Power Toolz Code Manager

Quick Start Guide

Add a new folder to code tree.   You will be prompted for a folder name.

Add a new code snippet to the selected folder.   An edit box will appear so you can include a description.

Search for a keyword.   You will be prompted for a keyword to search for.   Any folder   or snippet title containing that keyword will be selected.   You can continue your search by pressing the ALT + F3 key.

Install library from disk.   When pressed the program will scan all floppy drives for PTCM libraries.   You will then be prompted for the libraries to install.

Opens the AutoDoc window. AutoDoc is a utility that assists in   documenting routines.   In the VB environment open the edit window of the routine you wish to document and press this button or press Alt + F12.   You will be prompted for information such as routine parameter definitions, purpose of the routine, and return values (if any).   This information along with your name, date, version, copyright, and routine name will be converted into a standard header that will *automatically* be inserted into the routine you are in.

Exit PTCM.   Pressing Alt + F9 or double clicking on the control button also exits PTCM.

Paste data on clipboard into the code window.

Attach a form to code snippet.   You will be presented with a dialog box that allows you to select a form.   The program makes a copy of this form and attaches it to the code snippet.   When you add this snippet to your program the selected form will also get added.

Note: A master copy of all attached forms can be found in the PTCM working directory.

Capture selected text and place it in the code window.   This feature allows you to simply highlight the desired text instead of copying it to the clipboard.

Save all changes or additions to disk.

Cancels or aborts the present operation.

Toolbar Icons:

The toolbar icons serve as destinations for code you have stored in PTCM libraries. You simply drag a folder or snippet from the code tree to any of the following icons:

Sends the code to an open Visual Basic code window.

Places code on the clipboard.

Delete the code from the tree and stores it in an ascii trash file

Function keys:

The following system wide hotkeys are enabled when you run Power Toolz Code Manager:

F1   Open the on-line help file for PTCM or for the presently highlighted code snppet or folder.
ALT + **F10**   Exit PTCM.
ALT + **F11**   Open the resource monitor.
ALT + **F12**   Open AutoDoc.
ALT + **F3**   Opens the search window or continues a search in progress.

Menus:

PTCM is equipped with context sensitive menu system as well as standard pull down menus.   To use the context sensitive menus simply click the right mouse button any place on the PTCM window.

Basic Program Operation

Power Toolz Button

Power Toolz Code Manager is designed be a powerful utility that is easily accessable at all times.   For this reason we have chosen to make PTCM appear as a small button on your screen along with a standard icon which appears at the bottom of the screen.

The Power Toolz button will always be on top of all other windows and icons.   To access PTCM simply double click on the Power Toolz button and the code window will open.   To move the Power Toolz button, drag the Power Toolz button to the desired location.   This position will be saved so that in the future the Power Toolz button will remain in the same spot.

Toolbar Buttons

The following buttons are visible in the normal operating code window:

Add a new folder to code tree.   You will be prompted for a folder name.

Add a new code snippet to the selected folder.   An edit box will appear so you can include a description.

Search for a keyword.   You will be prompted for a keyword to search for.   Any folder or snippet title containing that keyword will be selected.   You can continue your search by pressing the F3 key.

Install library from disk.   When pressed the program will scan all floppy drives for PTCM libraries.   You will then be prompted for the libraries to install.

Display code window.   The code window is used to display and edit each code snippet.

Opens the AutoDoc window.   AutoDoc is a utility that assists in   documenting routines.   In the VB environment open the edit window of the routine you wish to

document and press this button or press ALT + F12.   You will be prompted for information such as routine parameter definitions, purpose of the routine, and return values (if any).   This information along with your name, date, version, copyright, and routine name will be converted into a standard header that will *automatically* be inserted into the routine you are in.

Exit PTCM.   Pressing Alt + F9 or double clicking on the control button also exits PTCM.

The following buttons are visible while editing or adding code to PTCM:

Paste data on clipboard into the code window.

Attach a form to code snippet.   You will be presented with a dialog box that allows you to select a form.   The program makes a copy of this form and attaches it to the code snippet.   When you add this snippet to your program the selected form will also get added.

Note: A master copy of all attached forms can be found in the PTCM working directory.

Capture selected text and place it in the code window.   This feature allows you to simply highlight the desired text instead of copying it to the clipboard.

 Save all changes or additions to disk.

 Cancels or aborts the present operation.

PTCM Hotkeys

A set of hotkeys are available to access all of the major functions in PTCM.   The scope of these keys is not limited to PTCM.   You may use them in any program at any time. For example, you just finished coding a routine and wish to use the AutoDoc feature to add a header.   You could press the Power Toolz button, then the AutoDoc button to access AutoDoc.   A quicker way is to simply press **ALT** + **F12** which will immediately bring up the AutoDoc window.

The following hotkeys exist for PTCM:

F1   Open the on-line help file for PTCM or for the presently highlighted code snippet or

folder.

**ALT** + **F10**   Exit PTCM.

ALT + **F11**   Open the resource monitor.

ALT + **F12**   Open AutoDoc.

ALT + **F3**   Opens the search window or continues a search already in progress.

Code Tree Hotspots

The graphical symbols in the tree are used to accomplish different tasks.   The plus and minus symbols may be pressed to expand and collapse branches of the tree.   You may simply click on the text to highlight a particular library or snippet.   The symbols next to each line may be dragged and dropped onto icons located in the upper right side of the code window.   This is how code snippets are sent to Visual Basic, the clipboard, or to the trash can.
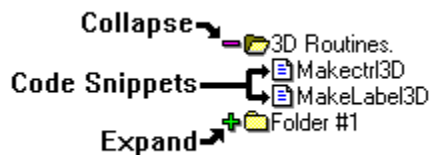
Using the Drag/Drop features

PTCM incorporates a drag/drop interface that allows you to take folders or snippets and drag them to one of three destinations:

Visual Basic:

 Sends the code to an open Visual Basic code window.   The code will be inserted in the open code window at the cursor position.   To use this option first open a code window and position the cursor where you want the text inserted, then open PTCM by pressing the Power Toolz button.   Next drag the code snippet that you want to insert over the VB icon and drop it.   The PTCM window will close and the code will be inserted in your Visual Basic code window.

Windows Clipboard:

 Places code on the clipboard. To use this option first open PTCM by double clicking the Power Toolz button.   Next drag the code snippet that you want to insert over the clipboard icon and drop it.   The PTCM window will close and the code will be on the
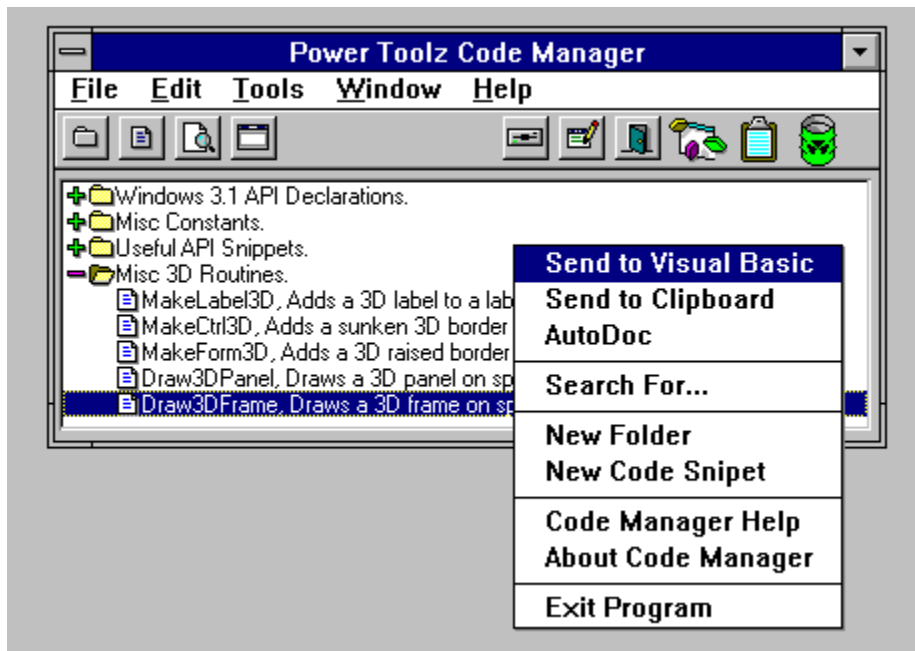
clipboard.

Trash Can:

Delete the code from the tree and stores it in an ascii trash file. To use this option first open PTCM by double clicking the Power Toolz button.   Next drag the code snippet or folder that you want to delete over the trash can icon and drop it.   The snippet or folder is then removed from the code.   A file 'TRASHCAN.DAT' is created in the PTCM working directory which contains any code that you may   delete.   If you double click on the trash can icon you may view this file in Windows Notepad.
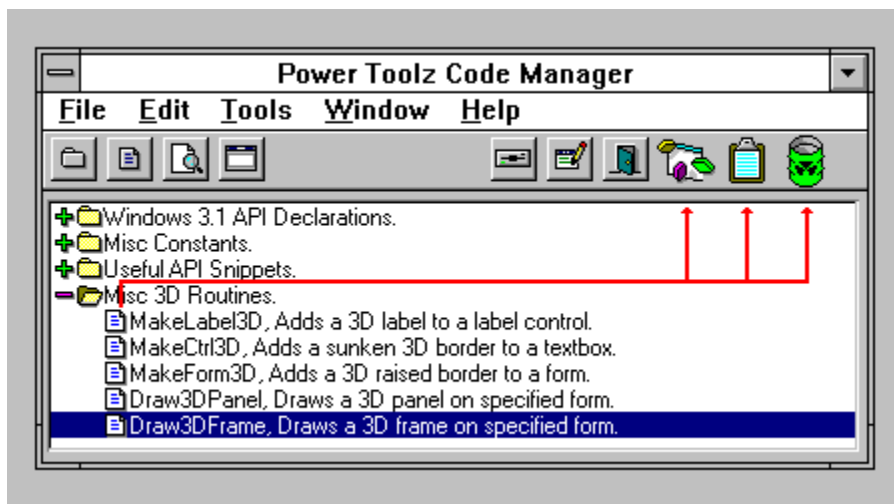
Menus:

PTCM is equipped with context sensitive menu system as well as standard pull down menus.   To use the context sensitive menus simply click the right mouse button any place on the PTCM window.

Power Toolz Code Manager

File Edit Tools Window Help

- Windows 3.1 API Declarations.
- Misc Constants.
- Useful API Snippets.
- Misc 3D Routines.
    - MakeLabel3D, Adds a 3D label to a lab
    - MakeCtrl3D, Adds a sunken 3D border
    - MakeForm3D, Adds a 3D raised border
    - Draw3DPanel, Draws a 3D panel on sp
    - Draw3DFrame, Draws a 3D frame on sp

Menu
Send to Visual Basic
Send to Clipboard
AutoDoc
Search For...
New Folder
New Code Snipet
Code Manager Help
About Code Manager
Exit Program

Installation Information

During installation files are copied to your hard drive.   The following listing describes all the files installed on your hard drive and their functions.

Working/Sample Directory Files:

PTCM.EXE   This is the Power Toolz Code Manager executable file.

README.DOC   This file contains any last minute changes that could not make it into the manual.   This file is in ascii format and can be viewed using the Windows Notepad program.

PTCM.HLP   This is the on-line documentation for PTCM.

TIP.DAT   This file contains 'Tip of the Day information.

Windows Directory:

PTCM.INI This file contains misc information used by the code manager.   This file is created the first time you run PTCM.

Windows/System Directory:

CMDIALOG.VBX   This file is used to create the common dialogs used in PTCM.

MSOUTLIN.VBX   This file is used to create the code tree in PTCM.

SBCKBD.VBX   This file is used when capturing the hotkeys used in PTCM.

DWSPYDLL.DLL   This file is used in conjunction with SBCKBD.VBX.

CTL3D.DLL   This DLL is used to provide the 3D dialogs in PTCM.

**VBRUN300.DLL**   This is the runtime module required by PTCM.

Please consult the README.DOC file to get the dates and sizes for the VBX and DLL files used.   Caution must be exercised when overwriting these files.   Generally newer versions of these files will be compatible, however older versions will cause general protection faults and unpredictable behavior.   PTCM is shipped with the newest VBX and DLL files available.   As new versions of these files become available, we will test their operation with PTCM and produce upgrades as needed.

Program Groups:

After all files are copied, a program group is created in the Program Manager.   An icon is created that will   start PTCM, launch Windows Notepad containing the README.DOC file, and create an icon for this help file.   The evaluation version of PTCM is not equipped with an installation program, so you will need to create a program group and icons manually.

PTCM Directory Structure:

All PTCM executable files are stored in the working directory.   The runtime libraries and custom controls are stored in the WINDOWS\SYSTEM directory.   All PTCM user created libraries are stored in the PTCM working directory.   All 3rd party libraries are stored in sub-directories under the working directory.

Program Files
User Libraries
├─ Add-On Library
├─ Add-On Library
└─ Add-On Library
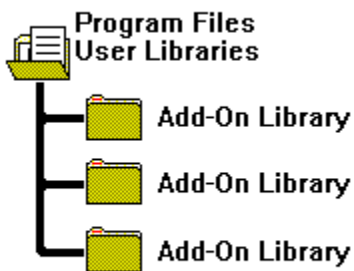
Adding Code Snippets

There are two different ways to add code to Power Toolz Code Manager.   This chapter deals with adding code manually.   See section 6 if you wish to install a code manager library using the automatic installation function.

All snippets must be stored in folders, this will help to logically group routines for later use.   So the first step is to create a folder to place some code snippets into.   Press the 'Add Folder' button on the toolbar.   Then supply a folder name.   Remember once full of libraries it may be more difficult to find this folder so be sure to add some keywords to the description so that you may search for it later using the search function.
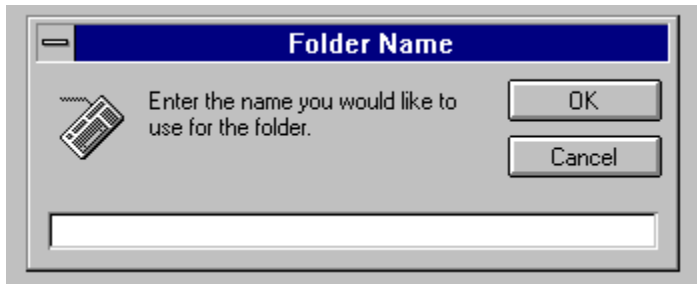
The folder descriptions are not unique (*you may have duplicate folder names*).   Any combination of characters may be used as a folder description.

After you have created a new folder or selected an existing one you can now add a snippet by pressing the 'Add Snippet' button.   At this point the toolbar changes revealing the keys you will need to enter a snippet.   You will also see that you have been provided a place to enter a snippet description and a place to insert the code snippet.

First enter a description by typing in the description box located on the tree.   Remember   to add keywords so you can locate this snippet easily using the search function.   Any combination of characters may be used as a snippet description.
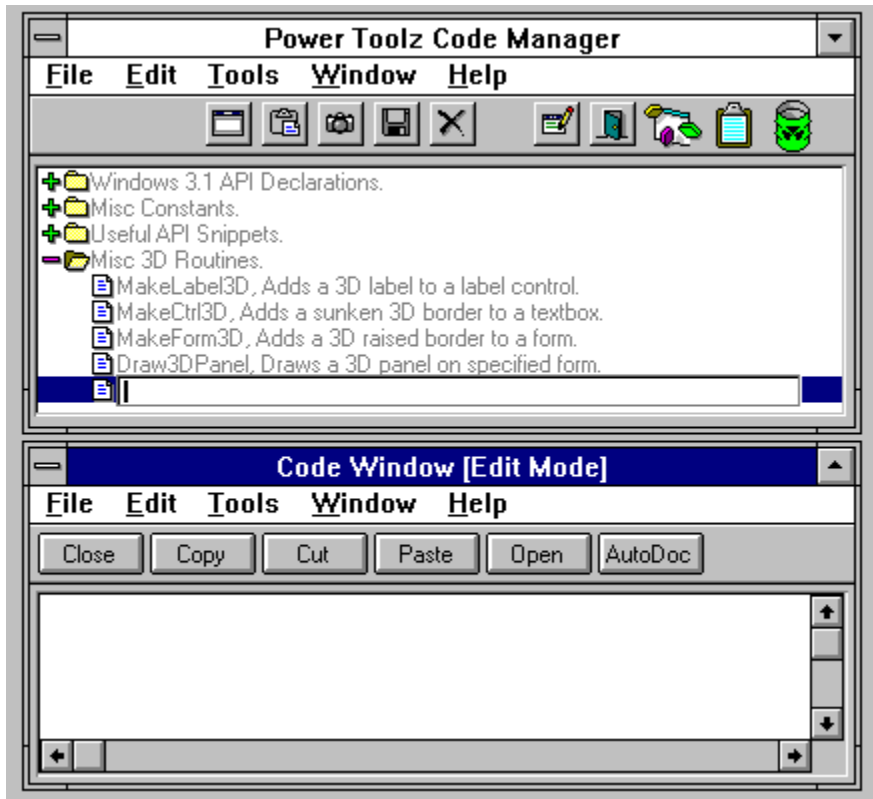
Once the snippet description is entered you then need to place your code into the code window.   You may type code directly in the code window if you wish.   Or if the code is on the clipboard, press the clipboard button and the code will be automatically pasted into the code window.   If you have highlighted some code in another application but have not copied it to the clipboard you can simply press the capture button and the code will be added to the code window.

Attaching a form

You can attach a form to the code snippet by pressing the 'attach form' button.   You will then be presented with a dialog box where you can select the form to attach.   A copy of the selected form is made and stored in the PTCM working directory.

It is also possible to attach a form manually.   First you must copy the form to the PTCM working directory.   Then add ">>Form=" and the form name including the extension.   For example to add a form named INBOX.FRM you would type ">>Form=INBOX.FRM".   The prefix ">>Form=" is case sensitive and needs to be entered exactly as shown.

Any forms that are attached to snippets need to be saved in binary format not text. This eliminates the need to keep track of FRX files.

Once attached to a snippet, the selected form will be automatically added to the Visual

Basic project when you send its code snippet to the project.   If a form exists with the same name in the project you will be prompted with a Visual Basic error message.

Abort!!

If you wish to abort the snippet addition simply click the 'Cancel' button.   Any descriptions, code, or forms that you started will be removed when you press the 'Cancel' button.

Save

When you are finished, press the save button and the snippet will be saved for later use.

Editing and Removing Code Snippets

Editing a Folder Description

To edit an existing folder description, you need to double click on the folders description in the code tree.   The editing box will appear containing the folders old description. You may change this description any way you choose.

Editing a Snippet Description

To edit an existing snippets description, you need to double click on the snippets description in the code tree.   The editing box will appear containing the snippets old description.   You may change this description any way you choose.
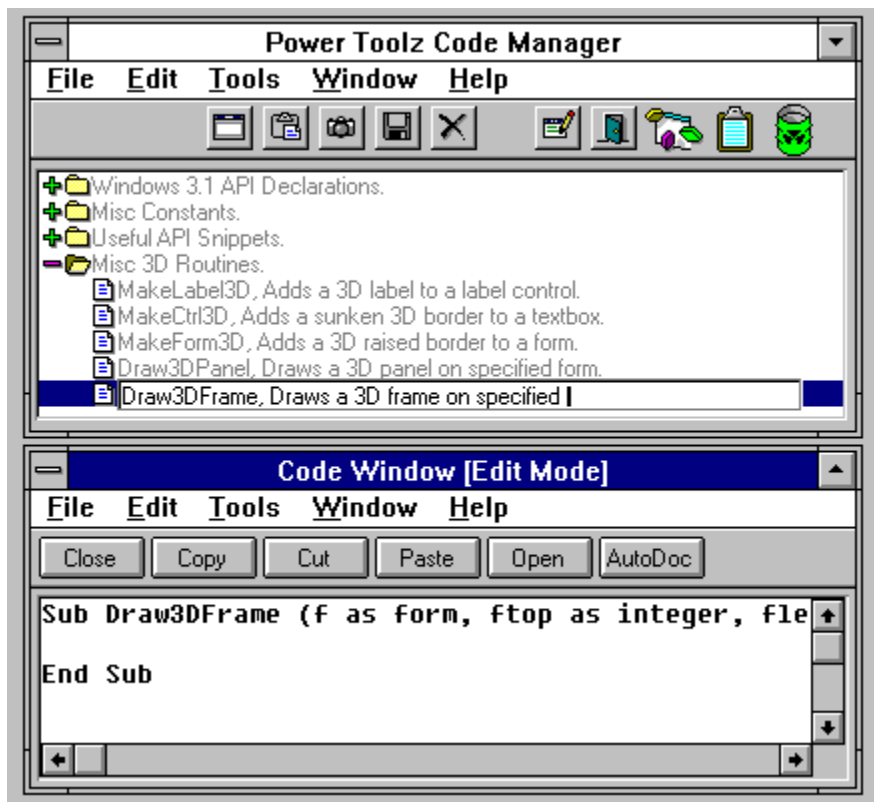
Editing a Snippets Code

To edit an existing snippets code, you need to double click on the snippets description in the code tree.   The editing box will appear containing the snippets old description. Additionally the code window will be unlocked allowing you to change the code.   You may change the code any way that you choose.

Abort!!

If you wish to abort the snippet changes, simply click the 'Cancel' button.   Any descriptions, or code that you changed will be returned to the condition before you started the changes.

Saving Changes

When you are finished, press the save button and the snippet will be saved for later use.

Deleting Snippets

Individual snippets can be deleted from folders by dragging them to the trash can.   First open the folder containing the snippet to be deleted.   Then drag   the snippet from the code tree to the trash can icon in the upper right of the PTCM window.   The snippet will be then be removed from the code tree.

The deleted snippet is stored in an ascii file called TRASHCAN.DAT in the working directory.   You can view this file by double clicking on the trash can icon.

Deleting Folders

Folders can be deleted by dragging them to the trash can.   First select the to be deleted.   Then drag   the folder from the code tree to the trash can icon in the upper right of the PTCM window.   The folder and all its snippets will be then be removed from the code tree.
The deleted folder and snippets are stored in an ascii file called TRASHCAN.DAT in the working directory.   You can view this file by double clicking on the trash can icon.
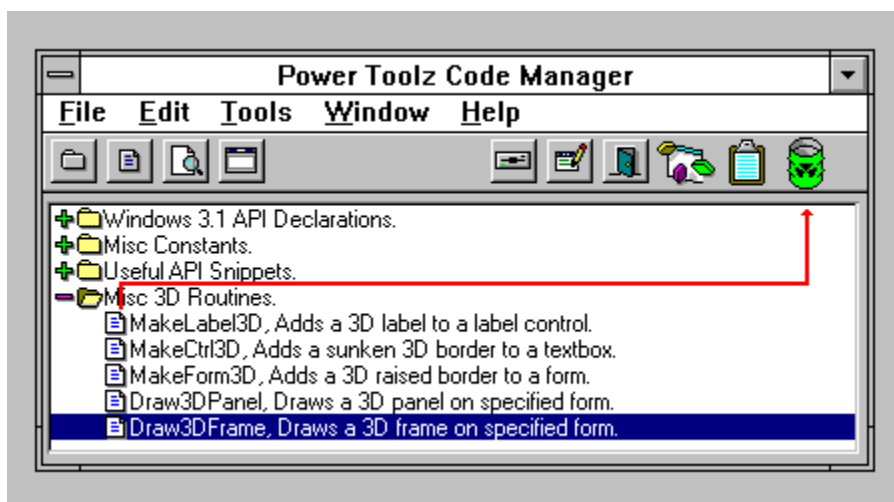
Using PTCM with Visual Basic

Using PTCM with Visual Basic is the easiest part of learning the program.   In fact the entire user interface has been designed around this one specific task of adding code to the Visual Basic development environment.

Sending Code to Visual Basic:

1.   Go to any code window and place the cursor where you want your snippet added.
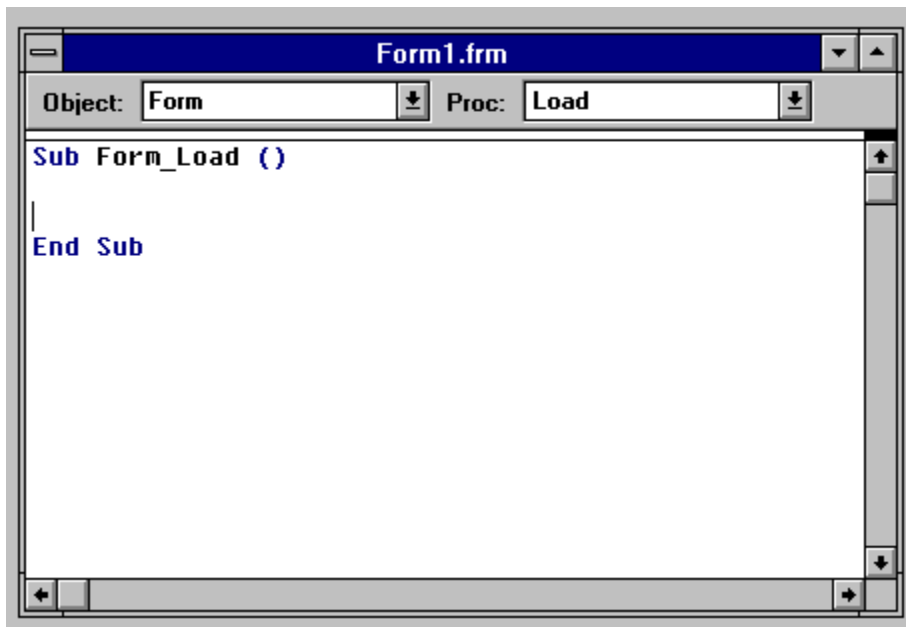
2.   Open PTCM and drag the desired snippet to the VB icon in the upper right side of the PTCM window.
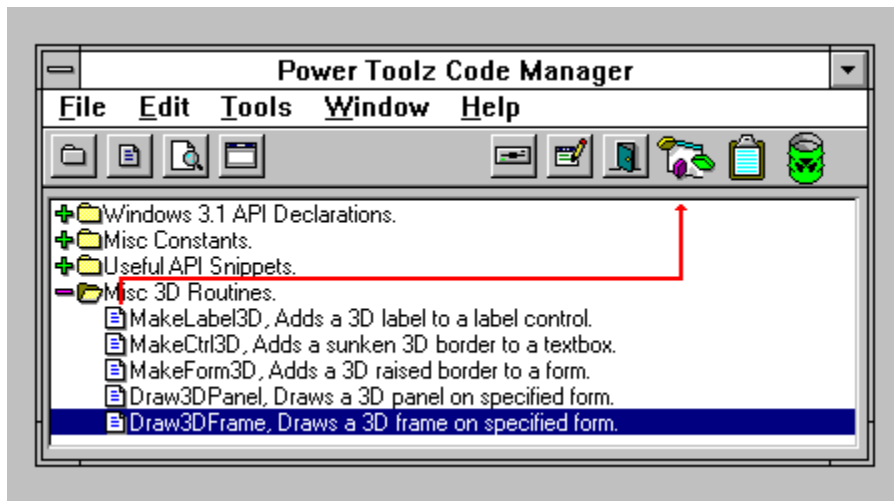
3. That's it!   The snippet is now part of your VB project.

Note: When adding entire subroutines to VB the cursor should be placed between Sub and End Sub statements to be recognized as a subroutine. Subs will also be recognized by VB when inserted in the declarations section of the code window

Sending Code to the Clipboard:

1.   Open PTCM and drag the desired snippet to the clipboard icon in the upper right side of the PTCM window.
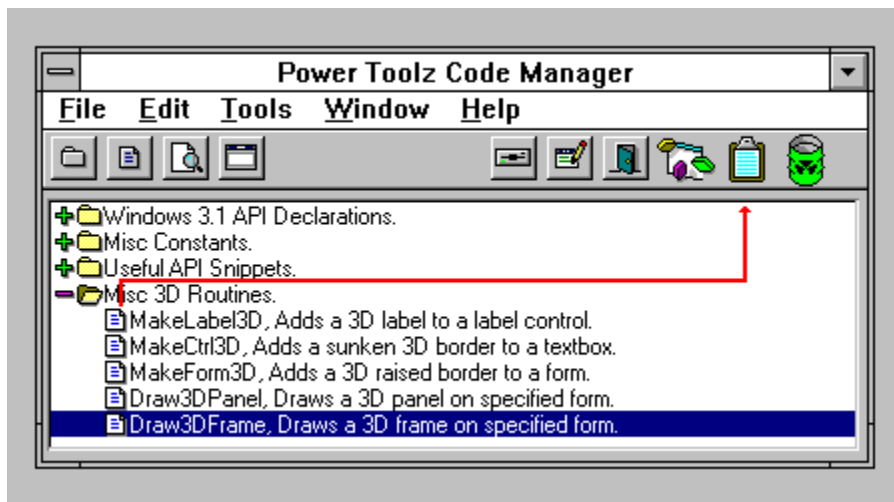
2.   That's it!   The snippet is now on the clipboard.   You may insert it into Visual Basic by using the Paste function

Note: When adding entire subroutines to VB the cursor should be placed between Sub and End Sub statements to be recognized as a subroutine. Subs will also be recognizes when inserted in the declarations section of the code window.

If you have a form attached to the snippet you must send the snippet to VB by dropping it on the VB icon instead of the clipboard icon.   If you do use the clipboard icon only the text that appears in the code window is transferred to the clipboard.

Library Help:

If you need to know more about the routine you wish to send to VB, highlight the routine and press **F1**.   If the library has an associated help file, it will be shown to you.

Installing Add On Libraries

Power Toolz Code Manager supports code libraries written by other authors.   Built in to PTCM is a library installation utility that makes adding 3rd party libraries a snap.

To open the installation utility, press the 'Install Library' button on the toolbar or press **ALT** + **F10**.

When the installation utility starts it automatically scans the floppy drives for compatible libraries to install.   Any available libraries that may be installed are shown.
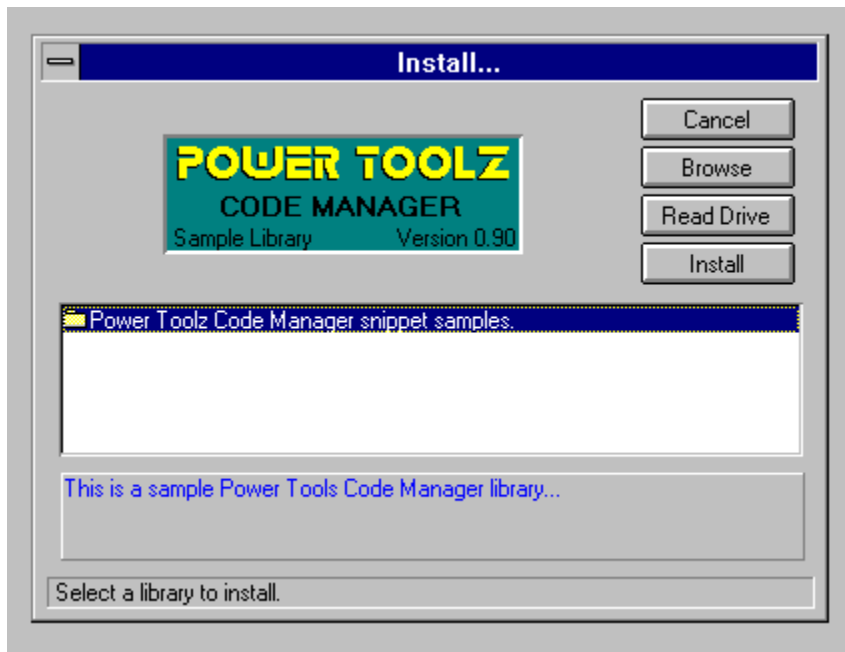
Viewing Library Descriptions:

To view descriptions for each of the different libraries (if more than one library exists), simply select the desired library with the mouse or cursor keys.   As you select a new library the picture and description boxes will change.

OK:

When you are finished installing libraries, or just want to leave the installation screen just press this button.

Browse:

To install libraries already on your hard drive press this button.   A dialog box will appear

that allows you to select any PTCM library files (CHKLST.CMX).   Select a library file and press 'OK'.   The selected library will automatically be installed.   Press 'Cancel' to abort the operation.

PTCM libraries are made up of many different files.   The file CHKLST.CMX contains a list of all files required to successfully install the library.   This is why the Browse dialog box looks for this file.

Read Drive:

If you forgot to insert the floppy disk when you opened the installation utility or if you changed disks click this button to relog the floppy drives.

Install:

Once you have selected the library to install from the list box press this button.   You will be presented with a dialog box asking for the name of a sub directory to install the library to.   A full path is not required.   Files will now be copied to the specified sub directory under the PTCM working directory.
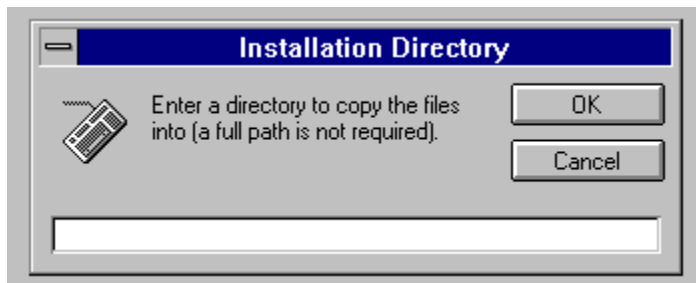
Using AutoDoc

The AutoDoc function of Power Toolz Code Manager allows you to easily and quickly add a comment header to any VB routine.

To access AutoDoc first open the code window containing the subroutine or function to document.   Next press the AutoDoc button or type **ALT** + **F12**.   The AutoDoc screen will scan the routine in the open code window and extract the name and all parameters. These will appear in the AutoDoc window.

Field Descriptions:

Routine Name: This field is extracted from the open Visual Basic code window.

Revision: This field can be user defined and consist of just about anything you want. If you make changes to this field they are saved automatically so you can use them again.

Author: This field can be user defined and consist of just about anything you want.   If you make changes to this field they are saved automatically so you can use them again.

Copyright Notice: This field can be user defined and consist of just about anything you want.   If you make changes to this field they are saved automatically so you can use them again.

Routine Description: This field allows you to enter a short description about the routine.

Return value: This field is only visible when you are documenting a function.   Use this field to define the return value of the function.

Parameters: This listbox contains the names of all passed parameters.   You can define each parameter by typing a short description in the textbox pointed to by the arrow.   When you are done with the first parameter you can continue selecting a new parameter until you have descriptions for each one.

Button Descriptions:

Cancel: This button closes the AutoDoc screen and you loose all data entered into the data entry fields.

Send to VB: When you are done entering descriptions for each of the fields, press this button to send a header to the open code window.   The following Is a sample routine documented by AutoDoc:

```
'+----------------------------------------------|PT AutoDoc|-------+
'|                                                                 |
'|Name: makelabel3d              Revision: 1.00    Date: 01-04-1995|
'|                                                                 |
'|Author: Author Name                                              |
'|                                                                 |
'|Operation: Adds a 3D border to any label control.                |
'|                                                                 |
'|Parameters Passed:                                               |
'|     f As Form: Name of the form containing the label control.   |
'|     c As Control: Name of the control to add a 3D border to.    |
'|     bevelwidth As Integer: Width of the 3D borders bevel in pixels.|
'|     pstyle As Integer: Style of the bevel 1= raised, 0 = sunken. |
'|                                                                 |
'|                                                                 |
'|           Copyright (C) 1994 All Rights Reserved                |
'+----------------------------------------------------------------+
```
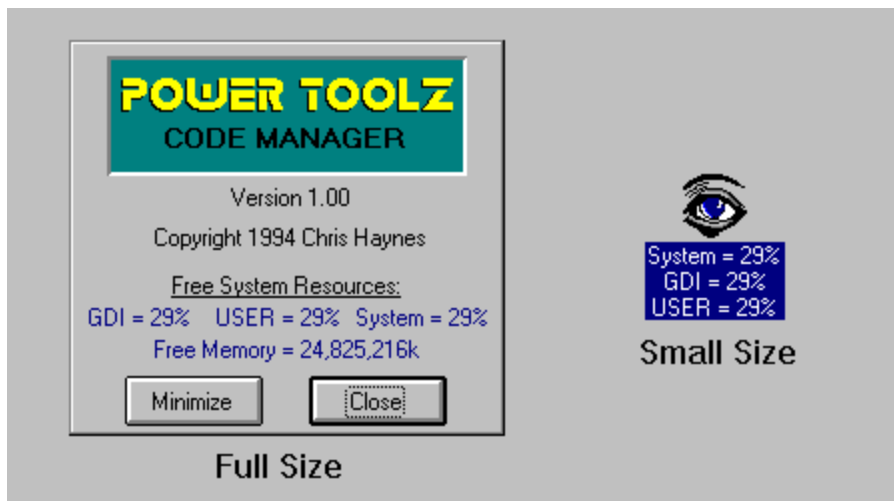
Using the Resource Monitor

Power Toolz Code Manager is equipped with a resource monitor that you can use to monitor the amount of free resources available to your programs.   To access the resource monitor you can use the pull down menus or press **ALT** + **F11**.

Resource Definitions:

There are three type of resources monitored by PTCM:

1.   *System* resources for general Windows use.

2.   Graphical device interface (*GDI*) resources used to store device context handles, graphic regions, fonts, and bitmaps.

3.   *User* resources that Windows uses to store form and menu handles.

Free Memory: The free memory indicator returns the amount of memory in the global heap that is not used and that is not reserved for code.   In 386-enhanced mode, this value is an estimate of the amount of memory available to an application. It does not account for memory held in reserve for non-Windows applications.

Version Information:

The resource monitor screen also shows you the revision number of Power Toolz Code Manager and copyright information.   You will need the information on this screen when you request technical support.

License Agreement

Power Toolz Code Manager is licensed, not sold to you by Chris Haynes of Ashby Massachusetts for use under terms of the license agreement.   You own the media on which Power Toolz Code Manager sold to you on, however the author retains ownership of the software itself.

1.   Under this license agreement you are authorized to use one copy of this software at a time.   to "use" the software means that the software is either loaded into the temporary memory (RAM) or installed on the permanent memory of a computer (Hard disk, etc).   Additionally you are authorized to make one copy of the software solely for backup purposes.   You are also authorized to place a copy of this program on a home computer that is owned by the primary user.

2.   You may not alter, reverse engineer, modify, distribute, lease, or create derivative works based upon Power Toolz or any part of Power Toolz.

3.   In no event will the author be liable to you for any consequential, incidental, indirect or special damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of business information and the like), regardless of the basis of the claim and even if the author has been advised of the possibility of such damage.   The authors liability to you for direct damages for any cause whatsoever, and regardless of the form of the action, will be limited to the money paid for the software that caused the damage.

4.   This license is effective until terminated.   This license will terminate immediately without notice from the author if you fail to comply with any provision of this license. Upon termination you must destroy all software and written materials that came with the software, and section 3 will survive any license termination.