

What is new in VFX 9.0

In this document are described new features in VFX 9.0

New features for developer

goProgram object

New properties

cConfigPassword – password used for encryption of config.vfx file. This password is used for security purposes, so the information about server name, user name and password being kept safety.

cFaxPrinterName – The name of Fax printer driver to be used when sending fax reports.

cMetadataTableName – Name of table where metadata definitions are kept.

lAllowMultipleLogin – Set this property to .T. if you want to allow same user to log in simultaneously from different workstation.

lAllowUserCustomization – Set this property to .T. if you wish to allow end-users to set their own environment preferences.

lInformUserForUpdate – Set this property to .T. if you want user to see a message when database update is started

lSaveExportPathPerUser – Set this property to .T. if you want export folder chosen by every user to be kept in the resource file per user.

lShowProgressOnUpdate – Set this property to .T. if you want user to see a progress indicator form when database update run.

lUseBCCRecipients – Set this property to .T. if you want to allow BCC input textbox in E-Mail details form.

Note: BCC option is only available when the e-mail client is Outlook Express.

nDockable – Defines if forms can be docked. -1 - Use form settings, 0 - All forms do not support docking. , 1 - All forms support docking and are dockable., 2 - All forms support docking but are not dockable.

Note: *nDockable* property is not considered for modal forms.

nHighlightStyle – This property is used to set globally *HighlightStyle* style for *cGrid* class.

nIndexInsteadOfFilter – Defines if filtered index will be used instead of filtering 0 - Use form setting, 1- Force to .t., 2 - Force to .f.

nNullValid – Defines if Null is valid value for *cPick* classes. 0 - Use Control Settings, 1 - Null is valid value, 2 - Null is not valid value

nSearchOnInit – This property is used in conjunction with *lSearchOnInit* property of *cDataForm* class. It specifies whether a form should come up with the last search or no.

New methods

OnPreStart – Designated to place additional code that needs to be executed before *Start* method of the object

OnPostStart – Designated to place additional code that needs to be executed after *Start* method of the object

Data Handling

cBaseDataAccess class

VFX 9.0 provides developers with new way to access data of different data sources. The new `cBaseDataAccess` class is based on `CursorAdapter VFP` class, providing VFX developers with extended data access functionality. The class `cBaseDataAccess` uses common application settings and allows you to develop applications which can be easily ported to different data sources. Data source is set in `Config.vfx` file

Properties:

cConnMgrName – Name of the `goProgram` object property used generally to keep settings and maintaining used connection. This object handles connection to data and provides connection handle to be used by `cBaseDataAccess` class.

cExecuteAfterCursorFill – command to be executed after `CursorFill` of the `CursorAdapter` object is executed. You can place here any data processing, needed to be applied on the retrieved data.

Filter – Logical expression to be applied on the cursor after retrieving data.

Order – Index tag to be used for ordering after retrieving data. This must be a name of a tag which is already created using `CreateIndexes` method.

Methods:

CreateIndexes – This method is mainly used by VFX DE Builder. Here are placed commands to create necessary indexes after `CursorFill` method is executed.

Accessing data using CursorAdapter

VFX 9.0 builders support now usage of `CursorAdapter` classes in form's data environment. You can use `CursorAdapter` classes in the data environment in same way as you use local and remote views.

VFX 9.0 provides a base `CursorAdapter` class, which implements base functionality that is needed to access application's data. This is `cBaseDataAccess` class in `Vfxctrl.vcx`. The class ensures that the entire application uses one common data handler and no redundant connections are created.

Properties:

cExecuteAfterCursorFill – This property is aimed to contain code which needs to be executed after `CursorFill` method is executed. You can place here code for any necessary data processing as well as code for creating indexes. This functionality is realized using property to make possible assigning such code when the object is created at run time and you do not have an easy way to write code in its methods.

Filter – filter expression to be applied on the retrieved data in the cursor

Order – name of index tag to be used.

Methods:

CreateIndexes – this method is called immediately after `CursorFill` method and is designated to build needed indexes. It is filled by VFX Dataenvironment Builder.

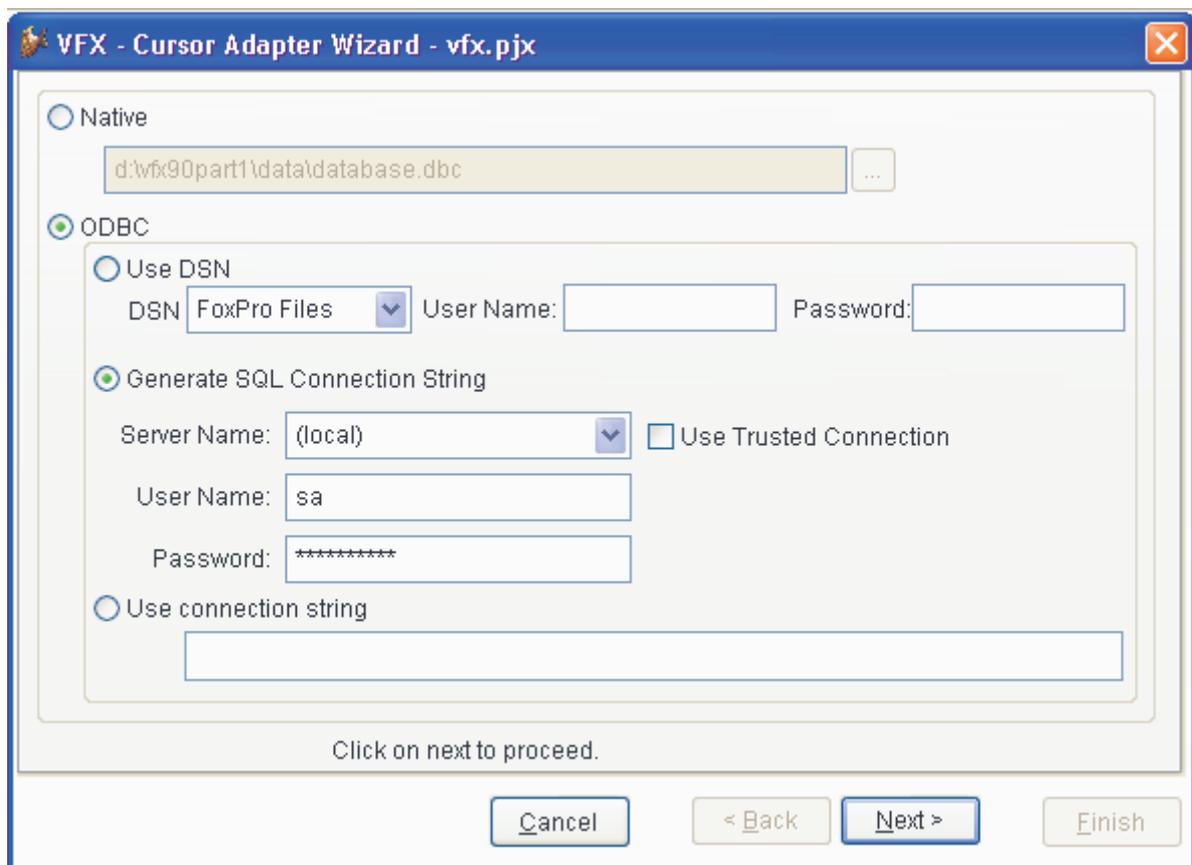
When an instance of the class `cBaseDataAccess` is created its properties are set accordingly correspondent `goProgram` properties. `goProgram.cDataSourceType` is used as `DataSourceType` for the newly created object. If `NATIVE DataSourceType` is used, the class uses the database referred by `goProgram.cDataDir` and `goProgram.cMainDataBase`

properties. When DataSourceType is other than NATIVE, to obtain DataSource value is called *GetConnection* method of the connection manager object, specified in cConnMgrName property.

For developers' use a subclass of cBaseDataAccess is placed in Appl.vcx. This is the place where you should define settings that are specific for your application.

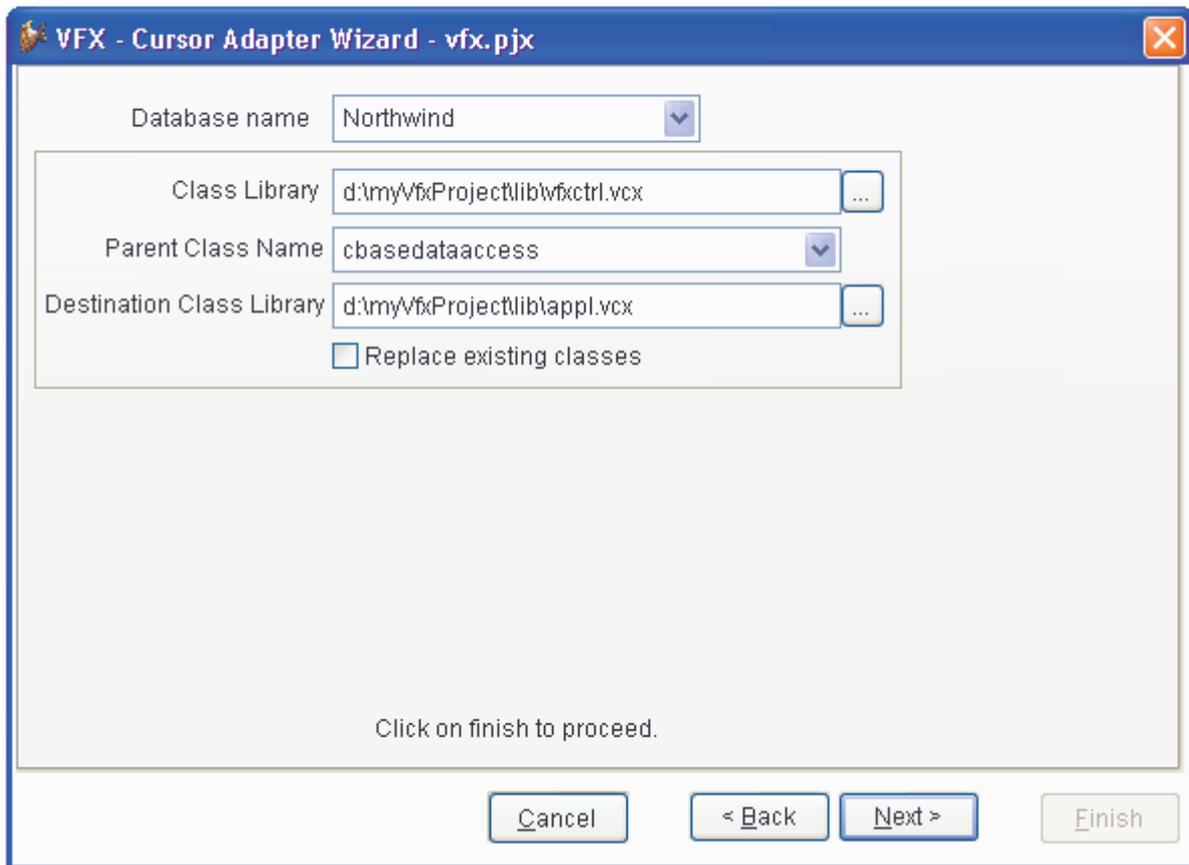
To help you handling the application data through CursorAdapter classes, VFX 9.0 includes a new CursorAdapter Wizard. CursorAdapter Wizard creates one CursorAdapter class for every table that is included in the database. By default these classes are based on cAppDataAccess class and are placed in appl.vcx. However, in the Wizard you can change underlying class and target class library. Wizard leads you through 3 steps:

At first step of the wizard you choose the data source. This is connection to the database that will be used to generate CursorAdapter objects.



This data source will not become data source for the compiled application. It is used for building CursorAdapter classes only. When application runs, the actual DataSource will be read from config.vfx file. In this way you can set different data source for the particular end-users. This approach allows you also to set more than one client in same way as you made that using vfxpath.dbf. At run-time when more than one client is set, the client selection dialog is invoked. How to fill the content of config.vfx file will be described later in this document.

If you select the option Generate SQL Connection String, on second step you must also choose database from the list of databases which reside on the selected SQL Server.



On this step underlying CursorAdapter class and destination library are to be chosen.

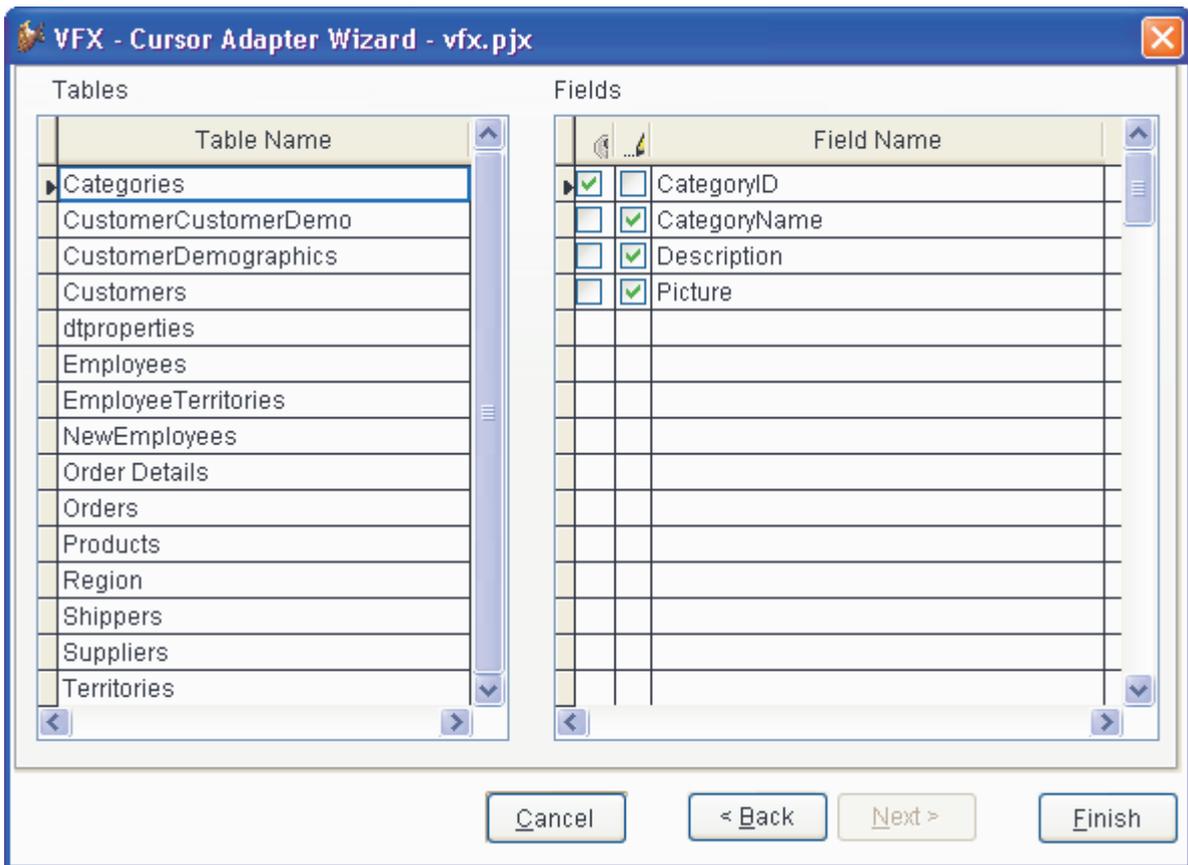
Default values are:

cAppDataAccess class in Appl.vfx (for underlying class)

Appl.vcx (for destination class library)

You can also set whether classes that already exist in destination class library should be overwritten by checking Replace existing classes.

Last step for building CursorAdapter classes shows a list of all tables and fields for every table. When you navigate through the table list in the left grid in the dialog, in the right grid is displayed the list of fields for correspondent table.



By default fields which are primary key for the table are marked as key fields for the created CursorAdapter class. All other fields by default are marked as updateable fields. You can change these preset values accordingly.

As a result the Wizard creates one CursorAdapter class, corresponding to every table in the chosen database. For generated classes are filled following properties: CursorSchema, Tables SelectCmd, KeyFieldList, UpdatableFieldList, and UpdateNameList.

Managing the content of Config.vfx file

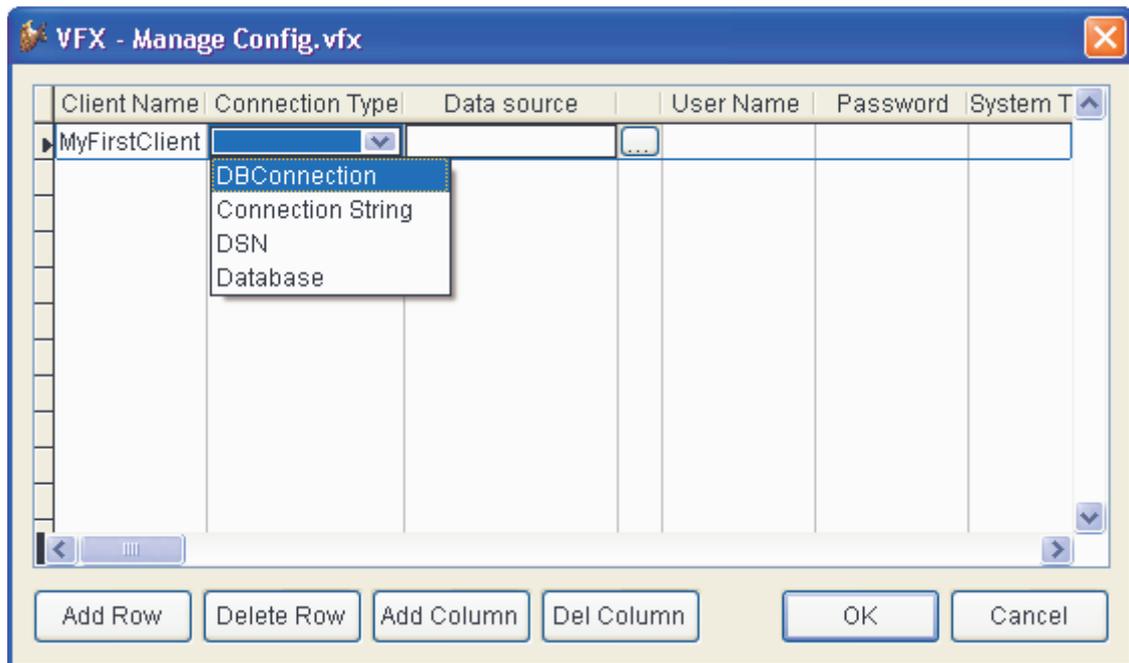
When you develop the application, the DataSource that you choose in the CursorAdapter Builder refers resources, local for your developer's computer. It is not always possible the users' computers to use resources named in same way. For instance, the name of SQL Server instance on customer's network is always different than name of your local SQL Server running on your computer and used in development.

This is the reason at runtime not to use DataSource which were set at design time. At run time cBaseDataAccess class uses the object goProgram.oConnMgr - an instance of class cConnectionMgr. Settings for goProgram.oConnMgr object in turn, are read at run time from the file Config.vfx

For the end-user you need to create and provide Config.vfx file. This file keeps data that are used to connect to the end-user data. There are store data source type, connection string and other information, necessary to establish a connection to remote data. For security reason this information is kept in encrypted format Encryption password is stored in *goProgram.cConfigPassword* and you can change it by your choice.

The file Config.vfx can be created either by the developer and deployed along with the application or created by the user Administrator. When the application starts first time and no

config.vfx file is found, the Manage Config.vfx dialog is invoked. This dialog can also be launched by Tools/Manage Config.vfx menu.



For every particular client you can define different DataSource type and corresponding VFP native database or ODBC connection to the database. ODBC connections can be defined in several ways.

If you choose *DBConnection* as connection type the application will use an existing connection from a VFP database. At run time the connection name is stored in *cDBCConn* property of *goProgram* object. In the Config.vfx file is defined which database should be used for different clients. When application starts the connection information is retrieved from the database set for the chosen current client and this connection is further used to connect to remote data.

To use ODBC connection, you can specify a connection string or an existing DSN.

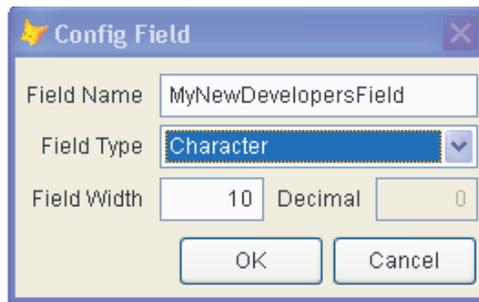
If a *connection string* is chosen for DataSource type the ellipsis button invokes a dialog that helps the user to build a valid ODBC connection string.

When you want to use *DSN*, you can also specify username and password to be used while establishing connection. If you do not provide username and password for the DSN and the connection requires authentication, a login screen will pop up asking user to enter username and password.

When the file Config.vfx is used, all information that was previously kept in Vfxpath.dbf is read from it. So you need to fill also all other columns, corresponding to Vfxpath.dbf fields.

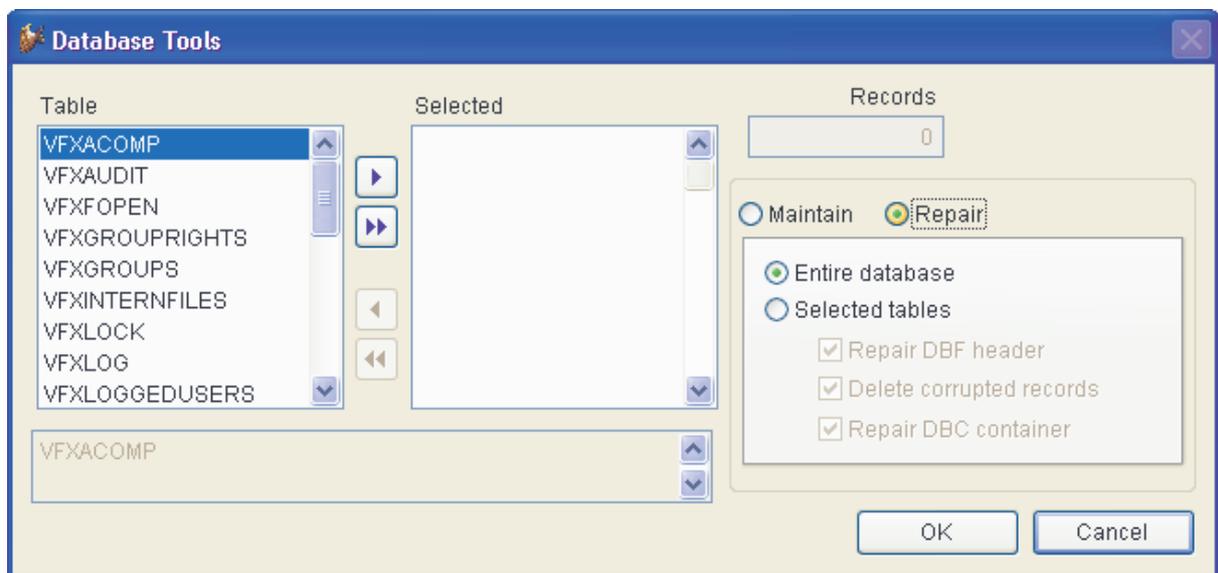
By analogy with Vfxpath.dbf, you can add your own fields to Config.vfx and use these fields later at run time.

The button *Add Column* invokes a dialog window, where you can specify name and type of the field to be added



Database Repair Tool

VFX 9.0 includes new tool for repairing corrupted data. On Database tools dialog form, now is available additional option- Repair



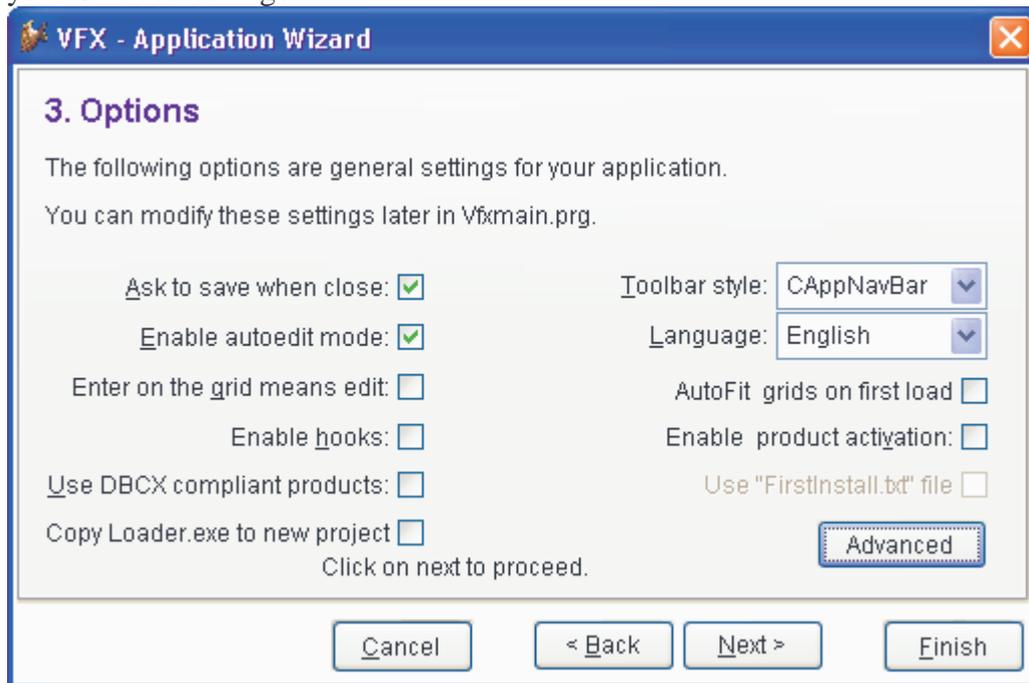
You can choose to repair the entire database on only selected tables. Depending of data corruption you can choose to repair only headers of tables or just to delete corrupted records. For repairing DBC container and DBF headers, an empty database with same structure as the operational one is used. When the executable file is built, the structure of database is included in it as a generated PRG. Later at run-time an empty database is created using this program. The newly created database is used to repair database container and DBF headers of the operational database.

If you chose to delete corrupted records, all records with empty or duplicated primary key values will be deleted from tables.

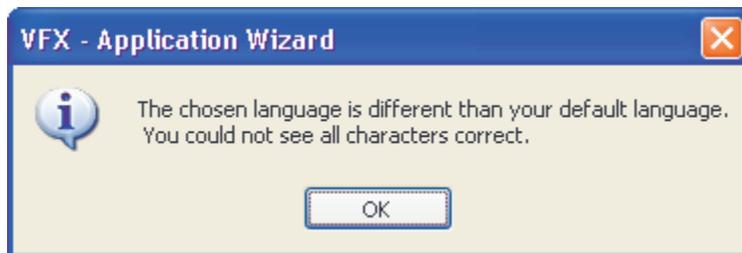
VFX Application Builder

Many of settings that you fill in the Application wizard are saved and later used for your newly created projects. The default path for creating new application is "My Documents\VFX Projects\". Once you select a different path for your project, same folder will be subsequently used for next projects that you create. By default is suggested a subfolder named *VFX APPLICATION* followed by a number, to form a unique folder name.

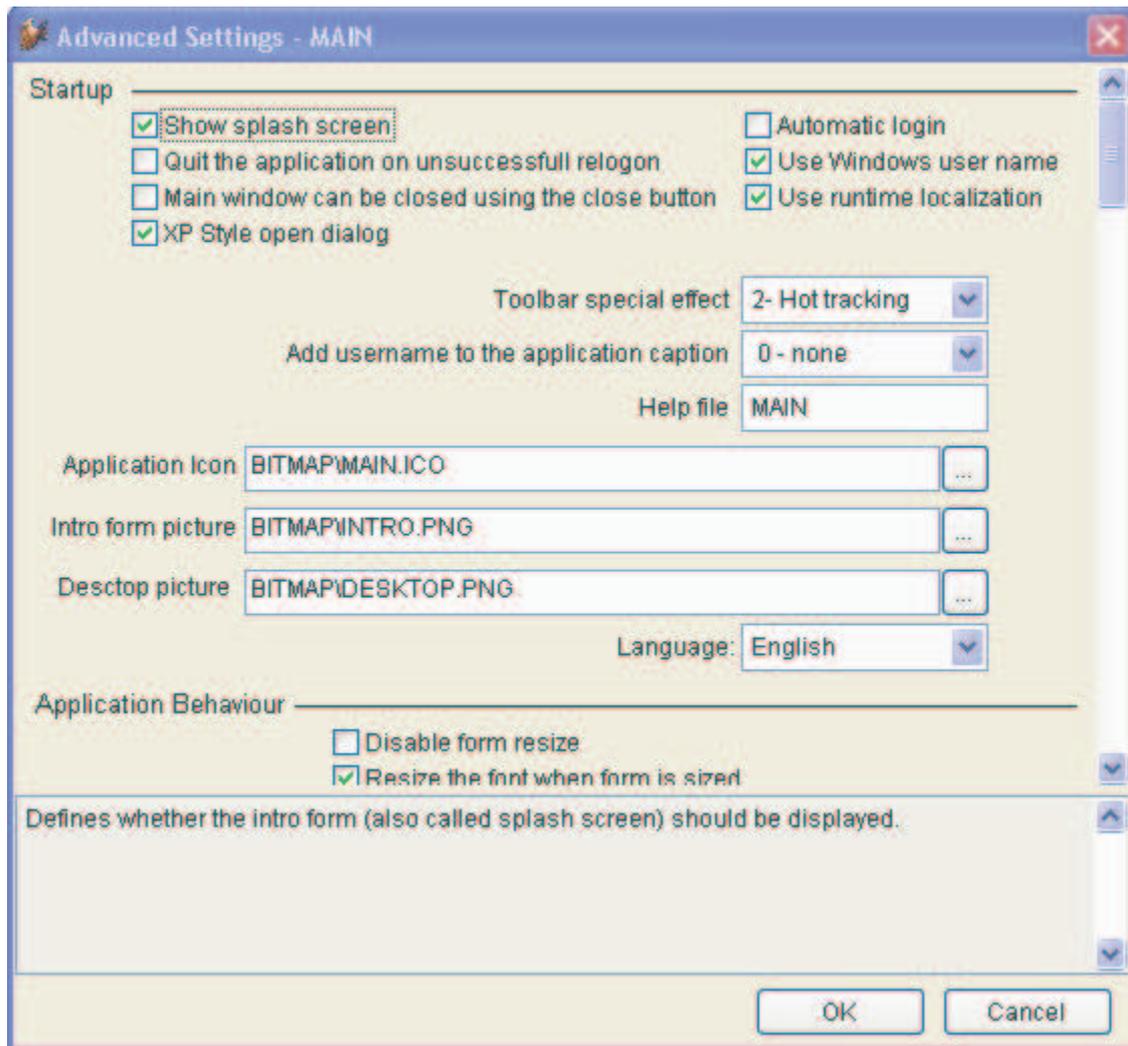
In the Application wizard is included check for compliance of selected project language and your Unicode settings.



If on step 3 of the application wizard, you choose a language which cannot be displayed properly on your computer a warning message is invoked.



On step Options of the Application wizard is also placed a new button *Advanced*. Advanced button invokes a new dialog, where can be set all properties of the goProgram object. In the lower part of the dialog are displayed explanations about every property, which you can set here.



You can also run the Application builder later to set these additional properties. Application builder invokes this dialog and allows you easy to change any property value.

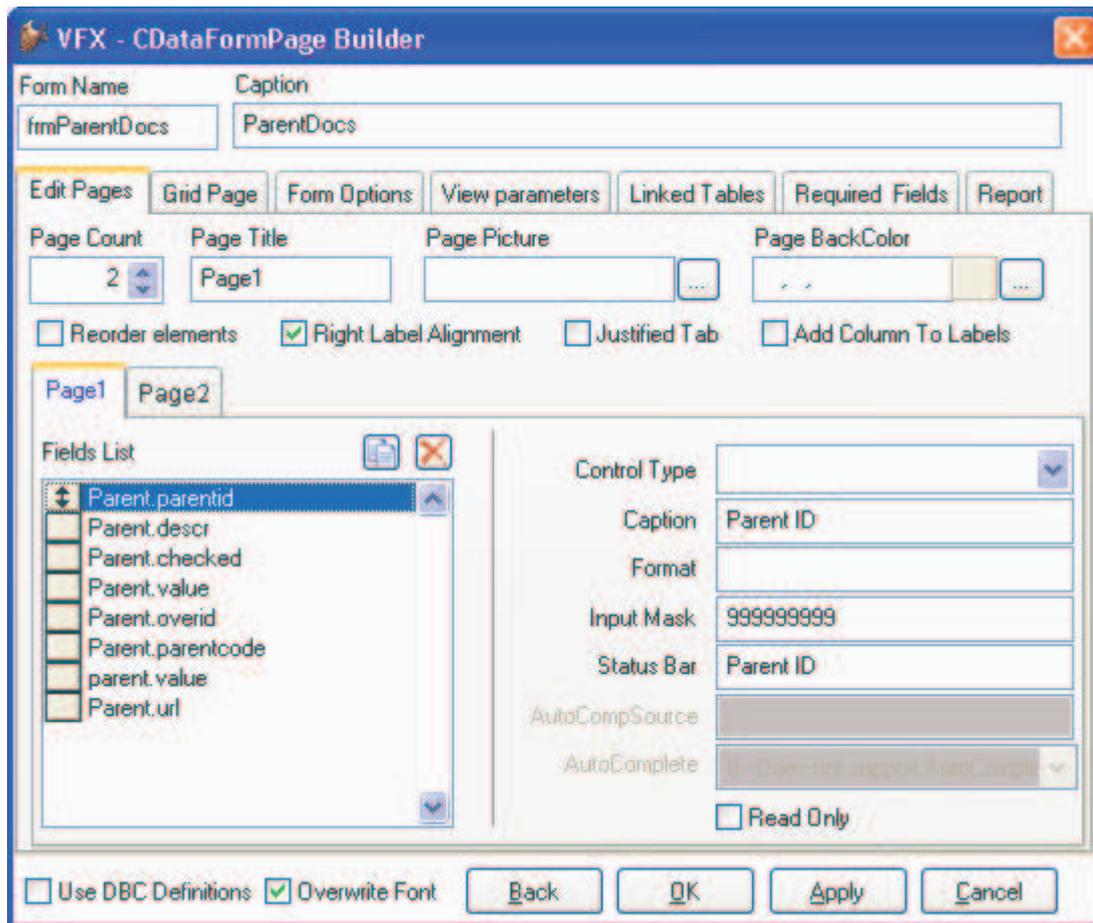
Note: Last saved property values in Application builder will be used next time when you create a new project.

VFX Form Builder

Managing Data Environment in Form Builders

VFX 9.0 form builders now give developers opportunity to set data environment along with building form layout. As first step of the form builder, the Dataenvironment builder is invoked. Dataenvironment builder gives you the possibility to add objects to form's data environment, to set order and filter of data, to define used indexes and to define relationship between objects in data environment.

On *Aliases* page you can add tables, existing CursorAdapter classes as well as creating new CursorAdapter objects.



If you mark the AddColumnToLabels checkbox, the builder will append a column char (:) at the end of all labels.

For all text fields you can specify if Autocomplete will be supported and table to be used to gather autocomplete texts. You don't have to deploy autocomplete table with your application. If autocomplete table does not exist, it is automatically created.

VFX - CDataFormPage Builder

Form Name: frmParentDocs Caption: ParentDocs

Buttons: Edit Pages | **Grid Page** | Form Options | View parameters | Linked Tables | Required Fields | Report

Grid Page Title: List Grid Class: cgrid Grid Page Picture: [Browse] Use Grid Page

Grid Page BackColor: [Color Picker]

Fields Selected:

- parent.parentid
- parent.descr
- parent.date
- parent.checked
- parent.value
- parent.ins_date
- parent.ins_usr
- parent.edt_date
- parent.edt_usr
- parent.overid
- parent.parentcode

Control Settings:

Control Type: [Dropdown]

Header: Parent ID

Control Source: parent.parentid

Output Mask: 999999999

AutoCompSource: [Empty]

AutoComplete: [Dropdown]

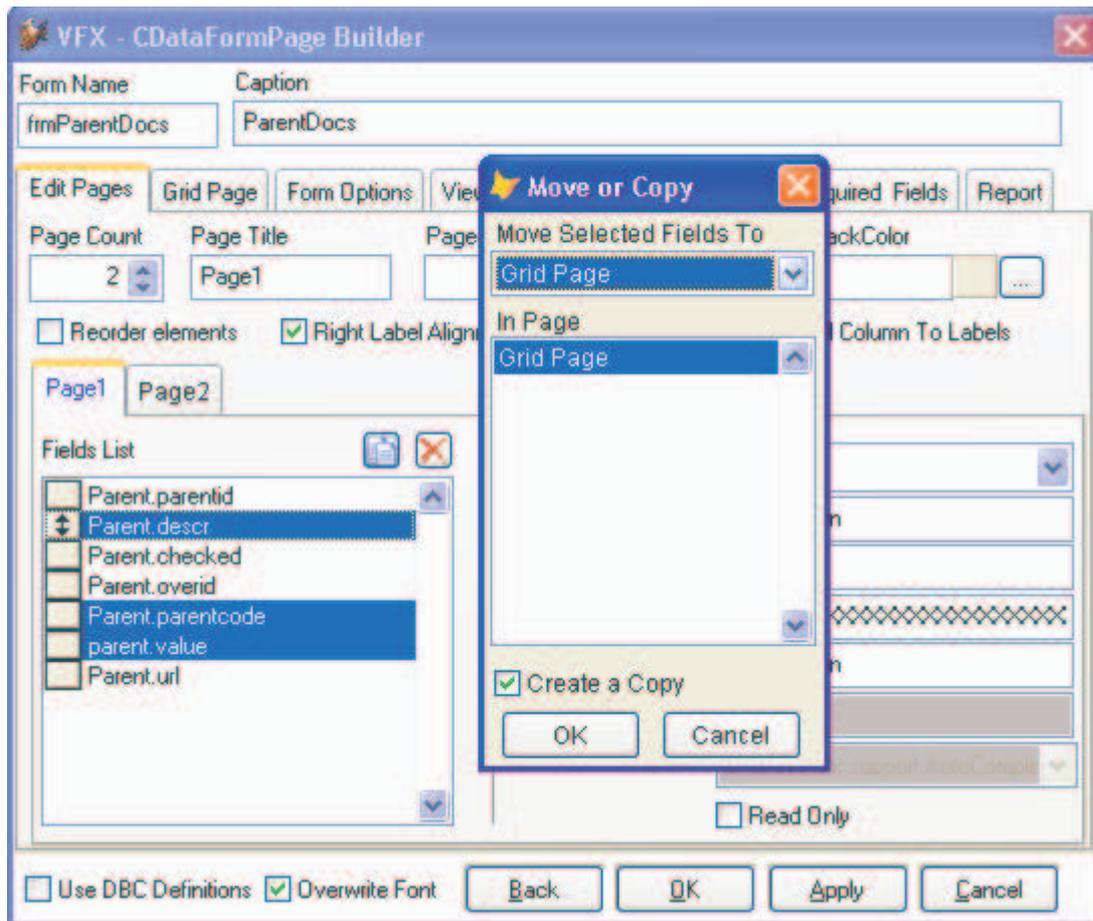
Read Only

Incremental Search

Use DBC Definitions Overwrite Font

Buttons: Back OK Apply Cancel

Very useful, when redesigning existing forms, is the new button  *Move or Copy Fields*. It would help you a lot if a page has to be moved or list was built on the wrong page by accident or even if grid list is similar to edit field list or vice versa. All the settings would not have to be keyed up again which is very important in case of mistakes or maintenance. After selecting necessary fields in *Fields Selected* list and click on the *Move or Copy Fields* button, a new dialog is invoked



Currently selected fields can be either moved or copied to another page in the form builder. From the *Move Selected Fields To* drop-down list you choose type of the destination page. Destination page type can be Edit page, Grid page or Report fields page. After you select type of destination page, the list *In Page* shows all existing pages of chosen type and you can select the needed page where controls will be placed.

When it is necessary not to move selected controls, but just to create a copy of them using all currently set properties, you should check the *Create a Copy* check box. In this case controls remain on their original place and a copy of each control is created on the destination page.

Four new pages are placed in Form Builders to allow you to manage additional features of VFX form classes:

View Parameters page

The screenshot shows the 'View Parameters' dialog box in the VFX - CDataFormPage Builder. At the top, the 'Form Name' is 'frmParentDocs' and the 'Caption' is 'ParentDocs'. Below this are several tabs: 'Edit Pages', 'Grid Page', 'Form Options', 'View parameters' (which is selected and highlighted in yellow), 'Linked Tables', 'Required Fields', and 'Report'. The main area of the dialog is divided into two sections. On the left is a 'Parameter List' with a list box containing 'Parent.overid'. On the right, there are several input fields: 'Parameter Name' (set to 'Parameter'), 'Caption' (set to 'Overid'), 'Format' (empty), 'Input Mask' (set to '999999999'), and 'Status Bar' (empty). A 'Reorder elements' checkbox is checked. At the bottom of the dialog, there are two checkboxes: 'Use DBC Definitions' (unchecked) and 'Overwrite Font' (checked). To the right of these are four buttons: 'Back', 'OK', 'Apply', and 'Cancel'.

In VFX 9.0 it is possible to place controls similar to AskViewArg form, so the end-users can query data using same form. *View Argument* controls are placed at upper part of the form (above the data pageframe). They are always visible and user can change arguments and invoke data reloading. For these controls a parameter name is specified instead of ControlSource property

Linked Tables Page

VFX 9.0 forms allow you to define additional list of tables which are 1:1 related to the main table. This feature gives the developers good support for more complex database design without need to write additional code to maintenance referential integrity. Having described the relational condition, VFX takes care to keep data consistent.

It is not required related tables to have same primary key name as the parent table, however it is good design to keep them same. This key is automatically filled in insert operations after inserting new record in master table. Related records are deleted when delete operation is performed on master table.

The screenshot shows the 'VFX - CDataFormPage Builder' application window. The 'Linked Tables' tab is selected. The 'Form Name' is 'frmParentDocs' and the 'Caption' is 'ParentDocs'. The 'Master Table' is set to 'Parent' and the 'ID Field' is 'Parentid'. The 'Parameter List' is currently empty. The 'Use DBC Definitions' checkbox is unchecked, and the 'Overwrite Font' checkbox is checked. The 'Back', 'OK', 'Apply', and 'Cancel' buttons are visible at the bottom.

On *Linked tables* page you have to select master table and its primary key field. In the Parameter List you can add fields from related table which maintain referential integrity. In the parameter list are listed alias and related field.

You are allowed to choose only one related field per table. When a second field from a table which is already in the list is chosen, it overwrites the field that is already placed in the list.

Required Fields

The new properties *cRequiredFields*, *cRequiredFieldInitProps*, *cRequiredFieldFailureProps* and *cRequiredFieldFailureForm* help you to avoid NULL and empty values in saved data.

This is very good way to show the end users which fields must be filled.

The list of properties and their values entered in *cRequiredFieldInitProps* property is applied at the form's *Init* method. The *ControlSource* of every control in the form is checked against content of the property *cRequiredFields*. If the *ControlSource* of a particular control is included in the list of required fields, this control is changed according *cRequiredFieldInitProps*. This means that to properties of that control are assigned corresponding values as specified in *cRequiredFieldInitProps* string.

The property name and value to be stored in that property must be specified in following format:

PropertyName = *cExpression* [; PropertyName = *cExpression*]

cExpression is an expression resulting in a value appropriate for the corresponding property.

If more than one properties need to be set, separate the list with semicolon character.

For example, if you'd like to show required fields initially with bold font, in red forecolor and light yellow background you need to write following expression in *cRequiredFieldInitProps*:

FontBold = .T.; ForeColor = RGB(255,0,0); BackColor = RGB(255,255,196)

All control which are mandatory to fill, regarding their *ControlSource* and *cRequiredFields* list will be displayed in that way.

When you use this great VFX feature, end users will be able to see what they need to fill before attempt saving data and this will help users a lot to be aware they filled all required information.

On *Valid* event of the form all the required fields are checked for valid input. If a missing value is encountered, for every field which is not filled as required are applied properties keyed up in *cRequiredFieldFailureProps* form's property.

The value of the property *cRequiredFieldFailureProps* is constructed in same way as for the *cRequiredFieldInitProps* property. The only difference is that *cRequiredFieldInitProps* settings are applied when form is instantiated and is used as advance information for end users, but setting defined with *cRequiredFieldFailureProps* string are used when user input is validated, before saving data in database.

All these properties can be set on *Required Fields* page in Form builders.

VFX - CDataFormPage Builder

Form Name: frmParentDocs Caption: ParentDocs

Required Fields List

- Parent.Parentcode
- Parent.Descr

Init Properties: ForeColor=RGB(255,0,0)

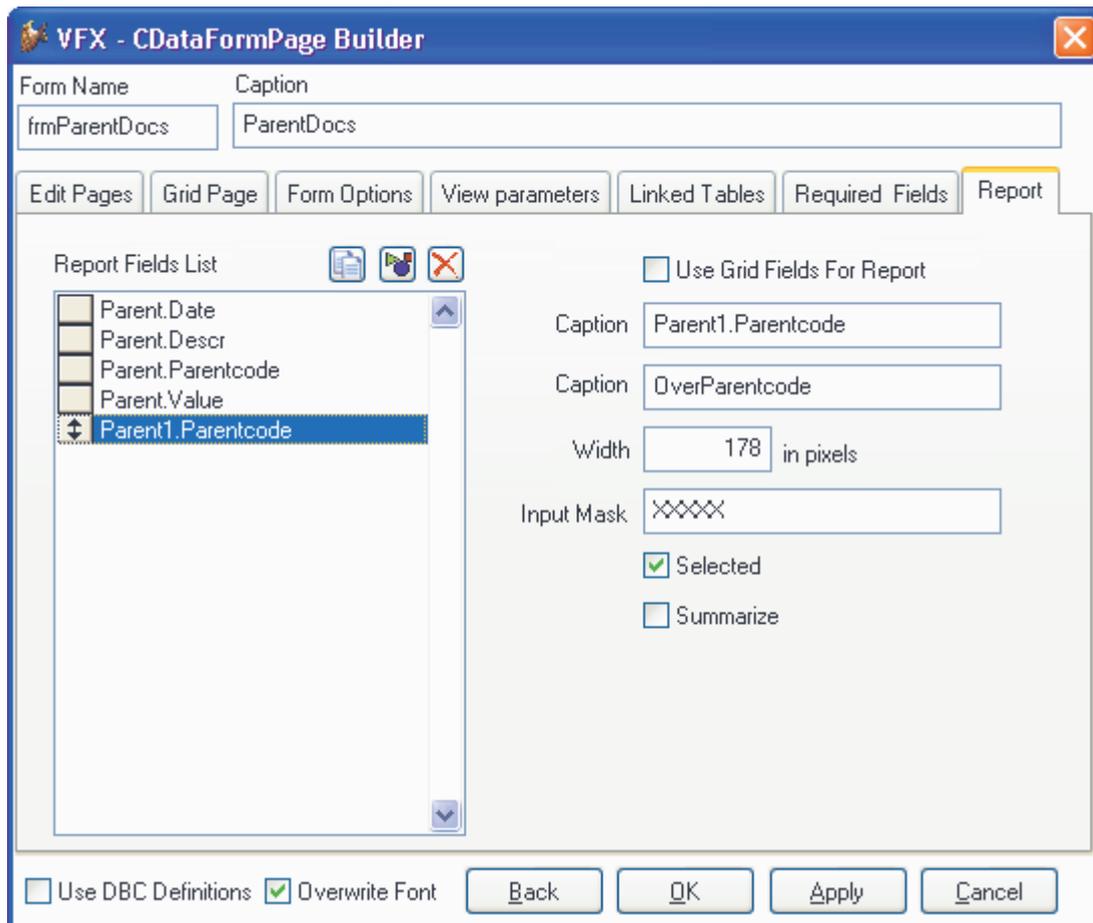
Failure Properties: BackColor=RGB(255,255,0)

Required Fields Form:

Use DBC Definitions
 Overwrite Font

Report

Often it is necessary to include in printed report some fields that are not needed in the grid field list. In other cases you may need to include in grid page fields, that do not need to be printed. *Report* page allows you to select fields that will be listed in on *Advanced* page of Grid report dialog. On *Advanced* page of Grid report dialog user can additionally specify which fields should be included in the report and which fields need to be summarized. You can predefine this selection here in Report fields page.



You need to specify size of the field in the report and Input mask that should be used. On this page you can also set if the field will be selected and summarized by default. When end-user runs grid report, he will see initially selected fields that you marked as *Selected* here. Fields, marked by you as *Summarize* will be initially marked for the end-user. This can help your end-users when they are not quite aware which fields are appropriate to summarize. If you want to keep former VFX behaviour and use in reports same fields, which are included in the search grid, you should just mark the *Use Grid Fields for Report* checkbox. In that case you do not need to define report fields list. When grid report dialog is invoked, the fields list on *Advanced* page will be formed based on search grid in the data form.

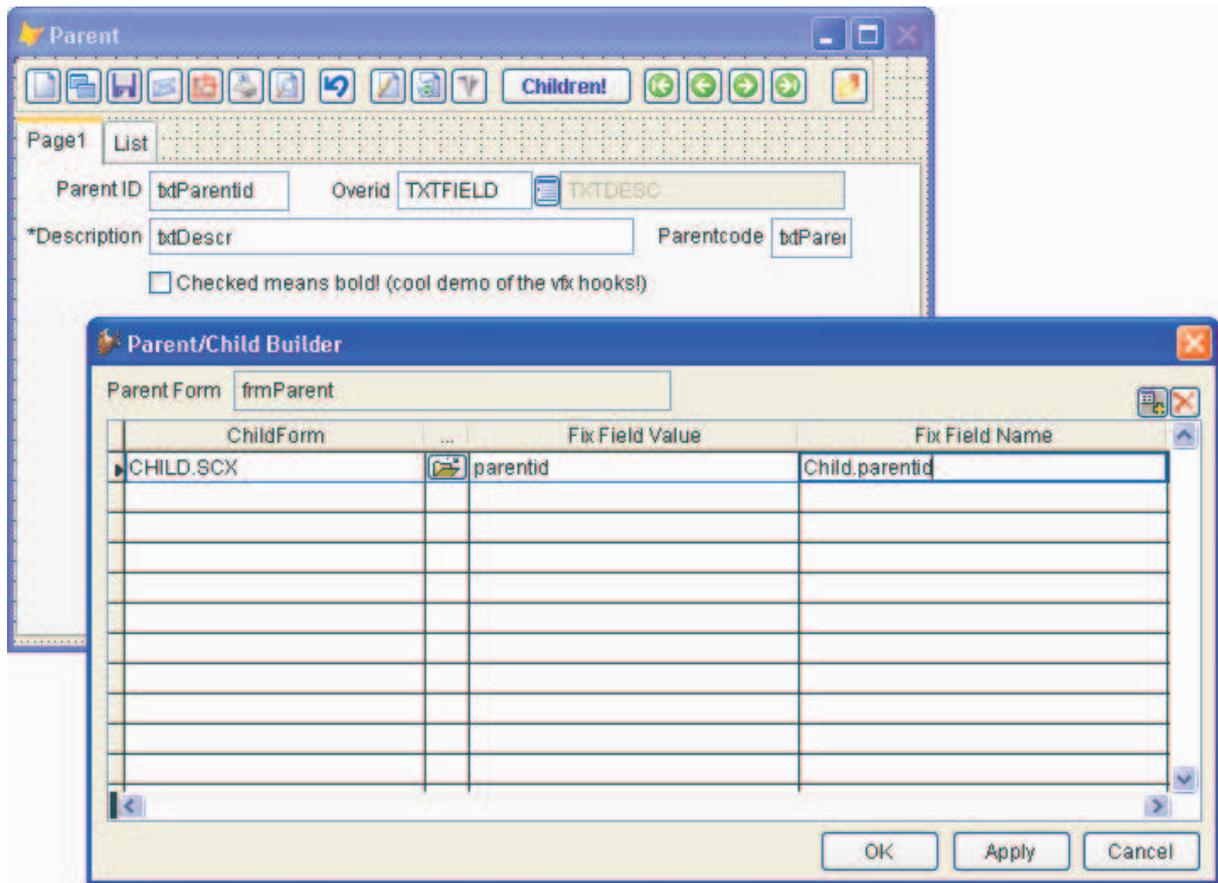
VFX Parent/Child Form Builder

Now, in VFX 9.0, you have a powerful new builder that helps creating code for managing parent-child relation between forms.

For easy handling the relationship code, on *cDataForm* class is placed an instance of the new class *cChildManager*.

To start the VFX Parent/Child Builder, you need to open the parent form. While the parent form is active choose the *Parent/Child Form Builder* from the VFX menu.

In the builder dialog you can add as many forms as needed.



In the Child Form column you enter the name of the child form or you can choose the child form using the Open button. In the column *Parent Field (Fix Field Value)* is entered the name of Id field in the parent alias. The value of this field will be passed to child form when the form is called and when refresh of child forms is performed.

In the column Child Field is entered the name of the field in the child alias which is foreign key to the parent alias. This field will be used for synchronizing the child form and to display only records correspondent to parent record.

After you fill all child form and click OK or Apply button, will be generated the necessary code for OnMore and OnSetChildData methods in the parent form and for OnSetChildData method in the child form.

cComboPicklist class

The class is designated for easy creation and maintenance of pick lists. Using this class you are able to define many pick lists without need to create pick table for every particular list.

The class *cComboPicklist* uses two system VFX tables: *Vfxpdef* and *Vfxplist*.

Vfxpdef table holds definition for pick lists. It contains one row for every defined pick list.

For picklist definition you are able to write a code that will be executed when user makes a selection. This is a common code and executes regardless of specific list item that is selected.

In *Vfxplist* table you can write more specific code that will be executed only if that item is selected.

Pick items are defined in *Vfxplist* table. Field *picklist* is the identifier of the picklist definition and is foreign key to *Vfxpdef* table. The fields *Code* and *Describe* in *Vfxplist* table are values used for creating the list. Depending on picklist definition, only code column or both code and description columns are displayed in the list. Fields *value* and *proccode* are designated for

developer's purposes. In proccode field you can write a specific code that will be executed when that particular item is selected.

For a particular instance of cComboPicklist class you can define if adding new records in the pick table is allowed and the level of user who is allowed to add new values.

Properties:

nParentID – ID key value from Vfxpdef table

Methods:

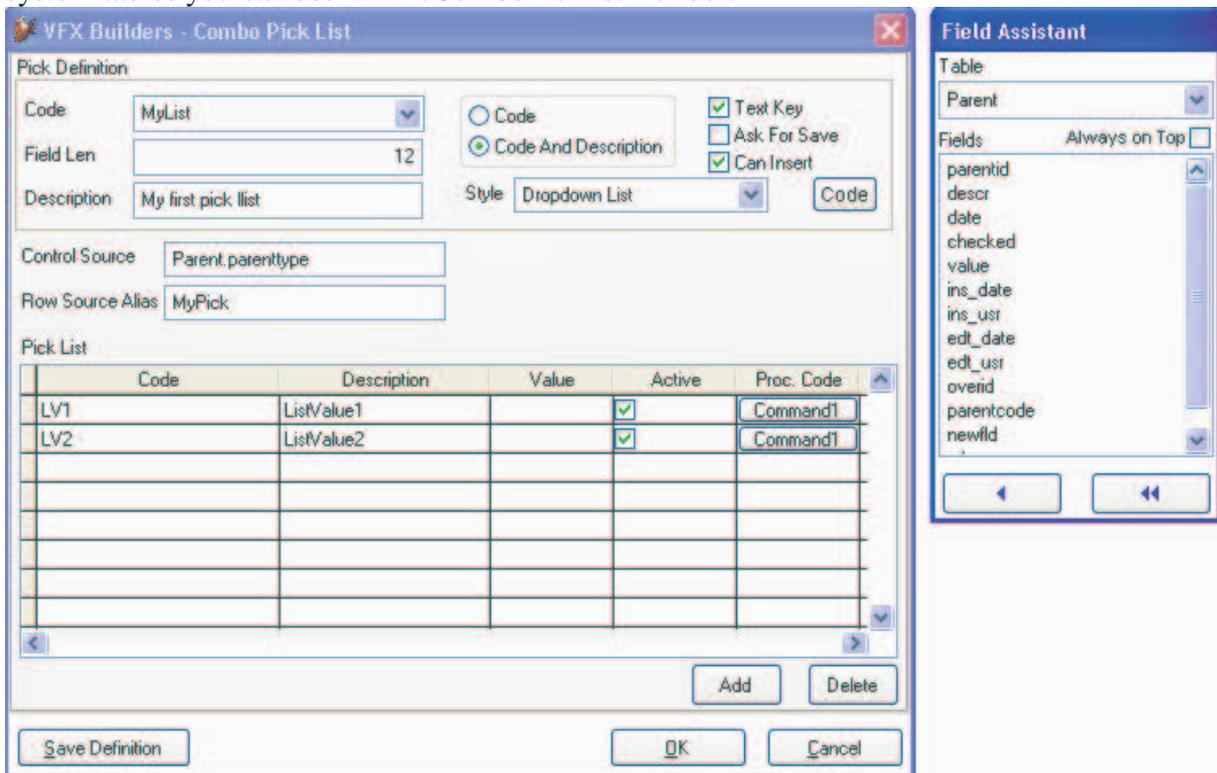
Addnewcode – this method is executed when the end-user adds a new value in the pick list table. If you need to perform additional processing, you should place your code in this method.

For *cComboPicklist* class you can you are able to define two different code snippets. In *Vfxpdef* table in *ProcCode* memo field and in *Vfxplist* table in *ProcCode* memo field. The code in *Vfxpdef* table is common for the cComboPicklist and is executed every time when the selected item of the Combobox is changed. The code in *Vfxplist* table is specific for the particular list item and is executed only when this item is selected from the list.

For list entries in *Vfxplist* table you can define if this is an active entry or not. Using this approach you do not need to delete entries which will not be used in the future and thus to cause orphan records in your database. When the field will not be used anymore, just set the values in its *Active* field to .F.

VFX cComboPicklist Builder

For easily keying up cComboPicklist class and also for maintaining Vfxpdef and Vfxplist system tables you can use VFX cComboPicklist Builder.



For the cComboPicklist you have to choose the control source and the row source alias. If the alias is not in the data environment of the form, the builder creates a cursor object and sets its

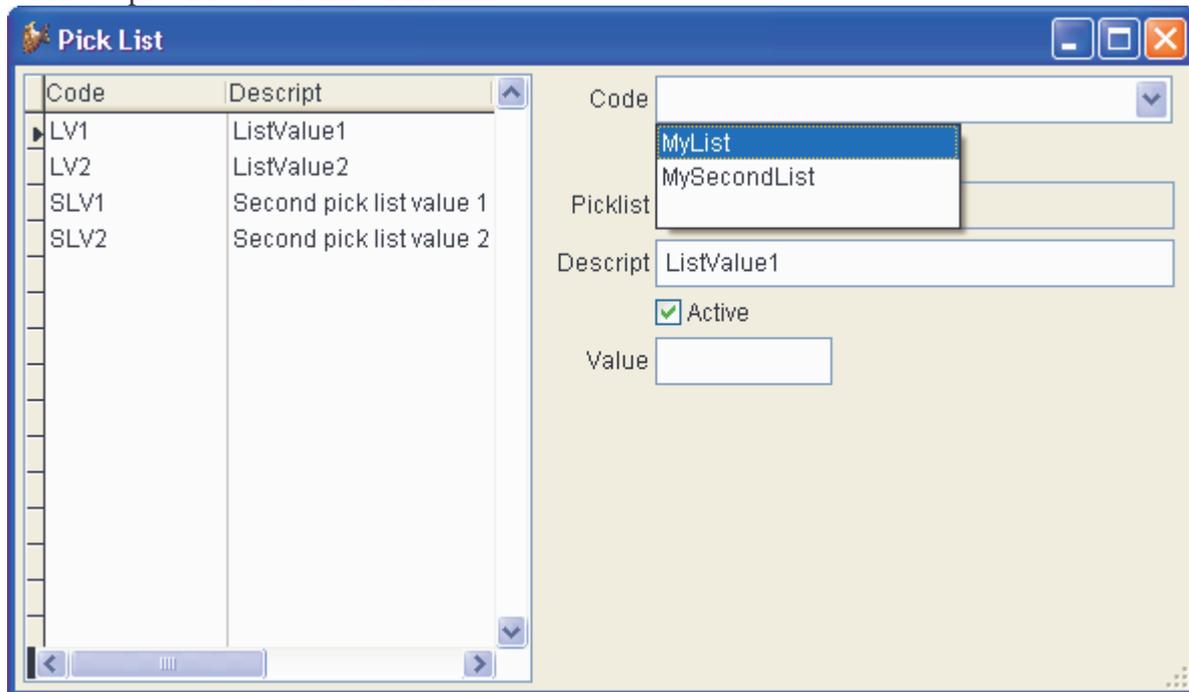
properties accordingly. If there is already such alias in the form's data environment, you are asked if that alias should be used or a new one has to be created.

In cComboPicklist Builder along with setting properties for currently selected object, you can also define Pick definitions, change the existing definitions and maintain the pick list data.

Here you can also write the code to be executed when an item is chosen from the list.

Pick list maintenance form

End users of your application will also need to maintain used pick list values. For this purpose VFX 9.0 provides the form VFXPlist.



User can navigate through the full list or to filter data by list code. It is allowed to delete a records from the table, but the proposed approach it to mark unused rows as not active, in order to avoid orphan records in the database.

One-To-Many enhancements

CChildGrid class, which is implemented in all one-to-many VFX forms, is now expanded with several new functions.

- When the child alias is based on a view or cursoradapter cursor, the cChildGrid class allows being defined incremental search for columns.
- Clicking on empty area of the cChildGrid class adds a new child record.

There are also improvements in VFX one-to-many forms – cOneToMany, cTreeviewOneToMany.

- Buttons for adding and deleting child records are now enabled only if the form is in Edit or in Insert mode.
- The child part now can contain also other controls than cChildGrid which can be placed similar to parent edit page.
- The Edit pages in child part of one-to-many VFX forms can be created using builder in same way as you create edit page in parent part of form.

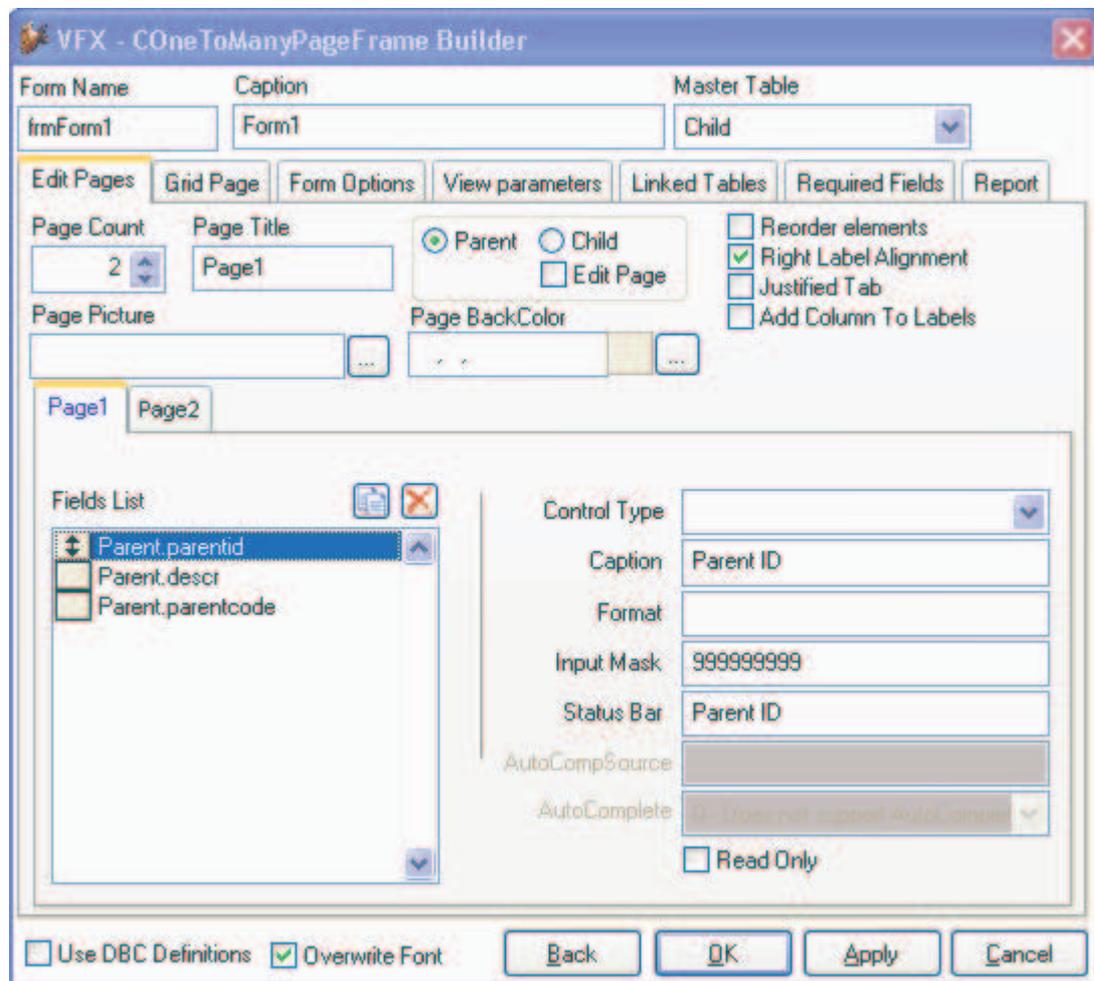
cOneToManyPageframe class

The new class `cOneToManyPageframe` gives the developers ability to place on different pages of one Pageframe both parent and child data. This class combines features of both pageframe controls of `cOneToMany` class.

When active page of the pageframe control is of type Parent, navigation buttons work for the alias `cWorkAlias`. If the active page is of type Child, navigation buttons. On child pages you can place either child grid or edit elements.

cOneToManyPageframe Form Builder

In addition to all other setting which the developer keys up in forms' builders, in `cOneToManyPageframe` form builder it is necessary to define for every page if it will be of Parent type or of Child type. If the page will be of Child type, additionally is selected if it is edit page or grid page.



Error handling

In VFX 9.0 is implemented an extended error handling. In addition to creating an error log, it is now possible for end user to send an error report e-mail to the developer. When in the error message dialog user selects Abort, the error report dialog is invoked informing the user what is included in generated report.

placed in any folder. Radmin works basically with the TCP/IP protocol. Normally, it uses the port 4899.

To initiate the remote control, the customer computer has to establish a connection to Internet. Usually, when dial-up connection is used, a dynamic IP address is assigned to the customer's PC. This IP address is not known to the Supporter. Hence, the VFX application registers the actual IP address of the customer's PC as a subdomain of DynDNS. Thus the Supporter can connect to the customer's PC via the subdomain name in the Internet.

Because to the Supporter this address is not known, the customer's PC registers its IP address on www.dyndns.org as a Sub-Domain.

Conditions

At first, the developer must prepare the VFX application for the Remote control.

First, for the Support of the own application, a Subdomain must be registered with DynDNS.

The keying up of the DynDNS-Registration information can be done using *Edit Vfxsys* option of *VFX 9.0* menu

The *memo field dyndns* contains encrypted string of four rows.

1. User's name for DynDNS
2. Password for DynDNS
3. Subdomain name
4. Password for the Radmin access to the customer's PC

To be able to use the remote control, the customer's PC must have an Internet connection. The IP address must be visible in the Internet. The port used by Radmin 4899 must not be blocked by a Firewall.

An additional software installation for the Remote control is not necessary on the customer's PC.

Registration of a Subdomain

Thanks to the organization Dynamic DNS Network Services it is possible to register Subdomains free of charge. Every developer should register a dynamic DNA with <http://www.dyndns.org/services/dyndns>.

User's name, a password and an E-mail address are necessary for the creation of an account in DynDNS.

The subdomain name can be chosen arbitrarily. It can be selected from a huge number of domain names.

Example: myCompany.dnsalias.com

In this example [myCompany](http://myCompany.dnsalias.com) is the self-chosen subdomain name. [Dnsalias.com](http://dnsalias.com) is the Domain name provided by DynDNS.

One can change the IP address bound with the Domain name arbitrarily often and with different methods. Detailed description of all methods can be found on the website www.dyndns.org <<http://www.dyndns.org>>.

The VFX application calls an URL to register the actual IP address of the customer's PC. The URL has the following format:

<http://username:password@members.dyndns.org/nic/update?hostname=myCompany.dnsalias.com>

If one gives this URL in the Internet Explorer, a HTML page is received as an answer with the word "Good".

When registering the subdomain, an account with user's name and password must be provided. With the login data the account can be configured. The login data are to be started also in the URL above.

Because the Internet browser transmits the own IP address to the server, the IP address must not be declared separately. The Internet server must know which address the answer must be sent back to. Dyndns uses automatically this IP address for the registration of the subdomain.

The Remote control program Radmin

The remote control program Radmin can be downloaded by this website www.radmin.com <http://www.radmin.com>. The documentation is on that website too.

Radmin is a shareware and can be registered at a reasonable price. The full version which is necessary for the Supporter workplace costs 45 US \$. A license for customers costs 15 US \$. Customer licenses can be acquired only in bundles of 50 licenses.

Like VFX, Radmin is also protected with an activation key.

If the customer wants to use the Remote control, Radmin can be used immediately. If after the 30-day trial period a connection is attempted to be established, the Supporter is required to transfer a registration key to the customer computer.

The registration key can be transferred through the Radmin connection by the Supporter toward the customer computer.

In the properties of the remote entry enter the subdomain name in the IP address box.

Beside the remote control, Radmin offers also the possibility for file transfer.

The access to the customer computer can be protected by a password.

The Remote control from the point of the Supporters

The customer should start the remote control only after consultation with the Supporter. The remote control program allows the unlimited access to the customer's PC and is a considerable security risk for the customers!

It is not very likely that awaiting Radmin server with a dynamically assigned IP address can be quickly found in the Internet by hackers. In addition, we can protect the customer's PC by a password which is entered when establishing connection to the customer's PC.

In the window of the Radmin-Viewers the customer's PC can be operated just like the own PC.

Now the customer computer can be found as a subdomain name in the Internet. On your Supporter workplace you need only one entry for the Remote control to all customer computers.

VFX Project Update Wizard

Now you can use the new VFX wizard to update your projects created with earlier VFX versions.

When you open a project via TaskPane the version of the project being open is compared with current VFX version and if they differ you are offered to update your project to the new version. Upon this action a backup of your project will be created.

VFX Project Update Wizard will renew your base cases libraries, system report files and procedure files. The Vfxmsg table in your project is merged with latest Vfxmsg table and all include files are generated.

System VFX tables are actualized with newly added fields and content.

You can also run Project Update Wizard from the VFX 9.0 main menu.

VFX Task Pane

Two new features are included in VFX Task Pane. You can now easy create backup of your projects using the new backup functionality. The  icon starts the project backup. You are asked to specify the name of the backup file. If the project is opened at the moment, it is closed before creating the backup.

The other great feature included in the VFX Task Pane is the Project Update Wizard. Project Update Wizard is automatically started when a project created with an older VFX version is opened and you are prompted if you want to update the project to the latest VFX version.

VFX Toolbox

If you like using the toolbox, now VFX libraries are added to toolbox when your project is opened. The toolbox is reloaded when you open another project, so you have libraries of your currently open project always available.

VMD

Now you do not need to recreate include files when new menu constants are added to Vfxmsg table. The new VMD automatically generates include files when menu file is saved.

VFX Online registration

Along with former ways to register your VFX copy, now it is possible to register VFX online and to receive your activation key immediately.

Using the *Register online* button in VFX registration form, your data are sent to the VFX site and your key is returned back. The registration form checks if you have valid registration and registering process completes automatically.

If you are not able to perform online registration, you can register your VFX copy by entering received activation key in the registration form.

If you are using a 30 days trial key, VFX automatically shows you the remaining time and you can also click on *Buy Now* button which will redirect you to VFX website and you can buy your license online.

Trifles

- The VFX Builder is called when a Pageframe is selected.
- Support of views and Cursoradapter by displaying the Audit Trails
- Support of all VFX classes in Builders
- As a separator in all VFX properties, comma or semicolon can be used alternatively
- Additional new fields cins_time and cedt_time to the keep the last processing time
- If readonly property is set to .T., tabstop is set to .F. automatically
- New methods of the application object onprestart and onpoststart as additional hooks
- cpickfield builder: the properties cfieldlist and cfieldtitle can be keyed up directly in a textbox in the Builder

New features for end user

Managing users

User list form

VFX 9.0 provides new advanced features in maintaining application users.

A new option for Administrator user is the ability to reset resource data for all users using *Reset All Users* button. This button clears saved resource records for all users without need to navigate through user records.

The screenshot shows the 'Edit - User List' dialog box. The 'List' tab is selected. The fields are as follows:

Username:	ADMIN	Password:		User Level:	1
Name:	Administrator	User Access:			
E-mail:					
Show Forms first in:	<input type="radio"/> Edit Mode	Form Size:	0.0	Number of Last File Entries:	4
	<input checked="" type="radio"/> List Mode				
	<input type="checkbox"/> Change password at NEXT login			Clear Resource	
	<input type="checkbox"/> Allow user to change password			Reset All Users	

At the bottom, a list box labeled 'UserGroup' is empty.

For every user can be set a requirement to change the password next time when login.

Administrator can also prohibit particular user to change his password.

In addition to former way of defining user right, the administrator can now define user groups, assign users to one or more user groups and give necessary right to user group.

In the lower part of the form resides the list of groups, this user is member of.

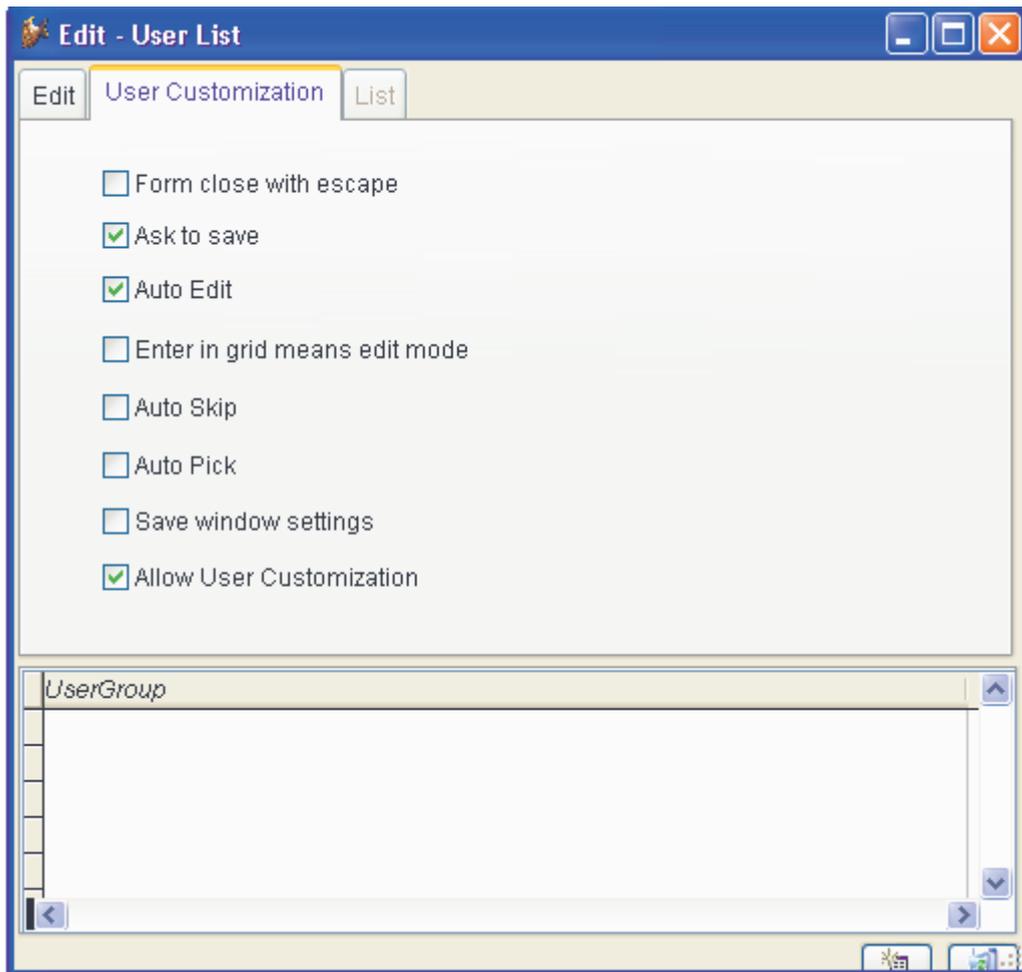
The user can be a member of more than one user group, combining right which every one of these groups permits. When a particular user is member of more than group the less restrictive rights are used. What does this mean? If the user is member of a group which do not has right to perform a particular action, but this user is also a member of another group, which gives him right to perform that action, as an effective right for this user execution of that action will be allowed.

How to assign rights to user groups will be discussed in Managing User Groups section below.

If a particular user is not assigned to any user group, access rights are defined by the user's security level as in former VFX versions.

It is also possible now, to give users better flexibility in setting environment. By setting the *AllowUserCustomization* property of goProgram object, the developer can allow end-users to define their own global environment.

When this property is set to .T. the end-user administrator can make global customization and in turn, enable or disable customization for all other users. The checkbox *Allow User Customization* is used to control whether other users are allowed make their own environment setting.



If the end-user administrator disables user customization option, the settings defined under Administrator's account become global settings for all users of the application.

Currently logged users

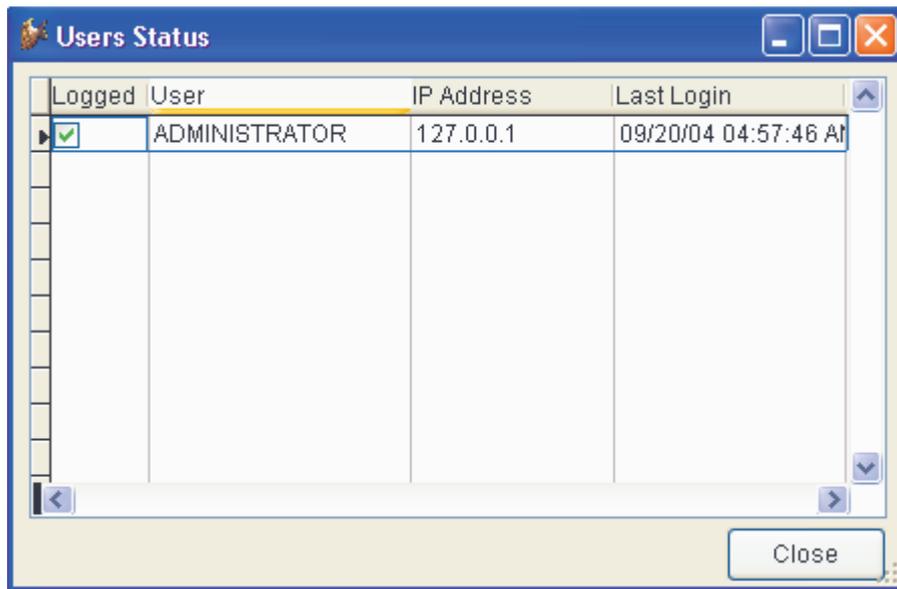
VFX 9.0 keeps track of logged users with possibility to block users from login more than once at a time.

The property of goProgram object, *lAllowMultipleLogin* controls the behavior of the application when at login time appears that the user is already logged in the system. For every user is kept the IP address from where he logged from. After user log out, IP address field is cleared. Depending of where VFX system tables reside, this feature uses different approach to identify if user is really active now or for some reasons his connection had broken and login attempt is to re-enter the system, but not to login simultaneously twice.

When DBF tables are used, the record for the user is locked all the time while he is logged in the system. In case of broken connection or another emergency exit, record lock is automatically released and when user attempt to login next time this action is not threat as multiple login.

When VFX sys tables reside in a SQL Server database, the system process Id is used. SPID is kept in vfxusr table and on second login attempt is checked if that process is still active. In case that process is active and multiple login is not disabled by the developer, user is not allowed to log in the application.

Along with that the administrator is able to see users' status by choosing *Tools / Users Status* from main menu.



Logged	User	IP Address	Last Login
<input checked="" type="checkbox"/>	ADMINISTRATOR	127.0.0.1	09/20/04 04:57:46 AM

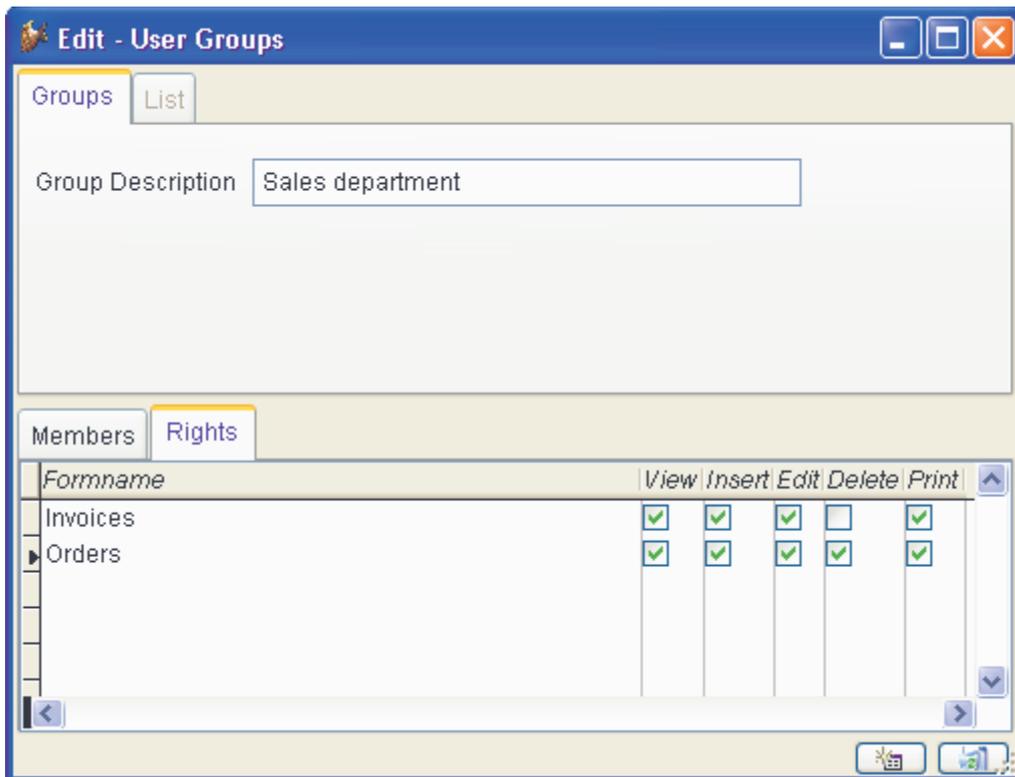
In *Last Login* column is displayed the time when user logged-in last time. For users, which are currently logged in the system, the column *IP Address* contains the IP address of the workstation where the user is working.

Managing User Groups

VFX 9.0 offers a new way to manage user access to application's modules.

Administrator can define user groups and specify access rights to every user group for every form in the application.

User rights are defined for all forms, included in Vfxfopen table.



To every user group are assigned users, which are members of that group. A particular user can be member of more than one user group. When the user is member of more than group rights which every one of these groups permits are combined and the less restrictive rights are used. What does this mean? If the user is member of a group which do not has right to perform a particular action, but same user is also a member of another group, which gives him right to perform that action, as an effective right for this user execution of that action will be allowed.

At run time is created a global object goUserRights. This object holds child objects for every form of the application and every child object has properties corresponding to user rights for the particular form.

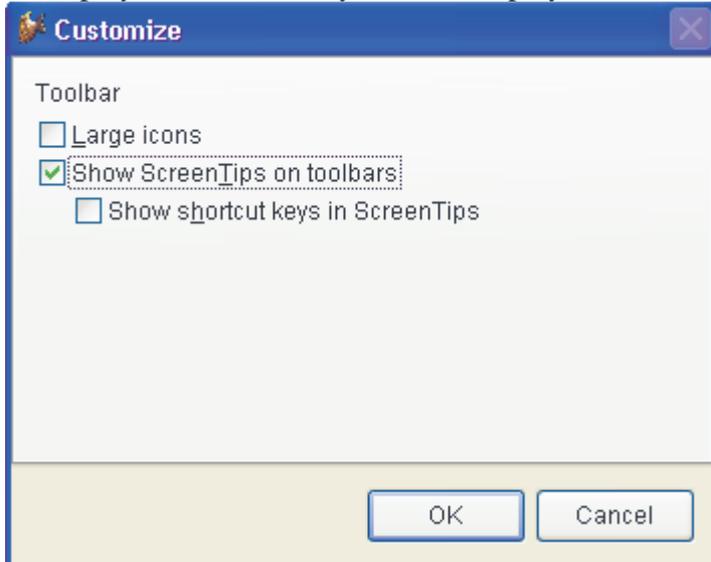
If a particular user is a member of the group “Sales department” shown on the screenshot above, the object goUserRights and its members will look in this way:

Name	Value	Type
goUserRights	(Object)	O
frminvoices	(Object)	O
deletepermit	.F.	L
editpermit	.T.	L
newpermit	.T.	L
printpermit	.T.	L
viewpermit	.T.	L
frmorders	(Object)	O
deletepermit	.T.	L
editpermit	.T.	L
newpermit	.T.	L
printpermit	.T.	L
viewpermit	.T.	L

If a particular user is not assigned to any user group, access rights are defined by the user's security level as in former VFX versions.

Layout

The layout of the application created with VFX 9.0 is improved with new XP style icons. New icons are developed for toolbar and menu entries. Users are able to define their own layout setting. The icon size can be set (large or small) along with setting if tool tip texts will be displayed and if hot keys will be displayed in tool tip texts.



If *Show shortcut keys in Screen Tips* checkbox is marked, in the texts, shown as tool tip is also included the hot key combination for that action. For example for *New* button the hot key combination is Ctrl+N

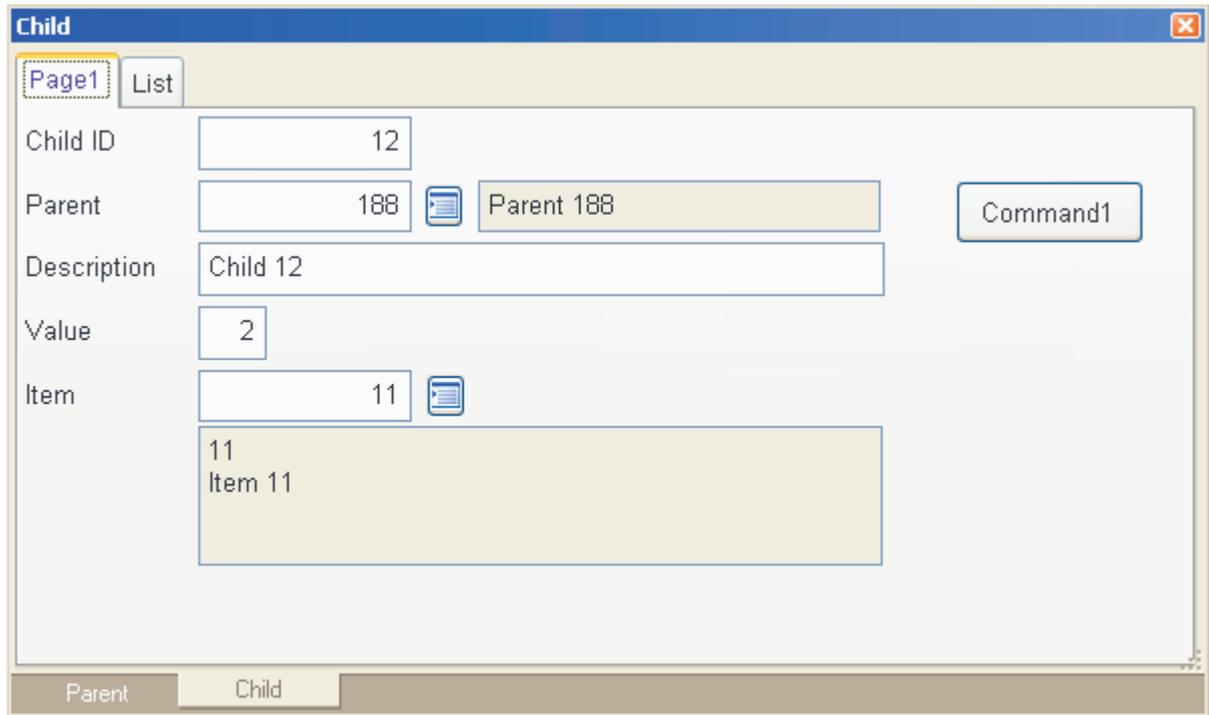


New for VFX 9.0 layout is also the ability to select a bitmap as background for a page in the pageframe in all data forms. The picture can be set in VFX form builders.

In VFX form builders it is also possible instead of setting background picture for a page, to define background color.

Docked forms

In VFX 9.0 is also supported forms' docking. Forms can be docked together, but not to the border of the application.



The docking behavior of forms is controlled by *goProgram.nDockable* property. If this property is set to -1, for every particular form are used its own settings. When *goProgram.nDockable* contains a value greater than -1, this value is stored into Dockable property of all modeless forms.

Note: When the WindowType of a form is *Modal*, *goProgram.nDockable* value is not applied to the form.

The docking state and dock position of forms is saved per form and per user in Vfxres table.

Fax

VFX 9.0 supports the fax programs FRITZ!fax from the AVM and Winfax from Symantec. VFX 9.0 automatically recognizes whether one of these fax programs is installed. If a fax program is recognized, the report output will be directed to the suitable fax printer driver.

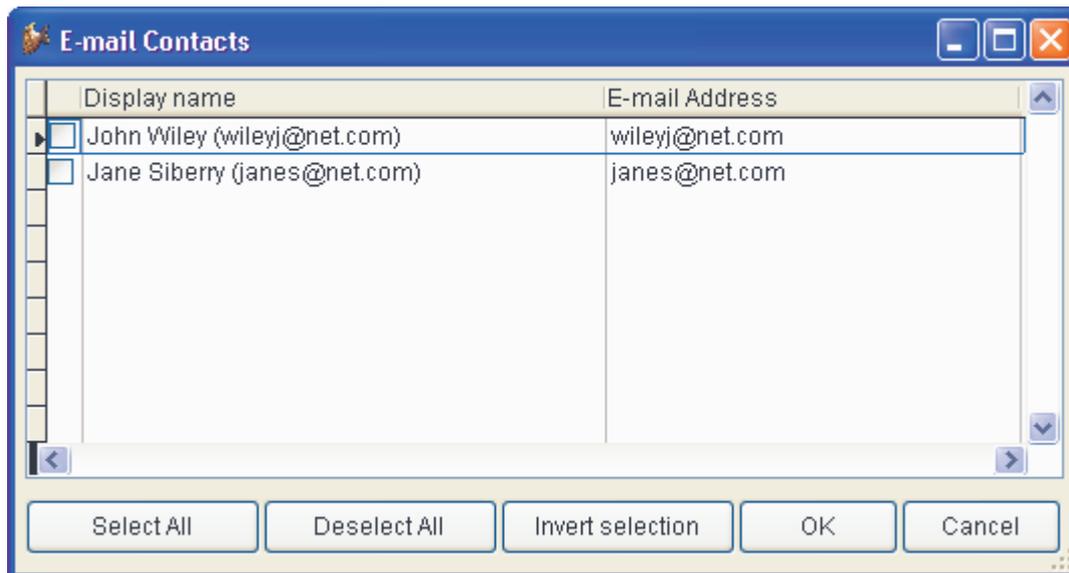
*** Screenshot from Grid report dialog

A VFX dialog requests the fax number in the application. Then the fax number will be passed from the VFX application directly to the VFX program. The end user does not see dialogs of the fax program.

*** Screenshot entering the fax number

Report issues: export to file and send in e-mail

VFX 9.0 supports several possibilities to save the VFP 9.0 report output in files. Generated reports can now be saved in several new formats. The supported formats are PDF, HTML, XML, TIFF and BMP. These file types can also be used to create an attachment and e-mail the report.



All addresses selected in *E-mail contacts* form are filled back into correspondent recipients list.

Business Graph

Statistical reports are an endless digits list, quite hard to read and analyze. The best way to demonstrate calculated results is to create colorful graphical presentation.

cBusinessGraph is a great new class, giving you a possibility better to represent application data as graph just in few minutes.

To draw the graph, *MSChart ActiveX* control is used and series data is read from a table or cursor. Every column in the cursor is considered to be one series in the graph. It is supposed one of fields in the cursor to hold label texts. If you do not specify field which contains labels, all fields in the cursor are used as series data. Fields, containing series data, should be of numerical data type. Additionally you have to provide series titles, which are displayed as legend text.

Properties:

cAliasName – alias to be used to obtain series data.

cGraphTitle – title for the graph.

cLabelField – name of the field which will be used to set data labels in the graph.

cLegendTitles – coma delimited list containing series titles.

lShowLegend – determines whether to show legend in the graph

nGraphType – type of the graph:

- 0 - 3D BAR
- 1 - 2D BAR
- 2 - 3D LINE
- 3 - 2D LINE
- 4 - 3D AREA
- 5 - 2D AREA
- 6 - 3D STEP
- 7 - 2D STEP
- 8 - 3D COMBINATION

14 - 2D PIE
16 - 2D XY

Methods:

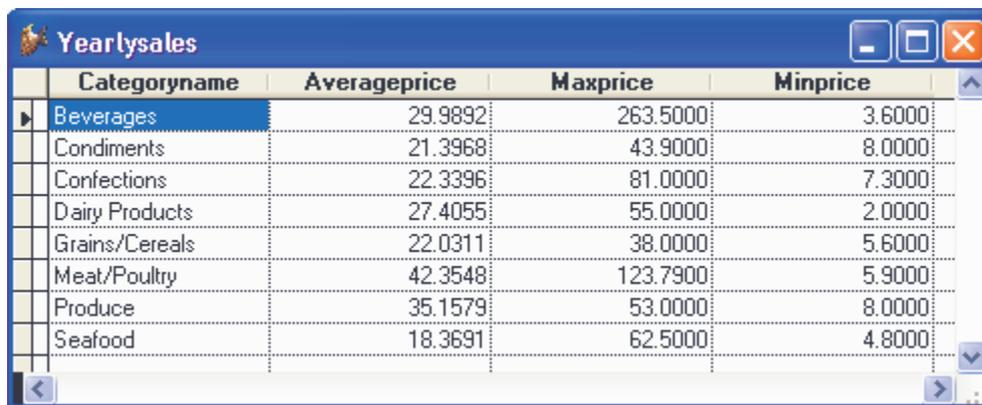
DrawGraph – Draws the graph using provided data and control's settings. All series data, labels and titles are set accordingly control settings.

OnPrint – Prints a copy of the graph using the hardcopy report.

Example:

Let's use a cursor containing yearly sales statistical information about average sale price, minimal sale price and maximal sale price for a year. We will need to group results by product group and finally our goal is to draw a graph.

Let's assume that the resultant cursor contains these data



Categoryname	Averageprice	Maxprice	Minprice
Beverages	29.9892	263.5000	3.6000
Condiments	21.3968	43.9000	8.0000
Confections	22.3396	81.0000	7.3000
Dairy Products	27.4055	55.0000	2.0000
Grains/Cereals	22.0311	38.0000	5.6000
Meat/Poultry	42.3548	123.7900	5.9000
Produce	35.1579	53.0000	8.0000
Seafood	18.3691	62.5000	4.8000

For the instance of `cBusinessGraph` should be made these settings:

`.cAliasName = "YearlySales"`

`.cGraphTitle = "Yearly sales statistic"`

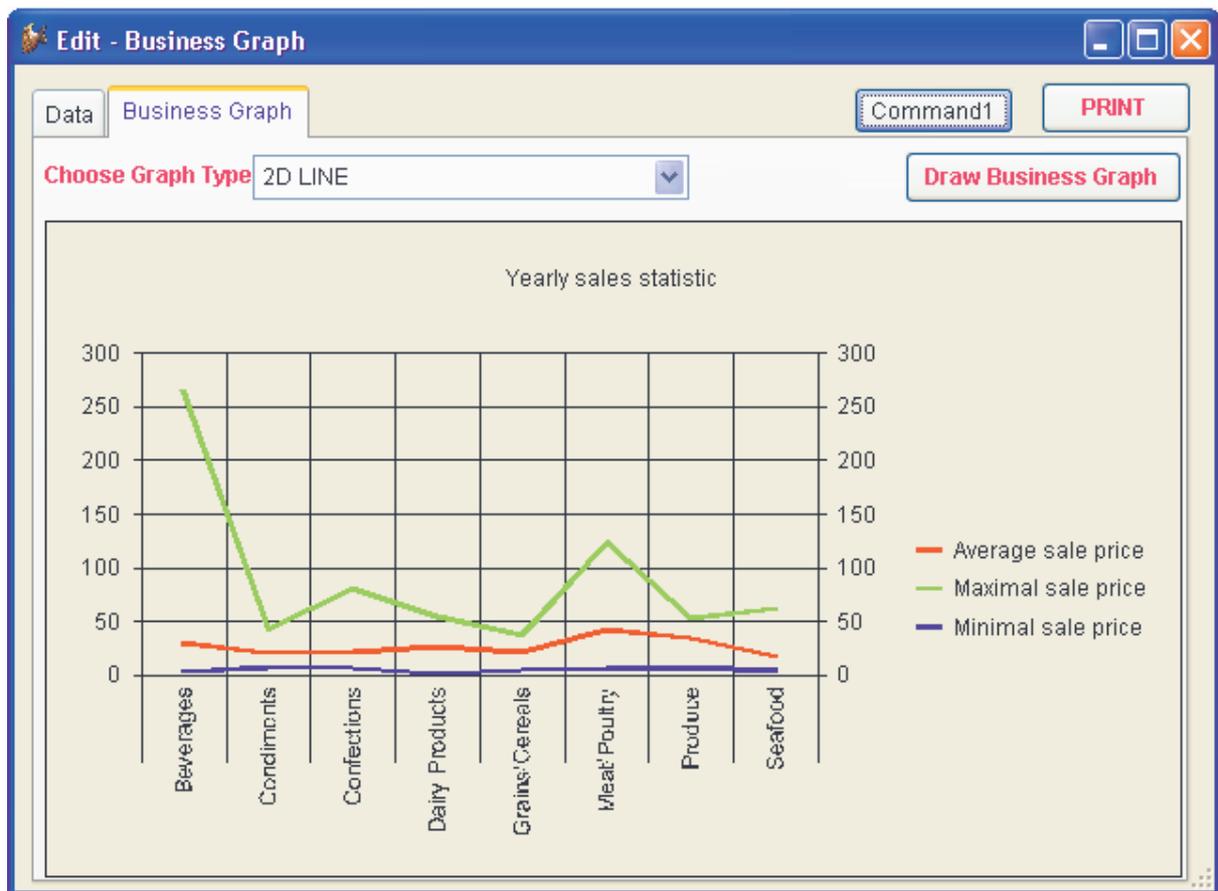
`.cLabelField = "CategoryName"`

`.cLegendTitles = "Average sale price, Maximal sale price, Minimal sale price"`

In property `cLabelField` is stored the name of the fields containing text description – `CategoryName`

In property `cLegendTitles` are enumerated texts, which need to be used as a legend for data series. The sequence of texts must be same as fields order in the cursor.

Now calling the `DrawGraph` method the graph is created



Runtime localization

In VFX 9.0 it is possible not only to develop localized applications, but also to allow end-user to change application language at run time.

Runtime localization feature is controlled by `goProgram.IRuntimeLocalization`. When this property is set to `.T.`, a new language selection combo box is available in Login dialog and in main application toolbar.



User can chose language at login time and he can change application language at any time while application runs.

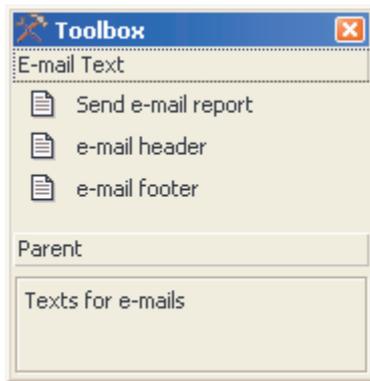
When runtime localization is used, a global object goLocalize is created at run time. This object has child properties, corresponding to messages in Vfxmsg table. For every record in Vfxmsg table is created a child property. The name of the child property is generated from message_id, prefixed with “c”. For instance if the message_id in Vfxmsg table is CAP_APPLICATION_TITLE, the property of goLocalize object will be named CCAP_APPLICATION_TITLE.

You can use goLocalize object properties every time you need.

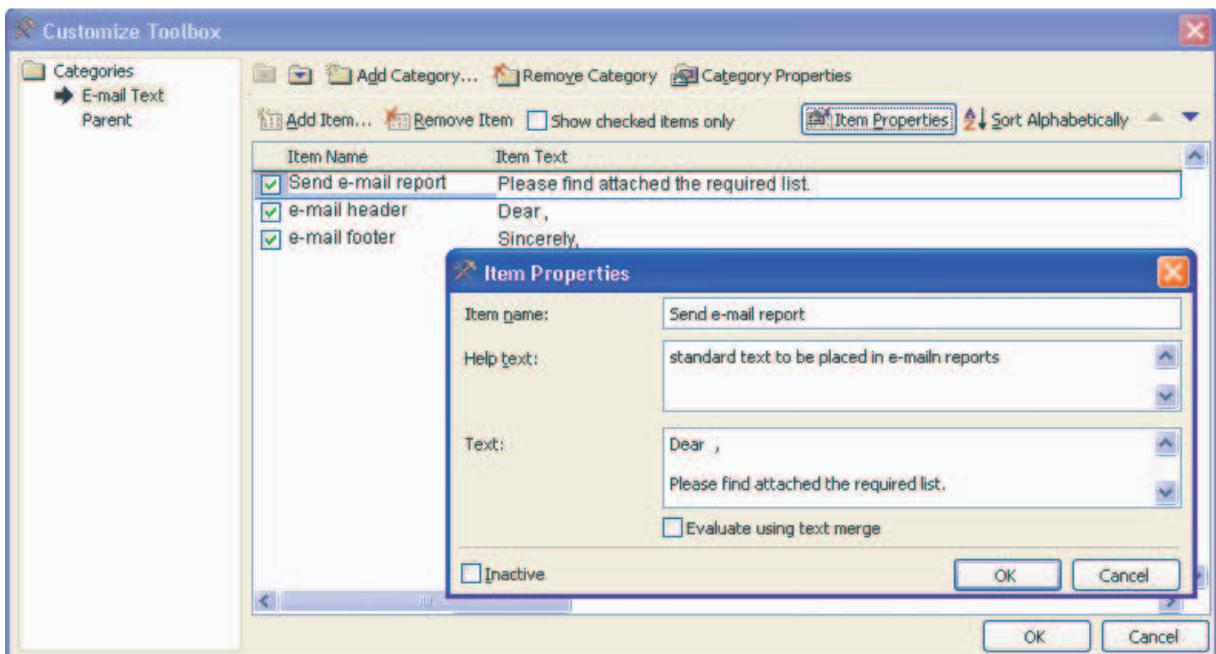
The lat used language is kept per user and next time when user logs in the system, the last chosen language will be automatically selected.

Toolbox for end-users

The useful toolbox functionality is extended to end-users. Users can define specific text that can be easily dropped-down into text boxes and edit boxes in forms. Similar to VFP toolbox, text items are grouped into categories. Dragging an item from the toolbox dropping it into a textbox or Editbox in a form will place the item text at the cursor position.



Users can customize toolbox by adding, editing and deleting categories and items



For category are specified category name and help text and for item are specified item name, help text and item text.

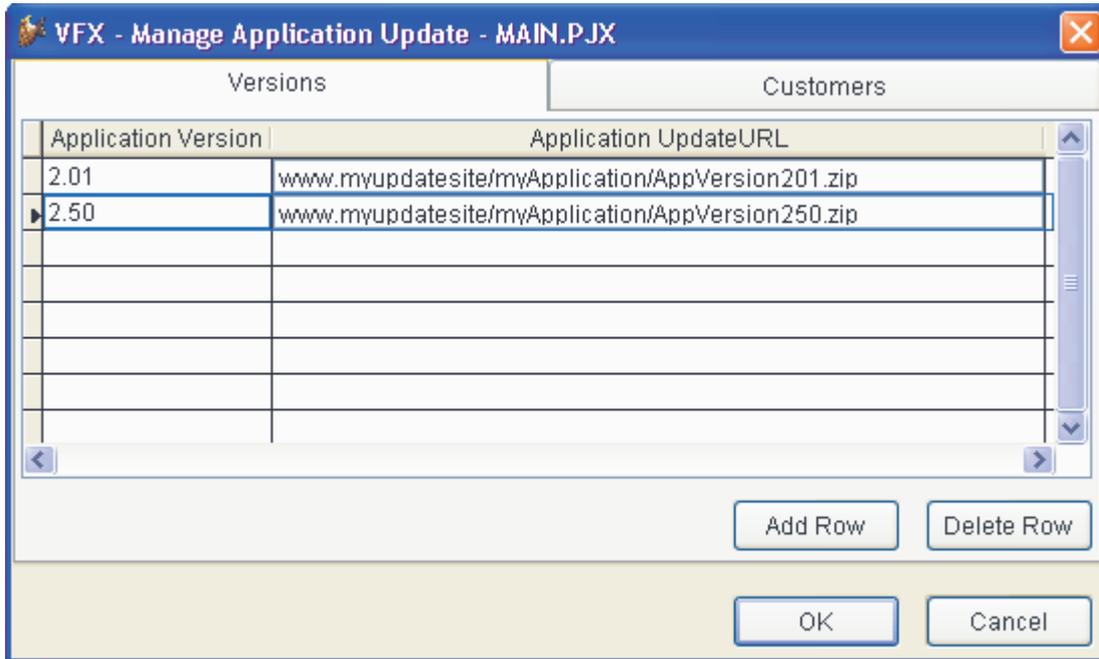
Item text is the text that is placed in the form when item is dropped. Category name text and Item name text is displayed in the toolbox form and the help text content is displayed at the bottom of the toolbox form as a description of the active item.

Using  and  icons, user can change the sequence in which categories are ordered in the toolbox window. Reordering text items can be made using  and  icons in *Customize Toolbox* dialog.

Application update

The ability to update the application at customer's side, by downloading the latest application version, is now extended. The developer defines the list of customers who are allowed to download and install the latest application version. The list is encrypted and saved in a file, which is downloaded and checked at the customer's side. Along with customers list developer is able to define different files to be download, depending on version which customer is currently running.

Defining version update files and customers list is started by choosing *Manage Application Update* option in VX 9.0 menu



In column Application version you have to enter the version number of existing customer's application in same way as it is defined in running application.

Two steps are executed at customer's side when application update runs. First step is to execute a macro script, which is designated to download customer's list (UpdateCustomer.vfx) and version list (UpdateVersion.vfx) files. This script has to be defined by the developer and is kept in UpdateApp field in VfxSys table.

Next step, after these two files are downloaded and decrypted, is to determine if current user is allowed to download the new application version and to find URL to application update file which need to be downloaded.

The file that will be downloaded is determined based on the version of currently running application.

Enhanced search functionality

The search dialog is improved by limiting end-users to enter only valid values for filtering data. The comparison operators depend on data type of the chosen data field in the column *Field*. When it is of character data type along with all common operation types user can choose also *contains*. In that case filter will be built using \$ operator. Additionally, in the last column of the grid – *A/a*, user can specify if filtering should be case sensitive. For logical data type fields, *True* and *False* values are listed in the *Operation* Combobox and it is not necessary to specify additional value in *Value* column of the grid.



The column *Value* contains several controls and *CurrentControl* is changed dynamically, depending on the type of data field chosen in *Field* column. When data type of chosen data field is character, the active control in column *Value* is a textbox formatted to accept character value with appropriate input mask. In case of numeric data type, the active control is a text box with input mask allowing only numeric value in respective format, considering size and decimal places of the field being compared. Same method is used for data and datetime data types. In this way user is protected against entering illegal values in *Value* column.

Enhancements concerning Treeview

cTreeView class is now enhanced to have better performance while loading and to restore the status of the TreeView control. The status of nodes in TreeView control is saved when form is closed and is restored on next form's load.

New properties of cTreeview class:

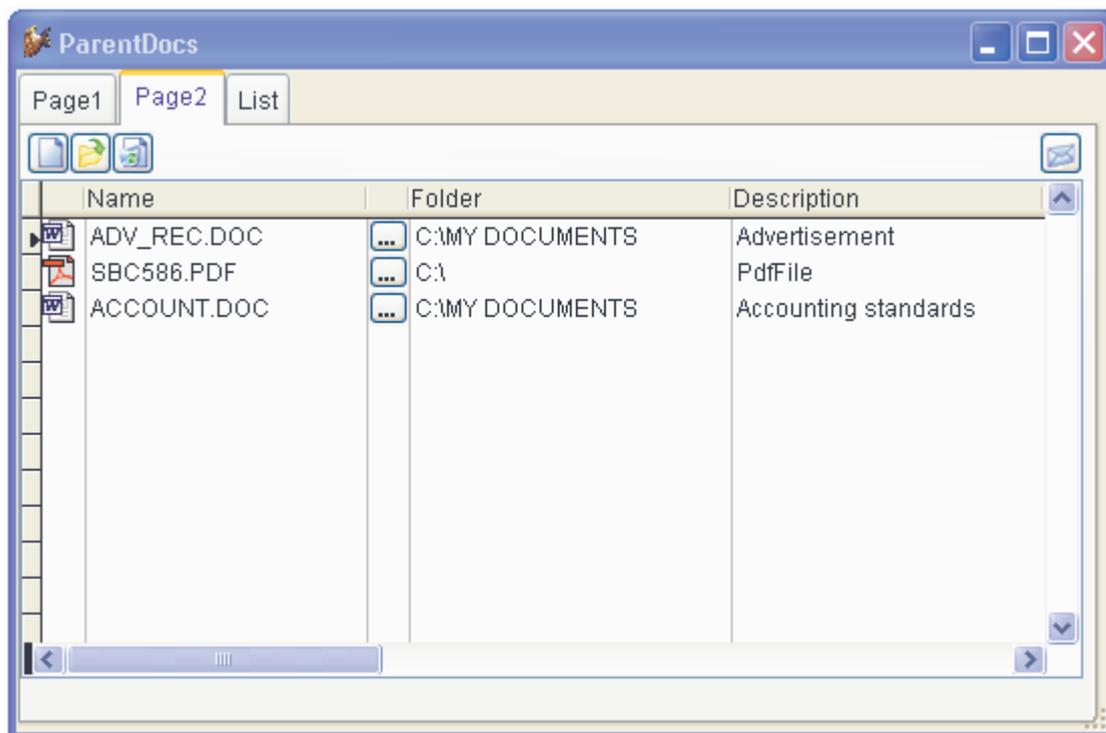
ILoadAllTreeviewNodes – if this property is set to .T. all nodes are loaded in the TreeView control when form loads. When the property value is .F., in the TreeView control are loaded only nodes at first level and nodes that are expanded when status of the control is restored. Child nodes that are not visible when form is loaded, will not load at that time. These nodes are loaded in the Treeview control when their parent node is expanded.

IRestoreTreeviewStatus – When this property is set to .T. a list of expanded nodes is saved when the form is closed and on next load of the Treeview control same nodes are expanded to restore treeview status.

The grid report in forms based on cTreeviewForm and cTreeviewOneToMany form classes is also printed as a treeview structure.

cDocumentManagement class

The new class cDocumentManagement is designated to maintain documents related to application data. cDocumentManagement class acts as a container for child data related to the record of main table. Class allows the end-user to open related documents and also to send them as attachment in an e-mail.



Properties:

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, specified in *cPicturePreviewFormname* property. If the property value is .F., picture documents are opened in the application associated with their file extension.

cPicturePreviewFormname – name of the form to be used to preview picture documents.

cPicturePreviewCaption – string to be passed to picture preview form and used by that form as caption.

About dialog

In the about dialog is placed a link to open the end user license agreement (EULA). This link invokes a dialog form where user can read the EULA and to print it.

Trifles

- Support of the incremental search even when the field contains .NULL.
- Localized Hotkeys for the class cpickdate and a multiline tool tip text
- New classes: e-mail with e-mail client call, hyperlink with Internet Explorer call, numerical text box with calculator call, TAPI, file selector with open file dialog
- The PDF export folder is saved per user
- Conform to visible =.F. for Grid-Columns for the search dialog and the print dialog
- Progress bar by the actualization of the customer database
- Script for download and install Adobe Acrobat reader (for PDF documents)
- Keyboard support in the XP opening dialog
- Drag and drop support in Mover dialog
- Position on last viewed record when form is reopened

- Support of the HighLightStyle property in Grids
- Improved memo fields visualisation in Grids
- When all entries in favorites menu are deleted, the corresponding menu bar is also removed