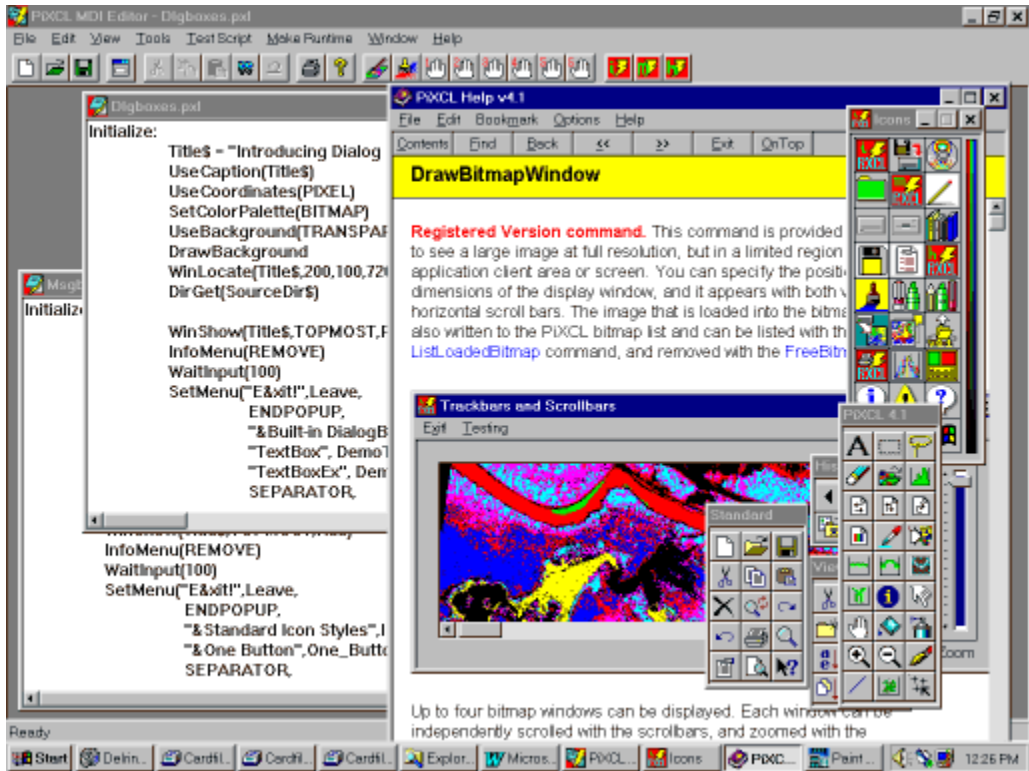


Introduction

You can easily create your PiXCL application programs (*.PXL) within one Integrated Development Environment, or IDE, with all of the editing operations, command reference Help, testing with the PiXCL interpreter and final EXE file building. An example screen shot is below, showing the multiple code windows, context sensitive help and one of the PiXCL Helper applications. Click on the screen shot below to see a short description of each region.



Since the editor is a multiple document interface, or MDI, application, you can have as many documents open as you need. This can be very useful when you need to cut and paste code fragments between PXL files. You can select a working file directory for your PXL sources, and this path is stored in the Windows registry, so that editor always starts in the last used directory. The current editor window dimensions and position are also stored in the Registry, and are accessed when the editor is started each time.

The PiXCL MDI Editor also provides user defined access to other useful tools. You can start an icon editor such as CursorArts IconForge™ (not included in PiXCL), a paint and draw tool such as Microsoft Paintbrush™ (part of Windows) or JASC Paint Shop Pro™ (not included in PiXCL), or the paint program of your choice, as well as any six additional helper applications you may need or create yourself. PiXCL provides two helper applications, with source code.

MDI Editor Toolbar Buttons

The PiXCL MDI editor displays a toolbar as shown in the image below. These buttons provide fast access to the various functions that are also available from the traditional menubar. There are actually two toolbars that are individually dockable. For the **Standard** set, buttons include the file access, edit buttons, find text and print buttons. For the **Developer Tools** bar, the buttons are for PiXCL code testing, EXE building and command reference help.



If you move the mouse over a toolbar button, a tooltip message is displayed that describes the function of the button. In the above image, the tooltip for the [Find Text](#) button is shown active. Click on the buttons in the above image to see a description of the function.

In addition, the toolbar is dockable, which means you can move it to anywhere along the vertical and horizontal main window frames by clicking in the background area and dragging the toolbar to the position you want. You can also set the toolbar to a floating mode by dragging it into the MDI Editor client area. Toolbars can also be removed with the **View:Standard Toolbar** and **View:Developer Toolbar** menu items.

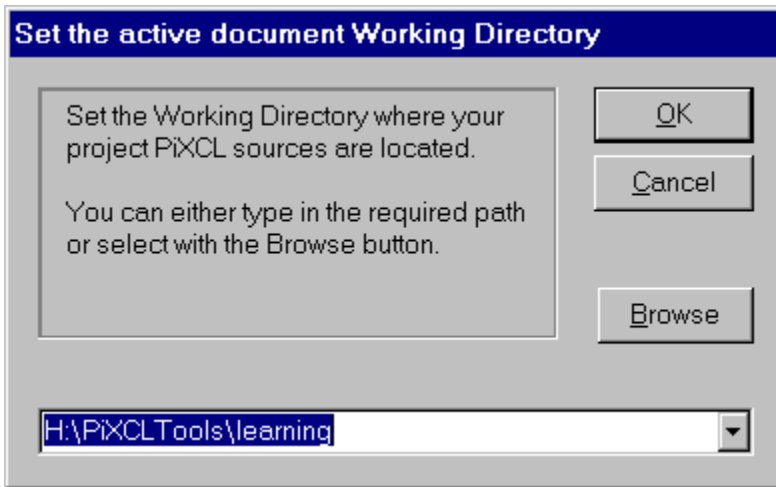
Setting the working file directory



When you start a PiXCL development project, the first thing you need to do is set a working file directory, by clicking on either the [Working Directory](#) button, or selecting the [Files: Set Working Directory](#) menu item.

This displays the dialog box below, with the current working directory setting read from the Windows registry shown as the default in the combobox. The current and previous three working directories are also stored in the Registry, and can be displayed in the combobox dropdown list.

To set the working directory, you can select from the combobox list, click the Browse button to locate a directory that you want, or type the desired directory into the combobox edit area.



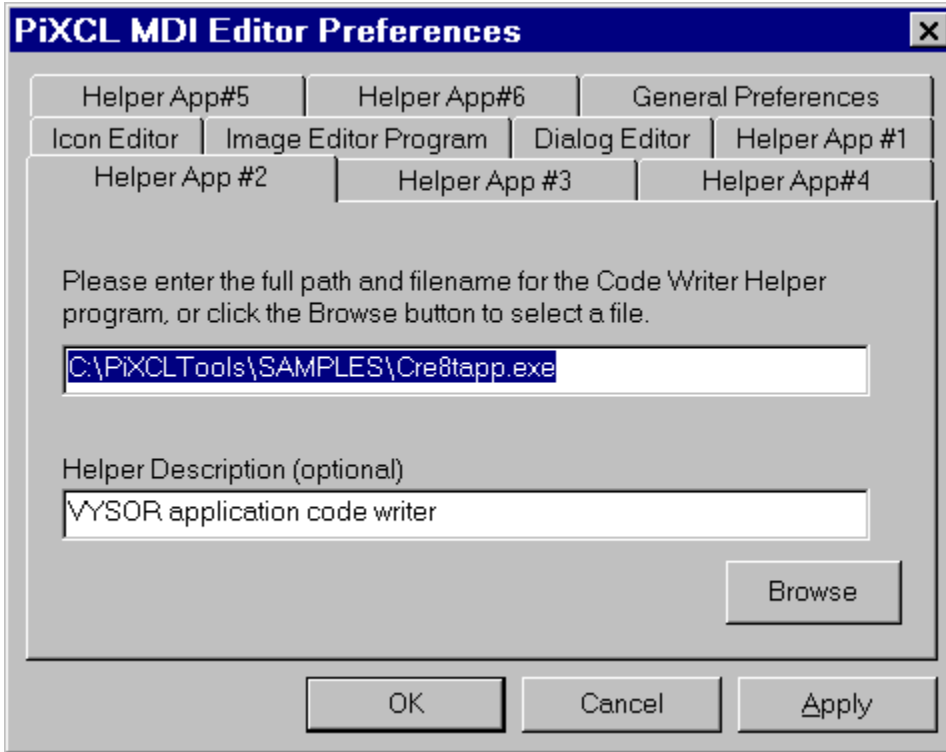
Click the OK button when done you are now ready to start coding a new file, or open an existing PXL file or files.

To create a new and fully functional skeleton source file, you may find that the **cre8tapp.exe** Helper will save you lots of time. It is installed as the default Helper App#2. The PiXCL source code **cre8tapp.pxl** is supplied in the **..\samples** directory so you can see how it works, and modify it to suit your own needs.

See also [Setting Optional Tool Preferences](#)

Setting Optional Tool Preferences

In the **File:Preferences** menu item, an ten page Preferences dialog appears asking for the location of the tools of your choice. You can browse your disks and directories to select these tools. Once you select a file, the path\name and optional description are written into the Registry.



The above image shows a selection for Helper App #2, in this case, the code writer assistant. To find the desired tool, press the Browse button and select the EXE file. The tool selection is written to the edit box. You can add a single line that describes the tool if you want. When you click the Apply or OK button, all tool selections and descriptions (if they exist) are written to the Registry.

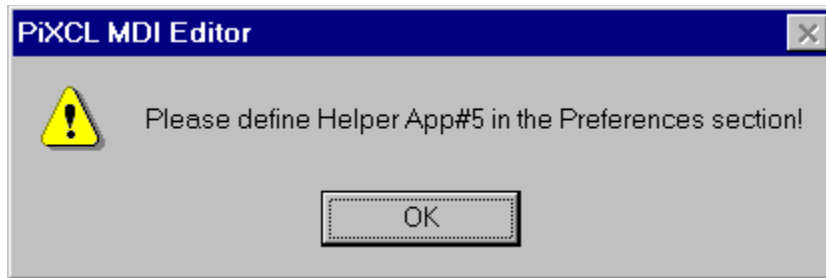
Two Helper Applications come with PiXCL, and are located in the installation **..samples** directory. Both the EXE and .pxl sources are provided, so you can customise them as needed. These Helper applications are

- **editools.exe**: the default Helper App#1, that aids the selection of messagebox icons and toolbar buttons; and
- **cre8tapp.exe**: the default Helper App#2, that assists you to write a working skeleton program, and to paste in DialogBox sample code.

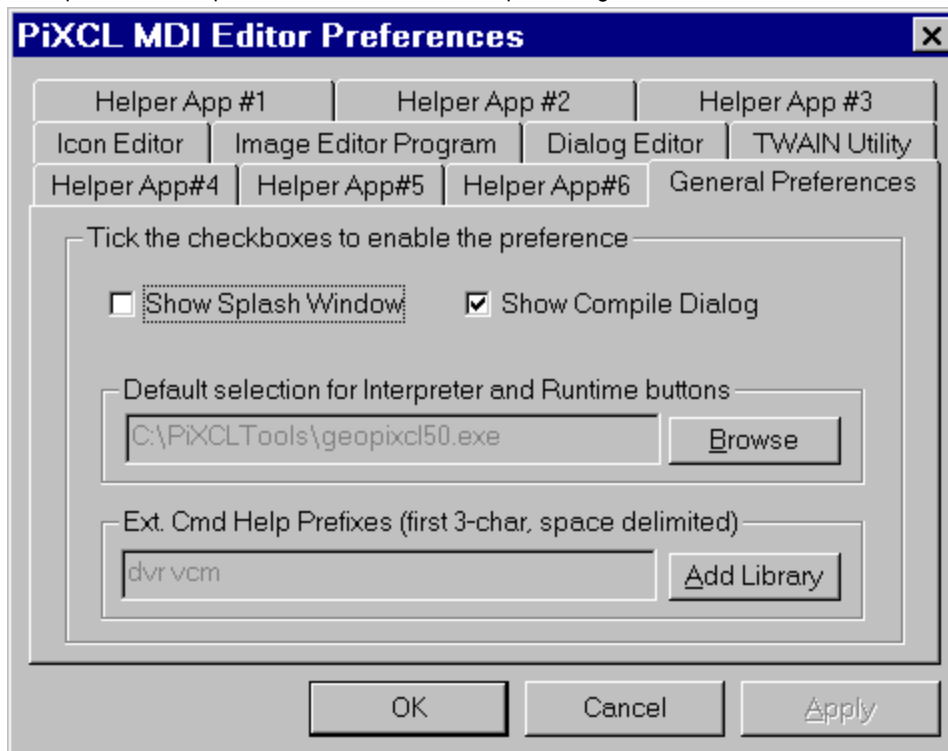
A dialog editor is included with PiXCL 5. This is also a Helper application but written in compiled code. It works in a similar fashion to **editools** and **cre8tapp**, such that according to your input, it generates PXL code and passes it to the Clipboard, from where it is pasted into your new application.

Tools are immediately available: there is no need to restart the MDI Editor.

If no Helper application has been defined then a MessageBox, shown below for Helper App#5, is displayed.



The General Preferences tag provides a way to turn off the splash screen and set the default PiXCL 4 or PiXCL 5 or geoPiXCL Interpreter and Compiler, and also to show the compiler dialog or not.



To select the interpreter, press the Browse button. A file select dialog is presented, and the interpreter you select is written to the read-only edit control, and stored in the Registry.

Using an icon editor program



PiXCL has 17 built in icons that can be edited using a suitable icon editor program. You may like to change the default icon in your EXE files.

In the **File:Preferences** menu item, a Preferences dialog appears asking for the location of the Icon editor of your choice. You can browse your disks and directories to select the Icon editor. Once you select a file, this is written into the Registry.

When you click on the Icon Editor toolbar button, the Registry is read and the file (if it exists) is executed. You can change the location or name of the icon editor at any time.

If an icon editor has not been defined, a message box appears when you click the Icon Editor toolbar button.

See sample program [shoicons.pxl](#) for a display of the icons built into PiXCL.

Starting a Paint and Draw Tool



Click this button to start a specified image editor tool, for example MS-Paintbrush, or for geoPiXCL users, the geoPiXCL display engine. You can use an editor program to create bitmaps that are needed for your PiXCL programs, such as button bitmaps and similar.

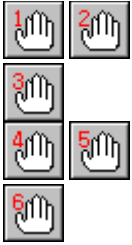
There is no default setting for an image editor program. If you want to use MS-Paintbrush, you will have to install it if you have not already done so, as it is an optional Accessory selected during Windows installation.

In the **File:Preferences** menu item, a Preferences dialog appears asking for the location of the image editor of your choice. You can browse your disks and directories to select the image editor. Once you select a file, this is written into the Registry.

When you click on the Image Editor toolbar button, the Registry is read and the file (if it exists) is executed. You can change the location or name of the image editor at any time.

If no application is installed, or the specified application cannot run, a message box will appear asking you to correct the situation.

Starting a Helper Application



Click these buttons to start a specified helper application, such as editools.exe (Helper #1) and cre8tapp.exe (Helper #2) which are supplied with PiXCL.

In the **File:Preferences** menu item, a Preferences dialog appears asking for the location of the helper applications of your choice. You can browse your disks and directories to select the tool. Once you select a file, this is written into the Registry.

When you click on any of the Helper application toolbar buttons, the Registry is read and the file (if it exists) is executed. You can change the location or name of the helper application at any time.

If you want to display another help file with a Helper button, use the following syntax, or just select the file.

path\application_help.hlp

If you have an HTML document with a Helper button, just select the file as shown below.

path\document.htm

and the current default browser will be started.

If you want a specific non-default browser, you have to set the full path and .exe name, as shown below.

path\netscape.exe path\document.htm

In general, a Helper application command line accepts either document names, executable, or executable with an argument list. If just a document name is specified, PiXCL searches for the associated executable for that document type, starts the executable and loads the document.

If no Helper application is installed, or the specified application cannot run, a message box will appear asking you to correct the situation.

Testing a PiXCL source file

Once you have written your PiXCL application code, you need to test it with the interpreter to verify the correct syntax and operation. You must have saved the file first. The default file extension is .PXL.

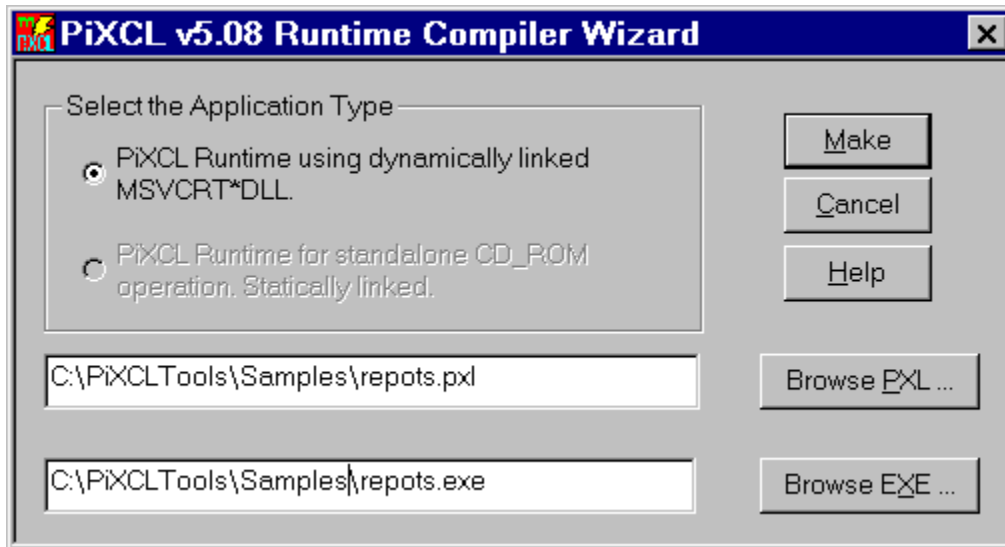


Click the button or select the Test Script menu item. A dialog box is displayed telling you the name of the PiXCL script, and an OK or Cancel button. Click OK to run the program, or Cancel to return to the editor.

Building a PiXCL executable



Click this button to compile a PiXCL Runtime EXE application. You will need to answer several questions on file names and locations, and optionally modify the run appearance (the window “style”) of the final EXE, using the Runtime Builder Wizard, shown below. It has its own help. See also the General Preferences property sheet when you can set the compile dialog to appear as below, or use default settings to compile the EXE without prompting for input.








Working with Text

All the usual text-editing functions are available within the MDI editor. If you are familiar with NotePad, you will see that the MDI editor is basically similar, plus the extra buttons for the PiXCL coding, help access and command testing.



These buttons are grayed until the current document has been edited in some way.

In order, left to right, these buttons are:

-  **Cut**: The section of code selected will be cut from the document and placed in the clipboard. Equivalent menu item is [Edit:Cut](#) or keyboard [Ctrl-X](#).
-  **Copy**: The section of code selected will be copied from the document and placed in the clipboard. Equivalent menu item is [Edit:Copy](#) or keyboard [Ctrl-C](#).
-  **Paste**: The contents of the clipboard are copied into the document at the current cursor position. Equivalent menu item is [Edit:Paste](#) or keyboard [Ctrl-V](#).
-  **Find Text**: The section of code selected appears as the default selection in the Find Text dialog box. Equivalent menu item is [Edit:Find](#).
-  **Undo**: Click this button to undo the last edit operation.

Finding specific characters or words

To find specific characters or words

- 1 Highlight the text in the document.
- 2 On the Search menu, click Find, or click the Find Text button.
- 3 In the Find What box, type the characters or words you want to find.
- 4 Click Find Next.

Left Mouse Selection Tips

1. double click on a word to select it. Delimiters are spaces, tabs, hyphens and brackets.
2. triple click to select an entire line.
3. click in the text, then shift click to select all the text between the markers.

Tip

For Help on an item, click at the top of the dialog box, and then click an item.

Deleting, cutting, copying and pasting text

- To delete characters to the left of the insertion point, press the BACKSPACE key.
- To delete characters to the right of the insertion point, press the DELETE key.
- To delete words, select them, and then press the BACKSPACE or DELETE key.
- To cut text so you can move it to another location, select the text. Then, on the Edit menu, click Cut.
- To copy text so you can paste a copy of it in another location, select the text. Then, on the Edit menu, click Copy.
- To paste text you have cut or copied, place the insertion point where you want to paste the text. Then, on the Edit menu, click Paste.

Getting PiXCL command reference Help



To bring up specific command help

- 1 Highlight the PiXCL command in the code. Click and drag or double click the command.
- 2 Press the **F1** key, the **Help** button or **menu** item. This will start the PiXCL command reference at the command specified. You can cut and paste from Help files into your PXL code.

Printing



You can print the current PiXCL source file by clicking this button. You can of course print a PiXCL source file with any other text editor or word processor.

New PiXCL Script button:



Click on this button to create a new text document window. You select the title when you first save the file.

Open File button:



Click this button to display the standard File Open dialog with a list of the available PXL source files in the current working directory.

Save PiXCL script button:



Click this button to save the currently selected open document. You will be prompted for a name if you have not previously saved the script.

Set Working Directory button:



Click this button to select a new working directory. The directory you select is stored in the Windows registry, and is the default when you start the PiXCL MDI Editor.

Edit: Cut function



Cut selected code from the document. Equivalent to menu [Edit:Cut](#) or keyboard [Ctrl-X](#)

Edit: Copy function



Copy selected code to the clipboard. Equivalent to menu [Edit:Copy](#) or keyboard [Ctrl-C](#).

Edit: Paste function



Paste clipboard contents into the. Equivalent to menu [Edit:Paste](#) or keyboard [Ctrl-V](#).

Edit: Find Text function



Selected code appears as default in the Find Text dialog box. Equivalent menu [Edit:Find](#).

Edit: Undo function



Click this button to undo the last edit operation.

Test PiXCL Script button

Once you have written your PiXCL application code, you need to test it with the interpreter to verify the correct syntax and operation. You must have saved the file first. The default file extension is .PXL.



Click the button or select the Test Script menu item. A dialog box is displayed telling you the name of the PiXCL script, and an OK or Cancel button. Click OK to run the program, or Cancel to return to the editor.

Make PiXCL Runtime button:



Click this button to compile a PiXCL or geoPiXCL EXE application. You will need to answer several questions on file names and locations.

Display PiXCL Command Help:



Click this button to display the PiXCL Command Reference Help file.

To bring up specific command help

- 1 Highlight the desired PiXCL command in the code
- 2 Press the F1 key or right click and Select the **What's this** menu item. This will start the PiXCL command reference at the command specified. You can cut and paste from Help files into your PXL code.

Icon Tool button:



PiXCL has 17 built in icons which can be edited using a suitable icon editor program. You may like to change the default icon in your EXE files.

Start Image Editor button:



Click this button to start an image editor. You can use an image editor to create bitmaps that are needed for your PiXCL programs, such as button bitmaps and similar.

Helper Application Buttons:



Click these buttons to start your helper applications, such as **editools.exe**, **cre8app.exe**, or other utilities such as HTML or WinHelp document builders.

Print Button:



Click this button to print the current PiXCL source document.

About Button:



Click this button to display this help file.

Option Menu and Toolbar buttons

These are the menu items and buttons that control the operation of the MDI editor. The toolbar can be turned off if you want, but most users will find editing code is faster with the toolbar present.

Multiple PiXCL code windows

The MDI editor can open more than one PiXCL source code file, making cutting and pasting between files a simple task. All the standard text search and replace functions are available.

The MDI editor can also be used for any non-formatted text file. That is, if NotePad can edit a file, then that file can be edited in the PiXCL MDI Editor.

Context Sensitive Command Help

Access into the PiXCL help file is integrated within the MDI Editor. Highlight the command and press F1 or right click and select the **What's this** menu item, and the specific Help topic will be displayed. You can set the Help window to be always on top if desired.

Toolbar and Icon Helper Application

One of the supplied helper applications that assists you in developing the code for toolbar and toolwindow button bitmaps, and messagebox icons. It also provides color selection,

The helper generates specific PiXCL code fragments that are copied to the clipboard, where you paste them into your code with a right mouse click.

The Windows Taskbar

This is the standard WindowsTaskbar as seen on your PC.

Code Writer Helper Application

One of the supplied helper applications that guides you in creating a working program skeleton with custom title, toolbar buttons, statusbar and other program options.

A skeleton code file is customised and written to your current working directory, and dragged and dropped from the Helper into the MDI Editor client area. The PiXCL source code for this Helper is included, and can be customised as needed.

