Why Thor?

Jim Nelson
The KONG Company
21 Los Vientos Drive
Newbury Park, CA 91320
Voice: (805) 498-9195
Fax: (805) 498-9195
JimRNelson@Gmail.com

Why Thor? Because

- *the IDE has been stagnant since version 9 was released about a decade ago, when its features were already lagging behind what was available in other environments; the IDE has the feel of a time long ago.*

- *the intricacies of SCXs and VCXs and the inheritance model make for a complex environment in which we have become accustomed to making do with the tools provided, as if that is all we could hope for.*

Introduction

Thor offers a wide number of tools to enhance the IDE and thus your productivity. Some of the concepts implemented in these tools came from other environments, most notably Visual Studio, while others come from brainstorming about the peculiar needs for handling SCXs and VCXs.

VFP's IDE provides us with manual tools; Thor offers power tools instead.

This paper begins with a broad overview of Thor and its features, and then describes some of the most useful tools that Thor makes available.

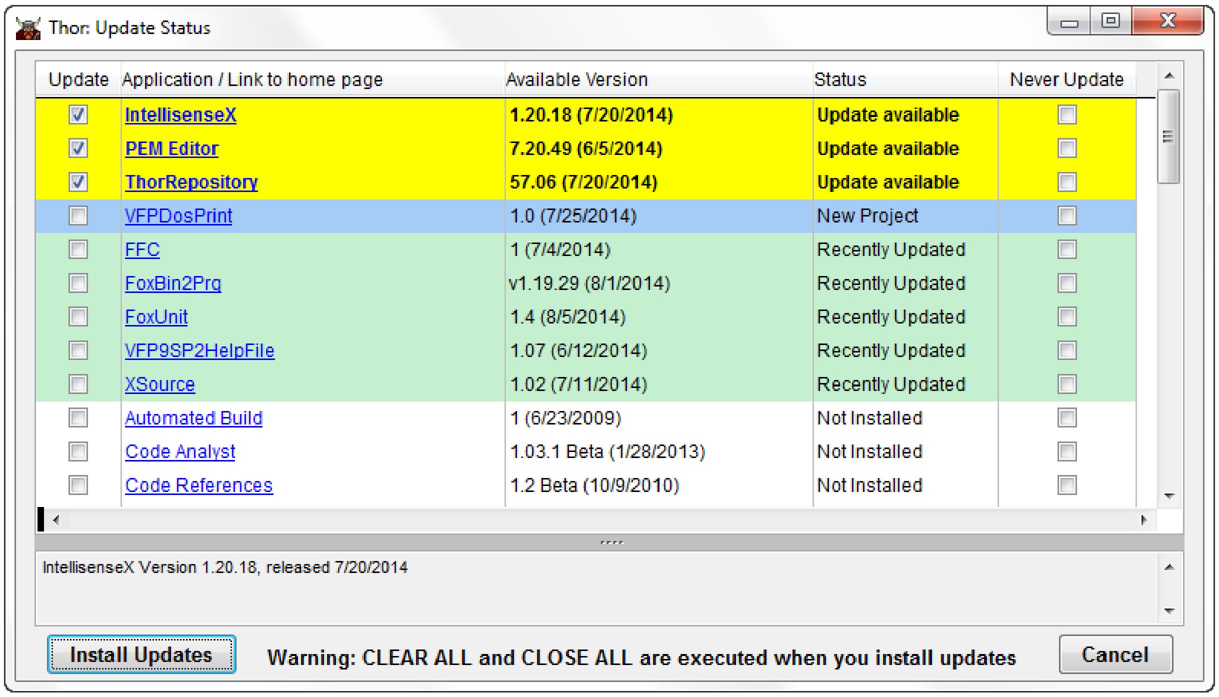
The Parts of Thor

There are three distinct parts to Thor:

- A form for downloading current releases of all VFPX Projects
- A suite of over 150 Thor tools
- UI forms that allow you to control how you access these Thor tools

Downloading Current Releases of all VFPX Projects

Thor's "Check For Updates" form, shown in F**igure 1**, available from the Thor menu pad, shows the list of all VFPX projects and their current version numbers and has links to their home pages.



Update	Application / Link to home page	Available Version	Status	Never Update
✓	IntellisenseX	1.20.18 (7/20/2014)	Update available	
✓	PEM Editor	7.20.49 (6/5/2014)	Update available	
✓	ThorRepository	57.06 (7/20/2014)	Update available	
	VFPDosPrint	1.0 (7/25/2014)	New Project	
	FFC	1 (7/4/2014)	Recently Updated	
	FoxBin2Prg	v1.19.29 (8/1/2014)	Recently Updated	
	FoxUnit	1.4 (8/5/2014)	Recently Updated	
	VFP9SP2HelpFile	1.07 (6/12/2014)	Recently Updated	
	XSource	1.02 (7/11/2014)	Recently Updated	
	Automated Build	1 (6/23/2009)	Not Installed	
	Code Analyst	1.03.1 Beta (1/28/2013)	Not Installed	
	Code References	1.2 Beta (10/9/2010)	Not Installed	

IntellisenseX Version 1.20.18, released 7/20/2014

Warning: CLEAR ALL and CLOSE ALL are executed when you install updates

Figure 1. "Check For Updates" displays the list of all VFPX projects, with links to their home pages, and downloads and installs updates for them.

Use this form to download and install any VFPX projects that interest you, allowing you to bypass the inconsistencies of the VFPX home page.

The projects downloaded from VFPX are divided into two categories, depending on how they are used.

Tools

"Tools" are those projects that you use in development, such as GoFish, PEM Editor, IntellisenseX, and many others. When downloaded, "Tools" become instantly accessible as "Thor tools" that you can execute by hot keys, the system menu, the Thor Tool Bar, or any of the other ways that Thor provides. See the video 50 ways to run a Thor tool (14:39).

Components

"Components" are those projects that you embed in your applications, such as FoxCharts, FoxBarCode, and DesktopAlerts. Components are downloaded into a sub-folder of

```
_screen.cThorFolder + 'Tools\Components'
```

It is then up to you to copy the relevant files from this sub-folder so that they can be accessed within your applications.

Suite of Thor Tools

The core of Thor is the suite of over 150 Thor Tools, contributed over the years by many members of the Thor community. The tools fall into a number of broad categories:

- Searching and "Go To" tools
- Objects and PEMs
- Parent and Containing Classes
- Manipulating text in code windows
- Tables

The Thor UI: Controlling access to Thor Tools

Thor provides UI forms so that you can access all of the Thor Tools in the way that you are accustomed to accessing your personal tools. You can make the tools accessible by

- Assigning hot keys (including combinations using two of Shift, Ctrl, and Alt)
- Creating menu pads
- Adding them to the Thor ToolBar
- Using pop-up menus (unique to Thor)
- Running from the Search page in Tool Launcher
- Running from the Favorites page in the Tool Launcher

- Running automatically when you start Thor
- Using the “Thor Tools” menu pad (part of the original design, but quite cumbersome now with so many nodes)

You may find it helpful to watch the video 50 ways to run a Thor tool (14:39).

The two Thor forms, Thor Configuration and Tool Launcher, are accessed from the Thor tool pad, as shown in **Figure 2**:

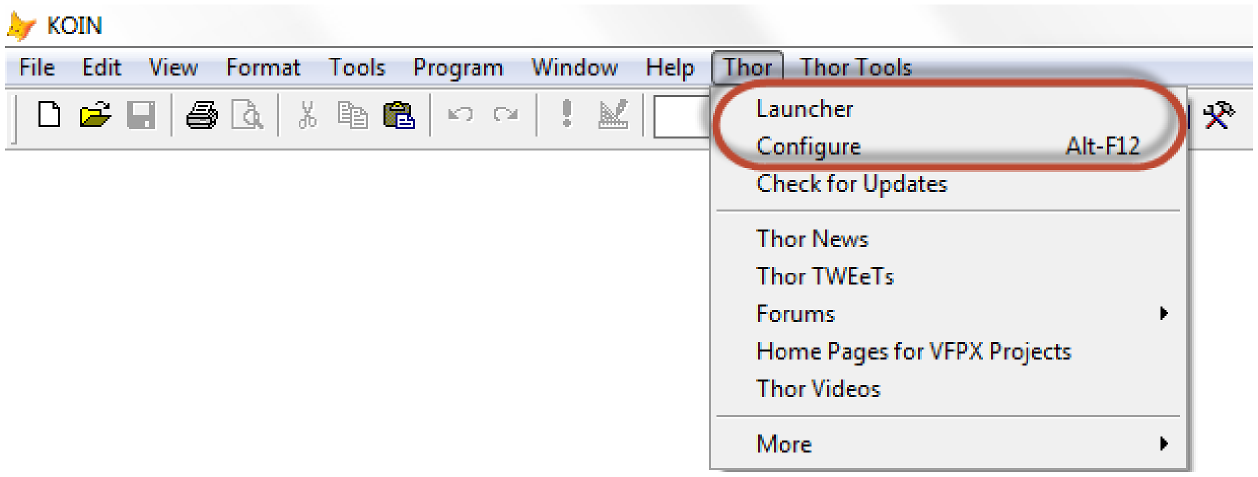


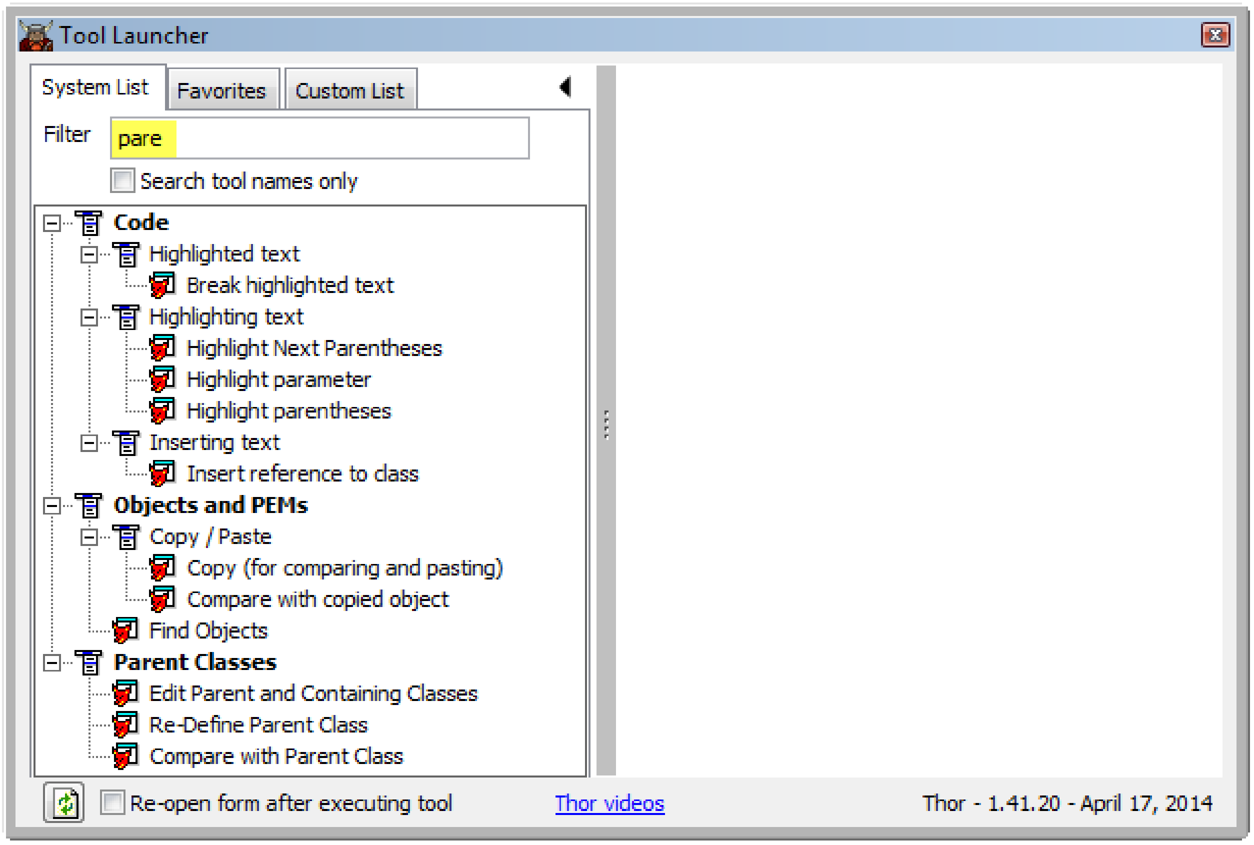
Figure 2. The Configuration and Launcher forms are accessed from the Thor menu pad.

The Configuration form provides a number of features:

- Creating menu pads and adding tools or sub-menus to them
- Creating pop-up menus
- Assigning hot keys to tools
- Browsing the list of all hot key assignments, including On Key Label and macros
- Determining which Thor tools run automatically when Thor starts
- Setting options (applicable to a number of Thor tools)

The Launcher form has some overlap with the configuration form, but its intent is to simplify the task of finding and executing Thor tools. The first page has a search text box that searches the names and descriptions for all Thor tools, as shown in **Figure 3**. Matches to the entered text of “pare” include both “parent” and “parenthesis”. Each word entered is searched for separately.

The Launcher form is the best place for both new and experienced Thor users to find and then execute Thor tools. Over time, as you identify those tools that you find useful, you can use the Configuration and Launcher forms to make them accessible via hot keys, from a custom menu pad in the system menu, from the Thor tool bar, and so on.


Tool Launcher

System List | Favorites | Custom List

Filter: pare

Search tool names only

- Code
  - Highlighted text
    - Break highlighted text
  - Highlighting text
    - Highlight Next Parentheses
    - Highlight parameter
    - Highlight parentheses
  - Inserting text
    - Insert reference to class
- Objects and PEMs
  - Copy / Paste
    - Copy (for comparing and pasting)
    - Compare with copied object
  - Find Objects
- Parent Classes
  - Edit Parent and Containing Classes
  - Re-Define Parent Class
  - Compare with Parent Class

Re-open form after executing tool   Thor videos   Thor - 1.41.20 - April 17, 2014

Figure 3. Entering "pare" in the Tool Launcher form shows tools where "pare" is found in either the name or description.

Pop-up menus

Thor has a unique feature called "Pop-up Menus", where a menu is activated by using a Hot Key, much as right-clicking pops up a context menu. **Figure 4** is an example of a pop-up menu.

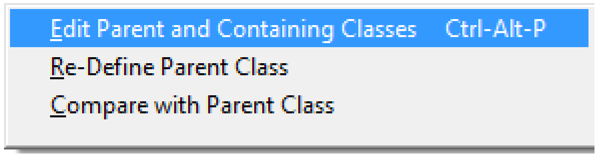

Edit Parent and Containing Classes   Ctrl-Alt-P
Re-Define Parent Class
Compare with Parent Class

Figure 4. A sample Pop-up menu, activated by hot key, with a number of related Thor tools as choices

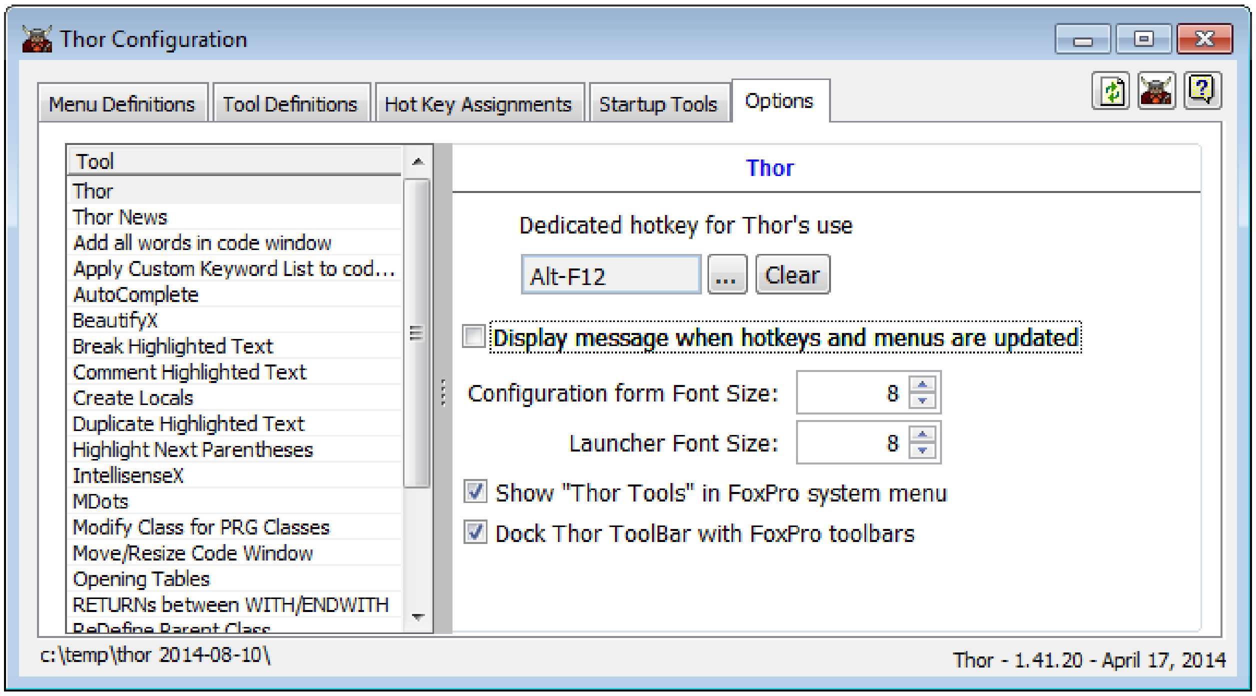
Pop-up menus are created on the first page of the Thor Configuration form. Don't forget to assign a hot key to any Pop-Up Menu you create.

Pop-up Menus can act like Visual Studio's key chords if you use "\<" in the prompts for the individual items; the first key of the chord is the hot key to show the pop-up menu and the second key selects the desired item from the pop-up menu.

You can create a starter set of Pop-up Menus, including the menus used in this document and a number of others, by executing the Thor Tool *Create Sample Menus*. These pop-up menus will not have any hot keys assigned to them, so you will need to do that in order to make them usable.

Configuring Thor

The last page ("Options") of the Configuration form controls settings for Thor and many of its tools. **Figure 5** shows common settings used by experienced Thor users.



Dedicated hotkey for Thor's use: Alt-F12

☐ Display message when hotkeys and menus are updated

Configuration form Font Size: 8

Launcher Font Size: 8

☑ Show "Thor Tools" in FoxPro system menu

☑ Dock Thor ToolBar with FoxPro toolbars

c:\temp\thor 2014-08-10\

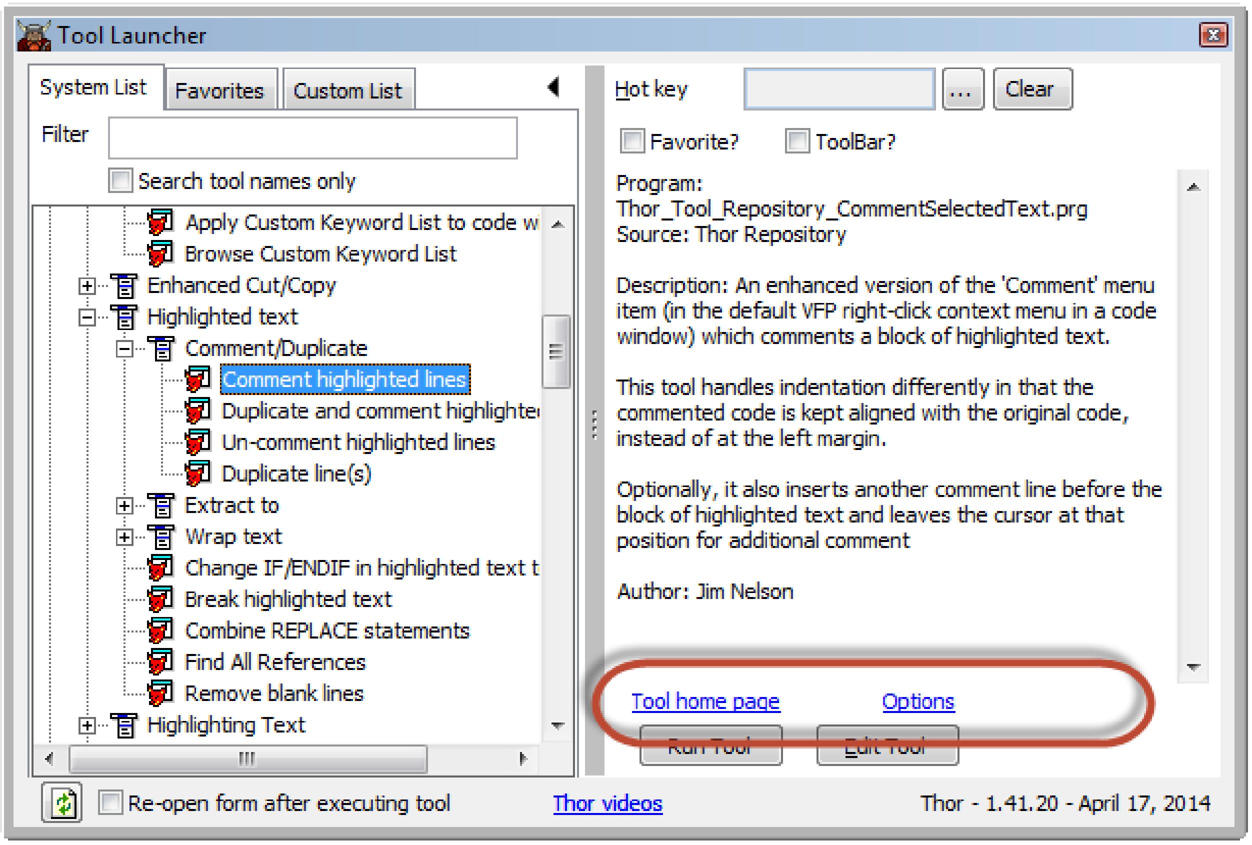
Thor - 1.41.20 - April 17, 2014

Figure 5. The Options page in the Configuration form controls settings for Thor and many of its tools

Tool-specific options

Some tools have more information available than is displayed in the right panel when selected from the left panel in the Launcher or Configuration forms. When available, one or more of the following links may appear at the bottom of the right panel, as shown in **Figure 6**:

- Tool Home Page
- Video
- Options (opens the Configuration form and navigates to the option for the tool on the last page.)
- Plug-ins (see Managing plug-ins, a video that demonstrates the use of two of the available plug-ins)



Program:
Thor_Tool_Repository_CommentSelectedText.prg
Source: Thor Repository

Description: An enhanced version of the 'Comment' menu item (in the default VFP right-click context menu in a code window) which comments a block of highlighted text.

This tool handles indentation differently in that the commented code is kept aligned with the original code, instead of at the left margin.

Optionally, it also inserts another comment line before the block of highlighted text and leaves the cursor at that position for additional comment

Author: Jim Nelson

Figure 6. Some tools have links in the right panel if the tool has a home page, video, or associated options or plug-ins.

Manage your own tools using the Thor UI

You can make the tools you have created over the years accessible thru the Thor UI as well. This will make them easy to find (by using the Tool Launcher) and you can run them any of the ways that Thor provides. See Creating New Thor Tools.

We suggest that you set the Category and Author properties for these tools to your name or initials so that they group together in the Tool Launcher display.

Installing Thor

To install Thor, follow these steps:

- Download Thor.Zip from http://vfpxrepository.com/dl/thorupdate/thor/Thor.zip
- Unzip it in a permanent folder (it will create a “Thor” sub-folder for all of its files)
- Double-click Thor.App
- Install all updates (as well as any other projects you are interested in) from the “Thor Update Status” screen that appears. We strongly suggest also downloading and using GoFish.

Running Thor

The installation procedure creates RunThor.PRG in the Thor folder. To run Thor, call this PRG as part of the startup of your VFP session. Since this PRG contains an explicit reference to the folder where Thor is installed, in may be moved or copied elsewhere for your convenience.

The recommended syntax is

```
    Do RunThor with 7
```

which will cause "Check For Updates" to be run once a week.

See also Running Thor.

Survey of Thor Tools

The remainder of this document provides a survey of Thor tools. This is not an exhaustive list, as there are far too many tools, nor an exhaustive description of the tools that do appear. Instead, it provides a broad overview of the types of tools that are available, with enough information that you can begin using them right away.

The tools are described in the following categories:

- Tools for Searching
- Tools for Objects and PEMs
- Tools for Parent Classes and Containing Classes
- Tools for Working with Highlighted Text
- Standardizing your code
- Miscellaneous Tools

When in doubt, or if you have questions, take advantage of the Thor Forum, a link that is also available from the Thor menu pad.

Tools for Searching

Thor provides a number of tools for searching. The first of those described here, *GoFish*, (itself a separate VFPX project), is easily recognized as a replacement for Code References. The others provide new capabilities, specific to the VFP environment.

GoFish

GoFish, a VFPX project that can be used independent of Thor, is the much applauded successor to Code References. It is blindingly fast, as much as ten times faster than Code References, and provides a wealth of new options in its UI. In addition, to ensure that it does not have the same problem as Code References, where making replacements can occasionally trash a VCX or SCX, it is slavish about making backups of files before replacements are made and recording the exact changes.

Author's note: I have been the project manager of Code References in VFPX for a number of years and I cannot remember the last time I used it; I use GoFish daily.

GoFish is amply documented on the GoFish Home Page and its UI is similar enough to Code References that most users will feel comfortable without any further instruction. Nonetheless, there are some powerful features that might be overlooked.

Wild card searching

GoFish provides for wild card searching using * and ?. This option is accessed near the bottom of the "Advanced Search" screen and is persistent like all GoFish options from one session to the next.

There is no time penalty for enabling wild card searching, as GoFish always uses regular expressions to do its searching, something that is completely transparent to you.

Author's note: I couldn't live without this feature.

Active Project / Current Directory

There are two different places where you can select the scope of your search, from the "Advanced Search" screen or the combobox for scope on the main screen. In both of these places, the options "Active Project" and "Current Dir" are present.

Selecting either of these two persists to the next session of GoFish. For example, if you have chosen "Active Project", not only does GoFish immediately set the scope to your active project, but when you next return to GoFish it will change the scope to whatever your active project is at that time.

Column Selection and Ordering

GoFish makes available an extensive list of columns to display in the grid; you can select which you wish to see on the "Options" screen. Note in particular the 'MatchType' column, which identifies the type of match for each row.

You can also re-order the columns and change their widths.

All of these changes persist to the next session.

Sorting

You can sort by a column in the grid, making it the primary sort, by clicking on the column header.

You can cause a column in the grid to be the secondary sort by right-clicking on the column header.

Author's note: For my own use, I have moved the MatchType column to the far left and made it the primary sort by clicking on the column header. Whenever I want to sort on another column, I always right-click on its column header so that the primary sort is still by MatchType.

Filtering

After the sort is done, you can filter the grid by any column or combination of columns by clicking the "Filter" command button just above the grid.

Replace Mode

GoFish offers three distinct Replace Modes, as shown in **Figure 7**:

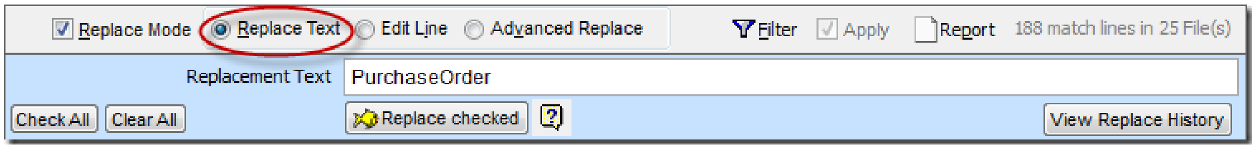


Figure 7. GoFish offers three Replace Modes, of which Replace Text is the most familiar.

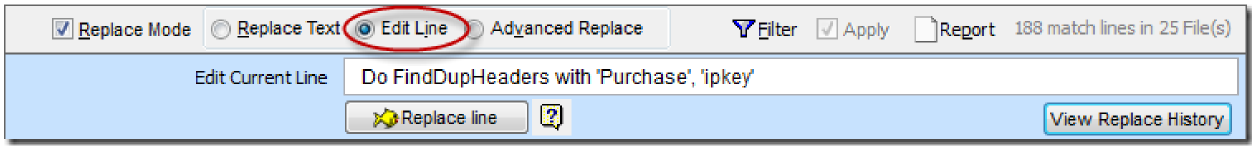
- Replace Text – the familiar mode you are accustomed to with Code References or Ctrl+F. (Be careful; don't use this mode if there are wild cards in your search!)
- Edit Line – allows you to edit the matched line directly without actually opening the file for editing, as shown in **Figure 8**. (Be careful here as well; don't use this on a file that is already open for editing)
- 

Figure 8. Edit Line allows you to edit and replace a line with actually opening the file.

- Advanced Replace – a powerful feature, but of limited applicability (see GoFish Replace Help)

Replacement History

GoFish makes a backup of all files before it attempts to make replacements in them. These backups are saved in folder Home(7) + "\GoFishBackups". A link to this folder can be found in the "Options" screen on the "Backups" page.

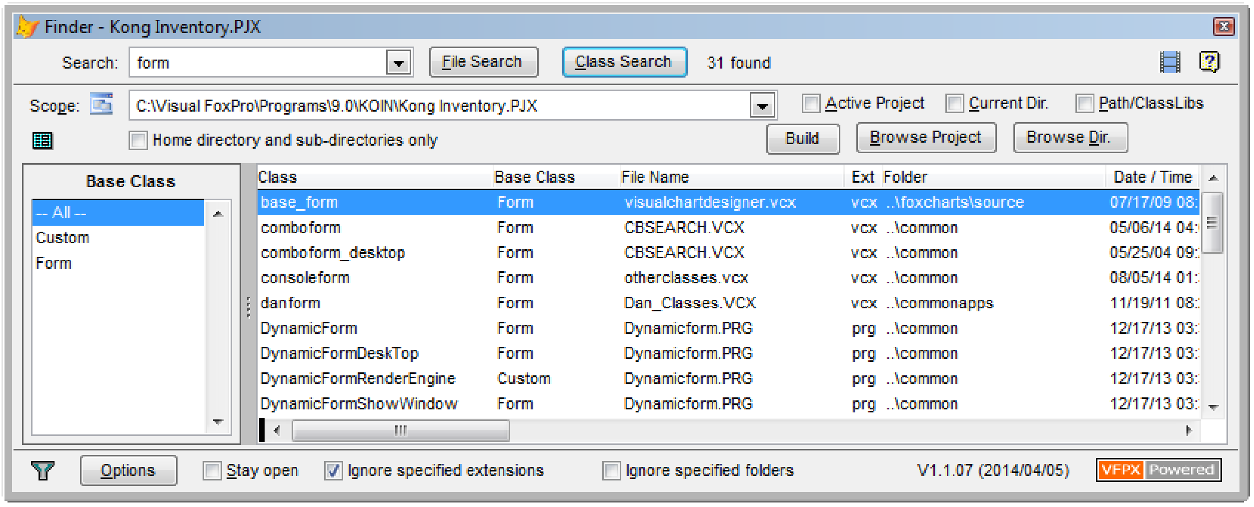
GoFish also maintains a history of all replacements that it has made. Use the "View Replace History" button at the bottom right in the Figure above.

Finder

Finder is a close relative of GoFish; see **Figure 9**. It differs from GoFish in that it is used specifically to look for the names (or parts of names) of classes or files, rather than looking for references to text in code, as GoFish does. It finds both VCX classes and PRG classes.

Finder is particularly suited for finding classes. It provides a large number of features for working with classes, including a number that are unique to Finder.

When working with files, Finder provides many features similar to the Project Manager, including the ability to filter by base class or folder location.



Search: form

31 found

Scope: C:\Visual FoxPro\Programs\9.0\KOIN\Kong Inventory.PJX

[ ] Active Project [ ] Current Dir. [ ] Path/ClassLibs

[ ] Home directory and sub-directories only

Base Class: -- All --, Custom, Form

Class	Base Class	File Name	Ext	Folder	Date / Time
base_form	Form	visualchartdesigner.vcx	vcx	..\foxcharts\source	07/17/09 08:
comboform	Form	CBSEARCH.VCX	vcx	..\common	05/06/14 04:
comboform_desktop	Form	CBSEARCH.VCX	vcx	..\common	05/25/04 09:
consoleform	Form	otherclasses.vcx	vcx	..\common	08/05/14 01:
danform	Form	Dan_Classes.VCX	vcx	..\commonapps	11/19/11 08:
DynamicForm	Form	Dynamicform.PRG	prg	..\common	12/17/13 03:
DynamicFormDeskTop	Form	Dynamicform.PRG	prg	..\common	12/17/13 03:
DynamicFormRenderEngine	Custom	Dynamicform.PRG	prg	..\common	12/17/13 03:
DynamicFormShowWindow	Form	Dynamicform.PRG	prg	..\common	12/17/13 03:

[ ] Stay open [✓] Ignore specified extensions [ ] Ignore specified folders

Figure 9. Finder looks like a close relative of GoFish, but search for names of classes or files.

Finder is amply documented on the Finder Home Page and there is also an introductory video. As with GoFish, you can open a class or form for editing by double-clicking on a row. There is much more to Finder than this, though; following is a list of the most notable features.

Timestamps for SCXs, VCXs, and Classes

The timestamps displayed for SCXs and VCXs do not show the timestamp of the last time the file was changed, which may have occurred when building a project, but instead the most recent timestamp from the rows in the SCX/VCX. More importantly, the timestamps for classes are based on the most recent timestamp from the rows for the class and are an accurate indication of the time of the last change to the class.

Filter Panel

There are a number of different filters that can be displayed in the panel on the left; by base class, class hierarchy, folder structure, or file type. The right-click context menu provides the list of available filters.

Dragging/Dropping a Class

Just as in the Class Browser, after selecting a class in the grid, you can drag the icon for it that appears under the "Scope" label and drop it onto a form or class that you are editing.

Alternatively, you can drop the class onto PEM Editor. In this case, the object is added as a child or sibling of the object that is current displayed in PEM Editor. This makes it very easy to drop classes into hard to get to places, such as columns in a grid or containers that are obscured in the form/class being edited. Simply navigate to the container object, using PEM Editor, Property Window, or the mouse, and then drop the class onto PEM Editor.

Context menu on rows in the grid

The right-click context menu for rows in the grid is sensitive to the extension of the class or file selected, and may include any of these options:

- Open (same as Dbl-Click)
- Run (for PRGs, SCXs, and FRXs)
- Open with HackCX (this option is always available but only works if HackCX is installed)
- Create VCA/SCA/etc. file using SCCTextX
- Pack
- Modify Structure
- Reindex
- Look up Reference (if GoFish is installed – finds all references to the form/class)
- NewObject (pastes call to NewObject into the current code window)
- Create subclass
- Create duplicate class
- Descendant Classes
- Sibling Classes
- Where Used (where this class or any of its descendant classes are used)
- Add to or Remove from current project
- Open folder in Explorer
- CD to this folder
- Set scope to this folder

Go To Definition

Go To Definition is one of the most useful Thor tools at the same time as being one of the easiest to use. It allows you to point to any user-defined name that is referenced in your code, and go to (that is, display and/or edit) its definition. It can also be used to create new methods and properties in a form or class.

The mechanics are simple:

- Click on the name.

- Run Go To Definition.

Table 1 lists all the different types of names that can be searched for.

Table 1. Go To Definition allows you to point to any user-defined name that is referenced in your code, and go to its definition.

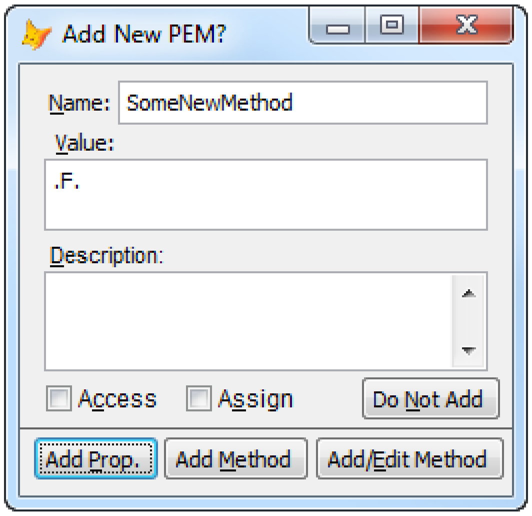
Type of Name	**Action taken**
Method or event	Opens up the method for editing; if there is no local non-default code, also opens up a txt file containing all the inherited code for the method.
Object	Selects that object, if possible, for display in PEM Editor and property window. There are some conditions where this fails: If the object is hidden beyond other objects or not otherwise visible or if it is on a different page of a pageframe.
Property	Same as selecting an object, but also tries to select that property in the PEM Editor grid. This only works if the PEM Editor form is open, and may also fail based on the filters in effect in the grid.
DODEFAULT	Opens up a txt file containing all the inherited code for the method.
PRG	Opens the PRG.
Procedure or Function in a PRG	Opens the PRG and highlights the start of the PROC or FUNC
Constant (#Define …)	Opens the #Include file, and highlights the constant
Form	Opens the form
Class	Opens the class, whether it is in a VCX or PRG
CREATEOBJECT or NEWOBJECT	Opens the class, whether it is in a VCX or PRG

If the search is to be conducted looking in file(s) other than the form or class being edited, the files in the active project, if any are searched; if there is no active project, then all files in the path are searched.

Creating New Properties and Methods

Go To Definition can also be used to create new properties and methods. Simply call Go To Definition when the cursor is in the name of a potential new property or method and the form for creating new properties or methods is opened; see **Figure 10**.

Author's note: I rely heavily on this tool. In fact, I use it to create most of my new properties and methods, rarely using PEM Editor for that purpose any more.



Name: SomeNewMethod
Value: .F.
Description:

Figure 10. Go To Definition can create new properties and methods if you execute it when the cursor is in the name of a potential new property or method.

Tools for Objects and PEMs

The tools in this group are used to copy and paste properties and method code from one object to another, to create new properties and methods, and to insert into your method code a reference to the object under the mouse.

The first three of these are closely tied to each other.

Copy (for comparing and pasting)

This tool makes a copy of the non-default properties and method code for the currently selected object when you are editing a form or class. It is a quiet tool; you will see a flash at the top of the screen as it operates, but nothing more.

This is similar in concept to Ctrl+C when copying text in a code window. *Author's note: I use Alt+C as the hot key for this tool.*

Compare with copied object

Once you have copied the properties and methods from one object, you can select another object (which need not be in the same form or class) and use this tool to see which properties and methods are different between the two objects.

Paste properties and method code

This tool is much like the previous one, *Compare with copied object*, except that the form that pops up allows you to select which properties and methods you want to paste into the new object. If the object being pasted into is the top level form or class, as distinct from an object in the top level form or class, you can create new properties and methods.

This is similar in concept to Ctrl+V when copying text in a code window. *Author's note: I use Alt+V as the hot key for this tool.*

Add property / method

This tool pops up a small form for creating properties and methods (a stripped down version of PEMEditor, but with all the same capabilities for creating properties and methods). See Add property / method.

Insert full name of object under mouse

This interesting little tool inserts into your code window a relative reference to the object that is under the mouse when you execute the tool. Thus, the tool can only be executed by hot key. The steps are:

- Click in the code window where you want the object reference to be inserted.
- Move your mouse over the object you want to reference.
- Execute the tool.

This tool can be used to obtain references to objects when editing a form or method and also forms you are executing. *Author's note: I use this when editing method code and also to paste a reference to an object in an executing form into the Command window.*

Tools for Parent Classes and Containing Classes

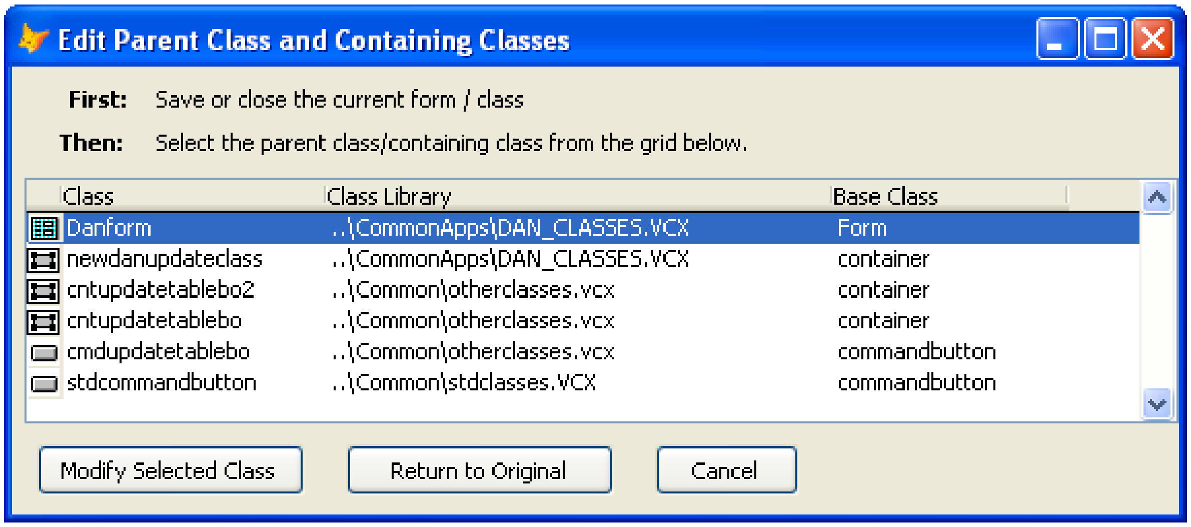
There are a number of tools that provide unique features for working with child classes, parent classes, and containing classes.

Edit Parent and Containing Classes

This tool opens a form you can use to open parent classes and containing classes of an object for editing. "Containing classes" in this context are to be understood as all class definitions in which the object is found.

The intent is to provide access to all classes in which properties or methods of the object could be defined or set. So, for example, for a control that's a member of a container class dropped onto a form, all of the parent classes of the control appear as well as the container class and any of its parent classes which contain the object, but the form and its hierarchy do not.

The form shows the entire parentage of the object, including parent classes and containing classes. **Figure 11** shows the form as it appears for a button inside a container on a form. The button is based on cmdUpdateTableBO, which is a child of stdCommandButton. It's inside an object of class newDanUpdateClass, which is derived from cntUpdateTableBO2. That class is derived from cntUpdateTableBo. The container is on a form based on class DanForm.



First: Save or close the current form / class

Then: Select the parent class/containing class from the grid below.

Class	Class Library	Base Class
Danform	..\CommonApps\DAN_CLASSES.VCX	Form
newdanupdateclass	..\CommonApps\DAN_CLASSES.VCX	container
cntupdatetablebo2	..\Common\otherclasses.vcx	container
cntupdatetablebo	..\Common\otherclasses.vcx	container
cmdupdatetablebo	..\Common\otherclasses.vcx	commandbutton
stdcommandbutton	..\Common\stdclasses.VCX	commandbutton

Figure 11. Edit Parent Classes shows all parent classes and containing classes for the current object.

As the directions on the form indicate, to open any of these classes, you:

1. Save or close the form or class you are working on (since FoxPro does not permit you to open a parent class or containing class otherwise)
2. Select the class (double clicking also works).

This form remains open after you have selected a class. This allows you to continue to open any of the other classes as well as allowing you to return to the class or form you were originally editing.

Compare with Parent Class

This tool compares the properties and method code from the currently selected object with its parent class. The grid that pops up shows all non-default properties and methods and indicates which are the same as in the parent class, allowing you to reset them to default.

Re-Define Parent Class

This tool is used to re-define the parent class of an object (as long as it does not contain any child objects, and is not a form or class).

The tool opens a dialog form to find the new parent class for your object. (This dialog form is a predecessor to the much newer tool, Finder.) Once the new parent class is selected, a new object is created, all properties and methods are copied from the old object to the new one, the old object is removed, and the new object is renamed to have the same name as the old object.

Tools for Working with Highlighted Text

There are more than a dozen Thor tools that deal directly with highlighted text in a code window. While most are very easy to understand, there are just too many to keep track of. Thus, two Pop-up Menus created by the tool *Create Sample Menus* create some organization out of the chaos.

"Extract to …" Pop-up Menu

This pop-up menu provides access to three different tools that extract the currently highlighted text and create a new method, variable, or constant from it. *Author's note: I use Ctrl+E as the hot key for this pop-up Menu.* See **Figure 12**.

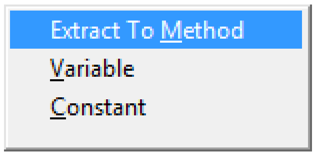
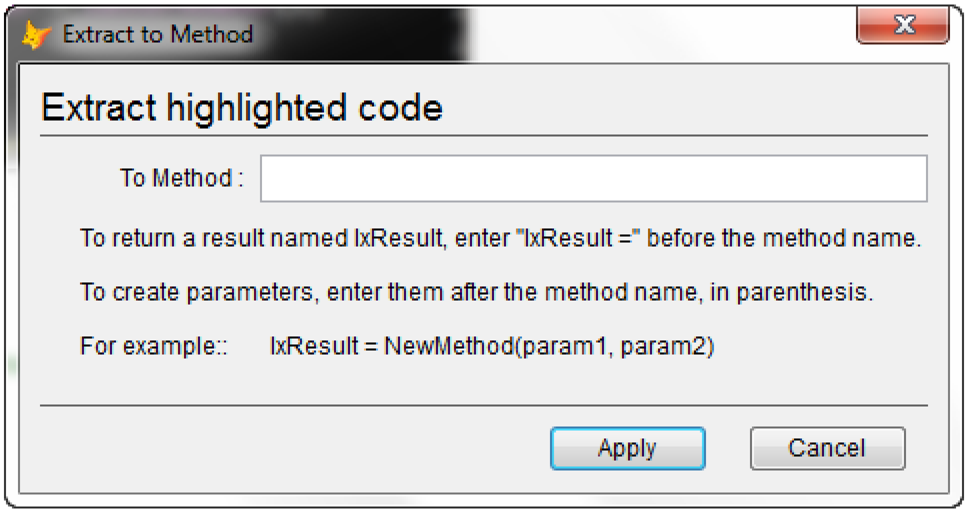


Figure 12. The "Extract to …" pop-up menus can extract the currently highlighted text and create a new method, variable, or constant from it.

Extract to Method

This tool, which only works on SCX or VCX code, extracts the currently highlighted code, creates a new method from it, and replaces the highlighted code with a reference to the new method.

As shown in **Figure 13**, you can indicate the parameters to be created in the new method as well as the result; these parameters and result are included in the code that is pasted in, replacing the highlighted code.


Extract to Method

Extract highlighted code

To Method :

To return a result named lxResult, enter "lxResult =" before the method name.

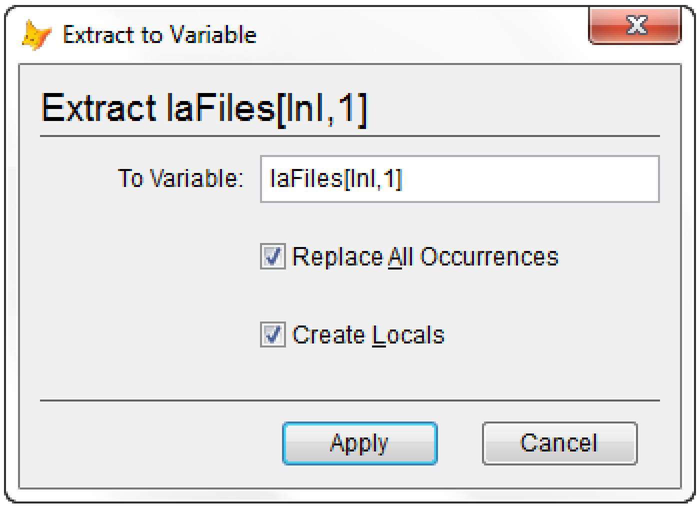
To create parameters, enter them after the method name, in parenthesis.

For example:: lxResult = NewMethod(param1, param2)

Figure 14. Extract to Method extracts the currently highlighted text and create a new method; the form allows you to also specify the parameters and result.

Extract to Variable

This tool extracts the highlighted text from within a single line of code and creates a variable that is assigned the highlighted text, as shown in **Figure 14**.


Extract to Variable

Extract laFiles[lnI,1]

To Variable: laFiles[lnI,1]

☑ Replace All Occurrences

☑ Create Locals

Figure 14. Extract to Variable extracts the currently highlighted text and creates a new variable on the line above; you can optionally replace all references and also add the new variable into a LOCAL statement.

Extract to Constant

This tool extracts the highlighted text from within a single line of code and creates a #Define statement, either in the current method or one of the .H files, as shown in **Figure 15.**

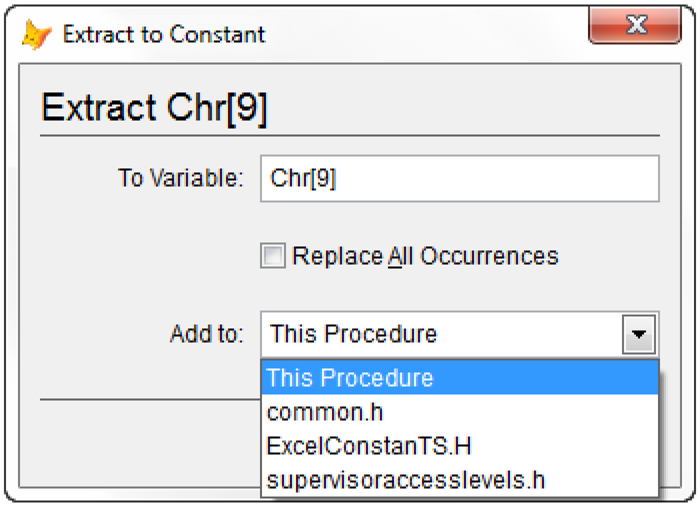


Figure 15. Extract to Variable extracts the currently highlighted text and creates a new constant, either in the current method or in one of the related .H files

"Highlight" Pop-up Menu

This pop-up menu provides access to a large number of closed related tools that either highlight text in the current method (the leading items) or modify the currently highlighted text (the remaining items). *Author's note: I use Ctrl+H as the hot key for this pop-up Menu.* See **Figure 16**.

These tools are pretty much self-explanatory, or can be easily understood with a little testing. Thus, descriptions are provided for only two of them.

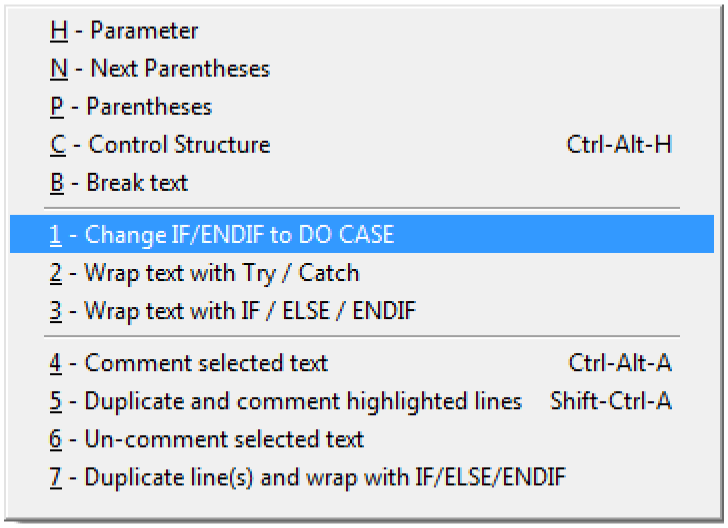

H - Parameter
N - Next Parentheses
P - Parentheses
C - Control Structure Ctrl-Alt-H
B - Break text
1 - Change IF/ENDIF to DO CASE
2 - Wrap text with Try / Catch
3 - Wrap text with IF / ELSE / ENDIF
4 - Comment selected text Ctrl-Alt-A
5 - Duplicate and comment highlighted lines Shift-Ctrl-A
6 - Un-comment selected text
7 - Duplicate line(s) and wrap with IF/ELSE/ENDIF

Figure 16. The "Highlight" provides a large number of closely related tools for either highlighting text or for modifying the currently highlighted text.

Comment Highlighted Text

This tool is an enhanced version of the Comment menu item in the default VFP context menu in a code window that comments a block of highlighted text.

This tool handles indentation differently in that the commented code is kept aligned with the original code instead of at the left margin.

It also inserts a new comment line before the block of highlighted text, inserts today's date, and leaves the cursor at that position for additional comment. You can change the text of this inserted line on the Options page for this tool in the Configuration form.

There is a companion tool, *Un-comment Highlighted Lines*, for removing the comments that were added by this tool. You can avoid using this second tool, however, by selecting an option on the Options page for *Comment Highlighted Text* – when this option is set, if the highlighted text is already commented, the comments are removed.

Close Control Structure

This tool closes the current control structure by pasting ENDIF, ENDFOR, ENDSCAN, ENDTRY, ENDWITH, ENDCASE, ENDDO, ENDTEXT, and so on, as appropriate.

The pasted text is sensitive to the setting for keywords in Visual FoxPro's Beautify, and can be modified by a Plug-in.

This tool has a sibling, *Highlight Control Structure*. Each reproduces behavior available in Visual Studio.

Standardizing Your Code

FoxPro gives us native Beautify to apply some familiar formatting to procedure code , fixing indentation and the selection of case for keywords and user-defined symbols.

There is a suite of Thor utilities that provide significant additional formatting features beyond what is available from native Beautify. All of these features, which are listed below, are available from the single tool *BeautifyX*; some are also available as separate, stand-along Thor tools. The features can be used in any combination, as they are all selectable on the Options page of the Configuration form (click on "BeautifyX" on the left).

These features are accessed only as part of BeautifyX:

- Apply native VFP Beautify (fixing the bug for mixed-case on continuation lines)
- Add a space around operators
- Remove spaces before commas
- Standardize use of string delimiters and the definitions for NOT and NOT EQUAL
- Align semi-colons in a standard column (except for lines that are too long)

- Create additional indentation for each level of parentheses on continuation lines
- Apply highly customizable formatting for SQL-SELECT, SQL-UPDATE, SQL-DELETE, and REPLACE statements, (conditionally within TEXT / ENDTEXT structures), by providing special indentations, both for VFP SELECT and within TEXT/ENDTEXT blocks.
    - for new fields and their continuation lines
    - for SELECT, UPDATE, DELETE, and REPLACE keywords in general (FROM, WHERE, INTO)
    - for some specific keywords individually (JOINs, UNION, interior SELECTs, and SET)
    - for nested level of parentheses
    - for alignment of 'AS fieldname' clauses in SELECT statements
    - for alignment of 'WITH' clauses and the expressions that follow them in REPLACE statements
- These features are available as stand-alone Thor tools and can also be invoked automatically by BeautifyX
- *Create Locals* – creates the list of LOCAL variables in a procedure. A plug-in is provided so that you can format the list of these variables to suit your taste.
- *Add M-Dots* – adds M-Dots to all local variables referenced
- *Check for RETURNs between WITH/ENDWITH* – executing a RETURN inside a WITH/ENDWITH block can cause a latent C5; this tool identifies any such problems. A related tool, "Remove WITH/ENDWITH", corrects such problems by removing the WITH/ENDWITH statements and adjusting the intervening code.
- *Apply Custom Keyword List* and *Add words to "Custom Keyword List* – the Custom Keyword List is a new concept explored in the next section.
-

Custom Keyword List

The Custom Keyword List, a concept unique to Thor, is a simple table with the list of all words from your programming universe, where each word is saved with the mixture of upper/lower case that you prefer. This table can be created easily, updated automatically, and applied to any block of code.

There are a number of Thor tools to create the Custom Keyword List and to keep it current, implemented so that your manual intervention is not required:

- Add highlighted word
- Add field names from current table
- Add PEMs from current class or form

- Add all words in code window (See also previous section; this can be run automatically as part of BeautifyX)
- Add all words from folder or project
- Apply Custom Keyword List to code window (See also previous section; this can be run automatically as part of BeautifyX)
- Browse Custom Keyword List

There is also an option in PEM Editor, on the Processing | General page, that causes PEM Editor to update the Custom Keyword List when you add new properties or methods.

See also the Custom Keyword List for more on this topic

Code Mechanic

Tool "Code Mechanic" can apply three Thor tools, listed below, to all code in all files in a selected project or folder:

- BeautifyX
- Create Locals
- Add M-Dots

Author's note: I use F6 as the hot key for BeautifyX and make it a practice to always use the hot key before saving a method or procedure, making sure that the formatting features I prefer are always applied.

Miscellaneous Tools

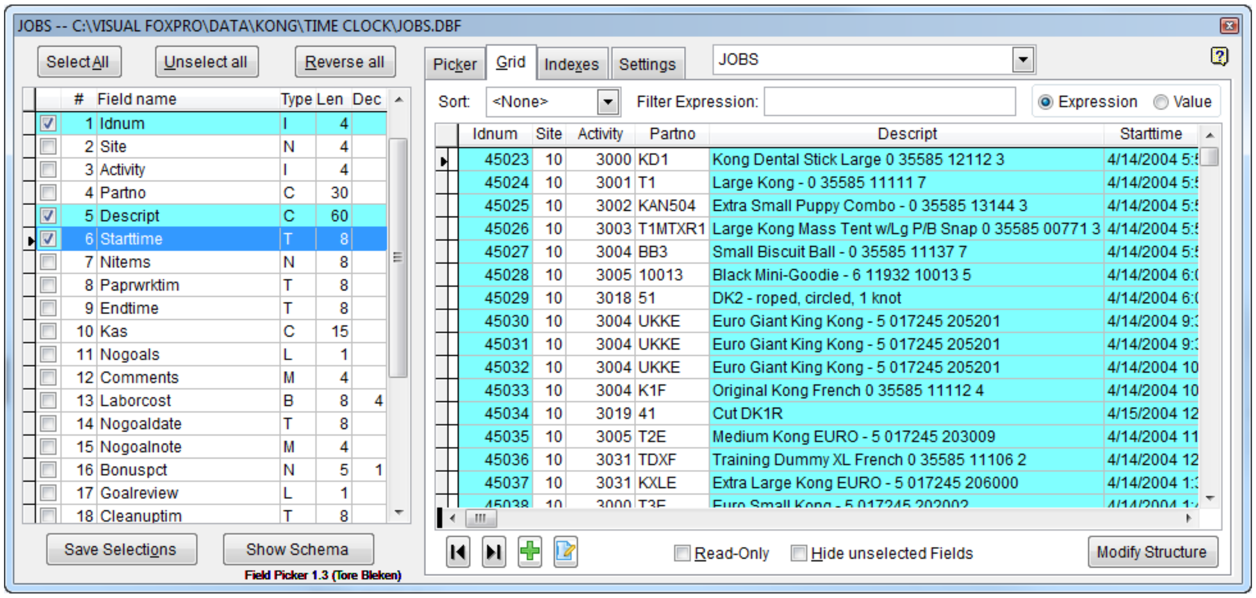
Finally, there's the list of those isolated, unrelated tools that don't fit into any category.

Super Browse

This tool has been described as "browse on steroids", combining a grid to browse a table with a wide number of other features; see **Figure 17.**

- Display of the structure of the table
- Filtering the rows in the table
- Selecting the sort order and sorting on any column in the grid
- Buttons to navigate the grid, adding or duplicating records, or modifying its structure.
- Pop-up form to edit any record (using Dynamic Forms)
- A separate page to create SQL statements such as Select, Update, Insert, and Create, using the grid on the left to select the fields to be used.

- Editing of the case of the field names in the grid on the left, auto-saving your changes to the Custom Keyword List described previously.



Sort: <None> Filter Expression: [● Expression] ○ Value

	#	Field name	Type	Len	Dec
✓	1	Idnum	I	4	
	2	Site	N	4	
	3	Activity	I	4	
	4	Partno	C	30	
✓	5	Descript	C	60	
✓	6	Starttime	T	8	
	7	Nitems	N	8	
	8	Paprwrktim	T	8	
	9	Endtime	T	8	
	10	Kas	C	15	
	11	Nogoals	L	1	
	12	Comments	M	4	
	13	Laborcost	B	8	4
	14	Nogoaldate	T	8	
	15	Nogoalnote	M	4	
	16	Bonuspct	N	5	1
	17	Goalreview	L	1	
	18	Cleanuptim	T	8	

Idnum	Site	Activity	Partno	Descript	Starttime
45023	10	3000	KD1	Kong Dental Stick Large 0 35585 12112 3	4/14/2004 5:5
45024	10	3001	T1	Large Kong - 0 35585 11111 7	4/14/2004 5:5
45025	10	3002	KAN504	Extra Small Puppy Combo - 0 35585 13144 3	4/14/2004 5:5
45026	10	3003	T1MTXR1	Large Kong Mass Tent w/Lg P/B Snap 0 35585 00771 3	4/14/2004 5:5
45027	10	3004	BB3	Small Biscuit Ball - 0 35585 11137 7	4/14/2004 5:5
45028	10	3005	10013	Black Mini-Goodie - 6 11932 10013 5	4/14/2004 6:0
45029	10	3018	51	DK2 - roped, circled, 1 knot	4/14/2004 6:0
45030	10	3004	UKKE	Euro Giant King Kong - 5 017245 205201	4/14/2004 9:3
45031	10	3004	UKKE	Euro Giant King Kong - 5 017245 205201	4/14/2004 9:3
45032	10	3004	UKKE	Euro Giant King Kong - 5 017245 205201	4/14/2004 10
45033	10	3004	K1F	Original Kong French 0 35585 11112 4	4/14/2004 10
45034	10	3019	41	Cut DK1R	4/15/2004 12
45035	10	3005	T2E	Medium Kong EURO - 5 017245 203009	4/14/2004 11
45036	10	3031	TDXF	Training Dummy XL French 0 35585 11106 2	4/14/2004 12
45037	10	3031	KXLE	Extra Large Kong EURO - 5 017245 206000	4/14/2004 1:3
45038	10	3000	T3E	Euro Small Kong - 5 017245 202002	4/14/2004 1:

Figure 17. Super Browse has been called "Browse on steroids".

Super Browse, like Browse, can be invoked to work on the current table. A much more interesting use, however, is to invoke after you have clicked on a table name in your code. Super Browse will work on that table, even opening it if necessary.

Author's note: Super Browse is one of my constant companions; I use Shift+F12 as the hot key for it.

See the Super Browse Home Page for a more complete description of its features.

Set Object Size and Position

This tool opens the Object Size and Position form (shown in **Figure 18**), which combines almost all of the features of the native Visual FoxPro Format Menu into one form. This form was created so that the features from the Format Menu could be available with one click, and to simplify access to those that may be used multiple times consecutively, such as increasing a dimension by a pixel.

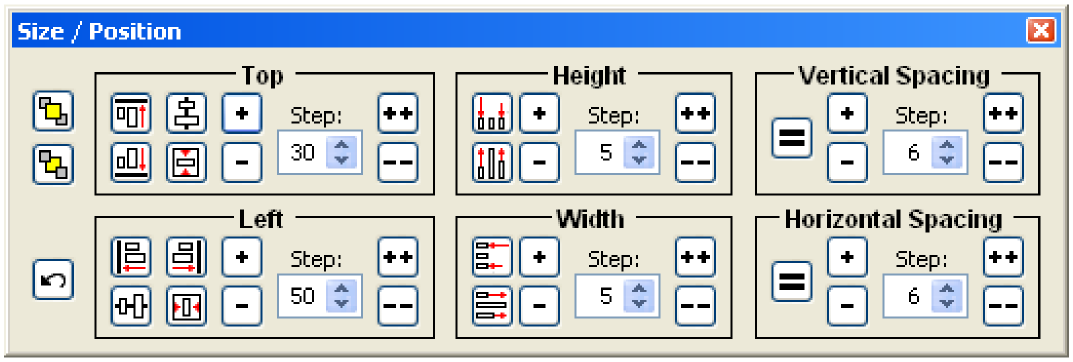


Figure 18. The Object Size and Position form combines many of the features of the Native Visual FoxPro Menu into one form.

The form also includes features not found in the Format Menu. You can increment or decrement each of the dimensions and spacings by a single pixel at a time or by a larger step size. There is an Undo button, which can reverse any actions taken since the form received focus. Each click reverses a single action.

The form is dockable. The six containers realign themselves according to the width of the form, allowing it to be docked either horizontally or vertically. You may need to adjust the height or width after docking.

Save as Class

Saves the currently selected object as a new class, for use when you are editing a class, since the menu item "Save as Class..." is not available in the File menu pad.

The pop-up form allows you to enter the name of the new class and the class library where it is to be stored.

Note that you can create a duplicate of a class by opening it and using this tool without selecting any child objects.

Find Objects

This tool opens a form for searching for objects in a form or class based on the values of any of their properties. The results form displays the list of objects found in a grid for easy review. From this results form, you can modify the values of properties for these objects, open methods for editing, or select any of the objects found.

The first step is to use the search form to enter the search expression. Search expressions are governed by a few simple rules, which provided considerable power and flexibility:

- The search expression you enter is executed for each object in the form or class. If the expression returns a non-empty result, that object is considered a match.
- The expression is executed within a WITH/ENDWITH structure for each object, so that properties can be referenced simply as .ControlSource or .Caption, for instance.

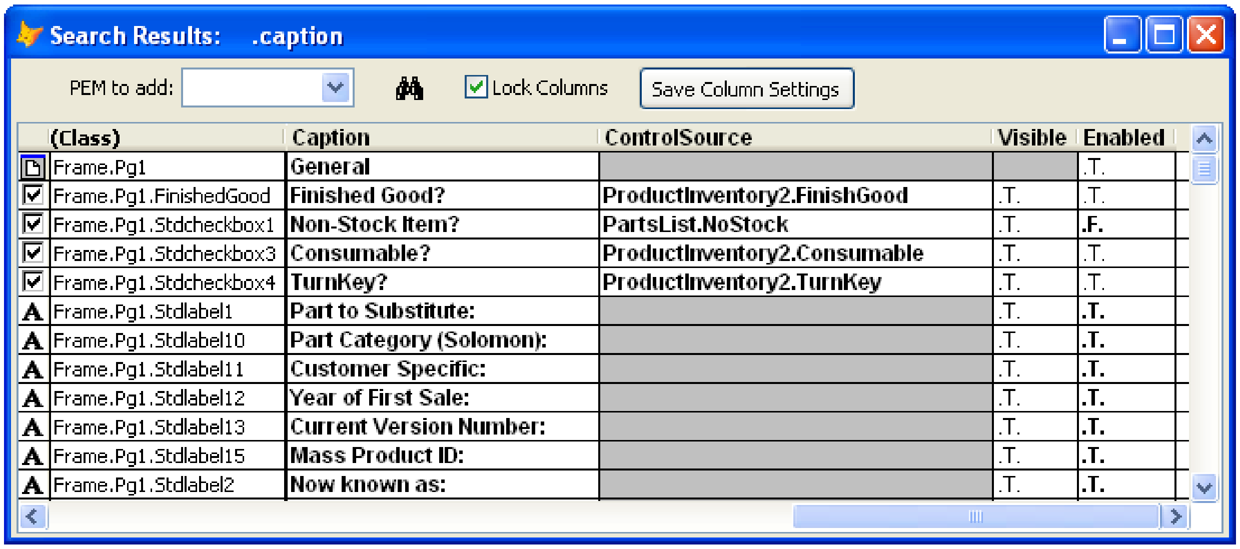
- The expression is executed within a TRY/CATCH structure, so there is no need to ensure that the properties you reference exit.
- The search expression can include any of the auxiliary functions shown in the form
- The search expression can call UDFs either to select objects or to modify their properties. A search expression that only calls a UDF causes that function to be called for each object.

The form that displays the results of the search has one line for each object, and columns that show the name of the object and various other properties, as shown in **Figure 1929**.

Property names referenced directly such as .Caption and .ControlSource in **Error! Reference source not found.**, are automatically shown as columns in the grid. Use the "PEMs to add" combobox to add other properties or methods or use the context menu on a column to remove it. You can save the list of columns that are saved, along with their order and width, by clicking "Save Column Settings."

You can edit a property by double-clicking it. Unfortunately, the change in the value may not be displayed in the grid until you move to another row in the grid. Similarly, you can edit the code for a method by double-clicking it.

You can also select (in the Property Sheet and PEM Editor) the object named in the first column by double-clicking it. When possible, this also causes the object to be selected in the form or class being edited.



(Class)	Caption	ControlSource	Visible	Enabled
Frame.Pg1	General			.T.
Frame.Pg1.FinishedGood	Finished Good?	ProductInventory2.FinishGood	.T.	.T.
Frame.Pg1.Stdcheckbox1	Non-Stock Item?	PartsList.NoStock	.T.	.F.
Frame.Pg1.Stdcheckbox3	Consumable?	ProductInventory2.Consumable	.T.	.T.
Frame.Pg1.Stdcheckbox4	TurnKey?	ProductInventory2.TurnKey	.T.	.T.
Frame.Pg1.Stdlabel1	Part to Substitute:		.T.	.T.
Frame.Pg1.Stdlabel10	Part Category (Solomon):		.T.	.T.
Frame.Pg1.Stdlabel11	Customer Specific:		.T.	.T.
Frame.Pg1.Stdlabel12	Year of First Sale:		.T.	.T.
Frame.Pg1.Stdlabel13	Current Version Number:		.T.	.T.
Frame.Pg1.Stdlabel15	Mass Product ID:		.T.	.T.
Frame.Pg1.Stdlabel2	Now known as:		.T.	.T.

Figure 192. The results form shows one row for each object matching the search criteria.

Coordination with PEM Editor

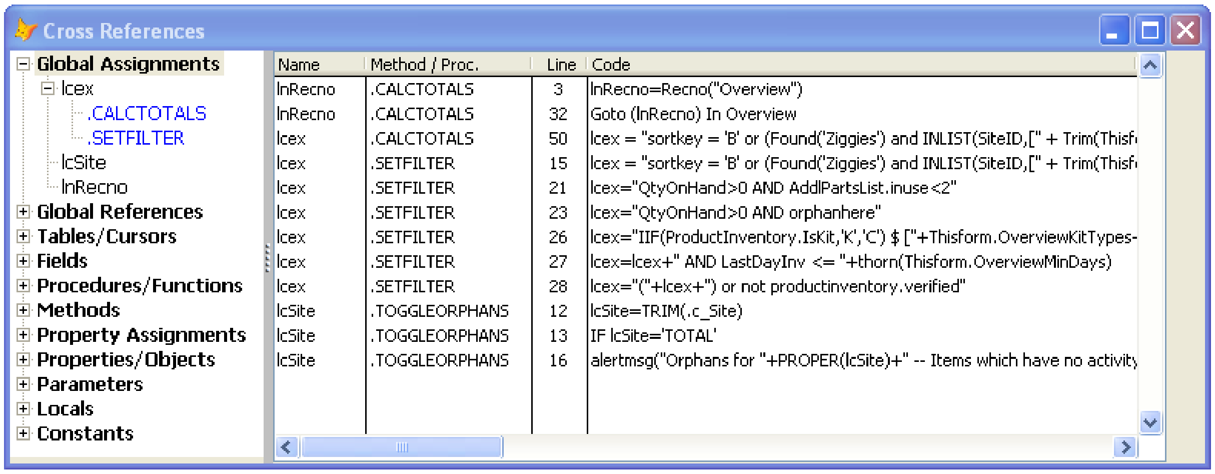
If the PEM Editor form is open when a search is done, it shows the list of all properties and methods for the objects shown in the results form, just as if you had selected a number of objects on the form or class being edited using the mouse or keyboard. This allows you to change the value of a property for all selected objects all at once, either by double-clicking the property, or by resetting the property to default in the context menu.

You can add properties to the results form by Ctrl-Clicking them in the PEM Editor form.

When you have selected a number of objects, whether by mouse or keyboard, in the form or class you are editing, you can use the results form to edit the values for single properties individually. Even if the results form is not open, Ctrl-Clicking on a property in PEM Editor when you have selected multiple objects opens the Search Results form and adds the property as a column there.

Cross References

This tool is similar to the Code References results form and shown in **Figure 20**, in which all names referenced in code are listed by category, such as Global Assignments or Global References. You can filter the references shown in the grid by clicking on the nodes in the treeview, and double-clicking on any row in the grid to go directly to the reference. The form is always on top and may be moved outside of the Visual FoxPro screen.



Name	Method / Proc.	Line	Code
lnRecno	.CALCTOTALS	3	lnRecno=Recno("Overview")
lnRecno	.CALCTOTALS	32	Goto (lnRecno) In Overview
lcex	.CALCTOTALS	50	lcex = "sortkey = 'B' or (Found('Ziggies') and INLIST(SiteID,[" + Trim(Thisf
lcex	.SETFILTER	15	lcex = "sortkey = 'B' or (Found('Ziggies') and INLIST(SiteID,[" + Trim(Thisf
lcex	.SETFILTER	21	lcex="QtyOnHand>0 AND AddlPartsList.inuse<2"
lcex	.SETFILTER	23	lcex="QtyOnHand>0 AND orphanhere"
lcex	.SETFILTER	26	lcex="IIF(ProductInventory.IsKit,'K','C') $ ["+Thisform.OverviewKitTypes-
lcex	.SETFILTER	27	lcex=lcex+" AND LastDayInv <= "+thorn(Thisform.OverviewMinDays)
lcex	.SETFILTER	28	lcex="("+lcex+") or not productinventory.verified"
lcSite	.TOGGLEORPHANS	12	lcSite=TRIM(.c_Site)
lcSite	.TOGGLEORPHANS	13	IF lcSite='TOTAL'
lcSite	.TOGGLEORPHANS	16	alertmsg("Orphans for "+PROPER(lcSite)+" -- Items which have no activity

Figure 20. Cross References displays a form showing all names referenced in code, grouped by category.

The names that appear in this report are all names referenced in code, excluding comments and character strings, subject to the following:

- FoxPro keywords (from table Home() + 'WIZARDS\FDKEYWRD') are excluded. This may filter out more than is desired, such as keywords 'Name' or 'Width'.
- References to objects ("This.lblName.SomeProperty") or tables ("Customers.City") are displayed as single items in the report rather than treating each level of the object hierarchy separately.
- References within WITH/ENDWITH blocks are resolved to include the object named in the WITH statement. This applies even to embedded WITH/ENDWITH blocks.

There are some configuration options available in the context menu of the left pane (click anywhere that is *not* a node) to control the appearance of the nodes when the form opens. You can specify which categories are automatically expanded when the form opens. (By default, all categories are collapsed). You can also select which single category is selected when the form opens, causing the references in that category to appear in the grid.

Author's note: I use Cross References to quickly identify global assignments and other global references in unfamiliar code.

Refactoring between Child Classes and Parent Classes

A number of tools can be combined to simplify the task of re-factoring code between child and parent classes. The steps are:

1. In the child class, use tool "Copy (for comparing and pasting)".
2. Use tool "Edit Parent and Containing Class" to get the list of all parent classes. (Leave this form open for later re-use in step 7).
3. Close the child class.
4. Use the form opened in step 2 to open the parent class of interest.
5. Use tool "Paste properties and method code" to paste into the parent class any or all properties or methods.
6. Save and close the parent class.
7. Use the form opened in step 2 to re-open the original class.
8. Use tool "Compare with Parent Class" to reset to default any properties or methods that are now identical to the parent class.

The "Refactoring" pop-up menu, shown **Figure 21**, can be used to remind you of these steps.

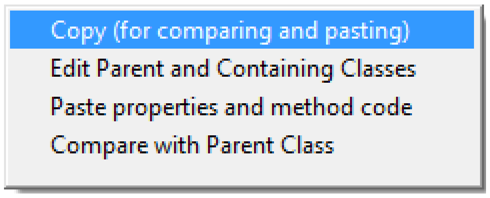

Copy (for comparing and pasting)
Edit Parent and Containing Classes
Paste properties and method code
Compare with Parent Class

Figure 21. The "Refactoring" reminds you of the steps to refactor code between a class and its parent.

Use the Thor Forum!

If you have questions, comments, suggestions, or bug reports about Thor or any of the Thor tools, or if you have tools to contribute to Thor, please do so using the Thor Forum, a link that is also available from the Thor menu pad. We love to hear from you.

Summary

Thor offers a wide number of tools to enhance the IDE and thus your productivity; tools for searching, tools for handling objects and PEMs, tools for working with parent classes and container classes, and a host of others. Many of these tools provide capabilities that can't be found any place else.

Thor provides a number of different ways for you to use these tools – hot keys, menu pads on the system menu, the Thor tool bar, pop-up menus (a feature unique to Thor), and the Thor Launcher – so that you can use them in a way that seems natural to you.

Thor also provides a single place for identifying new projects in VFPX and updates to existing projects and downloading them.

Finally, there is a place for you to address any questions or comments at the Thor Forum.

Biography

Jim Nelson spent the first thirty years of his professional life programming in APL, a long-since extinct programming language. For the last twenty of those years, he was the sole developer for a company that handled workers' compensation self-insurance. His claims handling system was the foundation on which the entire company was built. During those years, he had a reputation throughout the APL community as a builder of developer utilities not available within the APL language.

His VFP career began in 2003, working for the Kong Company in Golden, Colorado. He has worked full-time for them since then, acting as their software development department.

His involvement in VFPx projects (including Thor, IntellisenseX, Finder, PEM Editor, and FoxCharts) mirrors his long time interest in creating developers' tools.

Jim lives in Thousand Oaks, California, with his wife of 41 years.