Die Registrierungsinformationen des Kunden, der generierte Aktivierungsschlüssel sowie die Gültigkeitsdauer werden in der Kundendatenbank gespeichert.

Im VFX – Application Builder kann eingestellt werden, ob eine Anwendung zeitlich befristete Aktivierungsschlüssel verwendet. Die Einstellung wird nur unterstützt, wenn das Microsoft kompatible Format für Aktivierungsschlüssel verwendet wird. Der Standardwert für die Gültigkeitsdauer kann ebenfalls im VFX – Application Builder eingestellt werden.

Impersonation oder Personifizierung

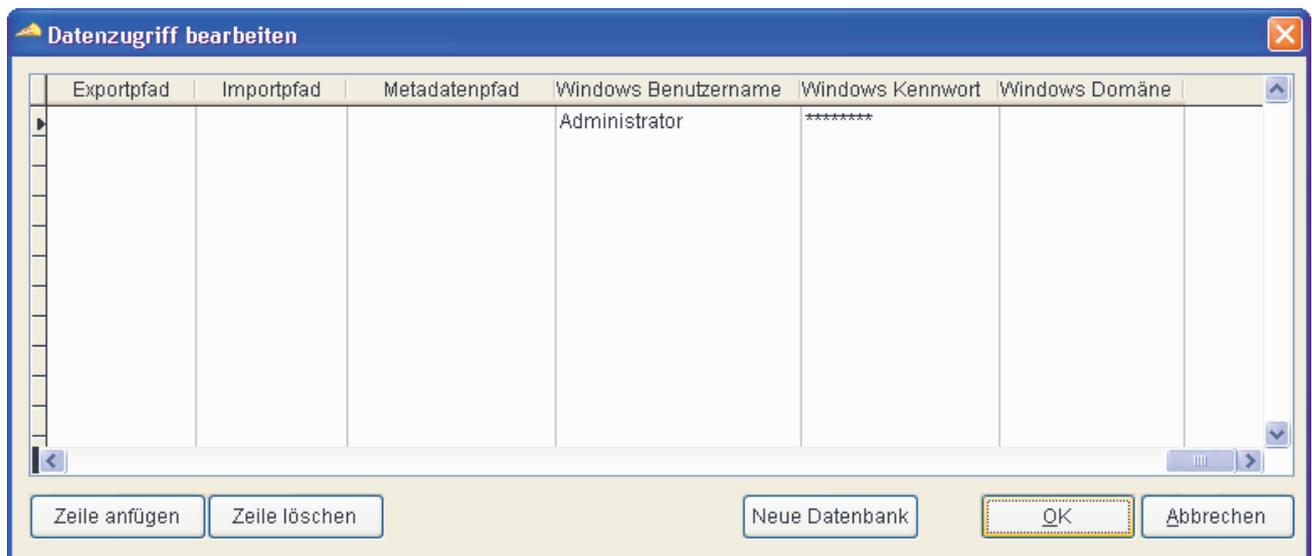
Durch Personifizierung kann die Sicherheit der Daten in VFX Anwendungen verbessert werden. Durch Personifizierung ist es möglich durch die Anwendung Zugriff auf die Daten zu erhalten, auch wenn der angemeldete Windows-Benutzer keinen Zugriff auf den Datenordner hat.

Die Verwendung der Personifizierung ist einfach. Es brauchen nur die Werte in drei Feldern in der Datei *Config.vfx* gefüllt werden.

Win User Name – Windows Anmeldename. Dies ist der Name des Benutzerkontos, das für die Personifizierung verwendet wird.

Win Password – Windows Kennwort.

Win Domain Name – Windows Domain Name (falls vorhanden).



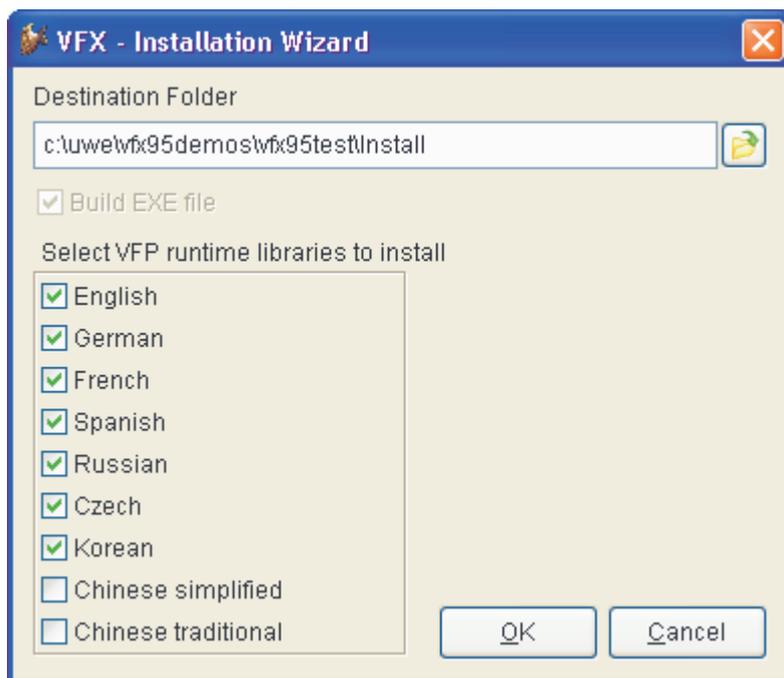
Wenn die Anwendung gestartet wird und eine Datenbank im Mandantenauswahldialog ausgewählt wird, meldet sich die Anwendung mit den Anmeldedaten aus der Datei *Config.vfx* an, wenn ein Benutzername und ein Kennwort angegeben wurden. Mit dem so angemeldeten Benutzerkonto wird der Zugriff auf die Datenbank sowie die VFX Tabellen durchgeführt.

Wenn Personifizierung verwendet wird, müssen für alle Datenbanken in der Datei *Config.vfx* Anmeldeinformationen eingetragen werden.

VFX – Installation Wizard

Der VFX – Installation Wizard sammelt alle benötigten Dateien um eine Anwendung auf einem Kundenrechner ausführen zu können. Wenn der VFX – Installation Wizard verwendet wird, braucht kein Installationsprogramm verwendet werden. Die gesammelten Dateien können in einen beliebigen Ordner auf dem Kundenrechner kopiert werden. Die Anwendung kann sogar von einem Memory Stick gestartet werden.

In dem erstellten Ordner befinden sich neben der Anwendung die VFP Runtime sowie alle benötigten ActiveX-Steuerelemente.



Der VFX – Installation Wizard kann aus dem VFX 9.5 Menü über den Menüpunkt *Project, Installation Wizard* gestartet werden.

Mit der Schaltfläche für die Ordnerauswahl kann der Ordner ausgewählt werden, in den der VFX – Installation Wizard alle gesammelten Dateien kopiert. Wenn der Ordner nicht existiert, wird er automatisch neu angelegt. Standardmäßig wird der Ordner *Install* unterhalb des Projektordners vorgeschlagen.

Unterhalb der Schaltfläche für die Ordnerauswahl befindet sich das Kontrollkästchen *Build EXE file*. Wenn dieses Kontrollkästchen markiert ist, wird eine Exe-Datei erstellt, bevor die Dateien in den Installationsordner kopiert werden. Wenn sich keine Exe-Datei im Projektordner befindet, ist das Kontrollkästchen *Build EXE file* markiert und disabled und es wird automatisch eine Exe-Datei erstellt.

Für jede Sprache, für die VFP eine Laufzeitbibliothek mit den Texten der jeweiligen Sprache bietet, ist ein Kontrollkästchen vorhanden. Standardmäßig sind alle Sprachen ausgewählt, ausgenommen vereinfachtes und traditionelles Chinesisch. Wenn auf dem Entwicklungsrechner nicht alle Laufzeitbibliotheken zur Verfügung stehen, sind die entsprechenden Kontrollkästchen disabled.

Durch einen Klick auf die Schaltfläche *OK* wird der Kopiervorgang gestartet. Dabei werden die VFP Laufzeitbibliotheken sowie alle benötigten ActiveX-Steuerelemente und DLLs in den Installationsordner kopiert. Während des Vorgangs wird auch ein Ordner *Data* angelegt. In diesen Ordner werden eine leere Datenbank sowie alle VFX Tabellen kopiert, die nicht in die Exe-Datei eingeschlossen sind.

Behandlung von Laufzeitfehlern

Wenn beim Lauf der Anwendung in der Entwicklungsumgebung Laufzeitfehler auftreten, wird ein spezieller Fehlerdialog angezeigt, der nur in der Entwicklungsumgebung zur Verfügung steht. In diesem Fehlerdialog hat der Entwickler die Möglichkeit den Debugger zu starten, den Fehler zu ignorieren oder die Programmausführung abzubrechen.

Im Fehlerdialog wird die Beschreibung des Fehlers angezeigt:



Ja – Die Programmausführung wird angehalten und der Debugger wird gestartet.

Nein – Der Fehler wird ignoriert.

Abbrechen – Die Programmausführung wird abgebrochen.

In jedem Fall werden die Fehlerinformationen in der Tabelle *Vfxlog* protokolliert.

Datenbankreparatur

Wenn bei der Datenbankwartung eine Neuindizierung gewählt wird und diese fehlschlägt, wird automatisch die Datenbankreparatur gestartet und die betroffenen Tabellen werden repariert.

Datenbankwartung bei Anwendungsstart

Eine Anwendung kann mit Parametern gestartet werden, um eine automatische Datenbankwartung durchzuführen. Die folgenden Parameter können der Anwendung übergeben werden:

tcDatabaseMaintenance – Der Wert von *tcDatabaseMaintenance* muss ein Wert aus der folgenden Liste sein: "\$REPAIR\$", "\$PACK\$", "\$PACKMEMO\$", "\$REINDEX\$" oder "\$TABLE\$<Tabellenname>".

Es darf immer nur einer dieser Werte übergeben werden. Wenn der Wert \$TABLE\$ übergeben wird, wird die angegebene Tabelle repariert.

tcPath – Eine mit \$ Zeichen separierte Liste mit Mandantennamen aus der Tabelle *Vfxpath.dbf* (diese Option kann nur in Verbindung mit *Vfxpath.dbf* genutzt werden, nicht jedoch mit *Config.vfx*). Wenn der Wert dieses Parameters leer ist, werden alle Mandantendatenbanken bearbeitet. Wenn dieser Parameter nicht mit einem \$ Zeichen beginnt, wird angenommen, dass es sich um den Pfad zu einer Datenbank handelt.

tcDBC – Name der Datenbank. Dieser Parameter muss übergeben werden, wenn in *tcPath* ein Ordnername angegeben ist.

Die Klasse *cDatabaseTools*

Eigenschaften

cCurrentTable – Name der aktuell ausgewählten Tabelle.

cDatabaseName –Name der Datenbank.

cDataDir – Pfad zur Datenbank.

cOldActiveDBC – Name der zuvor verwendeten Datenbank.

lSilentMode – Wenn *.T.* übergeben wird, werden keinerlei Meldungen angezeigt. Der Standardwert ist *.F.* Wenn das Formular mit Parametern gestartet wird, ist der Standardwert *.T.*

Methoden

Init(tcarg, tcAction, tcPath, tcDBC)

tcAction – Der Wert von *tcAction* kann "\$REPAIR\$", "\$PACK\$", "\$PACKMEMO\$", "\$REINDEX\$" oder ein Tabellenname sein. Wenn der Wert nicht mit einem \$ Zeichen beginnt wird angenommen, dass es sich um den Namen einer Tabelle handelt. Die Tabelle mit dem angegebenen Namen wird dann repariert. Es ist nicht zulässig mehrere Werte zu übergeben.

tcPath – Eine mit \$ Zeichen separierte Liste von Mandantennamen aus der Tabelle *Vfxpath.dbf*. Diese Option kann nur in Zusammenhang mit der Tabelle *Vfxpath.dbf* verwendet werden, nicht jedoch mit *Config.vfx*. Wenn ein leerer Wert übergeben wird, werden alle Mandantendatenbanken aus *Vfxpath.dbf* verarbeitet. Wenn der Wert nicht mit \$ beginnt, wird angenommen, dass es sich um einen Datenbankpfad handelt.

tcDBC – Name einer Datenbank.

ExcludeFiles(tcFileList, taFiles) – Diese Methode entfernt auszuschließende Dateien aus einem Array.

tcFileList – Eine Semikolon-separierte Liste der auszuschließenden Dateien.

taFiles – Ein Array mit einer Liste aller verfügbaren Dateien, aus denen *tcFileList* ausgeschlossen werden soll.

FillSourceList(taFiles,taTable, tlShowProgress) – Füllt ein Array mit Detailinformationen zu Dbf-Tabellen.

taFiles – Ein Array mit einer Liste von Dbf-Tabellen.

taTable – Ein Array mit Detailinformationen zu den Dbf-Tabellen in *taFiles*.

tlShowProgress – Wenn der Wert *.T.* übergeben wird, wird eine Fortschrittsanzeige angezeigt.

RunMaintenance(tcTable) – Durchführen der Datenbankwartung, Aufruf der Methode *Doit()*.

tcTable – Der Name der Tabelle, die den Parameter *tcAction* enthält. Wenn *tcAction* nicht den Namen einer Tabelle enthält, ist *tcTable* leer.

Neue Eigenschaften des Anwendungsobjekts goProgram

cAddFilesDownloadURL - Diese Eigenschaft enthält die URL der herunter zu ladenden und zu entpackenden Datei.

lCheckforDBUpdate – Mit dieser Eigenschaft kann eingestellt werden, ob die Struktur einer Datenbank zunächst überprüft werden soll, bevor die Aktualisierung der Struktur durchgeführt wird. Zur Prüfung sind keine Exklusivrechte erforderlich. Der Standardwert ist .F. Der Wert dieser Eigenschaft kann im VFX – Application Builder eingestellt werden. Die Aktualisierung der Struktur einer Datenbank kann nur durchgeführt werden, wenn Exklusivrechte vorhanden sind. Mit dieser Eigenschaft kann eine Prüfung der Struktur vorgenommen werden, ohne dass Exklusivrechte benötigt werden. Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird vor der Aktualisierung ohne Exklusivrechte überprüft, ob eine Aktualisierung überhaupt erforderlich ist.

lUseVfxFilterTable – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird die Tabelle *VfxFilter* zum Speichern der Filterdefinitionen verwendet. Wenn der Wert .F ist, wird die Tabelle *VfxRes* verwendet.

nXPdialogTotalTimeToSlideout – Gesamtzeit in Millisekunden, die benötigt wird, um den XP-Öffnendialog auszublenden (wenn das automatische Ausblenden aktiviert ist).

nWindowState – Einstellung von *_screen.windowstate* beim Start der Anwendung.

- 0 – Fenster
- 1 – minimiert
- 2 – maximiert

Neuheiten für Endbenutzer

Die Klasse *cComboPickList*

Diese Klasse ermöglicht die einfache, datengesteuerte Erstellung von Comboboxen. Basierend auf dieser Klasse können viele Auswahllisten erstellt werden, ohne für jede Auswahlliste eine eigene Tabelle erstellen zu müssen.

Die Klasse *cComboPicklist* verwendet die beiden VFX Systemtabellen *Vfxpdef.dbf* und *Vfxplist.dbf*.

Die Tabelle *Vfxpdef.dbf* enthält die Definitionen für die Auswahllisten. Für jede Auswahlliste ist ein Datensatz vorhanden. In den Definitionen kann Code hinterlegt werden, der bei jeder Auswahl des Benutzers ausgeführt wird. Dieser Code wird bei jeder Auswahl ausgeführt und ist nicht von der Auswahl eines bestimmten Eintrags abhängig. In der Tabelle *Vfxplist.dbf* kann für jeden Eintrag individueller Code eingetragen werden, der bei der Auswahl dieses Eintrags ausgeführt wird.

Die Einträge der Auswahllisten sind in der Tabelle *Vfxplist.dbf* definiert. Das Feld *Picklist* enthält den Schlüsselwert einer Auswahlliste und ist der Fremdschlüssel zur Tabelle *Vfxpdef.dbf*. Die Felder *Code* und *Descript* enthalten die Werte, die in der Auswahlliste angezeigt werden. Abhängig von der Definition der Auswahlliste wird entweder nur die Spalte *Code* oder die Spalte *Descript* oder es werden beide Spalten angezeigt. In das Feld *Proccode* kann Code eingetragen werden, der bei der Auswahl eines Eintrags ausgeführt wird.

Für jede Combobox basierend auf der Klasse *cComboPicklist* kann eingestellt werden, ob der Benutzer neue Einträge hinzufügen kann und welche Benutzerstufe hierfür ggf. als Berechtigung erforderlich ist.

Eigenschaften

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

cPickList – Enthält den Code aus dem Feld *Vfxpdef.Code*.

cProcCode – Enthält den Code aus dem Feld *Vfxpdef.ProcCode*.

lAutoAdjustColumnWidths – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird die Breite der Liste an die Breite des breitesten Eintrags angepasst.

lAutoGCode – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist und nur die Beschreibungsspalte sichtbar ist, wird der Wert für das Feld *Code* automatisch generiert.

lShowOnlyDescr – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird nur die Beschreibungsspalte angezeigt.

nUserLevel – Benutzerstufe, die erforderlich ist, um der Auswahlliste Einträge hinzufügen zu können.

nFieldLen – Nur intern verwendet. Enthält die Feldlänge der aktuellen Auswahlliste.

Methoden

AddNewCode – Diese Methode wird ausgeführt, wenn der Benutzer der Auswahlliste einen neuen Wert hinzufügt. Wenn hierbei zusätzlicher weiterer Code benötigt wird, kann der Code in dieser Methode platziert werden.

FillItems – Diese Methode füllt die Combobox mit den Einträgen aus der Tabelle *Vfxplist.dbf*.

SetComboboxSettings – Diese Methode wird bei der Initialisierung der Auswahlliste ausgeführt. Hier werden die Eigenschaften der Combobox entsprechend den Definitionen in der Tabelle *Vfxpdef.dbf* und entsprechend den Rechten des aktuellen Benutzers gemacht.

Für Steuerelemente basierend auf der Klasse *cComboPicklist* können an zwei Stellen Code-Teile hinterlegt werden. In der Tabelle *Vfxpdef* im Memofeld *ProcCode* und in der Tabelle *Vfxplist* im Memofeld *ProcCode*.

Der Code im Feld *Vfxpdef.ProcCode* wird jedes Mal ausgeführt, wenn ein Eintrag in der Combobox ausgewählt wird. Der Code im Feld *Vfxplist.ProcCode* gehört zu einem Eintrag in der Combobox und wird ausgeführt, wenn dieser Eintrag ausgewählt wird.

Für jeden Eintrag in der Tabelle *Vfxplist.dbf* kann definiert werden, ob der Eintrag aktiv ist. Dadurch ist es nicht erforderlich nicht mehr benötigte Einträge zu löschen und so möglicherweise verwaiste Datensätze in der Datenbank zu hinterlassen. Wenn ein Eintrag nicht mehr benötigt wird, kann der Wert des Feldes *Active* auf .F. gesetzt werden.

Verwaltung von Auswahllisten

Code	Beschreibung	Auswahlliste	aktiv	Wert
00000000000000000001	Mobiltelefon	Kommunikation	<input checked="" type="checkbox"/>	
00000000000000000002	Telefon	Kommunikation	<input checked="" type="checkbox"/>	
00000000000000000003	Telefax	Kommunikation	<input checked="" type="checkbox"/>	
00000000000000000004	E-Mail	Kommunikation	<input type="checkbox"/>	
00000000000000000005	Telex	Kommunikation	<input type="checkbox"/>	

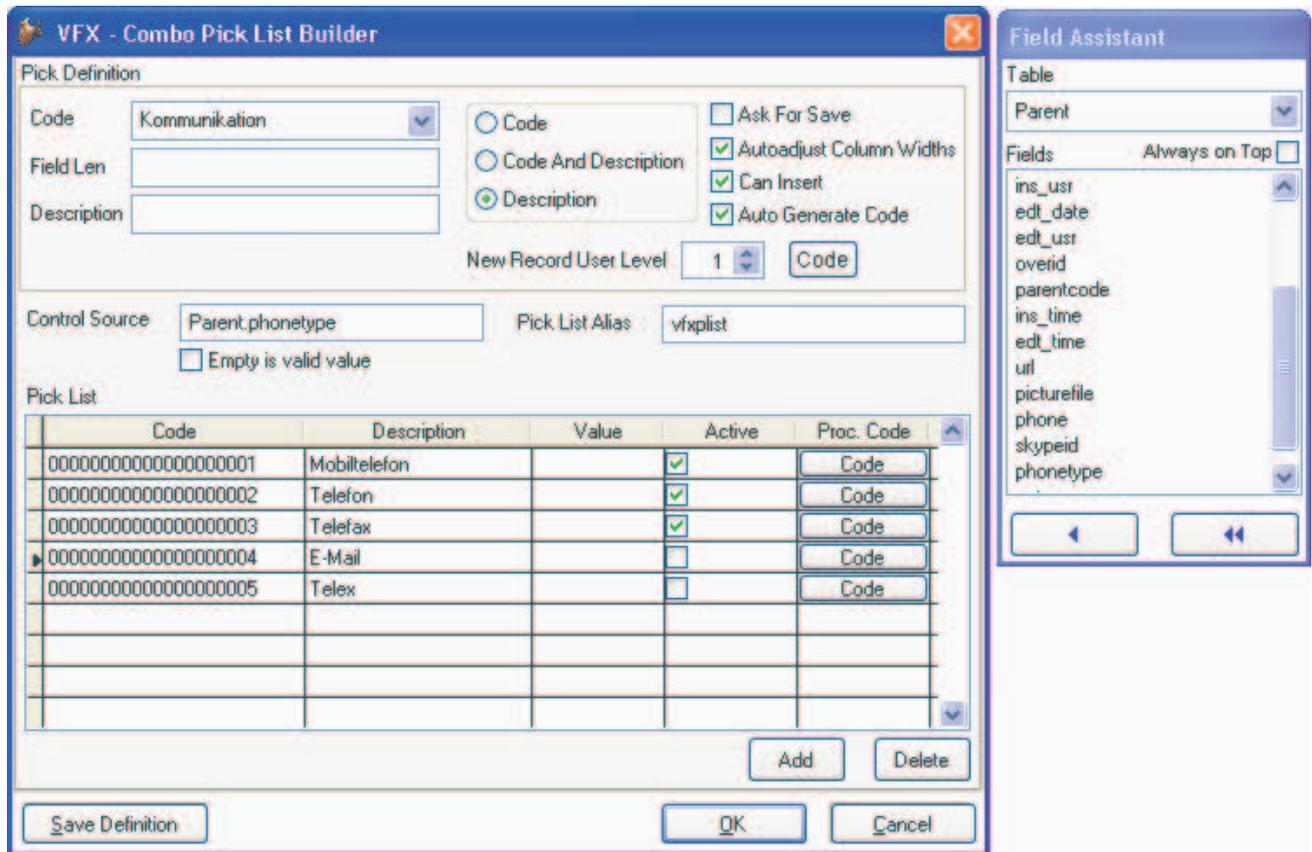
00000000000000000001	Mobiltelefon	Kommunikation	<input checked="" type="checkbox"/>	
----------------------	--------------	---------------	-------------------------------------	--

Mit diesem Formular hat der Benutzer die Möglichkeit die Werte in Auswahllisten zu bearbeiten. Das Formular Zu diesem Zweck ist das Formular *VFXPlist.scx* in Projekte eingeschlossen.

Benutzer können die vollständige Liste für alle Auswahllisten anzeigen und bearbeiten oder die Anzeige für eine Auswahlliste filtern. Es ist möglich Einträge zu löschen. Es ist jedoch empfehlenswert Einträge als inaktiv zu kennzeichnen, um verwaiste Datensätze in der Datenbank zu vermeiden.

VFX – Combo Pick List Builder

Die Einstellungen von Steuerelementen basierend auf der Klasse *cComboPicklist* sowie die Inhalte der Tabellen *Vfxpdef.dbf* und *Vfxplist.dbf* können mit dem VFX – Combo Pick List Builder bearbeitet werden.



Für Steuerelemente basierend auf der Klasse *cComboPicklist* müssen die *ControlSource* und der Arbeitsbereich für die *RowSource* angegeben werden. Wenn sich in der Datenumgebung des Formulars bereits der Aliasname befindet, der für die *RowSource* gewählt wird, erscheint eine Messagebox. Es kann gewählt werden, ob der vorhandene Aliasname verwendet werden soll, oder ob ein neuer Cursor der Datenumgebung hinzugefügt werden soll. Wenn der Aliasname für die *RowSource* nicht in der Datenumgebung gefunden wird, wird ein neuer Cursor erstellt und alle Eigenschaften werden automatisch eingestellt.

Wahlweise können Steuerelemente basierend auf der Klasse *cComboPicklist* nur die Beschreibung anzeigen. Wenn in diesem Fall die Option *Can Insert* auf *.T.* eingestellt ist, wird die Option *Auto Generate Code* automatisch auf *.T.* eingestellt. Wenn die Option *Auto Generate Code* auf *.T.* eingestellt ist und wenn ein Eintrag der Auswahlliste hinzugefügt wird, wird für das Feld *Code* der nächste aufsteigende Wert automatisch zugewiesen. In diesem Fall ist die Spalte *Code* im VFX – Combo Pick List Builder und im Formular „Verwaltung von Auswahllisten“ schreibgeschützt. Die Länge des generierten Codes wird durch die Angabe im Feld *Field Len* bestimmt. Wenn dieser Wert leer ist, wird die Länge durch die Breite des Feldes bestimmt, das als *ControlSource* angegeben ist.

Die Option *Auto Generate Code* steht nur dann zur Verfügung, wenn in der Optionsgruppe *Description* markiert ist. In allen anderen Fällen muss der Code manuell eingegeben werden.

Mit einem Klick auf die Schaltfläche *Code* wird ein Programmbearbeitungsfenster geöffnet. Der hier eingetragene Code gilt allgemein für die *cComboPicklist* und wird immer ausgeführt, wenn der Benutzer einen anderen Eintrag ausgewählt.

Mit einem Klick auf die Schaltfläche *Code* hinter einem Listeneintrag wird ein Programmbearbeitungsfenster geöffnet. Der hier eingetragene Code wird nur dann ausgeführt, wenn der Benutzer wenn dieser Eintrag ausgewählt wird.

Die Klasse *cDateTextBox*

Die Klasse *cDateTextBox* ist eine Textbox zur Eingabe von Datums- oder Datetime-Werten. Mit einem Doppelklick oder durch Betätigung der Funktionstaste wird ein Kalender angezeigt. Die *cDateTextBox* ist zur Verwendung im Suchdialog oder in Childgrids gedacht, wenn als Datenquelle ein Feld vom Typ *Date* oder *Datetime* verwendet wird.

In der Textbox können die folgenden Tasten zur schnellen Auswahl eines Datums verwendet werden:

+, =	nächster Tag
-, _	vorhergehender Tag
H, h	heute
B, b	Erster Tag des aktuellen Monats
L, l	Letzter Tag des aktuellen Monats
A, a	Neujahr
E, e	Silvester
V, v	vorhergehender Monat
N, n	nächster Monat

Die Tasten sind in der Konstanten *MSG_CPICKDATEHOTKEYS* definiert.

Wenn dem Steuerelement eine *ControlSource* vom Typ *DateTime* zugewiesen ist, wird der Zeitanteil durch Verwendung der oben angegebenen Tasten nicht verändert, ausgenommen sind die Tasten für Neujahr und Silvester.

Zur Anzeige des Kalenders wird das Microsoft MonthView ActiveX-Steuerelement verwendet. Wenn ein Setup zur Distribution einer Anwendung erstellt werden soll, muss dieses ActiveX-Steuerelement (Mscomct2.ocx) in das Setup eingeschlossen werden. Hierfür wird mit Installshield Express für Visual FoxPro ein Merge Module geliefert.



Die Klasse *cDocumentManagement*

Drag und Drop in *cDocumentManagement*

In der Klasse *cDocumentManagement* wird Drag und Drop unterstützt. Zur Laufzeit kann ein neuer Datensatz angelegt werden, in dem die gewünschte Datei aus dem Windows-Explorer auf den *cDocumentManagement* Container gezogen und fallengelassen wird. Außerdem ist es möglich Einträge aus Outlook in den *cDocumentManagement* Container zu ziehen. Auf diesem Weg können Einträge für E-Mails, Aufgaben, Kontakte oder Ordner erstellt werden.

Zur Unterstützung der Drag und Drop Funktionalität mit Outlook hat die Klasse *cDocumentManagement* spezielle Eigenschaften.

cDocumentTypeFieldName – Name des Feldes in der Dokumententabelle, in dem der Dokumenttyp gespeichert wird. Der Dokumenttyp wird intern verwendet. In diesem Feld können vier verschiedene Werte für den Dokumenttyp gespeichert werden: 0 – (Standardwert, wenn Datensätze mit der Schaltfläche *neu* hinzugefügt werden) Datei oder RTF Dokument, 1 – E-Mail aus Outlook, 2 – Aufgabe aus Outlook, 3 – Kontakt aus Outlook, 4 – Ordner aus Outlook. Der Standardwert für den Feldnamen ist *DocType* (siehe *VfxDocuments.dbf*),

cEntryIDFieldName – Name des Feldes in der Dokumententabelle, in dem der Schlüssel zu einem Element aus Outlook gespeichert wird. Diese Eigenschaft wird nur Elemente aus Outlook verwendet. Der Standardwert für den Feldnamen ist *EntryID* (siehe *VfxDocuments.dbf*).

Wenn der Wert einer der Eigenschaften *cDocumentTypeFieldName* oder *cEntryIDFieldName* leer ist, wird die Drag und Drop Funktionalität mit Outlook ausgeschaltet.

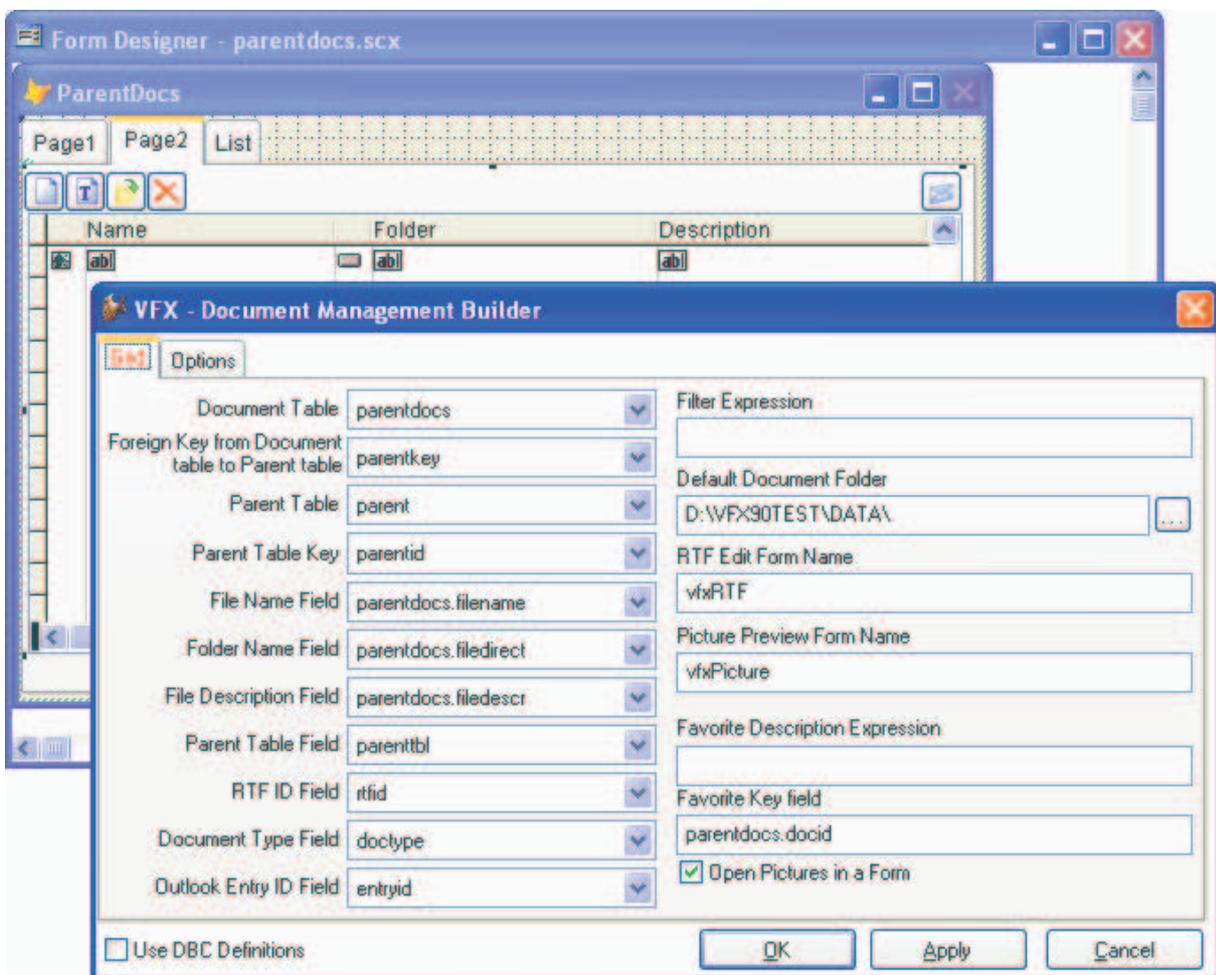
Dokumente aus dem *cDocumentManagement* Container können als Favoriten gespeichert werden. Über den Menüpunkt *Zu Favoriten hinzufügen* wird für das aktuelle Dokument ein Favoriteneintrag angelegt. Wenn der Favoriteneintrag geöffnet wird, wird der Satzzeiger auf den entsprechenden Datensatz im *cDocumentManagement* Container positioniert.

Das Ergebnis der Evaluierung aus dem Wert der Eigenschaft *cFavoriteDescr* wird als Menüeintrag im Favoritenmenü verwendet. Wenn der Wert dieser Eigenschaft leer ist, wird automatisch ein Text für den Menüeintrag generiert. Hierfür wird die Beschreibung für einen Favoriten auf Formular-ebene verwendet und es werden die ersten zehn Zeichen der Dateibeschreibung angehängt, wenn diese nicht leer ist. Wenn die Dateibeschreibung leer ist, wird der Dateiname angehängt, wie das folgende Beispiel zeigt:



VFX – Document Management Builder

Alle Eigenschaften der Klasse *cDocumentManagement* können im VFX – Document Management Builder eingestellt werden.



Im VFX – Document Management Builder werden alle Auswahlfelder für Feldnamen automatisch mit Standardwerten gefüllt, wenn die Tabelle *VfxDocuments* als Dokumenttabelle gewählt wird. Die Feldnamen werden auch dann automatisch eingetragen, wenn sich in der gewählten Dokumenttabelle die gleichen Feldnamen befinden, die auch in *VfxDocuments* verwendet werden.

In der Textbox *Favorite Key Field* muss ein Ausdruck eingetragen werden, der für die Eigenschaft *cFavoriteID* des Childgrid verwendet wird. Normalerweise wird hier Name des Primärschlüsselfeldes der Dokumenttabelle eingetragen. Dieser Wert wird vom VFX – Document Management Builder in der Regel automatisch richtig erkannt.

In der Textbox *Favorite Description Expression* muss ein Ausdruck eingetragen werden, der für die Eigenschaft *cFavoriteDescr* des Childgrid verwendet wird.

Die Klasse *cChildGrid*

Unterstützung von Favoriten

Child-Datensätze können jetzt den Favoriten hinzugefügt werden. Wenn sich der Fokus auf einem Child-Datensatz befindet, kann der Benutzer im Menü *Zu Favoriten hinzufügen* wählen. Der Favoriteneintrag für einen Child-Datensatz erscheint im Favoriten Menü in der Favoritengruppe für das Formular.

Eigenschaften

cFavoriteID – Ähnlich der Eigenschaft *cFavoriteID* von Formularen, enthält jedoch den Namen des Primärschlüsselfeldes der Child-Tabelle.

cFavoriteDescr – Ähnlich der Eigenschaft *cFavoriteDescr* von Formularen, enthält jedoch den Ausdruck für die Beschreibung für die Child-Tabelle. Dieser Ausdruck wird evaluiert und dient als Menüeintrag im Favoritenmenü.

Methode

ViaFavorite – Ähnlich der Methode *ViaFavorite* von Formularen, jedoch wird zusätzlich zum Öffnen des Formulars und Positionieren auf den Parent-Datensatz auch auf den gewünschten Child-Datensatz positioniert.

VFX – CChildGrid Builder

Im VFX – CChildGrid Builder sowie in allen Formular Buildern, die Childgrids unterstützen (cOneToMany Builder, cOneToManyPageFrame Builder und cTreeViewOneToMany Builder), kann für jede Spalte eingestellt werden, ob inkrementelle Suche erlaubt ist. Wenn die Child-Daten auf einem parametrisierten Cursoradapter oder auf einer Ansicht basieren, kann den Benutzern ermöglicht werden Daten im Grid zu bearbeiten und gleichzeitig kann eine Sortierung durch einen Klick auf die Überschrift zugelassen werden.

Incremental Search. Wenn diese Option markiert ist, ist inkrementelle Suche möglich. Durch Auswahl dieser Option wird die Spalte automatisch als schreibgeschützt markiert und es wird die Markierung bei *Allow Sort* gesetzt.

Allow Sort. Wenn diese Option markiert ist, kann der Benutzer nach diese Spalte durch einen Klick auf die Überschrift sortieren. Wenn diese Markierung gelöscht wird, wird auch die Markierung bei *Incremental Search* entfernt.

Positionierung des Satzzeigers in Onetomany Formularen

In den Child-Aliasbereichen von Formularen basierend auf der Klasse *Conetomany* wird vor dem Wechsel in den Bearbeitungsmodus die Position des Satzzeigers in der Formulareigenschaft *aChildAliasRecno[1,1]* gespeichert. Nach Abschluss der Bearbeitung durch Speichern oder Verwerfen wird in jedem Aliasbereich der Satzzeiger auf den ursprünglichen Datensatz positioniert.

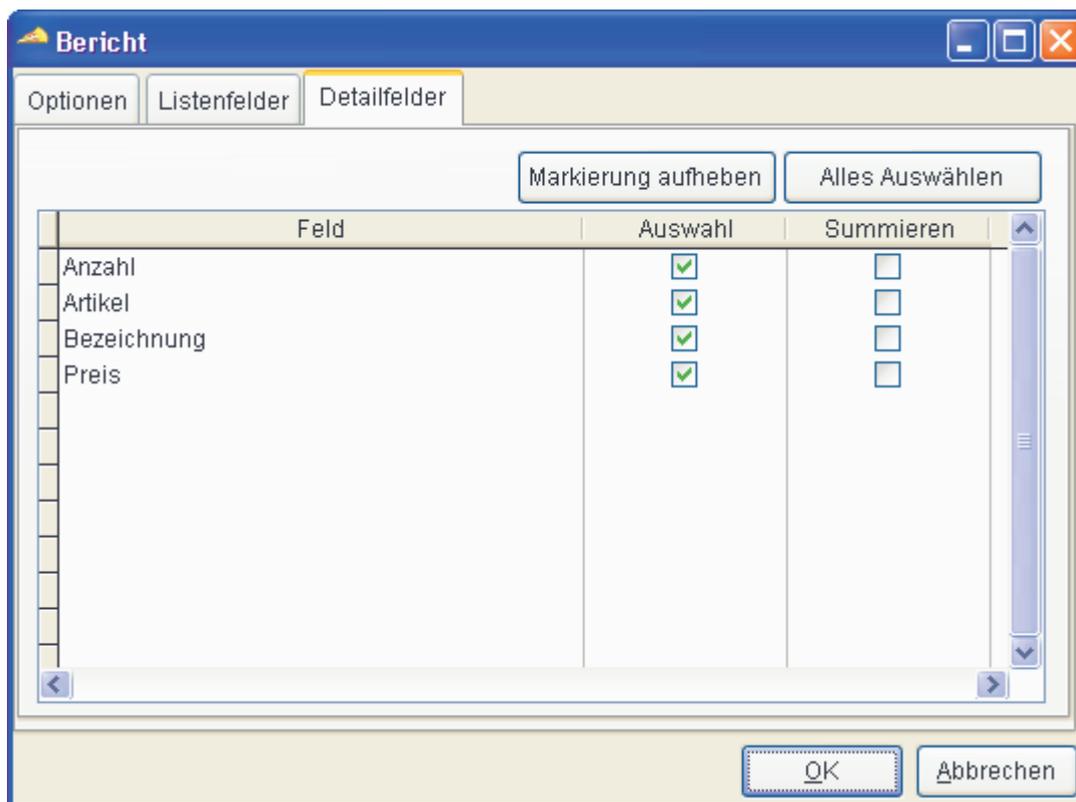
Berichte

In Formularen basierend auf der Klasse *OneToMany* oder *OneToManyPageFrame* können Berichte erstellt werden, die Detaildaten enthalten.

Im VFX – *cOneToMany* Builder und in VFX – *cOneToManyPageFrame* Builder kann auf der Seite *Report* in der Combobox *Band*, für jedes Feld ausgewählt werden, ob es im Parent-Band oder im Child-Band gedruckt werden soll.

Wenn das Kontrollkästchen *Use Grid Fields For Report* markiert ist, können die zu druckenden Felder nicht angegeben werden. In diesem Fall werden die Felder aus dem Arbeitsbereich *cWorkAlias* im Parent-Band gedruckt. In Formularen basierend auf der Klasse *COneToMany* werden im Child-Band alle Felder aus dem Childgrid von der ersten Child-Seite gedruckt. Wenn ein Formular auf der Klasse *COneToManyPageFrame* basiert, werden im Child-Band die Felder von der ersten Seite mit einem Childgrid gedruckt.

Im Berichtsdialog werden auf der Seite *Detailbereiche* alle Felder angezeigt, die im Child-Band gedruckt werden können. Die Felder, die im Parent-Band gedruckt werden, können auf der Seite *Feldauswahl* ausgewählt werden. Die Seite *Detailbereiche* wird nur bei Formularen basierend auf der Klasse *COneToMany* oder *COneToManyPageFrame* angezeigt. Der Benutzer kann auswählen welche Felder gedruckt werden sollen und für welche Spalten eine Summe gebildet werden soll.



Hier ein Beispiel für einen OneToMany Bericht:

Auftrag			Seite	1	Datum	11.10.06
					Zeit	15:05:33
<i>Auftragsdatum</i>		<i>Liefern an</i>				
04.07.96		Vins et alcools Chevalier				
<i>Anzahl</i>	<i>Artikel</i>	<i>Bezeichnung</i>	<i>Preis</i>			
	12 PCardi	Pizza Cardinale - To	6,40			
	10 Apfel	Apfelsaft	1,50			
	5 Tomate	Tomatensalat	2,10			
<i>Auftragsdatum</i>		<i>Liefern an</i>				
05.07.96		Toms Spezialitäten				
<i>Anzahl</i>	<i>Artikel</i>	<i>Bezeichnung</i>	<i>Preis</i>			
	9 Pveget	Pizza vegetarisch -	6,80			
	40 Jägerm	Jägermeister	1,80			
<i>Auftragsdatum</i>		<i>Liefern an</i>				
08.07.96		Hanari Carnes				
<i>Anzahl</i>	<i>Artikel</i>	<i>Bezeichnung</i>	<i>Preis</i>			
	10 Orange	Orangensaft	1,40			
	35 Jägerm	Jägermeister	1,80			
	15 Tirami	Tiramisu	2,80			

Wenn im Berichtsdialog auf der Seite Detailbereich keine Felder ausgewählt werden, wird ein Standardbericht ohne Child-Band erstellt.

Scannen von Bildern

In VFX Anwendungen können einfach und benutzerfreundlich Bilder gescannt werden. Hierfür gibt es zwei Klassen und zwei Funktionen in der VFX.fl. Die Klasse *cScannerControl* basiert auf der VFP Basisklasse *Custom*. Diese Klasse dient als Wrapper für die Scan-Funktion in der VFX.fl. Die Klasse *cScanForm* ist eine Formulareklasse und bietet die Benutzerschnittstelle zur Vorschau und zum Scannen von Bildern.

Die Klasse cScannerControl

Die Klasse *cScannerControl* befindet sich in der Klassenbibliothek *Vfxctrl.vcx*. Durch das Einstellen von wenigen Eigenschaften können Bilder programmgesteuert gescannt werden. Ein Beispiel zur Verwendung dieser Klasse befindet sich in der Demoanwendung VFX95Test im Formular *Item*.

Eigenschaften

cFileName – Vollständiger Pfad- und Dateiname der zu scannenden Bilddatei.

cFileFormat – Namensweiterung des Dateiformats. Der Wert kann *BMP*, *JPG*, *GIF*, *PNG* oder *TIFF* sein. Der Standardwert ist *TIFF*.

cRunOnFailure – Dieser Eigenschaft kann der Name einer Methode zugewiesen werden die aufgerufen wird, wenn ein Scannvorgang fehlgeschlagen ist. Die Methode sollte eine geeignete Fehlerbehandlung durchführen oder dem Benutzer eine Meldung anzeigen. Der Standardwert ist *This.ScanFailed()*.

cRunOnSuccess – Dieser Eigenschaft kann der Name einer Methode zugewiesen werden die aufgerufen wird, wenn ein Scannvorgang erfolgreich durchgeführt wurde. Die Methode sollte eine geeignete Meldung anzeigen. Der Standardwert ist *This.ScanSuccessful()*.

nScanResult – Diese Eigenschaft enthält den Rückgabewert der Funktion *GetTWAINImage()*. Der Wert dieser Eigenschaft kann in einer eigenen Methode verwendet werden, um eine detailliertere Fehlerbehandlung durchzuführen.

lUseDefaultTWAINSource – Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt ist, wird die Standard TWAIN Quelle verwendet. Der Standardwert ist *.T.*

cTWAINSource – Zeichenkette mit dem Namen der zu verwendenden TWAIN Quelle. Der Standardwert ist eine leere Zeichenkette. Wenn der Wert der Eigenschaft *lUseDefaultTWAINSource* auf *.F.* eingestellt ist und *cTWAINSource* leer ist und mehr als ein TWAIN Gerät angeschlossen ist, erscheint ein Dialog zur Auswahl des TWAIN Geräts.

lDisplayAdvancedSettingsDialog – Abhängig von der verwendeten TWAIN Quelle, steht der erweiterte Einstellungsdialog möglicherweise nicht zur Verfügung. Der Standardwert ist *.F.*

lDisplayProgressDialog – Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt ist, wird eine Fortschrittsanzeige angezeigt. Der Standardwert ist *.T.*

lDisplayMessages – Wenn der Wert dieser Eigenschaft auf *.T.* eingestellt ist, wird nach dem Beenden des Scan-Vorgangs eine Meldung angezeigt. Der Standardwert ist *.T.*

Methoden

GetTWAINSources – Aufruf der Funktion *GetTWAINSources* in der *VFX.FLL*.

Rückgabewert: Der Rückgabewert der Funktion *GetTWAINSources()*.

0 – Der Vorgang wurde erfolgreich abgeschlossen.

nErrorNumber – Bei Auftreten eines Fehlers wird die Fehlernummer zurückgegeben.

GetTWAINImage – Aufruf der Funktion *GetTWAINImage* in der *VFX.FLL*.

Rückgabewert: Der Rückgabewert der Funktion *GetTWAINImage()*.

0 – Der Vorgang wurde erfolgreich abgeschlossen.

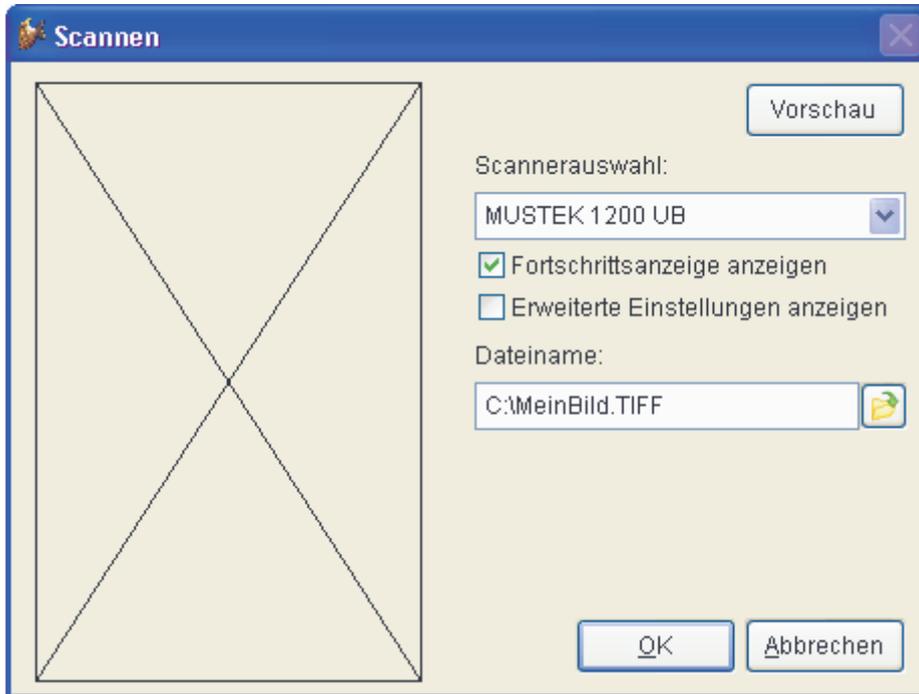
nErrorNumber – Bei Auftreten eines Fehlers wird die Fehlernummer zurückgegeben.

ScanFailed – Anzeige einer Fehlermeldung, wenn der Scannvorgang fehlgeschlagen ist.

ScanSuccessful – Anzeige einer Meldung, wenn der Scannvorgang erfolgreich war.

Die Klasse cScanForm

Die Formulkasse *cScanForm* bietet einen Scan-Dialog für Endbenutzer. Diese Klasse befindet sich in der Klassenbibliothek *Vfxformbase.vcx*. Diese Klasse kann programmatisch verwendet werden oder als modaler Dialog gestartet werden.



Wenn diese Klasse als modaler Dialog verwendet wird, wird als Rückgabewert der vollständige Pfadname des gescannten Bildes geliefert oder eine leere Zeichenkette, wenn der Vorgang abgebrochen wurde. Ein Beispiel zur Verwendung dieser Klasse befindet sich in der Demoanwendung VFX95Test im Formular *Item2*.

Eigenschaften

cImageFile – Vollständiger Pfad- und Dateiname des gescannten Bildes. Wenn abgebrochen wird, wird eine leere Zeichenkette zurückgegeben.

cTempImageFile – Nur intern verwendet. Hier wird der Pfad zu der temporären Datei gespeichert, die für die Vorschau verwendet wird.

lDoNotReleaseForm – Wenn der Wert dieser Eigenschaft auf *.T.* gestellt wird, wird das Formular nach einem Klick auf *OK* oder *Abbrechen* nicht geschlossen, sondern versteckt. Dies ist für die programmatische Verwendung des Formulars nützlich.

lScanned – Nur intern verwendet.

Methoden

Init(tlShowProgress, tlAdvancedSettings, tcFileName, tcTwainSource, tlDoNotReleaseForm)

Die Parameter werden beim Ausführen des Formulars ausgewertet und setzen die Formulareigenschaften.

tlShowProgress – Mit diesem Parameter wird die Eigenschaft *IDisplayProgressDialog* des Objekts *cScannerControl* eingestellt.

tlAdvancedSettings – Mit diesem Parameter wird die Eigenschaft *IDisplayAdvancedSettingsDialog* des Objekts *cScannerControl* voreingestellt.

tcFileName – Mit diesem Parameter wird der Wert des Objekts *cfileselector* voreingestellt.

tcTwainSource – Mit diesem Parameter wird der Anzeigewert des Auswahlfeldes für TWAIN Geräte voreingestellt.

tlDoNotReleaseForm – Mit diesem Parameter wird der Wert der Eigenschaft *IDoNotReleaseForm* des Formulars voreingestellt.

Scan(tcFileName, tcFileFormat) – Scannt ein Bild und speichert es in einer Datei. Diese Methode erwartet einen vollständigen Pfad- und Dateinamen sowie das Bilddateiformat als Parameter.

tcFileName – Vollständiger Pfad- und Dateiname. Hier wird das Bild gespeichert.

tcFileFormat – Bilddateiformat, in dem das gescannte Bild gespeichert wird. Mit dem Wert dieser Eigenschaft wird die Eigenschaft *cFileFormat* des Objekts *cScannerControl* eingestellt.

Rückgabewert: .T. – Wenn der Scann-Vorgang erfolgreich beendet wurde.

SaveLastFolder(tcExportFolderType) – Speichert den zuletzt für Bilddateien verwendeten Ordner in der Ressourcendatei.

tcExportFolderType – Zur internen Verwendung.

Zum Scannen benötigte Funktionen in der VFX.FLL

GetTWAINSources(@tcTwainSources) – Liefert eine Liste der verfügbaren TWAIN Geräte, die an den Computer angeschlossen sind.

tcTwainSources – Eine Zeichenkette, die per Referenz übergeben wird. In dieser Zeichenkette wird eine mit Semikolon separierte Liste verfügbarer TWAIN Geräte zurückgegeben.

Rückgabewert: 0 – Wenn die Operation erfolgreich abgeschlossen werden konnte.

nErrorNumber – Wenn ein Fehler aufgetreten ist. Siehe TWAIN Fehlermeldungen.

GetTWAINImage(tcFileName, tcFileFormat, tlUseDefaultTWAINSource, tcTWAINSource, tlDisplayAdvancedSettingsDialog, tlDisplayProgressDialog) – Scannt ein Bild und speichert es in der Datei *tcFileName*.

tcFileName – Vollständiger Pfad- und Dateiname der Datei, in der das gescannte Bild gespeichert werden soll.

tcFileFormat – Namensweiterung des Dateiformats. Der Wert kann *BMP*, *JPG*, *GIF*, *PNG* oder *TIFF* sein. Wenn dieser Parameter leer ist, ist der Standardwert *TIFF*.

tlUseDefaultTWAINSource – Wenn dieser Wert .T. ist, wird das Standard TWAIN Gerät verwendet. Wenn dieser Parameter leer ist, ist der Standardwert .T.

tcTWAINSource – Zeichenkette mit der zu verwendenden TWAIN Quelle. Wenn der Wert von *tlUseDefaultTWAINSource* .F. ist und wenn der Wert der Eigenschaft *tcTWAINSource* leer ist und wenn mehr als ein TWAIN Gerät angeschlossen ist, wird ein Dialog zur Auswahl des TWAIN Geräts angezeigt.

tlDisplayAdvancedSettingsDialog – Wenn dieser Wert .T. ist, wird ein Dialog mit erweiterten Einstellungen angezeigt. Abhängig von der verwendeten TWAIN Quelle, steht dieser

Dialog möglicherweise nicht zur Verfügung. Wenn dieser Parameter leer ist, ist der Standardwert ist .F.

tlDisplayprogressDialog – Wenn dieser Wert .T. ist, wird eine Fortschrittsanzeige angezeigt.

Wenn dieser Parameter leer ist, ist der Standardwert ist .T.

Rückgabewert: 0 – wenn der Vorgang erfolgreich durchgeführt werden konnte

nErrorNumber – wenn ein Fehler aufgetreten ist, TWAIN Fehlermeldung

Liste der TWAIN Fehlermeldungen

0x00000000 - 0x0000FFFF – Allgemeine Fehler

0x00000000 - Erfolg (Das Bild wurde erfolgreich gespeichert)

0x00000001 - Abbruch durch den Benutzer

0x00000002 - Die TWAIN Bibliothek konnte nicht geladen werden

0x00000003 - Die Initialisierung von GDI+ ist fehlgeschlagen

0x00000004 - Es wurde ein ungültiges Bildformat angegeben

0x00000005 - Nicht genügend Speicher verfügbar

0x00000006 - Ungültige Quelle angegeben

0x00000007 - GDI+ Fehler

0x00000008 - Ungültige Zeichen im Dateinamen (Unicode Konvertierung fehlgeschlagen)

0x00000009 - Fehler beim Speichern der Datei

0x00010000 - 0x0001FFFF - TWAIN spezifische Fehler (MSG_OPENDSM)

0x00010000 - Der Datenquellen-Manager kann nicht geöffnet werden

0x00010001 - Fehler aus unbekanntem Gründen

0x00010002 - Nicht genügend Speicher zur Durchführung der Operation

0x00010005 - DSM Fehler

0x00020000 - 0x0002FFFF - TWAIN spezifische Fehler (MSG_USERSELECT)

0x00020000 - Datenquelle kann nicht ausgewählt werden

0x00020001 - Fehler aus unbekanntem Gründen

0x00020002 - Nicht genügend Speicher zur Durchführung der Operation

0x00020005 - DSM Fehler

0x00030000 - 0x0003FFFF - TWAIN spezifische Fehler (MSG_GETFIRST)

0x00030000 - Die Aufzählung der Datenquellen kann nicht gestartet werden

0x00030001 - Fehler aus unbekanntem Gründen

0x00030002 - Nicht genügend Speicher zur Durchführung der Operation

0x00030005 - DSM Fehler

0x00040000 - 0x0004FFFF - TWAIN spezifische Fehler (MSG_GETNEXT)

0x00040000 - Die Aufzählung der Datenquellen kann nicht fortgesetzt werden

0x00040001 - Fehler aus unbekanntem Gründen

0x00040002 - Nicht genügend Speicher zur Durchführung der Operation

0x00040005 - DSM Fehler

0x00050000 - 0x0005FFFF - TWAIN spezifische Fehler (MSG_OPENDS)

0x00050000 - Datenquelle kann nicht geöffnet werden

0x00050001 - Fehler aus unbekanntem Grund
0x00050002 - Nicht genügend Speicher zur Durchführung der Operation
0x00050003 - Keine Datenquelle gefunden
0x00050004 - Die Datenquelle ist mit der maximal möglichen Anzahl von Anwendungen verbunden
0x00050005 - DSM oder DS Fehler

0x00060000 - 0x0006FFFF - TWAIN spezifische Fehler (CAP_XFERCOUNT)

0x00060000 - Anzahl der Bilder zur Übertragung kann nicht eingestellt werden (Wert 1)
0x00060001 - Fehler aus unbekanntem Grund
0x00060002 - Nicht genügend Speicher zur Durchführung der Operation
0x00060005 - DSM oder DS Fehler
0x00060006 - Unbekannte Eigenschaft
0x0006000A - Datenparameter außerhalb des gültigen Bereichs
0x0006000D - Eigenschaft von der Quelle nicht unterstützt
0x0006000E - Operation nicht unterstützt
0x0006000F - Eigenschaft hängt von anderen Eigenschaften ab

0x00070000 - 0x0007FFFF - TWAIN spezifische Fehler (MSG_ENABLEDS)

0x00070000 - Datenquelle kann nicht verwendet werden
0x00070001 - Fehler aus unbekanntem Grund
0x00070002 - Nicht genügend Speicher zur Durchführung der Operation
0x00070005 - DSM oder DS Fehler

0x00080000 - 0x0008FFFF - TWAIN spezifische Fehler (DAT_IMAGENATIVEXFER)

0x00080000 - Bildübertragung kann nicht durchgeführt werden
0x00080001 - Fehler aus unbekanntem Grund
0x00080002 - Nicht genügend Speicher zur Durchführung der Operation
0x00080005 - DSM oder DS Fehler

0x00090000 - 0x0009FFFF - TWAIN spezifische Fehler (MSG_ENDXFER)

0x00090000 - Bildübertragung kann nicht abgeschlossen werden
0x00090001 - Fehler aus unbekanntem Grund
0x00090002 - Nicht genügend Speicher zur Durchführung der Operation
0x00090005 - DSM oder DS Fehler