VFX 9.5 – What is new?
2. Quarter 2006

July 2006

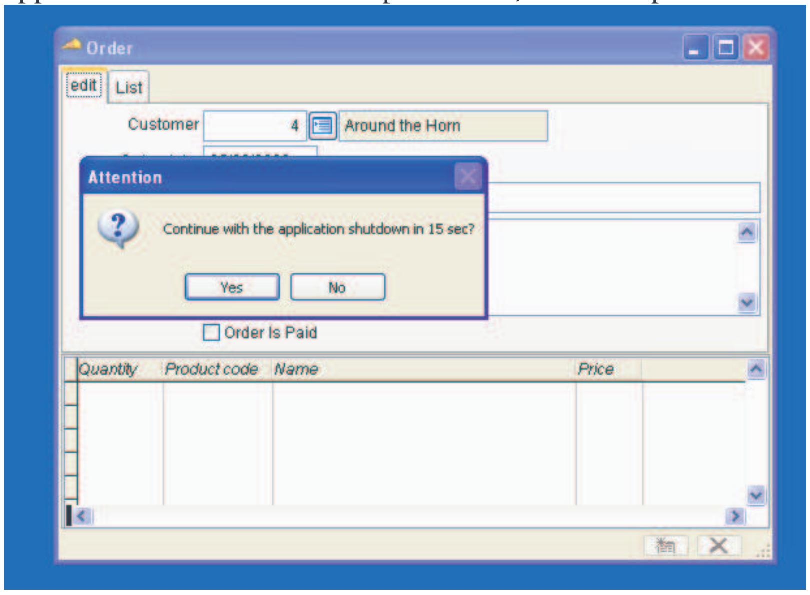
Uwe Habermann, Venelina Jordanova

Table of content

New features for developers

Automatic application termination

A new feature in VFX allows the built application to be terminated automatically after a given period without user activity. The feature is controlled by the property *lUseApplicationTimeout* of *goProgram*. To switch the feature on, set the value of this property to*.T.* The default values of *goProgram.lUseApplicationTimeout* is *.F.* The timeout is defined in minutes and its value is stored in *nApplicationTimeout* property of *goProgram*. The default value - 0 is equivalent to setting *lUseApplicationTimeout* = .F. (No application timeout).
When a timeout event occurs a warning messagebox is displayed. User is informed that an application shutdown will be performed, with an option to cancel it.



Continue with the application shutdown in 15 sec?

This messagebox is shown with timeout in seconds and the value of message timeout is stored in *goProgram.nAppTerminateMessageTimeout*. The default value of this property is 15 sec.

This feature is based on a global timer object. Two methods of *goProgram* are provided for controlling this timer – *AppTimerOnOff()* and *AppTimerReset()*.
The methode *goProgram.AppTimerOnOff()* expects a parameter, controlling if timer will be switched on or off. When the parameter value is .T., the timer is switched on. When a value .F. is passed, the timer is switched off. The develper can use this method to control the timer. When a longlasting operation is run, it is necessary to stop the timer and not to terminate the application, even if there is no end-user activity. The timer is turned off by VFX when creating/extracting archives, when database repair/pack/reindex operations are started and when database update is started. After finishing the opearion, the timer is switched on again.
The methode *goProgram.AppTimerReset()* resets the timer. This function is called with every user action, to restart timeout period.

Run backdoor programs

Under some circumstances it could be necessary to execute an additional program with the start of the application. This can be achieved using the VFX feature allowing backdoor programs to be run. This feature is turned on by setting the property *lRunBackdoorProgram* of *goProgram* to .T. (default is .F.). The available formats of program files for execution are PRG, FXP and APP. The name of the program file is obtaned from *goProgram.cBackdoorProgramName* and the file itself should be in the application folder, along with EXE file. After a successful execution the file is renamed by appending the date to its name.

Exanple: MyBackdoor.app is renamed to MyBackdoor_20060531.app

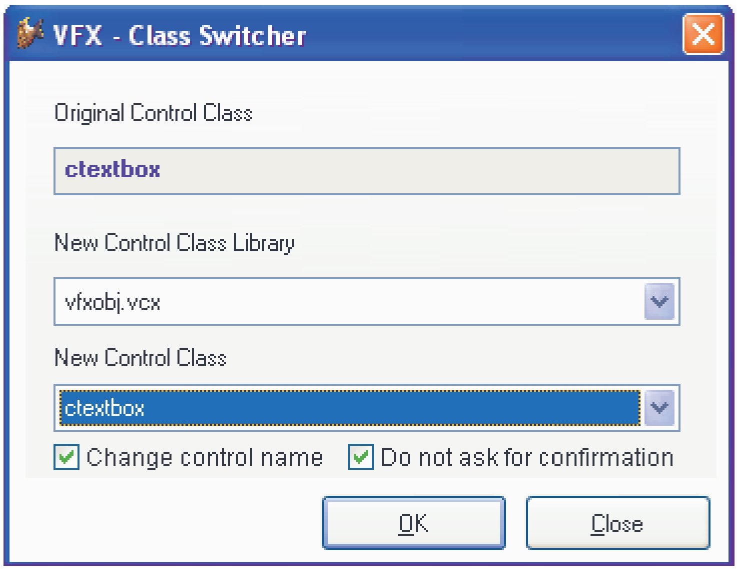
All other files with same filename, but different extensions (e.g. MyBackdoor.prg, MyBackdoor.fxp) are deleted. When checking for programs to execute the search strategy is as follows: APP, PRG, FXP. The properties *lRunBackdoorProgram* and *cBackdoorProgramName* are available for editing in VFX - Application Builder.

VFX – Class Switcher

Two new options are now available in VFX – Class switcher, when changing underlying class of a control.
If *ChangeControlName* checkbox is selected, the name of the new control is changed accordingly to its type, for instance if a textbox is switched to editbox the name is changed accordingly from *txtInput* to *edtInput.* If the control's name takes part in existing code it is better to keep the old name of the control, by not selecting this checkbox.

The *Do not ask for confirmation* checkbox controls if a confirmation message is displayed every time a class switching is made.

These two settings are stored in the VFX.INI file and used on next run of VFX – Class switcher.


VFX - Class Switcher

Original Control Class: ctextbox

New Control Class Library: vfxobj.vcx

New Control Class: ctextbox

✓ Change control name ✓ Do not ask for confirmation

VFX – Parent/Child Builder

In earlier VFX versions it was possible in the *OnMore* method to call Child forms or methods of the Parent form or to execute also a *Wait Window* command. Additionally to the programmatic way to do that, now in VFX 9.5, all these three possibilities can also be selected in the VFX –Parent/Child Builder.

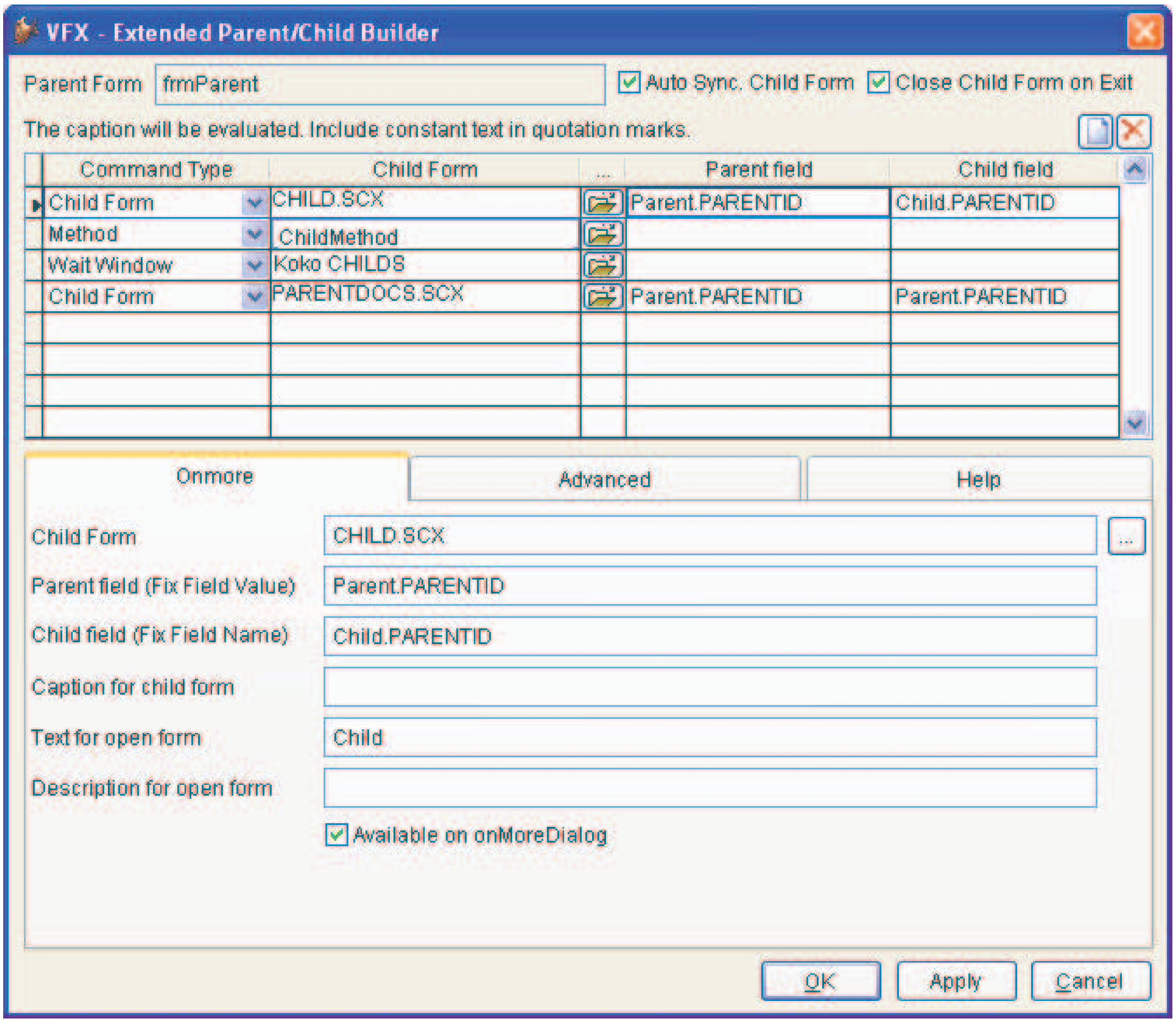
In *Command Type* column in the Grid, developer can choose if a Child form will be invoked, a method of the form will be exected or a *Wait Window* will be displayed.

Additionally to the chance to edit data in the Grid, all values can be entered alternatively also in textboxes below the Grid, in lower part of the form. Also there have been added more useful settings which allow:

- Displaying the child function in *OnMore* dialog;
- Synchronization per Child form;

- Automatically closing the Child form, when the Parent form is closed. This feature can also be set per Child form;
- Activate a specific page when open the Child form
- Position on a specific child record when open the Child form. This positioning is based on an expression, keyed up in builders;
- Selecting filter to apply on the Child form;
- Closing or hiding Parent form when Child form is open;
- Assigning an unique identifier to a child function
- Assigning a code identifier to a child function. This code is specified by the developer;
- Entering some Help information;

Now to the method onMore can alternatively be passed three different parameters. You can pass the consecutive number of the child function, corresponding to the sequence of the definition in *VFX – Parent/Child Builder*. You can pass the generated unique identifier, assigned to the child function by the builder. Or you can pass the code identifier, entered by you to identify the child fumnction in a more convenient way.
All these settings are grouped in three different pages in *VFX – Parent/Child Builder*. The two pages Advanced and Help are available only for Child forms and are disabled when the child function is *Method* or *Wait window.*



Parent Form: frmParent
[✓] Auto Sync. Child Form [✓] Close Child Form on Exit

The caption will be evaluated. Include constant text in quotation marks.

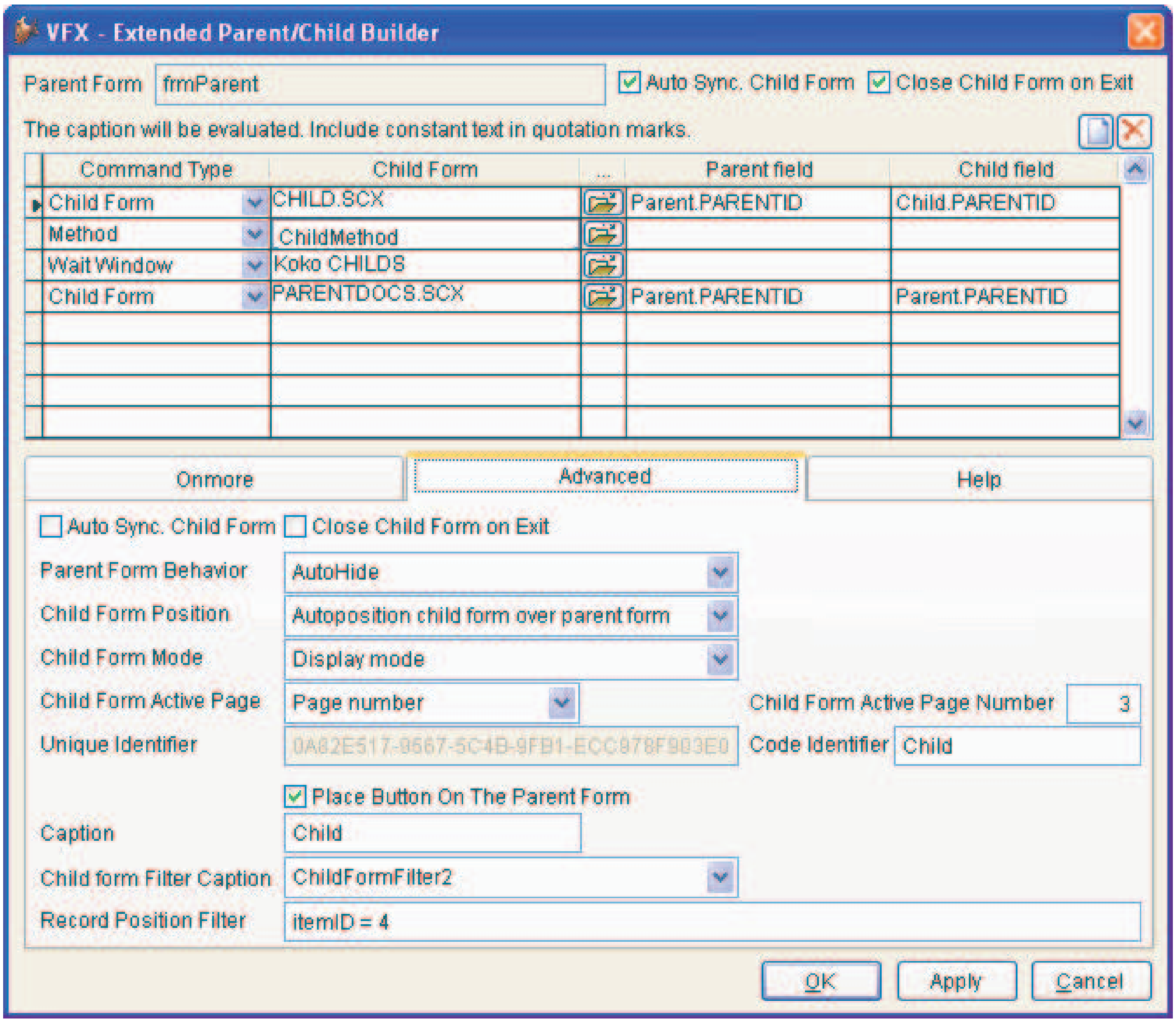
Command Type	Child Form	...	Parent field	Child field
Child Form	CHILD.SCX		Parent.PARENTID	Child.PARENTID
Method	ChildMethod			
Wait Window	Koko CHILDS			
Child Form	PARENTDOCS.SCX		Parent.PARENTID	Parent.PARENTID

Child Form	CHILD.SCX
Parent field (Fix Field Value)	Parent.PARENTID
Child field (Fix Field Name)	Child.PARENTID
Caption for child form	
Text for open form	Child
Description for open form	

[✓] Available on onMoreDialog

When the Child form option is selected, *VFX – Parent/Child Builder* retrieves information from Parent and the Child forms and automatically fills the values in fields. The builder also recognizes primary key setting when cursor adapter classes are used for data access and proposes this field to establish the relation between Parent and Child forms.

If the checkbox *Available on onMoreDialog* is marked, the respective child function will be visible in *onMoreDialog*.



VFX - Extended Parent/Child Builder

Parent Form: frmParent ☑ Auto Sync. Child Form ☑ Close Child Form on Exit

The caption will be evaluated. Include constant text in quotation marks.

Command Type	Child Form	...	Parent field	Child field
Child Form	CHILD.SCX		Parent.PARENTID	Child.PARENTID
Method	ChildMethod			
Wait Window	Koko CHILDS			
Child Form	PARENTDOCS.SCX		Parent.PARENTID	Parent.PARENTID

Onmore | Advanced | Help

☐ Auto Sync. Child Form ☐ Close Child Form on Exit

Parent Form Behavior	AutoHide
Child Form Position	Autoposition child form over parent form
Child Form Mode	Display mode
Child Form Active Page	Page number
Child Form Active Page Number	3
Unique Identifier	0A82E517-9667-5C4B-9FB1-ECC978F903E0
Code Identifier	Child
Place Button On The Parent Form	☑
Caption	Child
Child form Filter Caption	ChildFormFilter2
Record Position Filter	itemID = 4

OK | Apply | Cancel

If the checkbox *Auto Sync. Child Form* is marked, the records displayed in Child form are automatically synchronized while moving the record pointer in the Parent form.

If the checkbox *Close Child form on Exit* is marked, the Child form will automatically be closed when Parent form is closed. This checkbox can be marked only when *Parent Form Behavior* is not set to *Auto Close*.

When a new Child form is added these properties are set by default to values that are set in the Parent form. Values for Parent form are entered in the right top corner of Builder form. If these default (parent) settings are changed, a question is displayed if you want to change the corresponding properties for all Child forms.

From combobox *Parent Form Behavior* you can specify the form behavior among three available choises: *None*/*AutoClose*/*AutoHide*. When *None* is selected, the behaviour of Parent form will not be changed. When Auto Close is selected – the Parent form will be automatically closed when Child form is invoked. In this case, Child form can be open only if Parent form is in View mode (*thisform.nformstatus=0*). Selecting Auto Close will unmark the checkbox *Close Child form on Exit* to prevent cyclic forms closing. When *Auto Hide* is selected, the Parent form will be hidden when Child form is opened and will be shown again when Child form is closed.

With combobox *Child Form Position* can be defined where on screen will be positioned the Child form when it is open: *None*/*Autoposition child form over parent form*/*Autocenter*. When *None* is selected, Child form will be opened at the same place where it was closed last time. This is standart VFX behavior. When *Autoposition over parent form* is selected, Child form will be placed exactly at same top and left position as the Parent form. When *Autocenter* is selected, the Child form is autocentered on screen.

In combobox *Child Form Mode* it can be selected the mode which will be forced after the Child form is open: *Default*/*Display mode*/*Insert mode*/*Edit mode*. It is not allowed to select *Edit mode* when the Active Page for Child Form is set to *List Page*. When *Default* is selected the Child form will be opened according to application settings.

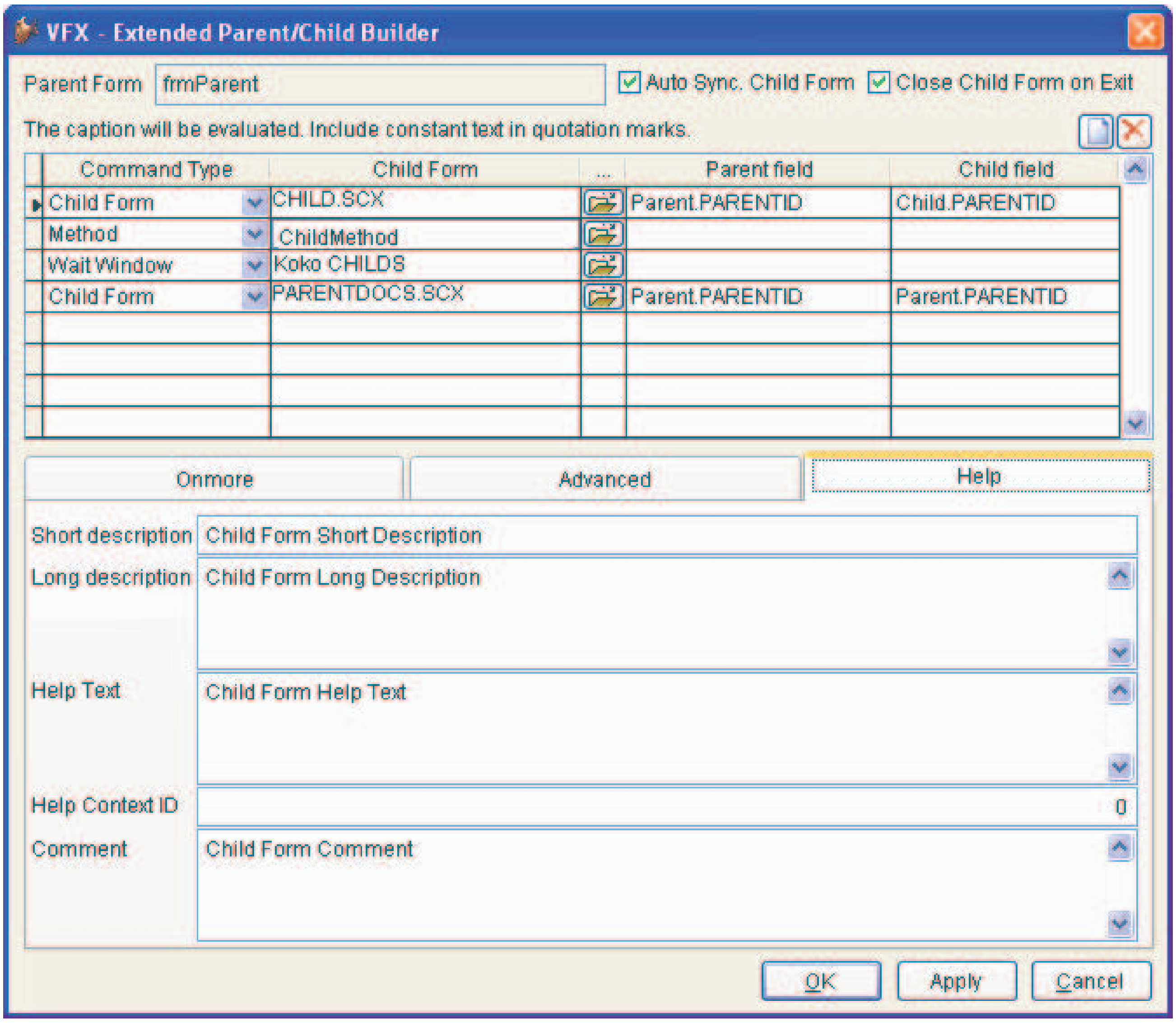
With combobox *Child Form Active Page* can be choosen the page, which will be activated when Child form will be opened: Default /Edit page /List page /Page number. When Page number is selected, the textbox *Child Form Active Page Number* is enabled and there must be entered the number of page to be activated. It is not allowed to select List Page when Child Form Mode is set to Edit Mode. Default means that the active page of the Child form will be activated according to resource file.

In the textbox *Unique Identifier* is shown the unique identifier, which is automatically generated when a new Child form is inserted. It cannot be modified. This value can be passed as a parameter to onMore method, for invoking the Child form.

In the textbox *Code Identifier* can be entered a short code for the seleclected Child form. This code should be unique among all codes for listed Child forms. The code can be edited later if needed. This value can also be used as a parameter in onMore method, for invoking the Child form.

The combobox *Child form Filter* is designated for selecting a filter which will be applied on Child form. The filters, among which it is possible to select, must be saved in advance as system filters for the selected Child form.

In the textbox *Record Position Filter* can be entered an expression which will be evaluated and the record pointer in Child form will be located to the corresponding recod.


VFX - Extended Parent/Child Builder

Parent Form: frmParent

[✓] Auto Sync. Child Form [✓] Close Child Form on Exit

The caption will be evaluated. Include constant text in quotation marks.

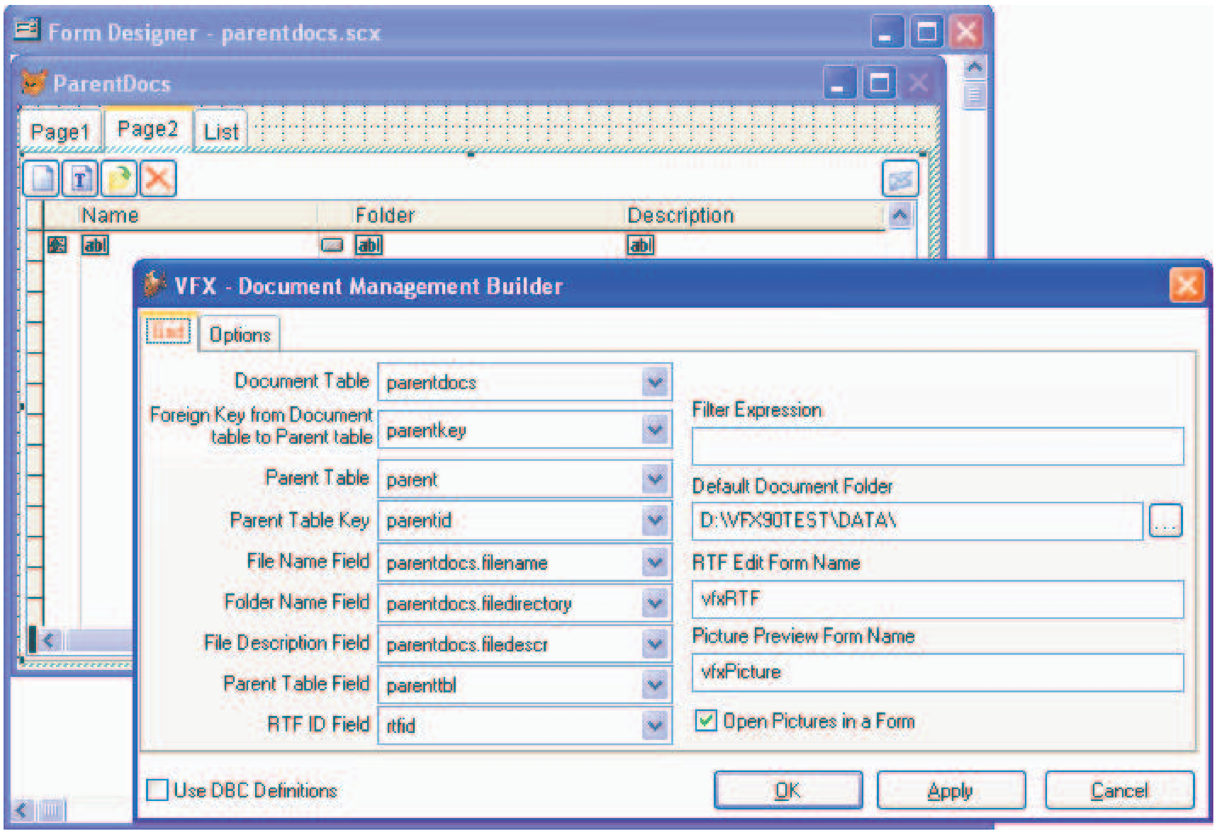
Command Type	Child Form	...	Parent field	Child field
Child Form	CHILD.SCX		Parent.PARENTID	Child.PARENTID
Method	ChildMethod			
Wait Window	Koko CHILDS			
Child Form	PARENTDOCS.SCX		Parent.PARENTID	Parent.PARENTID

Short description	Child Form Short Description
Long description	Child Form Long Description
Help Text	Child Form Help Text
Help Context ID	0
Comment	Child Form Comment

On the third page you can enter information that will be used as help text. This page is foreseen for later expansion and is not yet used.

VFX – Document Management Builder

The class *cDocumentManagement* is designated to maintain all kind of documents (i.e. Word, Excel or Zip files) related to application data. The class *cDocumentManagement* is a container that manages docuemtns as child records related to current data record in the active form. This class allows the end-user to open corresponcent documents as well as to send them as attachment in an e-mail. It also allows the end-user to enter and modify RTF texts as internally kept documents.



Document Table	parentdocs
Foreign Key from Document table to Parent table	parentkey
Parent Table	parent
Parent Table Key	parentid
File Name Field	parentdocs.filename
Folder Name Field	parentdocs.filedirectory
File Description Field	parentdocs.filedescr
Parent Table Field	parenttbl
RTF ID Field	rtfid
Filter Expression	
Default Document Folder	D:\VFX90TEST\DATA\
RTF Edit Form Name	vfxRTF
Picture Preview Form Name	vfxPicture

✓ Open Pictures in a Form

☐ Use DBC Definitions

The class *cDocumentManagement* can be added to any existing form.

cDefaultDocumentFolder – Default folder for documents.

cFilterExpression – Filter expression to be applied.

lOpenPicturesInForm – If this property is set to *.T.*, picture documents are opened in a VFX form. The name of this form is specified in *cPicturePreviewFormname* property. If the property value is set to *.F.*, picture documents are opened with the application, which is associated with their file extension in Windows Explorer. The default value is *.F.*

cPicturePreviewFormname – Name of the form to be used to preview picture documents. The default value is *VFXPicture*.

cPicturePreviewCaption – The string entered in this property will be passed to picture preview form and will be used by that form as a caption.

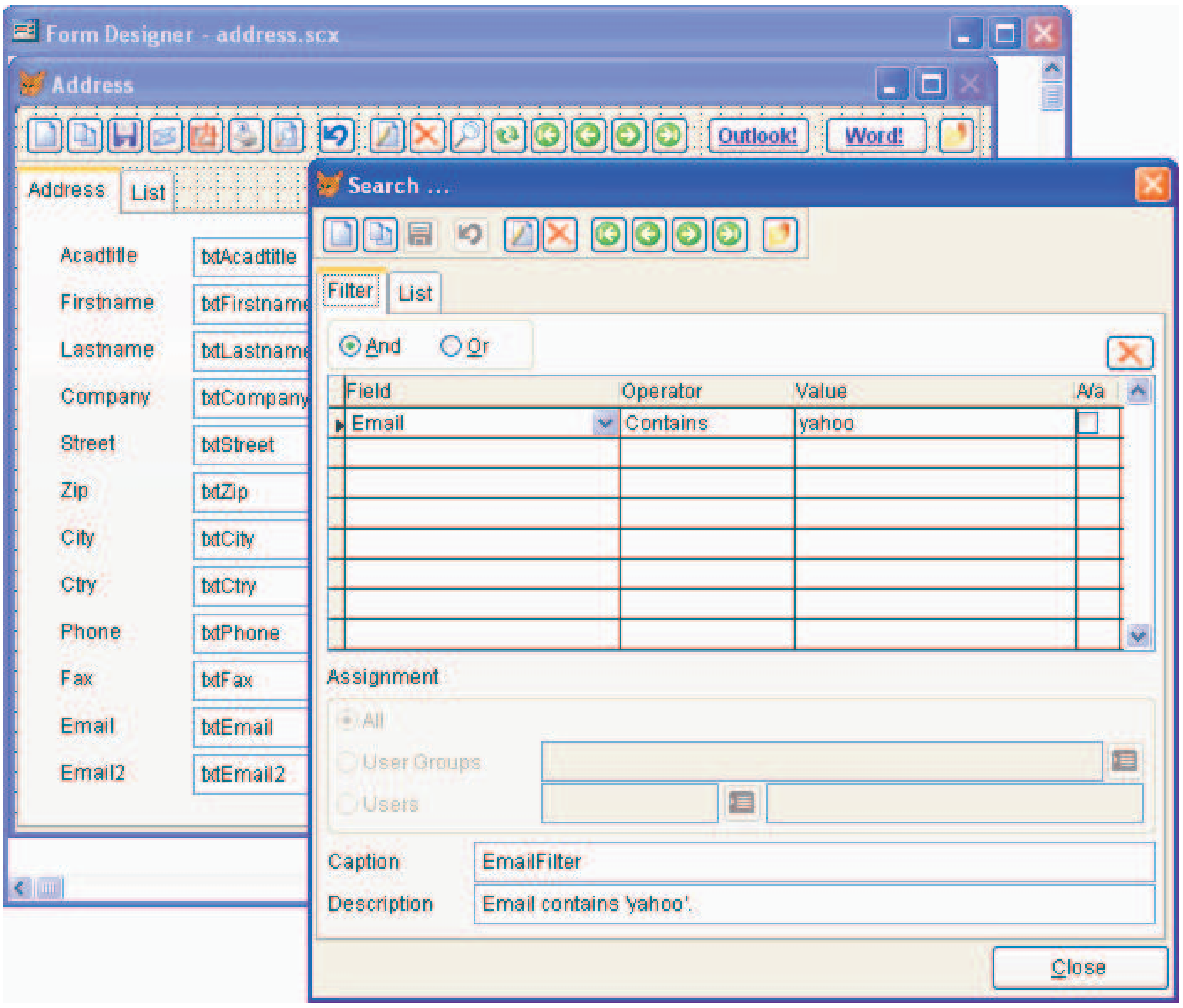
cRTFFormName – Name of the form to be used to edit RTF documents. The default value is *VFXRTF*.

cParentTableFieldName – Name of the field in the Document table, where is stored the name of parent table to which the current document belongs. The default value is *ParentTbl* (for table *vfxDocuments*).

cRTFIDFieldName – Name of the field in the Document table used as a foreign key to the related table used to keep RTF texts. In this field is stored value of *rtfID* field of *vfxRTF* table. The default value is *rtfID* (for *vfxDocuments* table).

VFX – Filter Builder

Using *VFX - Filter Builder*, in design time can be created system filters that at run time are considered as read-only filters and cannot be changed or deleted by end users.
To run the *VFX - Filter Builder*, it is necessary to have an open project and an open form.



Acadtitle	txtAcadtitle
Firstname	txtFirstname
Lastname	txtLastname
Company	txtCompany
Street	txtStreet
Zip	txtZip
City	txtCity
Ctry	txtCtry
Phone	txtPhone
Fax	txtFax
Email	txtEmail
Email2	txtEmail2

[✓ And] Or

Field	Operator	Value	A/a
Email	Contains	yahoo	☐

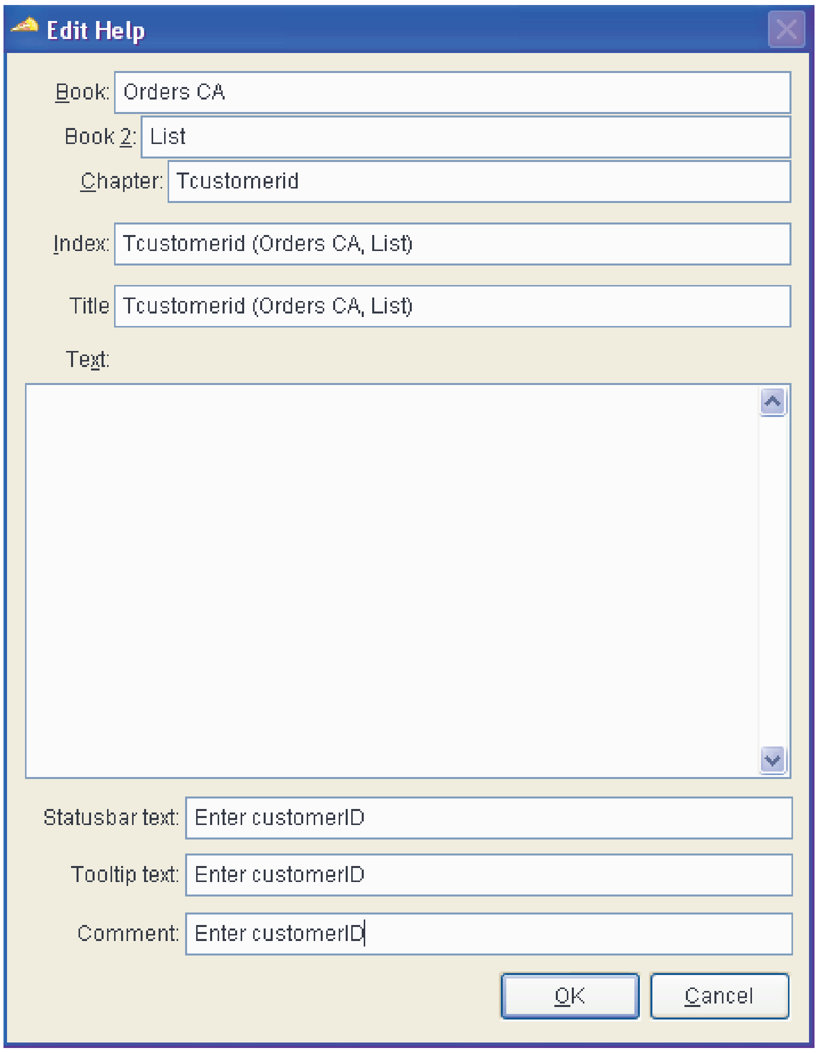
Assignment
[✓ All] User Groups Users

Caption	EmailFilter
Description	Email contains 'yahoo'.

The filter conditions are created is the same way as in Search dialog at run time.
The fields, available for constructing filter expressions are read from the active form in similar way.
On the *List* page can be seen all system filters that are already defined for current active form.
System filters are assigned always for all Users. So all users can read them but not modify them.

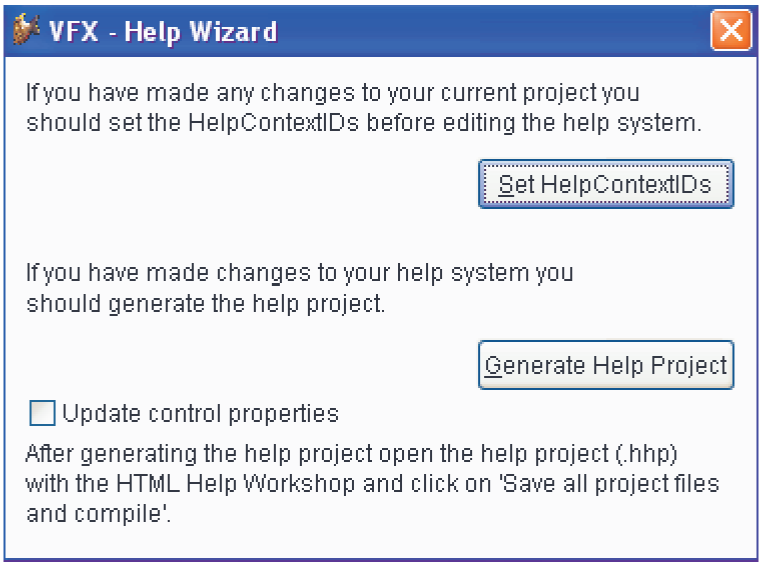
Extended help editor

With the help editor is also possible to enter and edit StatusBarText, ToolTipText and Comment properties of a control.



Book:	Orders CA
Book 2:	List
Chapter:	Tcustomerid
Index:	Tcustomerid (Orders CA, List)
Title	Tcustomerid (Orders CA, List)
Text:	
Statusbar text:	Enter customerID
Tooltip text:	Enter customerID
Comment:	Enter customerID

If the field, corresponding to the property in *vfxhelp.dbf* is empty, values of the control's properties is read.
The information entered in this form is saved in the table *vfxhelp.dbf* in the project folder. In order to update the properties of the controls with values saved in *vfxhelp.dbf* mark the checkbox *Update control properties* in *VFX – Help Wizard*, when creating a Help Project for your application. For all controls for which there is an entry in *vfxhelp.dbf*, this will overwrite StatusBarText, ToolTipText and Comment with the value stored in the table.


If you have made any changes to your current project you should set the HelpContextIDs before editing the help system.

If you have made changes to your help system you should generate the help project.

After generating the help project open the help project (.hhp) with the HTML Help Workshop and click on 'Save all project files and compile'.

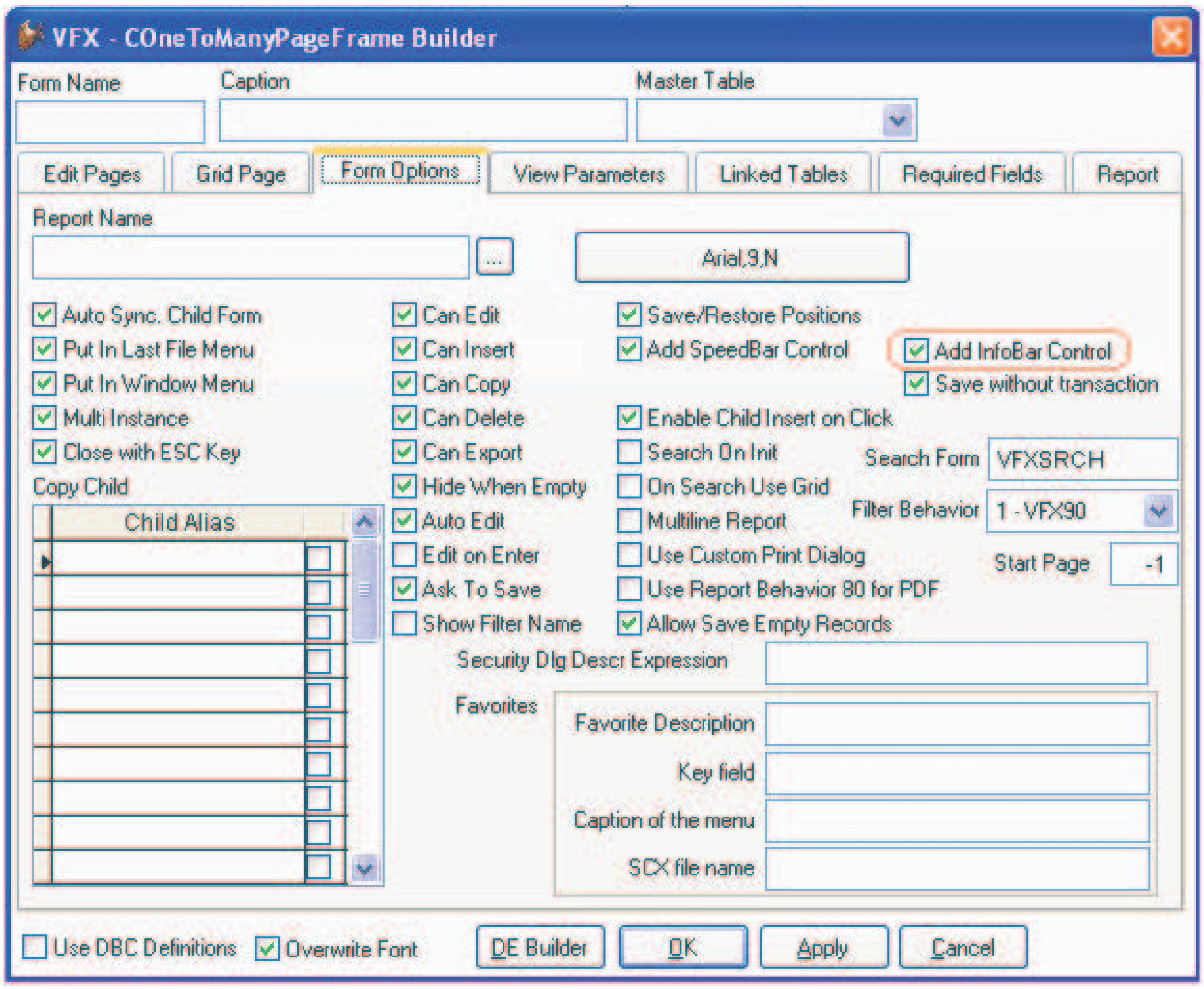
** Warning**
If the texts for StatusBarText, ToolTipText and Comment are empty in *vfxhelp.dbf* and the properties of the control are not empty, they will be erased. If that is the case do not mark the checkbox *Update control properties.*

Updating Config.vfx structure

When the structure of the file *Config.vfx* is modified at the developer's side, for instance when columns are added or deleted, a new file named *vfxconfigstructure* is created and included in the project, and thus in the EXE file at build. This file describes the new structure of *Config.vfx* file. When the exe file is run first tme at customers' side, the structure of *config.vfx* is automatically updated. This is done right before the database update by the function *UpdateConfigVFXStructure.* After that is started Database update, if necessary.

Record Information Container

The class *cInfoBar* is designated to show a set of important information to end-users at top part of the data form. The developers can this container to the forms and place there controls which show information to users. Form builders provide developers with an easy way to add *cInfoBar* control to a form. On options page there is a checkbox if *InfoBar* container will be placed on the form.


VFX - COneToManyPageFrame Builder

Form Name	
Caption	
Master Table	
Report Name	
Font	Arial,9,N
Search Form	VFXSRCH
Filter Behavior	1 - VFX90
Start Page	-1
Security Dlg Descr Expression	
Favorite Description	
Key field	
Caption of the menu	
SCX file name	

☑ Auto Sync. Child Form
☑ Put In Last File Menu
☑ Put In Window Menu
☑ Multi Instance
☑ Close with ESC Key
☑ Can Edit
☑ Can Insert
☑ Can Copy
☑ Can Delete
☑ Can Export
☑ Hide When Empty
☑ Auto Edit
☐ Edit on Enter
☑ Ask To Save
☐ Show Filter Name
☑ Save/Restore Positions
☑ Add SpeedBar Control
☑ Add InfoBar Control
☑ Save without transaction
☑ Enable Child Insert on Click
☐ Search On Init
☐ On Search Use Grid
☐ Multiline Report
☐ Use Custom Print Dialog
☐ Use Report Behavior 80 for PDF
☑ Allow Save Empty Records
☐ Use DBC Definitions
☑ Overwrite Font

When selected, the InfoBar control is placed on the form immediately below the SpeedBar Control.

Additional Automatic fields in tables

New automatic filled fields are added in the VFX application. These fields are designated to be used when records are synchronized. In these fields, in every data table, is saved additional information when lately the record has been modified. They hold information for date and time when the record has been changed and checksum used to check validity of the record. These fields are filled up automatically every time when the record has been inserted updated or deleted. Names of fields where sync information should be stored are kept in properties of *goProgram:*
cSync_Date – Name of the field in data table where date of last modification is kept;
cSync_Time – Name of the field in data table where time of last modification is kept;
cChkVal – Name of the field in data table where is kept checksum for data record.
The properties *cSync_Date*, *cSync_Time* and *cChkVal* of *goProgram* are keyed up in *VFX – Application Builder*.

Other enhancements for developrs

Function IsTerminalServer()

Using this function developer has the chance to check if the current machine is working works in terminalserver session. The function is placed in *vfxfunc.prg*. The VFP function OS(10) provides information only if terminal server is installed on the computer, not about the session mode.
The function *IsTerminalServer()* returns .T. if the application is running in a Terminalserver session.

Function GetColorDepth()

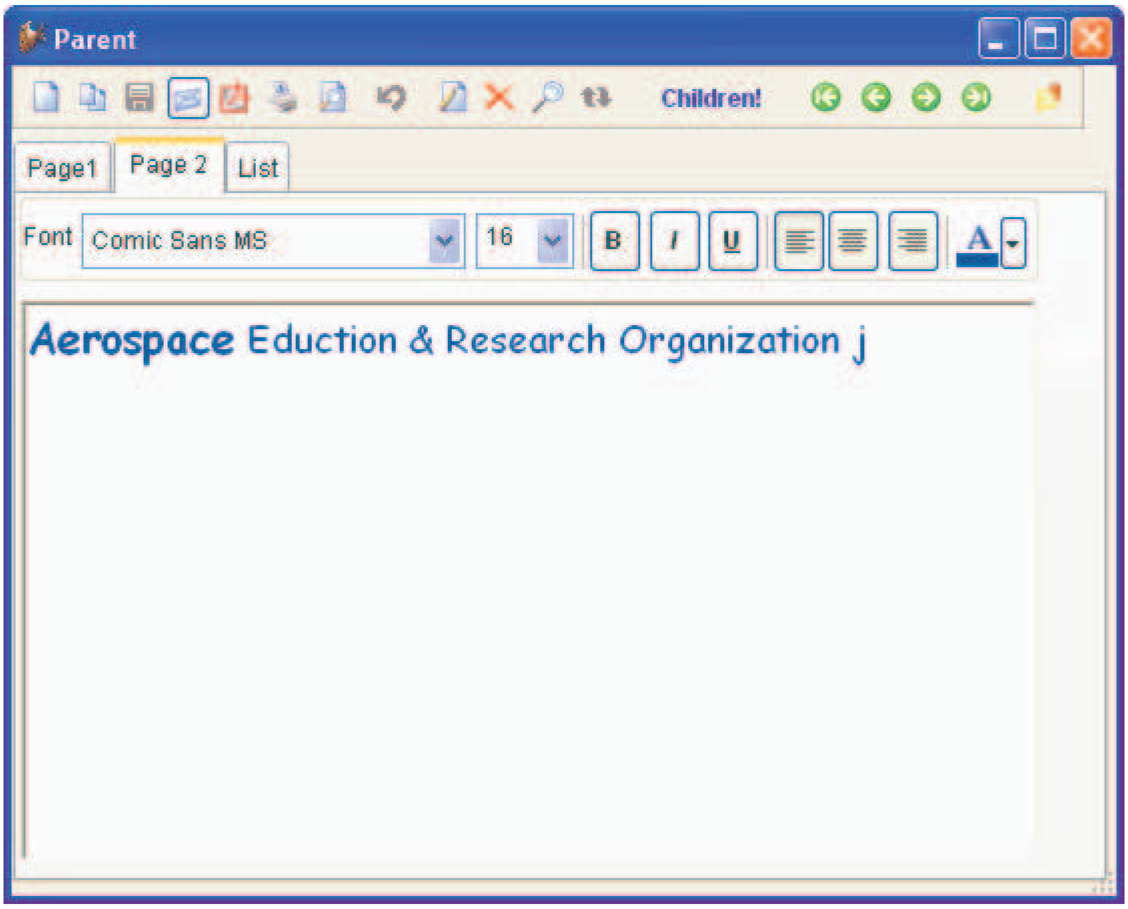
The system color depth can be retrieved using the function *GetColorDepth()*, placed in *vfxfunc.prg*. The function returns an integer value, corresponding to the color depth in bits. For instance the value 8 corresponds to 256 colors and the value 32 – to 32bit color settings.

New features for end-users

The class cRTFControl

This class is designated for easy editing texts in RTF format. From the toolbar, plced in this class, it is possible to set font, font style, font size, alignment and color for the text. The interface is intuitive and very similar to MS Word.The class *cRTFControl* is placed in *vfxCtrl* library.

In Form builders, the class *cRTFControl* can be used for fields of Memo or General data type.

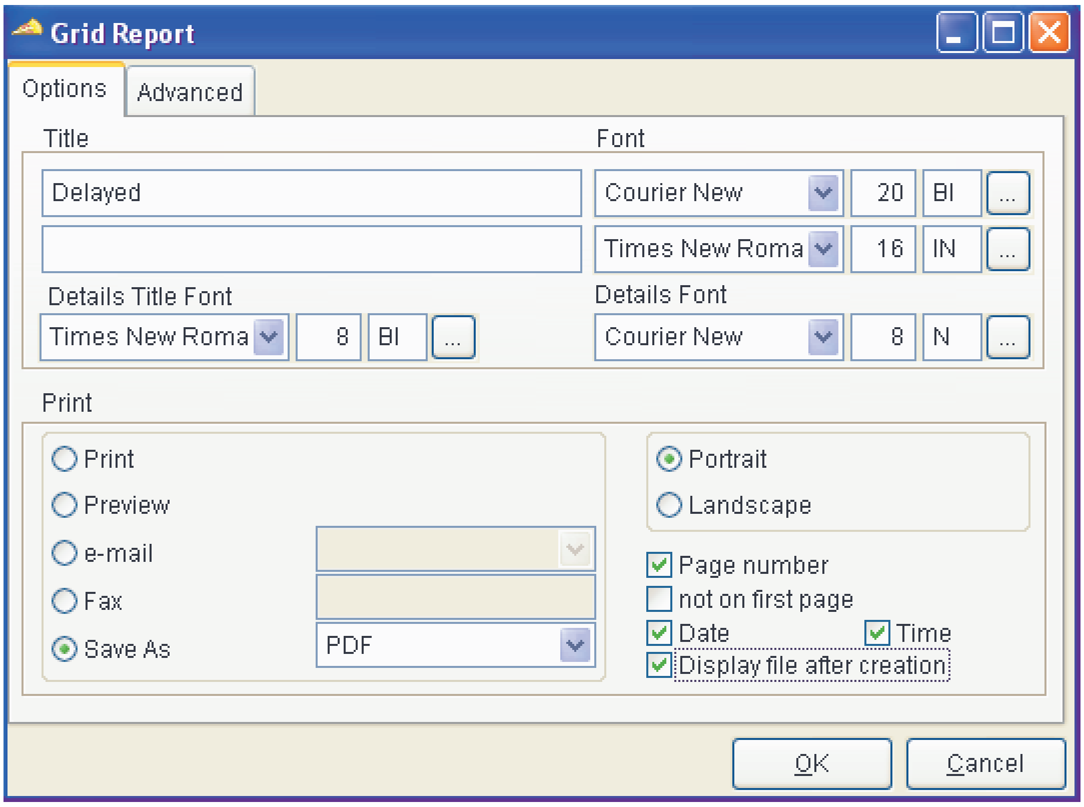


Aerospace Eduction & Research Organization j

Reports

Display generated file

This feature is available in Report dialog, when the report, based on the search grid, will be saved in a file. If *Display file after creation* checkbox is selected, after generating the file, it is displayed using the default associated application for selected file type.



Title: Delayed

Font: Courier New, 20, BI

Font: Times New Roma, 16, IN

Details Title Font: Times New Roma, 8, BI

Details Font: Courier New, 8, N

Print: ○ Print ○ Preview ○ e-mail ○ Fax [● Save As] PDF

[● Portrait] ○ Landscape

[✓ Page number] ☐ not on first page [✓ Date] [✓ Time] [✓ Display file after creation]

Enhanced Editbox

The functionality of *cEditBoxBase* class is extended. The property *lUseMemoForm* of the class, allows an additional form to be displayed to the end users, for better displaying andediting of the content. When the property is set to *.T.*, end users can invoke the memo form for edition. The feature is also available through a new entry in the context menu of the control – “Edit”. The edit form is based on the class *cMemoForm*. The form can be invoked either via the menu option or by a double mouse click in the editbox.

When the data form is in View mode, the memo form displays the contents of the editbox control readonly. If the data form is in Edit mode, the text of memo field can also be edited. Developer can control if before displaying memo form, the data form will be forced into edit mode. If *goProgram.lCallOnEditForEditBox* is set to *.T.* the form will be switched into Edit mode before opening the memo-form.

The property *lUseMemoForm* of cEditbox controls can be controlled globally by the property *goProgram.nUseMemoForm*. The value of *goProgram.nUseMemoForm* can be set in *VFX – Application Builder*

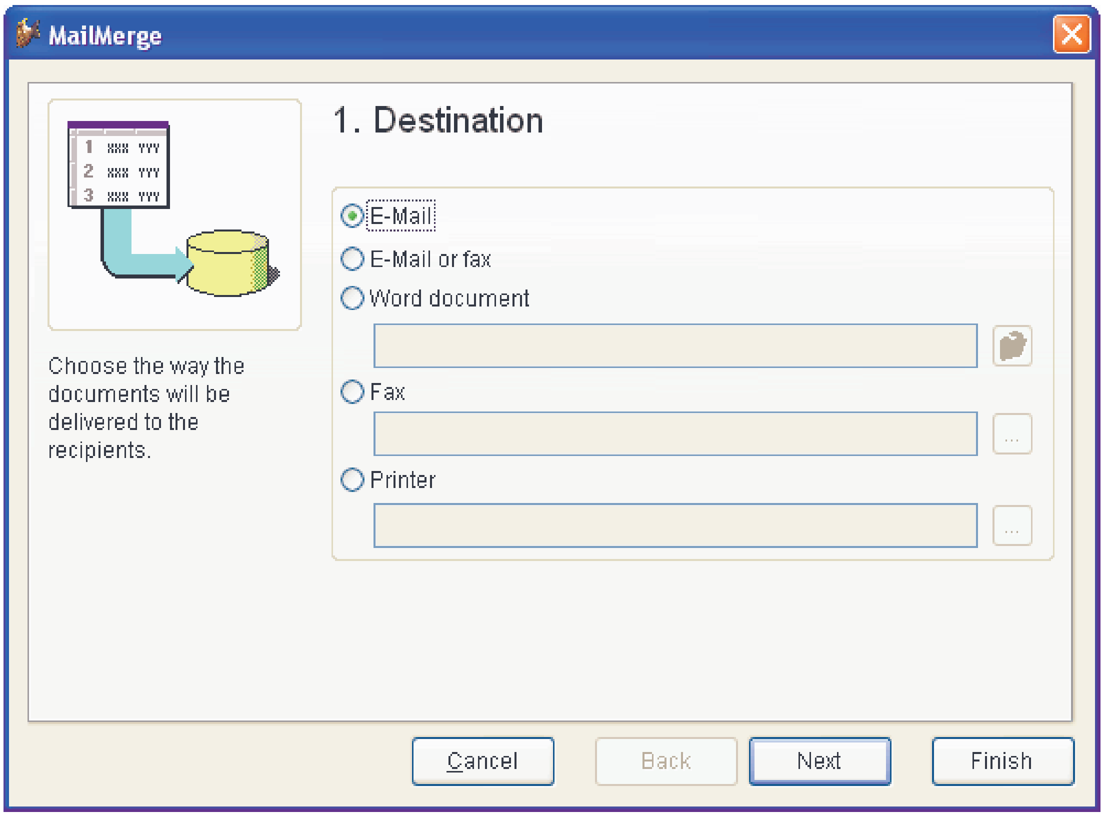
The property *lSingleLineEditBox* of *cEditBoxBase* objects controls if the control displays text in a single line, exactly like a textbox. If value of property *lSingleLineEditBox* is set to *.T.*, the text in the editbox control is displayed in one line. In this case calling the memo form is automatically disabled. It is also not possible user to insert carriage return chanacters in the text – Enter key is not includde in the text when pressed and when the text is processed programatically *CHR(13)* characters are removed. When the control works in this mode, scrollbars are switched off.

The property *lSingleLineEditBox* of *cEditBoxBase* class is globally controlled by the property *nSingleLineEditBox* of *goProgram*. The value of *goProgram.nSingleLineEditBox* can be modified in *VFX – Application Builder*.

Mail Merge documents

Witht the Mail Merge wizard, end-users have the amazing chance to use their data as a Mail Merge data source and to generate Mail Merge output. As source text for generated documents can be used MS Word Mail Merge document, text file, HTML file, RTF file or simply some text, entered by user in the wizard. The result can be saved as a word document, printed, faxed or sent as an E-Mail. User is guided by the wizard through a few very intuitive steps.

In the first step the user is asked to specify desired output type.

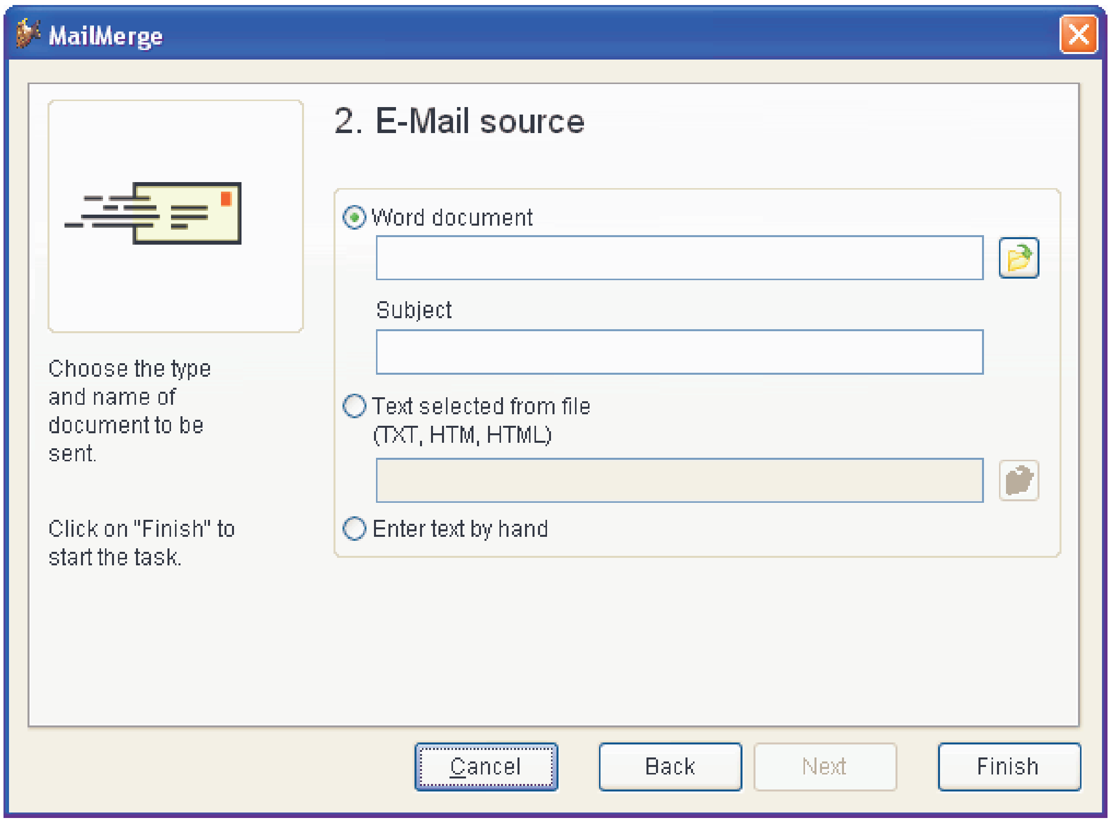

MailMerge

1. Destination

Choose the way the documents will be delivered to the recipients.

[✓ E-Mail] E-Mail or fax / Word document / Fax / Printer

On next step is selected the source for generated documents. If in first step has been selected E-Mail as output type, at second step user will be asked to select among three possible options for text for the the E-Mail body.



2. E-Mail source

[✓ Word document]

Subject

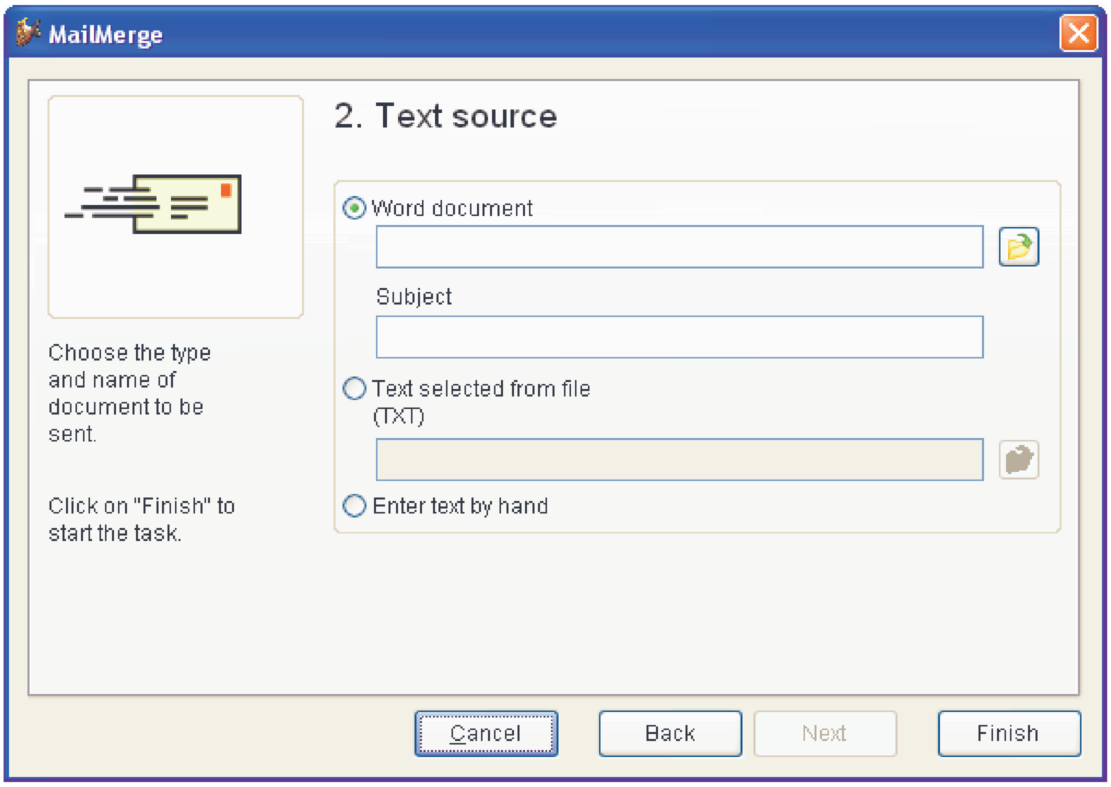
Text selected from file (TXT, HTM, HTML)

Enter text by hand

Choose the type and name of document to be sent.

Click on "Finish" to start the task.

If E-Mail or fax is selected in first step, at second step user will be asked to select among three possible sources for the e-mail (fax) body.



2. Text source

[✓ Word document]

Subject

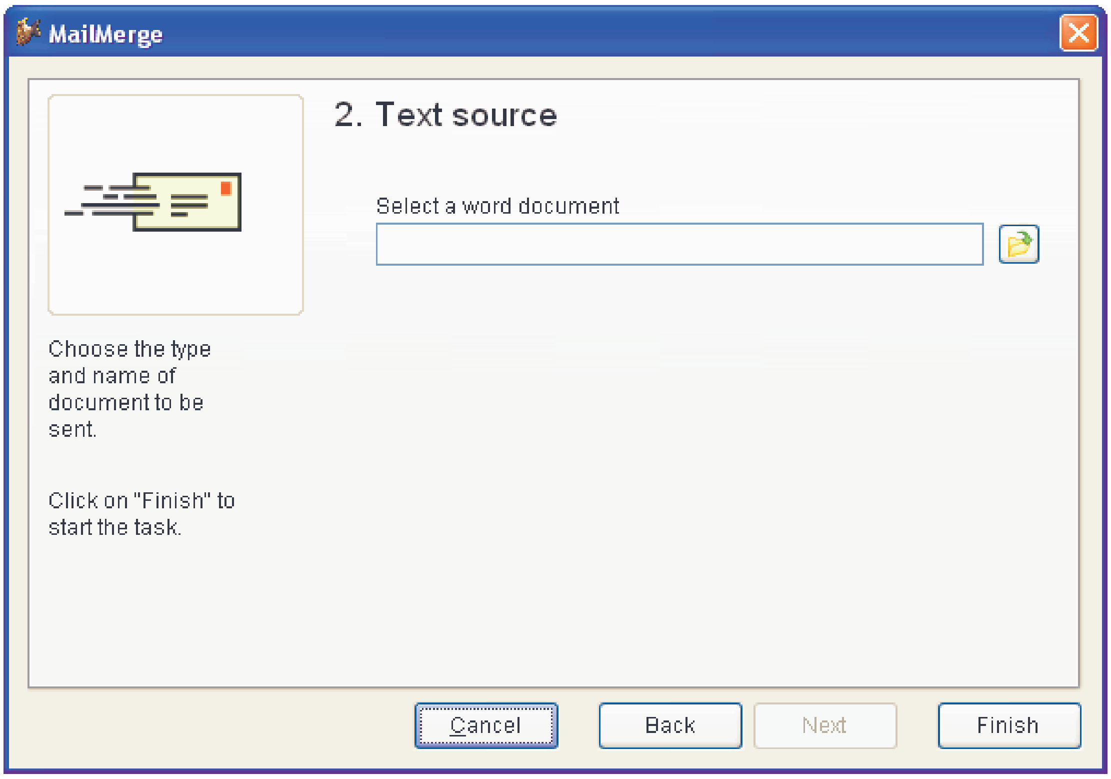
Text selected from file (TXT)

Enter text by hand

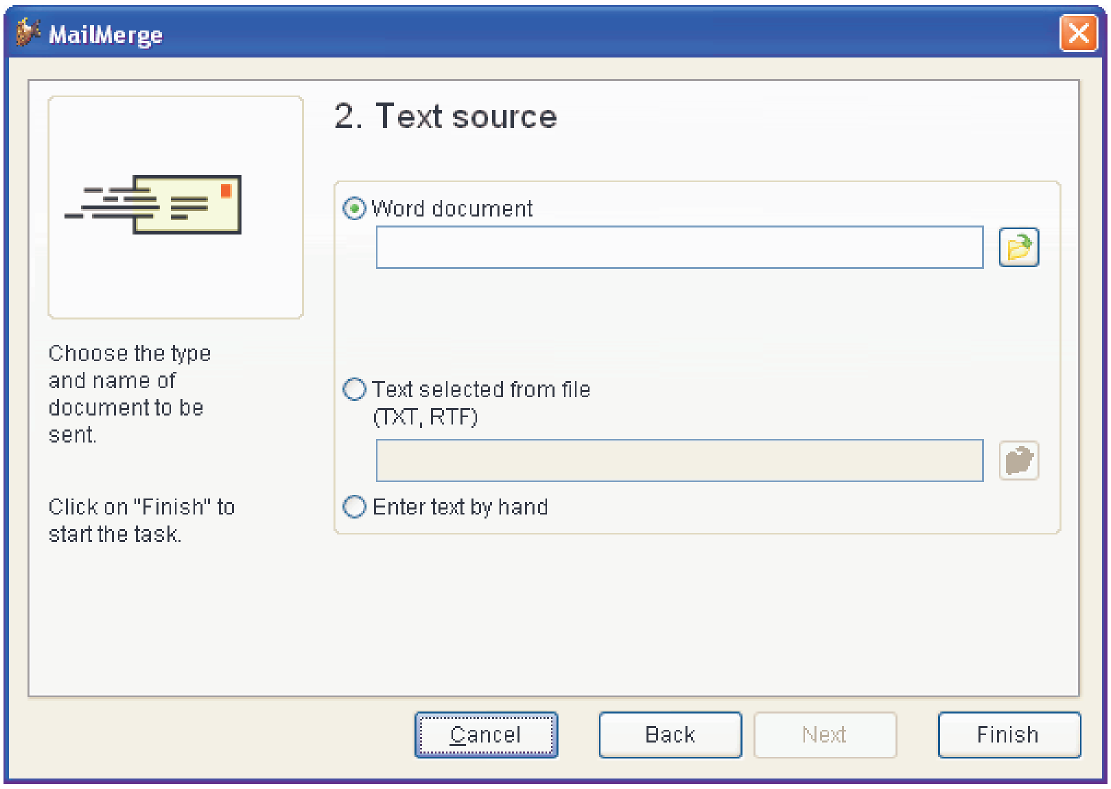
Choose the type and name of document to be sent.

Click on "Finish" to start the task.

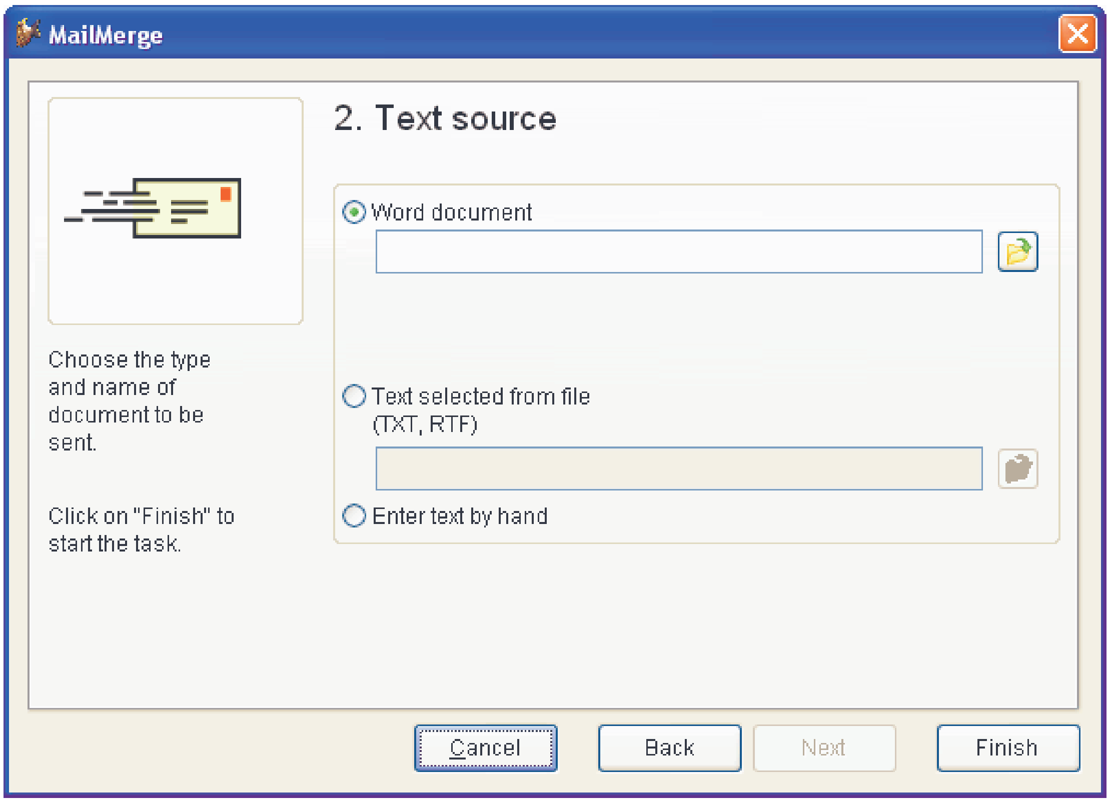
If in first step has been selected a Word Document as output, the shource text must also be a word document. At second step, user should specify the file path and name of a word document, which is mail merge document.


MailMerge
2. Text source
Select a word document
Choose the type and name of document to be sent.
Click on "Finish" to start the task.
Cancel
Back
Next
Finish

If in first step has been selected Fax as desired output, at second step user will be asked to select among three possible options for the fax body source.


MailMerge
2. Text source
Word document
Text selected from file (TXT, RTF)
Enter text by hand
Choose the type and name of document to be sent.
Click on "Finish" to start the task.
Cancel
Back
Next
Finish

If Printer is selected in first step, at second step user will be asked to select among three possible source types for text for the print document.


MailMerge

2. Text source

Choose the type and name of document to be sent.

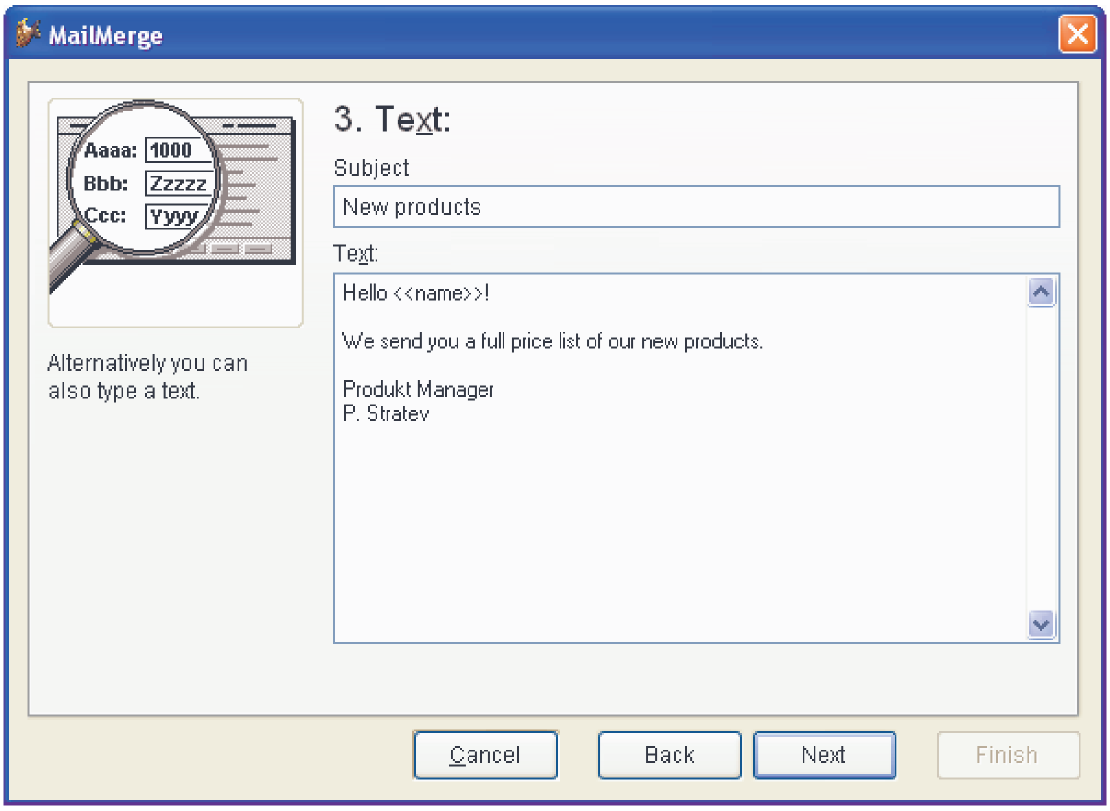
Click on "Finish" to start the task.

[✓ Word document]

Text selected from file (TXT, RTF)

Enter text by hand

When in second step has been selected that text, enered by hand will be used as document source, on third step user should write the text for document body in an *Editbox* or in *RTF* control. If output type will be Printer, only text will be entered. Additionally, if the output type is *E-Mail* or *E-Mail or fax*, is necessary user to write subject text.



3. Text:

Alternatively you can also type a text.

Subject: New products

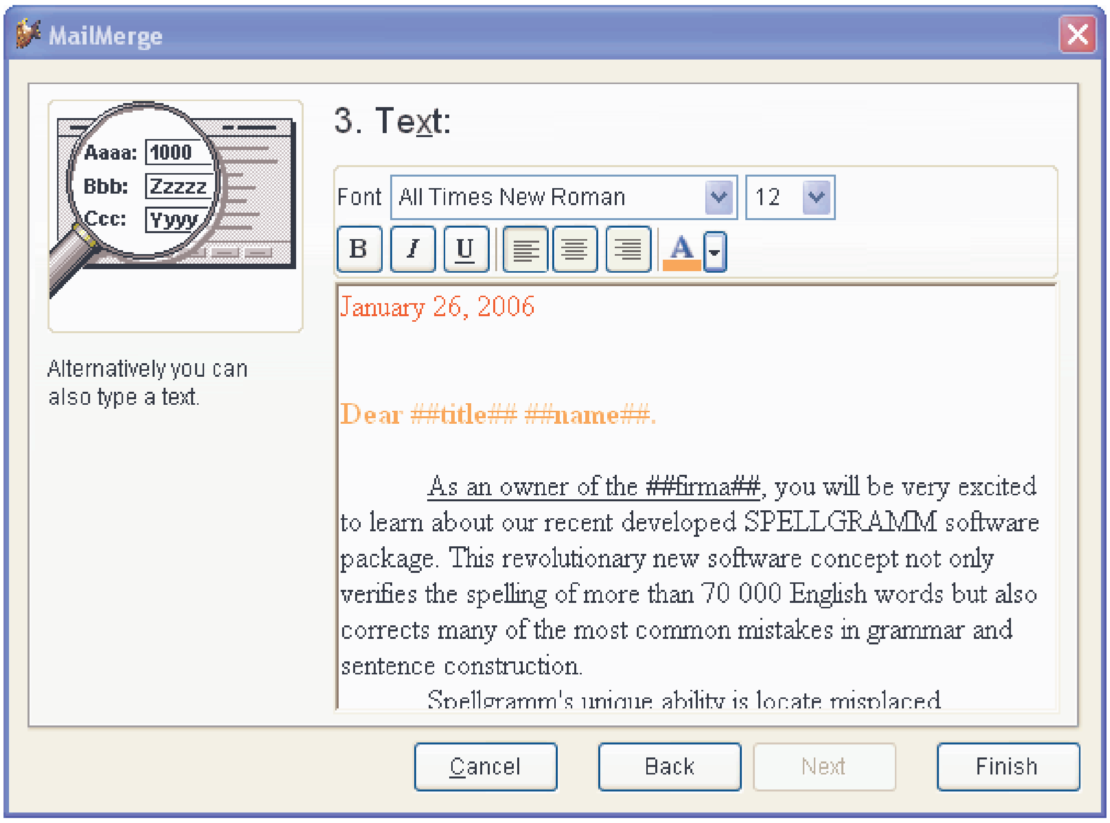
Text:

Hello <<name>>!

We send you a full price list of our new products.

Produkt Manager
P. Stratev

If a text file will be used as text source, its content is displayed here and can be edited if necessary. As an exception HTML files cannot be edited. Changes will affect the output document but will not be saved in the source file.



Alternatively you can also type a text.

January 26, 2006

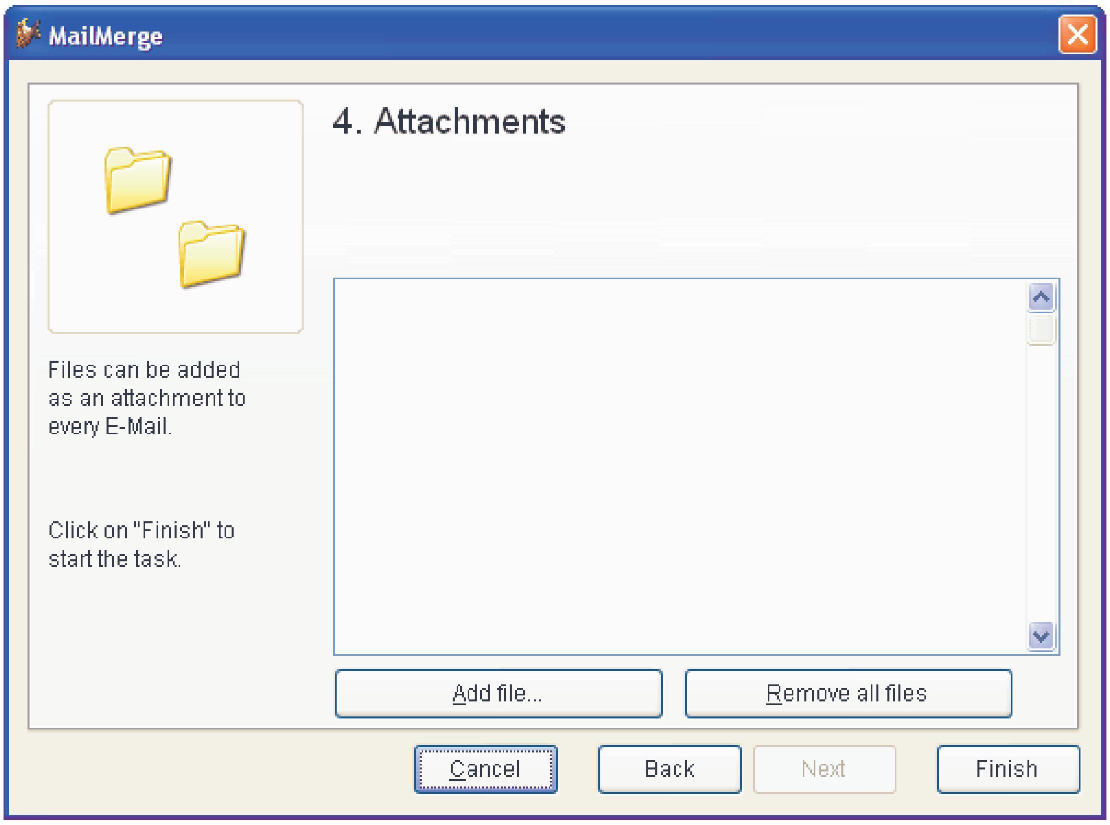
Dear ##title## ##name##.

As an owner of the ##firma##, you will be very excited to learn about our recent developed SPELLGRAMM software package. This revolutionary new software concept not only verifies the spelling of more than 70 000 English words but also corrects many of the most common mistakes in grammar and sentence construction.

Spellgramm's unique ability is locate misplaced

In this text, similar to the Word documents, can be used also data fields. The names of variables that user wants to use in the text, should be enclosed in special characters. By default the textmerge characters “<<” and “>>” are used. Developer can change this be setting *cLeftDelim* and *cRightDelim* properties of *cMailMerge* class, respectively in the derived form.

On next step the end-users have the chance to add additional files as attachment to the generated E-Mails.



Files can be added as an attachment to every E-Mail.

Click on "Finish" to start the task.

By clicking on Finish button, the mail merge documents will be generated according selected options. At last step of the wizard, after the requested action is completed, user is informed for the result – nimber of sent documents and number of fails, if there are any fails.

Class cMailMerge

This is a form class, placed in *VfxForm.vcx* class library

With this class the end users have the chance at run-time to build mail-merge documents using data kept in the application. Main features are:

1. E-Mail
   Generating bulk e-mails. The e-mail text can come from a Word document or a text file or be entered also by hand in an *Editbox*. When the e-mail text is generated, it is also possible additionally to attach to the e-mail as many files as necessary.
2. E-Mail or fax
   Sending document either per E-Mail or fax. If there is valid e-Mail address for current data record, the generated document will be sent as E-Mail. When for current record E-Mail address does not exist and there is a fax number, the document will be sent by fax.
3. Word documents
   Generating a MS Word mail merge output, based on a Word mail merge document. The Word mail merge document can be edited in Word as needed.
4. Fax
   Bulk sending faxes based on a Word mail merge document, a text or RTF files, or text entered by hand in RTF format.
5. Merge and Send documents to Printer.
   Printings letters based on a Word mail merge document, on a text or RTF files, or text entered by hand in RTF format.

In order to use the class, it is needed to have prepared a cursor with all fields used in merged documents or texts. Additionally it is needed in cursor to have fields for E-mail address, cc and bcc address, as well as faxNumber.

Properties

cDataSource	Contains the name of DataSource for the Word Document. It is used from word or wizard to merge given texts. It is expected all fields used in merged documents or texts to be available.
cMailAddressFieldName	Contains field name for e-mail address. It is expected this to be a field from *cDataSource* table. It is used for obtaining recipient address when sending emails.
cCCFieldName	Contains field name for CC e-mails addresses. It is expected this to be a field from *cDataSource* table. It is used when sending emails for filling cc recipients address.
cBCCFieldName	Contains field name for BCC e-mails addresses. It is expected this to be a field from *cDataSource* table. It is used when sending emails for filling bcc recipients address.
cFaxNumberFieldName	Contains field name for fax numbers. It is expected this to be a field from *cDataSource* table. It is used to obtain the recipient fax number when sending fax.
cLeftDelim	Left delimiter to use for the text merge option. By default it is “<<”. This string must be used as left delimiters in text files or texts enter by hand, which will be merged.
cRightDelim	Right delimiter to use for the text merge option. By default it is “>>”. This string must be used as right delimiters in text files or texts enter by hand, which will be merged.
cMergeText	For internal use only. Used to hold the text that will be merged.
nEmailsNotSent	For internal use only. Used to count emails or faxes faild to be sent and to display this value on last wizard step.
nEmailsSent	For internal use only. Used to count sent emails, prepared documents, sent faxes or printed sheets number and to display this value on last wizard step.
nPreviousPageNum	For internal use only. Used to keep subsequent number of displayed wizard step.

Methods

LoadFileContent	Loads the content of file that is selected in second step - *Source*, in field *Text*, available for edition in third step – *Text*.
SendMails	Performs the process of generating merge documents, according to selected options. It calls one of the following two methods: *SendThroughMapi* or *SendThroughOleWord*.
SendThroughMapi	Merges the text and sends emails using the standard VFX class *cEmail*. Also send faxes and prints documents.
SendThroughOleWord	This method is used when the source text is a word mail merge document. An automation instance of MSWord is created and its features are used to send merged document by e-mail, to make new word document or to send it to fax or to printer.

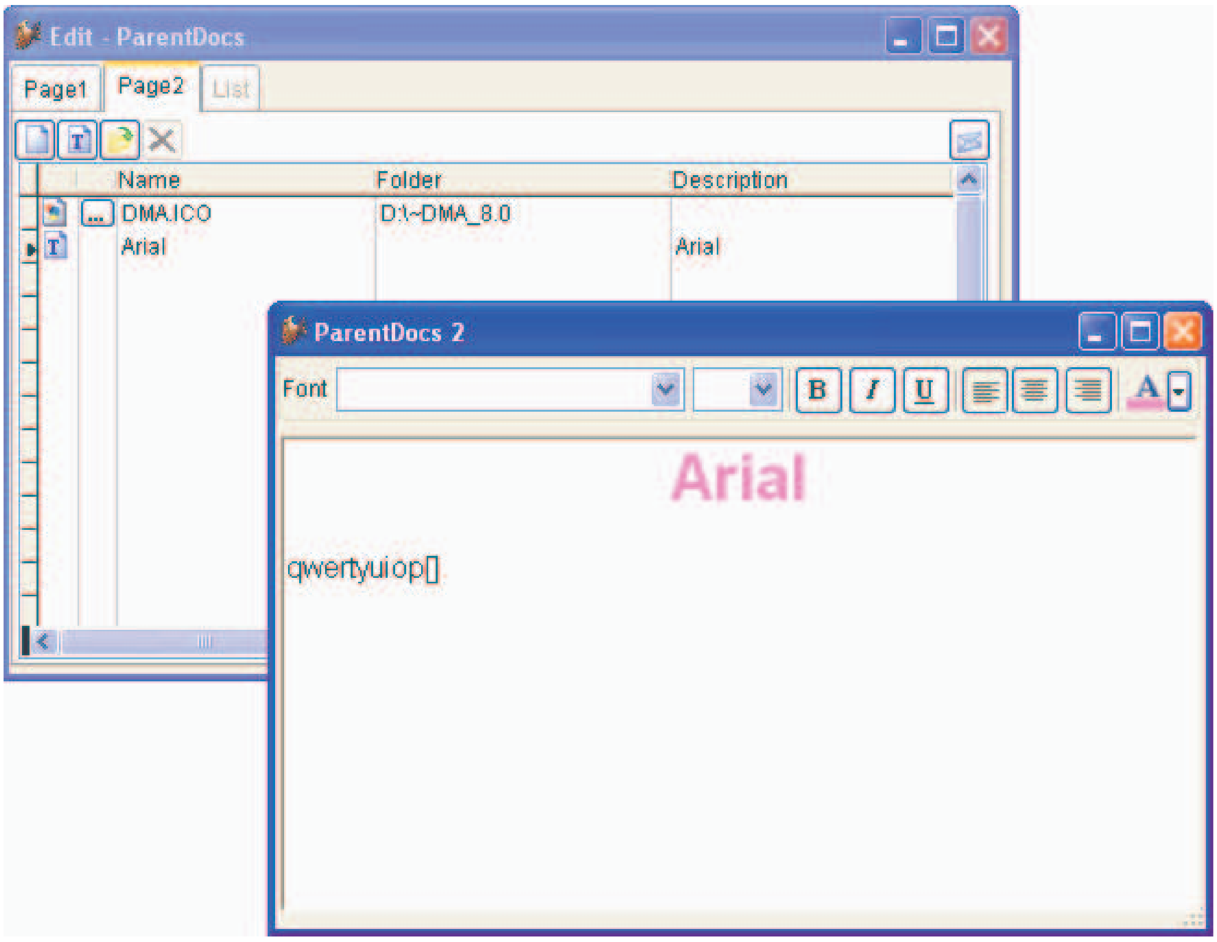
Extended audit information

In the *Tools* menu of the VFX application is added a new submenu– *Audit*.
It contains the currently available *Audit-trail* option, as well as a new one - *Audit Information*. At runtime *Audit-trail* opens the audit –trail for current data record. *Audit Information* option calls a new form, which displays detailed information about all the changes made in the main table for active data form. The *Audit Information* form has the full functionality of a regular VFX DataForm, including incremental search, grid report, export etc. In order to display audit information, it is required a form to be open.

Document management

The class *cDocumentManagement* is designated to maintain all kind of documents (i.e. Word, Excel, and PowerPoint) related to application data. The *cDocumentManagement* class is a container that manages child records related to current record of main table. The document management allows the end-user to open related documents and also to send them as attachment in an e-mail. It also allows the end-user to keep RTF texts related to current record and to modify them.

The class can easily be added to the existing forms.



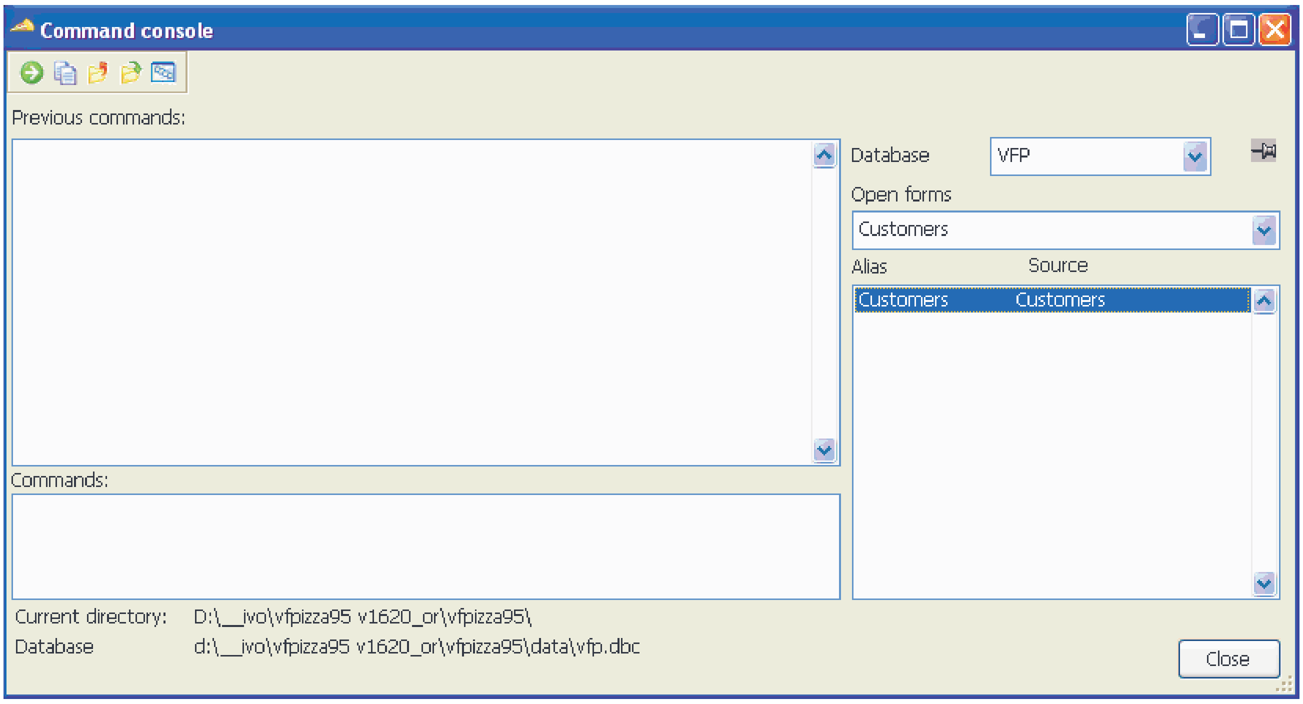
The RTF edit form is open as a child form, when in Document management is selected a record of type RTF text. The button *New RTF* adds a new empty RTF Document and also opens the RTF edit form if it is not open. When in Document management is selected a record of type RTF text and the user clicks *Open* button, the RTF edit form will open. If the user clicks the *E-mail* button, the RTF text will be saved in a file and send as e-mail attachment.

When an RTF document is deleted in Document management, the corresponding RTF text from VFXRTF table will be also deleted.

When a parent record is deleted, all documents corresponding to this data record and RTF texts that belong to these documents will be deleted.

VFX Command Console

For users with administrator rights is available a small Command Console form. Using it, it is possible to implement some base VFP functionality if needed. User with administrative rights can execute FoxPro commands from within a running VFX application.



Current directory: D:\__ivo\vfpizza95 v1620_or\vfpizza95\
Database d:\__ivo\vfpizza95 v1620_or\vfpizza95\data\vfp.dbc

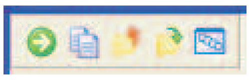
The availability of VFX Command Console is controlled by the option *Enable Command Console* in *VFX Application Builder*. The form can be started under *Tools* menu.

The commands that are to be executed should be entered in the *Commands* editbox. An entered command can be executed either by pressing the *Enter* key or with the form's toolbar, by clicking *Run* button. A history of all executed commands is kept in the listbox *Previous commands*. Commands that have already been executed can be copied into *Commands* editbox by Right click on the command in *Previous commands* list.

The *Database* combobox allows the selection among the open databases.
When a form is selected in the *Open forms* combobox, cursors in its data session are displayed in the list below. This list allows the selection of an alias. At the bottom of the form are displayed the current directory and current database.

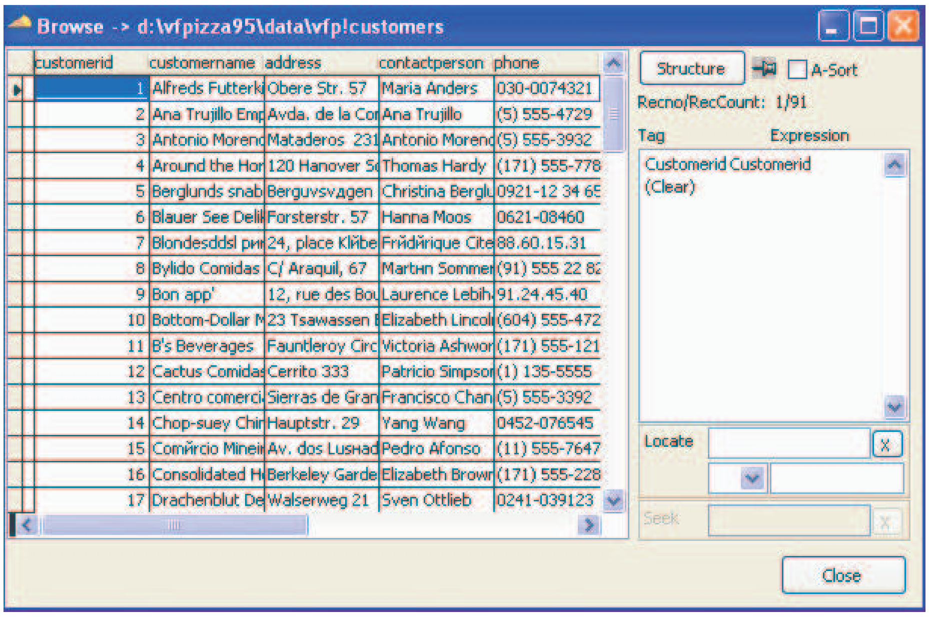
The button *Always on top* toggles the *AlwaysOnTop* property of the form On/Off. Using it, it is possible to control if the form will remain topmost form.

The toolbar, placed on the form, offers some additional functionality.



Run – Runs the command entered in *Commands* or the one selected in the *Previous commands* list
Copy – Copies the selected command from the *Previous commands* list to the *Commands* section
Use – Closes the currently selected alias (equal to USE command)
Use ? in 0 – Opens a table in a new work area (equal to USE ? IN 0 command)
Browse – Opens the *VFX Browse* form that allows viewing of the currently selected alias contents

When a BROWSE command is executed or the *Browse* button is pressed, the *VFX Browse* form is invoked.


Browse -> d:\vfpizza95\data\vfp!customers

customerid	customername	address	contactperson	phone
1	Alfreds Futterki	Obere Str. 57	Maria Anders	030-0074321
2	Ana Trujillo Emp	Avda. de la Cor	Ana Trujillo	(5) 555-4729
3	Antonio Morenc	Mataderos 231	Antonio Morenc	(5) 555-3932
4	Around the Hor	120 Hanover Sc	Thomas Hardy	(171) 555-778
5	Berglunds snab	Berguvsvдgen	Christina Berglu	0921-12 34 65
6	Blauer See Delil	Forsterstr. 57	Hanna Moos	0621-08460
7	Blondesddsl рин	24, place Klйbe	Frйdйrique Cite	88.60.15.31
8	Bylido Comidas	C/ Araquil, 67	Martнn Sommer	(91) 555 22 82
9	Bon app'	12, rue des Bou	Laurence Lebih	91.24.45.40
10	Bottom-Dollar M	23 Tsawassen E	Elizabeth Lincol	(604) 555-472
11	B's Beverages	Fauntleroy Circ	Victoria Ashwor	(171) 555-121
12	Cactus Comidas	Cerrito 333	Patricio Simpsor	(1) 135-5555
13	Centro comerci	Sierras de Gran	Francisco Chan	(5) 555-3392
14	Chop-suey Chir	Hauptstr. 29	Yang Wang	0452-076545
15	Comйrcio Mineir	Av. dos Lusнad	Pedro Afonso	(11) 555-7647
16	Consolidated H	Berkeley Garde	Elizabeth Browr	(171) 555-228
17	Drachenblut De	Walserweg 21	Sven Ottlieb	0241-039123

Structure

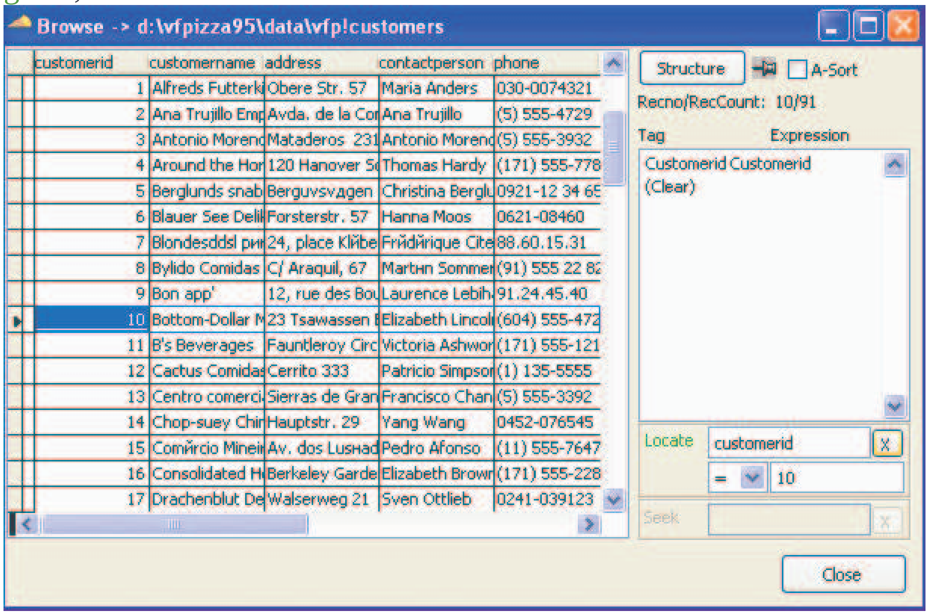
Recno/RecCount: 1/91

Tag	Expression
Customerid	Customerid
(Clear)	

The contents of the currently selected alias are displayed in the grid on the left part of the form.
On the right part of the form are placed additional controls:
Structure – this button displays the structure of the alias (equal to the MODIFY STRUCTURE command)
Always on top – toggles the form's AlwaysOnTop property on/off
A-Sort – When this checkbox is marked, the columns sequence in the grid is alphabetically
RecNo/RecCount – Displays current record number/total number of records
Tags list – Here are listed all defined tags for this alias. Double clicking an item in the list sets record order to the selected tag.
Locate / Seek – It is possible to locate or seek a record matching a certain expression

Locate / Seek

When the LOCATE or SEEK command finds a matching record, the *Locate* or *Seek* label turns green, otherwise it turns red.


Browse -> d:\vfpizza95\data\vfp!customers

customerid	customername	address	contactperson	phone
1	Alfreds Futterki	Obere Str. 57	Maria Anders	030-0074321
2	Ana Trujillo Emp	Avda. de la Cor	Ana Trujillo	(5) 555-4729
3	Antonio Morenc	Mataderos 231	Antonio Morenc	(5) 555-3932
4	Around the Hor	120 Hanover Sc	Thomas Hardy	(171) 555-778
5	Berglunds snab	Berguvsvдgen	Christina Berglu	0921-12 34 65
6	Blauer See Delil	Forsterstr. 57	Hanna Moos	0621-08460
7	Blondesddsl рин	24, place Klйbe	Frйdйrique Cite	88.60.15.31
8	Bylido Comidas	C/ Araquil, 67	Martнn Sommer	(91) 555 22 82
9	Bon app'	12, rue des Bou	Laurence Lebih	91.24.45.40
10	Bottom-Dollar M	23 Tsawassen E	Elizabeth Lincol	(604) 555-472
11	B's Beverages	Fauntleroy Circ	Victoria Ashwor	(171) 555-121
12	Cactus Comidas	Cerrito 333	Patricio Simpsor	(1) 135-5555
13	Centro comerci	Sierras de Gran	Francisco Chan	(5) 555-3392
14	Chop-suey Chir	Hauptstr. 29	Yang Wang	0452-076545
15	Comйrcio Mineir	Av. dos Lusнad	Pedro Afonso	(11) 555-7647
16	Consolidated H	Berkeley Garde	Elizabeth Browr	(171) 555-228
17	Drachenblut De	Walserweg 21	Sven Ottlieb	0241-039123

Structure

Recno/RecCount: 10/91

Tag	Expression
Customerid	Customerid
(Clear)	

Locate: customerid = 10

In order to use the *Seek* section, for selected alias should be set an active index tag. If there is no active order, the *Seek* section is disabled. Current tag can be changed by double click on a tag in the Tag List. The current tag is displayed under the grid.

The class cGridMover

The class *cGridMover* works similar to *cMover* class. The difference is that *cGridMover* class contains two grids instead of listboxes. Thus it is possible to have the base functionality of VFX grids like sorting and incremental search in the mover dialog. The grid at the left side of the form contains all elements available for selection. The right grid contains a list of selected elements. The user can select and deselect any number of elements (rows) using the selection arrow buttons.
The fields in source and destination aliases shown in two grids must have the same field names. Additionally for control to work, it is required to have a field, used internally, to mark fields in source alias, which are selected and should not be displayed in left grid in the control. The name of this field is kept in the property *cControlFieldName*. The field must be of Numeric or Logical data type and is used by *cGridMover* control as a control field.

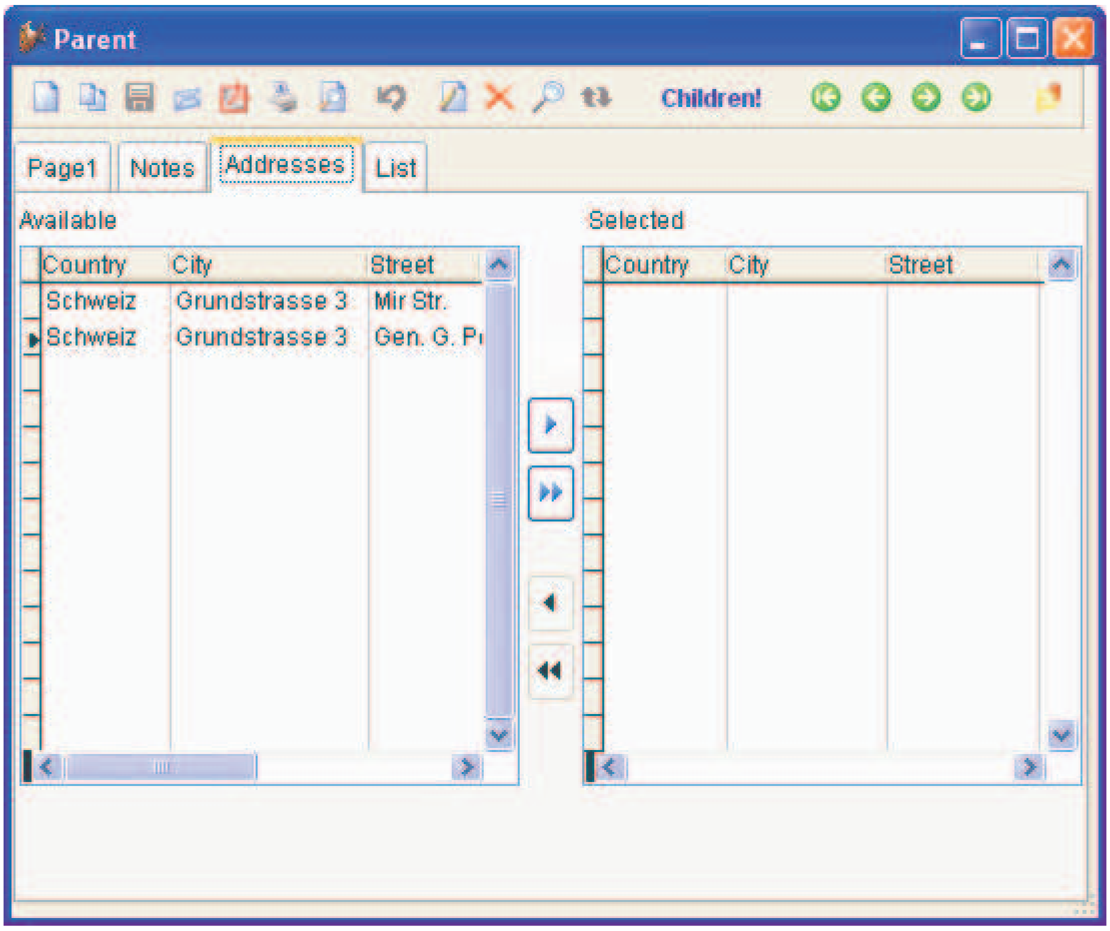
Properties

cSourceAlias	Name of the alias, used as RecordSource for source grid;
nControlFieldNameType	Internally used. Holds the type of Control field: 1 - Logical, 2 - Numeric.

When placed on a form, it is also necessary to be keyed up the record source properties of the two grids and the control source properties for columns.

Methods

onPostInsert	Called when a record is moved from source to destination list;
onPostSave	Called after save operation of the form;
onPreSave	Called before save operation of the form;
onUndo	Called by onUndo method of the form;
RefreshMoverButtons	Refreshes mover buttons;
RefreshSourceList	Refreshes source list according to destination list when loading source list. Called by onRecordMove method of the data form;
LangSetUp	Sets language depending text. Called by Langsetup method of the form.



Country	City	Street
Schweiz	Grundstrasse 3	Mir Str.
Schweiz	Grundstrasse 3	Gen. G. P

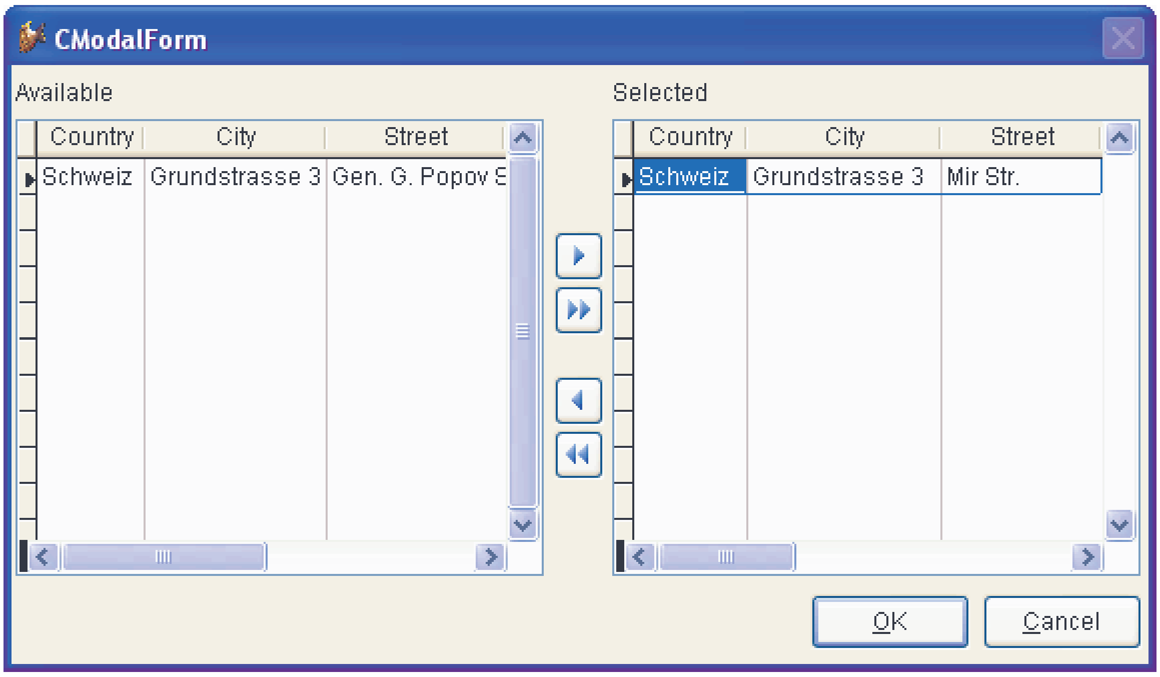
Country	City	Street

When selecting records, the record from *Source alias* (obtained from record source of grid *Available*) is scattered and data is gathered to *Destination alias* (obtained from record source of grid *Selected*). In this way it is also possible to copy also fields which have identical names but are not shown in grids. When a record is moved to the *Selected* grid, it is no more visible in *Available* grid.

When deselecting records, they are deleted from *Destination alias* and displayed again in *Available* grid.

Here comes example code which can be added in DblClick event of textboxes, after you add grid columns in grids at design time, to allow moving the records:
For *DblClick* in textboxes in *SourceGrid:*
```
    This.Parent.Parent.Parent.cmdAdd.Click()
```
For *DblClick* in textboxes in *DestinationGrid:*
```
    This.Parent.Parent.Parent.cmdRemove.Click()
```

The class cGridMoverDialog



Country	City	Street
Schweiz	Grundstrasse 3	Gen. G. Popov S

Country	City	Street
Schweiz	Grundstrasse 3	Mir Str.

The class *cGridMoverDialog* is an efficient dialog form, based on *cModalForm Class* which contains an instance of *cGridMover* class. This dialog provides the main functionality of *cGridMover* control.

When using *cGridMoverDialog* class, it is expected to have a Grid where the selected records are displayed. This grid is passed as parameter to Grid Mover Dialog control, in order to be refreshed when user confirms selected records.

Parameters

tcSourceAlias	Used to fill record source of Available grid in *cGridMover* control;
tcDestinationAlias	Alias to be filled with the selected data;
tcControlField	Used to fill *cControlFieldName* property of *cGridMover* control;
toGridDestination	Reference to Parent form grid. Kept in *oGridDestination* property;
tcCommaSeparatedFieldList	Comma separated list of field names. Used to fill column control sources of *Available* and *Selected* grids;
tcCommaSeparatedHeaderList	Comma separated list of strings used to set the header captions of *Available* and *Selected* grids.
tcCommaSeparatedColumnWidth	Comma separated list of numeric values used to set the column widths of *Available* and *Selected* grids.

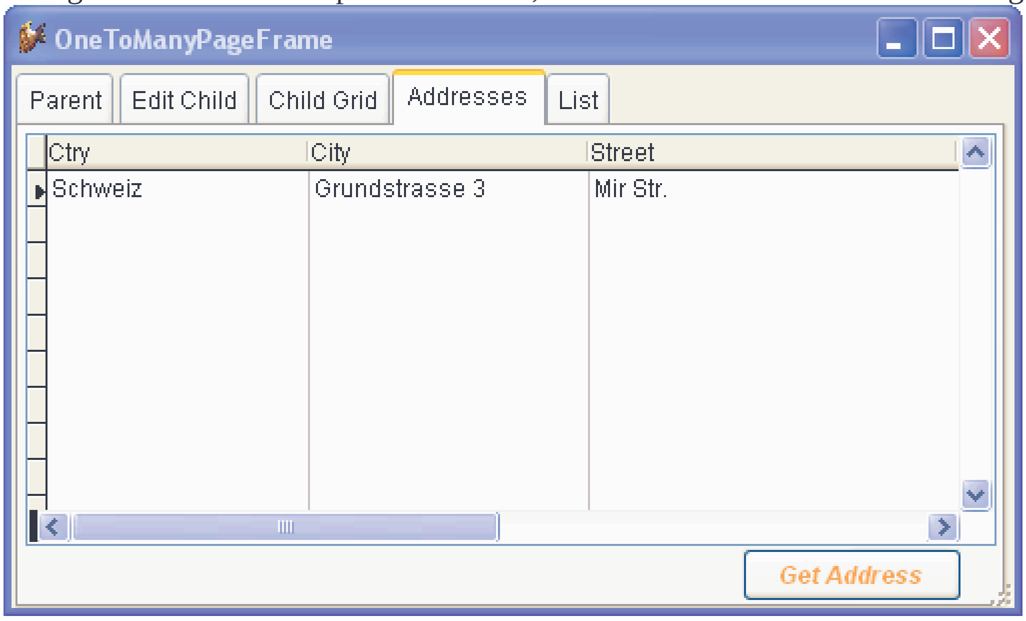
Properties

oGridDestination – Keeps a reference to Parent form grid, to be refreshed.

Methods

SetDestinationData – Refreshes destination alias, passed in *tcDestinationAlias* parameter with selected data.

Here is an example how to use the cGridMoverDialog class in practice:
Let's assume we have a one-to-many data form and it is required child alias to be filled using mover dialog. On Parent form is placed a button, which invokes the Grid mover dialog.

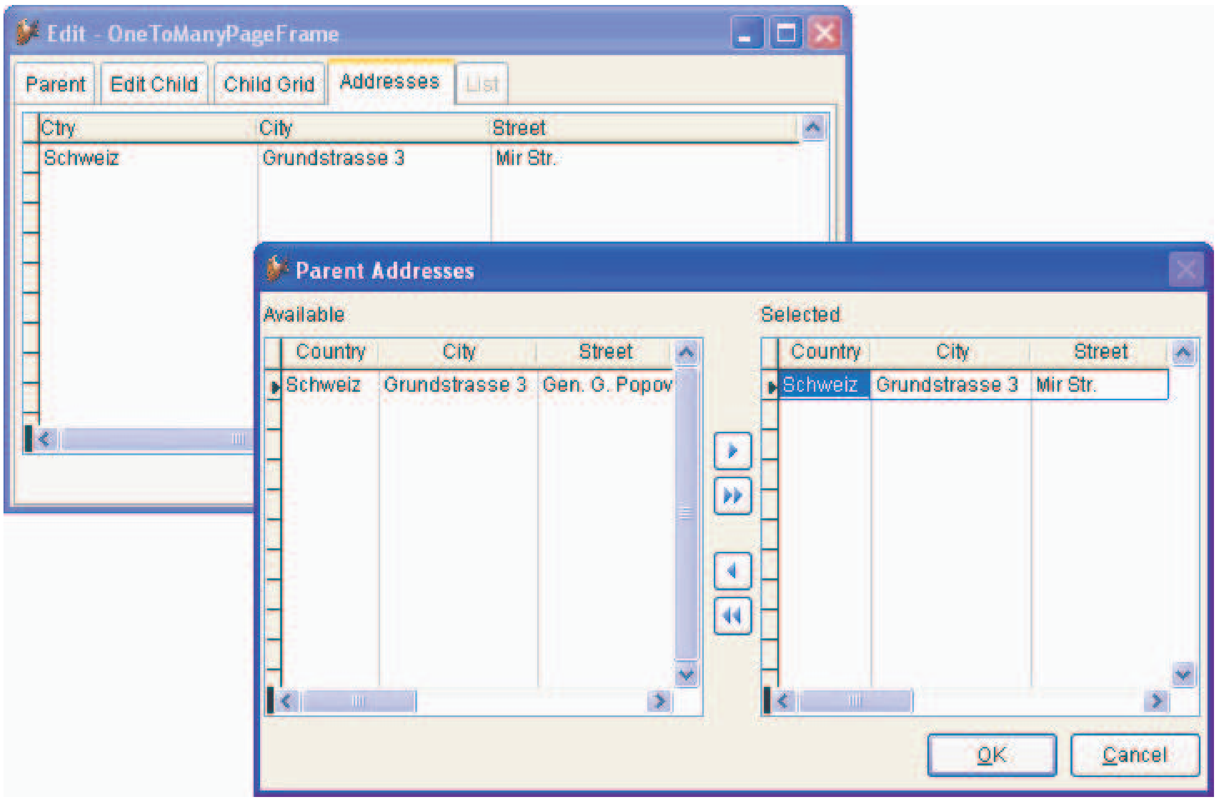


Ctry	City	Street
Schweiz	Grundstrasse 3	Mir Str.

Here is an example code in Click method of the *Get Address* button:

```
Local loGridMover
loGridMover =  CREATEOBJECT("cGridMoverDialog","caAddress", "caParentAddress",
"Selected", ThisForm.pgfPageFrame.Page4.Cchildgrid1, "ctry,city,street",
"Country,City,Street", "100,120,140")
loGridMover.Caption = "Parent Addresses"
loGridMover.Show()
```

After creating the object *loGridMover* developer has the complete control over it. It is also possible to change all desired properties and to call any method.
The Show method displays the modal dialog and the execution of code in click method of *Get Address* button will continue after the user closes mover dialog form.



Ctry	City	Street
Schweiz	Grundstrasse 3	Mir Str.

Available

Country	City	Street
Schweiz	Grundstrasse 3	Gen. G. Popov

Selected

Country	City	Street
Schweiz	Grundstrasse 3	Mir Str.

When the form opens, the *Selected* grid is filled whit the same data records as Parent form grid and the *Available* grid is filled with all records from *Source alias*, hiding the selected ones.
In the *Grid Mover dialog* the user can select and deselect the records as necessary. When the user clicks OK button in the dialog, the new selected records are written back into Parent form alias, specified in *tcDestinationAlias* parameter. The changes may be canceled using *Cancel* button. When saving records to *tcDestinationAlias*, the *onPostInsert* method of Parent form grid called, for each new selected record.

Small enhancements

1. When working with CAs a ChildGrid’s *OnPostInsert*() method gets filled only in the cases when the child alias is of CA and it’s *ForeignKeyName* and *ForeignKeyValue* properties are not empty.
2. At database update the tables that are included in the exe (project) and are located in the DATA folder are skipped
3. SQL Database update: when adding new columns to tables if DEFAULT is empty they always allow NULLs even if the update information for this column says that it should be NOT NULL
4. Database update runs only when application is run as EXE (VERSION(2) <> 2)
5. The column widths of cComboPickList are automatically adjusted only if its lAutoAdjustColumnWidths property is set to .T.
6. The textbox for the font style in grid report dialog is localized
7. goProgram.cCompanyName and goProgram.cAppName are used to create a folder in Documents and Settings system folder (AllUsers\Company Name\Application Name or CurrentUser\Company Name\Application Name) to store vfxacomp.dbf and vfxpath.dbf tables and application’s vfx.ini file. If those properties are empty the files are stored in the folder where the application’s exe resides.
8. VFX Application Builder executes ALLTRIM() on all char properties of goProgram when saving