STZUE – STEEZWARE ULTRAEDIT TOOLKIT
TOOL INSTRUCTIONS

31 March 2020 by Scott McDonell

CONTENTS

Overview

stzUE – *pronounced steeze-U-E* – is a set of UltraEdit tools that provide rich support for editing ProvideX and PxPlus source code (text) programs and Nomads libraries. The tools are based on those developed by Sage Software for the development of its Everest (4.x) version of MAS 90. Sage abandoned the tools in favor of the plug-in developed for the Eclipse IDE, but steezware teamed up with ERP Developers to update the tools to work with PxPlus as well as ProvideX.

stzUE is used by steezware and ERP Developers for developing PxPlus applications and Sage 100 / MAS 90 programs.

Tools

There are seven utility programs in the stzUE toolkit:

stzUEsave Checks the syntax of the current source file in the editor, and optionally compiles it, backs it up, and checks it for user-defined warnings.

stzUEsearch Searches programs and text files using complex search criteria.

stzUEinsert Displays data from a Pvx data dictionary, a message library, or a Pvx class and allows selected items to be inserted into the current source file opened in the UE editor.

stzUEinvoke Invokes the Pvx command-line, the Nomads designer, data dictionary maintenance, or PxPlus online help. Also allows user-defined actions to be added to the invoke-tasks window.

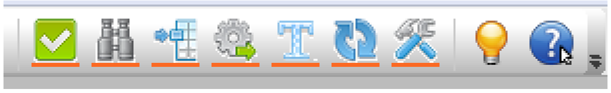
stzUEpgmtxt Converts a program file opened in the editor to a text file, optionally formatting it and removing its line numbers.

stzUEtrace Reformats a Pvx trace output file for easier display in the UE editor, allowing for code folding and color-coding of syntax.

stzUEproject Settings for maintaining programs and Nomads libraries are defined as projects, which are maintained with this tool. All stzUE tools except for stzUEtrace require an interpreter to be defined in a project.

Running the stzUE Tools

The buttons for the seven stzUE tools are found near the right end of the “Main” toolbar.



As you hover over each button, a floating tip indicates the tool name and a brief description. The stzUE tool icons have a steezware-orange line across the bottom, to distinguish them from other UE buttons.

EULA

The first time you select any tool, you are prompted to accept the End-User License Agreement.

Donate Window

After using the tools for about a week, a window will appear offering you the opportunity to make a donation to steezware for the stzUE toolkit. stzUE is free to use and there is no obligation or expectation that a donation be made, but keep in mind that literally hundreds of hours have been spent developing the tools and steezware is committed to continued enhancement of the tools and to respond to user questions and submissions in its support forum.

You will only be presented the donate window one-time, unless you delete the stzOptions.sw1 settings file or install stzUE on a new computer. Feel free to completely ignore this window, but if you want to donate at a later time, you may do so at steezware.com/donate.html

Tool Persistence

Each tool is invoked as a standalone Windows process, so when you close UltraEdit, any active stzUE tool will remain open and should be manually closed.

Support Forum

If you have a question about the features or operation of a tool that is not explained adequately in this document, please search the stzUE/Tools forums on steezware.com/support.html, or add a new topic of discussion if you do not find an answer to your search.

Installation Instructions

The instructions for installing stzUE are found in the stzUE-Install.PDF file, available at steezware.com/support.html.

Help

All stzUE windows include a help button that allows access to this document and the web-based support forums. Please use the forums to search for answers and to submit new questions, to help create a database of the real-world usage of the stzUE tools and to post suggestions on how to improve stzUE.

1 / stzUEProject

At the core of almost all stzUE tools is a connection to a project. The principal purpose of a stzUE project is to define the interpreter used for saving and searching programs, but projects also allow you to customize tool usage based on the tasks you are working on.

Note that stzUE projects should not be confused with UltraEdit projects – described below in Appendix D.

1 / Project Tab

Pvx Interpreter and Optional INI Files

Use the lookup buttons to select the ProvideX or PxPlus executable file to use. You may also specify INI files for running Pvx tasks or the Nomads designer.

Link to UE Project

If you are using UltraEdit projects, you can bypass the stzUE project selection prompt by linking your stzUE project to a UE project. All of the stzUE tools – with the exception of the trace formatter – use the current UltraEdit project to fetch settings from the stzUE project to which it is linked. If you are not familiar with UE projects, see Appendix D.

Linking stzUE and UE projects greatly increases the efficiency of the stzUE tools by allowing you to simply change the UltraEdit project you are working on, then running a tool and having the interpreter, folders and other settings automatically selected for you.

2 / Compile Tab

The settings on this panel are used by stzUEsave when compiling programs.

Password and Owner Code

The optional fields are used for securing compiled programs.

Refer to Appendix A for instructions on how to create the stzUEsave template programs if using owner codes.

Save Program
↳ Check Subfolders

If you select the Save Program option, you must enter a folder in which to save the compiled program. Selecting the “Check Subfolders” option instructs stzUEsave to scan under the save folder for any occurrence of the file being saved. This is useful – and necessary – when your programs are located in subfolders under a parent folder. For example, if you are saving AP_Vendor.pvc which resides in the “AP” subfolder under your specified save folder, stzUEsave will find it and save to that location.

Save Program
↳ Check Subfolders
↳ Exclude Subfolders

When checking subfolders during a save, if there are any that you do not want to be included in the scan, enter them here - separated by commas. Example:
ARCHIVE,BKUP2012,TMP

Save Program
↳ Check Subfolders
↳ Source File Leading Characters

This setting allows you to choose a number from 1 to 9 to indicate the number of characters that a program name begins with from which a subfolder name will be contrived. For example, if your “Save Folder” is defined as “c:\BizApps” and your AP_Vendor.pvc program resides in its “AP” subfolder, selecting 2 in this setting will cause stzUEsave to use the “c:\BizApps\AP” folder for saving the program.

Save Program
↳ Check Subfolders
↳ Extensions to Strip

UltraEdit and stzUE do not require source files to have a special extension, as does Eclipse. But, if you have source files with extensions that you want saved as programs with no extension, enter the extensions here - separated by commas. Example:
PVT,TXT,SRC

3 / Syntax Tab	*The settings on this panel are used by stzUEsave when initially checking the source file for syntax errors.*
Mark Syntax Errors with \ERR\	If errors are found in the source file, selecting this option inserts "\ERR\" at the beginning of the line containing the error. Note that if this tag is inserted into the source file, UltraEdit will prompt to reload the file so you can see the changes. UE may also prompt that the program is no longer available – you must click [Cancel] to keep the file open in UltraEdit. Also note that you must manually remove the \ERR\ flags inserted in your source code when you correct a line that contains the tag.
Click to Maintain Warnings File	While checking the text file for syntax errors, you can optionally specify that other items be found – and optionally modified – in the compiled code. The warnings are stored in the c:\steezware\stzUE\install\stzUEwarnings.txt file. Note that the compiled program is modified, not the source file.
Check UltraEdit Settings	When UE and stzUE are used by groups of people working on the same code, it is important that all users have identical settings for tabs and indentation and so forth. Select this option to cause stzUEsave to check that the user's settings are the same as those specified on this page.
4 / Tools Tab	
Run External Task Settings	Using stzUEinvoke to run the Nomads, Pvx Console Mode, and Data Dictionary Maintenance external tasks can be optimized by specifying the folder in which each task is started.
Insert Text Settings	If specified, stzUEinsert will default to the message library defined here. In addition, stzUEinsert uses the Run External Task Settings / DDF Folder defined on this panel to default the data dictionary location.
Nomads Panel Selector	The Library Copy-To Folder is used after editing a panel using the stzUE Nomads panel selector, which is invoked via an UltraEdit macro as described in Chapter 8 / Nomads Panel Selector elsewhere in this document. The macro works by opening a window from which a panel is selected for editing when you double-click on a Nomads library in the UE navigation pane. When you close Nomads, if any changes were made and saved, stzUE creates a copy of the library in the copy-to folder. This is useful if you have Nomads libraries in a "source" folder and want to push them to a build area for testing. Note that the rules for saving a Nomads library to a subfolder use the settings on the Compile tab for saving programs.
5 / UDBs	
Buttons 1 thru 4	Up to four user-defined buttons can be defined for each project. See the [Settings] section on stzUEinvoke below for more information.
Copy From	Select UDBs from another project or from the global buttons defined in stzUEinvoke to populate the UDB settings on this panel.
6 / Notes	*Up to 600 characters can be used for notes and comments about the project.*
Main Window	*The following options are invoked from buttons outside of the folder, on the main window.*
[New]	Creates a new project. If your new project closely mimics an existing project, using the [Copy] option could save some effort. Or, if you have a fellow programmer with a project definition useful to you, have him [Share] it then [Import] the file and change the project properties as needed.
[Copy]	Creates a copy of the current project with a new name.
[Delete]	Deletes the current project.

[Share] and [Import] Buttons

Use the [Share] option to create a file that contains all of the settings of a project. This file can then be shared with another stzUE user to import into his projects. When a project is imported, its name can be overridden with a new name.

[Common] Button

If you have not defined a link to a UE project (see above), each time you save a program or run another tool that is dependent on stzUE project settings, you are prompted to select the project. You can opt to always use a selected project to bypass being prompted each time, but if you want to reset such selections, click the [Common] button.

2 / stzUEsave

This tool checks the syntax of the current source file in the editor, and optionally compiles it, backs it up, and checks it for user-defined warnings

“Select the program you want to save…”
If the “Check Subfolder” project option is selected and the program being compiled is found in multiple subfolders under the main save folder, you are prompted to choose which program to save.

Save Program Checkbox
stzUEsave displays a window with the results of its syntax check and user-defined warnings/replacements processing. If the project “Save Program” setting is turned on, you will also see the path and name of the program that will be saved when you click [Close]. If you do not want to save the program, turn off this option.

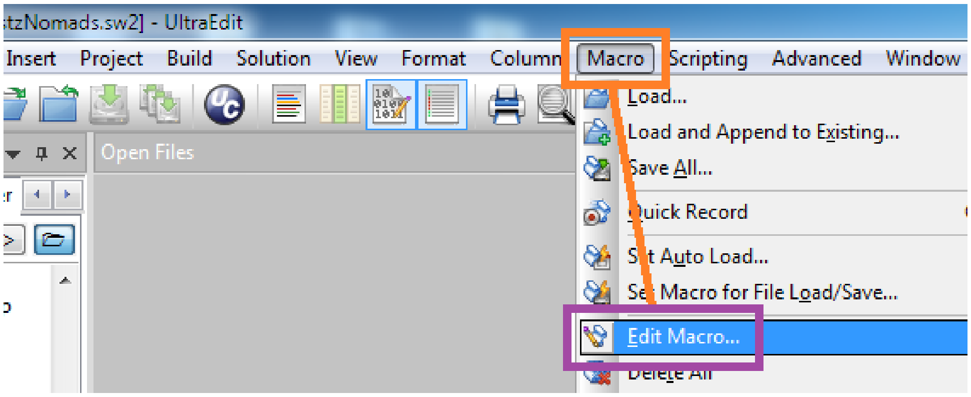
Save Program Path
Adjacent to the Save Program checkbox is the full path and name of the program that will be saved when you exit stzUEsave. You can type in a new path and/or name, or click the [Lookup Save Folder] button to select a save path, to the end of which the current program name will be appended for you.

Main Listbox / Syntax Errors Found
If there are syntax errors in the source file, they are displayed in the main display summary listbox at the top of the stzUEsave window. Double-click on a syntax error in that listbox to jump to the error on the [Listing] panel.

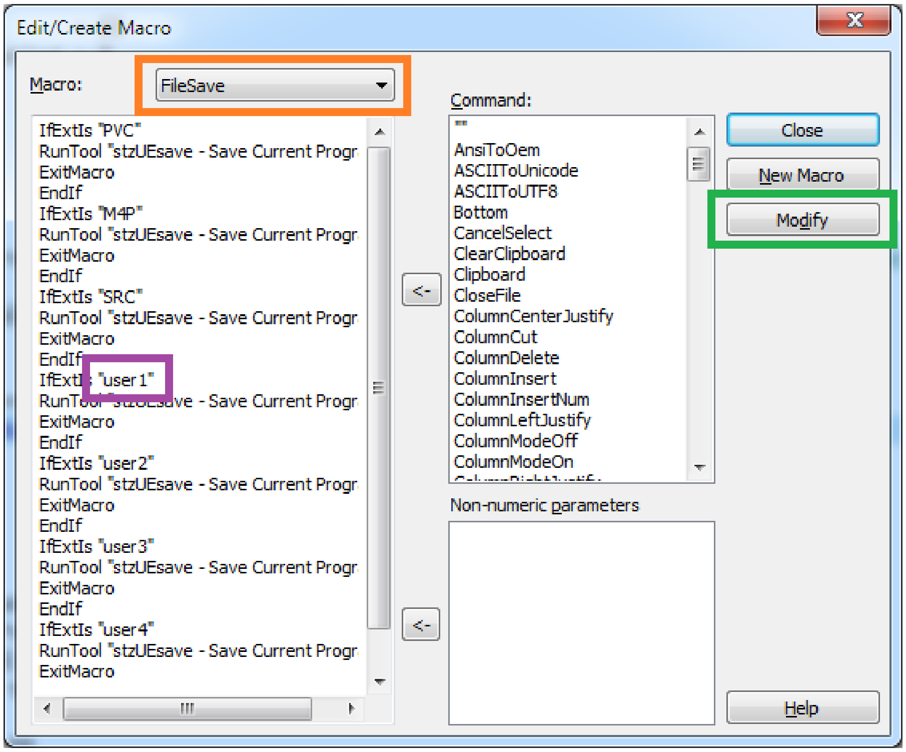
Save Macro

stzUE includes a macro file named stzUE.mac that includes a macro which can run stzUEsave when you save the file in UltraEdit. To use this macro, the file extensions in the macro may need to be modified, and it should be assigned to use the default UE save-file CTRL+S key combination, as described below.

Select [Macro] – [Edit Macro…]



Select the “FileSave” macro…



```
IfExtIs "PVC"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "M4P"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "SRC"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "user1"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "user2"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "user3"
RunTool "stzUEsave - Save Current Progr
ExitMacro
EndIf
IfExtIs "user4"
RunTool "stzUEsave - Save Current Progr
ExitMacro
```

```
""
AnsiToOem
ASCIIToUnicode
ASCIIToUTF8
Bottom
CancelSelect
ClearClipboard
Clipboard
CloseFile
ColumnCenterJustify
ColumnCut
ColumnDelete
ColumnInsert
ColumnInsertNum
ColumnLeftJustify
ColumnModeOff
ColumnModeOn
```

Note that the PVC, M4P and SRC extensions are already defined in the macro, but you may modify the “user1” value to match the extension of your source code programs, for example:

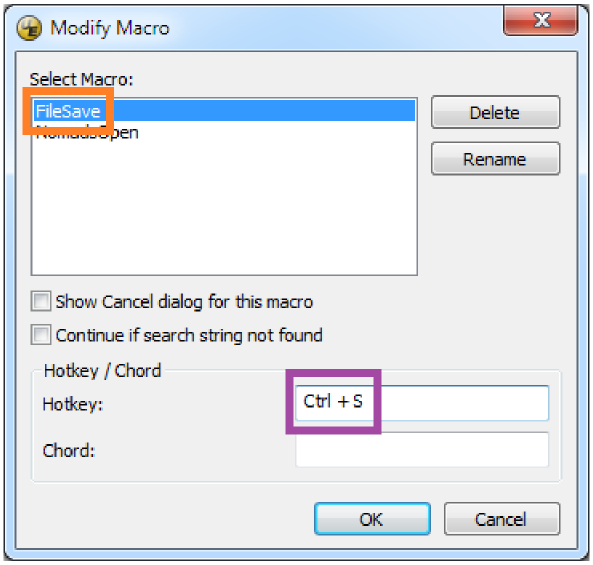
IfExtIs "**PVT**"

If you have additional extensions not already defined in the macro, there are placeholders where you can change “user2” through “user4”.

Note that your source programs must have an extension for this macro to work – source files without an extension require you to manually press the stzUEsave button.

After modifying the macro text – and make sure you do not change anything else or add blank lines or other commands to this macro – click the [Modify] button.

Select the FileSave macro, position the mouse pointer in the Hotkey field, then hold CTRL and press the “S” key. After doing so, “CTRL + S” will appear in the hotkey field…



Modify Macro

Select Macro:

FileSave
NomadsOpen

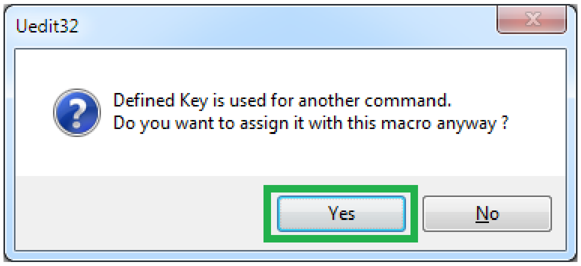
□ Show Cancel dialog for this macro
□ Continue if search string not found

Hotkey / Chord

Hotkey: Ctrl + S

Chord:

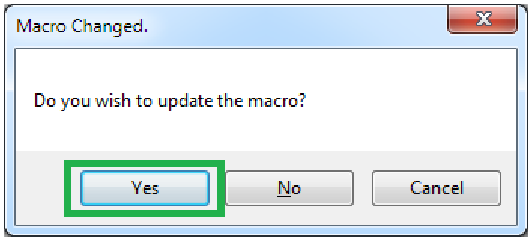
Click [OK] and when prompted to confirm the hotkey assignment, click [Yes]…



Uedit32

Defined Key is used for another command.
Do you want to assign it with this macro anyway ?

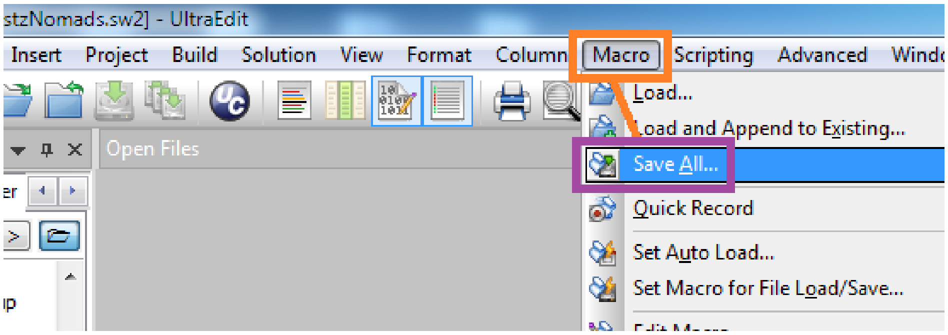
Click [Close] and select [Yes] when prompted to update the macro…



Macro Changed.

Do you wish to update the macro?

Finally, select [Macro] – [Save All…] and save over the stzUE.mac file in the stzUE\install file.



3 / stzUESearch

Searches programs and text files using complex search criteria.

4 / stzUEInsert

Displays data from a Pvx data dictionary, a message library, or a Pvx class and allows selected items to be inserted into the current source file opened in the UE editor.

5 / stzUEInvoke

Invokes the Pvx command-line, the Nomads designer, data dictionary maintenance, or PxPlus online help. Also allows user-defined actions to be added to the invoke-tasks window.

Project

The project selection only applies to the Nomads, Pvx Command-Line, and Data Dictionary Maintenance options, since these processes must be invoked with the appropriate interpreter.

[Settings]

You can remove standard buttons from the window, or add up to four of your own. You also have the option to keep the invoke window active after choosing a task, since its normal behavior is to close after a task button is pressed.

As mentioned above, UDBs can also be defined by project. This allows you to create, for example, a UDB that accesses TortoiseSVN for a specific folder by project.

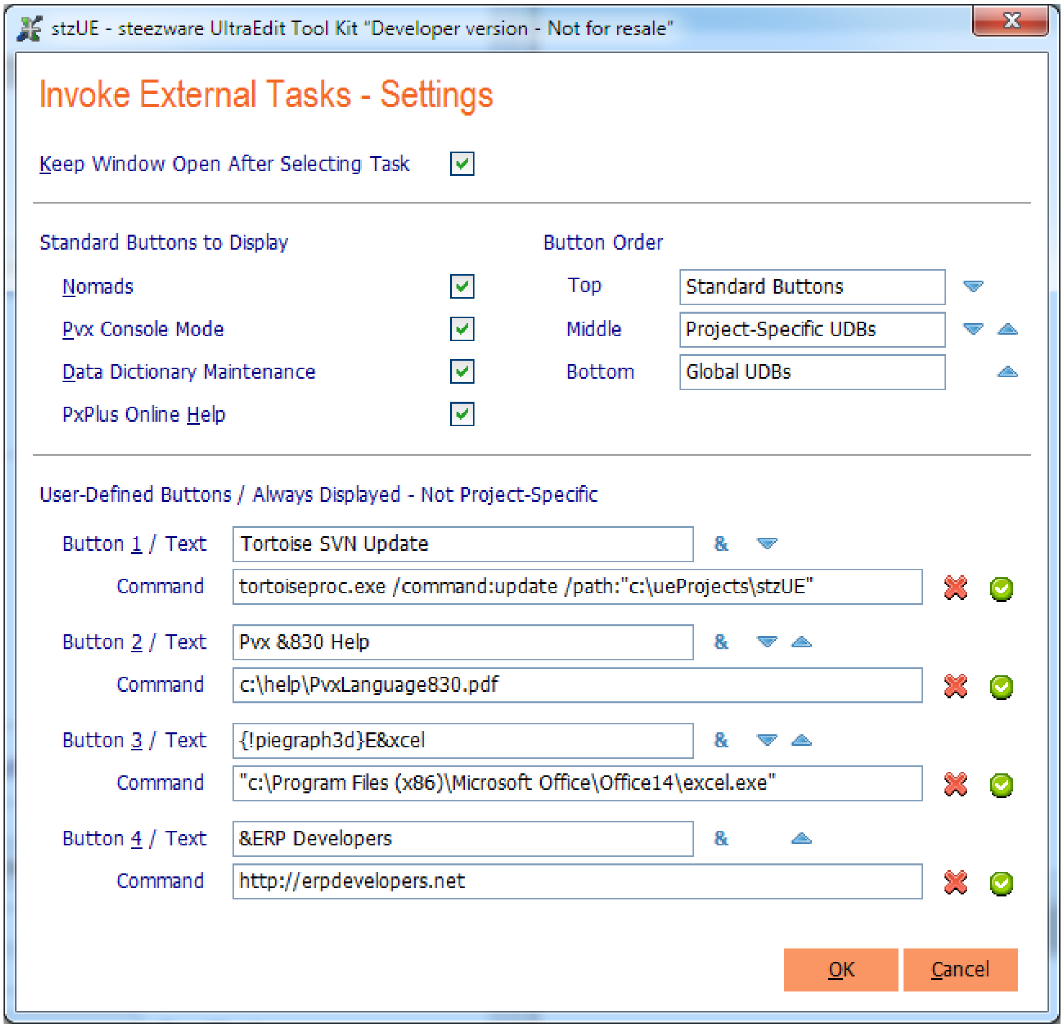
If no user-defined buttons are defined, the standard buttons appear down the middle of the invoke window. If you have a combination of standard buttons, global UDBs and project-specific UDBs, they will be shown in side-by-side columns, with the window automatically resized as necessary.

Notes on setting up your own buttons:

In the “Text” field enter the text to display on the button. To include a hotkey, precede the hotkey letter with an ampersand, or use the [&] button to toggle through the hotkey possibilities, since it is cognizant of other control hotkeys and will skip over them. Since the text format follows Nomads standards, you can even include an image as shown in Button 3 below by enclosing it
in { } curly braces.

If the command line has spaces in it – as in the fully-qualified path to Excel.exe as shown below for Button 3 – you may need to include it in quotes.

To experiment with your commands, click the [Test] – green checkmark – button.



stzUE - steezware UltraEdit Tool Kit "Developer version - Not for resale"

Invoke External Tasks - Settings

Keep Window Open After Selecting Task ☑

Standard Buttons to Display

- Nomads ☑
- Pvx Console Mode ☑
- Data Dictionary Maintenance ☑
- PxPlus Online Help ☑

Button Order

Top	Standard Buttons
Middle	Project-Specific UDBs
Bottom	Global UDBs

User-Defined Buttons / Always Displayed - Not Project-Specific

Button 1 / Text	Tortoise SVN Update
Command	tortoiseproc.exe /command:update /path:"c:\ueProjects\stzUE"
Button 2 / Text	Pvx &830 Help
Command	c:\help\PvxLanguage830.pdf
Button 3 / Text	{!piegraph3d}E&xcel
Command	"c:\Program Files (x86)\Microsoft Office\Office14\excel.exe"
Button 4 / Text	&ERP Developers
Command	http://erpdevelopers.net

6 / stzUEpgmtxt

Converts a program file opened in the editor to a text file, optionally formatting it and removing its line numbers.

7 / stzUEtrace

Reformats a Pvx trace output file for easier display in the UE editor, allowing for code folding and color-coding of syntax.

pgLVL>
pgLVL<
pgLVL-

Each program level in the trace is indicated with pgLVL> when a new program is executed, pgLVL< when a program exits to a lower level, and pgLVL- if the program remains at the same level. These indicators allow for code-folding sections of the trace, but it is possible and frequent for the trace fragment to have another closing code fold point without a matching opening fold point, which makes folding based on pgLVL> and pgLVL< impossible, so use code-folding with caution.

8 / Nomads Panel Selector

When a Nomads library file is opened in UltraEdit, a macro automatically invokes a stzUE program that allows you to select a panel from the library to modify. The interpreter and version of Nomads that are run are based on the stzUE project settings.

➡ If the stzUE NomadsOpen macro is missing or turned off, opening a Nomads library in UltraEdit will display its contents in hex format. **Do not edit a Nomads library or other Pvx data file opened in the UE editor or you could seriously damage the file.**

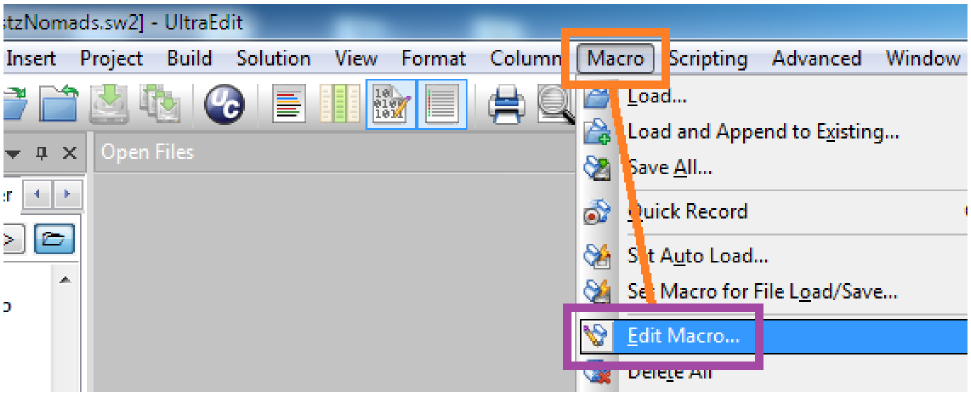
➡ When opening a library file, the macro attempts to automatically close it before running stzUEnomads. There is a known issue where a library file opened from "Recent Files" may remain open in the editor. As noted above, **you must manually close the file without editing it or saving it.**

➡ **If a Nomads library is open in the UE editor and you are prompted that changes were made to the file, click [Cancel] then close the file without saving it.**

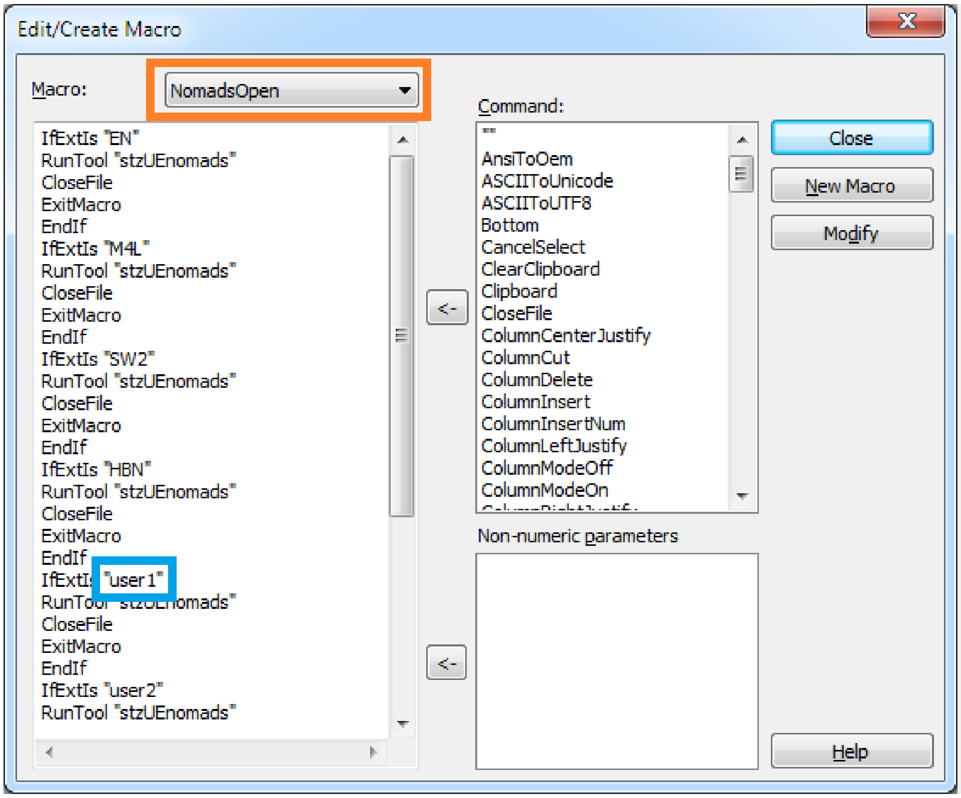
Nomads Macro

stzUE includes a macro file named stzUE.mac – which is found in the stzUE\install folder – that includes a macro that is run each time a file is opened. Its only purpose is to determine if the file being opened is a Nomads library to allow for the selection of a panel to edit in the Nomads designer. To access the stzUE Nomads panel selector, all you have to do is open a Nomads library in UltraEdit. The macro is defined to only run stzUEnomads if the file has a specific extension. The predefined extensions include EN and M4L, but if your application uses a different extension for Nomads libraries, you must edit the macro as described below.

Select [Macro] – [Edit Macro…]


tzNomads.sw2] - UltraEdit
Insert Project Build Solution View Format Column Macro Scripting Advanced Window
Open Files
Load...
Load and Append to Existing...
Save All...
Quick Record
Set Auto Load...
Set Macro for File Load/Save...
Edit Macro...
Delete All

Select the "NomadsOpen" macro…



```
IfExtIs "EN"
RunTool "stzUEnomads"
CloseFile
ExitMacro
EndIf
IfExtIs "M4L"
RunTool "stzUEnomads"
CloseFile
ExitMacro
EndIf
IfExtIs "SW2"
RunTool "stzUEnomads"
CloseFile
ExitMacro
EndIf
IfExtIs "HBN"
RunTool "stzUEnomads"
CloseFile
ExitMacro
EndIf
IfExtIs "user1"
RunTool "stzUEnomads"
CloseFile
ExitMacro
EndIf
IfExtIs "user2"
RunTool "stzUEnomads"
```

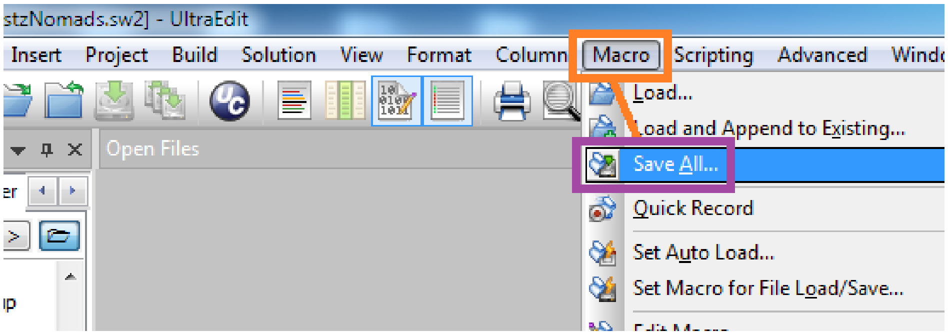
Change the “user1” value to be the extension of your Nomads library, for example:

IfExtIs "**LIB**"

If you have additional extensions not already defined in the macro, there are placeholders where you can change “user2” through “user4”.

After making the change – and make sure you do not change anything else or add blank lines or other commands to this macro – click [Close] and select [Yes] when prompted to update the macro.

Finally, select [Macro] – [Save All…] and save over the stzUE.mac file in the stzUE\install file.



Nomads Designer

To invoke the complete Nomads designer, use the stzUEinvoke tool.

Appendix A / Save Template Programs

This appendix contains information that is only applicable to users of stzUE that must save programs with an owner code or package flags. If you do not require an owner code or package flags on your compiled programs, you can skip this section.

The stzUEsave tool uses "template" programs when saving source files to program format. One template program is used for compiling programs with line numbers, and the other for programs that have no line numbers:

Line-Numbered Template	stzUEsave.cp1
Non-Line-Numbered Template	stzUEsave.cp2

stzUE also includes template programs that include the "90" owner code, for Sage 100 / MAS 90 developers:

Line-Numbered Template	stzUEsave-90.cp1
Non-Line-Numbered Template	stzUEsave-90.cp2

If you are a developer that assigns an owner code or a package flag to programs, you must manually create these two template programs in the c:\steezware\stzUE folder. Their names must begin with "stzUEsave" followed by a dash, then the numeric owner code and terminated with the .cp1 and .cp2 extensions. For example, if your owner code is 1234 you must create the following two template programs:

Line-Numbered Template	c:\steezware\stzUE\stzUEsave-1234.cp1
Non-Line-Numbered Template	c:\steezware\stzUE\stzUEsave-1234.cp2

If you assign a package flag or flags to compiled programs but do not have an owner code, you must create the two template files but can use any value you choose to name the files, for example, stzUEsave-999.cp1 and stzUEsave-999.cp2 (and enter "999" for the owner code in project maintenance as explained below).

Creating the Template Programs / Developers-Only

In order to save a program with an owner code, you must have a "developer" key. For example, Sage holds a developer key for creating programs that have the "90" owner code, but this key is not available for use by anyone outside of Sage.

If you have a developer key, creating the template programs can be done as follows:

1. Run the appropriate version of Pvx – that is, one that has an activation key that acknowledges the owner code - to open a new console mode
2. Create the line-numbered template program…
   2.1. Enter a single line remark:
   0010 ! stzUEsave.cp1 - Line-Numbered Template Save Program
   2.2. Type the following to save the program (note that FLG is optional and only used if you specify package flags):
   SAVE "c:\steezware\stzUE\stzUEsave-[owner-code].cp1", OWN=[owner-code] [, FLG=flg:flg:flg]
   - If you are saving a program that has package flags but not an owner code:
     SAVE "c:\steezware\stzUE\stzUEsave-[owner-code].cp1", FLG=flg:flg:flg
3. Creating the non-line-numbered template requires you to first create a serial file. The following steps may be done in the same console mode session started above…
   3.1. P$ = "c:\steezware\stzUE\stzUEsave-[owner-code].**cp2**"
   3.2. SERIAL P$
   3.3. OPEN (1) P$
   3.4. PRINT (1) "! stzUEsave.cp2 – Non-Line-Numbered Template Save Program
   3.5. CLOSE (1)

3.6. LOAD P$
3.7. ERASE P$
3.8. SAVE P$, OWN=[owner-code] [, FLG=flg:flg:flg]

- If you are saving a program that has package flags but not an owner code:
  SAVE P$, FLG=flg:flg:flg

CREATING THE TEMPLATE PROGRAMS / NON-DEVELOPERS

If you do not have a developer key but need template programs that contain a developer key, you can create the template programs by copying an existing owner-keyed program.

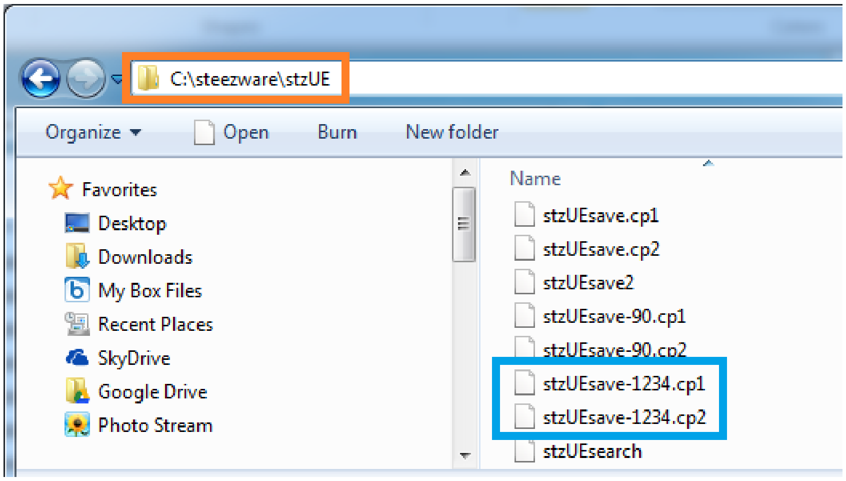
To create the line-numbered template program…

1. Run the appropriate version of Pvx – that is, one that has an activation key that acknowledges the owner code - to open a new console mode
2. LOAD a **line-numbered** program that has the developer key
3. LIST the first part of the program
4. DELETE all but the first remark in the program, that is, only have one line in the program and that line must be a remark – the contents of the remark are not important
5. SAVE "c:\steezware\stzUE\stzUEsave-[owner-code].cp1", OWN=[owner-code] [, FLG=flg:flg:flg]

Create the non-line-numbered template program in the same manner, except load a program that does not use line numbers…

1. LOAD a **non-line-numbered** program that has the developer key
2. LIST the first part of the program
3. DELETE all but the first remark in the program – the contents of the remark do not matter
4. SAVE "c:\steezware\stzUE\stzUEsave-[owner-code].cp2", OWN=[owner-code] [, FLG=flg:flg:flg]

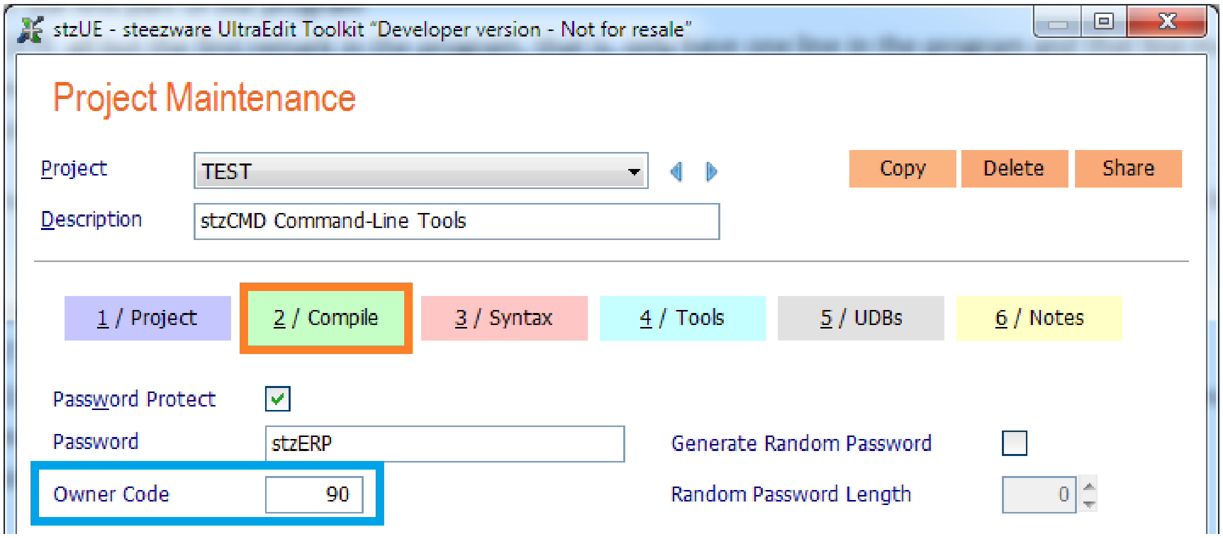
After successfully creating the two template programs, you should find them in the c:\steezware\stzUE folder:



stzUEsave.cp1
stzUEsave.cp2
stzUEsave2
stzUEsave-90.cp1
stzUEsave-90.cp2
stzUEsave-1234.cp1
stzUEsave-1234.cp2
stzUEsearch

SETTING THE OWNER CODE IN PROJECT MAINTENANCE

The owner code is defined on the [2 / Compile] panel in Project Maintenance:



stzUE - steezware UltraEdit Toolkit "Developer version - Not for resale"

Project Maintenance

Project: TEST

Description: stzCMD Command-Line Tools

Password Protect ☑

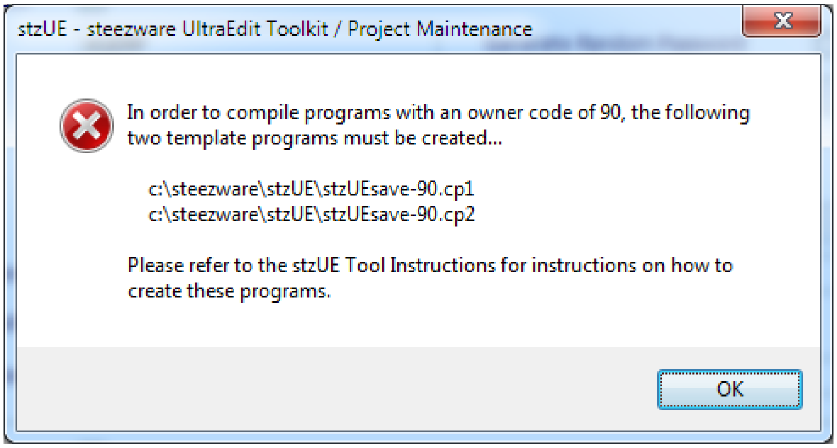
Password: stzERP

Generate Random Password □

Owner Code: 90

Random Password Length: 0

When you enter the owner code, a check is made to ensure that both template programs exist. If either or both templates are not found, the following message is displayed:



stzUE - steezware UltraEdit Toolkit / Project Maintenance

In order to compile programs with an owner code of 90, the following two template programs must be created...

c:\steezware\stzUE\stzUEsave-90.cp1
c:\steezware\stzUE\stzUEsave-90.cp2

Please refer to the stzUE Tool Instructions for instructions on how to create these programs.

Testing the Owner Code

1. Create a new source program or load an existing source program in the UE editor
2. Make sure your UE project is linked to the stzUE project that specifies the owner code, or select the appropriate project when saving the program
3. Click the stzUEsave button to save the program
4. Run the appropriate version of Pvx
5. If you have the steezware Command-Line Tools installed...
   - 5.1. >LOAD [program-name]
   - 5.2. >SP
     - The owner code will be shown under the "Info" tree node, if it exists
6. Or...
   - 6.1. >OPEN (1,ISZ=90) [program-name]
   - 6.2. >PRINT DEC(00+MID(RCD(1),83,2))
     - The owner code value should be displayed

Appendix B / Save Logic Override

stzUEsave includes a hook to a called program named “stzUEsaveovr.src”. This program allows you to override the normal save logic to assign a different save path, a different save program name, or to bypass saving the program altogether.

stzUEsaveovr.src is found in the c:\steezware\stzUE\install folder. It is provided as a formatted source code text file, with clearly defined code snippets and remarks to assist you in modifying the program. To modify and use this program, do the following:

1. Open c:\steezware\stzUE\install\stzUEsaveovr.src in UltraEdit.
2. To activate the program, you must remove the EXIT statement near the top of the program, or remark it out.
3. The full path and name of the source file is passed in the srcFILE$ variable and the path/name of the compiled program is passed as saveFILE$. The paths and names of the source and compiled files are parsed out for you.
4. You can set the return variable noSAVE to non-zero to instruct stzUEsave to not compile and save the program, or you can change the path and/or the name of the program to be saved.
5. After making your changes, save the file but **do not compile or move it** from the install folder.
    - Since there is a minimal amount of code executed in this program, it does not significantly hinder the save logic by leaving it as a text file.
    - stzUEsave expects to find this program in the “install” folder, so do not move or rename it.

To assist in debugging, feel free to add ESCAPEs to this program to break and step thru the code, but use caution when modifying its code since it could jeopardize the underlying compile/save logic if it introduces bad behavior or errors.

APPENDIX C / ULTRAEDIT TIPS

AUTOMATIC INDENTATION

When you begin a line with one of the following strings, the next line is automatically indented…	…and the following commands trigger the current line to automatically unindent…
{	}
FOR	NEXT
WHILE	WEND
REPEAT	UNTIL
SELECT	NEXT RECORD
SWITCH	END SWITCH
CASE	BREAK
DEFAULT	BREAK
DEF FN	END DEF

Note that IF, THEN, and ELSE are not automatically indented. Best practice for defining IF statements is to use { } curly brackets, which provide automatic indentation and allow for code-folding. Example:

```
IF condition {
    process this true-condition code
} ELSE {
   process this other code
}
```

CHANGING CASE

You can select a word or phrase and change its case with the following hotkeys:

- [F5] Capitalize
- [ALT+F5] Uppercase
- [CTRL+F5] Lowercase
- [Shift+F5] Invert Case

COLORS

To set colors not related to syntax highlighting – for example, how the current line is highlighted – select the [View] - [Set Colors…] menu option.

COLUMN MODE

Use [ALT+C] to switch between column mode and standard mode. Column mode may be used to select columnar sections of the source code.

COMPARING PROGRAMS

Select [FILE] - [COMPARE] to compare source code files.

CTRL-E – ERASE CURRENT LINE

Press [CTRL+E] to erase the current line in the editor.

If you want to change this to use [CTRL+D] - the Eclipse key combination for deleting a line - do the following:

1. Choose [Advanced] - [Configuration] from the UE menu bar
2. Find "Key Mapping"

3. Next to the "Commands:" caption, turn on the "Sort" checkbox
4. Scroll down to find **EditHexInsertDelete** in the list
   - This is the default UE option for the CTRL+D key combination, so...
5. Click [Remove] to release the key combination
6. Scroll up in the list to find **EditDeleteLine**
7. In the "Press New (multi-)key" field, click and hold the [CTRL] key and then press the letter "D" key
8. Click [Assign] to add CTRL+D to the "Existing Keys"
9. Select CTRL+E under "Existing Keys" and click [Remove]
   - The only assignment for "Existing Keys" should now be CTRL-D

ESCAPE

To set an escape into a program which automatically opens the debug/command-mode window, enter the following on a unique line in the source code:

```
PREINPUT -1301 ; ESCAPE
```

For this to open the command-mode debug window, you must have Debug=1 set in the [Config] section of your INI file:

```
[Config]
…
Debug=1
DebugPlus=1
```

The stzUE environment includes a template that will automatically insert this code for you. Simply type “ES” then CTRL+Spacebar and a small window will pop up that has “ESCAPE” with a small letter “T” icon beside it. Use your mouse or arrow key to select this template, and the code will be inserted for you.

Note that in this example, the Pvx system variable ESC is also displayed. This is because CTRL+Spacebar can be used with any introductory characters to open a list of suggestions – for example, enter just “E” (or “e”) then CTRL+Spacebar to see all directives and functions that begin with the letter “E”.

To add your own templates, go to [Advanced] – [Display/Modify Templates…].

FONTS

To set editor fonts, select the [View] - [Set Font…] menu option. Note that it is best to not use a variable-width font for editing, since the tab/indentation spacing may differ when viewed on another machine that uses a fixed-width font.

FUNCTION LIST

The UltraEdit “Function List” is used to display line labels in the current program. Its window can be docked or left free-floating – see “Moving the Function List” in Chapter 2.

1. Double-click on a label to jump directly to it.
2. Right-click on the list to sort the labels or to show labels in all project programs.

HIGHLIGHTING BRACKETED ITEMS

Use [CTRL+B] to search for and highlight bracketed items. You may then continue to press [CTRL+B] to expand to brackets outside the current bracketed selection.

UltraEdit defines bracketed items as those which are enclosed in parentheses ( ), brackets [ ], or braces { }.

Viewing Line Numbers

To view the physical line numbers of the source code, select the [View] - [Display Line Numbers] menu option.

Viewing Spaces and Tabs

To view spaces, tabs and paragraph marks, access the [View] - [Show Spaces/Tabs] menu option.

Appendix D / UltraEdit Projects

UltraEdit projects provide an easy way to manage and group associated files. When you open a project, all the programs open when last editing the project are automatically reopened. You can also search across all programs in a project and display a list of all line labels in all project files (see “Function List” above). By double-clicking on one of the line labels in the list, UltraEdit automatically jumps to the program and positions the current line at the label.

A project is defined by an INI-type text file that has an extension of .PRJ. These files may be maintained in any desired location, for example, c:\steezware\stzUE\ueProjects.

Creating a New Project

Select the [Project] - [New Project/Workspace] menu option and specify a name for your project. It is suggested that you keep all projects defined in the same location – using the “ueProjects” folder mentioned above, for example.

If you currently have files opened for editing, you will be prompted to close them. If the project will include the opened files, click [No] and you will be able to add them easily to the new project.

If you left files opened that you want added to the project, choose [+All Open Files], which will add them to Project Files list.

To add programs to the project, choose [Add File…].

To remove programs from the project, select one or more files and choose [Remove Item].

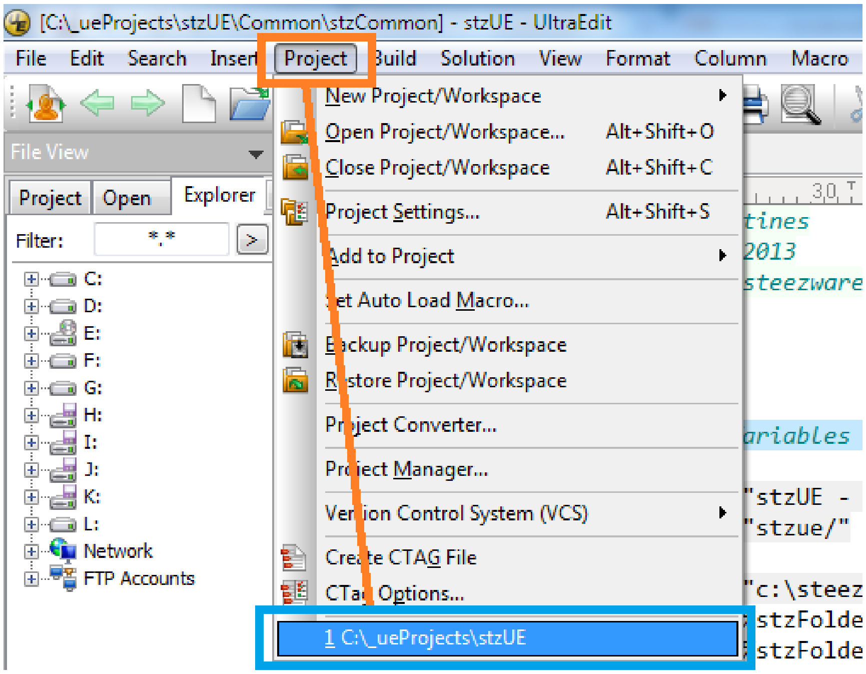
Leave “Project Directory”, “Optional Wordfile” and “Optional CTAG File” blank.

Click [Close] after all program files have been added to the project.

Opening a Project

Select the [Project] - [Open Project/Workspace] menu option. This merely opens a file explorer window from which you select a .PRJ project file.

A list of the recently used projects is also shown at the bottom of the [Project] menu, which allows you to quickly open a project:



Closing a Project

Select the [Project] - [Close Project/Workspace] menu option. When prompted to close the open files, it is recommended that you select [Yes].

Modifying a Project

The only maintenance a project requires is adding and removing programs in its file list.

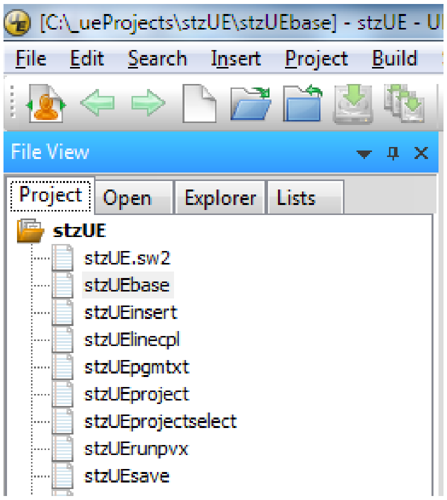
Select the [Project] - [Files/Settings] menu option. If you do not currently have a project open, this menu item is disabled.

Use the Add and Remove options to modify the file list.

If any project file is not currently opened for editing, select it from the file list and press Open.

Using a Project

The “File View” pane has a “Project” tab that displays the current project and a tree-view list of its files. If you do not see the File View pane, click CTRL-U to display it.



On the [Search] menu, the [Find in Files] option allows you to search all open files or all project files. When searching, a window opens showing the results of the search. You can double-click on a line in the results window and UltraEdit will jump to it.

Note that stzUEsearch only searches folders and cannot restrict a search to UE project files.