



Session D-DBI

Integration der DBI Controls aus Sedna in VFP am Beispiel von VFX

Uwe Habermann

Einführung

Mit Sedna werden acht ActiveX Steuerelemente von DBI geliefert. Hiermit lassen sich gut aussehende und leistungsfähige Elemente der Benutzeroberfläche aufbauen. Microsoft Outlook empfinden viele Benutzer als angenehm zu bedienen. Wie schön wäre es, eine „Office Bar“ in eigene Anwendungen integrieren zu können? Dazu ein Menü und eine Symbolleiste wie in Office 2003 und perfekte Eingabe von Datums- und Zeitwerten. In dieser Session wird gezeigt wie es geht. So bekommt jede VFP Anwendung ein neues und modernes Look & Feel.

Übersicht DBI Controls aus Sedna

Schon seit einiger Zeit sind von DBI Technologies die „Studio Controls R3“ erhältlich. Bei den Studio Controls handelt es sich um ein umfangreiches Paket von ActiveX Steuerelementen, die in Projekte verschiedener Programmiersprachen eingebunden werden können.

Für Sedna stehen acht der ActiveX Steuerelemente aus den Studio Controls kostenlos zur Verfügung. Diese ActiveX Steuerelemente mit dem Namen „DBI Sedna Components“ heruntergeladen werden und dürfen mit VFP 9 SP 2 Anwendungen an Endkunden weitergegeben werden. Hier der Download-Link:

http://www.dbi-tech.com/Special_Offers_SednaComponents.asp

Nach der Installation müssen die DBI Sedna Components mit dem DBI License Manager lizenziert werden. Zur Lizenzierung muss eine E-Mailadresse angegeben werden. Der Aktivierungsschlüssel wird sofort per E-Mail zugestellt.

Ohne Aktivierung ist eine Verwendung der DBI Sedna Components in der Entwicklungsumgebung von VFP nicht möglich.



Mit diesen acht ActiveX Steuerelementen werden die Wünsche der VFP Programmierer wahr. Endlich können VFP Anwendungen mit einer modernen Benutzeroberfläche versehen werden.

ctContact

Mit diesem ActiveX Steuerelement wird eine Art mehrspaltiges Grid mit einer dynamischen Zeilenhöhe bereitgestellt. Es besteht die Möglichkeit die Einträge zu bearbeiten. Eine Kontaktliste im Stil von Microsoft Outlook ist eine Anwendungsmöglichkeit.

ctDays

Mit diesem ActiveX Steuerelement steht uns ein leistungsfähiger Kalender in einer Tagesansicht und vielen Darstellungsmöglichkeiten zur Verfügung.

ctDEdit

Dieses ActiveX Steuerelement ermöglicht die Eingabe von Datums- und Zeitwerten.

ctDropMenu

Ein kontextsensitives Menü mit dem Aussehen von Microsoft Office 2003 Anwendungen wird durch dieses ActiveX Steuerelement bereitgestellt. Jedem Menüeintrag kann ein Icon zugeordnet werden.

ctFrame

Dieses ActiveX Steuerelement ist ein Container Steuerelement. Mit ctFrame können sehr schöne Hintergründe gestaltet werden.

ctListBar

Dieses ActiveX Steuerelement erinnert an Microsoft Outlook oder an die aus VFP bekannte Toolbox. In einem Container können Gruppen geöffnet und geschlossen werden. Jede Gruppe kann Einträge enthalten. Beim Klick auf einen Eintrag wird eine Aktion ausgelöst. Dieses Steuerelement bietet sich zum Öffnen von Formularen an. Es besteht die Möglichkeit mit mehreren Spalten zu arbeiten.

ctToolBar

Dieses ActiveX Steuerelement ist universell für die Anzeige von Menüs und Symbolleisten einsetzbar. Das Aussehen kann auf einfache Weise so eingestellt werden, dass Menü und Symbolleiste den Anwendungen aus Microsoft Office 2003 ähneln.

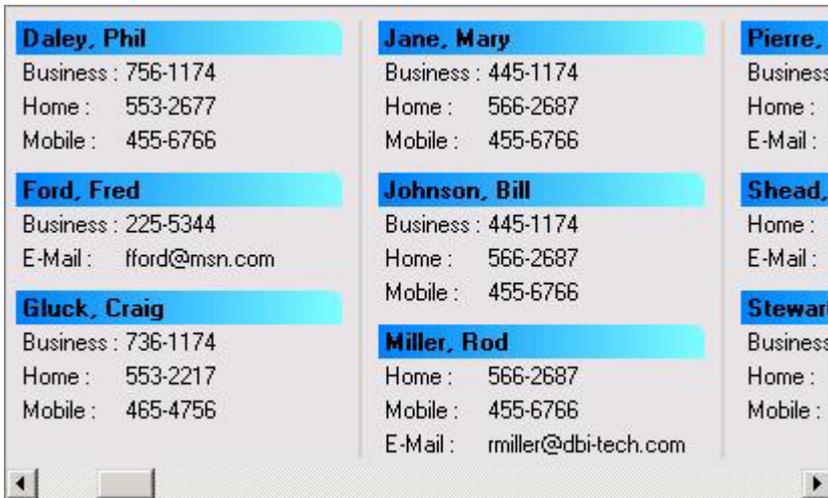
ctTray

Dieses ActiveX Steuerelement bietet dem Entwickler die Möglichkeit eine Anwendung mit einem Icon im Windows System Tray zu verstecken.

Die ActiveX Steuerelemente im Einzelnen

ctContact – Kontaktverwaltung

Das Steuerelement *ctContact* wurde speziell für die Verwaltung von Kontaktdaten entwickelt. In diesem Steuerelement kann eine Liste von Datensätzen in einem Kartenformat angezeigt werden. Jede Karte hat dabei eine variable Größe. Der Anwender kann die Daten in diesem Steuerelement bearbeiten.

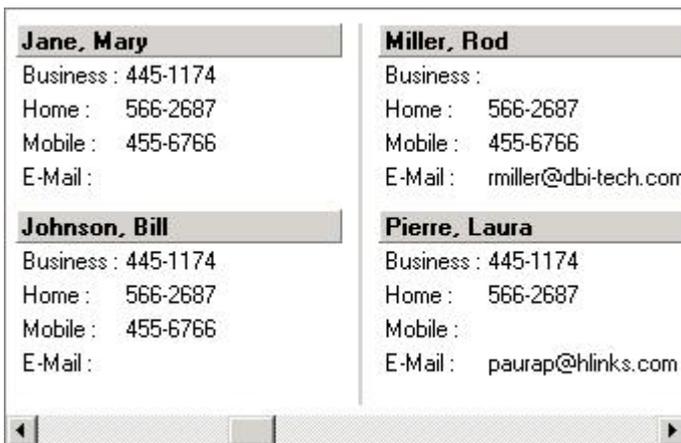


Es kann eingestellt werden, dass nur Felder mit Inhalt auf einer Karte angezeigt werden. Dadurch erhalten die Karten eine dynamische Länge.

Mit einer Suchmethode kann ein Datensatz gefunden werden, der einen bestimmten Text enthält. Die Methode positioniert die Ansicht so, dass der gefundene Datensatz in den sichtbaren Bereich kommt.

Mit einem Mausklick auf einen Eintrag kann der Anwender mit der Bearbeitung beginnen. Bei der Bearbeitung wird ein Ereignis ausgelöst.

Für die Darstellung der Kontaktansicht gibt es vielfältige Einstellmöglichkeiten. Es können verschiedene Farben und Füllstile gewählt werden. Diese Einstellungen können für Überschriften und den Kartenbereich unabhängig voneinander gemacht werden. Als Füllstil sind auch Farbverläufe in allen Richtungen möglich.



Um das Steuerelement programmatisch mit Daten zu füllen, müssen einige Schritte durchgeführt werden.

Zunächst müssen mit der Methode *AddTitle* die anzuzeigenden Felder definiert werden. Diese Methode bekommt als Parameter die anzuzeigende Bezeichnung für das Feld. Die folgenden Code Zeilen entsprechen den Einstellungen für den Screenshot von oben.

```
ctContact1.AddTitle "Business :"  
ctContact1.AddTitle "Home :"  
ctContact1.AddTitle "Mobile :"  
ctContact1.AddTitle "E-Mail :"
```

Mit der Methode *AddHeader* wird ein neuer Datensatz hinzugefügt und für diesen Datensatz wird die Überschrift eingetragen. Rückgabewert ist die ID für diesen neuen Datensatz

```
nIndex = ctContact1.AddHeader("Gluck, Craig")
```

Mit der Methode *ItemText* werden die Daten für einen Datensatz hinzugefügt. Als Parameter muss der Index für den Datensatz sowie die Nummer des Feldes übergeben werden.

```
ctContact1.ItemText(nIndex, 1) = "736-1174"  
ctContact1.ItemText(nIndex, 2) = "553-2217"  
ctContact1.ItemText(nIndex, 3) = "465-4756"
```

Genauso werden Daten für weitere Datensätze hinzugefügt. Es muss nicht jedes Feld mit Daten gefüllt werden. Dadurch wird eine dynamische Zellhöhe erreicht.

```
nIndex = ctContact1.AddHeader("Shead, Alan")  
ctContact1.ItemText(nIndex, 2) = "365-1174"
```

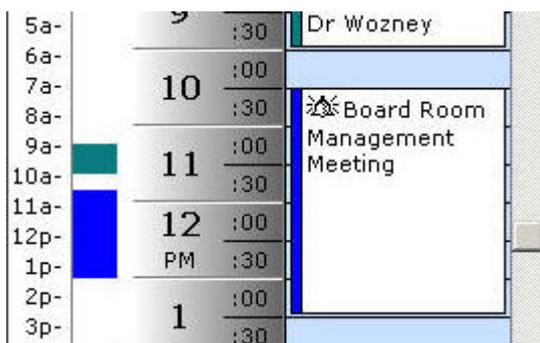
ctDays – Kalender Tagesansicht

Das ActiveX Steuerelement *ctDays* stellt einen grafischen Kalender dar. Die Anzeige ist ähnlich zur Anzeige in Microsoft Outlook. Es ist die Anzeige von Termin für einen Tag möglich. Dies entspricht der Tagesansicht von Outlook. Eine Wochen- oder Monatsansicht ist mit anderen ActiveX Steuerelementen von DBI möglich, die jedoch nicht mit Sedna geliefert werden. Es kann eine Zeitachse angezeigt werden, die grafisch die Termine eines Tages symbolisiert.

Neue Termine können durch einen Mausklick auf die gewünschte Zeit angelegt werden. Zu jedem Termin kann ein Beschreibungstext eingegeben werden.

Bei einem Mausklick auf einen Termineintrag, kann dieser direkt im ActiveX Steuerelement bearbeitet werden.

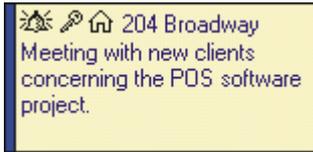
Termine können auch programmatisch angelegt, geändert oder gelöscht werden.



Standardmäßig wird dieses Steuerelement im Windows XP Stil angezeigt.

Die Zeiteinteilung kann in Schritten zu 5, 10, 15, 30 oder 60 Minuten erfolgen. Unabhängig von der Anzeige können Termine minutengenau erfasst werden. Die Start- und Endzeit in der Tagesansicht können frei gewählt werden.

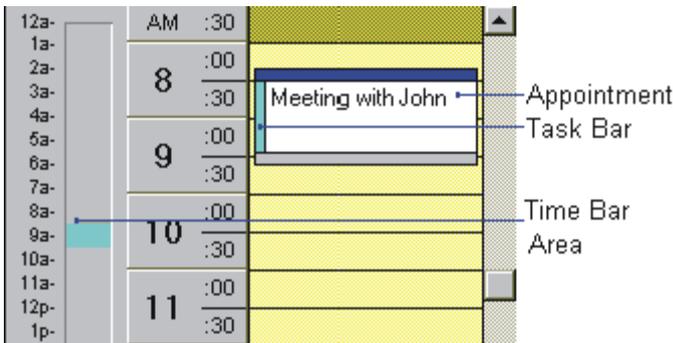
Es gibt Methoden und Eigenschaften die es ermöglichen sich überschneidende Termine zu erkennen. Überschneidungen können im Kalender auch angezeigt werden.



Diese Abbildung zeigt einen privaten Termin mit einer Erinnerung und einem Bild sowie einer Ortsangabe und einem Beschreibungstext.

Jedem Termin können bis zu vier Bilder zugeordnet werden, die in dem Steuerelement enthalten sind. Dies sind die Bilder Erinnerung, vorläufig, privat und Ortsangabe. Zusätzlich können bis zu drei weitere eigene Bilder hinzugefügt werden. Dem Steuerelement können insgesamt bis zu 100 Bilder hinzugefügt werden, von denen in einem Termin drei individuelle Bilder angezeigt werden können.

Mit einer einzigen Code Zeile kann die Ansicht des Steuerelements gedruckt werden. Mit der Methode *PrintInWindow* kann die Ansicht des Steuerelements in einem anzugebenden Seitenbereich gedruckt werden.



Die Start- und Endzeit eines Termins können durch Ziehen mit der Maus in der Tagesansicht verschoben werden.

Zu jedem Termin kann automatisch eine ID generiert werden. Dadurch können die Termine einfach in einer Tabelle gespeichert werden.

Es können mehrere Zeiten für verschiedene Zeitzonen angezeigt werden.

12 :00 PM :30	10 :00 :30
1 :00 :30	11 :00 :30
2 :00 :30	12 :00 PM :30
3 :00 :30	1 :00 :30

In dieser Abbildung werden zwei Zeitzonen angezeigt.

Hier Beispiele für weitere mögliche Ansichten des Kalenders:

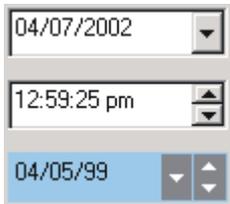
12a-	7:30 AM	
1a-	8:00 AM	
2a-	8:30 AM	
3a-	9:00 AM	
4a-	9:30 AM	
5a-	10:00 AM	Staff Meeting
6a-	10:30 AM	
7a-	11:00 AM	
8a-	11:30 AM	
9a-	12:00 PM	Lunch with George
10a-	12:30 PM	
11a-	1:00 PM	
12p-	1:30 PM	Haircut Banking
1p-	2:00 PM	
2p-	2:30 PM	

12a-	11	:00	
1a-	AM	:30	
2a-	12	:00	Lunch with Mary
3a-	PM	:30	
4a-	1	:00	
5a-		:30	
6a-	2	:00	Staff Meeting
7a-		:30	Conference Call
8a-	3	:00	
9a-		:30	
10a-	4	:00	
11a-		:30	
12p-	5	:00	Plan Vacation
1p-		:30	
2p-	6	:00	
3p-		:30	
4p-			
5p-			
6p-			
7p-			
8p-			
9p-			
10p-			
11p-			
12a-			

ctDEdit – Datum- und Zeitbearbeitung

Dieses Steuerelement ermöglicht die Eingabe und Anzeige von Datum- und Zeitwerten. Mit einer Eigenschaft kann eingestellt werden, ob Datum- oder Zeitwerte eingegeben werden sollen. Viele weitere Eigenschaften ermöglichen die Steuerung einer formatierten Eingabe.

Bei der Verwendung für die Zeiteingabe kann zwischen 12 Stunden und 24 Stunden Format gewählt werden. Die Datumeingabe wird in vielen internationalen Formaten unterstützt. Es werden alle Formate unterstützt, die auch in VFP unterstützt werden.



Die Anzeige ist in verschiedenen Präsentationsstilen möglich. Mit einfachen Einstellungen, kann ein typischer Windows XP Stil eingestellt werden. Es kann auch eine Darstellung gewählt werden, bei der das Steuerelement flach erscheint und sein Aussehen wechselt, wenn es den Fokus erhält.

Am rechten Rand des Steuerelements kann ein Pfeil nach unten oder ein Spinner angezeigt werden. Beim Klick auf diese Elemente wird ein Ereignis ausgelöst, aber es wird standardmäßig keine Aktion ausgeführt. Es ist jedoch einfach in diese Ereignisse Code unterzubringen, der einen Kalender startet oder das Datum tageweise vor- oder zurückzählt. Die Farbe und das Aussehen der Schaltflächen kann eingestellt werden.

Das Steuerelement kann so eingestellt werden, dass beim anhaltenden Klick auf eine der Schaltflächen das entsprechende Ereignis in einem einstellbaren Intervall wiederholt gefeuert wird. Damit ist zum Beispiel ein fortlaufendes hochzählen des Datums bei gedrückter Maustaste möglich.

Alle Einstellungen am *ctDEdit* Steuerelement können in einer XML Datei gespeichert werden. Genauso können auch alle Einstellungen aus einer XML Datei gelesen werden. Dies ist eine Alternative zur programmatischen Einstellung aller Eigenschaften.

Das Steuerelement *ctDEdit* unterstützt eine intelligente Datumeingabe. Unvollständige Datumeingaben werden beim Verlassen des Steuerelements vervollständigt.



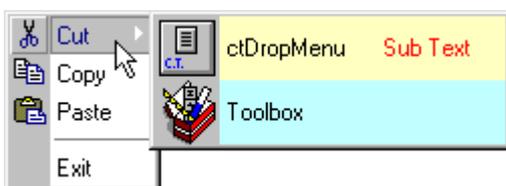
Bei der Verwendung für die Zeiteingabe kann ein gültiger Zeitraum vorgegeben werden.

Ähnlich wie in VFP wird ein Century Rollover unterstützt.

Wesentlich besser als in VFP, ist die Fehlerbehandlung gelöst. Wenn in VFP eine Textbox eine Controlsource vom Typ Datum hat und der Anwender ein ungültiges Datum eingibt, kann er die Textbox nicht verlassen, bis er seine Eingabe korrigiert und ein gültiges Datum eingegeben hat. In *ctDEdit* wird ein Ereignis gefeuert, wenn der Anwender versucht das Steuerelement mit einer ungültigen Eingabe zu verlassen. Standardmäßig wird eine ungültige Eingabe beim Verlassen des Steuerelements gelöscht.

ctDropMenu – Rechtsklickmenü

Dieses ActiveX Steuerelement erstellt Rechtsklickmenüs. Es sind viele Darstellungsmöglichkeiten einstellbar. Neben der Darstellung im Windows XP Stil können insbesondere Bilder, Texte, Hintergrund und Farben eingestellt werden.



Jedem Menüeintrag kann ein individuelles Bild zugeordnet werden. Jedem Menüeintrag kann ein zusätzlicher Hinweistext zugeordnet werden.

Dem Menü können Farben für alle Menüeinträge, für den ausgewählten Menüeintrag und für den linken Randbereich zugeordnet werden. Es können auch Füllmuster verwendet werden.

Rechtlickmenüs können kaskadierend in bis zu vier Ebenen erstellt werden.



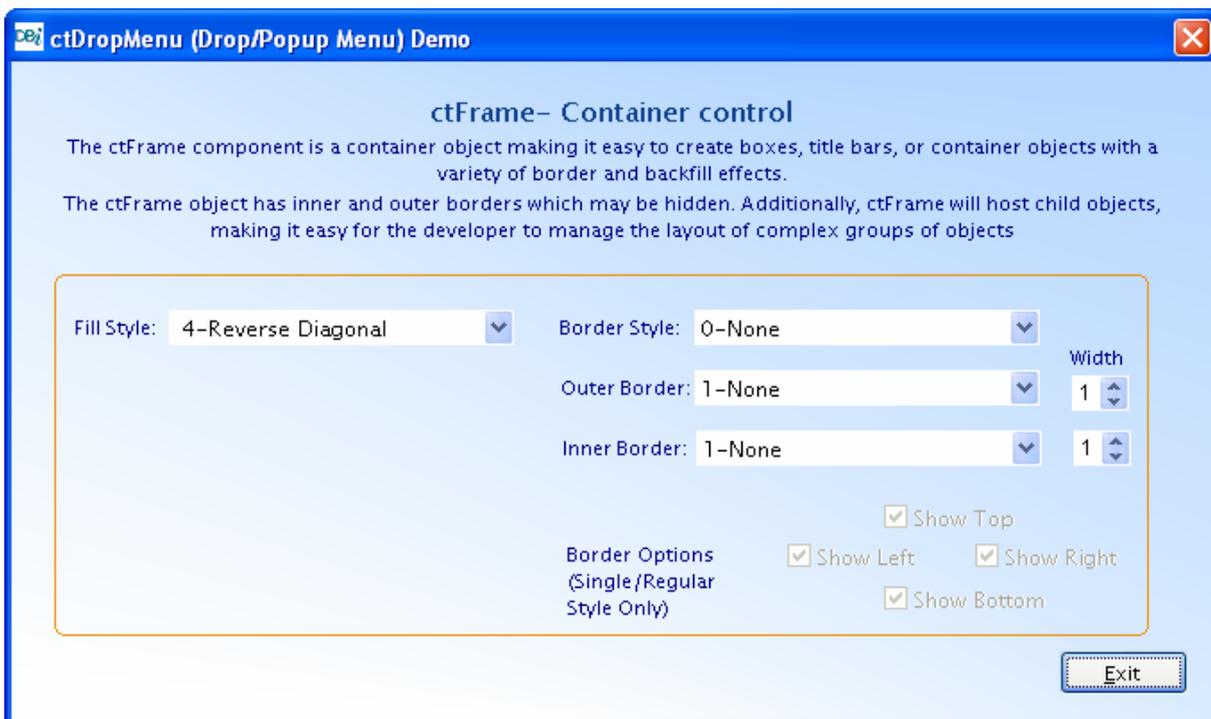
Integration in VFX

In der Klassenbibliothek Vfxformbase.vcx befindet sich die Formulareklasse *cDBIDropMenuVfxBase*. Auf dieser Formulareklasse befindet sich das ActiveX Steuerelement *ctDropMenu*. In der Klassenbibliothek Vfxform.vcx befindet sich die Ableitung *cDBIDropMenu*.

Wenn die Eigenschaft *goProgram.nMenuandToolbarBehaviour* den Wert 1 hat, wird automatisch das *ctDropMenu* verwendet.

ctFrame

Das ActiveX Steuerelement *ctFrame* verhält sich in VFP wie eine Container Klasse. Innerhalb des Steuerelements können VFP Steuerelemente platziert werden.

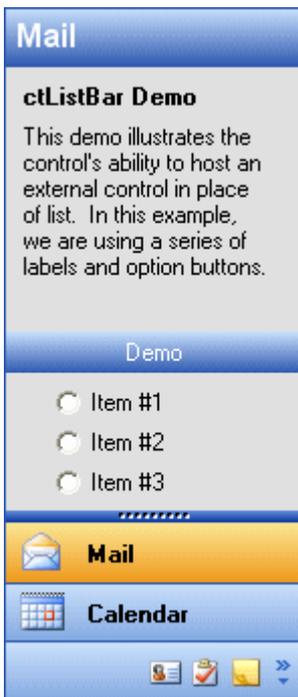


Der *ctFrame* Container hat einen einstellbaren Hintergrund mit unterschiedlichen Farbverläufen. Bei einem Resize des Formulars, wird der Farbverlauf für die neue Größe des Formulars neu berechnet. Der Hintergrund sieht also auch während oder nach einer Änderung der Größe gut aus.

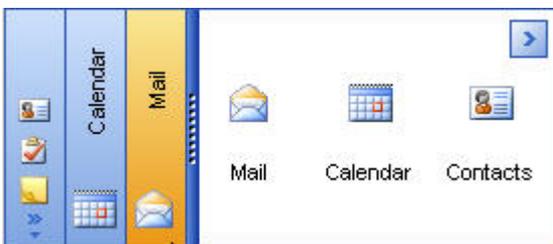
Neben der Darstellung eines Hintergrunds ist die Einstellung von verschiedenen Rändern, wahlweise an allen Seiten des Steuerelements möglich. Diese Möglichkeit stellt für VFP Entwickler aber nichts besonders dar.

ctListBar – Öffnen-Dialog

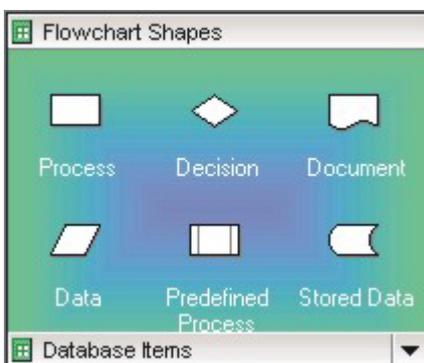
In einem *ctListBar* Steuerelement werden mehrspaltige Listen angezeigt, in denen der Anwender durch auf- und zuklicken von Gruppen navigieren kann. Jede Gruppe kann individuell andersartige Einträge enthalten.



Ein Listeneintrag innerhalb einer Gruppe kann ein Text oder ein Bild oder eine Kombination aus beidem sein. Wenn auf eine Gruppenüberschrift geklickt wird, wird die dazugehörige Gruppe geöffnet. Das Steuerelement führt dabei eine kleine Animation (smooth scrolling) aus, die dem Anwender den Vorgang verdeutlicht. Wenn mehr Gruppen vorhanden sind, als im anzeigbaren Bereich dargestellt werden können, werden automatisch Pfeile angezeigt, die ein Scrollen innerhalb der Gruppen ermöglichen.



Alle Einstellungen in einem *ctToolbar* Steuerelement können in einer XML Datei gespeichert werden. Alle Einstellungen können auch wieder aus einer XML Datei gelesen werden.



Die Elemente innerhalb einer *ctToolbar* können horizontal, vertikal oder mehrspaltig angezeigt werden. Es ist möglich mehrspaltige Listen aufzubauen.



Außer der Darstellung im Windows XP Stil ist es möglich den Hintergrund der Listen, der Gruppenüberschriften und der Navigationsschaltflächen mit verschiedenen Mustern oder Farbverläufen zu füllen. Als Hintergrund für die Liste kann auch ein Bild verwendet werden.

Das Element, über dem sich gerade die Maus befindet, kann durch eine andere Hintergrundfarbe, einen anderen Zeichensatz oder Schriftstil oder auch durch eine andere Schriftfarbe deutlich gemacht werden.

Ähnlich wie in der Toolbox von VFP wird auch in *ctToolbar* Drag & Drop unterstützt. Über Eigenschaften kann eingestellt werden, ob eine geschlossen Gruppe automatisch geöffnet werden soll, wenn mit der Maus ein Objekt über die Gruppe bewegt wird.



Listeneinträgen können Tooltips zugewiesen werden. Die Bezeichnung eines Listeneintrags kann umgebrochen werden. Es können bis zu 100 Bilder verwaltet werden.



Wenn die Maus über ein Element bewegt wird, können die Zeichensatzeinstellungen geändert werden und es kann auch ein anderes Bild angezeigt werden.

Die Schaltflächen zur Navigation können innerhalb der Liste an einer Seite oder an einer Seite einer Gruppenüberschrift platziert werden.

Integration des Öffnen-Dialogs in VFX

Die Klasse für den Öffnen-Dialog basierend auf der Klasse *ctListBar* heißt *cdbiopenformvfxbase*. Von allen Öffnen-Dialogen gemeinsam verwendeter Code befindet sich in der Klasse *copenformvfxbase*.

Die Klasse *cdbiopenformvfxbase* ist eine Formulareklasse, auf der das ActiveX Steuerelement *ctListBar* platziert ist.

Methoden

Makegroup() – Mit dieser Methode wird eine neue Gruppe angelegt. Dafür wird die Methode *addList* des *ctListBar* Objekts aufgerufen. Rückgabewert ist der Index des eingefügten Elements.

Makeitem() – Mit dieser Methode wird ein neuer Eintrag einer Gruppe hinzugefügt. Hierfür wird die Methode *addItem* des *ctListBar* Objekts aufgerufen. Rückgabewert ist der Index des eingefügten Elements.

Eigenschaften

ControlStyle – Diese Eigenschaft steuert das Aussehen des Steuerelements. Mögliche Werte sind:

- 0 – Standardwert.
- 1 – Listenansicht wie in Outlook 97.
- 2 – Symbolleiste wie in Visual Studio.
- 3 – Listenansicht wie in Outlook XP mit blauem Farbverlauf.
- 4 – Listenansicht wie in Outlook XP mit olivfarbenem Farbverlauf.

TipsDisplay – Mit dieser Eigenschaft können Tooltips eingeschaltet werden.

ctToolBar – Menüs und Symbolleisten

ctToolBar ist eine Komponente zur horizontalen Darstellung von Menüs und Symbolleisten. Diese Beschreibung lässt schon eine Einschränkung erkennen. Im Gegensatz zu Symbolleisten, die mit VFP erstellt werden, kann eine *ctToolBar* nicht an der rechten oder linken Seite des Bildschirms gedockt werden.

Symbolleiste und Menü werden in diesem Steuerelement gleich behandelt. In beiden Fällen sind Schaltflächen vorhanden, die auf einen Mausklick reagieren. Beim Menü enthalten diese Schaltflächen einen Text als Beschriftung. In der Symbolleiste werden üblicherweise Symbole statt einer Beschriftung verwendet.

Die Schaltflächen einer Symbolleiste reagieren in der Regel direkt auf einen Mausklick und führen eine Aktion aus.

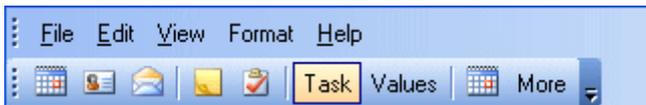
Beim Klick auf eine Schaltfläche in einem Menü wird in der Regel ein Menü-Pad geöffnet. Der Klick auf einen Eintrag in diesem Menü-Pad führt dann eine Aktion aus oder öffnet sogar ein weiteres Menü-Pad.

In Menü-Pads können verschiedene Elemente dargestellt werden, zum Beispiel markierbare Einträge oder Separatoren. In Menüeinträgen kann neben Text ein Bild angezeigt werden,



Alle Einstellungen einer Symbolleiste oder eines Menüs können in eine XML Datei exportiert werden. Alle Einstellungen können auch aus einer XML Datei eingelesen werden. So ist es einfach ein ganzes Menü oder eine Symbolleiste durch das Lesen einer Datei zu konfigurieren.

Durch einfache Einstellung von Eigenschaften kann das Aussehen so eingestellt werden, wie es typisch für Windows XP Anwendungen ist.



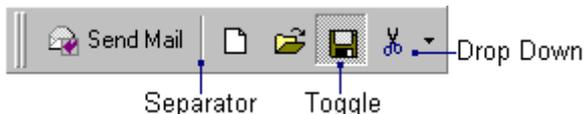
Es sind Ereignisse vorhanden, die ausgelöst werden, wenn der Anwender die Maus über ein Element bewegt oder den Status eines Elements ändert. Dadurch können dynamische Anzeigeeffekte erreicht werden. Zum Beispiel kann ein Menüeintrag, der während er von der Maus überfahren wird, mit fetter Schrift dargestellt werden.

Alle Elemente im *ctToolbar* Steuerelement sind in Collections beschrieben. Auf einzelne Elemente kann nur über einen Index referenziert werden. Der Index für ein Element wird beim Hinzufügen des Elements von der Methode *Additem* zurückgegeben. Dieser Index muss verwendet werden, um die Prozedur zu finden, die beim Klick auf einen Eintrag ausgeführt werden soll.

Zu beachten ist, dass wenn ein Element mit der Methode *RemoveItem* entfernt wird, alle Elemente der *ctToolbar* neu indiziert werden. Bei Entfernung eines Elements muss die selbst geführte Liste mit Indexen also auch selbst aktualisiert werden. Eine Alternative ist, nicht benutzte Elemente als unsichtbar einzustellen.

Ähnlich verhält es sich, wenn in ein bestehendes Menü neue Einträge eingefügt werden. Die *ctToolbar* aktualisiert alle Indexe und wir müssen die von uns geführte Liste mit Indexen ebenfalls aktualisieren, damit zu den Menüeinträgen die richtigen Aktionen ausgeführt werden. Die Indexe der *ctToolbar* können also nicht als konstante Werte verwendet werden.

Schaltflächen in Symbolleisten können eine Bezeichnung oder ein Bild oder beides haben. Eine gesunkene Darstellung ist möglich. Separatoren können verwendet werden.



Die Abbildung zeigt einige der Darstellungsmöglichkeiten in einer Symbolleiste.

Symbolleisten und Menüs können mit Farbverläufen angezeigt werden. Auch einzelne Einträge können mit einem Farbverlauf angezeigt werden. Farbverläufe können einzelnen Elementen in Menüs oder Symbolleisten oder auch zu einer ganzen Symbolleiste oder einem ganzen Menü zugewiesen werden.

Ein Menü kann bis zu vier Ebenen von Menüeinträgen enthalten.



In einem Menü oder einer Symbolleiste können bis zu 100 Bilder verwaltet werden. Zu jedem Bild kann eine Maskendatei verwendet werden, die Teile des Bildes transparent erscheinen lässt.

Das Steuerelement selbst enthält Bilder für häufig verwendete Anwendungszwecke, wie zum Beispiel für die Funktionen der Zwischenablage kopieren, ausschneiden und einfügen.

Es werden Bilder mit einer Größe von 16 x 16 Pixel angezeigt, wenn der Wert der Eigenschaft *SmallIcons* auf *.T.* eingestellt ist. Wenn der Wert dieser Eigenschaft *.F.* ist, werden Bilder in der Größe 32 x 32 Pixel angezeigt.

Eine Symbolleiste kann auf einem individuellen Hintergrundbild angezeigt werden.

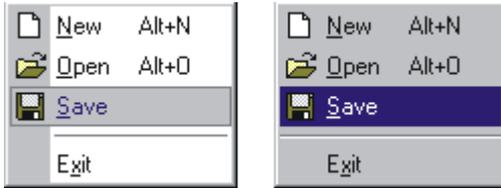


Zu jedem Element in einer Symbolleiste kann ein Tooltip angezeigt werden. Am linken Rand einer Symbolleiste kann zusätzlich ein Bild angezeigt werden.

Ähnlich wie in Symbolleisten und Menüs von Office 2003 Anwendungen, kann ein kleiner Pfeil am rechten Rand der Symbolleiste oder des Menüs anzeigen, dass nicht genug Platz zur Anzeige aller Elemente vorhanden ist. Wenn der Anwender auf diesen Pfeil klickt, werden die nicht dargestellten Elemente in einem Popup angezeigt. Menüeinträge, können auch so eingestellt werden, dass sie ausschließlich über dieses Popup erreicht werden und nie im normalen Menü angezeigt werden.



Diese Abbildung zeigt eine *ctToolBar* mit einem Bild am linken Rand und einem Pfeil zur Anzeige weiterer Elemente.

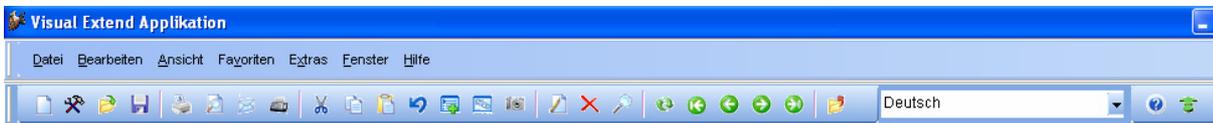


Beispiele für die Darstellungsmöglichkeiten von Menüs mit unterschiedlichem Hintergrund und Icons.

Zum Steuerelement *ctToolBar* gibt es einen Builder, der es ermöglicht viele Gestaltungseinstellungen zu machen.

Integration von Menü und Symbolleiste in VFX

In der Klassenbibliothek *Vfxappl.vcx* befinden sich die neuen Klassen *cDBiAppMenuBar* und *cDBiAppNavigatorBar*.



Symbolleiste

Den Schaltflächen in einer Symbolleiste können keine Namen zugeordnet werden. Es wird daher ein Array angelegt, in dem zu jeder Schaltfläche in der Symbolleiste neben dem Namen die Indexnummer gespeichert wird. Dieses Array wird vom Manager für Symbolleisten verwendet. Der Manager für Symbolleisten kapselt den Code, der gemeinsam in allen Symbolleisten verwendet wird, zum Beispiel für die Ereignisse Click und Refresh. Der Manager für Symbolleisten wird von den Symbolleisten *cOfficeBar*, *cNavigatorBar*, *cDBiAppMenuBar* und *cDBiAppNavigatorBar* verwendet.

Klasse *cToolBarManager* in *vfxAppl.vcx*

Diese Klasse wird von den bisherigen Symbolleistenklassen *cOfficeBar* und *cAppNavBar* sowie von der neuen Symbolleiste basierend auf *ctToolBar* verwendet.

Methoden

onRefresh(toItemsEnabled) – Hier ist der für alle Symbolleisten gemeinsame Refresh Code untergebracht. In dieser Methode wird zu allen Elementen der Symbolleiste der Status enabled bzw. disabled geschaltet.

onItemClick(tcItemName) – Hier ist der für alle Symbolleisten gemeinsame Click Code untergebracht. In einer Case Konstruktion, wird abhängig vom Wert des übergebenen Parameters der Code für die gewünschte Schaltfläche ausgeführt.

```
DO CASE
CASE tcItemName = "CMDNEW"
    * Code für das Click Ereignis der Schaltfläche "neu"
CASE tcItemName = "CMDSAVE"
    *
    ...
ENDCASE
```

Die leere Ableitung *cAppToolBarManager* von dieser Klasse befindet sich in der Klassenbibliothek *Appl.vcx*. Hier kann der Entwickler individuelle Anpassungen vornehmen, ohne den Code von VFX zu verändern.

Zur Laufzeit wird eine Referenz auf ein Objekt dieser Klasse in der Eigenschaft *oToolBarManager* des Anwendungsobjekts gespeichert. Bei der Initialisierung des Anwendungsobjekts wird dieses Objekt instanziiert.

Die Klassen *cOfficeBar*, *cNavigatorBar* und *cDbiNavigatorBar* verwenden die Klasse *cToolBarManager*.

Menü

Ein VFP Menü wird mit dem Menü-Designer erstellt. Das Design wird in einer Tabelle mit der Namensweiterung *mnx* gespeichert. Der VFP Menü-Designer kann daraus eine Programmdatei mit der Namensweiterung *mpr* erstellen. Das Kompilat dieser Datei hat die Namensweiterung *mpx* und wird schließlich in Exe Dateien eingebunden und zur Laufzeit der Anwendung ausgeführt. Das ist der bekannte Weg, mit dem ein VFP Menü angezeigt wird und funktioniert.

VFX hat einen eigenen Menü-Designer mit erweiterten Möglichkeiten. Die Struktur der Menütabelle wurde erweitert um alle diese Möglichkeiten speichern zu können. Die Tabelle hat die Namensweiterung *vmx*. Auch der VFX Menü-Designer generiert aus der Tabelle eine Programmdatei mit der Namensweiterung *vmr*. Das Kompilat mit der normalen Namensweiterung *fxp* wird in Exe Dateien eingebunden. Dieser Weg unterscheidet sich nicht wesentlich von dem von VFP verwendeten Verfahren.

Um eine *ctToolbar* als Menü aufzubauen ist die Tabelle mit den Menüinformationen in das Projekt eingeschlossen und wird zur Laufzeit ausgelesen um das *ctToolbar* Steuerelement mit Informationen zu füllen. Dieser Weg ist voll generisch implementiert.

Nun gibt es in einem Menü aber einige Code Teile, die ausgeführt werden, wenn der Benutzer einen Menüeintrag auswählt. Weiterer Code wird benötigt um Einträge zu enablen oder diablen. Dieser Code befindet sich in der Tabelle mit der Menübeschreibung. Es können aber auch Code Teile verwendet werden, die sich an anderer Stelle im Projekt befinden. Dieser Code lässt sich zur Laufzeit nicht ohne weiteres aus der Tabelle ausführen. Beim Erstellen einer Exe Datei wird daher im Project Hook eine Programmdatei mit allen erforderlichen Code Teilen erstellt und automatisch in das Projekt eingebunden.

In *ctToolbar* gibt es für Menüeinträge keine Skip For Klausel. Menüeinträge können nur über die Eigenschaft *ItemEnabled* aktiviert oder deaktiviert werden. Der Wert dieser Eigenschaft wird im *Refresh* Ereignis ausgeführt. Das *Refresh* wird im *Gotfocus* Ereignis ausgeführt.

Wenn ein Menüeintrag disabled erscheint, wird das Bild angezeigt, das in der Eigenschaft *ItemPicDisabled* eingetragen ist. Wenn dieser Eigenschaft kein Wert zugewiesen ist, wird das gleich Bild angezeigt, das auch im enabled Status verwendet wird.

Entsprechend der Darstellung in Office 2003 Anwendungen hat das Menü einen Farbverlauf im Hintergrund. Bilder sehen darauf nur dann gut aus, wenn sie einen transparenten Hintergrund haben. Leider ist der einzige Dateityp, der von *ctToolbar* mit transparentem Hintergrund unterstützt wird, der Typ *ico*.

Eine Alternative besteht darin, für Bilder anderer Dateitypen eine Farbe zu bestimmen, die transparent erscheinen soll. In der Regel wird dies die Farbe weiß sein. Dafür muss der Wert der Eigenschaft *MaskBitmap* mit *.T.* eingestellt werden und in der Eigenschaft *MaskColor* wird der Farbwert der transparent zu erscheinenden Farbe abgelegt. Diese Art der Darstellung funktioniert mit allen Bilddateiformaten.

Zwischenablage

Die *ctToolbar* enthält Standardelemente, die grundsätzlich auch in VFP Anwendungen funktionieren. Wie wir wissen, haben viele ActiveX Steuerelemente in VFP ein etwas ungewöhnliches Verhalten. Dies ist bei den Funktionen der Zwischenablage der Fall. Während eine Funktion für die Zwischenablage ausgewählt wird, hat das ActiveX Steuerelement den Fokus, aber nicht unsere darunter liegende VFP Anwendung. Ein Workaround ist, programmatisch mit der Maus auf das aktive Formular zu klicken und anschließend mit dem Keyboard Befehl die Funktion der Zwischenablage auszulösen. Hier der Code für „ausschneiden“.

```
lnLeft = _SCREEN.left + _SCREEN.ACTIVEFORM.Left
lnTop = _SCREEN.top + _SCREEN.ACTIVEFORM.Top
MOUSE CLICK AT lnTop, lnLeft PIXELS
KEYBOARD "{CTRL+X}" CLEAR
```

Integration in eigene Projekte

Eigene VFX Projekte können mit wenigen Einstellungen so konfiguriert werden, dass das ActiveX Steuerelement *ctToolbar* zur Anzeige von Menü und Symbolleiste verwendet wird.

Das Projekt muss auf den aktuellen Build von VFX 10.0 aktualisiert werden. Dann sind die folgenden Eigenschaften einzustellen:

```
cFoxAppl.nMenuAndToolbarBehaviour = 1  
cFoxAppl.cMainMenu = "cDBiAppMenuBar"  
cFoxAppl.cMainToolbar = "cDBiAppNavigatorBar"
```

Anschließend muss eine Exe Datei aus dem Projekt generiert werden, weil der Project Hook Code generiert, der auch für Tests in der Entwicklungsumgebung benötigt wird.

ctTray – VFP Anwendung im Windows System Tray

Mit dem ActiveX Steuerelement *ctTray* ist es möglich eine VFP Anwendung im Windows System Tray zu verstecken. Dieses Verhalten ist insbesondere bei Anwendungen nützlich, die ständig laufen sollen oder müssen, bei denen aber nur selten oder gar nicht die Bedienung über die Benutzeroberfläche benötigt wird.

Während sich eine Anwendung im Windows System Tray befindet, ist diese Anwendung nicht in der Windows Startzeile und auch nicht im Task Manager auf der Seite Anwendungen sichtbar.

Das Steuerelement selbst ist nicht sichtbar. Es platziert zur Laufzeit ein Icon im Windows System Tray. Wenn der Anwender zur Laufzeit auf dieses Icon klickt, werden verschiedene Ereignisse ausgelöst.



Das Steuerelement *ctTray* erfordert nur wenig Programmieraufwand um eine Anwendung im Windows System Tray mit einem Icon zu platzieren und auf Ereignisse zu reagieren.

Das Steuerelement kann bis zu vier Icons verwalten. Zur Laufzeit kann der Entwickler eins dieser Icons zur Anzeige im System Tray auswählen. Icons können während der Entwicklungszeit oder zur Laufzeit geladen werden.

Der Entwickler kann auf die Ereignisse Klick und Doppelklick mit der Ausführung beliebigen Codes reagieren.