
xHarbour.com
Visual xHarbour
Getting started manual
for Microsoft Windows
(version 1.0, July 2007)

Copyright © 2007 xHarbour.com Incorporation. All rights reserved. All trademarks appearing in this manual are trademarks or registered trademarks of their respective companies. This documentation as well as the software described herein is delivered under license and may only be used or copied according to the license agreement, as can be found in the file "copying" on the xHarbour Builder CD. This documentation only serves as a source of information and may be subject of changing without notice. xHarbour.com Inc. neither takes responsibility nor any liability for errors or inaccuracies in this manual.

Content

1.	PREFACE	1-13
1.1.	Introduction	1-13
1.2.	Tell us what you think!	1-14
2.	INSTALLING VISUAL XHARBOUR	2-16
2.1.	What's in the box?	2-16
2.2.	Download Visual xHarbour from the web.....	2-16
2.3.	Your personal Visual xHarbour Product Key.....	2-17
3.	VISUAL XHARBOUR IDE.....	3-19
3.1.	The ToolBox	3-20
3.2.	The Work Area.....	3-20
3.2.1.	Start Page	3-20
3.2.2.	xHarbour Developer Network	3-21
3.2.3.	Source Code Editor	3-21
3.2.4.	Form Designer	3-21
3.2.5.	Build / Error / Log Window	3-22
3.2.6.	Component Window	3-22
3.3.	The Object Manager.....	3-22
3.3.1.	Properties and Events	3-23
3.3.2.	Object / File view	3-23
4.	HOW VISUAL XHARBOUR WORKS	4-24
4.1.	The physical structure of a VXH project.....	4-25
4.1.1.	The VXH file.....	4-26
4.1.2.	The Resource folder	4-26
4.1.3.	The Source folder	4-26
4.1.4.	The Bin folder	4-27
4.1.5.	The Obj folder	4-27
4.2.	Building the Getting Started sample.....	4-27

5.	CREATING PROJECTS WITH VISUAL XHARBOUR.....	5-30
5.1.	Create a new project.....	5-30
5.2.	Create a new form.....	5-31
5.3.	Adding controls to the form	5-32
5.4.	Manipulating controls.....	5-33
5.4.1.	Properties	5-33
5.4.2.	Events.....	5-34
5.5.	Adding code to the interface	5-34
5.6.	Build / run a project	5-35
6.	VISUAL XHARBOUR CONTROLS.....	6-37
6.1.	System	6-37
6.1.1.	Properties	6-37
6.1.2.	Events.....	6-37
6.2.	Application	6-38
6.2.1.	Properties	6-38
6.2.2.	Events.....	6-40
6.3.	Form	6-41
6.3.1.	Properties	6-41
6.3.2.	Events.....	6-42
6.4.	Button	6-42
6.4.1.	Properties	6-42
6.4.2.	Events.....	6-43
6.5.	CheckBox	6-43
6.5.1.	Properties	6-43
6.5.2.	Events.....	6-44
6.6.	ComboBox.....	6-44
6.6.1.	Properties	6-44
6.6.2.	Events.....	6-45
6.7.	CursorComboBox.....	6-46
6.7.1.	Properties	6-46
6.7.2.	Events.....	6-46
6.8.	FontComboBox.....	6-46
6.8.1.	Properties	6-46
6.8.2.	Events.....	6-46

6.9. Edit	6-46
6.9.1. Properties	6-46
6.9.2. Events	6-47
6.10. GroupBox	6-49
6.10.1. Properties	6-49
6.10.2. Events	6-49
6.11. HeaderStrip	6-49
6.11.1. Properties	6-49
6.11.2. Events	6-50
6.12. HeaderItem	6-50
6.12.1. Properties	6-50
6.12.2. Events	6-51
6.13. Label	6-51
6.13.1. Properties	6-51
6.13.2. Events	6-52
6.14. LinkLabel	6-52
6.14.1. Properties	6-52
6.14.2. Events	6-53
6.15. ListBox	6-53
6.15.1. Properties	6-53
6.15.2. Events	6-54
6.16. RadioButton	6-55
6.16.1. Properties	6-55
6.16.2. Events	6-56
6.17. Animation	6-56
6.17.1. Properties	6-56
6.17.2. Events	6-56
6.18. ColorPicker	6-57
6.18.1. Properties	6-57
6.18.2. Events	6-57
6.19. ComboBoxEx	6-57
6.19.1. Properties	6-57
6.19.2. Events	6-57
6.20. CoolBar	6-57
6.20.1. Properties	6-57
6.20.2. Events	6-58
6.21. CoolBarBand	6-58

6.21.1.	Properties.....	6-58
6.21.2.	Events	6-59
6.22.	CoolMenu.....	6-60
6.22.1.	Properties.....	6-60
6.22.2.	Events	6-60
6.23.	CoolMenuItem	6-60
6.23.1.	Properties.....	6-60
6.23.2.	Events	6-61
6.24.	DataGrid.....	6-61
6.24.1.	Properties.....	6-61
6.24.2.	Events	6-62
6.25.	GridColumn	6-63
6.25.1.	Properties.....	6-63
6.25.2.	Events	6-65
6.26.	DateTimePicker.....	6-65
6.26.1.	Properties.....	6-65
6.26.2.	Events	6-66
6.27.	DriveComboBox.....	6-67
6.27.1.	Properties.....	6-67
6.27.2.	Events	6-67
6.28.	ExplorerBar	6-67
6.28.1.	Properties.....	6-67
6.28.2.	Events	6-67
6.29.	Expando.....	6-67
6.29.1.	Properties.....	6-68
6.29.2.	Events	6-68
6.30.	FolderList.....	6-68
6.30.1.	Events	6-68
6.31.	FolderTree	6-69
6.31.1.	Properties.....	6-69
6.31.2.	Events	6-69
6.32.	ListView	6-69
6.32.1.	Properties.....	6-69
6.32.2.	Events	6-70
6.33.	ListViewColumn	6-70
6.33.1.	Properties.....	6-70
6.33.2.	Events	6-70

6.34. ListViewGroup	6-71
6.34.1. Properties	6-71
6.34.2. Events	6-71
6.35. MaskEdit	6-71
6.35.1. Properties	6-71
6.35.2. Events	6-71
6.36. MonthCalendar	6-72
6.36.1. Properties	6-72
6.36.2. Events	6-72
6.37. OptionBar	6-73
6.37.1. Properties	6-73
6.37.2. Events	6-73
6.38. OptionBarButton	6-73
6.38.1. Properties	6-73
6.38.2. Events	6-74
6.39. PageScroller	6-74
6.39.1. Properties	6-74
6.39.2. Events	6-74
6.40. Panel	6-74
6.40.1. Properties	6-74
6.40.2. Events	6-74
6.41. PictureBox	6-75
6.41.1. Properties	6-75
6.41.2. Events	6-75
6.42. ProgressBar	6-75
6.42.1. Properties	6-75
6.42.2. Events	6-76
6.43. Splitter	6-76
6.43.1. Properties	6-76
6.43.2. Events	6-77
6.44. StatusBar	6-77
6.44.1. Properties	6-78
6.44.2. Events	6-78
6.45. StatusBarPanel	6-78
6.45.1. Properties	6-78
6.45.2. Events	6-79
6.46. TabControl	6-80

6.46.1.	Properties.....	6-80
6.46.2.	Events	6-80
6.47.	TabPage.....	6-82
6.47.1.	Properties.....	6-82
6.47.2.	Events	6-83
6.48.	TabCloseButton	6-83
6.48.1.	Events	6-83
6.49.	TabPinButton	6-83
6.49.1.	Events	6-83
6.50.	ToolBar	6-84
6.50.1.	Properties.....	6-84
6.50.2.	Events	6-85
6.51.	ToolButton.....	6-86
6.51.1.	Properties.....	6-86
6.51.2.	Events	6-88
6.52.	TreeView	6-88
6.52.1.	Properties.....	6-88
6.52.2.	Events	6-89
6.53.	TreeListView.....	6-90
6.53.1.	Properties.....	6-90
6.53.2.	Events	6-90
6.54.	TreeviewItem	6-90
6.54.1.	Properties.....	6-90
6.54.2.	Events	6-91
6.55.	WebBrowser	6-91
6.55.1.	Properties.....	6-91
6.55.2.	Events	6-92
6.56.	ContextMenu	6-92
6.56.1.	Properties.....	6-92
6.56.2.	Events	6-92
6.57.	CMenuItem.....	6-92
6.57.1.	Properties.....	6-92
6.57.2.	Events	6-93
6.58.	ImageList	6-94
6.58.1.	Properties.....	6-94
6.58.2.	Events	6-95

6.59. MemoryTable	6-95
6.59.1. Properties	6-95
6.59.2. Events	6-95
6.60. MemData	6-96
6.60.1. Events	6-96
6.61. NotifyIcon	6-96
6.61.1. Properties	6-96
6.61.2. Events	6-97
6.62. SerialPort	6-97
6.62.1. Properties	6-97
6.62.2. Events	6-97
6.63. Service	6-97
6.63.1. Properties	6-97
6.63.2. Events	6-98
6.64. Timer	6-98
6.64.1. Properties	6-98
6.64.2. Events	6-98
6.65. WinSock	6-99
6.65.1. Properties	6-99
6.65.2. Events	6-99
6.66. DataConnector	6-99
6.66.1. Properties	6-99
6.66.2. Events	6-100
6.67. DataTable	6-100
6.67.1. Properties	6-100
6.67.2. Events	6-101
6.68. Data	6-102
6.68.1. Properties	6-102
6.68.2. Events	6-103
6.69. ColorDialog	6-103
6.69.1. Properties	6-103
6.69.2. Events	6-103
6.70. FolderBrowserDialog	6-103
6.70.1. Properties	6-103
6.70.2. Events	6-103
6.71. OpenFileDialog	6-104
6.71.1. Properties	6-104

6.71.2.	Events	6-104
6.72.	PrintDialog.....	6-104
6.72.1.	Properties.....	6-104
6.72.2.	Events	6-105
6.73.	SaveFileDialog	6-105
6.73.1.	Properties.....	6-105
6.73.2.	Events	6-105

1. Preface

1.1. Introduction

Welcome to Visual xHarbour!

This is the “Getting started” manual for Visual xHarbour, xHarbour.com Inc.’s Integrated Development Environment (IDE) for xHarbour. It will help you to get started with Visual xHarbour. For the time being, Visual xHarbour is only available for the Windows platform.

A more in-depth technical reference about the xHarbour language itself can be found in our “xHarbour Language Reference Guide” which is available from xHarbour.com.

xHarbour originated as an “Open Source - Free Software project”. “Open Source” and “Free Software”, are terms referring to a new trend in the software market, publicized first by software products such as the “Linux Operating System”. The term “Free Software” is not intended to suggest “free of cost”, but rather “free” as in “freedom”. Free to be modified, extended, etc., as well as the freedom to sell. The “Free Software” concept is actually not new, but became more popular in recent years as a direct result of the phenomenal success of Linux, and related products.

Visual xHarbour is xHarbour.com Inc.’s Integrated Development Environment (IDE) for xHarbour. So, how did **Visual** xHarbour actually get started? The xHarbour Project (the foundation of Visual xHarbour), was started by Ron Pinkas in late 2001, as a separate project from the Harbour Project (<http://www.harbour-project.org>) where he was one of the leading developers. xHarbour was established to provide as a more aggressive alternative to the conservative development style of the Harbour Project.

Shortly thereafter, xHarbour has fulfilled its promise, and introduced numerous new features, performance improvements, and greater levels of reliability compared with the original Harbour compiler.

Soon, It became clear to Ron Pinkas that xHarbour could benefit greatly from commercial distribution and support. Early in December 2002, Ron Pinkas contacted Patrick Mast to discuss that idea and shortly after xHarbour Inc. was established.

xHarbour.com Inc. aims to establish xHarbour as the Development Product of choice, for the extended xBase Developer Community. xHarbour.com Inc. attempts to follow in the footsteps of Red Hat Inc and similar companies, merging the stability and longevity that commercial distribution offers with the freedom and security offered by Open Source -

Free Software projects. This powerful combination has already demonstrated its tremendous market appeal and the viability of this business model.

Visual xHarbour development started not long after xHarbour.com Inc. was established.

We used xHarbour as the development tool to build Visual xHarbour. No other tool or 3rd party libraries have been used. This again shows the strength of xHarbour as a language and development tool. The xHarbour language together with a RAD (Rapid Application Development) tool as Visual xHarbour; building modern Windows business application has never been faster and easier!

1.2. Tell us what you think!

As the reader of this book, YOU are our most important critic and commentator. We value your opinion and want to know what we're doing right, what could be better, what areas you'd like to see covered in a future revision of this book, and any other words of wisdom you're willing to pass our way.

We tried to prepare this book carefully; however we are only human and there is only so much time available for writing and tweaking. So, you may find some errors, inconsistencies, or subjects that could be described better.

You can write to us directly or email to VXH-Getting-Started@xHarbour.com to let us know what you did or didn't like about this book—as well as what we can do to improve any future books.

Please note that we cannot personally help you with technical problems related to the topic of this book. You should use xHarbour's technical newsgroups for direct technical support. Also note that due to the high volume of email I receive, we might not be able to reply to every message personally. We apologize for this in advance.

When you write, please be sure to include the version of this book as well as your name. We will carefully review your comments and handle them with the most attention. Thank you.

2. Installing Visual xHarbour

This chapter provides complete instructions on how to install Visual xHarbour on a Windows based computer. At this moment, Visual xHarbour is only available for Microsoft Windows operating system.

Visual xHarbour works on top of xHarbour Builder and is included in all xHarbour Builder installation packages from version 'July 2007' on. If you have a previous xHarbour Builder version installed on your system, you need to download Visual xHarbour's installer from xHarbour.com and install it in the same folder as where your current xHarbour Builder is installed.

We refer to the xHarbour Builder Getting started manual on how to install of xHarbour Builder.

If a new Visual xHarbour version is released without a new xHarbour Builder release, you can install the new Visual xHarbour in the same folder as where your current xHarbour Builder is installed. See also item 2.2 "Download Visual xHarbour from the web"

2.1. What's in the box?

The list of contents for the Visual xHarbour box is available on the Visual xHarbour CD. Look for the file "package.txt" in the root directory of the Visual xHarbour CD.

If you have an "Update and service subscription" plan from xHarbour.com Inc. or you just purchased a product, you can also download your installation program from our web site.

We recommend downloading the installation file from our web site. It saves transportation costs and by downloading the software, you can start using Visual xHarbour within minutes.

2.2. Download Visual xHarbour from the web

As mentioned in the paragraph above, it's possible and even recommended to download the purchased software from our web site. Go to <http://www.xHarbour.com> and login to your "My xHarbour Account" using your user name and password. If you have forgotten your password, click on the "Send my password" link. The xHarbour.com website will send you an email with your current password.

Once logged on, click on the “downloads” link in the top menu of your account. The page loaded offers an overview of xHarbour products available for download. Click the Visual xHarbour link to navigate to the available Visual xHarbour downloads.

Here you will find a list of all currently available Visual xHarbour releases. Click the name of the Visual xHarbour version you have purchased. The download of the installation program will start. You may save the downloaded installer on your system or you may click “Open” to start the installation. We encourage you to make a backup copy of your purchased software after downloading it, in case you have computer or hard disk problems later.

The installation program will be available for download for a period of 30 days after your purchase date and/or as long as you have a active Update and Service Subscription. For more details on the Update and Service Subscription, go to <http://www.xHarbour.com/subscription>.

2.3. Your personal Visual xHarbour Product Key

Before you can install the Visual xHarbour installer package, you need to get your personal Visual xHarbour Product Key. This key can be reviewed on the Visual xHarbour download page from your “My xHarbour Account” on xHarbour.com website.

See section 2.2 on how to connect to your “My xHarbour Account” page at xHarbour.com.

At the bottom of the Visual xHarbour download page, you will find the product key(s) for your purchased software. Please keep your product key in a safe place, as it is your **personal** key. Do **NOT** share this key with anyone else, as doing so will violate your Visual xHarbour License.

Your Visual xHarbour Product Key only changes if you upgrade your Visual xHarbour version. If for example, you upgrade your Personal Visual xHarbour to a Professional version, you will receive a new Visual xHarbour Product Key. Once upgraded, your previous Visual xHarbour Product Key will expire and should be discarded.

3. Visual xHarbour IDE

Before we begin to build our first VXH (“Visual xHarbour”) project, or even explain how VXH works, let’s spend some time looking at the interface VXH has to offer. The screenshot below displays the default, empty environment which is loaded when VXH is opened.



Apart from the common menu- and a toolbar at the top of the IDE, VXH can be split up in three major sections. At the left side, a richly filled “ToolBox” shows all available objects which can be dragged and dropped on the form. The right side has the “Object Manager” which manages all objects and files in your VXH project. And finally, in the middle of the IDE, the “Work Area” where your form and source code will be displayed.

Let’s have a closer look at these three sections.

3.1. The ToolBox

The ToolBox contains every possible object which can be used by VXH. The objects are categorized by type and listed alphabetically. You can simply add an object to your project by clicking the object of your choice and clicking the form where the object should be placed. It's also possible to just drag and drop the object on the form.

At the bottom of the ToolBox you'll notice a category 'COM Objects'. You can add 3rd party COM objects via the 'View' -> 'Registered COM Objects' menu option. The shortcut for this window is F11. Select the COM object(s) you wish to add to your ToolBox and click 'OK'. When scrolling the ToolBox downward, the newly added COM objects are listed alphabetically in the 'COM Objects' category. A more in-depth usage of COM objects can be found later in this book.

A important source of information is also available on our xHarbour Developers Network at <http://www.xHarbour.com/xHDN/>.

Note:

The information on the reference guide pages of the xHarbour Developer Network are only accessible for users which have purchased our documentation or have an active subscription plan. More information about this can be found on our corporate website at <http://www.xHarbour.com/xHDN/referenceguide/index.asp?page=readme>.

3.2. The Work Area

This is probably the most frequently used section in VXH as it allows you to design the GUI of your applications and add the actual code. Let's not forget that VXH is a tool to rapidly create good-looking applications!

Notice that the Work Area is a tabbed environment with multiple functions. Let's have a closer look at each of the tabs in this section.

3.2.1. Start Page

The VXH Start Page offers an overview of the most recently opened VXH projects. From this page you can also open an existing, or to create a new project. It basically offers the functions that are most likely to be used by the user when he or she opens VXH. Notice that these functionalities are also available from the "File" menu.

3.2.2. xHarbour Developer Network

The second tab is a direct link to our xHDN section on our corporate website. The xHarbour Developer Network is a resource for documentation and xHarbour related information. It features a reference guide covering the xHarbour Language, articles, project samples and demo movies.

Please note that a part of our xHDN service is limited to customers with active subscriptions or to customers which have bought our xHarbour Language Reference Guide. When browsing the xHDN reference guide pages without an active subscription or the purchase of the Reference Guide, users will see the main part of the documentation scrambled. More information about the benefits and restrictions of our xHDN pages can be found on <http://www.xHarbour.com/xHDN>.

3.2.3. Source Code Editor

The Source Code Editor enables you to add code to the GUI you designed using the Form Designer (see 3.2.4 Form Designer). The code editor has all the features you'd expect from a code editor:

- Colorized syntax recognition;
- Collapse / expand source level entities (functions, procedures, classes, class methods);
- Markup for changed code since last save;
- Rectangular selection;
- User defined bookmarks;
- Block indentation;
- Seamless xHDN integration.

The Source Code Editor uses xEdit as its engine. More information about this free xHarbour editor can be found on our website at: www.xHarbour.com/xEdit.

3.2.4. Form Designer

The Form Designer is the last and possibly the most frequently used tab in the Work Area. It's the tab where you can visually create the GUI for your VXH project. To most new VXH users, this approach can be strange compared to the previous technique of designing a GUI.

The technique is rather simple. First create the IDE by dragging objects to the form (window) of your application, then add events and actions to the objects in order to create

the functionality you wish for your application to offer to its users. It's no longer needed to code the IDE itself, that's what VXH does for you. Main idea is that VXH saves valuable time at this stage in development. Nice advantage is that you can actually see the IDE growing with each object added to your form without ever having to build the application to see the look and feel of your application.

3.2.5. Build / Error / Log Window

When building your VXH project, a dockable window pops up at the bottom of the Work Area. This window has three tabs available: Build, Errors and Log.

The "Build" tab displays information when building the project. When all goes well, it should state "<Your Project Name> Built" at the bottom. If not, "Errors!" is displayed and the next tab, the "Errors" tab is activated.

As mentioned above, the "Errors" tab is automatically activated whenever an error is encountered while trying to build the VXH project. It offers some information about the error encountered which should help you to find the problem. When for example the error states that an incomplete statement was found on line 10 in file x.prg, you are able to double click the error entry in the Errors tab to navigate to the referenced file and line where you can correct the problem.

3.2.6. Component Window

When adding components to your VXH project (objects listed under the "Components", "Data" and "Dialogs" categories in the ToolBox), a dockable window is added at the bottom of the Work Area. When no components are added to the VXH project, the window is not visible and it is not possible to open the window manually.

3.3. The Object Manager

The Object Manager is, as the name might suggest, a tool to manager every possible object in VXH. The Object manager can by itself be split up in two sections.

The upper part has a two tabbed panel which features all properties and events of the currently selected object in VXH. Information about the selected object is displayed in a box underneath the tabs. It displays the name and type of the object.

Below is a separately dockable tree view with two possible views: the objects in your VXH project and the files uses by your VXH project.

3.3.1. Properties and Events

Since virtually all objects have a set of properties and events, this part of the Object Manager is a must-have when creating GUI based applications. It allows you to customize the appearance of an object in a matter of seconds. By browsing through the possible properties of an object, you can learn a lot. Setting values for the properties shows you on the fly what that property does for the object. Changing the “BackColor” property of the Form for example will show you immediately what effect that will have for your GUI.

Adding events to an object has never been easier. Just browse the list with possible events available for the currently selected object and double click the event name. VXH automatically activates the event and links it to the correct object. All you have to do is add the code to be executed when the event is fired.

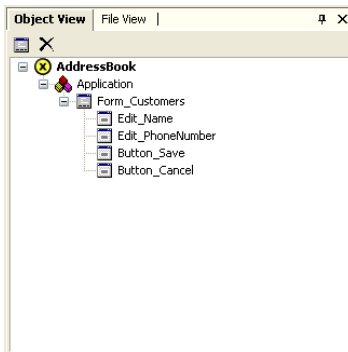
Note:

When double clicking on any object in the “Form Designer” will activate the default event for that object. Double clicking on a Button object for example will automatically generate the OnClick event as if you were to select the OnClick event from the Events list.

3.3.2. Object / File view

The bottom part of the Object Manager features a tree view which, depending on the tab activated, shows two different views.

The “Object View” shows all objects used in your VXH project in relation to other objects. The screenshot below shows a project with one form and four objects added to the form.



Depending on the location of the object in the tree it's possible to see what the relationship of the object is to other objects in the same VXH project.

When looking at the screenshot aside, have a look at the Edit_Name object. This is an Edit object to receive the name of a customer. Its parent object is the Form_Customers object (our main form). The other “Edit_” and “Button_” objects listed are on the same level as the Edit_Name object.

It's possible to have an almost infinite path of objects

that are linked to each other with a parent/child relationship. Take for example the Panel control in the VXH ToolBox. It's used to bundle a group of objects together. Imagine a Panel object which has a Panel object inside, which has a Panel object inside which has ... If the complexity and readability of this kind of structure can be justified, that's another story.

Notice that eventually all object in your VXH project are child objects of the root object which has your projects name.

The "File View" tab shows all files used by your VXH project. By default, you'll see a .prg file which is named after your Project and, when already created a form, a .prg file that is named after your initial Form object.

The files added by the "File", "Add to project", "Binary Files" menu option are listed in this tree view under the "External Binary Files" tree item.

Note that not all files used by your VXH project are listed in this tree view since some files are not intended to be changed by the user. These files include the .xfm files used to define the IDE created and resource files.

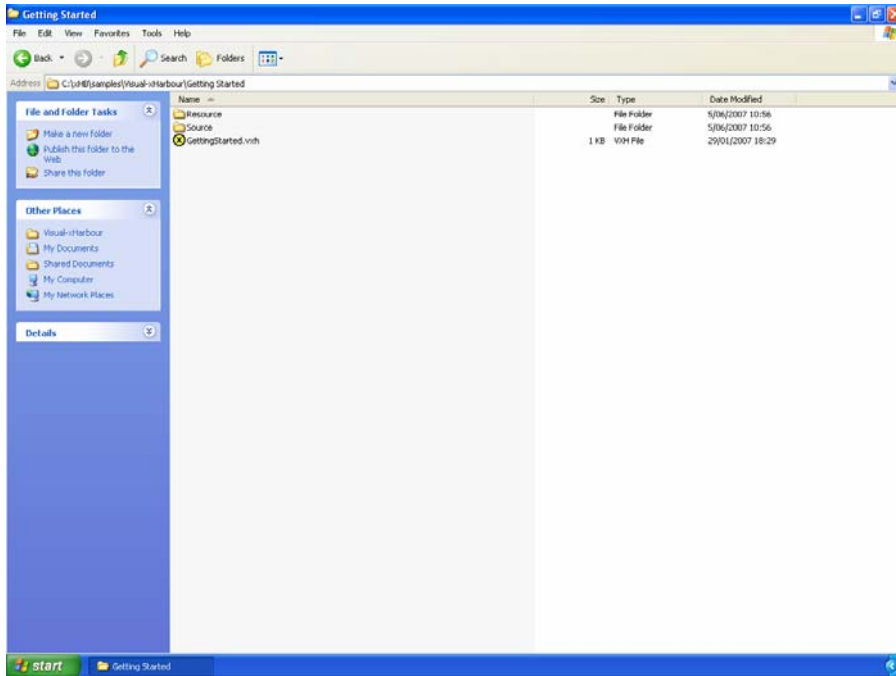
4. How Visual xHarbour Works

Now that you are a little more familiar with the VXH IDE, let's see how VXH works and how we can use it to create great looking applications.

The best way to learn how to work with VXH is ... to work with VXH. Therefore, let's use one of the existing VXH samples to learn how to manage a project and how to build it.

Let's first open the samples folder of your xHarbour Builder installation. Go to the Windows "Start" button, and open your "xHarbour Builder" folder. Now open the "Samples folder" and its subfolder "Visual-xHarbour". Look for the "Getting Started" folder and double click it.

Depending on the view chosen in your file browser, the Getting Started sample folder looks like this:



4.1. The physical structure of a VXH project

A VXH project always has a pre-defined structure. This structure contains the following files and folders:

- A “Bin” folder;
- An “Obj” folder;
- A “Resource” folder;
- A “Source” folder;
- A .vxh project file.

The “Bin” and the “Obj” folder are only added to the project’s root folder when the project was build for the first time. When creating backups of your VXH project, you can omit these folders.

Note that it’s possible to change the name of these folders in each VXH project individually by changing the “Target Folders” properties of the project.

4.1.1. The VXH file

The .vxh file is the actual project file holding information about the VXH project in general. Such information includes: project name, version number, various paths, compiler flags etc. Opening the .vxh file in a regular text editor shows all the settings of the project file.

Since this is the file that holds information about the VXH project, it's pretty clear there can be only one .vxh file in a project.

Please note that even though this file is an important one, it doesn't contain any source code or information about the content of the actual project. When creating backups, always backup the entire VXH project folder and, if necessary, linked files on other locations. As noted above, the "Bin" and "Obj" folders can be excluded by a backup.

4.1.2. The Resource folder

The "Resource" subfolder in the VXH project's root folder contains a "ProjectX.rc" and a "ProjectX.xml" file where "ProjectX" is the name of your VXH project. It's quite possible that there are other files in this folder (images for example).

The .rc file contains resource information. When your project uses linked images for example, you'll find information about that resource in this file.

Other files that can reside in the Resource folder are for example icons or bitmaps. These files are automatically placed in the Resource folder when imported in VXH.

Note:

Importing the files into the Resource folder can be discarded when you decide this isn't necessary. However, importing these files into the Resource folder is mandatory to maintain portability.

4.1.3. The Source folder

The "Source" subfolder in the VXH project's root folder contains all the source files necessary for your project. The folder contains .prg, .xfrm and .bak files.

The .prg file contains the actual code of a VXH project. For each Form object created in VXH, a "FormX.prg" file is created and stored in the Source folder.

Each .prg file, except for the manually created / imported .prg files in that folder, has a .xhm file with the same filename. The .xhm file contains automatically generated code used by VXH.

The .bak files are backup files created by VXH whenever you are modifying your .prg file(s) from within VXH.

4.1.4. The Bin folder

This folder is only visible when the project was built at least once. It contains the projects executable and .map file. When an error occurred, the Bin folder will have an error.log file with additional information about the error.

Since this folder is used to store the executable, it's the projects current directory.

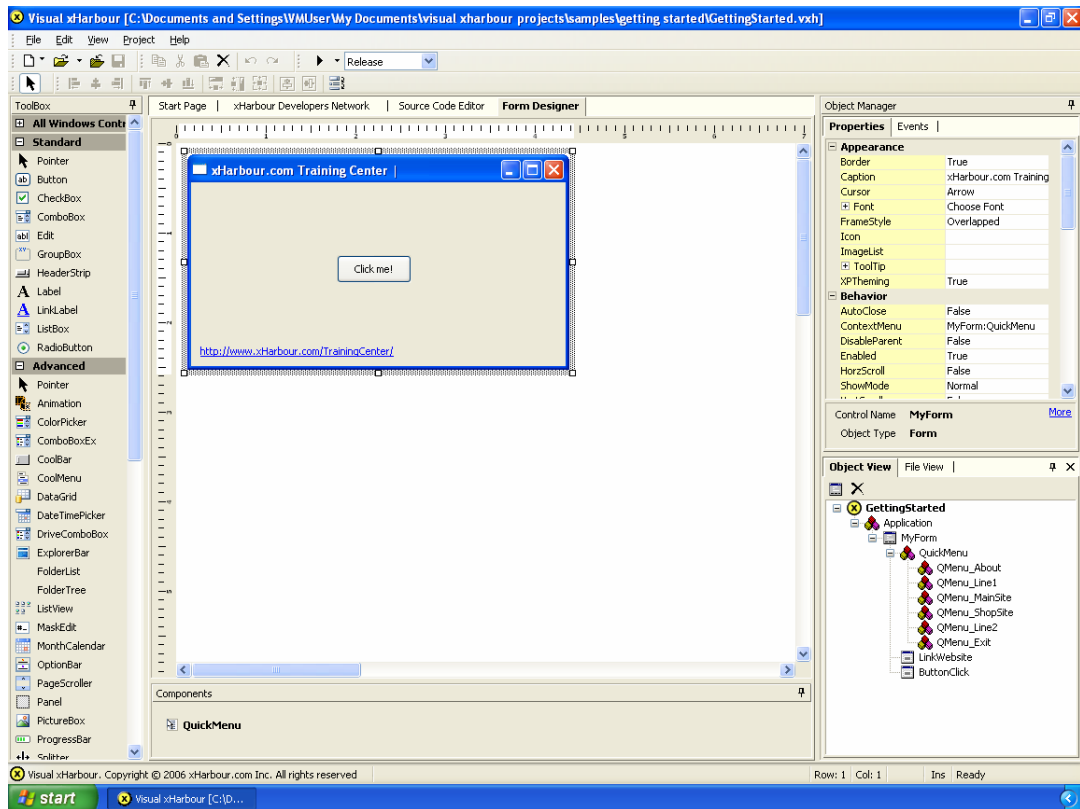
4.1.5. The Obj folder

The "Obj" folder contains all .c and .obj files produced when building your project.

4.2. Building the Getting Started sample

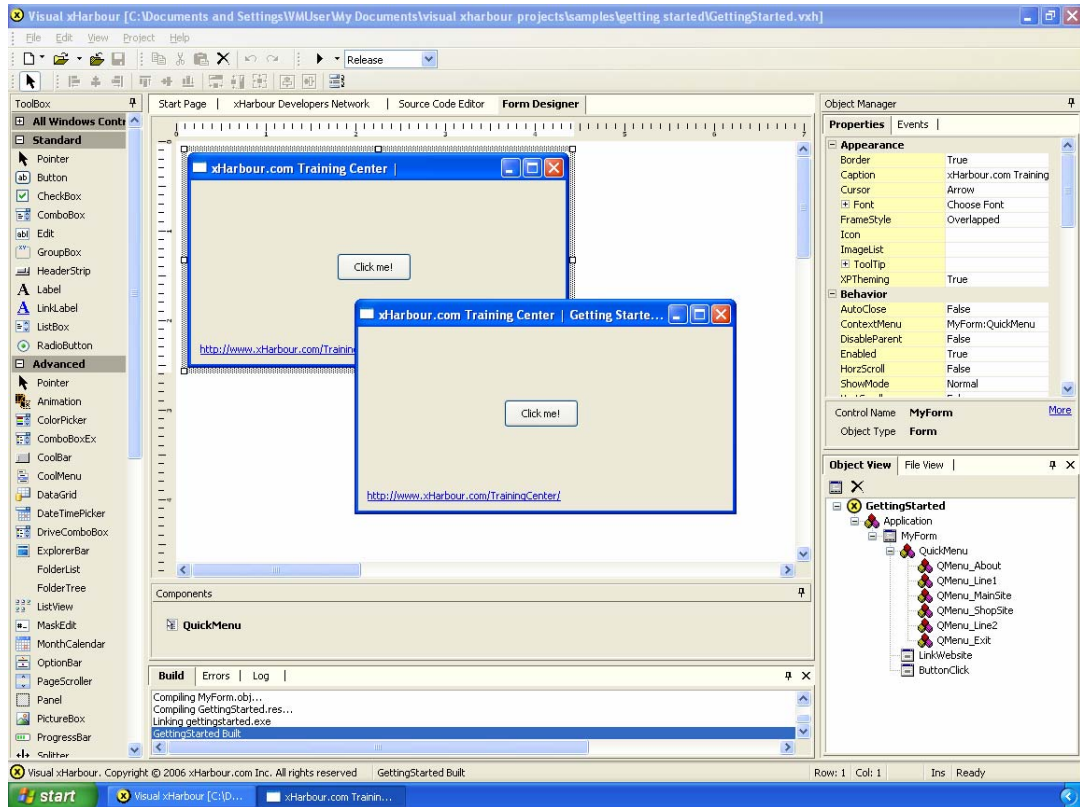
Open the Getting Started VXH project by double clicking the GettingStarted.vxh file in the Getting Started subfolder of the Visual xHarbour sample folder.

Visual xHarbour is opened and the Getting Started project is loaded into the development environment. The VXH interface will look like this:



Depending on your settings your VXH environment may differ from the one pictured above.

To simply build and run the project, make sure you select "Release" from the dropdown list next to the play button and press play. When no errors were encountered, the dialog below is shown.



Congratulations! Your first VXH project file was built successfully. It's just that simple.

5. Creating Projects With Visual xHarbour

In the previous chapter we built an existing VXH project which was pretty easy and didn't take too much time. Building an entire project from the ground up is time-consuming and intensive work you'd think? Well, it used to be. VXH allows you to create your application's GUI in a jiffy.

Thanks to a wide choice of controls in VXH's ToolBox, you have every control you dreamed of at your fingertips. By just dragging and dropping your components on the form, you'll have a professional looking application running before you know it.

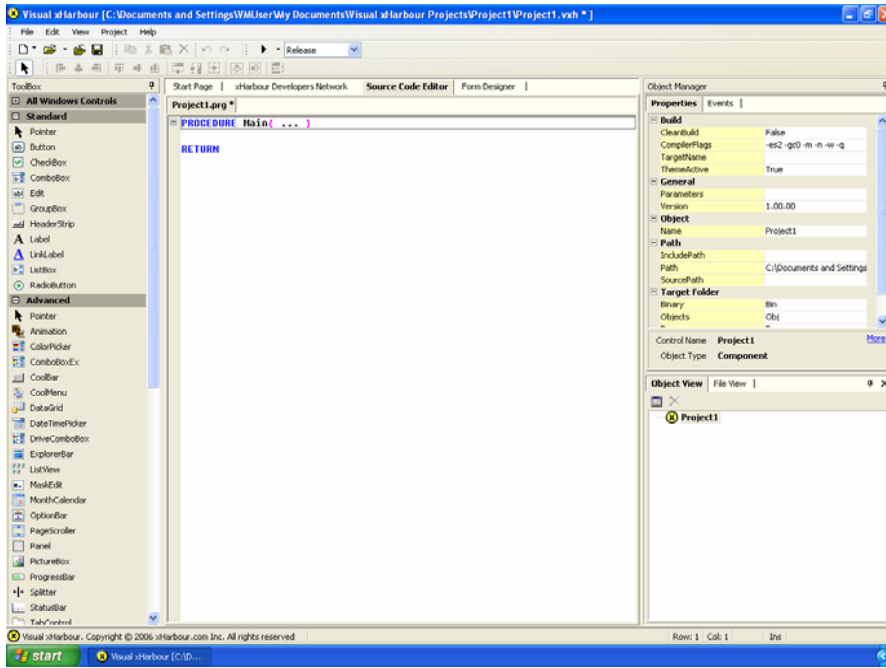
Let's create a simple application, step by step, and don't forget to watch the clock ;-)

5.1. Create a new project

The first step would be to create a new blank project file. Open VXH and select "Project" via "File", "New". Notice that you can also use the CTRL+SHIFT+N shortcut or by using the "New" icon on the toolbar.

You should see an empty environment with the Source Code Editor active. Within the Source Code Editor you'll notice a single tab. The name of the tab is "ProjectX.prg" where the X is replaced by a number.

VXH uses a counter to automatically create a unique project name. When you save and close this project file, the next project created will have the name "Project(X+1)". Needless to say, you can change the Project's name at any time without a problem.



5.2. Create a new form

Since Visual xHarbour is a tool created for designing smooth looking applications in a matter of minutes, we will be working with the Form Designer most of the time. Notice that activating the Form Designer at this time is impossible. Makes sense since we didn't create a form to design yet.

Click the "Form" menu option via "File", "New". Notice that a new form is created and the Form Designer is activated. You'll notice that the form is added to your project's Object View at the lower right corner of the VXH window. At this point, the form is named "Form1". The name of each form added is automatically generated by VXH using an incremental number.

The first form created in a VXH project is the main form. This form will be loaded automatically when you build and run the project.

Note:

Once the main form is added to the project, it can't be removed from the project.

We now have our main form in front of us. Let's start by adjusting the Form's height, and width. Select the Form object and scroll through its properties in the Object Manager. In the category "Size", you'll find both the Height and Width property. Set the Height to 300 and the Width to 420.

Note:

If you'd like to lock both the width and height of this form at runtime, you can set the `MaxHeight`, `MinHeight`, `MaxWidth` and `MinWidth` properties for the form. By assigning the same value as the initial Height and Width, it's impossible to resize the form.

We'd like to initially center the form on the screen. If the user wants to drag it elsewhere, that's no problem, but the initial position should be centered. Before you start working on how to calculate the screen dimensions, VXH has a Center property for Forms.

Scroll the Properties list upward until the "General" category slides by. Look for the Center property and set its value to true. More detailed information about manipulating objects will follow later on in this chapter.

It's about time to reveal what we are trying to create. Scroll the properties list completely upward and change the Caption property to "Hello World" (without the " off course).

At this point, we're done with the form. Now that didn't take too long, did it? Time to add our other objects.

5.3. Adding controls to the form

Let's start by having a look at what we want to do with this example. We'd like to have an input box where we can enter our name. Furthermore a list with countries to choose from and a button which displays a message "Hello <name> from <country>!" when it is clicked.

This shouldn't be that difficult. And it isn't! Let's have a look at the various steps in creating this sample.

Choose the "Edit" control from the ToolBox and click on the Form where the control should be placed (or drag it). Place the mouse on top of the Edit object (the white area) until the mouse pointer changes to a move icon (cross with arrows) and drag the Edit object to its definite place. For precision positioning, use the arrow keys.

As mentioned before, we also want to display a list with countries. A pretty simple and obvious method is by using a ListBox object. Select the ListBox object from the ToolBox and drag it to the Form. Place the ListBox so it's aligned to the Edit object.

Finally, we need a button to push when we want the popup screen to say hello. Select the Button object from the ToolBox and drop it on the form.

Note:

When adding a series of the same objects, it's handy that you can add them without re-selecting the object from the ToolBox. Disable the "ToolBox Pointer Auto Selection" feature by clicking the arrow button on the second toolbar (left side). To re-enable the auto selection, simply click the button again.

5.4. Manipulating controls

We now have some controls on our form, but how can we change the look and feel of these controls to meet our requirements? As expected, this is pretty easy.

5.4.1. Properties

Properties describe the characteristics of an object. To simplify, it defines the way an object looks. Let's look at a simple example to demonstrate.

Imagine an object car with the following properties:

- Model
- Name
- Color

The property color defines how the car looks. If the value of Color is blue, the car is blue, if the value of Color is red, the car is red. Same goes for the Model and Name property.

Now looking at our Button in VXH, the property Caption defines the text displayed on the button. If the value is "Button1", it will have "Button1" written over it, plain simple.

The best way to discover all the properties of an object is to simply browse the list and have a look at the names of the properties (and their value). Luckily, the properties have a clear name and often need no explanation of "what they do".

Back to our sample. Standard, the Edit control has its name filled in in the Caption property. Let's change this to "Enter your name". Simply select the Edit object, look for its 'Caption' property and change the value to "Enter your name".

The Edit control is a bit too narrow, let's enlarge it by increasing its Width property (for example 170). Select the Edit control and look for the Width property. Increase the value to 170.

5.4.2. Events

Events define the way an object reacts on, as the name would suspect, events in its environment. Applied to the car example mentioned earlier we could state that a car object could have the following events:

- OnStart()
- OnTurn()
- OnBreak()

The event OnBreak() defines what action(s) should be taken when the driver uses the break. First thing that comes to mind is light up the taillights and activate the break mechanism.

5.5. Adding code to the interface

We now have a nice looking GUI for our user, but until now, it's pretty boring since the project does nothing except take memory space and fills up our screen. Time to add some events to our project.

Let's start with an obvious one. We want to display a popup when the button is pressed. Therefore we'll need to browse the available events of the Button using the Object Manager. Scroll the list with events until you find the OnClick event and double-click it.

Notice that the view has switched from Form Designer to Source Code Editor. Also, the cursor is placed inside existing code which should look like the code below.

```
//-----//  
METHOD Button1_OnClick( Sender ) CLASS Form1  
  
RETURN Self
```

Let's first see what this code means. By adding an OnClick event to the button, the object Form1 (our main form) has received a new method named "Button1_OnClick()". This method is fired whenever the button is clicked.

Note:

It's possible to call this method from within another method by using the following code: `::MethodName(::ObjectName)`. In the example above, this would translate to: `::Button1_OnClick(::Button1)`.

So, whatever code we add to the method will be executed when the button is clicked. Time to add the code which pops up a message box greeting our user.

Copy / paste the code below to your method.

```
MessageBox(, "Hello " + ::Edit1:Caption + "! Nice to see you.", "Hello")
```

The `MessageBox()` function creates a default Windows message box used for displaying simple messages on the screen. You might have noticed that the `MessageBox()` function has more than two parameters. Have a look at the syntax of the `MessageBox()` function on our xHDN pages for more information and possibilities on this function.

This is basically the general idea of an event. It is triggered by an action of a user or a timer. You can add whatever code you want to these methods and use user defined functions if you'd like. The possibilities are virtually limitless.

5.6. Build / run a project

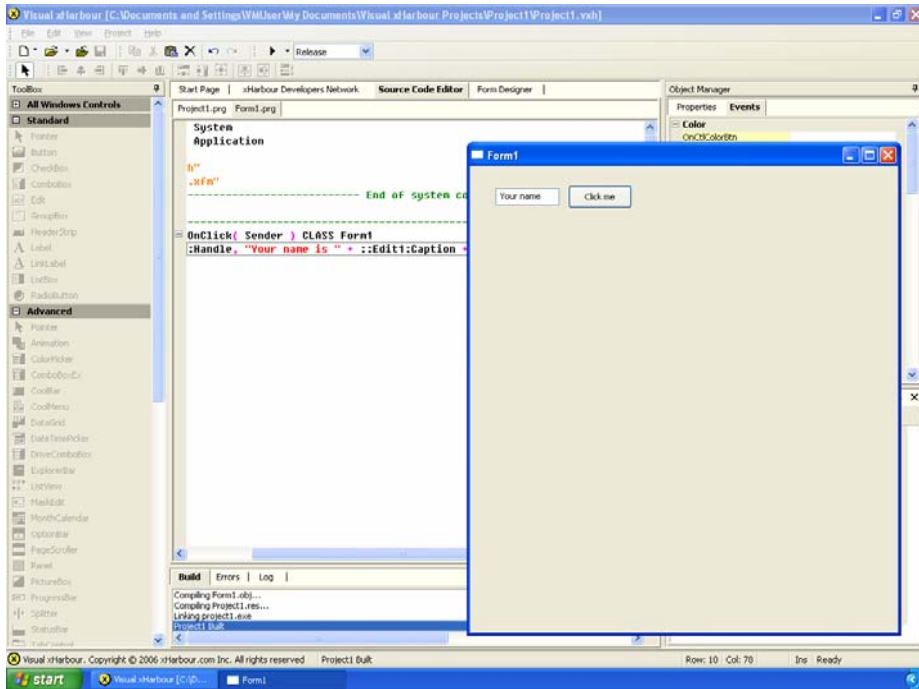
Building your project is done by choosing the "Build" menu option from the "Project" menu, or by choosing "Build" from the "Save / Build run" dropdown on the toolbar. This dropdown is visible by clicking the small triangle next to the play button.

To run the project after a successful build, choose "Run" in the menu or the dropdown list mentioned above.

Note that there's a difference when running in "Release" mode or "Debug" mode. When running in release mode, the project is run as it was released. When building in debug mode, VXH's visual xDebugger is launched and the project is debugged while running.

In most cases however, building a project is immediately followed by executing your project. That is why VXH has a Build and Run option which combines the two actions. It's executed by pressing the play button on the toolbar, or by simply hitting the F5 key on the keyboard.

Let's see what the result is when we try to build our sample. Make sure the build mode is set to "Release" (dropdown list on the upper toolbar) and hit F5 to build and run the project. It should look like this:



Click the button to test if the button works as expected. Congratulations, your first VXH application works as expected.

6. Visual xHarbour controls

When browsing through the ToolBox, you'll notice a wide range of controls. Some will seem familiar while others are brand new. Below is a list of all available controls and their properties and events.

6.1. System

6.1.1. Properties

- Color
- ColorScheme
- CurrentScheme
- Cursor
- DataDrivers
- DateTimeFormat
- DropDownStyle
- ExplorerBar
- HandShake
- ImageList
- ListBox
- ListViewGroupAlign
- LocalTime
- OleVerb
- Parity
- ProfessionalColorTable
- RootFolders
- Services
- SockProtocol
- StdIcons
- StopBits
- TextAlignment
- Time

6.1.2. Events

- GetFocus()
- GetLocalTime()
- Init()
- Update()
- UpdateColorSchemes()

6.2. Application

6.2.1. Properties

- AppIniFile
- AppIniFileName
- AutoProperty
- Children
- ClsName
- CommonHotImageList
- CommonImageList
- CommonImageListSmall
- Components
- Cursor
- Events
- Form
- hAccel
- hMainHook
- hMenuHook
- hRich20
- IdeActive
- ImageLists
- IniFile
- IniFileName
- InsKey
- Instance
- IsThemedXP
- IExit
- LocalIniFile
- LocalIniFileName
- MainForm
- MainWindow
- MainWindow
- MDIClient
- Modal
- Msg
- MultInstance
- Mutex
- Name
- oCurMenu

- OfficeXPLook
- OnFunctionKey
- OsVersion
- Params
- Parent
- Path
- Property
- Running
- SetAlternate
- SetAltfile
- SetAutopen
- SetAutorder
- SetAutoshare
- SetBell
- SetCancel
- SetConfirm
- SetCount
- SetDateFormat
- SetDbflockscheme
- SetDebug
- SetDecimals
- SetDefault
- SetDeleted
- SetDevice
- SetDircase
- SetDirseparator
- SetEol
- SetEpoch
- SetErrorlog
- SetErrorloop
- SetEscape
- SetEventmask
- SetExact
- SetExclusive
- SetExit
- SetExtra
- SetExtrafile
- SetFilecase
- SetFixed
- SetForceopt
- SetHardcommit

- SetIdlerepeat
- SetInsert
- SetLanguage
- SetMargin
- SetMblocksize
- SetMcenter
- SetMessage
- SetMfileext
- SetOptimize
- SetOutputsafety
- SetPath
- SetPrinterjob
- SetPrintfile
- SetScrollbreak
- SetSoftseek
- SetStrictread
- SetTrace
- SetTracefile
- SetTracestack
- SetTrimfilename
- SetTypeahead
- SetUnique
- SetVideomode
- SetWrap
- Skin
- SysColors
- TempDir
- ThemeActive
- UserVariables
- Version

6.2.2. Events

- AddAccelerators()
- Close()
- CoolMenuProc()
- Create()
- DelAccelerators()
- Exit()
- Init()
- IsThemeActive()
- LoadIcon()

- MainHook()
- OnExit()
- RestorePrevInstance()
- Run()
- Set()
- TranslateAccelerator()
- Yield()

6.3. Form

6.3.1. Properties

- AutoClose
- CaptionBar
- ControlId
- DlgModalFrame
- FrameStyle
- HorzScroll
- Icon
- ImageList
- IsMDIContainter
- MaskBitmap
- MaxHeight
- MaximizeBox
- MaxWidth
- MDIChild
- MDIClient
- MDIClient
- MdiContainer
- MinHeight
- MinimizeBox
- MinWidth
- Modal
- Opacity
- ShowMode
- SysMenu
- TabOrder
- ThickFrame
- TopMost
- UserVariables
- VertScroll

- Visible

6.3.2. Events

- Create()
- CenterWindow()
- GetNextControllId()
- Init()
- Maximize()
- MdiActivate()
- MdiCascade()
- MdiClose()
- MdiDestroy()
- MdiGetActive()
- MdilconArrange()
- MdiMaximize()
- MdiNext()
- MdiRestore()
- MdiTileHorizontal()
- MdiTileVertical()
- Minimize()
- Restore()
- SetFormIcon()
- SetOpacity()
- Show()

6.4. Button

6.4.1. Properties

- AllowClose
- AllowUndock
- Border
- Checked
- DefaultButton
- DrawFocus
- DrawTheme
- Group
- ImageAlign
- ImageIndent
- ImageIndex
- ImgInst

- OwnerDraw
- Theme
- xState

6.4.2. Events

- Click()
- Create()
- DrawFrame()
- GetState()
- Init()
- IsPushed()
- OnCtlColorBtn()
- OnCtlColorStatic()
- OnParentDrawItem()
- OnLButtonDownClk()
- Push()
- Release()
- SetBackColor()
- SetFocus()
- SetForeColor()
- SetImageIndex()
- SetState()
- SetStyle()

6.5. CheckBox

6.5.1. Properties

- Border
- Check_Styles
- CheckStyle
- DefaultButton
- Group
- ImageIndex
- ImageList
- State
- States
- Theme
- Transparent

6.5.2. Events

- Check()
- Checked()
- Create()
- DrawFrame()
- Init()
- Indeterminate()
- OnCtlColorStatic()
- OnDestroy()
- OnEraseBkGnd()
- OnParentNotify()
- SetCheckStyle()
- SetParent()
- SetState()
- UnCheck()

6.6. ComboBox

6.6.1. Properties

- AllowClose
- AllowUndock
- AutoEditHorzScroll
- AutoHorzScroll
- Border
- CurSel
- DisableNoScroll
- DropDownStyle
- Enabled
- FitToolBar
- HasStrings
- HorzScroll
- ImageIndex
- ItemHeight
- LowerCase
- NoIntegralHeight
- OemConvert
- OnCBNKillFocus
- OnCBNSelEndCancel
- OnCBNSelEndOk
- OwnerDrawFixed

- OwnerDrawVariable
- SelectionHeight
- Sort
- UpperCase
- VertScroll

6.6.2. Events

- AddItem()
- AddString()
- Create()
- DeleteString()
- Dir()
- DrawFrame()
- Enable()
- FindString()
- FindStringExact()
- GetCount()
- GetCurSel()
- GetDroppedState()
- GetDroppedWidth()
- GetEditSel()
- GetHorizontalExtent()
- GetInfo()
- GetItemData()
- GetItemHeight()
- GetLBText()
- GetLBTextLen()
- GetLocale()
- GetSelString()
- GetString()
- GetTopIndex()
- HideDropDown()
- Init()
- InitStorage()
- InsertString()
- OnParentCommand()
- OnWindowPosChanged()
- ResetContent()
- SelectString()
- SetCurSel()
- SetDropDownStyle()

- SetExtendedUi()
- SetHorizontalExtent()
- SetItemData()
- SetItemHeight()
- SetLocale()
- SetTopIndex()
- ShowDropDown()

6.7. CursorComboBox

6.7.1. Properties

- Cursors

6.7.2. Events

- Create()
- Init()
- OnParentDrawItem

6.8. FontComboBox

6.8.1. Properties

- Fonts
- Owner

6.8.2. Events

- Create()
- Init()
- OnParentDrawItem

6.9. Edit

6.9.1. Properties

- Alignments
- AllowClose
- AllowUnDock
- AutoHScroll
- AutoVScroll
- BackSysColor

- Border
- Button
- ButtonAction
- ButtonPushed
- Case
- CaseTypes
- Enabled
- ForeSysColor
- HorzScroll
- ImageIndex
- ImageList
- MultiLine
- NoHideSel
- OemConvert
- Password
- ReadOnly
- SelBackColor
- SelfForeColor
- Transparent
- VertScroll
- WantReturn
- Modified

6.9.2. Events

- CanUndo()
- CharFromPos()
- Create()
- EmptyUndoBuffer()
- Enable()
- FmtLines()
- GetCueBanner()
- GetEditRect()
- GetFirstVisibleLine()
- GetHandle()
- GetImeStatus()
- GetLimitText()
- GetLine()
- GetLineCount()
- GetMargins()
- GetModify()
- GetPasswordChar()

- GetSel()
- GetThumb()
- GetWordBreakProc()
- HideBalloonTip()
- Init()
- LimitText()
- LineFromChar(n)
- LineIndex()
- LineLength()
- LineScroll()
- OnCtlColorEdit()
- OnCtlColorStatic()
- OnDestroy()
- OnEraseBkGnd()
- OnHorzScroll()
- OnHScroll()
- OnKeyDown()
- OnNCCalcSize()
- OnNCHitTest()
- OnNCLButtonDown()
- OnNCLButtonUp()
- OnNCPaint()
- OnParentCommand()
- OnVertScroll()
- OnVScroll()
- PosFromChar(x,y)
- ReplaceSel()
- Scroll()
- ScrollCaret()
- SetAlignment()
- SetCase()
- SetCueBanner()
- SetHandle()
- SetImeStatus()
- SetLimitText()
- SetMargins()
- SetModify()
- SetPasswordChar()
- SetReadOnly(lSet)
- SetRect()
- SetRectNP()

- SetSel()
- SetSelColor()
- SetTabStops()
- SetWordBreakProc()
- ShowBalloonTip()
- Undo()
- SetParent()

6.10. GroupBox

6.10.1. Properties

- ForeSysColor
- ImageIndex
- ImageList
- Theme
- Transparent

6.10.2. Events

- Create()
- GetSysColor()
- Init()
- OnEraseBkGnd()
- OnPaint()
- OnSize()

6.11. HeaderStrip

6.11.1. Properties

- AllowClose
- AllowUnDock
- ImageList
- ImageMargin
- OnBeginDrag
- OnBeginTrack
- OnEndDrag
- OnFilterChange
- OnGetDisplInfo
- OnItemChanged
- OnItemChanging

- OnItemClick
- OnItemDbClick
- OnTrack

6.11.2. Events

- Create()
- DeleteItem()
- DrawFrame()
- GetItemRect()
- Init()
- InsertItem()
- OnHScroll()
- OnParentNotify()
- OnSize()
- SetArrow()
- SetImageList()
- SetImageMargin()
- SetItemCaption()
- SetItemWidth()

6.12. HeaderItem

6.12.1. Properties

- Alignment
- Alignments
- AllowDrag
- AllowResize
- Anchor
- Caption
- Children
- ClsName
- Dock
- EventHandler
- Events
- Form
- Height
- hWnd
- ImageIndex
- IsContainer
- Left

- Name
- Owner
- Parent
- Position
- Top
- Width
- xName

6.12.2. Events

- Create()
- D GetRectangle()
- Destroy()
- GetChildFromPoint()
- GetRect()
- GetRectangle()
- GetSize()
- Init()
- IsWindowVisible()
- MoveWindow()
- OnParentMove()
- Refresh()
- SetAlignment()
- SetCaption()
- SetHeight()
- SetImageIndex()
- SetWidth()
- SetWindowPos()
- UpdateWindow()

6.13. Label

6.13.1. Properties

- AllowClose
- AllowUnDock
- Border
- CenterText
- ImageIndex
- ImageList
- Noprefix
- RightAlign

- Simple
- Sunken

6.13.2. Events

- DelEntry()
- DelSection()
- Flush()
- GetEntries()
- GetSections()
- Read()
- ReadDate()
- ReadInteger()
- ReadLogical()
- ReadNumber()
- ReadString()
- Write()
- WriteArray()
- WriteDate()
- WriteInteger()
- WriteLogical()
- WriteNumber()
- WriteString()
- ReadArray()

6.14. LinkLabel

6.14.1. Properties

- ActiveLinkColor
- AllowClose
- AllowUnDock
- AutoSize
- Cursor
- DefaultButton
- FocusRect
- ForeColor
- Group
- LinkColor
- LinkSysColor
- LinkVisited
- OwnerDraw

- SelBackColor
- Url
- VisitedColor

6.14.2. Events

- Create()
- Init()
- OnEraseBkGnd()
- OnKillFocus()
- OnLButtonDown()
- OnLButtonUp()
- OnNCDestroy()
- OnSetFocus()
- SetLinkColor()
- SetParent()
- SetSelColor()
- SetVisitedColor()
- SetWindowText()

6.15. ListBox

6.15.1. Properties

- Border
- CurSel
- DisableNoScroll
- Enabled
- ExtendedSel
- HasStrings
- HorzScroll
- ImageIndex
- ImageList
- IntegralHeight
- MultiColumn
- NoRedraw
- Notify
- OnChange
- OwnerDraw
- OwnerDraw_Styles
- Sort
- UseTabStops

- VertScroll
- WantKeyboardInput

6.15.2. Events

- AddFile()
- AddItem()
- AddString()
- Create()
- DeleteItem
- DeleteString()
- Dir()
- FindExact()
- FindString()
- GetAnchorIndex()
- GetCaretIndex()
- GetCount()
- GetCurSel()
- GetHorizontalExtent()
- GetItemData()
- GetItemHeight()
- GetItemRect()
- GetItemText()
- GetLocale()
- GetSel()
- GetSelCount()
- GetSelItems()
- GetString()
- GetText()
- GetTextLen()
- GetTopIndex()
- Init()
- InitStorage()
- InsertString()
- ItemFromPoint()
- OnCtlColorListBox()
- OnDbcClick()
- OnErrSpace()
- OnGetDlgCode()
- OnHScroll()
- OnLBNKillFocus()
- OnLBNSetFocus()

- OnParentCommand()
- OnSelChange()
- OnVScroll()
- ResetContent()
- SelectString()
- SellItemRange()
- SellItemRangeEx()
- SetAnchorIndex()
- SetCaretIndex()
- SetColumnWidth()
- SetCount()
- SetCurSel()
- SetDrawStyle()
- SetHorizontalExtent()
- SetIntegralHeight()
- SetItemData()
- SetItemHeight()
- SetLocale()
- SetSel()
- SetTabStops()
- SetTopIndex()
- Enable()

6.16. RadioButton

6.16.1. Properties

- Border
- Checked
- DefaultButton
- Group
- ImageIndex
- ImageIndex
- ImageList
- ImageList
- OwnerDraw
- State
- States
- Theme

6.16.2. Events

- Create()
- GetState()
- Init()
- OnEraseBkGnd()
- OnParentNotify()
- SetState()

6.17. Animation

6.17.1. Properties

- AutoPlay
- Border
- Centered
- ClipChildren
- ClipSiblings
- FileName
- FromFrame
- ImageIndex
- ImageList
- Repeat
- SysAnimations
- System
- TabStop
- ToFrame
- Transparent

6.17.2. Events

- Close()
- Create()
- Init()
- OnCtlColorStatic()
- OnEraseBkGnd()
- Open()
- Play()
- Seek()
- Stop()

6.18. ColorPicker

6.18.1. Properties

- AllowCustomColor
- AllowSystemDefault
- ColorSelected
- Colors
- Custom
- SysDefault

6.18.2. Events

- Create()
- Init()
- OnParentCommand()
- OnParentDrawItem
- SelectColor()

6.19. ComboBoxEx

6.19.1. Properties

- ClientEdge
- hControl
- ImageList

6.19.2. Events

- AddItem()
- AddString()
- Create()
- Init()
- OnMove()
- OnSize()
- SetHorizontalExtent()
- SetImageList()
- SetItemHeight()

6.20. CoolBar

6.20.1. Properties

- Repaint

- Bands
- IDestroyed
- hWndBand
- rbi
- nmr
- ImageList
- BackColor
- Vertical
- AllowUndock
- AllowClose

6.20.2. Events

- Init()
- GetBandInfo()
- GetBandRect()
- SetImageList()
- GetBkColor()
- SetTextColor()
- SetBkColor()
- GetHeight()
- OnParentNotify()
- Create()
- OnMouseMove()
- BandFromhWnd()
- OnNCDestroy()
- UpdateSize()
- FindRemoveBand()

6.21. CoolBarBand

6.21.1. Properties

- AllowUndock
- Anchor
- BackColor
- Backup
- BandChild
- Break
- Caption
- Chevron
- ClsName

- Dock
- FixedSize
- Grippers
- Height
- Image
- Index
- ICreated
- Left
- MinHeight
- MinWidth
- Name
- oStruct
- Parent
- ToolTip
- Top
- Width

6.21.2.

Events

- Create()
- Delete()
- Destroy()
- GetBandInfo()
- GetRect()
- GetRectangle()
- Hide()
- Init()
- InvalidateRect()
- OnChevronPushed()
- OnParentMove()
- ReflectColor()
- RemoveChild()
- RestoreChild()
- SaveChild()
- SetCaption()
- SetChevron()
- SetChild()
- SetGrippers()
- SetHeight()
- SetImage()
- SetMinHeight()
- SetMinWidth()

- SetStyle()
- SetWidth()
- Show()
- IsWindowVisible()

6.22. CoolMenu

6.22.1. Properties

- Editor
- ForeColor
- hBackColor
- hMenuHook
- HotImageList
- ImageList

6.22.2. Events

- CheckMenuItem()
- Create()
- D OnKeyDown()
- DeleteMenuItem()
- EnableMenuItem()
- GetItem()
- GetItemCount()
- Init()
- OnGetDlgCode()
- OnLButtonDown()
- OnMenuChar()
- OnParentCommand()
- OnParentSysCommand()
- OnSysChar()
- OnSysKeyDown()
- OnSysKeyUp()
- OnNCDestroy()

6.23. CoolMenuItem

6.23.1. Properties

- AutoSize
- Check

- CheckGroup
- Components
- DropDown
- Font
- ImageIndex
- Index
- Item
- Menu
- MixedButtons
- Name
- SetChildren
- ShowText
- ToolTip
- WholeDropDown
- Width

6.23.2. Events

- Create()
- Init()

6.24. DataGrid

6.24.1. Properties

- AutoHorzScroll
- AutoUpdate
- AutoVertScroll
- ColCount
- ColPos
- Columns
- ConvertOem
- DataSource
- ExtVertScrollBar
- FullRowSelect
- GridColor
- GridSysColor
- HeaderHeight
- HighlightColor
- HighlightSysColor
- HighlightTextColor
- HighlightTextSysColor

- HitBottom
- HitTop
- HorzScrollPos
- ImageList
- ItemHeight
- Record
- RowCount
- RowCountUsable
- RowCountVisible
- RowHeight
- RowPos
- ShadowRow
- ShowGrid
- ShowHeaders
- VertScrollPos

6.24.2.

Events

- ArrowLeft()
- ArrowRight()
- AutoAddColumns()
- Create()
- DeleteColumn()
- Disable()
- Down()
- Enable()
- End()
- GetControl()
- GetItemRect()
- GoBottom()
- GoTop()
- Home()
- Init()
- OnBeginDrag()
- OnBeginTrack()
- OnChar()
- OnClick()
- OnColChanged()
- OnColChanging()
- OnEndDrag()
- OnEraseBkGnd()
- OnGetDlgCode()

- OnHeaderClick()
- OnHorzScroll()
- OnItemChanged()
- OnKeyDown()
- OnKillFocus()
- OnLButtonDblClk()
- OnLButtonDown()
- OnMouseMove()
- OnNCDestroy()
- OnPaint()
- OnParentSysCommand()
- OnRowChanged()
- OnRowChanging()
- OnSetFocus()
- OnSize()
- OnTimer()
- OnVertScroll()
- PageDown()
- PageUp()
- Refresh()
- Skip()
- Up()
- Update()
- UpdateRow()

6.25. GridColumn

6.25.1. Properties

- Active
- Alignment
- Alignments
- AllowDrag
- AllowSize
- AutoEdit
- BackColor
- BackSysColor
- Caption
- Cargo
- Children
- ClsName

- Control
- ControlAccessKey
- ControlAction
- ControlAlign
- ControlBackColor
- ControlForeColor
- ControlHeight
- ControlHide
- ControlMultiline
- ControlObject
- ControlSaveArrayPos
- ControlStatus
- ControlValid
- ControlWidth
- Data
- EventHandler
- Events
- ForeColor
- ForeSysColor
- Form
- HeaderImageIndex
- hWnd
- ImageIndex
- IsContainer
- Locked
- MDIContainer
- Name
- OnSave
- Owner
- Parent
- Picture
- Representation
- ShowControls
- TabOrder
- ToolTip
- Type
- Width
- xHeight
- xLeft
- xPosition
- xTop

6.25.2. Events

- Create()
- Destroy()
- GetChildFromPoint()
- GetEditValue()
- GetRect()
- GetRectangle()
- GetSize()
- HasMessage()
- Init()
- InvalidateRect()
- IsWindow()
- IsWindowVisible()
- MoveWindow()
- Refresh()
- SetAlignment()
- SetCaption()
- SetColor()
- SetHeaderImageIndex()
- SetImageIndex()
- SetOrderArrow()
- SetPosition()
- SetRepresentation()
- SetSize()
- SetWidth()
- SetWindowPos()
- UpdateWindow()

6.26. DateTimePicker**6.26.1. Properties**

- Border
- Caption
- CustomFormat
- Date
- Enabled
- Format
- MaxDate
- MaxRange

- MaxTime
- MinDate
- MinRange
- MinTime
- OnDTNCloseUp
- OnDTNDateTimeChange
- OnDTNDropDown
- OnDTNFormat
- OnDTNFormatQuery
- OnDTNKillFocus
- OnDTNSetFocus
- OnDTNUserString
- OnDTNWMKeyDown
- Parse
- SmallCaption
- Time
- UpDown

6.26.2. Events

- Create()
- Enable()
- GetCalendarColor()
- GetCalendarFont()
- GetCalendarHandle()
- GetRange()
- GetSystemTime()
- Init()
- OnCloseUp()
- OnDateTimeChange()
- OnDropDown()
- OnFormat()
- OnFormatQuery()
- OnKillFocus()
- OnParentNotify()
- OnSetFocus()
- OnUserString()
- OnWMKeyDown()
- SetBkColor()
- SetCalendarColor()
- SetCalendarFont()
- SetCustomFormat()

- SetFormat()
- SetMonthBackColor()
- SetRange()
- SetSystemTime()
- SetTextColor()
- SetTitleBackColor()
- SetTitleTextColor()
- SetTrailingTextColor()

6.27. DriveComboBox

6.27.1. Properties

- Drives

6.27.2. Events

- Create()
- Init()
- OnParentDrawItem

6.28. ExplorerBar

6.28.1. Properties

- BackColor
- ForeColor
- ImageList

6.28.2. Events

- Create()
- Init()
- OnDestroy()
- OnEraseBkGnd()
- OnPaint()
- OnSize()
- OnThemeChanged()
- OnVertScroll()

6.29. Expando

6.29.1. Properties

- Anchor
- BackColor
- Caption
- Dock
- Expanded
- ForeColor
- HeaderHeight
- ImageList
- Left
- PaneHeight
- Special
- Top
- Width

6.29.2. Events

- Create()
- Expand()
- GetRectangle()
- Init()
- OnEraseBkGnd()
- OnMouseLeave()
- OnMouseMove()
- OnPaint()
- OnParentDrawItem()
- OnSize()
- OnThemeChanged()
- SetExpand()

6.30. FolderList

6.30.1. Events

- Create()
- Init()
- OnParentNotify()
- Populate()
- SetFolder()

6.31. FolderTree

6.31.1. Properties

- InitialPath
- RootFolder

6.31.2. Events

- Create()
- GetPath()
- Init()
- OnKeyDown()
- OnParentNotify()
- Update()

6.32. ListView

6.32.1. Properties

- AlignLeft
- AlignMask
- AlignTop
- AutoArrange
- BackColor
- Columns
- DataSource
- EditLabels
- FlatScrollBar
- ForeColor
- FullRowSelect
- GridLines
- Groups
- ImageList
- ImageListSmall
- Items
- LvExStyle
- nCurRec
- NoColumnHeader
- NoLabelWrap
- NoScroll
- NoSortHeader
- OwnerData

- OwnerDrawFixed
- View_Styles
- ViewStyle

6.32.2. Events

- AddColumn()
- AutoAddColumns()
- Create()
- DeleteColumn()
- EnsureVisible()
- GetCurPos()
- GetVirtualValue()
- Init()
- InsertItem()
- InsertItems()
- OnParentNotify()
- OnSize()
- ResetContent()
- SetDataSource()
- SetImageList()
- SetImageListSmall()
- SetItemText()
- SetLVExStyle()
- SetViewStyle()

6.33. ListViewColumn

6.33.1. Properties

- Align
- Caption
- ImageIndex
- Index
- Lvcolumn
- Name
- Parent
- Width

6.33.2. Events

- Create()
- Destroy()

- Init()

6.34. ListViewGroup

6.34.1. Properties

- Alignment
- Caption
- ICreated

6.34.2. Events

- AddItem()
- Create()
- GetRectangle()
- Init()
- SetCaption()
- SetAlignment()

6.35. MaskEdit

6.35.1. Properties

- Caption
- IInvalid
- NoEdit
- NoOverStrike
- oGet
- OnCharacter
- OnKey
- Picture
- PostBlock
- PreBlock
- ReadOnly
- VarGet

6.35.2. Events

- Create()
- Init()
- OnChar()
- OnClear()
- OnCopy()

- OnCut()
- OnGetDlgCode()
- OnKeyDown()
- OnKillFocus()
- OnPaste()
- OnSetFocus()
- OnUndo()
- OnUserMsg()
- SetGetProp()
- VarPut()

6.36. MonthCalendar

6.36.1. Properties

- BackColor
- Border
- Caption
- Date
- DayState
- ForeColor
- MultiSelect
- NoToday
- NoTodayCircle
- TitleBackColor
- TitleForeColor
- Today
- TrailingTextColor
- WeekNumbers

6.36.2. Events

- Create()
- GetCalendarColor()
- Init()
- OnParentNotify()
- SetCalendarColor()
- SetCurSel()
- SetRange()
- SetToday()

6.37. OptionBar

6.37.1. Properties

- HotImageList
- ImageList
- IsContainer
- ICreated
- IDown
- PagerSize
- xButtonCheckColor
- xButtonCheckSolid
- xCheckGroup
- xImageList
- xList

6.37.2. Events

- CheckButton()
- Create()
- GetChildFromPoint()
- Init()
- IsButtonChecked()
- OnSize()
- UnDock()

6.38. OptionBarButton

6.38.1. Properties

- AutoSize
- Check
- CheckGroup
- ContextMenu
- DropDown
- IsContainer
- Position
- Separator
- ShowText
- WholeDropDown
- Width
- Wrap

6.38.2. Events

- GetRect()
- Init()

6.39. PageScroller

6.39.1. Properties

- AllowClose
- AllowUnDock
- ButtonHeight
- ImageIndex
- ImageList
- Page_Positions
- PageChild
- Position
- ScrollUnits

6.39.2. Events

- Create()
- Init()
- OnParentNotify()
- RecalSize()
- SetButtonHeight()
- SetChild()
- SetPosition()

6.40. Panel

6.40.1. Properties

- Border
- ImageIndex
- ImageList
- oLastFocus
- SmallCaption

6.40.2. Events

- Create()
- Init()
- OnSetFocus()

6.41. PictureBox

6.41.1. Properties

- CenterImage
- ImageIndex
- ImageList
- ImageName
- Instance
- KeepAspectRatio
- PictureHeight
- PictureWidth
- pPicture
- ResourceName
- Stretch
- Type

6.41.2. Events

- Init()
- Create()
- Draw()
- Kill()
- OnPaint()
- SetImageName()
- OnGetDlgCode()
- OnEraseBkGnd()
- OnSize()
- Update()

6.42. ProgressBar

6.42.1. Properties

- AllowClose
- AllowUnDock
- BackColor
- Border
- ForeColor
- ImageIndex
- ImageList

- MaxRange
- MinRange
- Position
- PropParent
- Range
- Smooth
- Step
- Vertical

6.42.2. Events

- Create()
- Init()
- SetBarColor()
- SetBkColor()
- SetPosition()
- SetRange()
- SetStep()
- StepIt()

6.43. Splitter

6.43.1. Properties

- AllowClose
- AllowUndock
- Anchor
- BackColor
- Border
- Caption
- ClientEdge
- ClipChildren
- ClipSiblings
- ContextMenu
- Cursor
- Dock
- Enabled
- Font
- ForeColor
- Height
- HorzScroll
- hWinDC

- InvRect
- IDown
- Left
- ISizing
- MaxBottom
- MinTop
- MoveWhileDrag
- NoParentAlign
- Noresize
- Owner
- Position
- SmallCaption
- StaticEdge
- TabOrder
- TabStop
- ToolTip
- Top
- TopMost
- Transparent
- VertScroll
- Visible
- Weight
- Width
- XPTheming

6.43.2. Events

- Create()
- GetSizes()
- Init()
- OnEraseBkGnd()
- OnLButtonDown()
- OnLButtonUp()
- OnMouseMove()
- OnParentMove()
- SetSizes()
- SplitOn()

6.44. StatusBar

6.44.1. Properties

- AllowClose
- AllowUndock
- Anchor
- BackColor
- Border
- Caption
- ClientEdge
- ClipChildren
- ClipSiblings
- Dock
- Enabled
- ImageIndex
- ImageList
- Left
- nCurPanelTip
- SmallCaption
- StaticEdge
- TabStop
- Top
- Transparent
- Width

6.44.2. Events

- Create()
- GetPanelRect()
- Init()
- OnMouseMove()
- SetImageIndex()
- SetPanels()

6.45. StatusBarPanel

6.45.1. Properties

- AllowClose
- AllowUndock
- Anchor
- BackColor
- Border
- Caption

- ClientEdge
- ClipChildren
- ClipSiblings
- ContextMenu
- Cursor
- Dock
- Enabled
- Font
- ForeColor
- Height
- hWnd
- ImageIndex
- Index
- Left
- SmallCaption
- StaticEdge
- TabOrder
- TabStop
- ToolTip
- ToolTipText
- ToolTipTitle
- Top
- Transparent
- Visible
- Width
- XPTheming

6.45.2. Events

- Create()
- DeferWindowPos()
- Destroy()
- GetRect()
- GetRectangle()
- GetWindowRect()
- Init()
- MoveWindow()
- SetImageIndex()
- SetText()
- SetWindowPos()
- Update()

6.46. TabControl

6.46.1. Properties

- BackParent
- BoldFont
- BoldSelection
- Bottom
- CloseBtn
- ColorInactiveHeader
- CurSel
- FocusFont
- FocusNever
- Frame
- ImageList
- IFlat
- MultiLine
- OnSelChanged
- OwnerDrawFixed
- PinBtn
- VertFocusFont
- VertFont
- Vertical

6.46.2. Events

- AdjustRect()
- AlignBottom
- AlignLeft
- AlignRight
- AlignTop
- Create()
- DeleteAllTabs()
- DeleteTab()
- DeselectAll()
- GetCurFocus()
- GetCurSel()
- GetExtendedStyle()
- GetItem()
- GetItemCount()
- GetItemRect(nItem)
- GetItemText()

- GetObj()
- GetRowCount()
- GetTabPosByName()
- GetToolTips()
- GetUnicodeFormat()
- Hide()
- HighlightItem()
- HitTest()
- Init()
- InsertTab()
- OnEraseBkGnd()
- OnHScroll()
- OnMove()
- OnNCDestroy()
- OnParentNotify()
- OnSysKeyDown()
- OnTabChanged()
- OnTabChanging()
- OnTabInsert()
- OnWindowPaint()
- ReDock()
- Refresh()
- RemoveImage()
- SetCurFocus()
- SetCurSel()
- SetExtendedStyle()
- SetImageList()
- SetItem()
- SetItemExtra()
- SetItemSize()
- SetItemText()
- SetItemValues()
- SetMinTabWidth()
- SetPadding()
- SetPos()
- SetToolTips()
- SetUnicodeFormat()
- Show()
- UnDock()

6.47. TabPage

6.47.1. Properties

- Action
- AutoClose
- Border
- Caption
- CaptionBar
- Center
- ClientEdge
- ClipChildren
- ClipSiblings
- ColorMode
- ControlParent
- DisableParent
- DlgModalFrame
- FrameStyle
- Height
- Icon
- Icon
- ImageIndex
- Index
- ItemSize
- LastFocus
- Left
- MaximizeBox
- MdiContainer
- MinimizeBox
- Modal
- Opacity
- ShowMode
- StaticEdge
- SysMenu
- TabOrder
- TabOrder
- TabStop
- ThickFrame
- Top
- TopMost
- Transparent

- Visible
- Width

6.47.2. Events

- Create()
- Delete()
- Destroy()
- Init()
- OnCtlColorDlg()
- OnGetDlgCode()
- OnHotKey()
- Select()
- SetCaption()
- SetImageIndex()
- Show()

6.48. TabCloseButton

6.48.1. Events

- DrawClose()
- Init()
- OnEraseBkGnd()
- OnLButtonDown()
- OnMouseLeave()
- OnMouseMove()
- OnParentDrawItem()

6.49. TabPinButton

6.49.1. Events

- DrawPin()
- Init()
- OnClick()
- OnEraseBkGnd()
- OnLButtonDown()
- OnMouseLeave()
- OnMouseMove()
- OnParentDrawItem()

6.50. ToolBar

6.50.1.

Properties

- aBitmaps
- aDeleted
- altems
- AllowClose
- AllowUnDock
- aReset
- Banner
- Border
- ButtonCheckColor
- ButtonCheckSolid
- Caption
- Chevron
- ChevronRect
- ClientEdge
- ClipChildren
- ClipSiblings
- DrawArrows
- ExtStyle
- Flat
- hBmpBkGnd
- HideClippedButtons
- HorzPadding
- HotImageList
- hPrevWndMenu
- hWndHook
- hWndMenu
- ImageList
- IDestroyed
- List
- IKeyboard
- IToolMenu
- MenuWindow
- MenuWnd
- MixedButtons
- nItem
- NoThemeCheck
- nPressed

- nPushed
- oMenu
- ParProc
- pRect
- PrevFoc
- PrevMenuProc
- PushPos
- RowCount
- SelPopup
- ShowMDIButtons
- StaticEdge
- TabStop
- ToolTips
- Transparent
- VertPadding

6.50.2.

Events

- AddBitmap()
- AddControl()
- AddCoolMenuItem()
- CheckButton()
- Create()
- CreateDropDown()
- Disable()
- Disableall()
- DisableBtn()
- DrawArrow()
- Enable()
- Enableall()
- EnableBtn()
- GetButton()
- GetButtonBitmap()
- GetButtonCount()
- GetButtonInfo()
- GetButtonSize()
- GetButtonsWidth()
- GetGapWidth()
- GetHeight()
- GetImageList()
- Init()
- IsButtonChecked()

- IsButtonClicked()
- MoveMenu()
- OnDestroy()
- OnDrawItem()
- OnInitMenuPopup()
- OnKillFocus()
- OnLButtonDblClk()
- OnMeasureItem()
- OnParentCommand()
- OnParentNotify()
- OnSize()
- OnToolTipNotify()
- OnUserMsg()
- OpenPopup()
- SetBanner()
- SetBitmapSize()
- SetButtonSize()
- SetButtonState()
- SetButtonWidth()
- SetColorScheme()
- SetDrawTextFlags()
- SetExtStyle()
- SetHotImageList()
- SetImageList()
- SetListStyle()
- SetPadding()
- UpdateMenu()

6.51. ToolButton

6.51.1. Properties

- Action
- Active
- AllowClose
- AllowUnDock
- AutoSize
- Caption
- Cargo
- Check
- CheckGroup

- Children
- CIsName
- Color
- ContextMenu
- DropDown
- Enabled
- EventHandler
- Events
- Font
- Form
- Height
- HotImageList
- hWnd
- Id
- ImageIndex
- ImageList
- IsChecked
- IsMenuItem
- KeepParentActive
- ICreated
- Left
- MdiMenu
- Menu
- Message
- Name
- nHScrollPos
- oBtnInfo
- oStruct
- Parent
- Position
- Property
- Separator
- ShortCut
- ShowText
- State
- Style
- This
- ToolTip
- Top
- VertScrollPos
- Visible

- WholeDropDown
- Width
- Wrap
- xTabOrder

6.51.2. Events

- AddMenuItem()
- CheckButton()
- Create()
- Delete()
- Destroy()
- Disable()
- Enable()
- GetBitmap()
- GetClientRect()
- GetRect()
- GetRectangle()
- GetWindowRect()
- Init()
- InvalidateRect()
- IsWindowVisible()
- MoveWindow()
- OnClick()
- OnParentMove()
- Press()
- Refresh()
- Release()
- SetImageIndex()
- SetPosition()
- SetState()
- SetStyle()
- SetText()
- SetWidth()
- Show()

6.52. TreeView

6.52.1. Properties

- BackColor
- Columns

- Columns
- Header
- Header
- Items
- Items
- Level
- Level
- OnBeginDrag
- OnBeginDrag
- OnBeginLabelEdit
- OnBeginLabelEdit
- OnBeginRDrag
- OnBeginRDrag
- OnEndLabelEdit
- OnEndLabelEdit
- OnLabelEdit
- OnLabelEdit
- OnSelDelete
- OnSelDelete
- SelectedItem
- SelectedItem

6.52.2.

Events

- AddItem()
- Create()
- D OnNotify()
- EditLabel()
- EnsureVisible()
- Expand()
- ExpandAll()
- GetChild()
- GetCount()
- GetFirstVisible()
- GetItem()
- GetItemRect()
- GetItemState()
- GetNextItem()
- GetNextVisible()
- GetRoot()
- GetSelected()
- GetSelText()

- GetVisibleCount()
- HitTest()
- Init()
- OnClick()
- OnParentNotify()
- OnRightClick()
- OnSelChanged()
- OnSelChanging()
- ResetContent()
- SearchString()
- SelectItem()
- SetBkColor()
- SetImageList()
- SetIndent()
- SetItemHeight()
- SetScrollTime()
- SetTextColor()
- Toggle()
- TvnKeyDown()

6.53. TreeListView

6.53.1. Properties

- Header

6.53.2. Events

- Create
- Init()

6.54. TreeviewItem

6.54.1. Properties

- Action
- Bold
- Caption
- Cargo
- CollItems
- Expanded
- Expanded

- Form
- hItem
- HotImageIndex
- ImageIndex
- InsertAfter
- Items
- Level
- Owner
- Parent
- PointerItem
- Property
- tvI

6.54.2. Events

- AddItem()
- Create()
- Delete()
- EditLabel()
- EnsureVisible()
- Expand()
- ExpandAll()
- GetChild()
- GetItemRect()
- GetItemState()
- GetNextItem()
- Init()
- SearchString()
- Select(nFlag)
- SetFont()
- SetItemState()
- SetStyle()
- SortChildren()
- Toggle()

6.55. WebBrowser

6.55.1. Properties

- ClsID
- ProgID
- Url

6.55.2. Events

- Create
- Init()
- WebNavigate()

6.56. ContextMenu

6.56.1. Properties

- Caption
- Menu
- MenuWnd
- Parent
- Property
- SelfPopup
- this

6.56.2. Events

- AddCoolMenuItem()
- Create()
- Init()
- MenuDesignHook()
- Show()

6.57. CMenuItem

6.57.1. Properties

- Action
- altems
- Bitmap
- BkBrush
- BmplItem
- Caption
- Cargo
- Children
- ClsName
- Components
- Default
- Enabled

- EventHandler
- Events
- Font
- ForeColor
- ForeSysColor
- Form
- Height
- hMenu
- HorzScrollPos
- HotImageList
- hWnd
- Id
- ImageIndex
- ImageList
- Left
- lInserted
- Menu
- MenuBreak
- MenuItemInfo
- Message
- Name
- Parent
- Position
- Property
- RadioCheck
- SetChildren
- ShortCut
- ShortCutText
- Status
- Style
- This
- ToolTip
- Top
- Type
- VertScrollPos
- Visible
- Width
- xForeColor

6.57.2. Events

- AddMenuItem()

- Check()
- CheckForDefaultItem()
- Create()
- Delete()
- Destroy()
- Disable()
- DrawItem()
- DrawItemText()
- Enable()
- GetClientRect()
- GetItem()
- GetMenuByHandle()
- GetMenuById()
- GetMenuByKey()
- GetRectangle()
- GetSubMenu()
- GetWindowRect()
- HiLite()
- Init()
- InvalidateRect()
- IsChecked()
- IsEnabled()
- IsMenuItem()
- MeasureItem()
- OnClick()
- Select()
- SetDefaultItem()
- SetStatus()
- SetSubMenu()
- SetText()
- UnCheck()

6.58. ImageList

6.58.1. Properties

- Count
- Events
- Handle
- IconHeight
- IconWidth

- Images
- MaskColor
- Palette

6.58.2. Events

- AddBitmap()
- AddIcon()
- AddImage()
- Clean()
- ComboBox()
- Create()
- Destroy()
- DrawDisabled()
- DrawImage()
- FromToolBar()
- GetImage()
- GetImages()
- Init()
- SetIconProp()
- SetMaskColor()

6.59. MemoryTable

6.59.1. Properties

- Alias
- bOnFileClosed
- bOnFileNameChanged
- Fields
- IBof
- IEof
- Parent
- Record
- Structure
- Table
- Tag

6.59.2. Events

- Append()
- Bof()
- Close()

- Create()
- Delete()
- Deleted()
- Eof()
- FieldGet()
- FieldPos()
- FieldPut()
- Found()
- GoBottom()
- GoTo()
- GoTop()
- Init()
- Insert()
- IsOpen()
- OrdKeyCount()
- OrdKeyGoTo()
- OrdKeyNo()
- OrdKeyRelPos()
- OrdSetFocus()
- RecCount()
- RecLock()
- RecNo()
- Seek()
- Skip()
- UnLock()
- Zap()

6.60. MemData

6.60.1. Events

- FieldGet()
- FieldPut()
- Put()

6.61. NotifyIcon

6.61.1. Properties

- Balloon_Icons
- BalloonTipIcon

- BalloonTipText
- BalloonTipTitle
- ContextMenu
- Flags
- hWnd
- Icon
- ID
- Message
- Parent
- Text
- Visible

6.61.2. Events

- Destroy()
- Init()
- SetBalloonTipIcon()
- SetVisible()

6.62. SerialPort

6.62.1. Properties

- BaudRate
- DataBits
- Events
- Handle
- HandShake
- Parity
- PortName
- StopBits

6.62.2. Events

- Close()
- Init()
- Open()

6.63. Service

6.63.1. Properties

- EventBlock

- File
- hService
- hServiceManager
- ServiceName
- ServiceStatus
- ServiceStatusHandle
- StopServiceEvent
- TimeOut

6.63.2. Events

- Delete()
- Install()
- Run()
- Start()
- Stop()

6.64. Timer

6.64.1. Properties

- AutoRun
- CIsName
- Delay
- Events
- hProc
- Id
- IRunning
- Parent

6.64.2. Events

- Create()
- Destroy()
- Init()
- OnTimeOut()
- SetDelay()
- Start()
- Stop()
- TimeProc()

6.65. WinSock

6.65.1. Properties

- ClsName
- Events
- Handle
- IListener
- LocalIP
- LocalPort
- nRecId
- pCallBack
- Protocol
- RecData
- RemoteHandle
- RemoteIP
- RemotePort
- Status

6.65.2. Events

- Close()
- Connect()
- Create()
- Destroy()
- Disconnect()
- Init()
- Listen()
- Send()
- SockControlPoc()

6.66. DataConnector

6.66.1. Properties

- AutoConnect
- cLogFile
- ClsName
- Connected
- ConnectionID
- ConnectionString
- Events
- Sql

6.66.2. Events

- Commit()
- Connect()
- Create()
- Disconnect()
- Exec()
- Execute()
- Fetch()
- FieldGet()
- Getline()
- GetStruct()
- Init()
- RollBack()

6.67. DataTable

6.67.1. Properties

- Alias
- AllowClose
- AllowUndock
- Area
- AutoOpen
- BackColor
- Bitmap
- bOnFileClosed
- bOnFileNameChanged
- Border
- Caption
- ClientEdge
- ClipChildren
- ClipSiblings
- CodePage
- DataConnector
- Dock
- Events
- Exists
- Fields
- FileName
- Font

- ForeColor
- Height
- Id
- IDEButton
- IndexOrder
- MDIClient
- OwnerDraw
- Parent
- Path
- ReadOnly
- ServerType
- Shared
- SmallCaption
- StaticEdge
- Structure
- ToolTip
- Transparent
- Visible
- Width
- xDriver
- xFilter

6.67.2. Events

- Append()
- Blob2File()
- Bof()
- CheckAlias()
- Close()
- Commit()
- Create()
- CreateIndex()
- Delete()
- Deleted()
- Destroy()
- Eof()
- File2Blob()
- FileLock()
- Found()
- Gather()
- GoBottom()
- GoTo()

- GoTop()
- IndexOrd()
- Init()
- IsOpen()
- KillScope()
- Open()
- OrdKey()
- OrdKeyCount()
- OrdKeyGoTo()
- OrdKeyNo()
- OrdKeyRelPos()
- OrdName()
- OrdSetFocus()
- Recall()
- RecCount()
- RecLock()
- RecNo()
- Reindex()
- Scatter()
- Seek()
- Select()
- SetAlias()
- SetBottomScope()
- SetDriver()
- SetFileName()
- SetIndex()
- SetOrder()
- SetRelation()
- SetTopScope()
- Skip()
- Struct()
- UnLock()
- UnLockAll()
- Used()
- Zap()

6.68. Data

6.68.1. Properties

- Parent

6.68.2. Events

- Init()
- FieldGet()
- FieldPut()
- Put()

6.69. ColorDialog**6.69.1. Properties**

- aCustom
- AnyColor
- Color
- Colors
- Custom
- FullOpen
- PreventFullOpen
- ShowHelp
- SolidColor
- SysDefault

6.69.2. Events

- Init()
- Show()

6.70. FolderBrowserDialog**6.70.1. Properties**

- Description
- RootFolder
- SelectedId
- SelectedPath
- ShowNewFolderButton

6.70.2. Events

- BrowseForFolderCallBack()
- Init()
- SetShowNewFolderButton()
- Show()

6.71. OpenFileDialog

6.71.1. Properties

- AddExtension
- CheckFileExists
- CheckFileExists
- CheckPathExists
- DefaultExt
- DeferenceLinks
- FileName
- Filter
- FilterIndex
- InitialDirectory
- Multiselect
- ReadOnlyChecked
- RestoreDirectory
- ShowHelp
- ShowPlacesBar
- ShowReadOnly
- Title

6.71.2. Events

- Init()
- SetInvStyle()
- Show()

6.72. PrintDialog

6.72.1. Properties

- AllowCurrentPage
- AllowPrintFile
- AllowSelection
- AllowSomePages
- Copies
- Default
- DriverName
- FromPage
- Port

- PrinterName
- ToPage

6.72.2. Events

- Init()
- SetInvStyle()
- Show()

6.73. SaveFileDialog

6.73.1. Properties

- AddExtension
- CheckFileExists
- CheckPathExists
- CreatePrompt
- DefaultExt
- DeferenceLinks
- FileExtension
- FileName
- Filter
- FilterIndex
- InitialDirectory
- RestoreDirectory
- ShowHelp
- ShowPlacesBar
- Title

6.73.2. Events

- Init()
- Show()
- SetInvStyle()