



VFX 9.5 – Was ist neu? 1. Quartal 2006

März 2006



Uwe Habermann, Venelina Jordanova

Inhaltsverzeichnis

Neuheiten für Entwickler.....	3
Vererbungsarchitektur	3
Vfxformbase.vcx	3
Die Klasse cDataMgr (Vfxappl.vcx)	4
Builder	6
Formular Builder	6
VFX - Language Management Builder	6
VFX – Help Wizard.....	8
Die Tabelle Vfxfopen	8
Neue Eigenschaften des Anwendungsobjekts.....	8
Versenden eines Fehlerberichts an einen Web Service	9
VFX – Kundenverwaltung	11
Neue Eigenschaften für Endanwender	12
Benutzerverwaltung	12
Liste der erlaubten/verbotenen IP Adressen für die Anmeldung	12
Verwaltung von Sicherheitsrechten.....	13
Erweiterungen der Benutzeroberfläche	15
XP Öffnen Dialog automatisch ausblenden	15
Unterstützung geringer Farbtiefen.....	15
RTF Unterstützung in Formularen (VfxCtrl.vcx, cRTFControl)	16
RTF Unterstützung in Berichten	17
cMemoform für Steuerelemente aus der Klasse cEditbox.....	17
Anpassen-Dialog	18
Suchdialog.....	18

Neuheiten für Entwickler

Vererbungsarchitektur

Vfxformbase.vcx

Die Vererbungsarchitektur der VFX-Klassen wurde erweitert. In bisherigen Versionen von VFX konnte auf die Funktionalität und das Layout nur mit Hooks Einfluss genommen werden. Wenn der Entwickler beispielsweise in der gesamten Anwendung einen anderen Zeichensatz verwenden wollte, konnte dies nur mit Hooks im *Init* Ereignis erreicht werden. Dies hatte den Nachteil, dass die Formulare in den VFP Designern immer mit dem VFX Standardzeichensatz Arial angezeigt wurden, und dass der durch Hooks eingestellte Zeichensatz nur zur Laufzeit der Formulare sichtbar war.

In VFX 9.5 gibt es nun eine Zwischenschicht in der Vererbung. Die VFX Formularklassen, die sich in bisherigen Versionen von VFX in der Klassenbibliothek *Vfxform.vcx* befanden, sind jetzt in der Klassenbibliothek *Vfxformbase.vcx*. An die bisherigen Klassennamen wurde der Zusatz *vfxbase* angefügt.

Für jede Klasse aus dieser Bibliothek aus der ersten Schicht, gibt es eine Ableitung in die Klassenbibliothek *Vfxform.vcx*. Dadurch steht die Klassenbibliothek *Vfxform.vcx* Entwicklern für eigene Erweiterungen zur Verfügung. Hier kann zum Beispiel der Zeichensatz in einer Klasse geändert werden. Diese Einstellung betrifft dann alle Formulare, die auf dieser Klasse basieren in der gesamten Anwendung.

Bei einer Aktualisierung des Projekts mit dem VFX - Update Project Wizard wird die Klassenbibliothek *Vfxform.vcx* nicht mehr aktualisiert, wenn die Klassenbibliothek *Vfxformbase.vcx* existiert.

Die Klasse *cDataMgr* (Vfxappl.vcx)

Die Klasse *cDataMgr* ist eine neue Zwischenschicht zwischen der Klasse *cDataTableMgr* und den Formularen der Anwendung.

Diese Klasse hat die gleichen Methoden und Eigenschaften wie die Klasse *cDataTableMgr* aus *Vfxctrl.vcx*. Es werden die gleichen Parameter an die Methoden wie in *cDataTableMgr* übergeben. Die Eigenschaften besitzen Assign- und Access-Methoden.

```
Init()  
This.AddObject("oDataTableMgr","CDataTableMgr")
```

```
cDataMgr.<method_name>  
LPARAMETERS tParam1, tParam2, ...  
IResult = this.oDataTableMgr.<method_name>(tParam1, tParam2, ...)  
Return IResult
```

```
cDataMgr.<property_assign_method>  
LPARAMETERS tNewVal  
this.oDataTableMgr.<property_name> = tNewVal  
this.<property_name> = tNewVal
```

```
cDataMgr.<property_access_method>  
LOCAL IValue  
IValue = this.oDataTableMgr.<property_name>  
RETURN IValue
```

In *cDataMgr.Init()* wird ein Objekt der Klasse *cDataTableMgr* der Eigenschaft *oDataTableMgr* hinzugefügt.

Wenn eine Methode aufgerufen wird, wird der Aufruf mit den Parametern an die Methoden mit den entsprechenden Namen in *oDataTableMgr* weitergegeben. Die Werte, die von *oDataTableMgr* zurückgegeben werden, werden an die aufrufende Methode zurückgegeben.

Beispiel:

```
cDataMgr.Method1()  
LPARAMETERS Param1, Param2, ...  
lResult = This.oDataTableMgr.Method1(Param1, Param2, ...)  
RETURN lResult
```

Bei Eigenschaften werden *Access* und *Assign* Methoden verwendet, um die Werte der Eigenschaften an das Objekt *oDataTableMgr* weiterzugeben. Bei *Assign* wird der Wert *cDataMgr.PropertyName* an die entsprechende Eigenschaft *oDataTableMgr.PropertyName* weitergegeben. Bei *Access* wird der Wert der Eigenschaft von *oDataTableMgr.PropertyName* gelesen und zurückgegeben.

Beispiel:

```
cDataMgr.Property1_assign()  
LPARAMETERS vNewVal  
This.oDataTableMgr.Property1 = vNewVal  
This.Property1 = vNewVal
```

```
cDataMgr.Property1_access()  
RETURN This.oDataTableMgr.Property1
```

Eigenschaften:

aCargo – array
aLinkedtable
cAliasName
cIDfieldname
cLinkedTables
cValidchildpositionexpr – Enthält einen Ausdruck, der zur Validierung eines Child-Datensatzes verwendet wird.
Extrabuffer – Ein Puffer zur freien Verwendung.
nLinkedtable
Setfilter

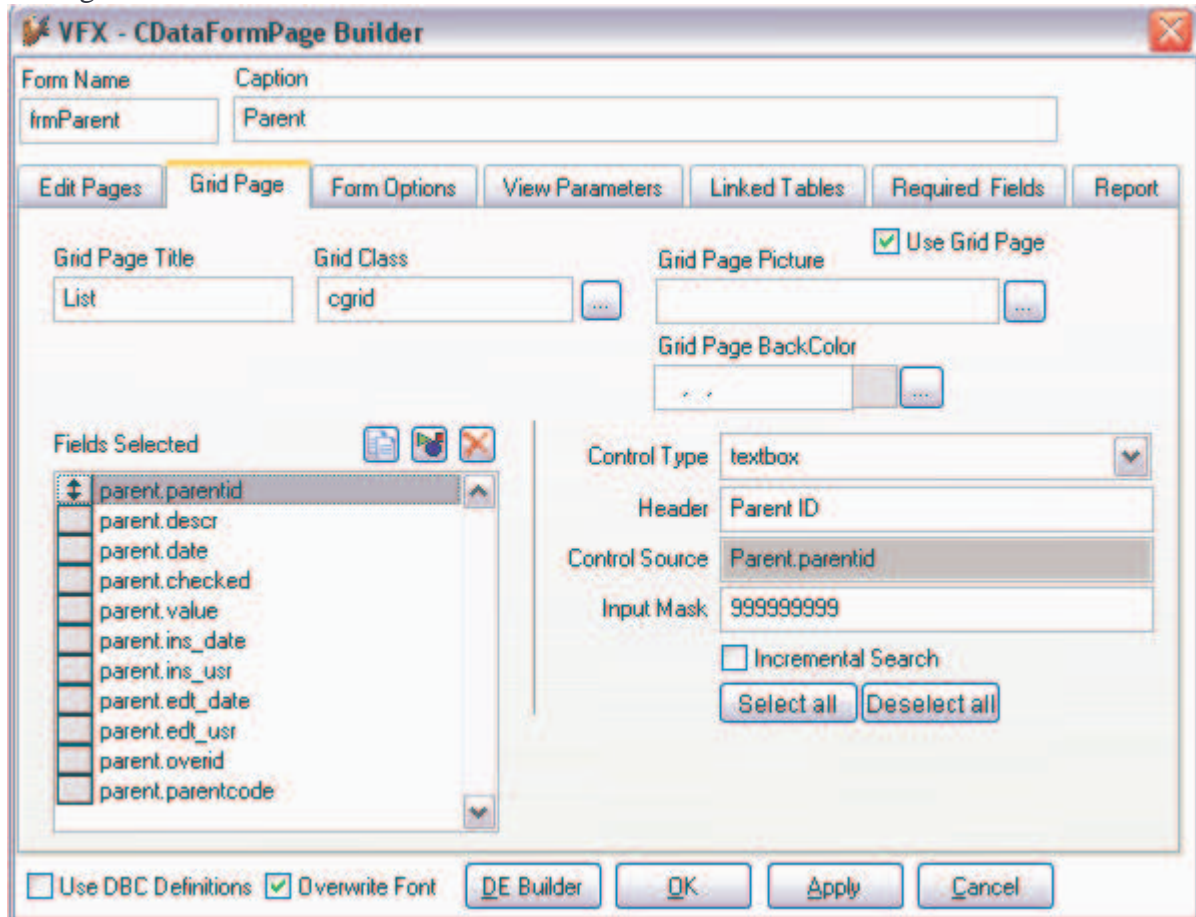
Methoden:

About()
CheckForValidChildPosition() – Prüfen ob der aktuelle Child-Datensatz in Beziehung zum aktuellen Datensatz in *cWorkAlias* steht.
Delete() – Löschen des aktuellen Datensatzes.
GoBottom() – Bewegen des Datensatzzeigers auf den letzten Datensatz.
GoTop() – Bewegen des Datensatzzeigers auf den ersten Datensatz.
Insert()
LogError() – Speichern von Fehlerinformationen in der Tabelle Vfxlog.
MakeValidChildPositionExpr() – Erstellen eines gültigen Ausdrucks. Dieser Ausdruck wird evaluiert um festzustellen, ob ein Child-Datensatz in Beziehung zu einem Parent-Datensatz steht.
Modified() – Liefert .T. zurück, wenn der Datensatz verändert wurde.
Next() - Bewegen des Datensatzzeigers auf den nächsten Datensatz.
Previous() – Bewegen des Datensatzzeigers auf den vorhergehenden Datensatz.
Release() – Freigeben dieses Objekts.
Requery() – Erneutes lesen des aktuellen Datensatzes.
Save() – Speichern des aktuellen Datensatzes.
SynchronizeLinkedTables()
Undo() – Änderungen rückgängig machen.
Update() – Aktualisieren mit *TABLEUPDATE()*.
UpdateForeignKey() – Aktualisieren von Fremdschlüsselfeldern.

Builder

Formular Builder

Auf der Grid Seite aller Formular Builder können Entwickler jetzt schnell die Einstellungen zur inkrementellen Suche für alle Spalten in einem Grid ändern. Dies kann mit den neuen Schaltflächen *Select all* und *Deselect all* auf der Grid Seite unterhalb der Checkbox *Incremental search* der Builder durchgeführt werden.



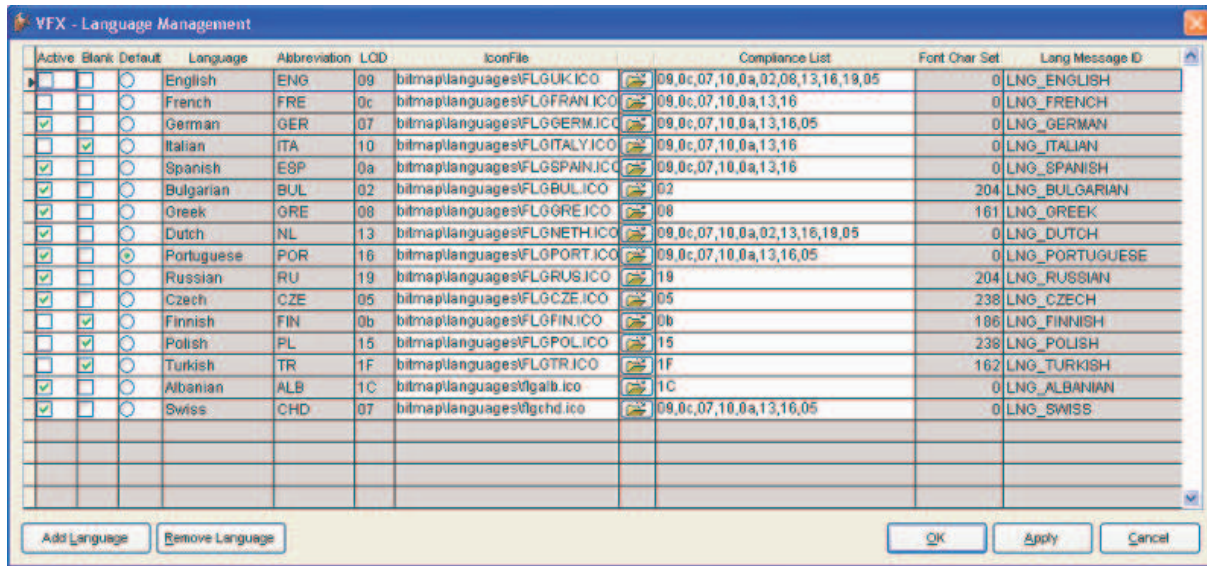
VFX - Language Management Builder

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

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

Wenn für eine Anwendung eine bestimmte Sprache nicht benötigt wird, können die Texte für diese Sprache gelöscht werden. Wenn diese Sprache zu einem späteren Zeitpunkt wieder hinzugefügt werden soll, kann die Markierung in der Checkbox *Blank* wieder entfernt werden. Dadurch werden die Texte aus der Vorlagetabelle *Vfxmsg.dbf* aus der VFX-Installation wieder in die Tabelle *Vfxmsg* des aktuellen Projekts kopiert.

In der Spalte *Default* kann der Entwickler die Standardsprache für eine Anwendung festlegen.



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

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

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

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

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

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

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

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

VFX – Help Wizard

Der VFX – Help Wizard trägt bei einem Klick auf die Schaltfläche *Set HelpContextIDs* bei allen Steuerelementen eines Projekts automatisch eine eindeutige *HelpContextID* ein. Falls in dem Projekt bereits nicht eindeutige *HelpContextIDs* vorhanden sind, werden die doppelt vorhandenen *HelpContextIDs* in der Datei *VFXHelpDoubledEntries.txt* im Projektordner gespeichert und nach Beenden des VFX – Help Wizard angezeigt. Der Entwickler kann so einfach doppelt vorhandene Einträge prüfen und ggf. korrigieren.

Die Tabelle Vfxfopen

Die Tabelle *Vfxfopen* enthält zwei neue Felder:

<i>VFXFOpen-Field</i>	<i>Beschreibung</i>	<i>Beispiel</i>
IconFile	Pfad- und Dateiname zu einem Icon, das im XP-Öffnen-Dialog angezeigt werden soll.	Bitmap\customers.ico
NotVisible	Wenn der Wert dieses Feldes .T. ist, wird der Eintrag nicht im XP-Öffnen-Dialog und auch nicht im alten Öffnen-Dialog angezeigt.	.T.

Neue Eigenschaften des Anwendungsobjekts

cRequiredFieldInitProps – Gibt die Initialisierungswerte für Pflichtfelder an. Beispielsweise kann hier die Hintergrundfarbe oder die Schriftfarbe für Pflichtfelder global eingestellt werden. Diese Werte werden verwendet, wenn die Eigenschaft *cRequiredFieldInitProps* eines Formulars leer ist.

cRequiredFieldFailureProps – Gibt die Eigenschaften für Pflichtfelder an, wenn der Benutzer keinen Wert eingegeben hat. Diese Werte werden verwendet, wenn die Eigenschaft *cRequiredFieldFailureProps* eines Formulars leer ist. Beispielsweise kann hier die Hintergrundfarbe oder die Schriftfarbe für Pflichtfelder global eingestellt werden, die gesetzt werden, wenn der Benutzer keinen Wert eingibt.

cSecurityTablesList – Diese Eigenschaft enthält eine semikolon-separierte Liste mit den Namen aller Sicherheitstabellen, die aktualisiert werden müssen, wenn ein Benutzername geändert wird oder wenn ein Benutzer gelöscht wird.

lCloseReportDialogOnExecution – Wenn der Wert dieser Eigenschaft auf .T. eingestellt ist, wird der Druckdialog nach der Ausgabe eines Berichts geschlossen.

nShowFilterName – Mit dieser Eigenschaft kann eingestellt werden, ob der Name eines gespeicherten Filters in der Titelzeile eines Formulars angezeigt werden soll. 0 – Es gilt die Einstellung des Formulars. 1 – Der Name des Filters wird in allen Formularen in der Titelzeile angezeigt. 2 – Der Name des Filters wird nicht angezeigt. Diese Eigenschaft wird nur berücksichtigt, wenn *ThisForm.nFilterBehavior* = 2 - VFX95 eingestellt ist.

lActivateThemes – Mit dieser Eigenschaft kann eingestellt werden, ob Windows Designs angezeigt werden sollen. Diese Eigenschaft wird nur berücksichtigt, wenn Anpassungen je Benutzer nicht erlaubt sind.

nSaveWithoutTransaction – Hier kann eingestellt werden was geschehen soll, wenn eine Transaktion nicht abgeschlossen werden kann. Diese Einstellung wird nur in Formularen basierend auf der Klasse *cOneToMany* berücksichtigt. 0 – Die Einstellung des Formulars ist gültig. 1 – Ohne Transaktion speichern. 2 – Nicht ohne Transaktion speichern.

nUseMemoForm – Verwendung eines Formulars zur Bearbeitung von Memofeldern. 0 – Die Einstellung der Editbox ist gültig. 1 – Das Memo-Bearbeitungsformular wird in allen Formularen verwendet. 2 – Das Memo-Bearbeitungsformular wird nicht verwendet.

nIPAddressesListAllowed – Definiert wie die Liste der IP Adressen in *Vfxipaddresses.dbf* interpretiert wird. 0 – Die Liste der IP Adressen wird nicht verwendet. 1 – Die Liste der IP Adressen enthält erlaubte Adressen. 2 – Die Liste der IP Adressen enthält nicht erlaubte Adressen.

cMainMenu – Name des Hauptmenüs einschließlich Namensweiterung.

cTableManagerClass – Name der Datenzugriffsklasse.

lAutoHideXPOpen – Einstellung für das automatische Ausblenden des XP-Öffnen-Dialogs. .T. – Automatisches Ausblenden aktiviert, .F. – Automatisches Ausblenden ausgeschaltet.

nXPDialogAutoHideInterval – Zeitspanne in Sekunden, nach der der XP-Öffnen-Dialog automatisch ausgeblendet wird.

Versenden eines Fehlerberichts an einen Web Service

Endanwender können Laufzeitfehler aus dem Fehlerdialog direkt an einen Web Service senden. Dafür müssen die folgenden Einstellungen gemacht werden:

Anwendungseinstellungen:

Die folgenden Eigenschaften des Anwendungsobjekts müssen eingestellt werden, bevor ein Fehlerbericht an einen Web Service gesendet werden kann:

cWSDL – URL zu der WSDL Datei.
(Zum Beispiel: <http://register.something.com/regservice.wsdl>)
cServiceName – Name des Web Service
(Standardwert: vfxregservice)
cRemoteMethod – Name der Methode des Web Service, die den Fehlerbericht empfängt.
(Standardwert: ReceiveErrorInfo)

Die Werte dieser Eigenschaften können im Abschnitt Error Handling im VFX – Application Builder eingestellt werden.

Der Fehlerbericht wird als XML Zeichenkette versendet und enthält die folgenden Felder:

FirstName – Vorname des Anwenders.
LastName – Nachname des Anwenders.
Company – Firmenname des Anwenders.
Email – E-Mailadresse des Anwenders.
RegKey – Registrierungsnummer der Anwendung.
Version – Version der Exe-Datei.
Producer – Name des Herstellers der Anwendung. Es wird der Wert der Eigenschaft *goProgram.cCompanyName* übermittelt. Dieser Wert darf nicht leer sein.
AppName – Name der Anwendung. Es wird der Wert der Eigenschaft *goProgram.cAppName* übermittelt. Dieser Wert darf nicht leer sein.
ErrorInfo – Fehlerbericht.

Der Versand eines Fehlerberichts an den Web Service wird mit der Methode *SendToWS()* durchgeführt.

Wenn ein Fehlerbericht nicht erfolgreich an den Web Service gesendet werden kann, wird der Bericht per E-Mail versendet.

Einstellungen beim Server:

Alle empfangenen Fehlerberichte werden in der Tabelle *ReceivedErrors* in der Datenbank der Kundenverwaltung *RegData* gespeichert.

Zu jeder Anwendung, zu der Fehlerberichte vom Web Service empfangen werden können, sollte ein Eintrag in der Datei *Config.vfx* angelegt werden. Die ersten drei Felder sollten wie folgt eingestellt werden:

ClientName – Name der Anwendung, die den Fehlerbericht sendet. Der Name sollte dem Wert von *goProgram.cAppName* entsprechen.

ConnectionType – Database oder SQL Server.

Data Source – Pfad zur Datenbank *RegData* auf dem Server.

Der Name der Anwendung wird verwendet um die Datenbank zu ermitteln, in der der Fehlerbericht gespeichert werden soll.

Die Tabelle *ReceivedErrors* hat die folgende Struktur:

ErrID [*I(Autoinc)*]

DateRecv[*DateTime*] – Zeitstempel des Empfangs.

ErrorInfo[*Memo*] – Fehlerbericht.

FirstName[*C(15)*] – Vorname des Anwenders.

LastName[*C(15)*] – Nachname des Anwenders.

Company[*C(30)*] – Firmenname des Anwenders.

Email[*C(30)*] – E-Mailadresse des Anwenders.

RegKey[*C(30)*] – Registrierungsnummer der Anwendung.

Version[*C(15)*] – Version der Exe-Datei.

Producer[*C(20)*] – Name des Herstellers.

Application[*C(20)*] – Name der Anwendung.

Der Rückgabewert des Web Service ist eine XML Zeichenkette mit zwei Feldern:

nStatus

cText

Wenn der Fehlerbericht erfolgreich gespeichert werden konnte, wird in *nStatus* der Wert 0 zurückgegeben. Wenn ein Fehler aufgetreten ist, enthält *nStatus* einen Wert > 0 und *cText* enthält eine Beschreibung des Fehlers.

VFX – Kundenverwaltung

Die VFX – Kundenverwaltung enthält ein Formular zur Verwaltung der Laufzeitfehler, die in der Anwendung aufgetreten sind. Das Formular *Receivederrors.scx* verwendet als Datenquelle die Tabelle, in der vom Web Service empfangene Fehlerberichte gespeichert werden. In diesem Formular kann der Status von Fehlerberichten verfolgt werden und es können auch manuell Fehlerberichte erfasst werden. Die Informationen, die die Anwendung an den Web Service gesendet hat, enthält einen Zeitstempel, die Version der Anwendung, die Registrierungsnummer, Name, Firma und E-Mailadresse des Benutzers sowie Details über den Fehler. Zu jedem Eintrag kann ein Status ausgewählt werden: Empfangen, Gelesen, in Bearbeitung, Erledigt.

The screenshot shows a web browser window titled "New - Receivederrors". The window contains a form with the following fields and controls:

- Buttons: "Edit" (selected) and "List".
- Fields: "Daterecv" (date and time pickers), "Version" (text box), "Errorinfo" (text area), "First name" (text box), "Last name" (text box), "Company" (text box), "E-mail" (text box), "Regkey" (text box), and "Errorstatus" (dropdown menu).
- Value: The "Errorstatus" dropdown is set to "PROCEEDING".

Neue Eigenschaften für Endanwender

Benutzerverwaltung

Liste der erlaubten/verbotenen IP Adressen für die Anmeldung

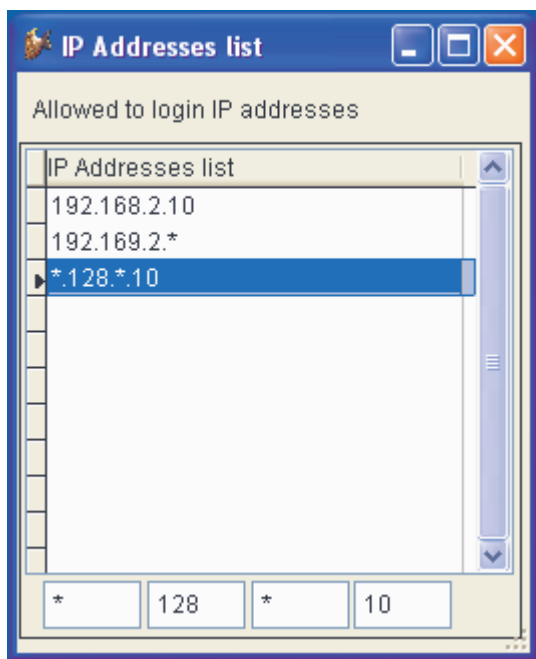
Benutzer mit Administratorrechten haben jetzt die Möglichkeit eine Liste mit IP Adressen anzugeben, von denen eine Anmeldung zur Anwendung erlaubt ist. Wahlweise kann eine Liste mit IP Adressen angegeben werden, von denen eine Anmeldung nicht erlaubt sein soll. Ob diese Eigenschaft zur Verfügung steht, kann mit der Eigenschaft *goProgram.nIPAddressesListAllowed* eingestellt werden. Die Liste der IP Adressen wird in der Tabelle *Vfxipaddresses* gespeichert. Die Bedeutung der IP Adressen wird mit der Eigenschaft *nIPAddressesListAllowed* gesteuert.

goProgram.nIPAddressesListAllowed

- 0 – Die Liste der IP Adressen wird nicht verwendet.
- 1 – In der Liste befinden sich die erlaubten IP Adressen.
- 2 – In der Liste befinden sich die nicht erlaubten IP Adressen.

Benutzer mit Administratorrechten können sich in jedem Fall von jeder IP Adresse aus anmelden.

Die Liste der IP Adressen kann nur von Benutzern mit Administratorrechten bearbeitet werden. Die Liste kann über den Menüpunkt Extras, IP-Adressen bearbeitet werden.



Wahlweise können vollständige IP Adressen, beispielsweise 192.168.2.10 oder Eingabemasken mit einem Stern (*) als Platzhalter für beliebige Werte verwendet werden, zum Beispiel *.128.*.10 oder 192.168.*.*.

Benutzern mit Administratorrechten wird in jedem Fall der Zugang gewährt. Bei der Anmeldung anderer Benutzer wird die IP Adresse überprüft.

Wenn die Liste erlaubte IP Adressen enthält und wenn die IP Adresse des aktuellen Computers in der Liste enthalten ist, wird dem Benutzer der Zugriff gewährt.

Wenn die Liste nicht erlaubte IP Adressen enthält und wenn die IP Adresse des aktuellen Computers in der Liste enthalten ist, wird dem Benutzer der Zugriff verweigert und es erscheint ein Hinweis in einer Messagebox.

Wenn die Liste der IP Adressen leer ist, dürfen sich Benutzer von allen IP Adressen anmelden.

Verwaltung von Sicherheitsrechten

Mit der Verwaltung von Sicherheitsrechten kann die Ansicht, und somit auch die Bearbeitung, auf bestimmte Datensätze je Benutzer eingeschränkt werden. Für jeden Datensatz aus einer Tabelle kann eine Liste mit Benutzern definiert werden, die diesen Datensatz ansehen und ggf. Bearbeiten dürfen.

Neue Eigenschaften der Klasse *cBaseDataAccess*:

cSecurityTable – Name der Sicherheitstabelle.

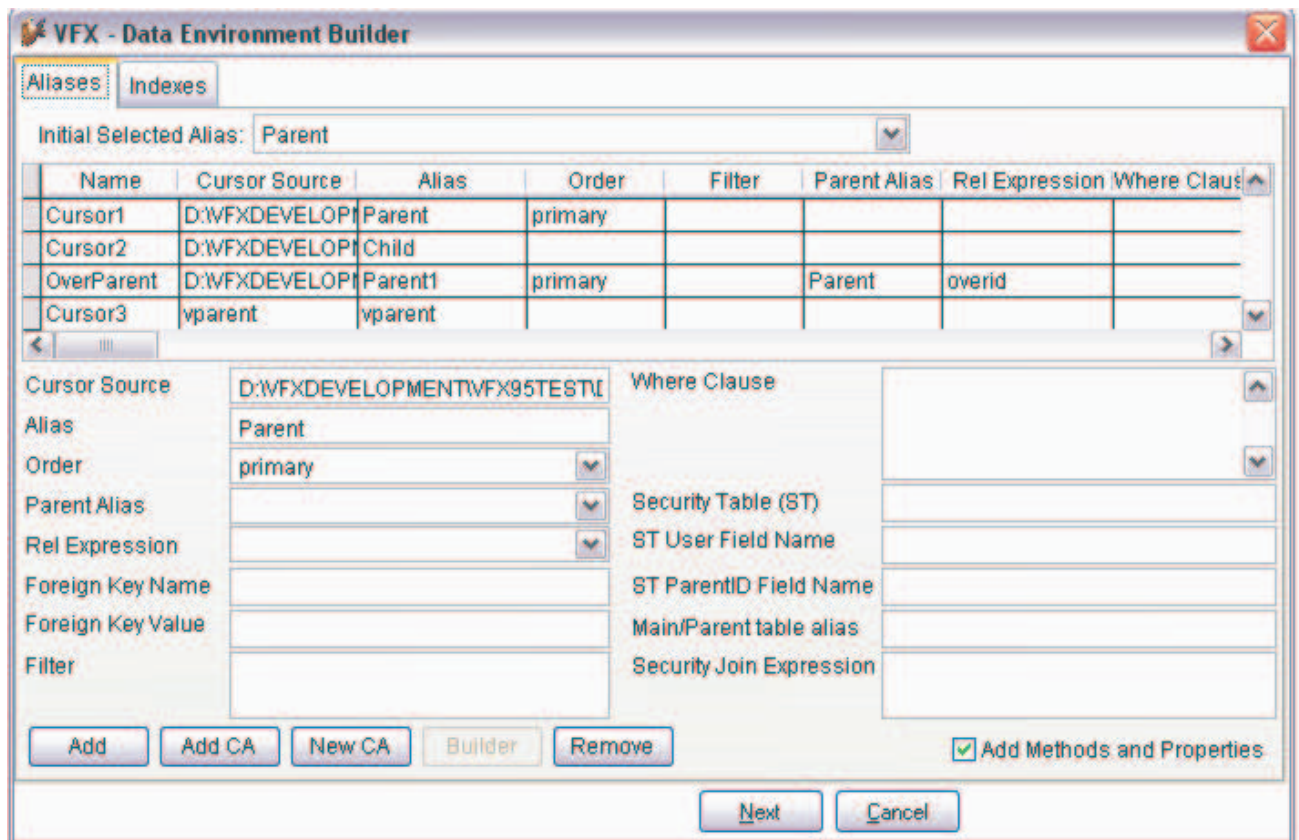
cSecurityUserFieldName – Name des Feldes mit dem Benutzernamen in der Sicherheitstabelle.

cSecurityParentIDFieldName - Name des Schlüsselfeldes in der Bearbeitungstabelle.

cParentAlias – Aliasname der Haupttabelle wenn mit einem Join-Ausdruck gearbeitet wird, siehe auch Beispiel unten.

cSecurityJoinExpression – Zusätzlicher Join-Ausdruck zur Sicherheitstabelle. Wird benötigt, wenn die aktuelle Bearbeitungstabelle eine Child-Tabelle ist, siehe auch Beispiel unten.

Die Werte dieser Eigenschaften können für Cursoradapter im VFX - Data Environment Builder eingestellt werden.



Für jede Tabelle einer Anwendung, für die Sicherheitsrechte eingestellt werden sollen, wird eine Tabelle zur Verwaltung der Sicherheitsrechte mit zwei Feldern benötigt. Das eine Feld enthält den Primärschlüssel des zu schützenden Datensatz. Das zweite Feld enthält den Namen des Benutzers,

der diesen Datensatz ansehen und ggf. bearbeiten darf. Das Schlüsselfeld muss den gleichen Namen haben, wie das Schlüsselfeld in der Bearbeitungstabelle.

Wenn die Daten eines Cursoradapters angezeigt werden sollen, fügt die Cursoradapter-Klasse den Select-Befehl automatisch eine Join-Klausel hinzu, so dass der Benutzer nur die Datensätze angezeigt bekommt, für die er eine Berechtigung hat.

Wenn der Bearbeitungstabelle ein neuer Datensatz hinzugefügt wird, wird auch der Tabelle mit den Sicherheitsrechten ein Datensatz hinzugefügt. Dadurch erhält der Benutzer die Möglichkeit den selbst angelegten Datensatz auch künftig angezeigt zu bekommen. Wenn in der Bearbeitungstabelle ein Datensatz gelöscht wird, werden auch alle dazugehörenden Datensätze in der Tabelle mit den Sicherheitsrechten gelöscht.

Für Child-Datensätze brauchen keine Sicherheitsdatensätze verwaltet werden, solange die Child-Tabelle keine eigene Tabelle für Sicherheitsrechte besitzt.

Um zu erkennen ob der aktuelle Datensatz aus einer Parent- oder Child-Tabelle stammt, wird wiederum der Wert der Eigenschaft *cSecurityJoinExpression* verwendet. Wenn dieser Wert leer ist, handelt es sich um eine Parent-Tabelle. Wenn in dieser Eigenschaft ein Join-Ausdruck gespeichert ist, ist die aktuelle Tabelle eine Child-Tabelle.

Beispiel:

Für eine Tabelle mit Auftragsdetails sollen Sicherheitsrechte verwaltet werden. Die Tabelle Auftragsdetails ist die Child-Tabelle von Aufträge. Die Relation zwischen den Tabellen Aufträge und Auftragsdetails wird mit dem Feld *Orderid* gebildet.

Eigenschaften des Cursoradapters:

cSecurityTable = "OrderSecurity"

cSecurityUserFieldName = "UserName"

cSecurityParentIDFieldName = "OrderId"

cSecurityJoinExpression = "JOIN Orders ord on OrderDetails.OrderID= ord.OrderID"

cParentAlias = "ord"

Der zur Laufzeit daraus erstellte Select-Befehl wird so aussehen:

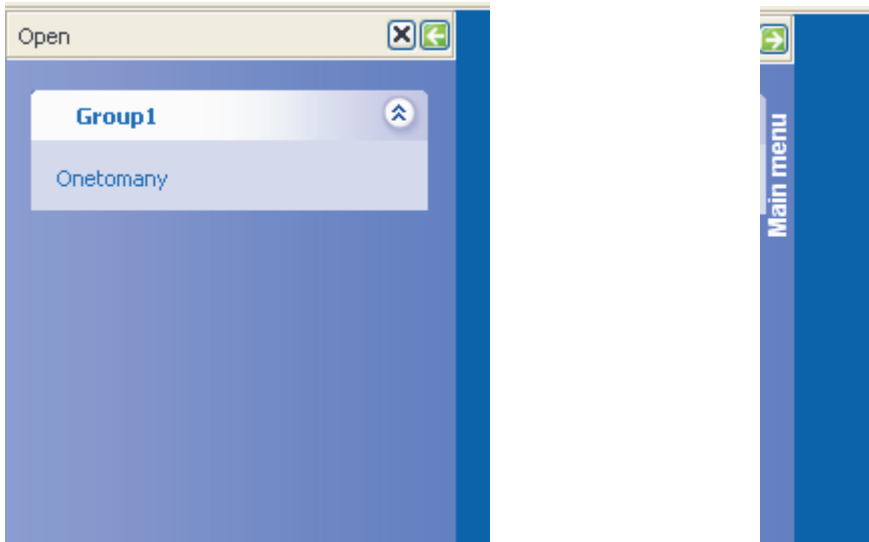
```
Select * from OrderDetails JOIN Orders ord on OrderDetails.OrderID=
ord.OrderID
JOIN OrderSecurity SecurityTable on SecurityTable.OrderId = ord.OrderId AND
UserName = ?goUser.User
```

Durch die Farbgebung ist leicht erkennbar welche Werte an welcher Stelle in den Select-Befehl eingefügt werden.

Erweiterungen der Benutzeroberfläche

XP Öffnen Dialog automatisch ausblenden

Der XP-Öffnen-Dialog kann automatisch ausgeblendet werden. Wenn diese Eigenschaft verwendet wird, erscheint in der Titelzeile des XP-Öffnen-Dialog ein kleiner Pfeil der anzeigt, dass der Dialog ausgeblendet werden kann.



Der XP-Öffnen-Dialog kann durch einen Klick auf den kleinen Pfeil manuell ausgeblendet werden. Durch eine Mausbewegung über den sichtbar verbleibenden Teil des XP-Öffnen-Dialogs, kann dieser wieder eingeblendet werden.

Das Ausblenden des XP-Öffnen-Dialogs wird durch einen Timer gesteuert. Mit der Eigenschaft *goProgram.nXPDialogAutohideInterval* kann eingestellt werden, nach welcher Zeitspanne der Dialog ausgeblendet werden soll. Der Wert dieser Eigenschaft kann im VFX – Application Builder eingestellt werden.

Wenn Anpassungen je Benutzer erlaubt sind, kann jeder Benutzer im Anpassen-Dialog individuell für sich einstellen, ob das Ausblenden des XP-Öffnen-Dialogs aktiviert werden soll. Die Steuerung erfolgt über die Eigenschaft:

```
goUser.AutoHideXPOpen  
0 – EIN  
1 – AUS
```

Wenn Anpassungen je Benutzer nicht erlaubt sind, wird das Verhalten des XP-Öffnen-Dialogs über die Eigenschaft *goProgram.lAutoHideXPOpen* kontrolliert. Der Wert dieser Eigenschaft kann zur Laufzeit durch den Benutzer nicht verändert werden.

```
goProgram.lAutoHideXPOpen  
.T. – EIN  
.F. – AUS
```

Unterstützung geringer Farbtiefen

Wenn eine Farbtiefe von maximal 256 Farben eingestellt ist, werden im XP-Öffnen-Dialog Bilddateien verwendet, die mit dieser geringen Farbtiefe gut angezeigt werden können. Die Farbtiefe wird beim Start der Anwendung ermittelt und in der Eigenschaft *goProgram.nColorDepth* gespeichert. Der Wert ist eine ganze Zahl, zum Beispiel *nColorDepth = 32* für 32-bit Farbtiefe.

Wenn dieser Wert maximal 8 beträgt, werden im XP-Öffnen-Dialog die Bilddateien *blue16.bmp*, *tabclose16.bmp* und *tabopen16.bmp* verwendet. Bei einer höheren Farbtiefe werden die Bilddateien *blue.bmp*, *tabclose.bmp* und *tabopen.bmp* verwendet.

RTF Unterstützung in Formularen (VfxCtrl.vcx, cRTFControl)

Dieses neue Steuerelement kann mit den VFX – Form Buildern jedem Formular hinzugefügt werden, wenn ein Feld vom Typ *Memo* oder *Objekt* als Datenquelle verwendet wird.



Dieses neue Steuerelement ermöglicht die Bearbeitung von Text im RTF Format. Zur Textbearbeitung steht eine Symbolleiste zur Verfügung. Über die Symbolleiste können der Zeichensatz, die Schriftgröße und die Effekte fett, kursiv und unterstrichen sowie die Ausrichtung und die Schriftfarbe eingestellt werden.

Dieses Steuerelement kann nur mit Feldern vom Typ *Memo* oder *Objekt* als Datenquelle verwendet werden.

RTF Text kann in Berichten gedruckt werden oder als BMP oder RTF Datei gespeichert werden.

RTF Text kann auch als E-Mail mit BMP- oder RTF-Anhang versendet werden.

Es ist empfehlenswert in der Methode *OnPrint* des Formulars folgenden Code einzufügen, wenn ein Bericht mit RTF Inhalt gedruckt werden soll:

```
PARAMETERS tpreview

LOCAL lcMessageText
IF Type("tpreview") == "N" AND INLIST(tpreview, 1,2,6,7,11,12)  &&1 - email
with PDF,
&&11 - email with HTML,
&&12 - email with XML,
&&2 - save as PDF,
&&6 - save as HTML,
&&7 - save as XML
MESSAGEBOX("RTF is not supported in this format!", 0+48, ThisForm.Caption)
RETURN .F.
ELSE
DODEFAULT(tpreview)
ENDIF
```

RTF Unterstützung in Berichten

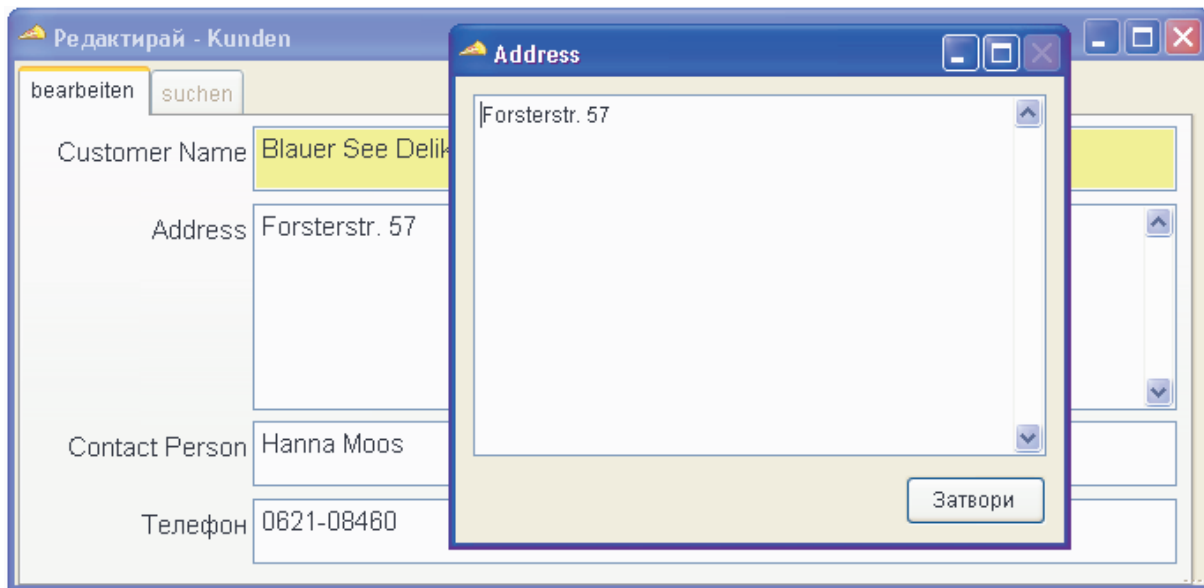
Um RTF Text zu drucken, muss dem Bericht ein Rechteck hinzugefügt werden, in dem der RTF Text zur Laufzeit gedruckt werden soll. Die Höhe des Rechtecks wird nicht beachtet. Die Höhe des Rechtecks wird zur Laufzeit so erweitert, dass der RTF Text vollständig gedruckt werden kann. In den Eigenschaften des Rechtecks muss auf der Seite *Other, Edit settings* geöffnet werden. Im Dialog *Run-time extensions* kann über die Schaltfläche *Edit XML* der Eintrag „*rptctrl*“ hinzugefügt werden. Die vollständige XML Zeichenkette sieht so aus:

```
<VFPData>
  <reportdata name="" type="R" script="" execute="" execwhen="" class=""
  classlib="" declass="" declasslib=""/>
  <rptctrl class="RTFHandler" expr="Parent.notes"/>
</VFPData>
```

MyTable.rtfmemo gibt dabei das Feld mit dem zu druckenden RTF Text an. Der Bericht kann jetzt gespeichert werden.

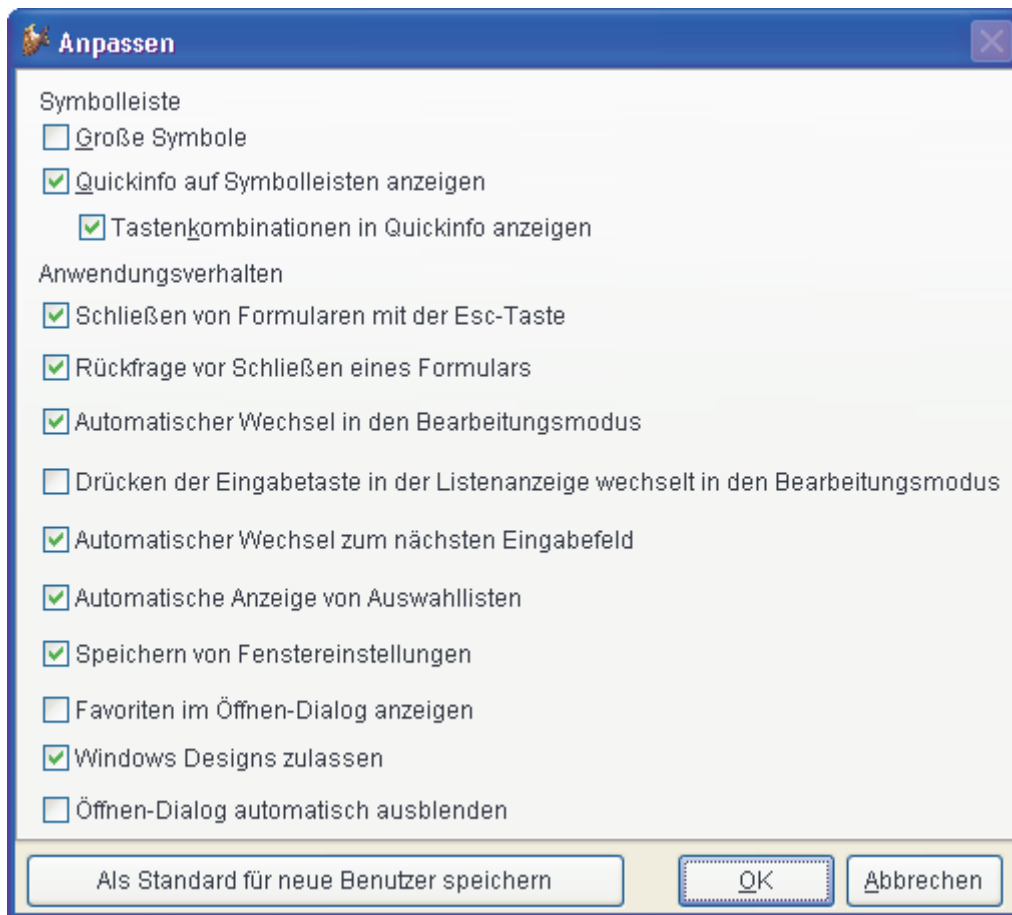
cMemoform für Steuerelemente aus der Klasse cEditbox

Umfangreiche Texte in Steuerelementen basierend auf der Klasse *cEditbox* können jetzt bequemer bearbeitet werden. Durch einen Doppelklick auf eine *cEditbox* öffnet sich ein Formular mit einem größeren Eingabebereich.



Anpassen-Dialog

Die neuen Einstellungen *Windows Designs verwenden* und *Öffnen-Dialog automatisch ausblenden* kann sich jeder Benutzer im Anpassen-Dialog individuell einstellen.



Suchdialog

Dem Suchdialog wurde der neue Operator *enthält nicht* hinzugefügt. Dieser Operator kann bei den Datentypen *Zeichen* und *Memo* verwendet werden.