

New Enhancements and Bug Fixes Completed for Next APL64 Update

Table of Contents

Overview	3
Enhancements/Changes	3
Scalar operations have been optimized in APL64.....	3
New ☐ SQL (Microsoft SQL Server), ☐ SQLDB (IBM DB2), ☐ SQLMY (Oracle MySql) and ☐ SQLITE (Sqlite)	3
GUI Tools for Command Line Applications for ☐ ASKx system functions	3
☐ ASKx optional caption argument supported.....	4
☐ ASKE query the value of an APL64 executable expression	4
☐ ASKN scalar and vector numeric argument and result supported	4
☐ ASKL new system function for list element selection	4
☐ User documentation updated in Help APL Language Using ☐ASKx System Functions:	4
New ☐ XML actions: Descendants, NDescendant, DescendantsAndSelf, and NDescendantAndSelf	4
New ☐ XL actions: SortRange and UsedRange	5
Updates to ☐ EDSS and)EDSS	5
Updates to SaveAs options: For xlsx, xls, csv and Unicode formats implemented	5
New Sort Rows options for a user-selected range.....	6
New Print Option for the current worksheet selection	7
New Find & Replace Options	8
New ☐ EDITINFO action: DeleteEditEventsSubscriptions.....	9
☐ EDITEVENTS 'PreviewObjectNameSelection' event simplified	10
Quad (☐) input behavior improved	10
Implemented an option to exclude the CommandLine from the Ctrl+Tab list when the Classic History is editable	11
Toolbar images were added to some menu items in the File, Edit and Objects drop down menus.....	11
The Find/Replace Tool now closes when pressing the Esc key in the window	13
New option added to download all online documents to a target folder on the local machine	13
The Outer Syntax Error message was improved for a case with an ill-formed :if clause without an argument in a local inner function.....	14
The description for the ☐ trace system function was added to the System Functions document.....	15
New response code (~7) for ☐ copy and ☐ pcopy system functions.....	15

Additional information about creating and editing a user-defined function was added to the Help APL Language User-defined Functions document	15
Bug Fixes	15
⌕ '(1 2 3', ⌕ tclf, '4 5 6)' returned a result different from APL+Win.....	15
The auto underlining of strings with leading http in the editor stopped at the Quad (⌕) character..	16
A Quote-quad input could be repeated in the history when in the editable classic history mode.....	16
The ⌕ EDITEVENTS handler didn't produce an error message in the history	16
1 ⌕ EDITEVENTS 'EditStarted' 'OnEditStarted' had no effect during a suspension in the workspace ...	17
⌕ PW wasn't immediately updated after resizing the session window	17
)reset or)sic failed to reset the stack (state indicator) after suspending the workspace	18
Reduced the undesirable intermittent flicker in the editable Classic history when executing an APL expression	18
The cursor remained at the left edge instead of a six-space indent after the ⌕ Session Log Started message... displayed in the History.....	18
Unpredictable results with ⌕ ROWFIND when the size of the right argument was 64 rows or more..	19
The user-selected sizes and locations for the Open Object and Fetch Object dialogs were fixed	19

Overview

This document describes the enhancements and bug fixes that are planned for the next production release of APL64.

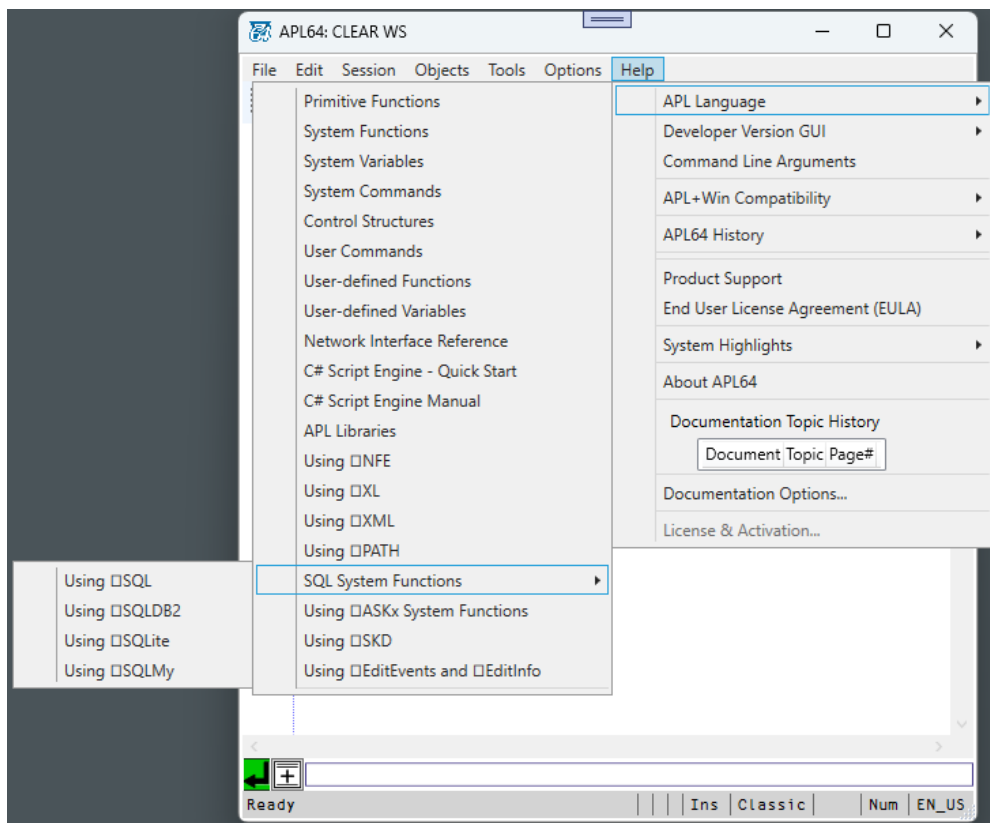
Enhancements/Changes

Scalar operations have been optimized in APL64

This was made possible with the new ☐it64 'ExecuteCache' parameter. The default value is 100.

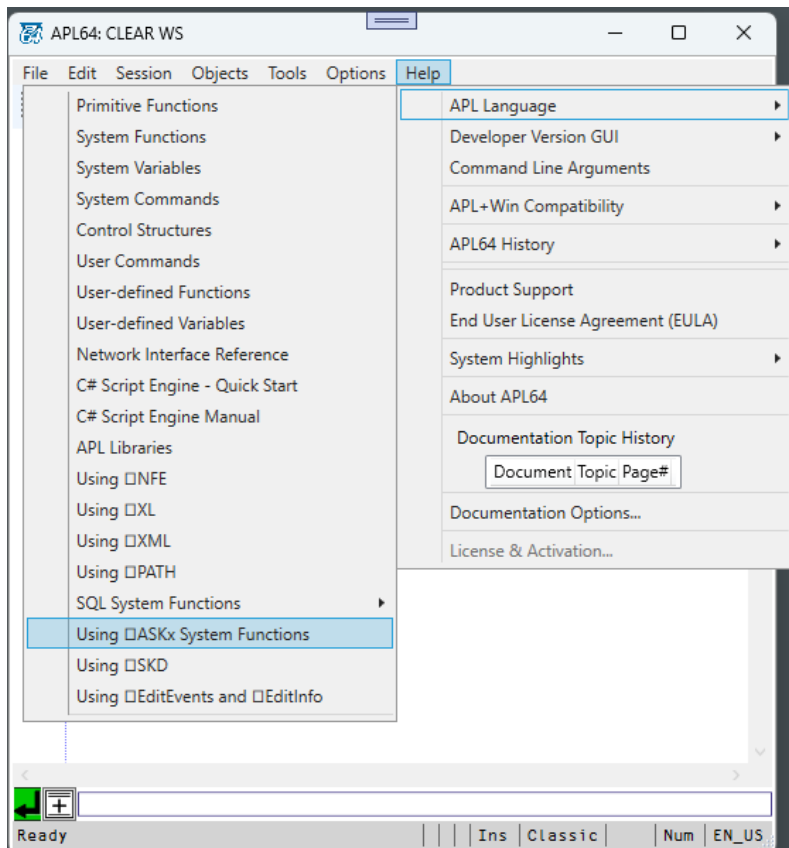
New ☐SQL (Microsoft SQL Server), ☐SQLDB (IBM DB2), ☐SQLMY (Oracle MySql) and ☐SQLITE (Sqlite)

New system functions to access relational databases using structured query language. Documentation and extensive examples are provided in the **Help | APL Language SQL System Functions** menu:



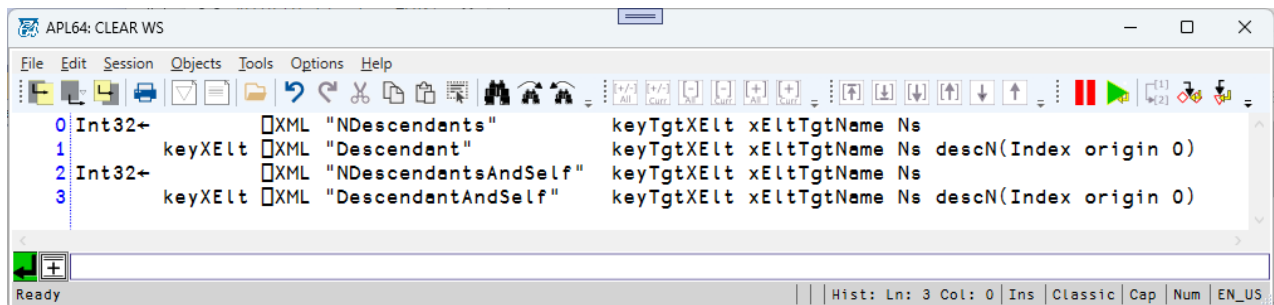
GUI Tools for Command Line Applications for ☐ASKx system functions

- ☐ ASKx optional caption argument supported
- ☐ ASKEX query the value of an APL64 executable expression
- ☐ ASKN scalar and vector numeric argument and result supported
- ☐ ASKL new system function for list element selection
- ☐ User documentation updated in [Help | APL Language | Using ☐ASKx System Functions:](#)



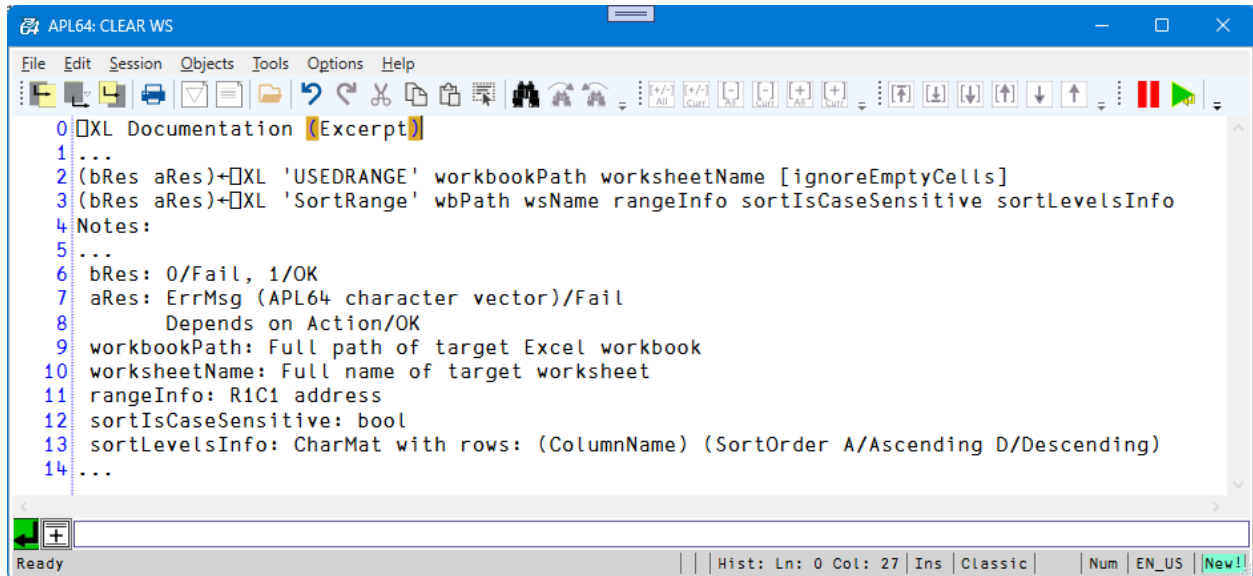
New ☐XML actions: Descendants, NDescendant, DescendantsAndSelf, and NDescendantAndSelf

Refer to [Help | APL Language | Using ☐XML](#) for the details.



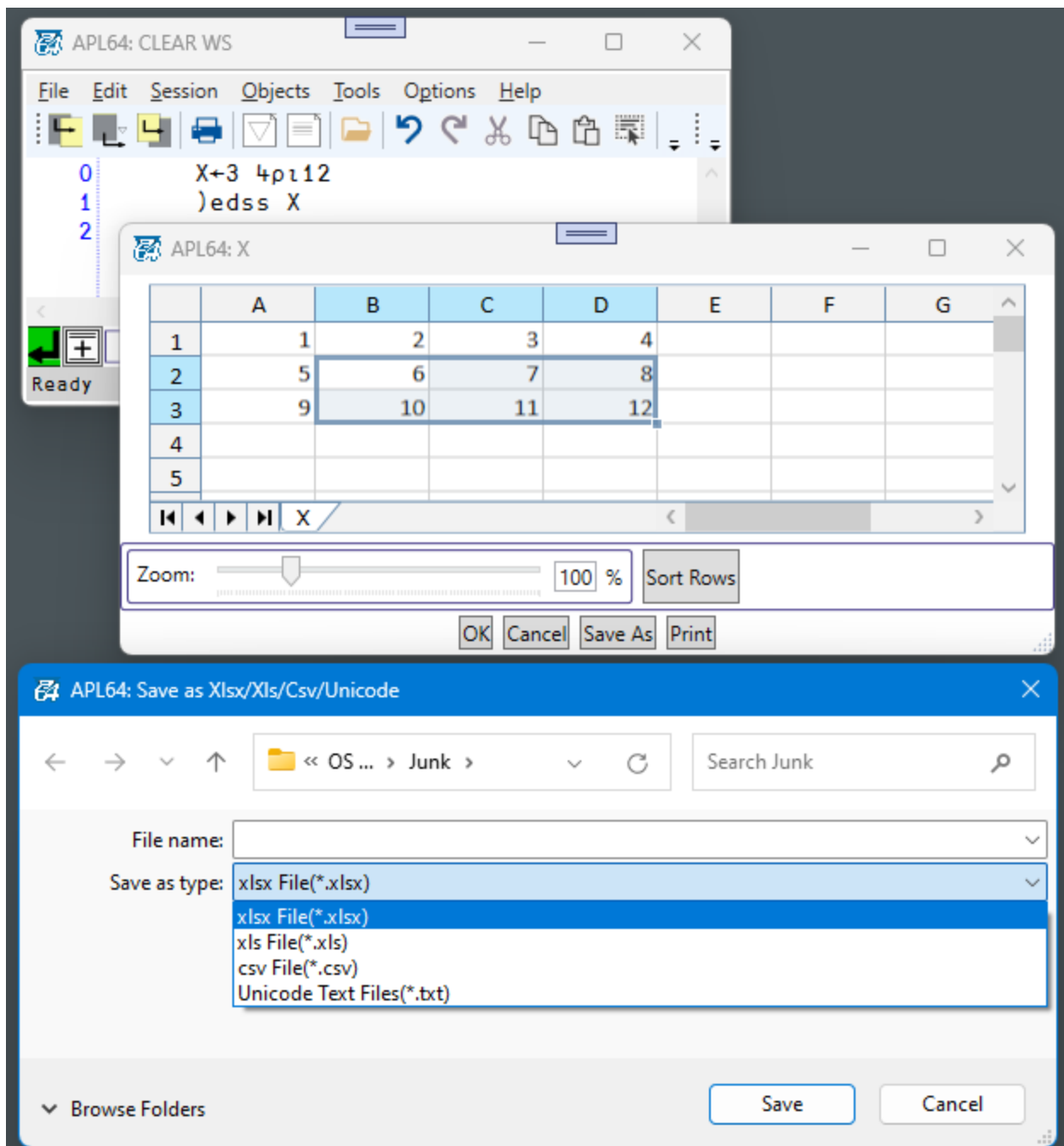
New XL actions: SortRange and UsedRange

Refer to the [Help](#) | [APL Language](#) | [Using !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\)XL](#) for the details.



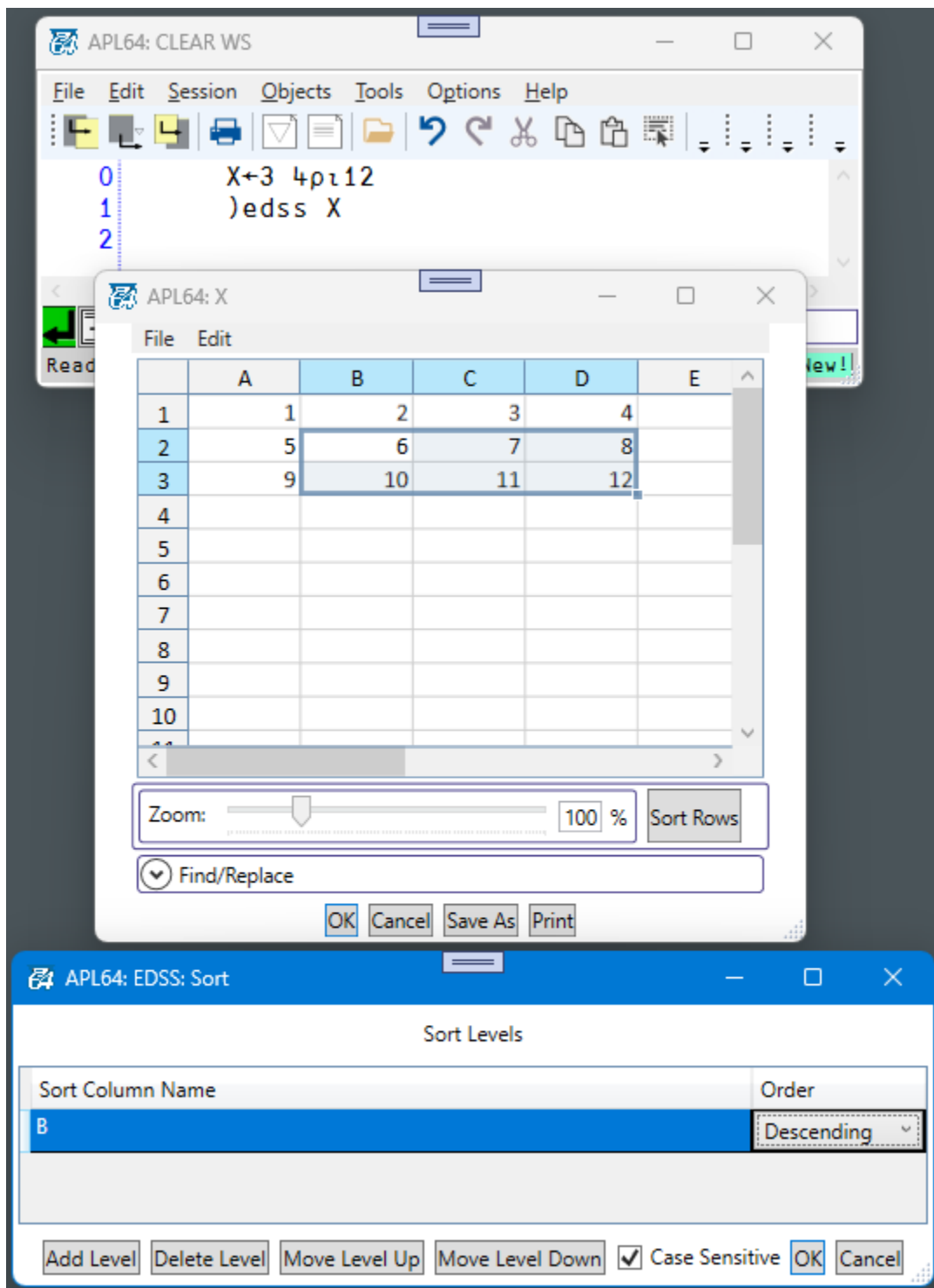
Updates to EDSS and)EDSS

Updates to SaveAs options: For xlsx, xls, csv and Unicode formats implemented

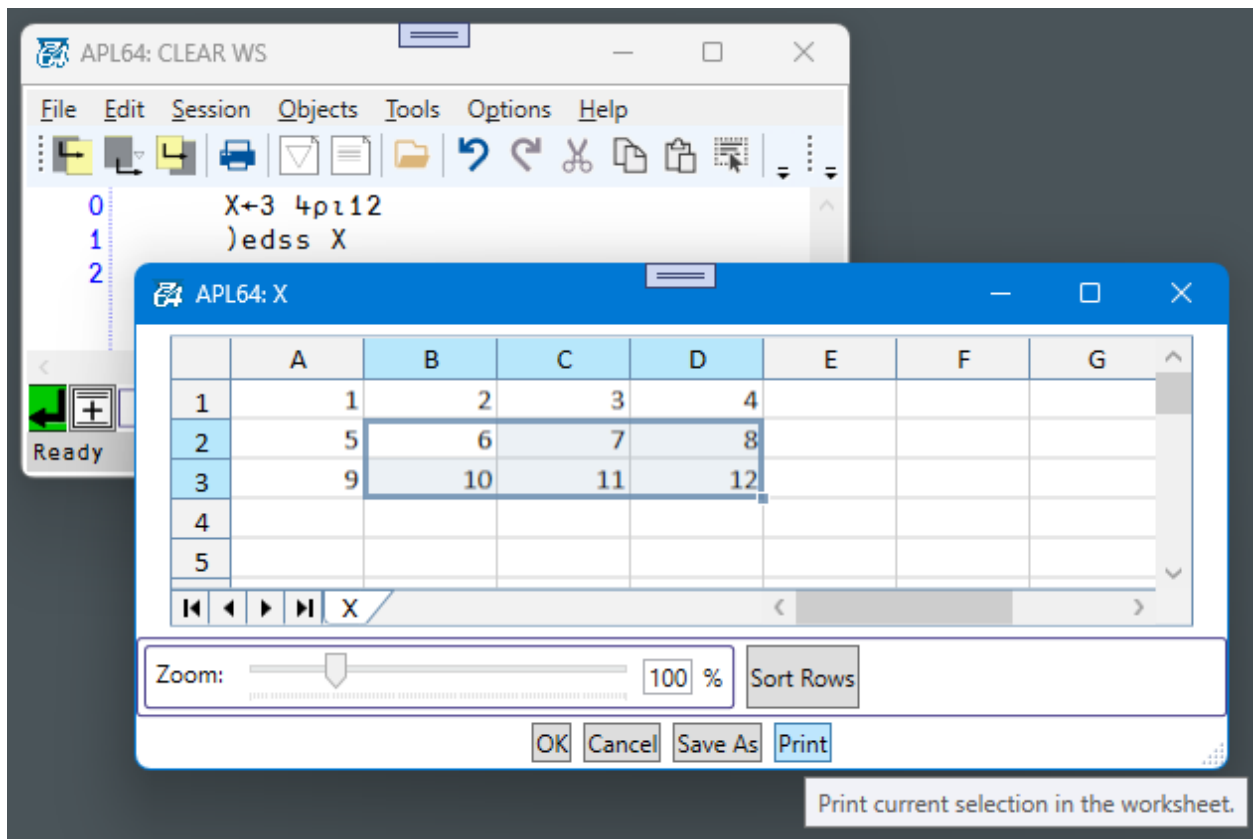


New Sort Rows options for a user-selected range

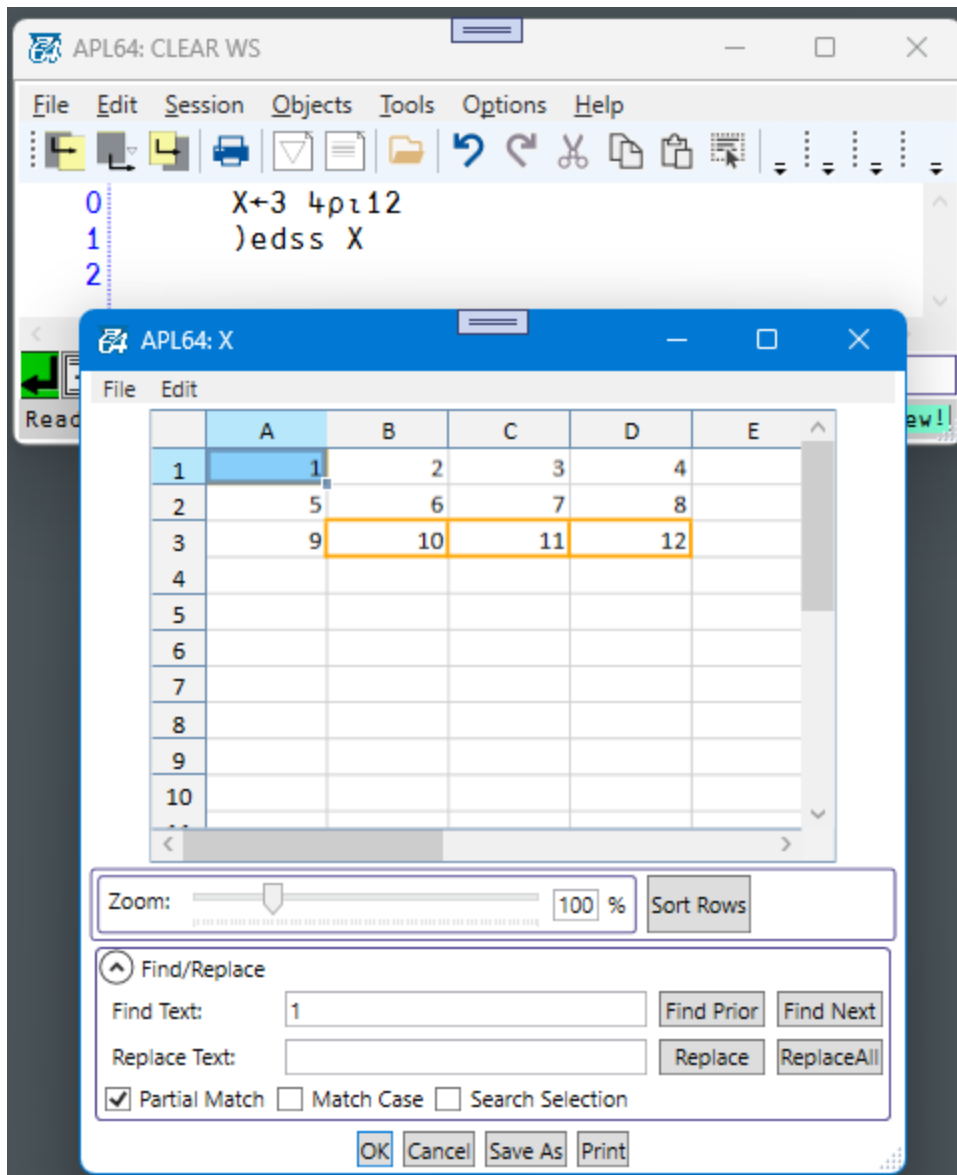
Select a single contiguous range, or the entire used range, in the worksheet and click the 'Sort Rows' button to be prompted for the sort levels:



New Print Option for the current worksheet selection



New Find & Replace Options



New EDITINFO action: DeleteEditEventsSubscriptions

Action: [DeleteEditEventsSubscriptions](#)

Use this action to delete all subscriptions or all subscriptions of a specified [EventType](#).
The abbreviated 'DeleteSubs' action name can also be used for this action.

Syntax:  editinfo 'DeleteEditEventsSubscriptions' [[EventType](#)]

The [EventType](#) argument is optional. If present, the [subscriptionInfo](#) result is filtered to contain event subscriptions of the type specified by the [EventType](#) argument. The available values for the [EventType](#) argument are:

- [EditEnded](#)
- [EditSaved](#)
- [EditStarted](#)
- [ObjectNameSelected](#)
- [PreviewObjectNameSelection](#)

□ EDITEVENTS 'PreviewObjectNameSelection' event simplified

PreviewObjectNameSelection

This event fires immediately before the object selector dialog is activated for user interaction. The object selector dialog is activated:

- When the Objects | Open or Objects | Fetch menu is used
- When the Ctrl+O or Ctrl+G shortcut is used
- When a visible object selector dialog regains the keyboard focus after losing the keyboard focus

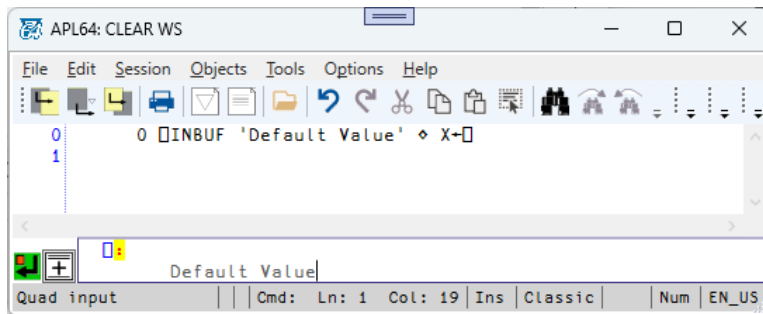
The purpose of a PreviewObjectNameSelection handler is to instantiate or delete APL64 variables or user-defined functions in the current workspace. When the PreviewObjectNameSelection handler is executed, the specified objects are instantiated in the current workspace and the object selector dialog will display those object names and type along with any previously-existing workspace objects.

eventName: 'PreviewObjectNameSelection'

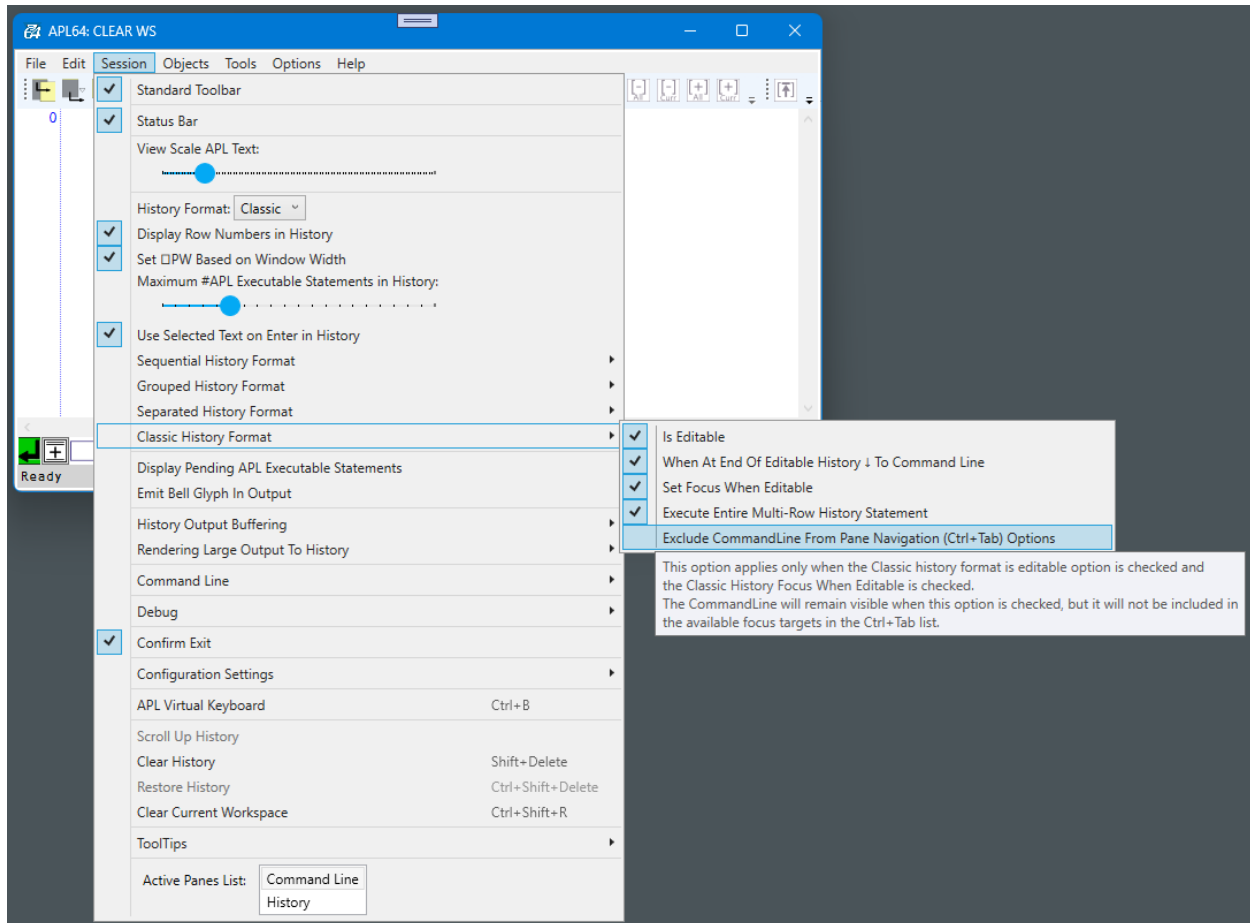
ehText: The PreviewObjectNameSelection event handler can be an APL64 user-defined function or an APL executable statement:

- If ehText is the name of a user-defined APL function
 - It must have no result
 - It must no left argument
 - It must have a right argument and currently the right argument is reserved for future use
 - It must not produce any output to the history
- If ehText is the text of a user-defined APL executable statement
 - It must not produce any output to the history

Quad (□) input behavior improved

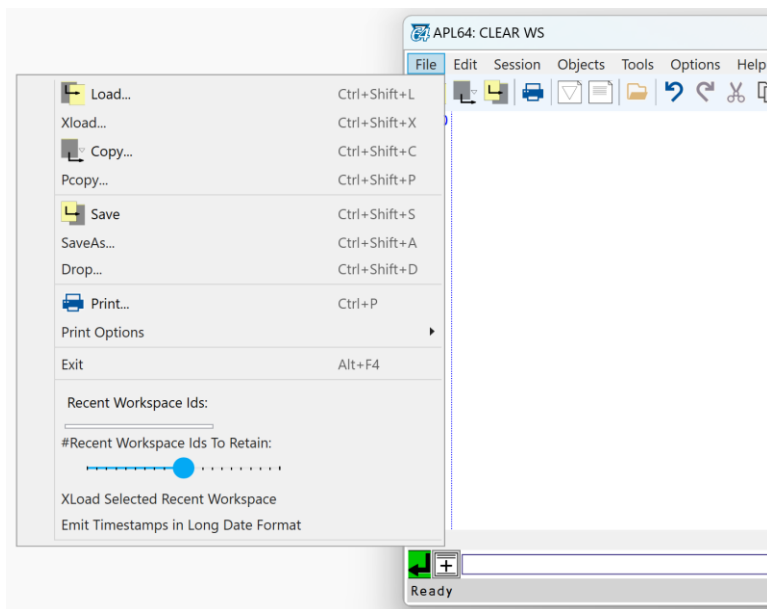


Implemented an option to exclude the CommandLine from the Ctrl+Tab list when the Classic History is editable

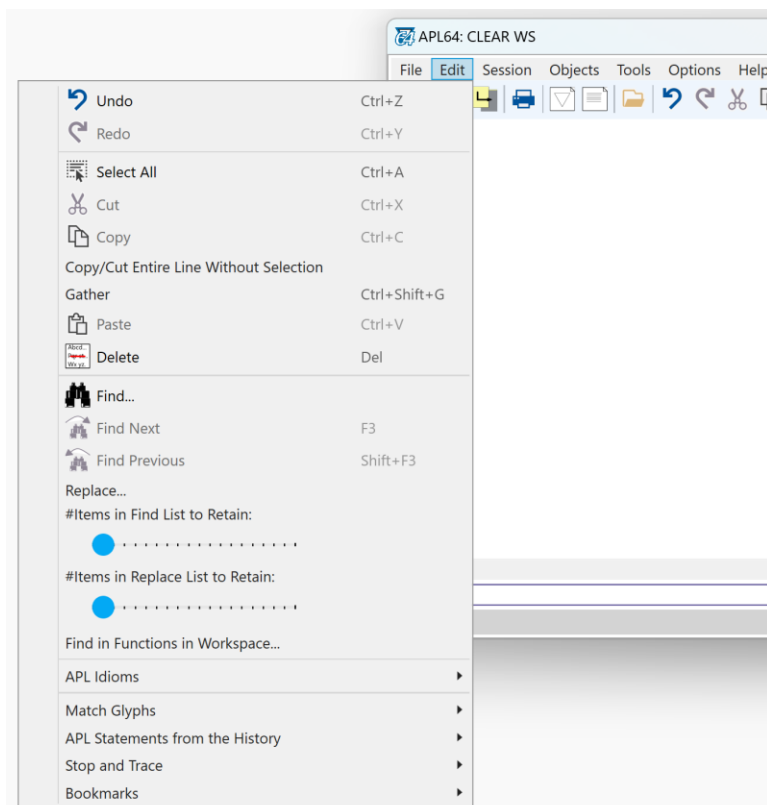


Toolbar images were added to some menu items in the File, Edit and Objects drop down menus

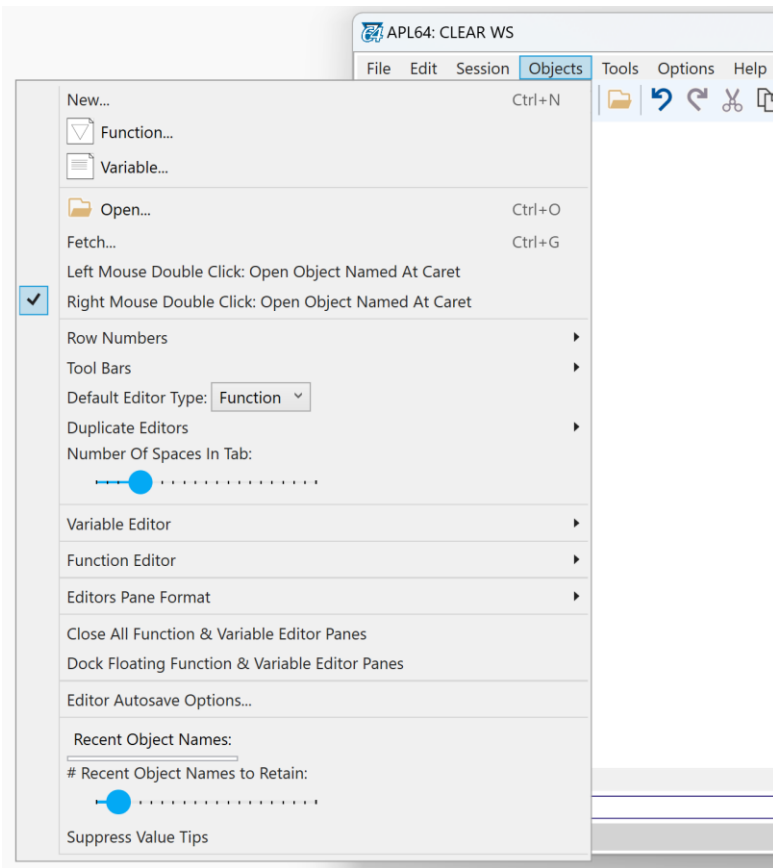
Images on Load, Copy, Save and Print on File menu:



Images on Undo, Redo, Select All, Cut, Copy, Paste, Delete, Find, Find Next and Find Previous on Edit menu:



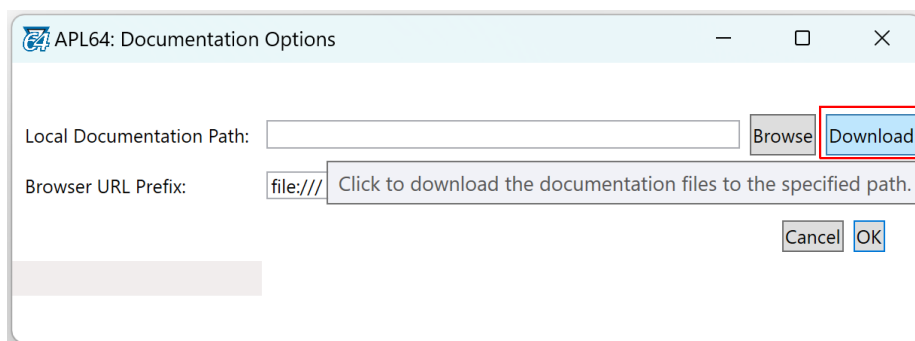
Images Function, Variable and Open menu items on Objects menu:



The Find/Replace Tool now closes when pressing the Esc key in the window

New option added to download all online documents to a target folder on the local machine

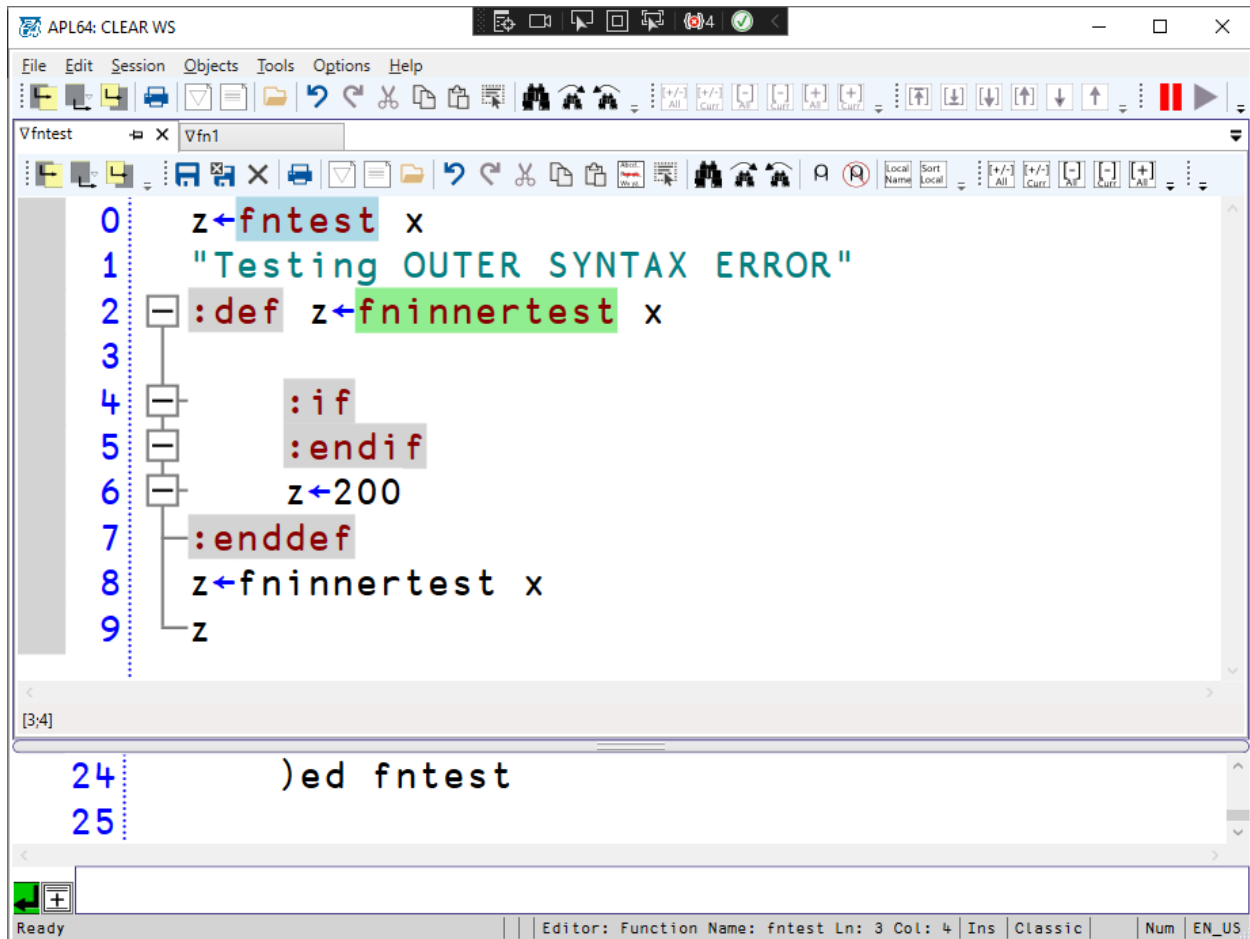
Click Download to download all the documentation files related to this specific version to the specified path.



Note: This does not change the fact that a local pdf document file cannot be opened to a specified page or bookmark under program control.

The Outer Syntax Error message was improved for a case with an ill-formed :if clause without an argument in a local inner function

E.g.,

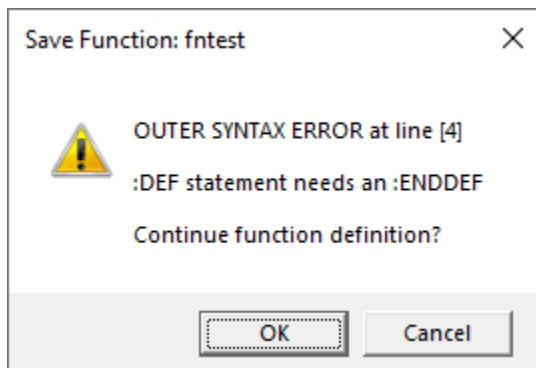


The screenshot shows the APL64: CLEAR WS editor window. The main editor area displays the following code:

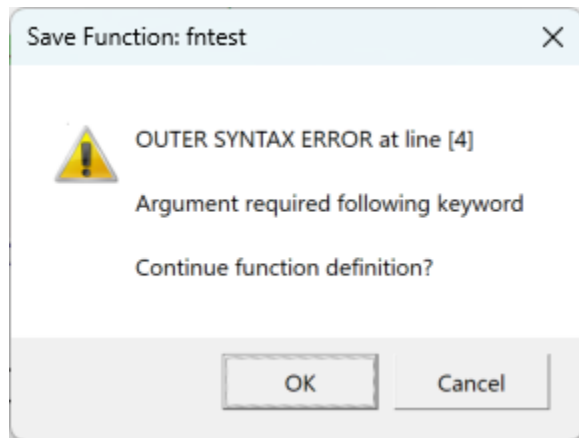
```
0  z←fntest x
1  "Testing OUTER SYNTAX ERROR"
2  :def z←fninnertest x
3  |
4  |      :if
5  |      :endif
6  |      z←200
7  |      :enddef
8  |      z←fninnertest x
9  |      z
```

The code is highlighted with various colors: blue for line numbers, red for keywords, and green for function names. A vertical line is positioned at line 4, column 4. Below the main editor, there is a status bar showing "Ready" and "Editor: Function Name: fntest Ln: 3 Col: 4 Ins Classic Num EN_US".

resulted in the dialog error message below when saving and exiting (Ctrl+E) the function editor:



Below is the new and improved dialog error message:



The description for the `⎕trace` system function was added to the System Functions document

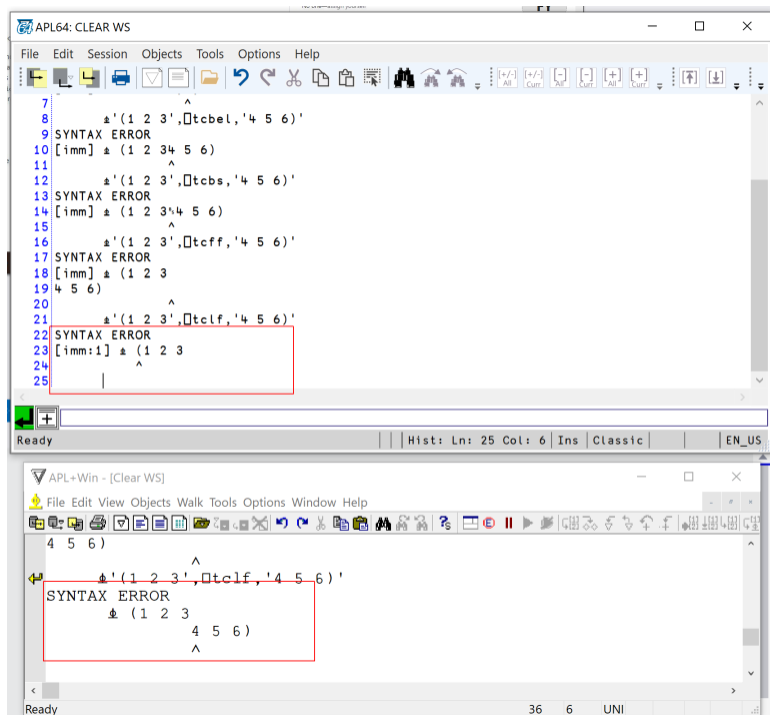
New response code (~7) for `⎕copy` and `⎕pcopy` system functions

This response code indicates the system was unable to copy the named object (e.g., `⎕mom`) from a workspace when unsupported in APL64.

Additional information about creating and editing a user-defined function was added to the [Help | APL Language | User-defined Functions](#) document

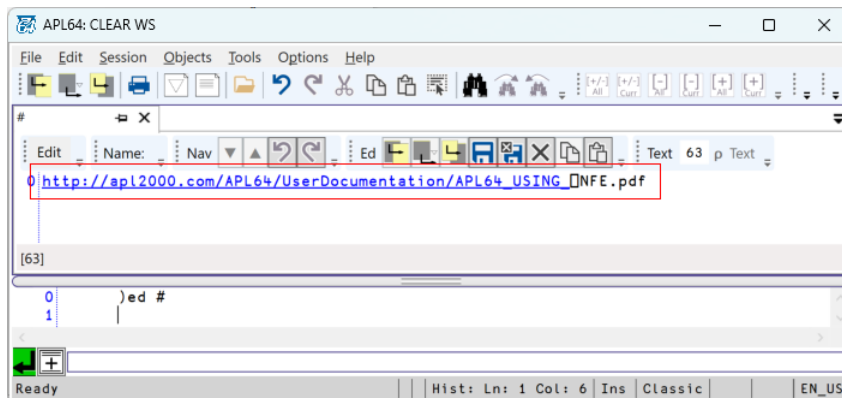
Bug Fixes

`⌵'(1 2 3',⎕tclf,'4 5 6)'` returned a result different from APL+Win
E.g.,



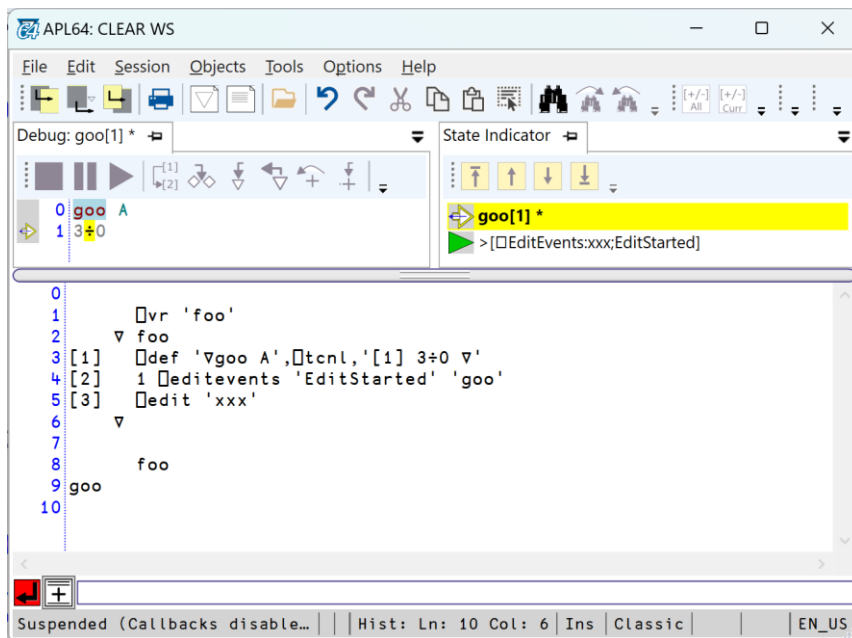
The auto underlining of strings with leading **http** in the editor stopped at the Quad (⍺) character

E.g.,

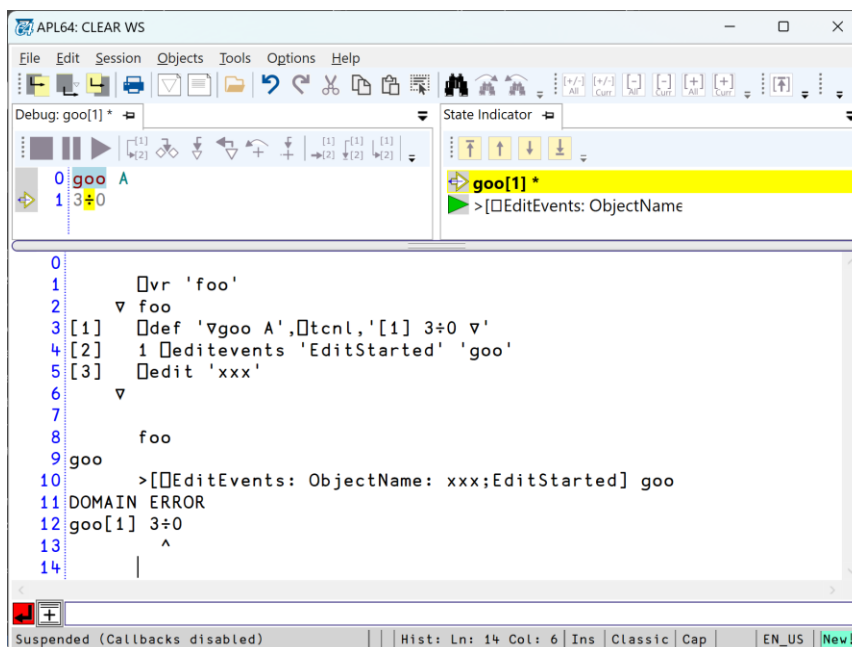


A Quote-quad input could be repeated in the history when in the editable classic history mode

The ⍺EDITEVENTS handler didn't produce an error message in the history
Example of the pre-fix behavior:



Same example with the post-fix behavior:



1 ☐ EDITEVENTS 'EditStarted' 'OnEditStarted' had no effect during a suspension in the workspace

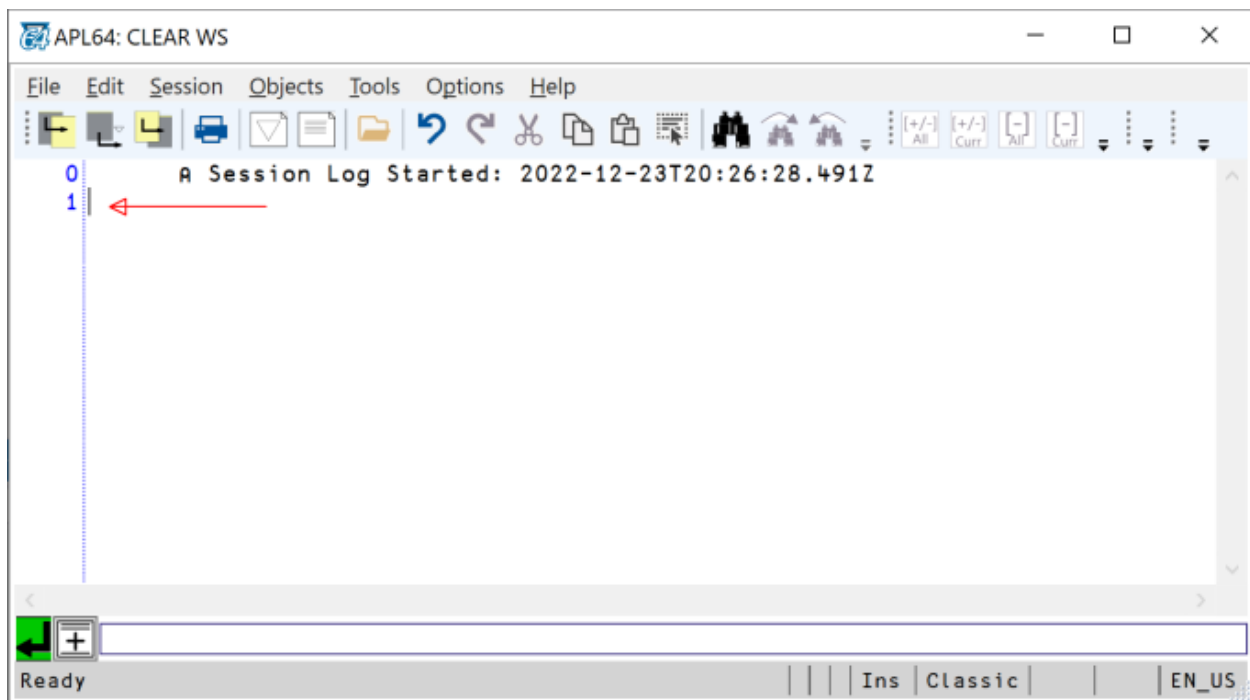
☐ PW wasn't immediately updated after resizing the session window

With the **Session | Set ☐ PW based on Window Width** menu enabled, resizing the session window then querying the value of ☐ PW did report the new value.

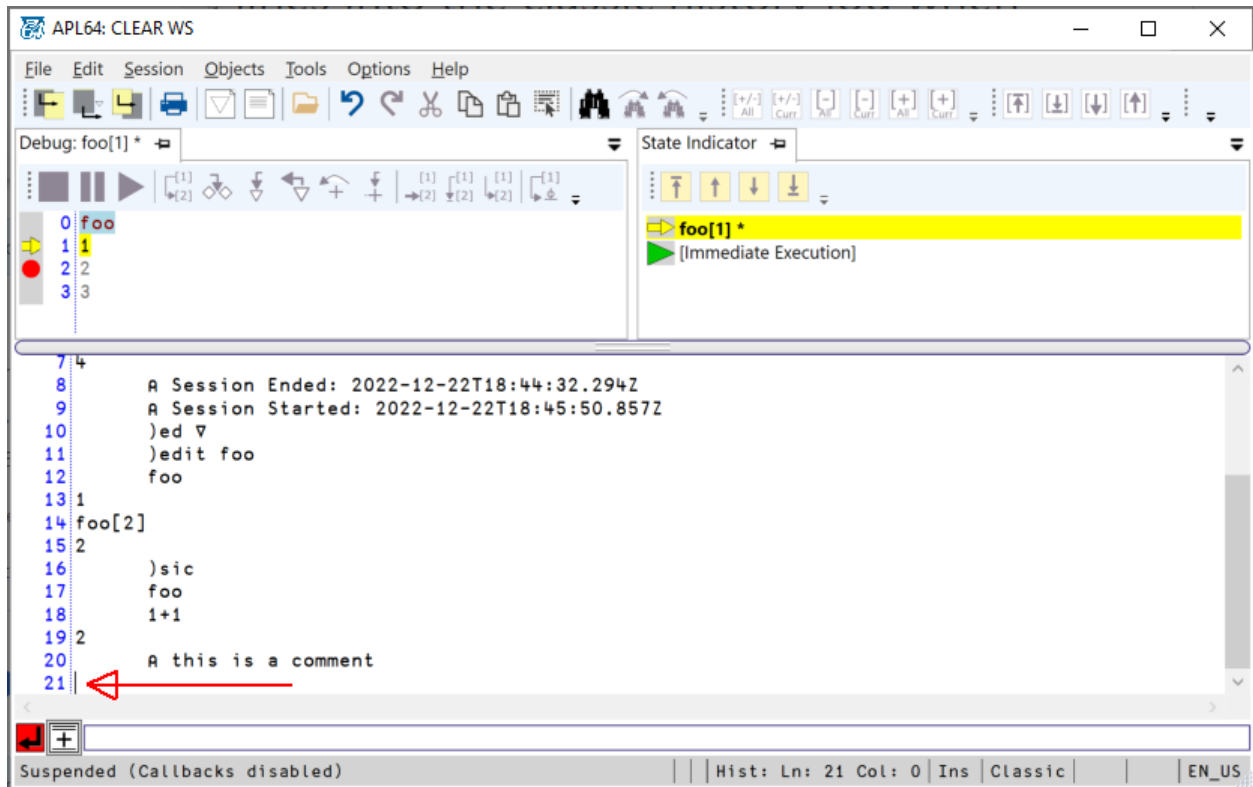
)reset or)sic failed to reset the stack (state indicator) after suspending the workspace
Addressed multiple scenarios that would prevent the)reset and)sic commands from clearing the state indicator after a debug suspension in the workspace that resulted in ☐wi forms from responding.

Reduced the undesirable intermittent flicker in the editable Classic history when executing an APL expression

The cursor remained at the left edge instead of a six-space indent after the
[@ Session Log Started message...](#) displayed in the History



A similar behavior was observed while suspended in the debugger then simply executing a comment line in the editable history as demonstrated below:



Unpredictable results with ☐ ROWFIND when the size of the right argument was 64 rows or more

The user-selected sizes and locations for the Open Object and Fetch Object dialogs were fixed

APL64 now records the user-selected display sizes and locations when these dialogs are closed to be reused the next time they're opened.