APL64 Developer Version User Tools

Contents

Summary 1

Configuring A User Tool 1

Example: User tool to create a named timestamp 2

User Tool Menu Text 4

User Tool Action Text 5

User Tool Keyboard Key and User Tool Modifier Key(s) 6

User Tool Documentation 6

Running a User Tool 6

From the Tools Menu 6

Using the Short Cut Keystrokes 8

User Tool Output Targets 9

Only Explicit User Tool Output is Considered 9

Example #1 9

Example #2 10

APL+Win User Tools Compatibility 12

Summary

Initially, the Tools menu has no list items. You can create your own selection of user tools to perform useful actions. A user tool includes a menu text, an action text which will be executed in the APL64 developer version when the tool is run, an optional keyboard short cut, and optional documentation text.

A user tool can execute a user command, system command, system function, user-defined function or any APL executable statement.

The user tool description appears in a list accessed by the Tools menu item in the APL64 developer version.

Clicking on a user tool description in the list, or using the optional keyboard short cut, runs the selected user tool.

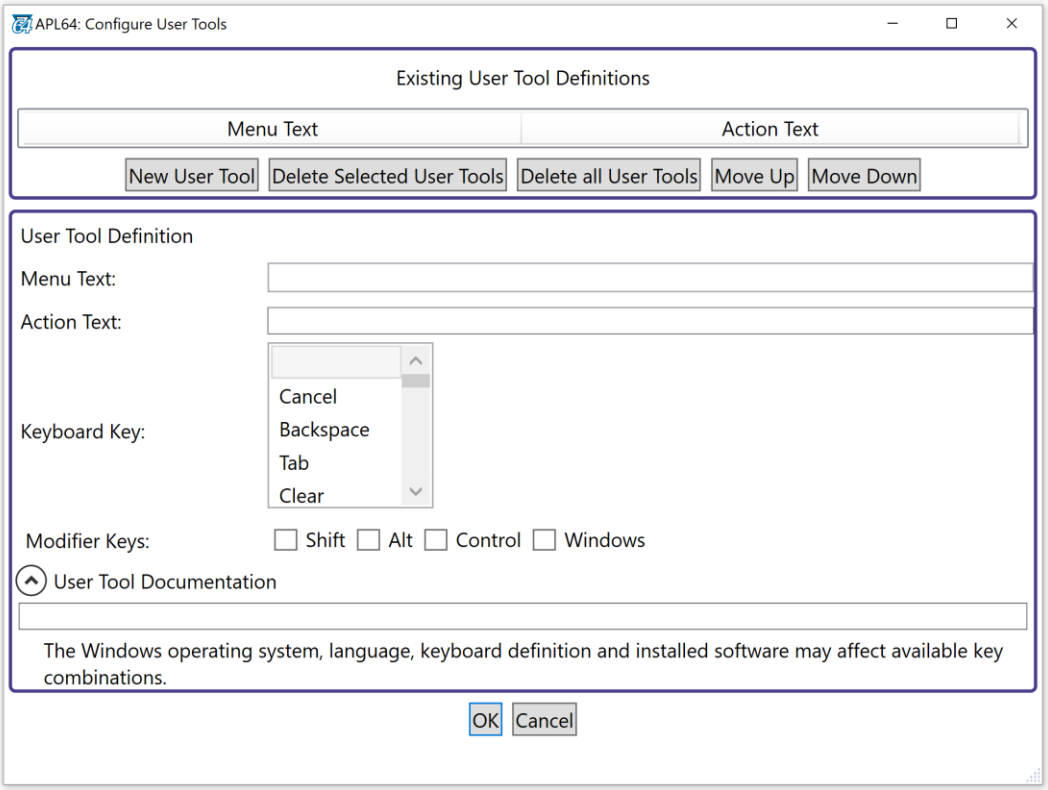
Output of a user tool, if any, is directed to the pane with the keyboard focus in the APL64 developer version.

User tool definitions are preserved in the APL64 developer version xml-format configuration file.

Configuring A User Tool

User tools may be defined and existing tool definitions edited using the Configure User Tools dialogue accessed from the menu **Options | Configure User Tools**. Tooltips for each user entry field summarize the

values required. Click the ‘New User Tool’ button to create a new user tool. The order of the menu items in the Tools menu may be modified using the ‘Move Up’ and ‘Move Down’ buttons.



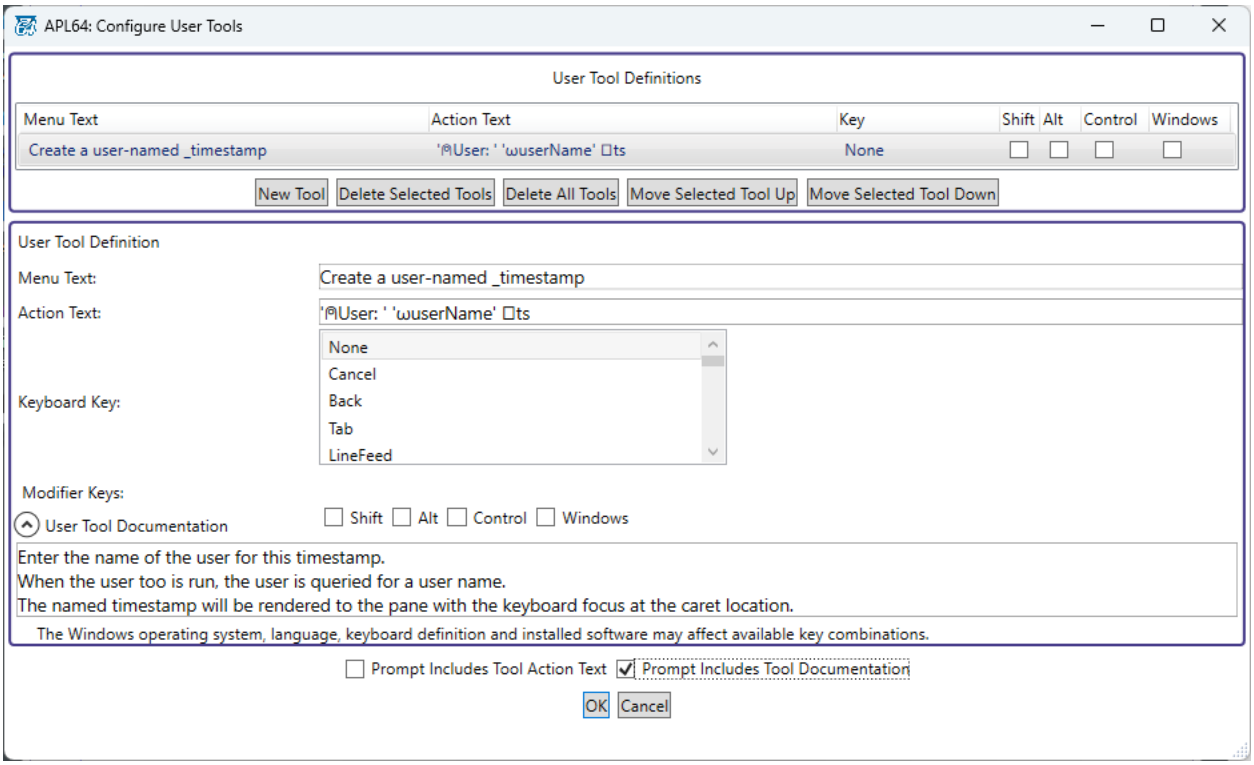
Example: User tool to create a named timestamp

In this example a named timestamp user tool has been created with:

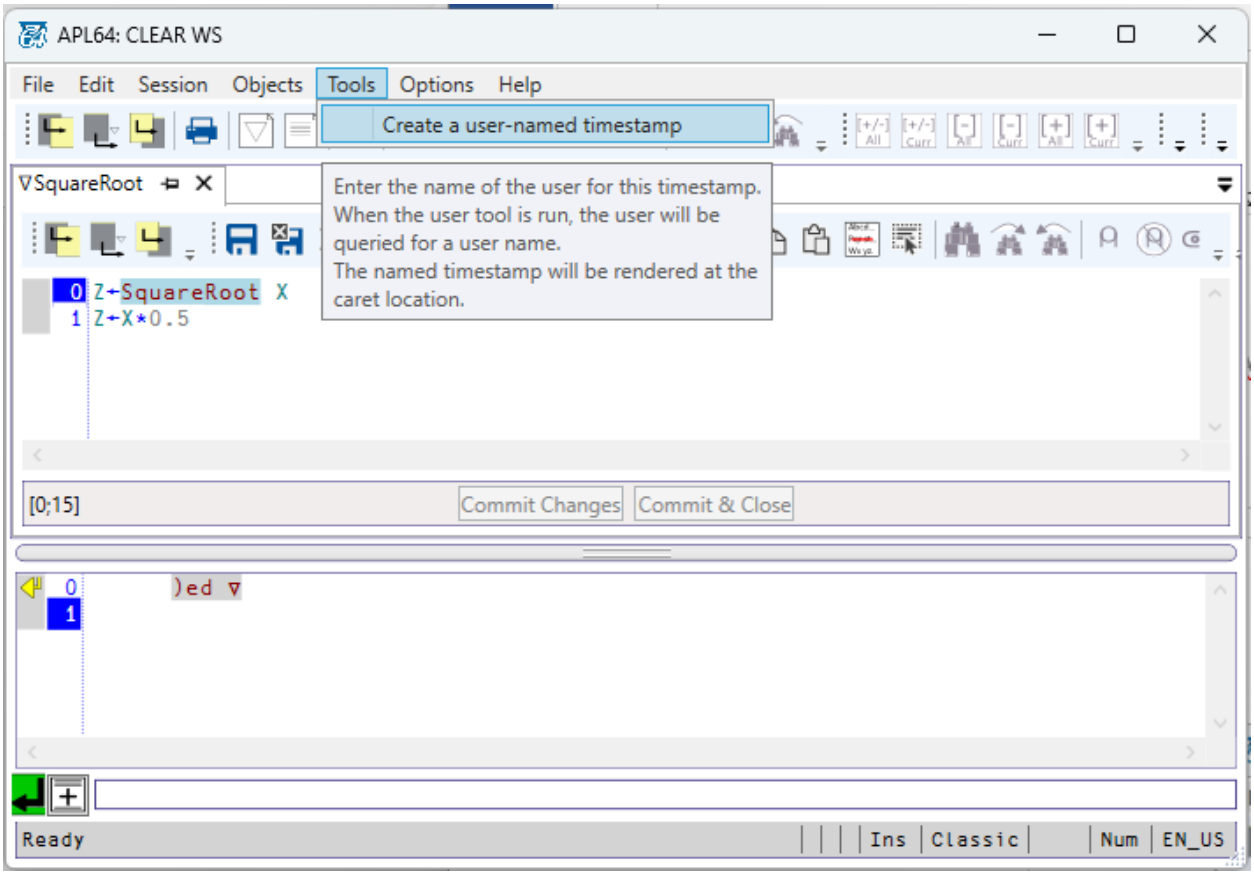
Menu text: Create a user-named timestamp, with one underscore prefix of a character in the menu text to support the Tools menu accessibility features.

Action text: The action text, User: 'ωuserName' ⍞ts , indicates that when this user tool is run, the user will be presented with an input window to enter the desired value of the username when the user tool text is inserted into at the caret position in the APL64 developer version pane with the keyboard focus.

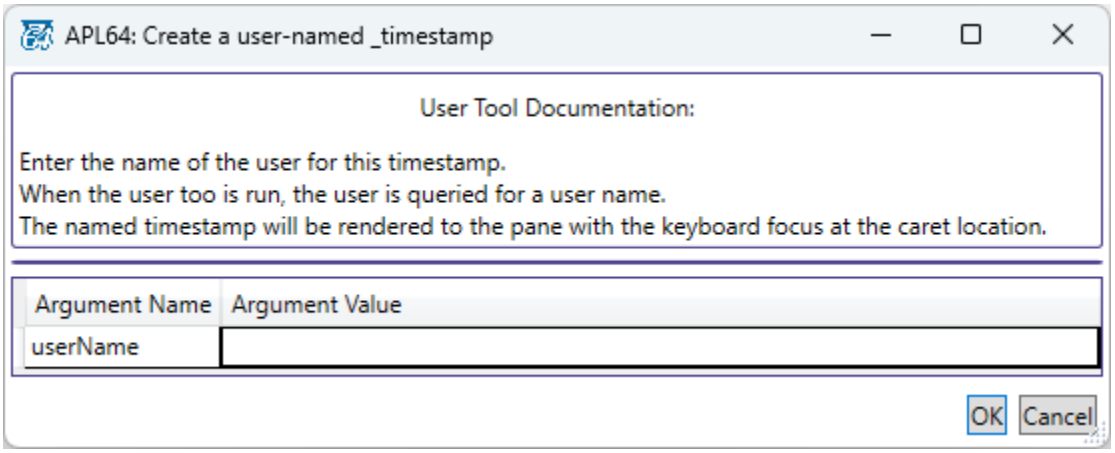
- Input field: ωuserName
- Keyboard short cut: not defined
- Documentation text: Enter the name of the user for this timestamp.
- Prompt will include the tool’s action text
- Prompt will display the documentation text



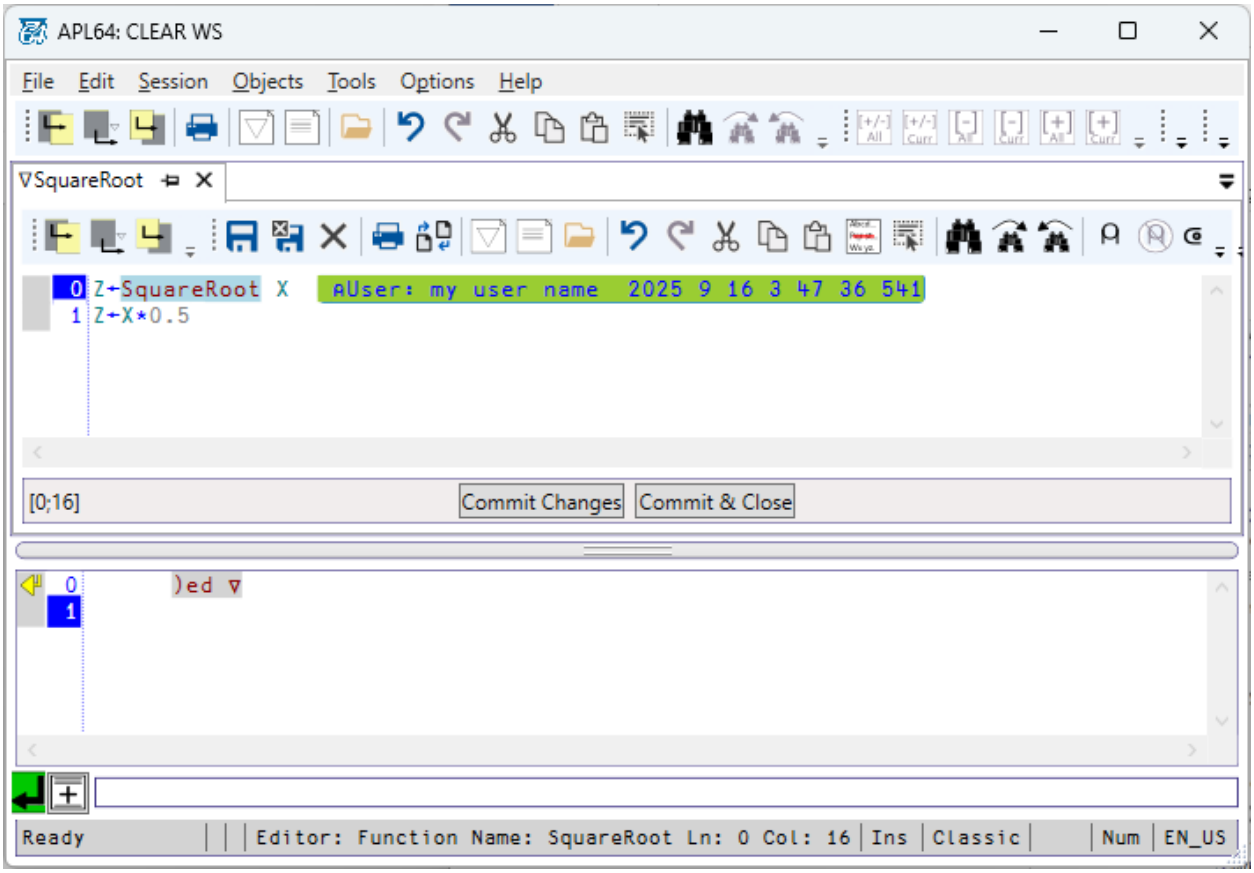
The user tool's menu text becomes a Tools menu item:



When this user tool is run, the user is prompted for the ωuserName input value:



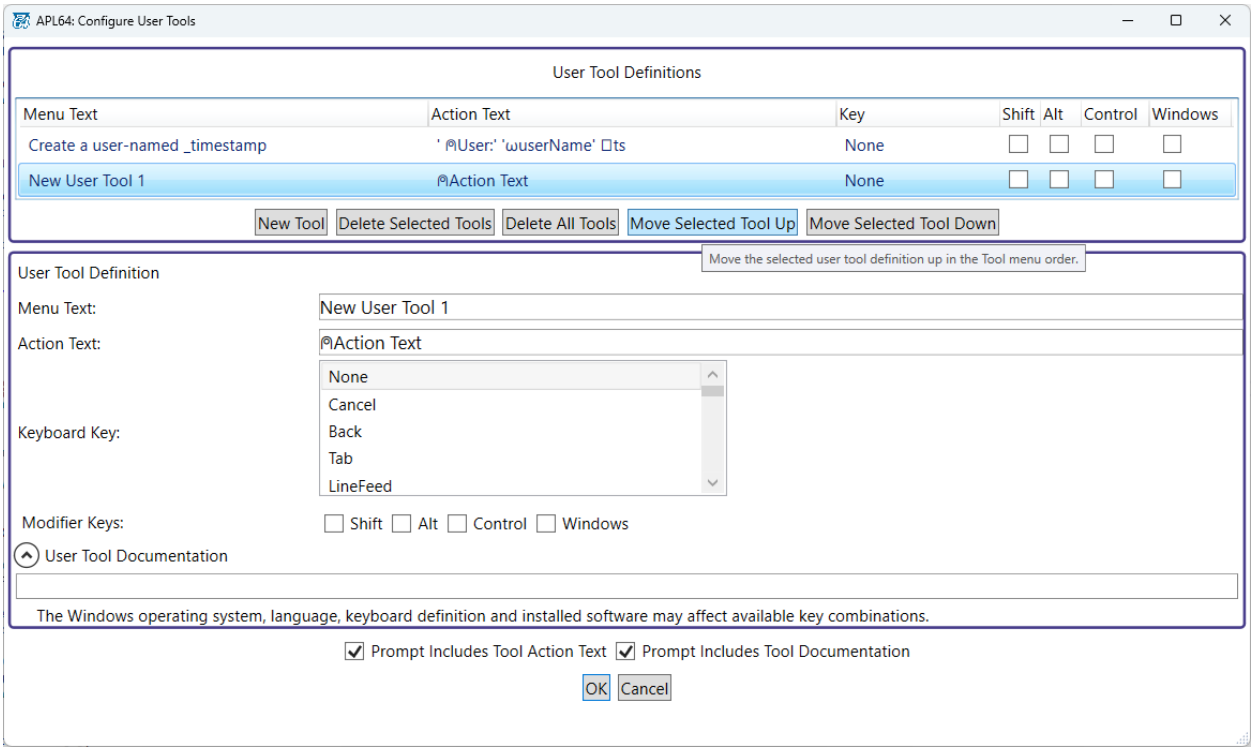
When the user OK's the input value, the user tool output is rendered to the pane with the keyboard focus:



User Tool Menu Text

The user tool menu text becomes a user tool item in the Tools list of the APL64 developer version. Prefix a character in the user tool menu text with the underscore (_) glyph to designate that character as a 'keyboard menu access character'.

The order of the menu items in the Tools menu may be modified by selecting a user tool in the Configure User Tools dialog and clicking the ‘Move Up’ and ‘Move Down’ buttons in that dialog.



User Tool Action Text

Enter an APL executable expression, which can be multi-line and contain APL glyphs.

Runtime-provided Arguments: If the user tool requires optional runtime arguments, include runtime argument names in the user tool action text. The position of the optional runtime arguments in the user tool action text is significant.

User tool runtime-provided arguments are values which will be user-entered at runtime by the APL64 programmer using the APL64 developer version. Runtime-provided arguments in the user tool action text are identified with the APL glyph omega (ω) prefix. In addition to the omega APL glyph, the user tool runtime argument text identifier may include descriptive text. Multiple runtime-provided user tool arguments are supported. For example:

- ωArgument1
- ωArgument2
- ω

Selection-provided Argument: A user tool argument which will be provided by selection available at runtime is identified by the APL glyph del (∇) prefix. At runtime the current selection will be obtained from the APL64 Developer session GUI control which has the keyboard focus. One selection-provided argument is supported, but the selection value may be used more than once in the user tool’s action text.

Examples:

- MYFN ωArgument1 ωArgument2 ∇

- ωArgument1 MYFN ωArgument2
- MYFN ∇ ωArgument1 ωArgument2
- MYFN ∇ ω ω
- ∇ can be … when ∇ is present

The user tool's action text, once filled with the optional runtime provided arguments and selection-provided argument, if any, must be successfully executed by the APL64 interpreter to be an effective user tool. If the execution of the user tool's action text fails at runtime, no exception will be thrown.

The Action Text text box in the Configure User Tools dialog supports multi-line text to improve the legibility of the action text while entering it into this dialog. When the Action Text is executed at runtime, multiple lines in the Action Text are analogous to using diamond statement separators, and do not render as multiple lines to the target pane of the APL64 developer version.

If it is desired that the Action Text generate multiple inserted lines at runtime, formatting characters, such as □tcnl, □tclf, «\r», «\n», would need to be included in the Action Text in the Configure User Tools dialog.

If a user tool action text will run an APL64 programmer-defined function, implicit output of that function is not captured and rendered to the target editor by the user tool mechanism in APL64.

User Tool Keyboard Key and User Tool Modifier Key(s)

Select the optional keyboard key and modifier keys which will be used in the short-cut for the user tool menu item from the list of available keyboard keys. The user tool keyboard short cut can override pre-existing keyboard short cuts in the APL64 developer version. The Windows operating system, language, keyboard definition and installed software may affect available key combinations.

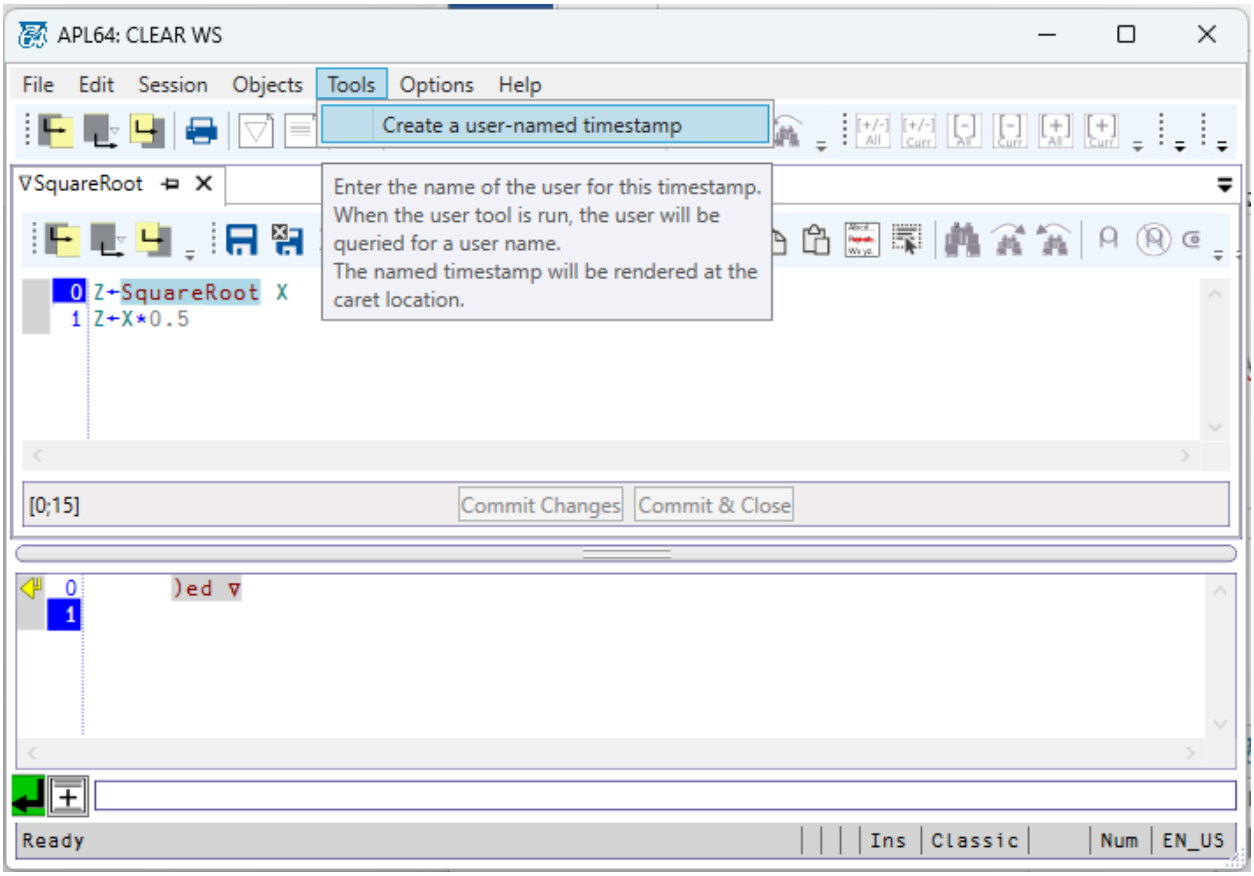
User Tool Documentation

Enter the (optional) documentation of this User Tool, which can be multi-line and contain APL glyphs. This documentation will be used as the User Tool menu item tooltip. This documentation will be displayed when User Tool argument values are requested at runtime, if the 'Prompt Includes Tool Documentation' check box of the 'Configure User Tools' dialog was checked in the user tool definition.

Running a User Tool

From the Tools Menu

The user tool menu text will be included in the APL64 Developer version Tools menu. The optional user tool documentation text will be used as the user tool menu item tooltip:

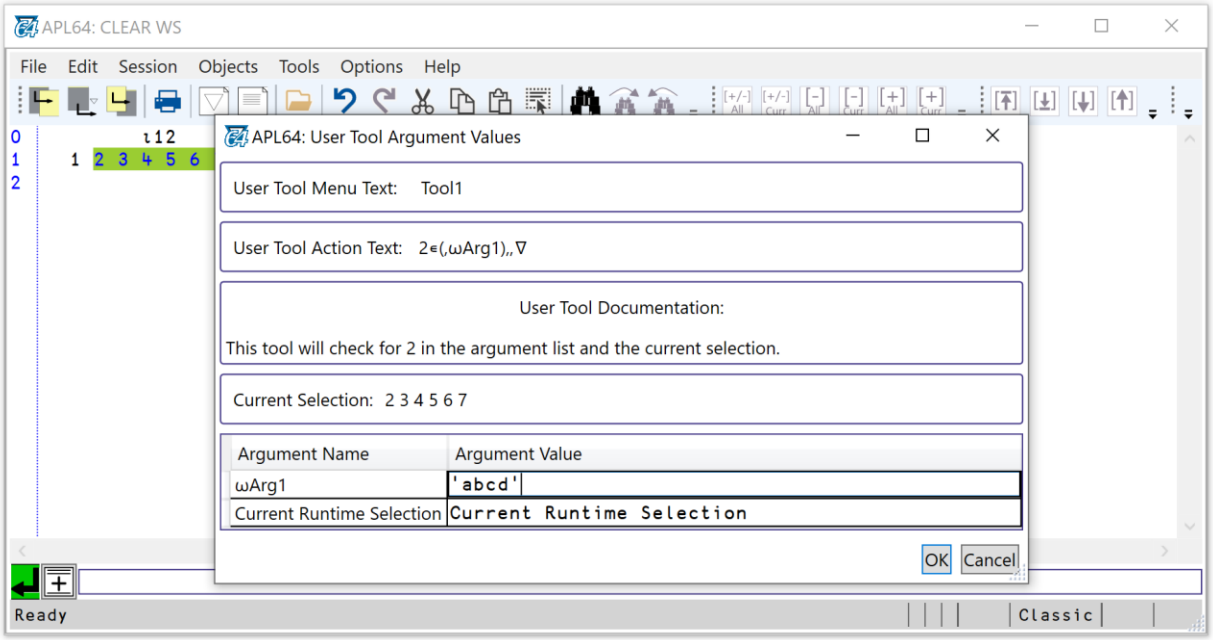


If the user tool requires the runtime selection, because the del (∇) glyphs was included in the user tool's action text, put the keyboard focus on the selected GUI control in the APL64 Developer version instance and make the desired selection within that GUI control.

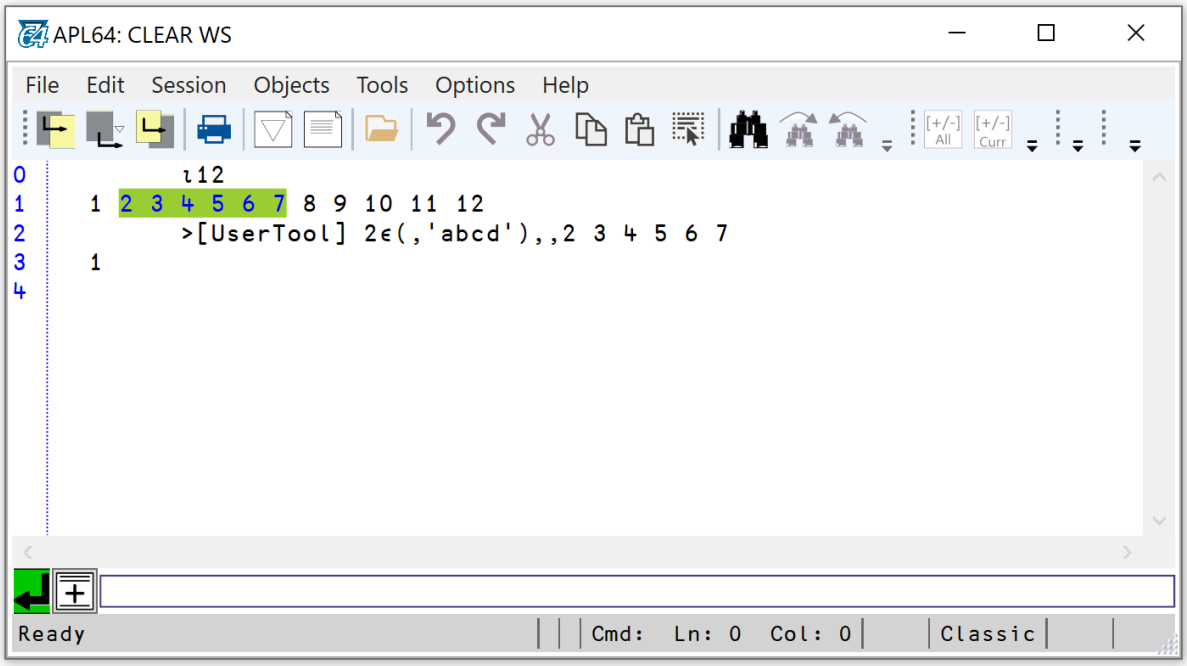
- If the selection is a continuous, linear selection which spans lines, the selection content will include the new line elements.
- If the selection is a rectangular selection, all selection segments in the rectangular selection will be concatenated into a linear selection.
- If the user tool requires the current selection and no selection has been made by the user, the implied selection will be the object name nearest the caret position in the GUI control with the keyboard focus. If there is no nearest object name, the implied selection will be ''.

Click on the selected user tool menu item to execute the user tool.

If the user tool requires user-provided runtime arguments, the user tool argument dialogue will be presented. In this dialogue enter the required runtime argument values. The optional user tool documentation may be presented in this dialogue. If the user tool requires the runtime selection, the selection content will also be presented in this dialogue.



Click the 'OK' button when the required argument values have been provided and the user tool will be executed:



Using the Short Cut Keystrokes

The user tool may be invoked from within the APL64 Developer version instance using the short cut keystrokes in the user tool definition.

To direct the output, if any, of a user tool to a function or variable editor or the editable classic history pane, set the focus on the desired target and click the user tool's short cut keystroke.

User Tool Output Targets

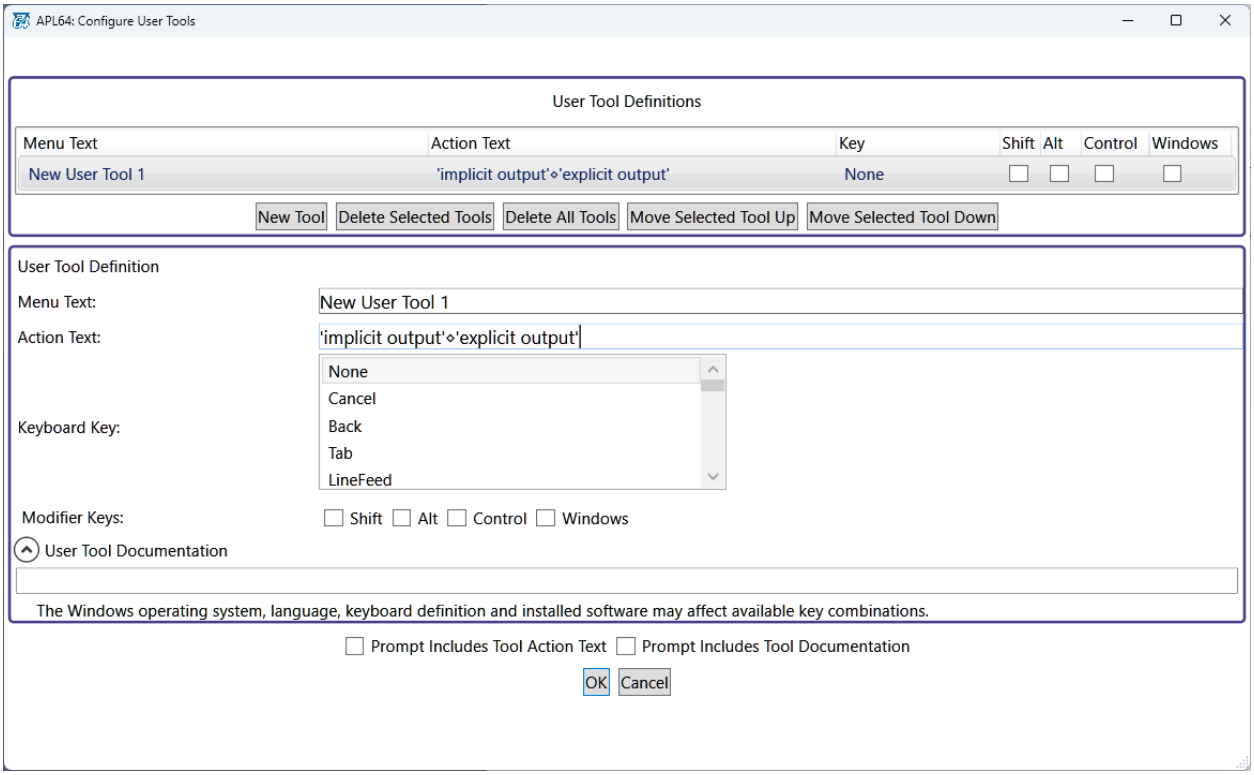
By selecting the keyboard focus, the user tool's output, if any, can be directed to the editable classic history pane, the command line, a function editor or a pure text element in a variable in the variable editor.

Only Explicit User Tool Output is Considered

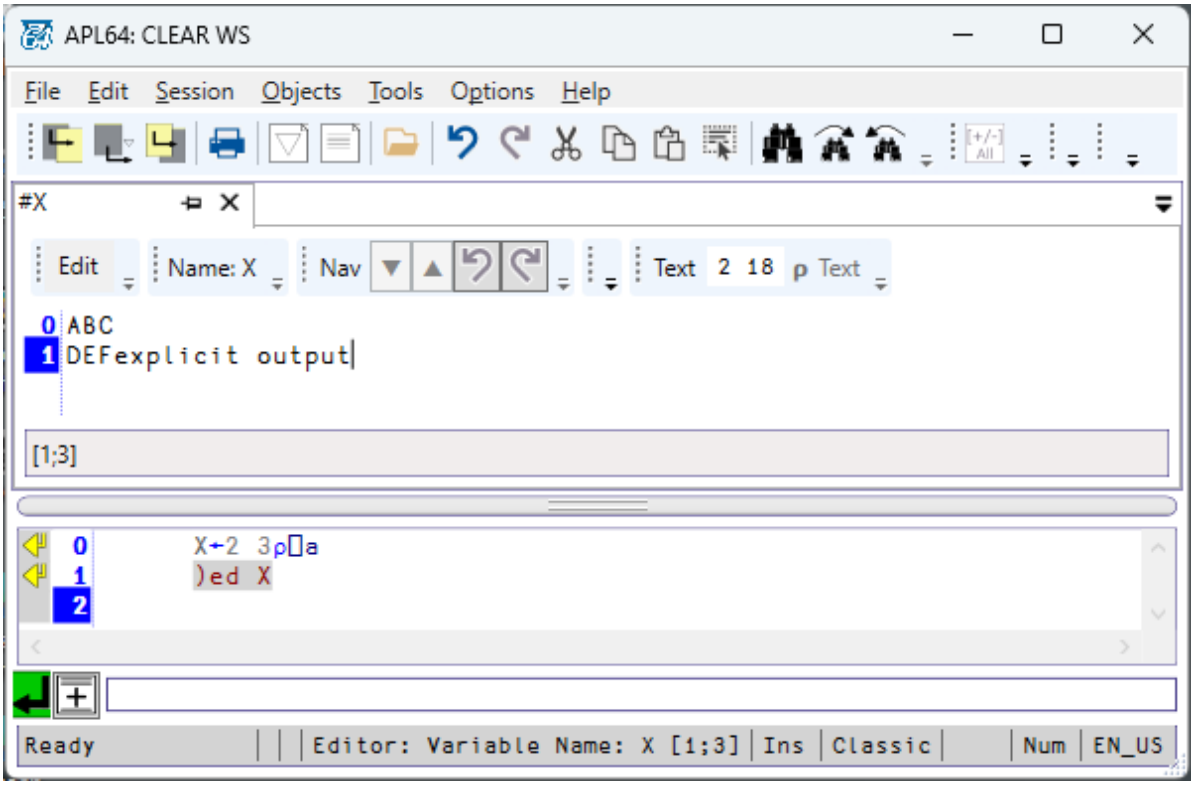
In APL64, running a user tool renders explicit output, if any, of the execution of the user tool's action text to the pane with keyboard focus. Implicit output, if any, of the execution of the user tool's action text is not considered.

Example #1

In this example the user tool action text is diamondized text:



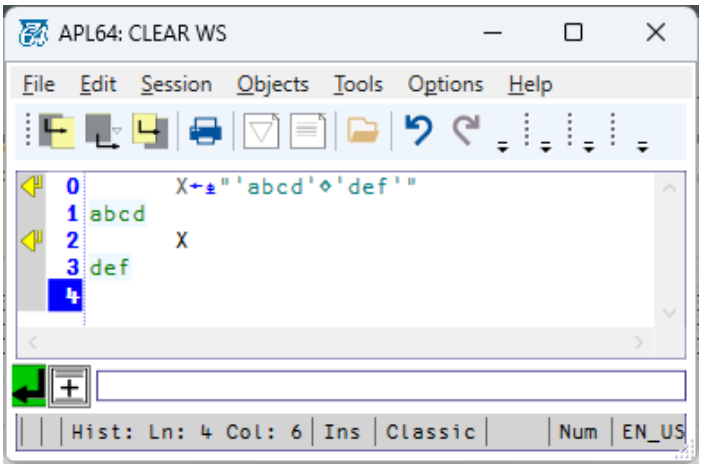
In APL64 the explicit output of this user tool is rendered to the APL64 variable editor, which has keyboard focus, at the caret position, and the implicit output is not considered, and the updated caret position is immediately after the insertion of the 'explicit output' text to the variable editor.



To understand this example, try this:

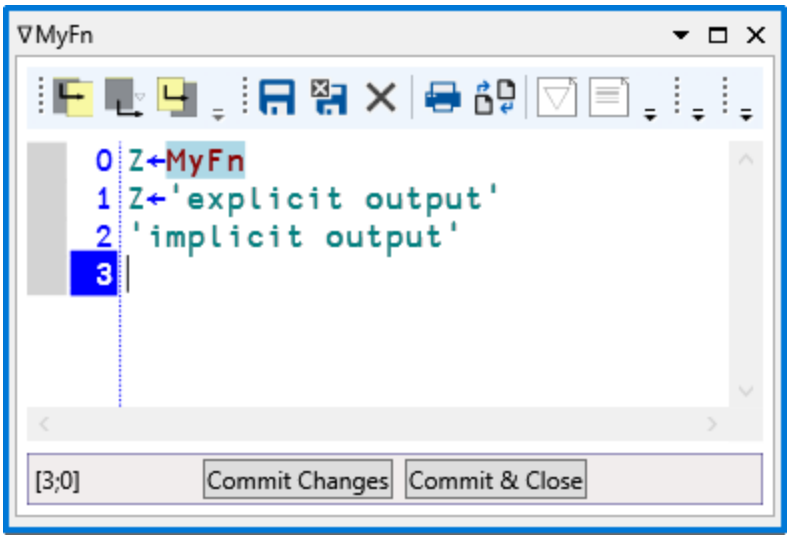
```
X← ⍎ "'abcd'⋄'def'"
```

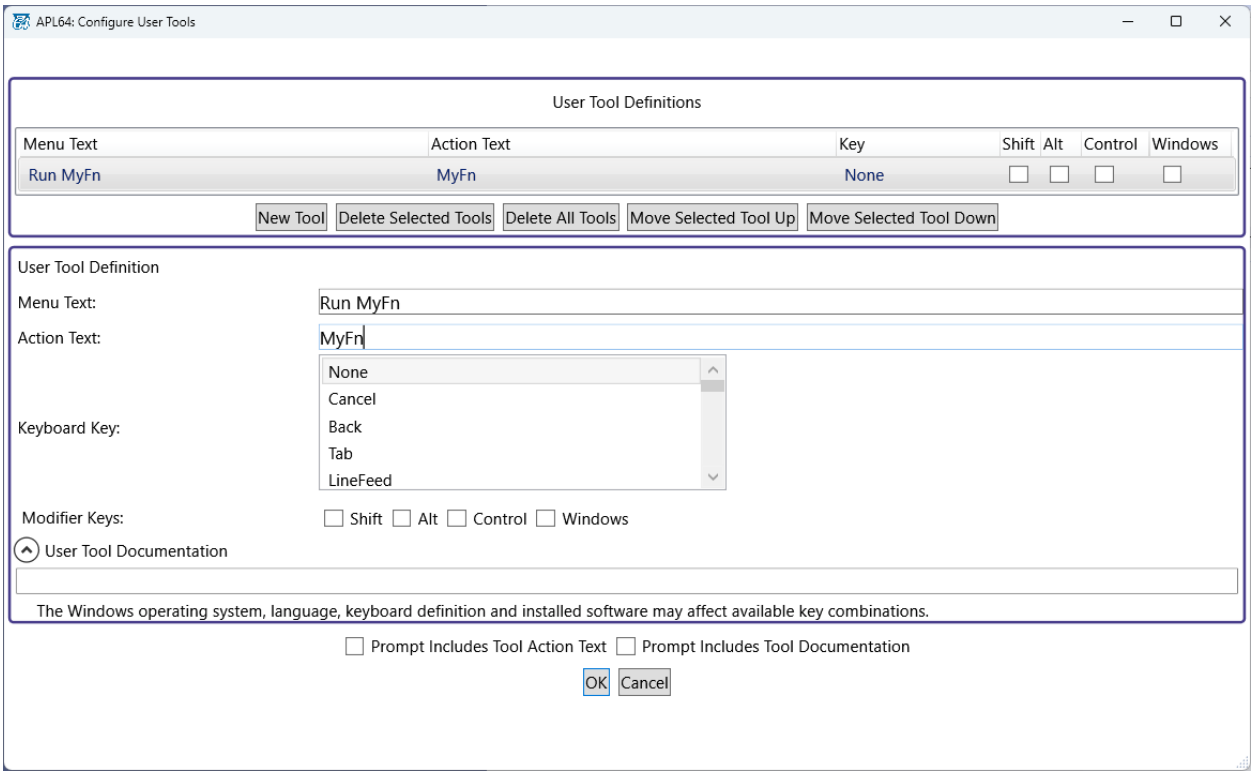
In APL64 the interpreter considers only the last diamondized statement to be explicit output. Explicit output is that output which can be assigned. Executing this text is analogous to the running of a user tool. The interpreter executes the user tool's action text, and the explicit output, if any, is rendered at the caret location in the APL64 developer version pane which has keyboard focus.



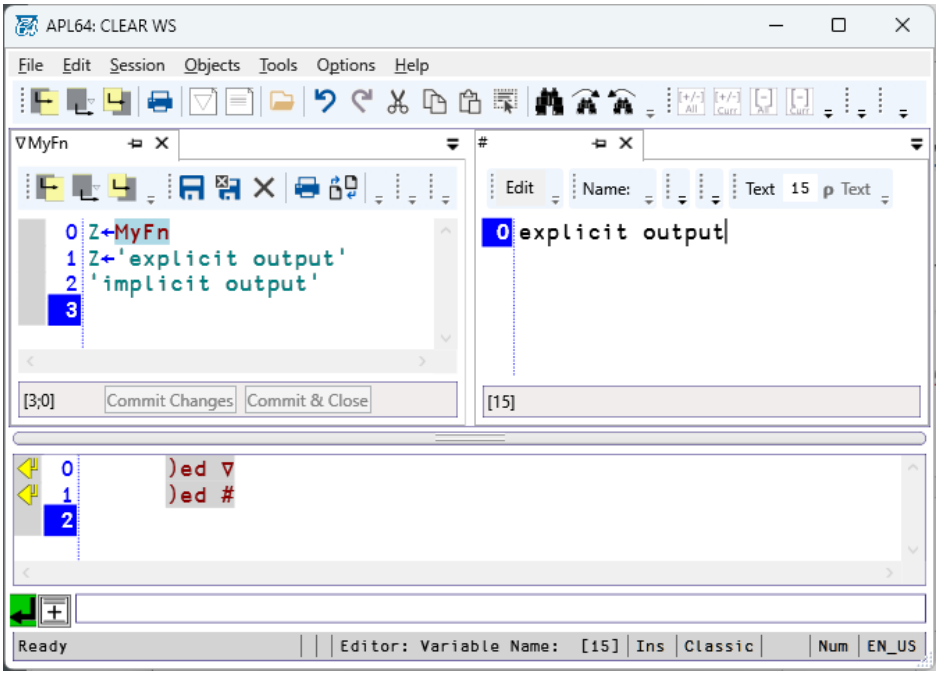
Example #2

In this example the user tool action text runs the 'MyFn' function:



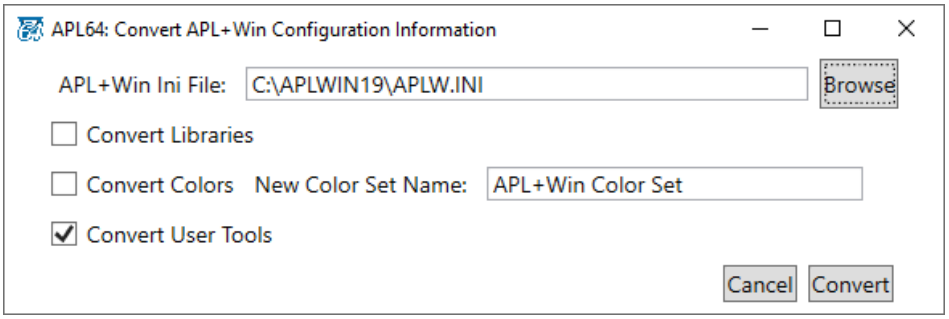


In APL64 the explicit output of this user tool, which is the explicit output of executing MyFn, is rendered to the APL64 variable editor, which has keyboard focus, at the caret position, and the implicit output is not considered, and the updated caret position is immediately after the insertion of the ‘explicit output’ text to the variable editor:



APL+Win User Tools Compatibility

Existing APL+Win tools can be converted to AP64 user tools in the Convert APL+Win Configuration Information dialogue accessed from the menu **Options | APL+Win Configuration Conversion**.



In APL+Win it may have been necessary to use □inbuf to direct the output of a user tool to an editor. APL64 eliminates the need for the □inbuf system function in a user tool definition. APL64 automatically directs the output, if any, of a user tool to the pane which has the keyboard focus. The APL64 □inbuf system function always directs it output to the editable history pane, or for other history pane formats, to the command line. Generally, the APL64 □inbuf system function should not be included in the action text of a user tool, unless output to the history pane is desired.