



Version 1.12 (Updated September 2002)

Welcome to GExperts, a Delphi and C++Builder IDE enhancements package.

GExperts is a set of tools built to increase the productivity of Delphi and C++Builder programmers by adding several features to the IDE. GExperts is developed as Open Source software and encourages user contributions to the project.

GExperts would not exist without the help and advice of many people. In particular, Gerald Nunn was the lead developer of GExperts for almost two years, before turning the project over to Erik Berry in 1998. Gerald allowed Erik to make the project Open Source to speed development and further enhance the quality of the experts. For a list of credits, see the change log in the Readme.txt, or open the GExperts about box.

Please take a moment to read the [GExperts License Agreement](#). Also, remember that the Readme.txt file included with GExperts will often contain information not included in this help file including the latest changes, known bugs, and plans for the future.

The latest version of GExperts and the FAQ are always available at: <http://www.gexperts.org/>. You can send bug reports and suggestions using the Feedback Wizard in the GExperts about box.

GExperts is copyright 1996-2002 by GExperts, Inc, Erik Berry, and several other authors who have submitted their code for inclusion. This license agreement only covers code written by GExperts, Inc and Erik Berry. You should contact the other authors concerning their respective copyrights and conditions.

The rules governing the use of GExperts and the GExperts source code are derived from the official Open Source Definition, available at <http://www.opensource.org/>. The conditions and limitations are as follows:

- Usage of GExperts binary distributions is permitted for all developers. You may not use the GExperts source code to develop proprietary or commercial products including plugins or libraries for those products. You may use the GExperts source code in an Open Source project, under the terms listed below.
- You may not use the GExperts source code to create and distribute custom versions of GExperts under the "GExperts" name. If you do modify and distribute custom versions of GExperts, the binary distribution must be named differently and clearly marked so users can tell they are not using the official GExperts distribution. A visible and unmodified version of this license must appear in any modified distribution of GExperts.
- Custom distributions of GExperts must include all of the custom changes as a patch file that can be applied to the original source code. This restriction is in place to protect the integrity of the original author's source code. No support for modified versions of GExperts will be provided by the original authors or on the GExperts mailing lists.
- All works derived from GExperts must be distributed under a license compatible with this license and the official Open Source Definition, which can be obtained from <http://www.opensource.org/>.
- Please note that GExperts, Inc. and the other contributing authors hereby state that this package is provided "as is" and without any express or implied warranties, including, but not without limitation, the implied warranties of merchantability and fitness for a particular purpose. In other words, we accept no liability for any damage that may result from using GExperts or programs that use the GExperts source code.

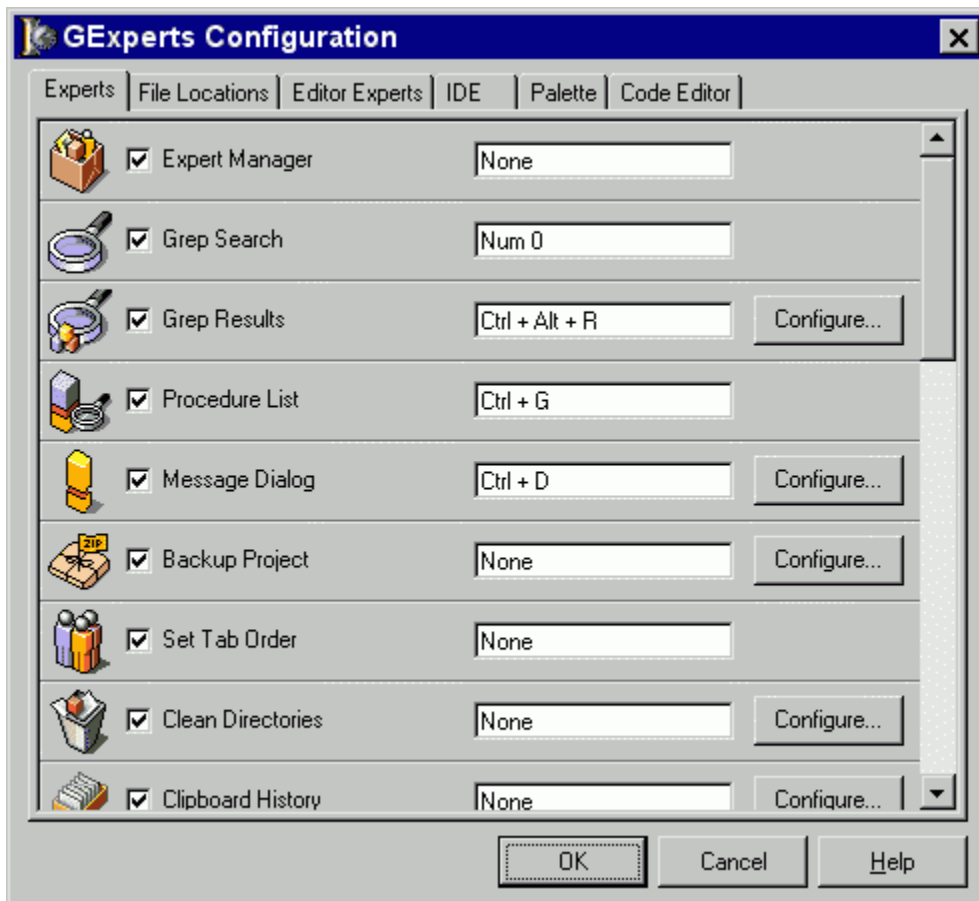
If you have questions about this license, please email Erik Berry at eberry@gexperts.org.

Here you can configure the various parts of GExperts. The window has tabs for configuration of the individual experts, file locations, editor experts, IDE enhancements, component palette enhancements, and code editor enhancements.

The first tab, shown below, allows you to turn on and off the individual experts using the checkbox next to the expert name. If you uncheck an expert, it will be removed from the GExperts menus and toolbars. You can also configure which shortcut key you want to use to activate each expert. GExperts makes no attempt to ensure a selected shortcut key is unique and many common shortcuts are already in use by the main menu or the code editor. Note that you can scroll down the list on this tab to see the remainder of the experts.

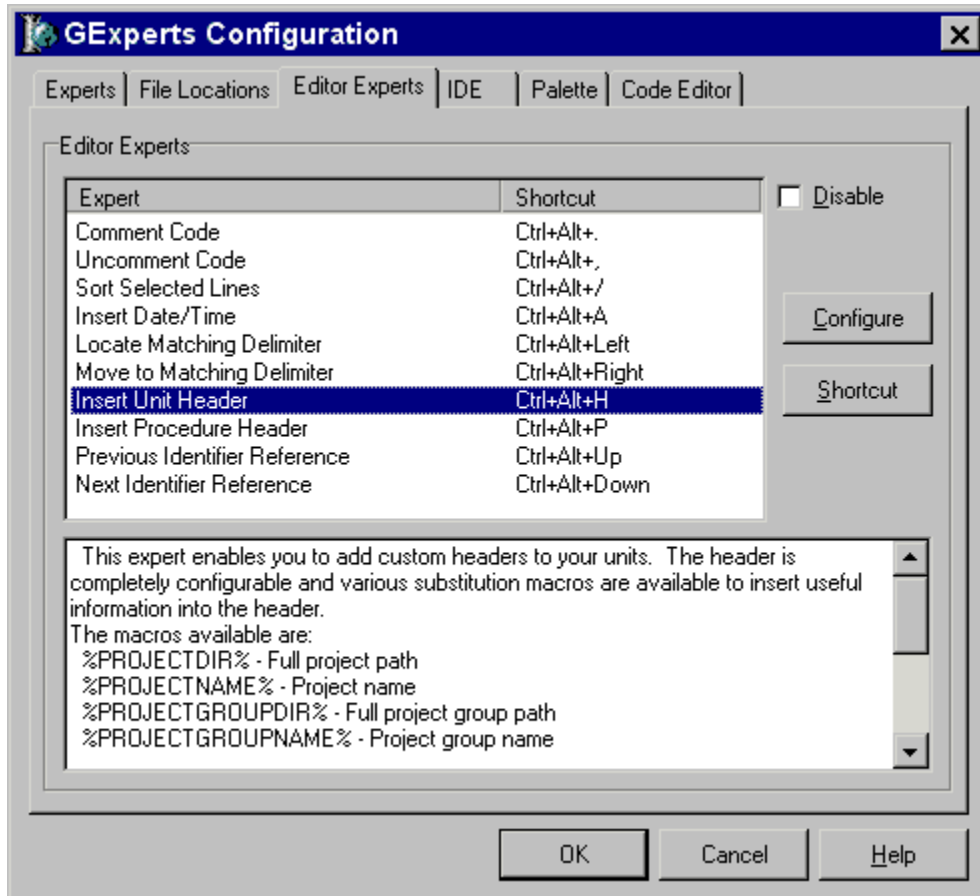
The File Locations tab allows you to set locations for the VCL source, the default GExperts data storage directory, and the GExperts help file.

The editor experts, IDE enhancements, component palette enhancements, and code editor enhancements are described in separate help sections.



The editor experts are active only when the IDE code editor has focus. They perform various functions on the source. The shortcut key combination for each editor expert can be changed via the Shortcut button, but you should verify the shortcut is unused by the editor itself (see Key Mappings in the IDE help file for more information) and the main menu.

Some of the experts have options that can be changed using the Configure button. Selecting such an expert enables the button. The function of each of these experts is described in the help memo at the bottom of the tab.



Note that two of these experts offer macro support - the Unit Header Expert and the Procedure Header Expert. These macros allow you to substitute dynamic values into the inserted code using utilizing % delimited keywords such as %MONTH%.

The GExperts IDE enhancements use low-level techniques to interface directly with the IDE and add often-requested features. You can disable all of the IDE enhancements at once using the "Disable IDE enhancements" checkbox.

Tab Dock Hosts

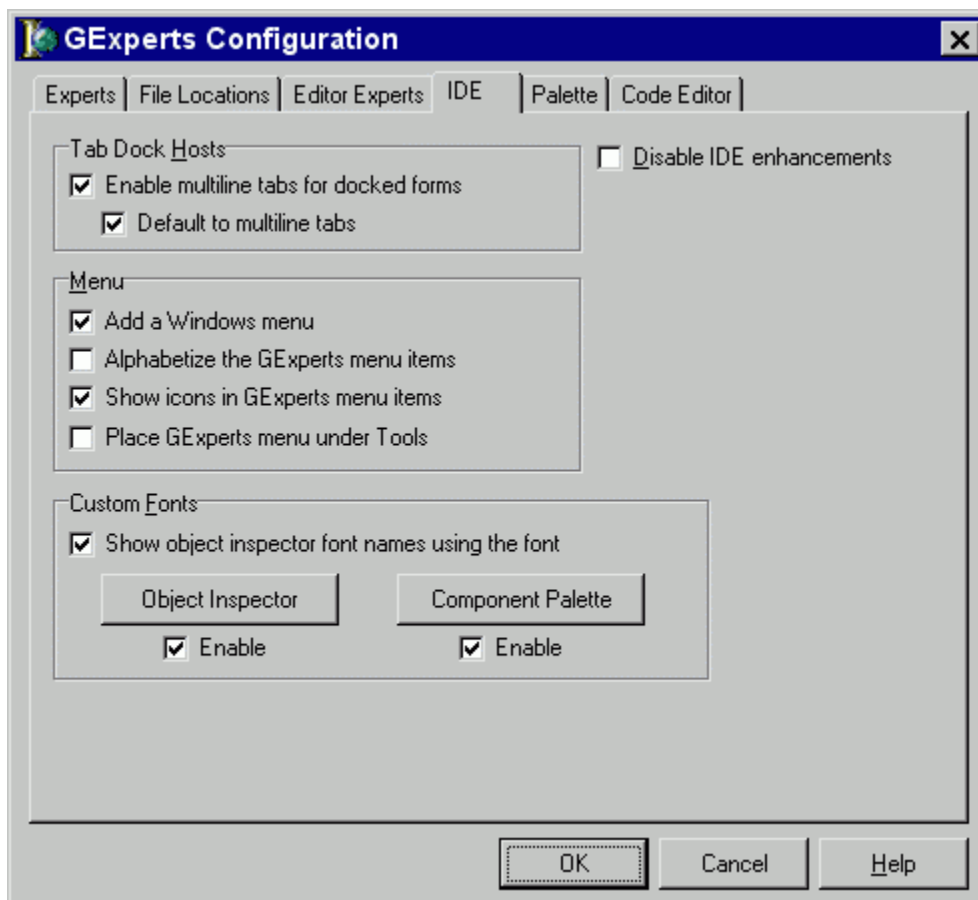
When docking multiple IDE tool windows together, GExperts can force the IDE to use multiline tabs when the tabs do not fit in the available space. The multiline tab setting can be changed on a per-dock host basis from the right-click menu of the tabs, or set as the default for all new dock hosts.

Menu

GExperts can modify the IDE main menu by adding a top level Windows menu that lists the visible IDE windows for quick selection (already supported by the IDE in Delphi 6+). You can also choose to sort the GExperts menu items in alphabetical order, show icons in the GExperts menus, and place the GExperts menu in the Tools menu instead of as a top-level menu.

Custom Fonts

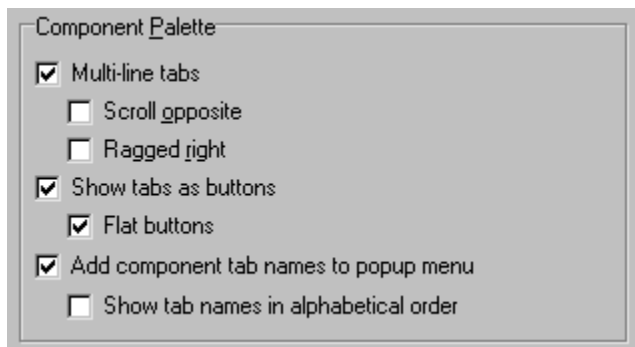
You can edit the font used in the component palette and the object inspector by checking the appropriate checkbox and then clicking the associated font button. You can also tell the object inspector to display font name dropdowns selections using the actual font face.



GExperts can enhance the component palette in many ways. You have the option of showing multiple rows of tabs to see more component categories at once. If this option is set, you can also enable the ScrollOpposite and RaggedRight properties of the tab control. Scroll opposite determines where previous rows of tabs in a multi-line tab control move when the user selects a tab in another row. When scroll opposite is checked, previous rows of tabs are moved to the bottom of the tab control. Ragged right determines whether or not incomplete rows of tabs stretch to fill the width of the tab control. See the VCL help for more information on these TCustomTabControl properties.

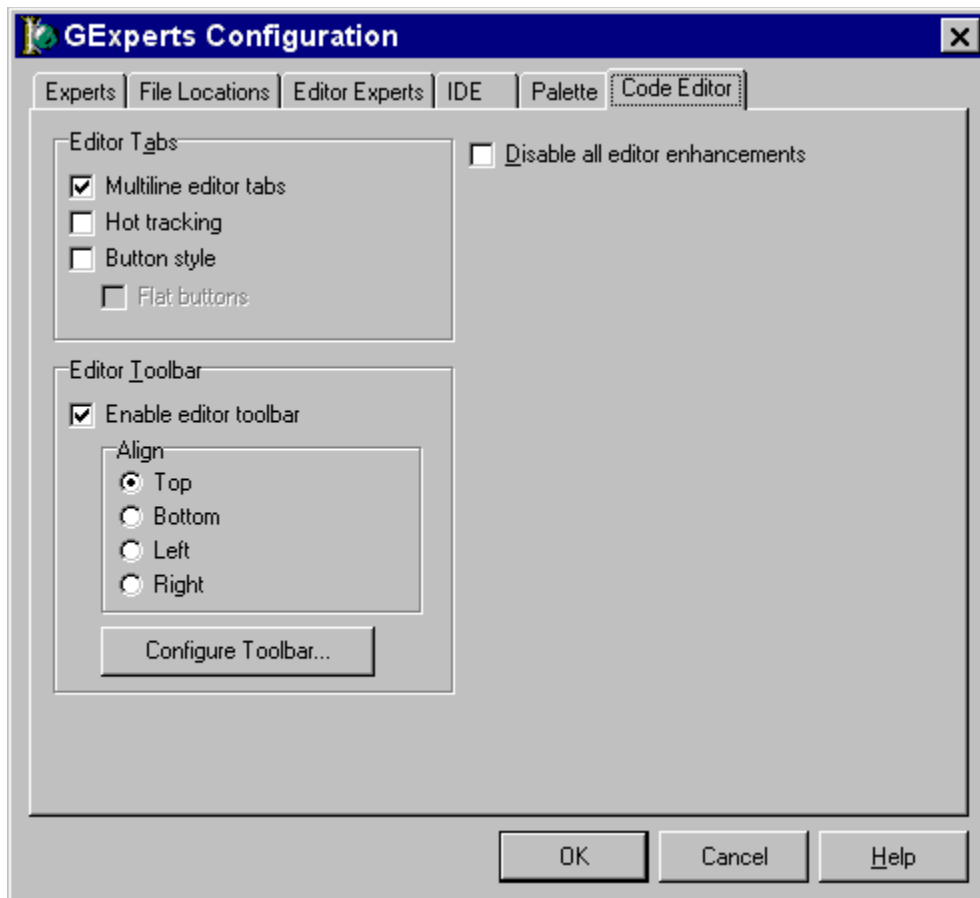
You can also choose to show the component palette tabs in a raised button style (see TCustomTabControl.Style) or a flat button style.

If you want quick access to lots of component categories, but don't want to enable the multi-line tab option, you can add the component categories to the palette's popup menu. Choosing one of these menu items selects the corresponding tab on the component palette. Finally, you can force the menu items to be sorted alphabetically. This option is also provided by the IDE in Delphi 6+.



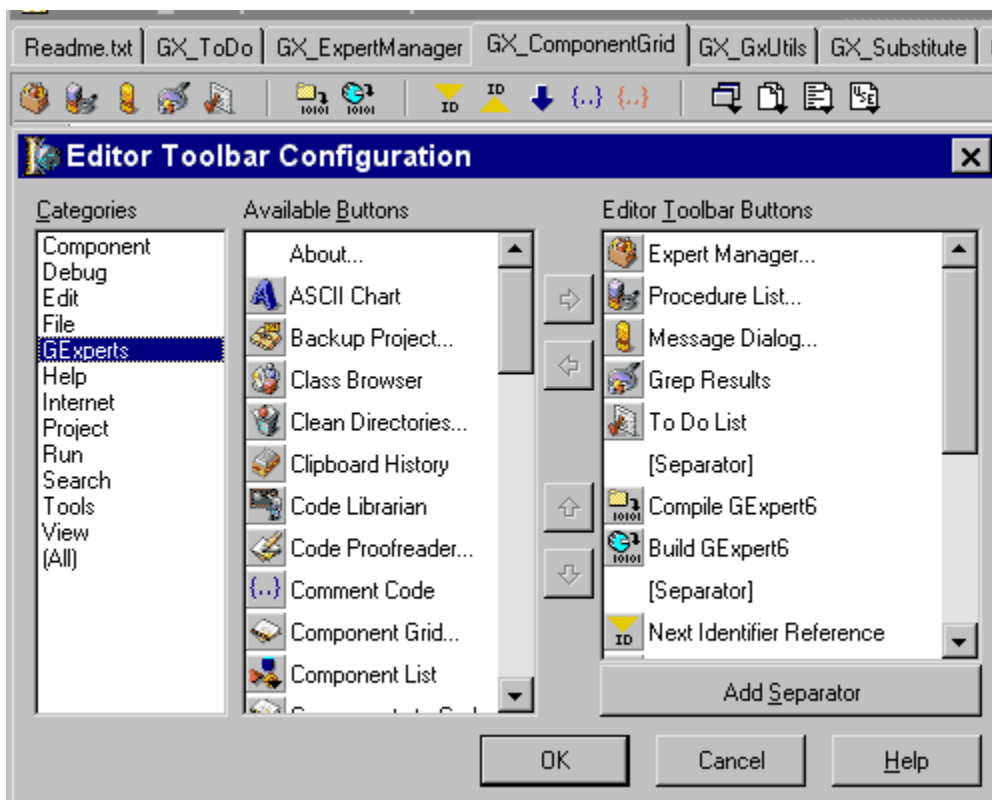
The GExperts editor enhancements can improve the functionality of the IDE editor in the following ways:

- Make the code editor tabs multiline instead of utilizing horizontal scrolling
- Turn on/off hot tracking for the editor tabs (see TCustomTabControl.HotTrack)
- Show the code editor tabs in a raised or flat button style
- Enable and then position the GExperts editor toolbar to the top, left, right, or bottom of the code editor window



GExperts includes the ability to place a user-defined toolbar in the IDE code editor windows (see the [Code Editor](#) tab in the GExperts configuration dialog). To select which buttons are visible on the toolbar, drag buttons from the list of Available Buttons to the Toolbar Buttons list. Reordering buttons in the selected Buttons list is also possible using drag-and-drop or the Up and Down arrows. To remove a button from the toolbar, you can drag it from the Toolbar Buttons list and drop it back onto the Available Buttons list

The majority of the available buttons are taken from IDE's main menu, but there are also several special buttons in the GExperts category that perform custom actions. These custom actions can execute GExperts experts, execute editor experts, show a list dropdown list of forms or units in the current project, or allow you to select a component on the current form by class and name. There is also a button to jump to specific positions in the current Object Pascal unit. An example editor toolbar is shown here behind the toolbar configuration dialog:



Tip: The keyboard is always faster than the mouse! Long live UNIX!

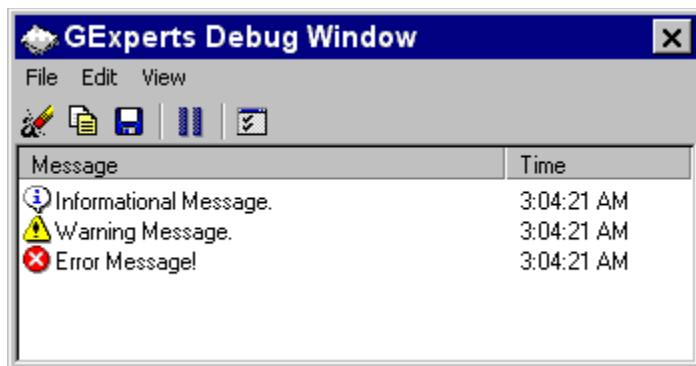
Note: Vertical toolbars do not support button separators.

The debug window is useful in those situations where you can not use the regular IDE debugger. To use the debug window, include the unit `DbugIntf.pas` that comes with GExperts in your unit's uses clause and then in the spot you need a debug statement inserted, type `SendDebug('Debug note');`. You may want to wrap your debug statements in a conditional define such as `{IFDEF DEBUG}` and create an IDE Code Template to easily insert new debug statements into your code.

Three types of messages can be sent to the debug window: informational, warning, and error. These message types are sent using the `mtInformation`, `mtWarning` and `mtError` constants that are defined in the VCL's Dialogs unit and can be sent using the `SendDebugEx` command: `SendDebugEx('Debug message', mtInformation);`. `DbugIntf.pas` also defines several other useful methods to send data types directly and signal entry into and exiting from methods, etc.

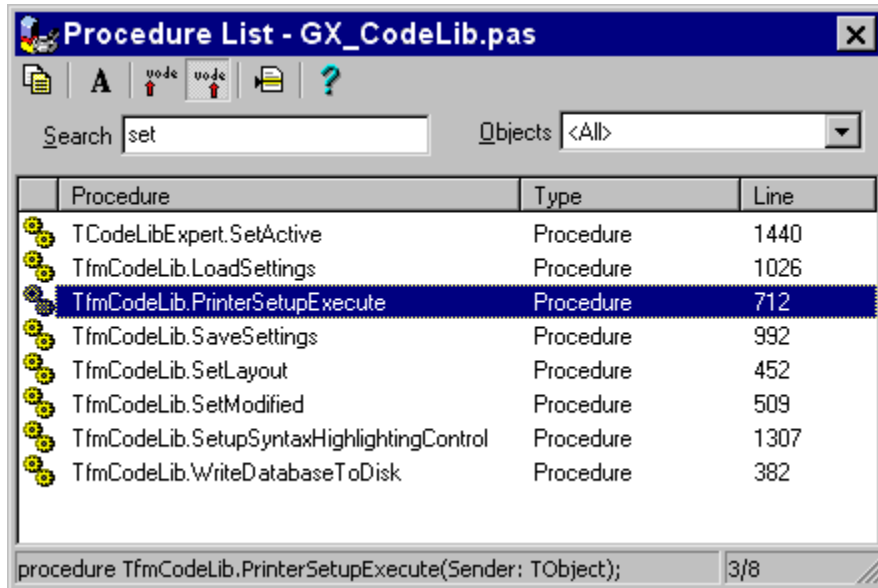
The GExperts Debug Window is a separate program (`GDebug.exe`) that sits in the task tray of the Windows shell. If you try to send a debug message and the Debug Window is not running, GExperts will try to start the window automatically. Whenever the window receives a new message, the icon in the task tray will change from a plain paper icon to a paper icon with a pen writing on it.

The Debug Window allows you to clear the message display, copy the messages to the clipboard, save them to a file, pause collection of messages, or configure some basic options.



Tip: The debug window can be used to debug painting routines or components at design-time, because these tasks are sometimes difficult with the standard Borland IDE debugger.

The procedure list enables you to view a list of procedures in the current unit and quickly jump to the implementation of a given procedure. The procedure list window is pictured below:



To search for a procedure, start typing in the Search edit box. As you type, the characters will appear at the top next to the word **Search**. To jump to a selected procedure, double click or highlight it and press enter.

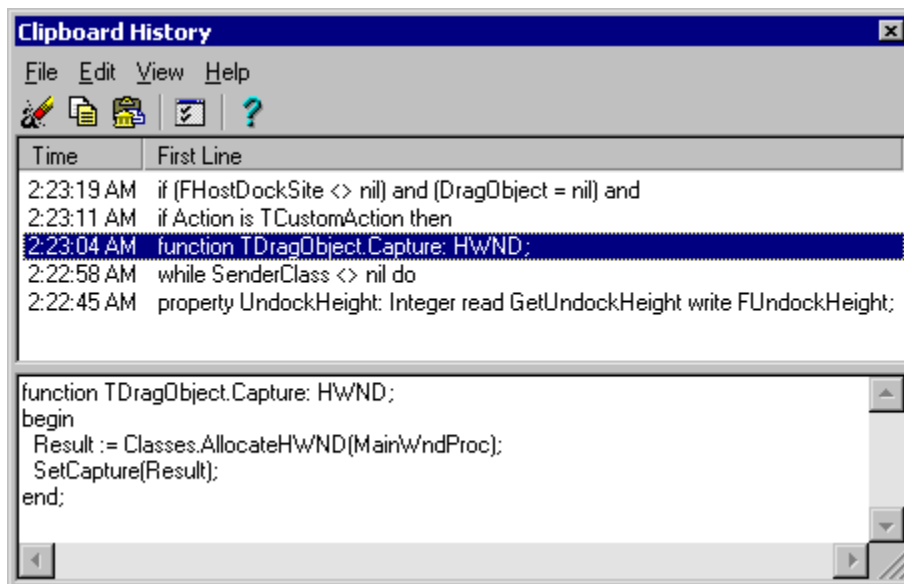
Two search modes are provided. In the first search mode (match only from the start), searches are conducted only on the beginning of the procedure name (after an optional object reference). In the second search mode (match anywhere), the search string can match at any point in the procedure name. For example, if you had two functions, **TMyObject.Assign** and **TMyObject.AssignWidget**, searching for "assign" would return both methods in either mode, whereas searching for widget would return neither when searching from the start, while **TMyObject.AssignWidget** would appear if searching for a match at any point.

Using the Objects combobox on the right, you can filter the procedure list to display all procedures, only those without an associated object, or only those associated with a specific object. The Copy button will copy all of the visible procedure details to the clipboard.

The Clipboard History expert allows you to view the current and previous text contents of the clipboard. You can re-copy any of the saved items by double-clicking the item or pressing the Copy button. The saved items may be purged at any time by choosing Clear. This expert supports IDE docking.

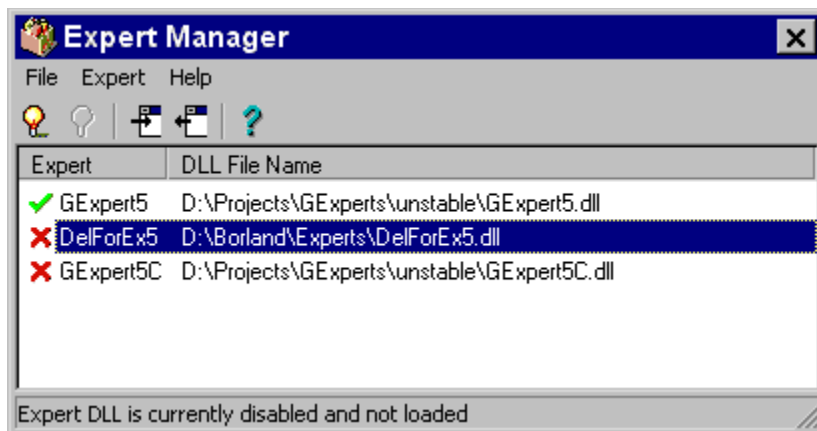
Several options can be set via the configuration form, including the maximum number of items to store, whether or not Clipboard History starts capturing when the IDE starts, and whether or not to hide the form after pasting the selected item into the code editor. The clipboard history data is persistent across IDE sessions.

Note: The Clipboard History expert only tracks standard ASCII (CF_TEXT) items on the clipboard. It does not maintain a history of other clipboard items such as bitmaps, components, etc.



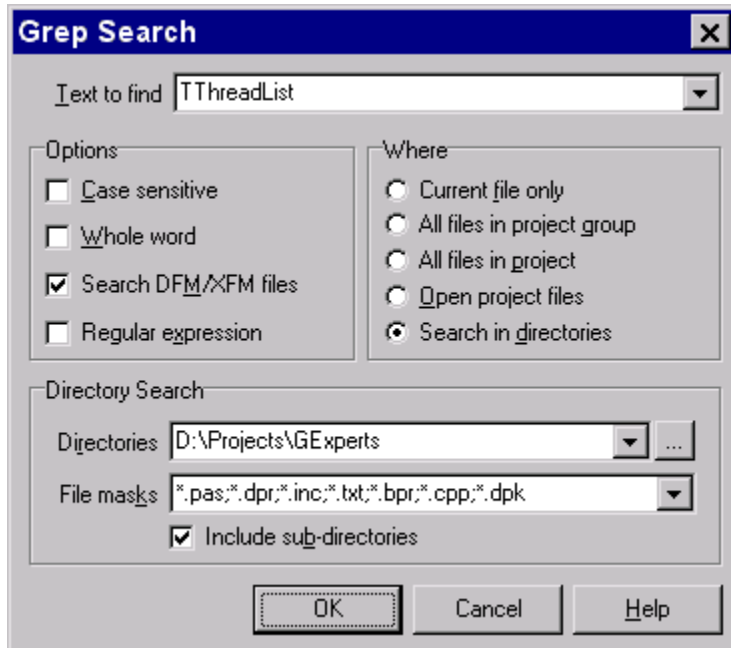
The expert manager enables you to add, enable, disable, and remove DLL experts from the IDE without having to directly edit the registry. The experts marked with a red X are disabled and those with the green checkmark are enabled. If an expert has a red dot in front of the graphic, the expert's state has been changed since the IDE was started, and you will not see the effect of the change until you restart the IDE. Dynamically loading DLL experts is dangerous and rarely works, therefore GExperts does not permit this.

Note that the expert manager does not allow you to turn on and off the various GExperts experts. This can be done in the [GExperts Configuration](#) window using the checkboxes on the Experts tab.



GExperts incorporates a powerful search facility that enables you to quickly locate text strings in files. Using Grep, you can search the current file, all open project files, all files in your project or project group, or all files in a directory (optionally including sub-directories). The grep search is also available outside the IDE by executing the standalone Grep.exe application.

To begin a search, select Grep Search from the GExperts menu. A dialog will appear like the one below into which you can enter your search criteria. Note that if you have a string highlighted in the code editor when calling up this dialog, it will be used as the default search string.



The various options in the search dialog are as follows:

Text to Find: The text or [regular expression](#) to search for

Options:

- Case sensitive: Search is case sensitive (a and A are different characters)
- Whole word: Return matches that are whole words (delimited by whitespace or punctuation such as "(), <> - {} ! @ # \$ %"). Note that 0-9 and _ are treated as part of a word.
- Search DFM/XFM files: Include form files in the search
- Regular expression: The text to find is a [regular expression](#)

Where (search scope):

- Current file only: Only the file that is currently in focus for editing
- All files in project: All files that are part of the current project
- All files in project group: All files that are part of the current project group
- Open project files: All project files that are currently open in the editor
- Search in directories: All files specified by the Directory Search options (see below)

Directory Search:

Note: This portion of the dialog is only enabled if "Search in directories" is selected.

- Directories: A semicolon separated list of directories to search
- "..." Button: Allows browsing for a search directory

File masks: Limits the search to a semicolon separated list of file extensions
Include subdirectories: Enables recursive searching of the search directories

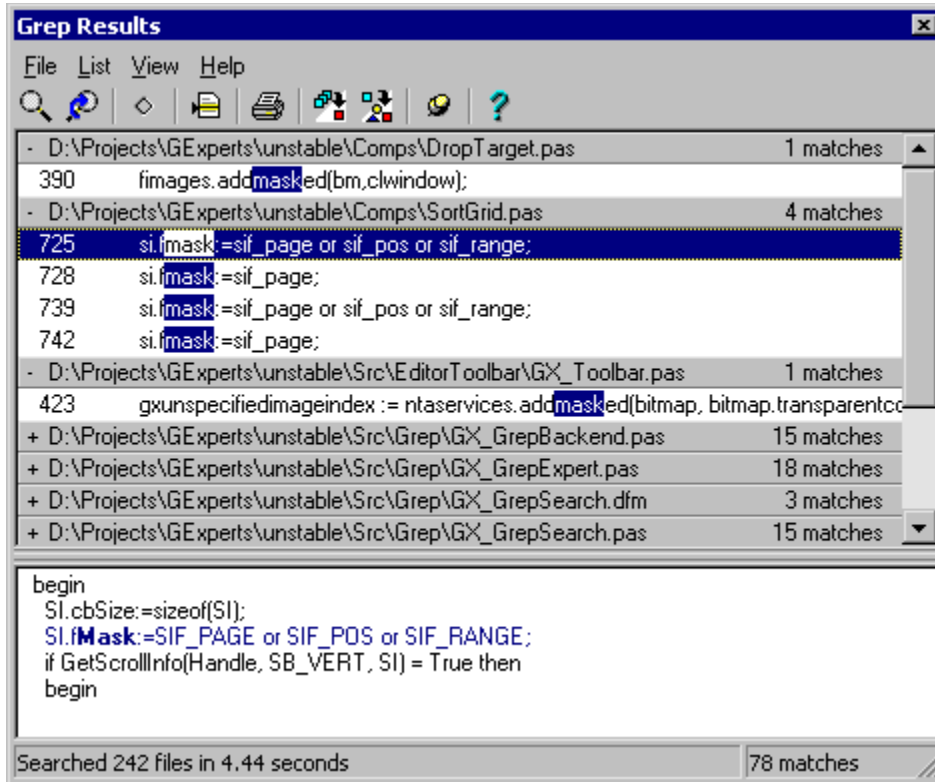
Once you have entered the search criteria, click the OK button or press enter to initiate the search. As the search progresses, results will be shown in the Grep Results window.

The GExperts Configuration dialog allows setting of some Grep Search options such as saving of recent searches and ANSI compatible case insensitive searching (which is slower).

Using Grep regular expressions, more complex searches can be formulated. For example, to locate all instances of the words **unit** and **init**, you can search for **[ui]nit**. A description of the regular expressions supported by GExperts is below. The GExperts regular expression engine does not support the '*' modifier (which usually matches 0 or more copies of the previous character or block) or character groups using brackets '{}'.

- \ The backslash escapes any character and can therefore be used to force characters to be matched as literals instead of being treated as characters with special meaning. For example, '\[' matches '[' and '\\ matches '\\.
- ^ A circumflex at the beginning of an expression matches the beginning of a line.
- \$ A dollar sign at the end of an expression matches the end of a line.
- .
- :
- [...] A string enclosed in square brackets matches any character in that string, but no others. If the first character in the string is a circumflex (^), the expression matches any character except the carriage return (ASCII 0x0D) or line feed (ASCII 0x0A) and the characters in the string. For example, '[xyz]' matches only 'x', 'y', or 'z', while '[^xyz]' matches 'a' but not 'x', 'y', or 'z'. A range of characters may be specified by two characters separated by '-'. Note that '[a-z]' matches alphabetic characters, while '[z-a]' never matches. Special characters do not need to be escaped when they are inside brackets, because they are treated as literals. Nesting of brackets is not supported.

The Grep Results window is where the results of a [Grep Search](#) are shown. It uses a folding display of matches to allow you to easily locate a particular match. This expert supports IDE docking. An example of this window appears below.



The results window displays all files which contained one or more matches for the search term. Under each file, a list of matches for that particular file can be shown. To expand or contract a file's matches, click on the filename, press enter, or use the '+' and '-' keys. When a specific match line is selected, the window can show any number of lines of match context using the Show Match Context and Set Context Lines menu items. Note that the match context might not be accurate if you have edited the searched files since the search.

The number to the left of each match is the line number where the match was found. The results list highlights the matching characters in each entry to indicate where the match occurred.

To jump to a match in the IDE editor, double click the desired line, press enter, or use the Goto button. You can control whether the result is positioned at the top of the IDE editor or in the middle through an option in the [GExperts Configuration](#) for the Grep Results expert.

To start a new search, click the Search button and the [Grep Search](#) dialog will appear. As the search progresses, the new search button will be disabled and the abort button will be enabled to cancel the current search. Once a search is completed, the results window displays the number of files searched, the search time, and the total number of matches.

You can expand all items in the list by clicking the expand button in the toolbar. Similarly, clicking the contract button in the toolbar will contract all result items. You can also change the font the grep results are displayed in by clicking the font button. The entire match list can be copied to the clipboard or saved

to a file using the items on the File menu.

Tip: You can drag and drop a file from the results list onto a running program like notepad, or you can drag it onto a program icon or shortcut to execute the program with the selected file. When running the standalone Grep.exe, double clicking a file will open the associated program for that file type.

The Message Dialog expert provides a convenient way to enter MessageDlg function calls code into your source code. To use this expert, fill out the fields shown and press the OK button. The expert will generate the required MessageDlg code and insert it into the editor. The expert is smart enough to recognize text that spans multiple lines. It will generate code that results in a dialog box identical to the one you see when you click the Test button.

This expert can generate an "if" statement to test the user's response to the dialog box against the set of responses marked in the Function Results box. To set the current options as the default, check the "Save as default" checkbox before pressing OK.

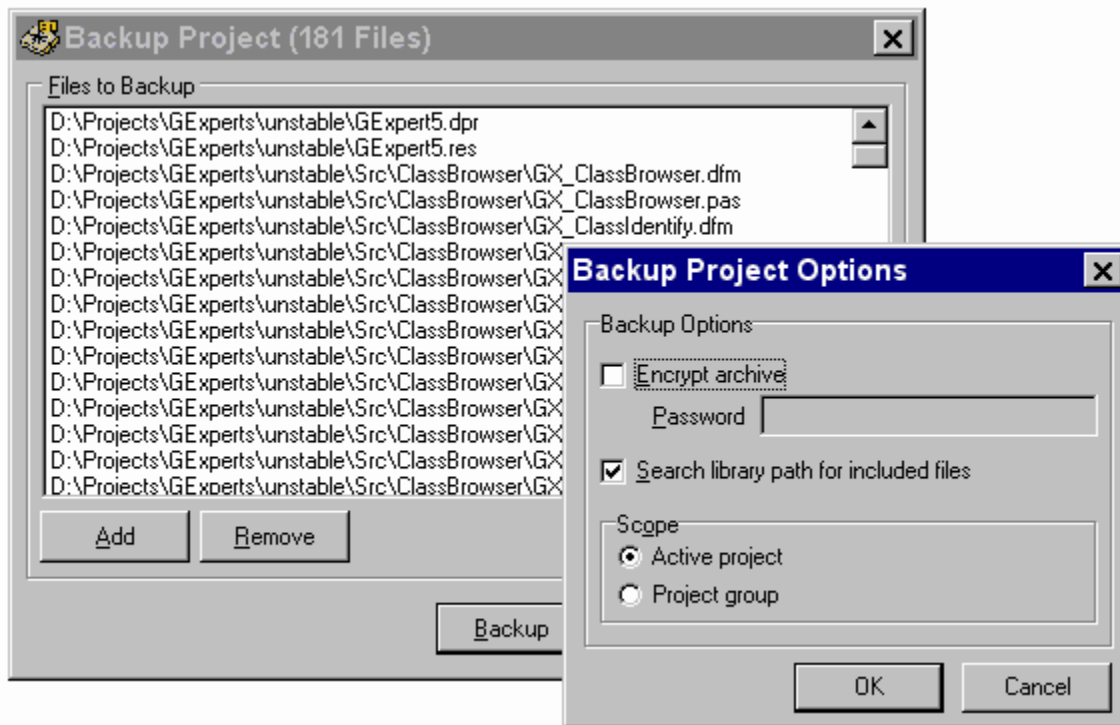
The screenshot shows the "Message Dialog" expert dialog box. It has a title bar with "Message Dialog" and a close button. The main area is divided into several sections:

- MessageDlg**: A text field containing "Warning! GExperts may save you hours of coding!".
- Dialog Type**: Radio buttons for `mtWarning` (selected), `mtError`, `mtInformation`, `mtConfirmation`, and `mtCustom`. Each has a corresponding icon (warning, error, information, confirmation).
- Dialog Buttons**: Checkboxes for `mbYes`, `mbNo`, `mbOK` (checked), `mbCancel`, `mbAbort`, `mbRetry`, `mbIgnore`, `mbAll`, `mbNoToAll`, `mbYesToAll`, and `mbHelp`.
- Function Results**: Checkboxes for `mrYes`, `mrNo`, `mrOK` (checked), `mrCancel`, `mrAbort`, `mrRetry`, `mrIgnore`, `mrAll`, and `mrNone`.
- Help Context**: A dropdown menu showing "0".
- Logic**: Checkboxes for `If statement` (checked) and `Negate if`.
- Buttons**: "Test", "OK", "Cancel", and "Help".
- Save as default**: A checkbox at the bottom left.

Tip: You can press Ctrl+Enter while typing in the message memo to confirm the dialog (click OK).

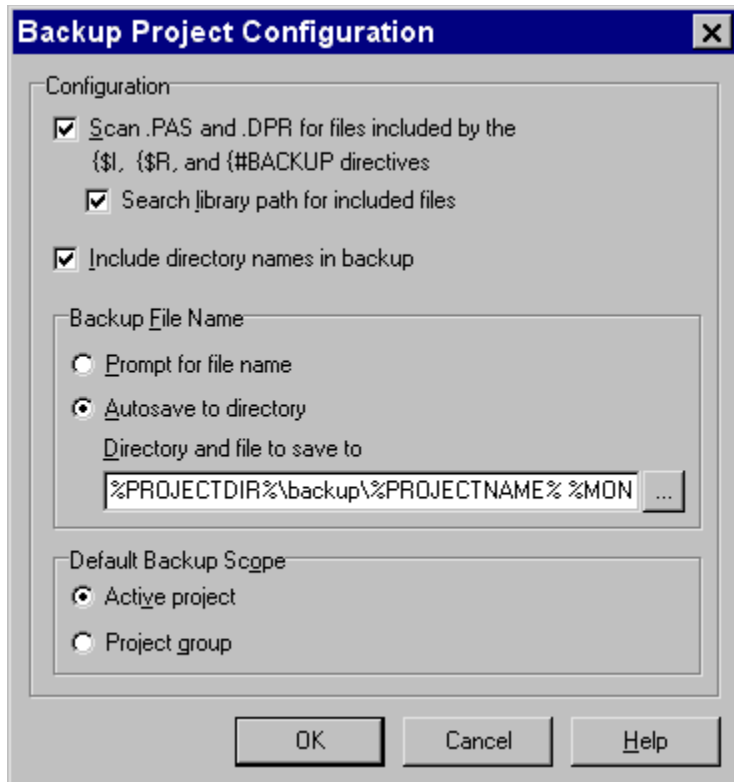
The project backup expert enables you to backup an entire project to a zip file. Always save all files in your project before running this expert or your latest changes will not be included in the zip archive. When this expert is executed, a list of all files in the project appears in the dialog. You can add or remove files from the list using the Add and Remove buttons. To backup the project, click the Backup button.

Quick-access options such as password protection for your archive, the backup scope, and an option to search for included files on the library path are available via the Options button. These options are not saved between invocations of the expert; you must use the Backup Project Configuration dialog described below to set the default options.



The default configuration options are set from the [GExperts Configuration](#) window. You can configure the backup project expert to scan all of the source files for resource files and other files which should be included in the backup. These files are indicated by the \$I, \$R, \$INCLUDE, and \$RESOURCE directives, and the included files can be anywhere in the project directory or the IDE/project library path. The backup can also default to backup the project group or just the current project.

If you have enabled the option to scan for included files, you can force certain files to be included in the backup by adding a command directive line to the interface of any unit in the project. For example, if you have a PAS file that always accesses a certain INI file and you want it included in the backup, just add a `{#BACKUP MYINI.INI}` line to the top of the file and from then on GExperts will include it in the backup. This expert has support for wildcards and paths in the filename such as: `{#BACKUP *.INI}`, and `{#BACKUP 'C:\Program Files\test.dat'}`.



The expert can prompt for a file name for each backup, or you can instruct the expert automatically backup to a specified directory and file without having to specify a file name each time. Various macros are available that allow you to specify the directory and filename used:

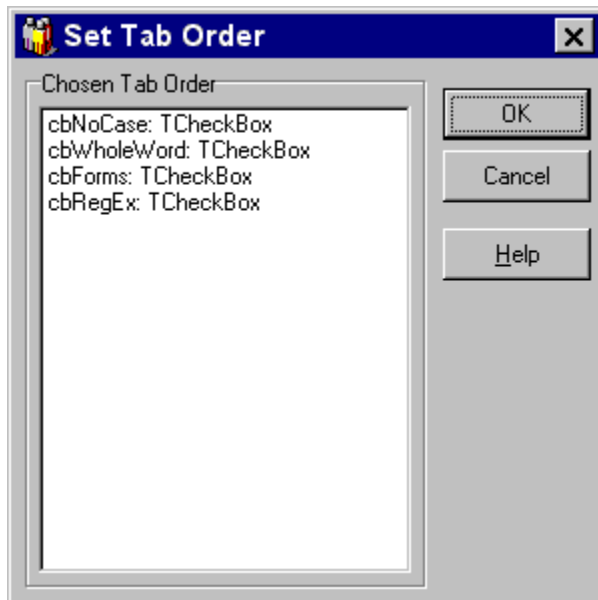
%PROJECTDIR%	The project directory
%PROJECTNAME%	The project name
%PROJECTGROUPDIR%	The project group directory
%PROJECTGROUPNAME%	The project group name
%YEAR%	The year, in four digits
%MONTH%	The month in two digits
%MONTHSHORTNAME%	The short month name
%MONTHLONGNAME%	The long month name
%DAY%	The day in two digit format
%DAYSHORTNAME%	The short day name
%DAYLONGNAME%	The long day name
%HOUR%	Hour in two-digit format
%MINUTE%	Minute in two-digit format
%SECOND%	Second in two-digit format
%USER%	The currently logged in username

Examples:

%PROJECTDIR%Backup\%YEAR%-%MONTH%-%DAY% will produce a backup file in a subdirectory called "Backup" off of your main project directory. The filename will consist of the year, month, and day the backup was made.

C:\Backups\%ProjectName%-%DAYSHORTNAME%-%DAY%.zip will produce a backup file named similar to C:\Backups\Project1-Wed-26.zip.

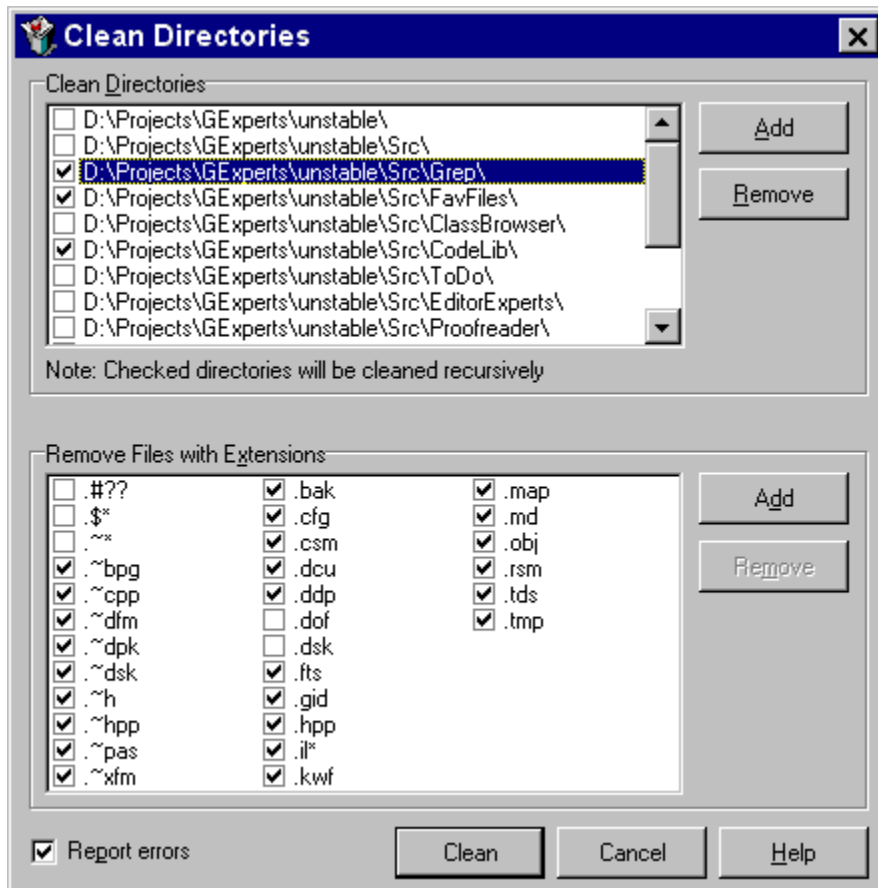
This expert enables you to define a form's tab order visually. To use this expert, hold down the shift key and click on the form's controls in the desired tab order. Then, select the Set Tab Order menu option and a dialog will appear as below. This dialog enables you to review the order of controls you selected to ensure you didn't make any errors. Finally, to set the tab order, press the OK button.



Notes: All of the controls you select must have the same immediate parent component, or the expert will issue an error. Also, this expert does not modify the TabStop property of your controls and will ignore components without TabOrder properties.

This expert enables you to quickly delete several types of non-critical files that can accumulate during development. When it is activated, a list of all directories that project files reside in will appear. Check the checkbox next to a directory to scan it recursively. You can then select which types of files to delete from those directories by checking the various checkboxes in the file extensions listbox. To start the cleaning process, press the Clean button. Once the process is complete, you will be notified how many files were deleted and how much disk space was reclaimed.

You can add and delete directories and extensions to be cleaned using the provided Add and Remove buttons. The Using the "Report errors" checkbox you can request that GExperts notifies you when files can't be deleted, such as when the files are read only or in use by applications.



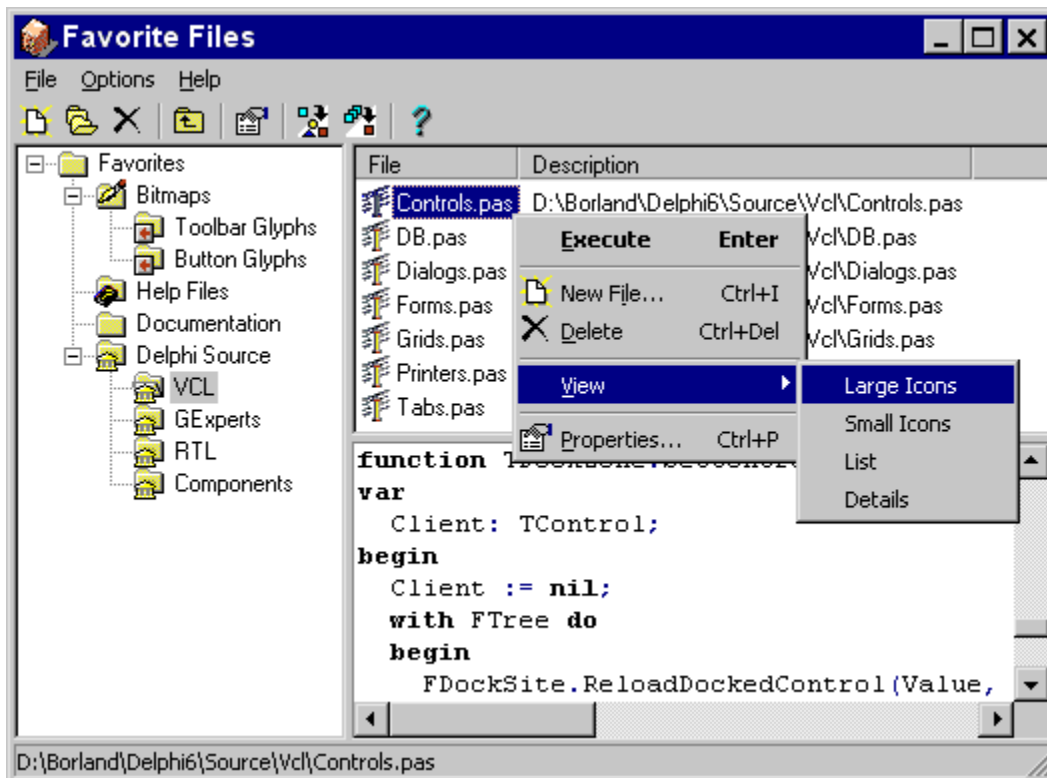
The Favorite Files expert provides a convenient way to organize commonly used files. You can create as many nested folders as you like, and each folder can be one of several types representing bitmaps, source code, help files, folders, glyphs, etc. To add a new folder, right click the folder pane and select the new menu. A window will appear prompting you to enter a folder name and type. The folder type is used to determine which icon is used represent the folder.

Similarly, you can add files to a folder by right clicking on the file pane and selecting new. An open file dialog will appear, from which you can select multiple files to add to the current folder. After adding the files, you can edit their properties by bringing up the property dialog. Each favorite item has a filename, a name, a description, and an execution type. The execution type determines how the manager deals with the file if it is double clicked. The execution types are as follows:

- IDE:** Load the file into the IDE
- Shell:** Use the ShellExecute API call to run the file with the associated program
- Custom:** Manually specify an application to use to open the file
- Project:** Load the file as a new project into the IDE

A file viewer is available in this expert to preview files as they are selected. It can preview several file types such as PAS, DPR, RTF, HTML, BMP, SQL, C, CPP, TXT, etc. To speed up this expert, you can hide the preview panel via the Options menu.

Tip: You can drag and drop files from Windows Explorer into the file listing to quickly add new entries.



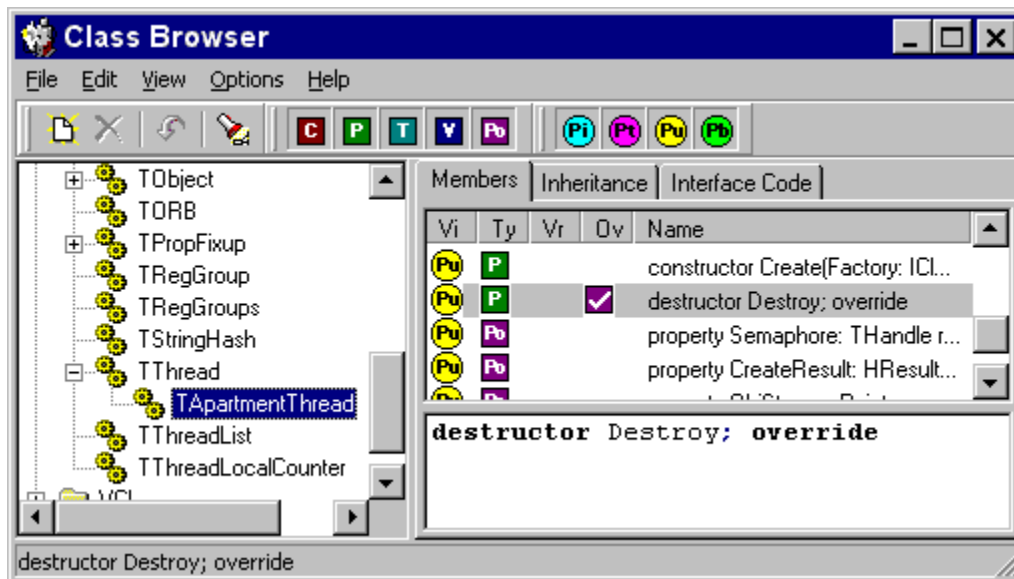
The class browser enables you to view the classes in your project or in external directories, such as the VCL classes. External class groups can be added using the add button on the toolbar or by selecting Add from the File menu. Your project classes are automatically added into the class browser when you open a project in the IDE. You can refresh the display of classes by pressing the refresh button.

Classes can be viewed as a flat list or tree-structured hierarchy. You can choose which method you prefer using the View menu. To find a specific class, choose Find. The details shown for the selected class include the class members, the inheritance tree (this works best after you have added the RTL and VCL directories to your class database), and the interface code for the selected class member.

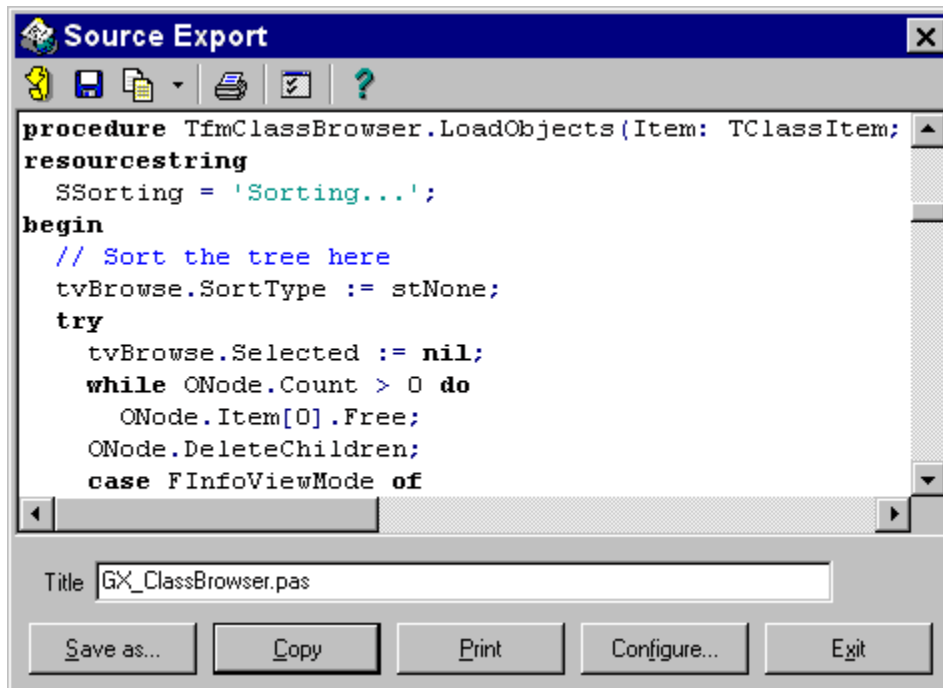
Two reports are available from the File menu. The first report prints the currently selected class and its methods. The second prints a hierarchical report based on the selected class. Both reports have print preview capability.

There is a filter toolbar to select which member types (properties, methods, etc.) and what visibility classes (Private, Protected, etc.) are listed in the Members tab. An options dialog provides access to the default filter settings, font settings, a setting to enable recursive parsing for classes, and other options.

You can jump to the source code for a class or method in the code editor by right clicking on the object and selecting the Goto menu option.



This expert enables you to save source code to HTML or RTF formats or to the clipboard with syntax highlighting preserved. You can export the whole file, the portion that is selected in the IDE, or you can edit the shown source to customize the output. The print button allows syntax-highlighted printing as well. You can also set the colors used for the syntax highlighting and the default clipboard format by clicking the Configure button.



Tip: This expert supports syntax highlighting of Object Pascal, C/C++, HTML, and SQL text files.

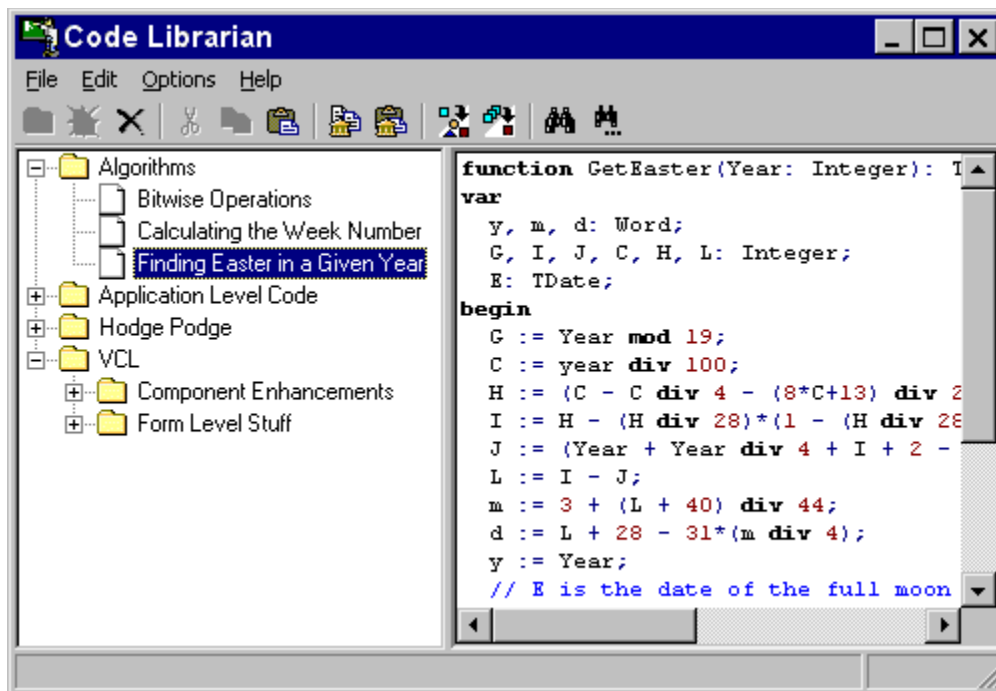
The Code Librarian enables you to organize and access often-used code snippets. It is an easy way to store the various code items that you use often, but do not fit into a general-purpose library unit.

The Code Librarian is composed of two panes. The pane on the left is the folder view that allows you to create nested categories to organize for your code snippets. You can rename, add, and delete folders, or move them using drag and drop. Folder titles can be edited by clicking a selected folder.

The Code Librarian supports copying code from the IDE into the code pane and vice versa.

You can search for keywords by clicking the Find button. Both the topic title and the code pane are searched.

The Options selection allows you to change the display fonts, layout mode, and the location of the Code Librarian snippet database. The syntax highlighting mode for the code snippet can also be toggled between none, Pascal, C++, HTML, and SQL from the context menu of the source editor.



Tip: A starter database of code snippets (shown in the screen shot above) that you can integrate directly into the Code Librarian is available on the [GExperts home page](#).

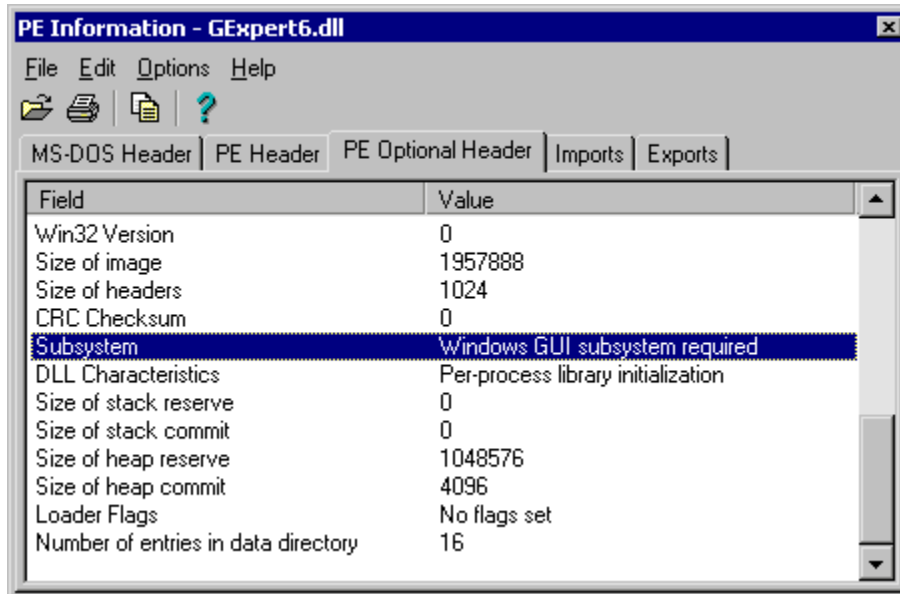
The ASCII Chart expert shows ASCII values for all of the characters in a given font. You can set various options by right clicking on the grid. You can choose to view the high or low half of the ASCII character range (0-127, 128-55) and choose between HEX and decimal number representations for a character using the buttons at the top of the window. You can also choose the display font and size for the ASCII chart. Several other options are available in the context menu for the ASCII chart.

Clicking a character in the ASCII character grid will copy that character to the edit box at the top right of the window. From there, you can copy any characters to the clipboard for use in your program.

The screenshot shows a window titled "ASCII Chart" with a standard Windows interface. At the top, there are controls for "Lo" and "Hi" (set to 16 and \$F), a font dropdown menu (set to "MS Sans Serif"), and a font size spinner (set to 10). The main area is a grid of 16 rows and 16 columns. Each cell contains a hex value, a name, a decimal value, and a character. The characters shown are the standard ASCII set from 0 to 127.

Lo	Hi	16	\$F	MS Sans Serif 10											
00	NUL	10	DLE	20		30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(38	8	48	H	58	X	68	h	78	x
09	TAB	19	EM	29)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	!

The PE Information expert enables you to examine the contents of a Portable Executable (PE) file. Almost all Win32 EXEs and DLLs are PE format files. Many other files are also PE files, such as Borland packages (DPL/BPL), Active X controls (OCX), etc. This expert supports IDE docking and drag and drop of files from Windows Explorer.

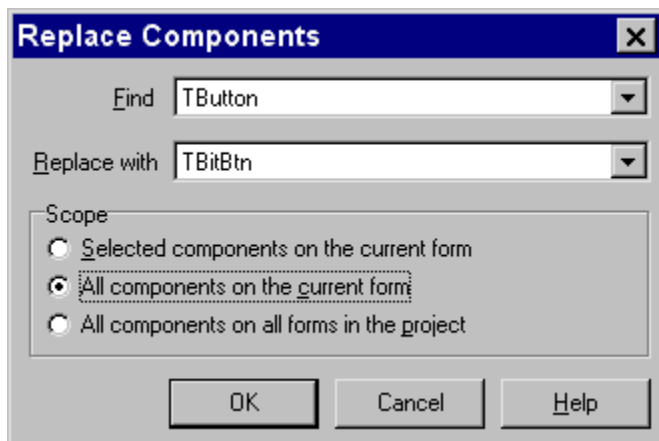


To view the contents of a PE format file, click the Open button and select the file or drag and drop a file from your file manager. There are five tabs in the expert as follows:

- **MS-DOS Header:** All windows programs contain the MS-DOS header, sometimes referred to as the MS-DOS stub.
- **PE Header:** The portable executable header contains data such as what machine the binary is targeted for (Intel 0x86, Alpha, MIPS, etc.).
- **PE Optional Header:** Contrary to its name, this header is required in PE files. It contains information regarding the subsystem required, the version number of the OS required, etc.
- **Imports:** This section contains a list of all files required by this file to run, including Borland packages. Selecting a file will cause a list of all functions imported from the selected file to appear.
- **Exports:** Exported functions are the ones that are made publicly available for other applications call. This tab is mostly of use when examining DLLs or packages.

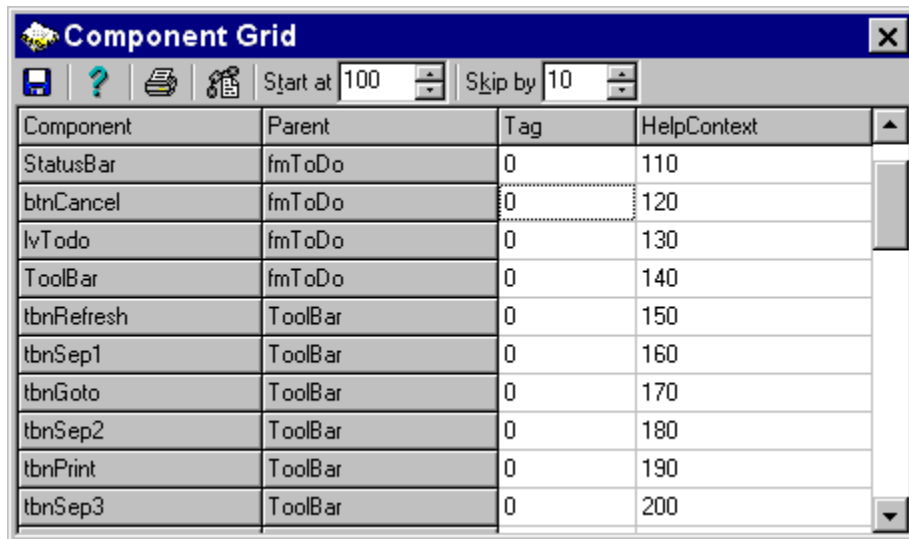
This expert enables you to easily replace components on a form or in a project. For example, have you ever wanted to replace all TBitBtn components in a project with TButton components? This expert does its best to preserve the properties of the original component. To use this expert, first select the component you want to replace, followed by the component you want to replace it with. Then click OK to have the expert perform its magic. If you select the component to replace in the IDE before calling up this expert, you won't have to select it in this dialog box.

Note: This expert can not preserve properties that reference other components, TCollection type properties, and other more complex properties such as TStringList and TFont. It should preserve ordinal, string, and other simple types. If you use this expert to replace non-visual components, you will need to save the form before the components will show again at their previous location.



This expert enables you to edit the Tag and HelpContext properties of all components on a form at once instead of having to individually select components. You can sort the component list by name or parent by clicking the relevant column header. When you want to save your changes, press the Save button.

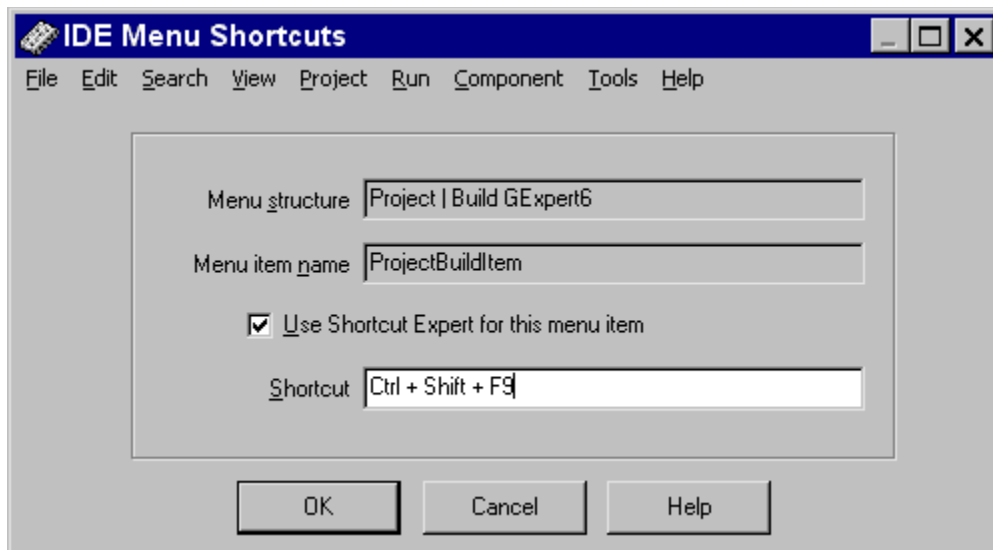
The toolbar controls provide support for renumbering help context IDs starting from a given number and skipping by another number and the ability to print a simple report listing all of the components on the current form.



The screenshot shows a dialog box titled "Component Grid" with a toolbar at the top. The toolbar includes icons for Save, Help, Print, and Refresh, followed by two numeric input fields: "Start at" with the value 100 and "Skip by" with the value 10. Below the toolbar is a table with four columns: Component, Parent, Tag, and HelpContext. The table lists ten components with their respective parent names, tags, and help context IDs.

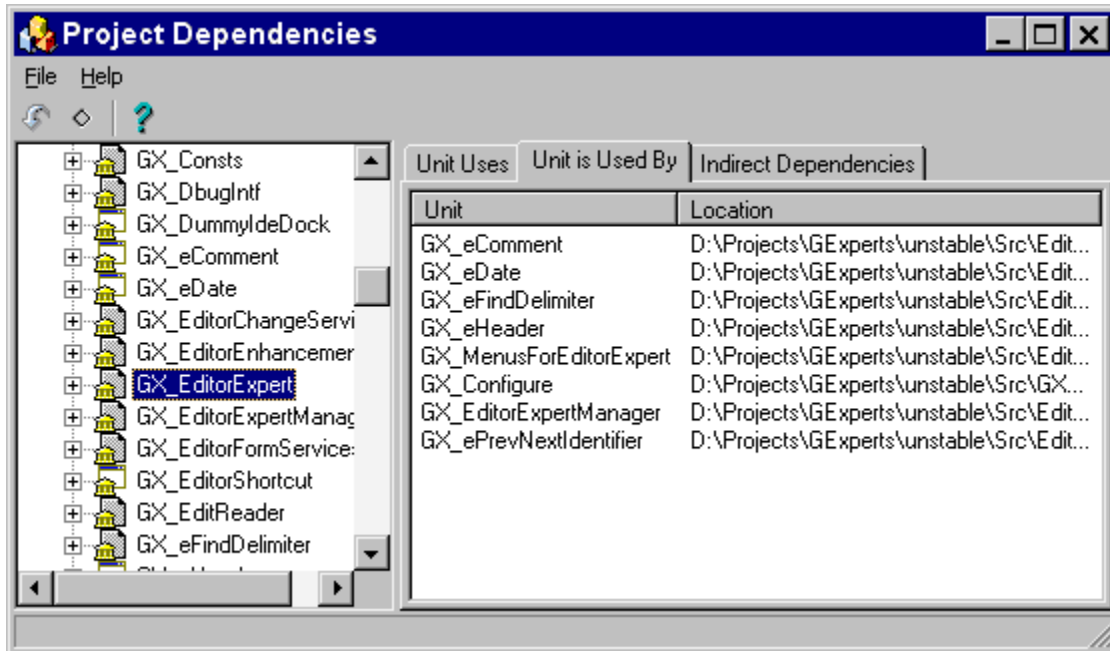
Component	Parent	Tag	HelpContext
StatusBar	fmToDo	0	110
btnCancel	fmToDo	0	120
lvToDo	fmToDo	0	130
ToolBar	fmToDo	0	140
tbnRefresh	ToolBar	0	150
tbnSep1	ToolBar	0	160
tbnGoto	ToolBar	0	170
tbnSep2	ToolBar	0	180
tbnPrint	ToolBar	0	190
tbnSep3	ToolBar	0	200

This expert enables you to configure any of the IDE menu shortcuts. To change the shortcut of a menu item, select the appropriate menu item, click "Use Shortcut Expert for this menu item" and type the desired shortcut into the Shortcut edit box. If you find your chosen shortcut does not work, it is likely because the shortcut is reserved by the code editor or another menu item, and you'll need to select a different, unused shortcut. This expert can not change the shortcuts for the GExperts menu items; for this purpose, use the [GExperts Configuration](#) dialog.



The project dependency expert enables you to see what units a particular unit uses, and in turn what units use a particular unit. When this expert is activated, it parses all of the current project's source code for uses clauses and builds up a list of dependencies. To view the dependency information for a particular unit, click on it in the left pane. The right pane will contain the dependency information. Indirect dependencies are units that are used by used units of a particular unit.

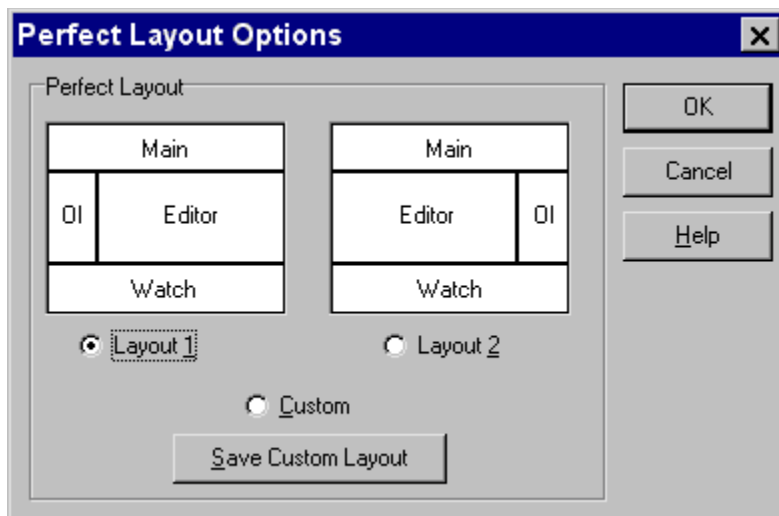
You can refresh the dependency information at any time by clicking the refresh button on the toolbar and you can sort the file listing by clicking on the column headers.



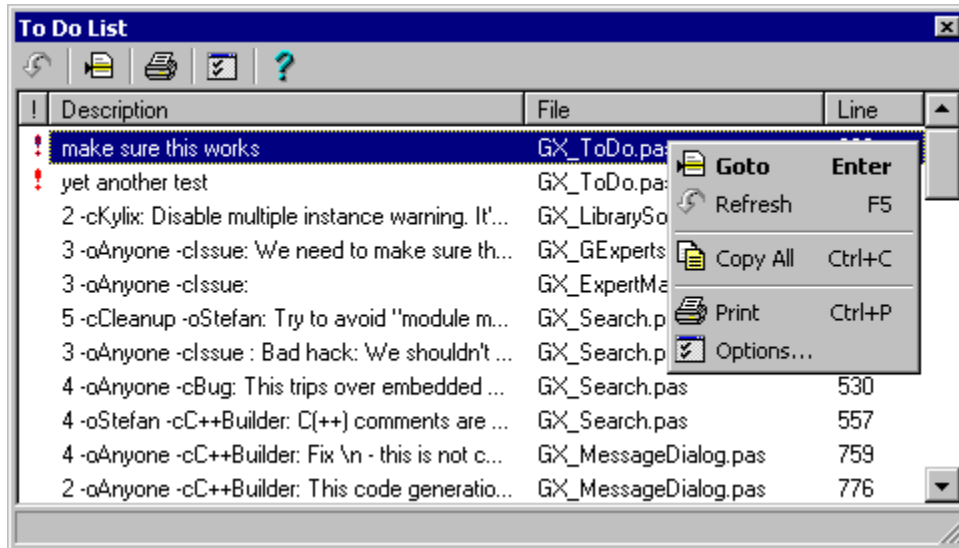
The Perfect Layout expert repositions the IDE's windows so that they are located exactly where you want them. This expert supports two pre-defined layouts, which can be selected via the GExperts Configuration dialog.

You can also create a custom layout by positioning the IDE windows as you want them, and using the Save Custom Layout button. You can change your custom layout at any time by repeating this process.

Note: This expert is deprecated and does not work well with docked window configurations. If available, use the named desktop saving features of your IDE instead.



The To Do List helps you organize a list of items in your source that need special attention. You can click a column header to sort by any column in the list.



To add new to do items, type comments in your code such as:

```
{#ToDo1 Rewrite this code to work under NT}  
//#ToDo2 Add support for Oracle here later
```

The to do keywords (such as ToDo1, ToDo2, etc.) can each have an assigned priority of High, Medium, or Low. You can also add new keywords with associated priorities using the configuration dialog. By adding a token called TODO this expert can read the standard to do entries generated by your IDE. This expert can scan open project files, all files in the current project, or a complete directory for to do items. It may be helpful to create an IDE Code Template to quickly create new to do items while you are coding.

Note: You can not configure the To Do list to use tokens that begin with the dollar sign (\$) because this conflicts with Object Pascal compiler directives.

The Code Proofreader is a powerful tool unique to GExperts. It can correct your typing in the code editor as-you-type based on a list of custom dictionary words, a list of AutoCorrect entries, or context sensitive information supplied by the Object Pascal compiler.

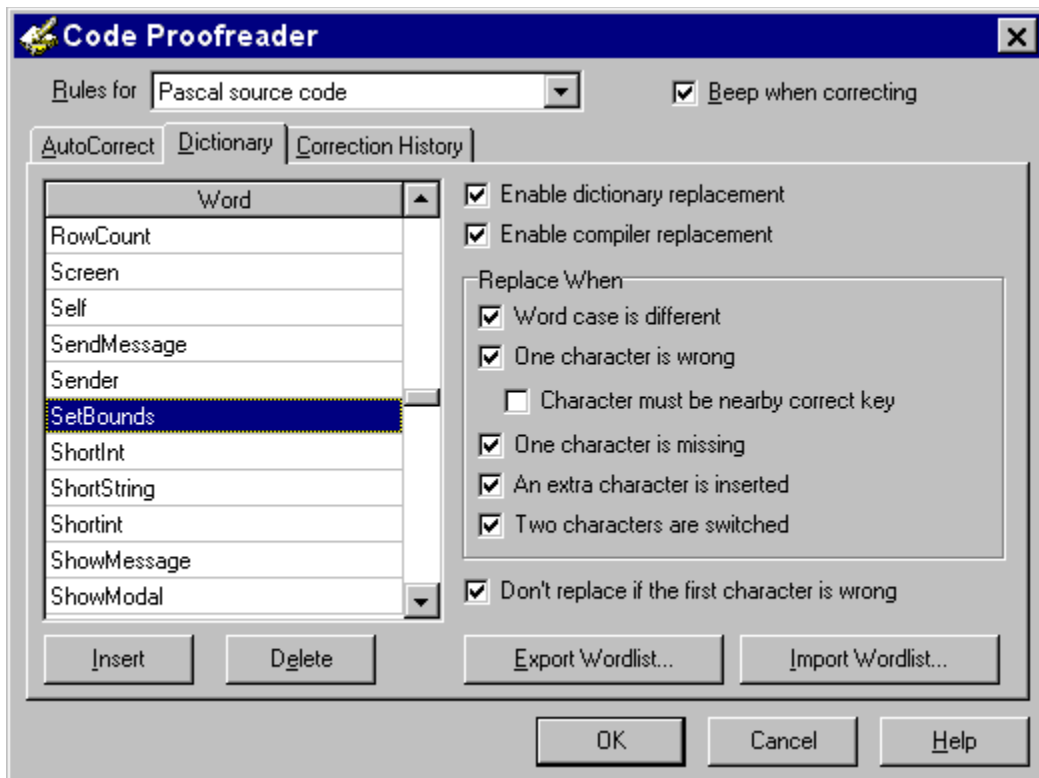
The corrections can be customized per source file type and even for different syntax entities within a file by using the dropdown box at the top of the window. You can import or export dictionary words from/to ASCII files that contain one word per line, or you can insert or delete single words using the provided buttons.

AutoCorrect entries wait for you to press a certain sequence of keys, and then replace that string with another string you define. This can be used to correct typing mistakes you commonly make, or it can be used to expand simple macros. AutoCorrect entries support usage of the "|" character to determine the position of the cursor after the correction takes place, similar to the IDE's Code Templates.

The compiler replacement option allows GExperts to interface with the Object Pascal compiler to generate a list of context-sensitive matches for the current cursor position. Be careful because this option causes the IDE to attempt to recompile parts of the project each time you press a non-alphanumeric keystroke in the source code window, and can slow down your coding on older machines. This option is useful to auto-correct identifiers to match the case they were declared in. Note that compiler replacement is not supported under C++Builder.

Several options allow you to tune the dictionary and compiler correction. You can choose to correct or not correct based on word case; errors based on incorrectly hitting nearby keys on the keyboard; and missing, inserted, or switched characters in the word. Finally, the Code Proofreader can issue a beep after correcting to make sure you know it was activated.

The Correction History tab on this form shows the last few corrections that were made, so you can review exactly what the Code Proofreader has changed, and what rule triggered the correction, since this isn't always obvious.

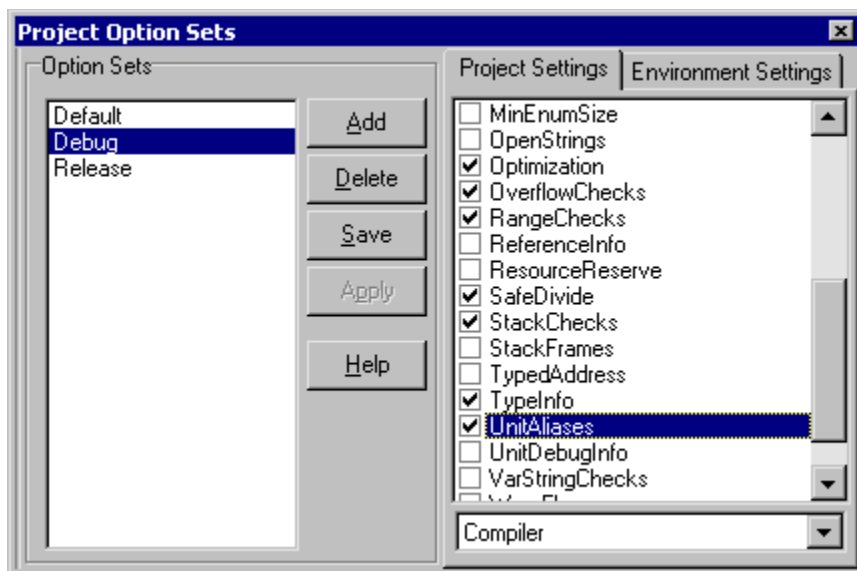


With this expert, you can save off sets of project options and recall them at a later time. Along with the project options, a few environment options can be saved as well. This expert supports IDE docking.

To save a given set of project options, you should first execute the Project Option Sets expert and type a set name into the edit box. Click Add to create a new named option set (such as "Debug" or "Release"). Next, open the IDE's Project Options dialog and verify that the settings you wish to save are set correctly. Note that you can open the Project Options dialog from the right-click menu of settings listbox. Let the IDE store those settings by confirming the Project Options dialog. Next, check the settings on the Project Settings or Environment Settings tabs that you want to be saved. Settings you don't check here are not affected when you load the stored settings later. Finally, use the Save button to save the settings. Your saved sets are stored in an OLE Structured Storage file called GXPoSetsX.bin. The storage location of this file is defined in the [GExperts Configuration](#) dialog. To load a previously saved set of options, click the set name in the list and click the Apply button.

Note: The option name lists match those used internally by the IDE, so it may take some searching to determine which option name matches a given setting in the project options dialog. The listbox provides some popup hints to help you, and you can use the combobox below the list to filter the available options based on certain categories.

Note: Delphi 5.00 has serious bugs that prevent most of these settings from working. You must upgrade to Delphi 5.01 or greater for full functionality in this expert.



This expert allows you to select one or more components on a form and copy the appropriate Object Pascal code to the clipboard that would create those same components at runtime with any modified property settings intact. From the configuration dialog, you can instruct it to include a copy of the original component so you can later paste it back on a form.

Note: This expert will not work with binary properties such as bitmaps, but it can create commented or uncommented property creation code so you can fix this by hand. Also, it will not work with some components with subcomponents such as TTreeChart and third party tabsheets that do not descend from TTabSheet.

