

New Updates and Bug Fixes Since Production Version 2023.0.1

Table of Contents

Overview	3
Enhancements	3
Scalar operations have been optimized in APL64.....	3
New <input type="checkbox"/> SQL (Microsoft SQL Server), <input type="checkbox"/> SQLDB (IBM DB2), <input type="checkbox"/> SQLMY (Oracle MySql) and <input type="checkbox"/> SQLITE (Sqlite)	3
GUI Tools for Command Line Applications: <input type="checkbox"/> ASKx system functions	4
<input type="checkbox"/> ASKx optional caption argument supported.....	4
<input type="checkbox"/> ASKEX query the value of an APL64 executable expression	4
<input type="checkbox"/> ASKN scalar and vector numeric argument and result supported	4
<input type="checkbox"/> ASKL new system function for list element selection	4
<input type="checkbox"/> User documentation updated in Help APL Language Using <input type="checkbox"/>ASKx System Functions:	4
New <input type="checkbox"/> XML actions: Descendants, NDescendant, DescendantsAndSelf, and NDescendantAndSelf	4
New <input type="checkbox"/> XL actions: SortRange and UsedRange	5
Updates to <input type="checkbox"/> EDSS and)EDSS	6
Updates to SaveAs options for xlsx, xls, csv and Unicode formats:.....	6
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 <input type="checkbox"/> EDITINFO action: DeleteEditEventsSubscriptions.....	9
<input type="checkbox"/> EDITEVENTS 'PreviewObjectNameSelection' event simplified	10
Quad (<input type="checkbox"/>) input behavior improved	10
Option to exclude the CommandLine from the Ctrl+Tab list in editable Classic History:	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 <input type="checkbox"/> trace system function was added to the System Functions document.....	15
New response code (~7) for <input type="checkbox"/> copy and <input type="checkbox"/> 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

The user-selected sizes and locations for the Open Object and Fetch Object dialogs are retained	15
Performance improvements for <code>⎕DLB</code> , <code>⎕DLTB</code> , <code>⎕DTB</code> , and <code>⎕ROWFIND</code>	15
Bug Fixes	16
<code>⊘ '(1 2 3', ⎕tclf, '4 5 6)'</code> returned a result different from APL+Win.....	16
The auto underlining of strings with leading <code>http</code> in the editor stopped at the Quad (<code>⎕</code>) character..	16
The <code>⎕TRACE</code> output in APL64 was different and didn't display the value of the calculation on the traced lines like APL+Win.....	17
A Quote-quad input could be repeated in the history when in the editable classic history	17
The <code>⎕EDITEVENTS</code> handler didn't produce an error message in the history	17
1 <code>⎕EDITEVENTS</code> 'EditStarted' 'OnEditStarted' had no effect during a suspension in the workspace ...	18
<code>⎕PW</code> wasn't immediately updated after resizing the session window	18
<code>)reset</code> or <code>)sic</code> 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 <code>⌘ Session Log Started</code> message... displayed in the History.....	19
Unpredictable results with <code>⎕ROWFIND</code> when the size of the right argument has 64 or more rows ...	20
Grade-up (<code>⌈</code>) and grade-down (<code>⌋</code>) with Booleans returned incorrect results.....	20
<code>⎕MF</code> of a function with a <code>:returnif</code> clause in a <code>:case</code> block returned a SYSTEM ERROR	20
Saving a function with system functions in the header row didn't report an error on line 0	20
<code>ρ⎕NA 'xxx'</code> returned the result <code>0 0</code> instead of <code>0</code>	20
The interpreter status displayed "Running" when the initial function editor was created in a new APL64 session.....	21
The interpreter status displayed "Running" when loading some APL+Win workspaces from the File Load menu	21
Every load of an APL+Win workspace caused an APL32Now32 Runtime System (32-bit) process to accumulate in the Windows Task Manager.....	21

Overview

This document describes the updates and bug fixes incorporated into this 2023.0.2 production release of APL64.

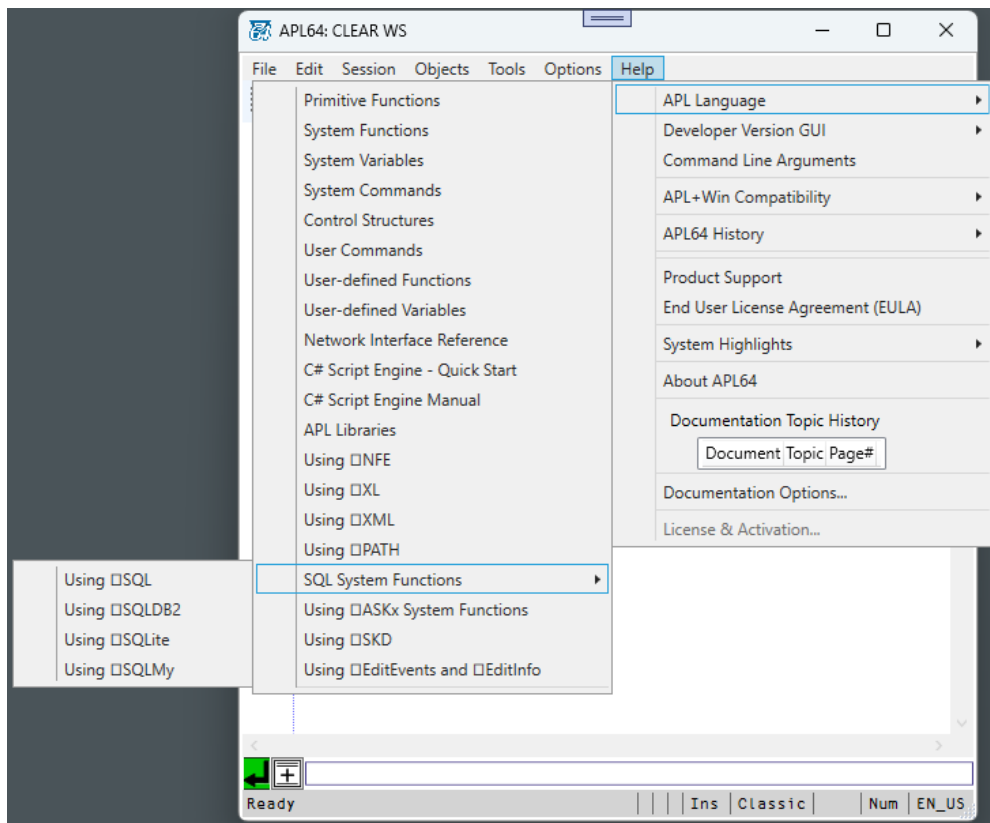
Enhancements

Scalar operations have been optimized in APL64

The new `⎕it64` 'ExecuteCache' setting controls how many recent code blocks are cached to improve performance in scalar operations. The default is 100 with a max of 100000. . 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: ASKx system functions

ASKx optional caption argument supported

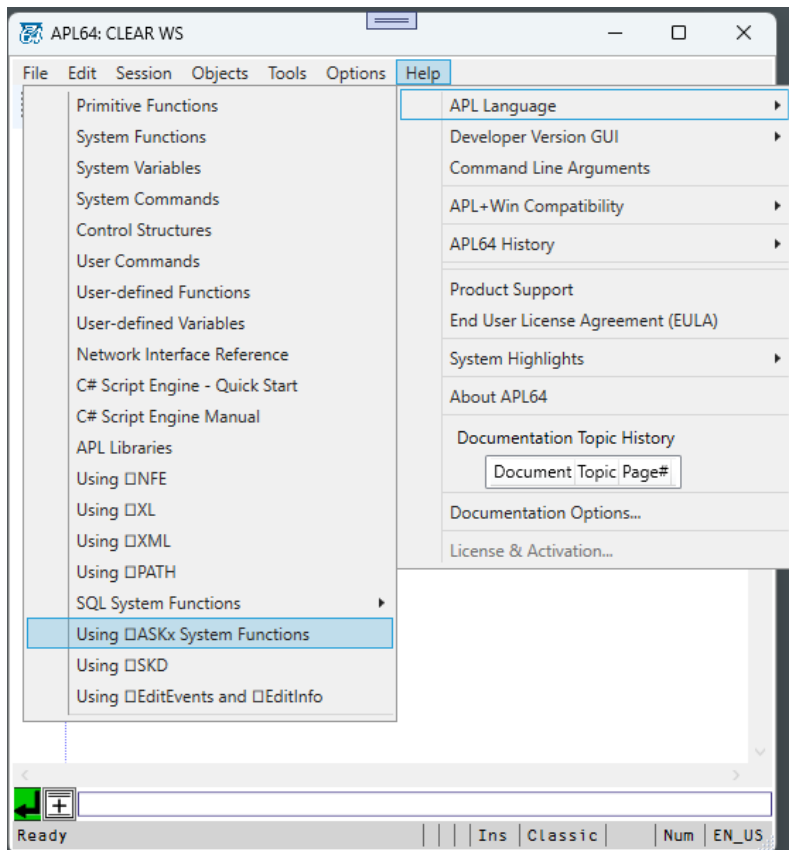
ASKEX query the value of an APL64 executable expression

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:**

Documentation and examples are provided:



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

Refer to **Help | APL Language | Using XML** for the details.

```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
0 Int32← □XML "NDescendants" keyTgtXElt xEltTgtName Ns
1 keyXElt □XML "Descendant" keyTgtXElt xEltTgtName Ns descN(Index origin 0)
2 Int32← □XML "NDescendantsAndSelf" keyTgtXElt xEltTgtName Ns
3 keyXElt □XML "DescendantAndSelf" keyTgtXElt xEltTgtName Ns descN(Index origin 0)
Ready | Hist: Ln: 3 Col: 0 | Ins | Classic | Cap | Num | EN_US

```

New □XL actions: `SortRange` and `UsedRange`
Refer to the **Help | APL Language | Using □XL** for the details.

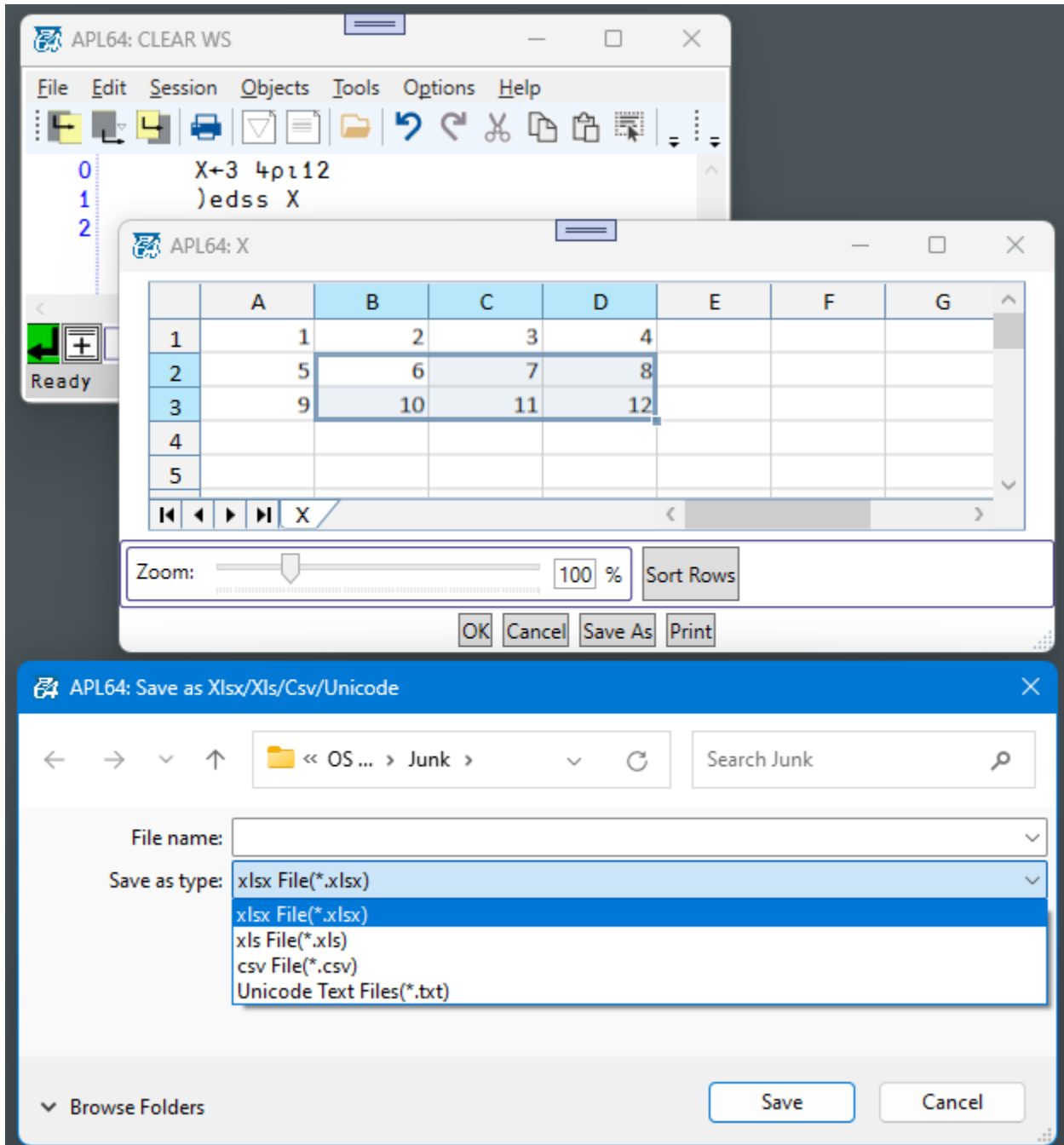
```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
0 □XL Documentation (Excerpt)
1 ...
2 (bRes aRes)+□XL 'USEDRANGE' workbookPath worksheetName [ignoreEmptyCells]
3 (bRes aRes)+□XL 'SortRange' wbPath wsName rangeInfo sortIsCaseSensitive sortLevelsInfo
4 Notes:
5 ...
6 bRes: 0/Fail, 1/OK
7 aRes: ErrMsg (APL64 character vector)/Fail
8 Depends on Action/OK
9 workbookPath: Full path of target Excel workbook
10 worksheetName: Full name of target worksheet
11 rangeInfo: R1C1 address
12 sortIsCaseSensitive: bool
13 sortLevelsInfo: CharMat with rows: (ColumnName) (SortOrder A/Ascending D/Descending)
14 ...
Ready | Hist: Ln: 0 Col: 27 | Ins | Classic | Num | EN_US | New!

```

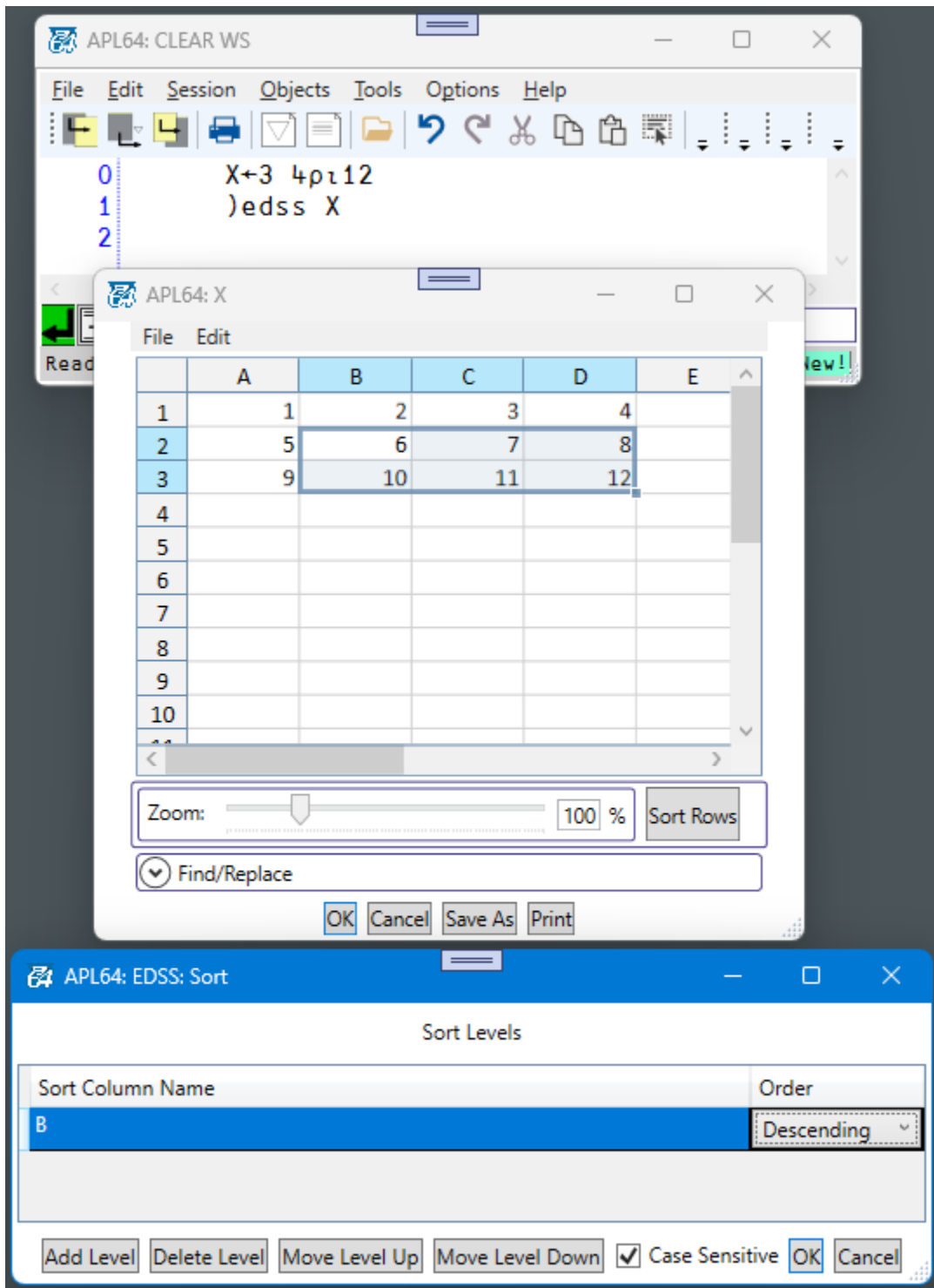
Updates to □EDSS and)EDSS

Updates to SaveAs options for.xlsx, .xls, .csv and Unicode formats:

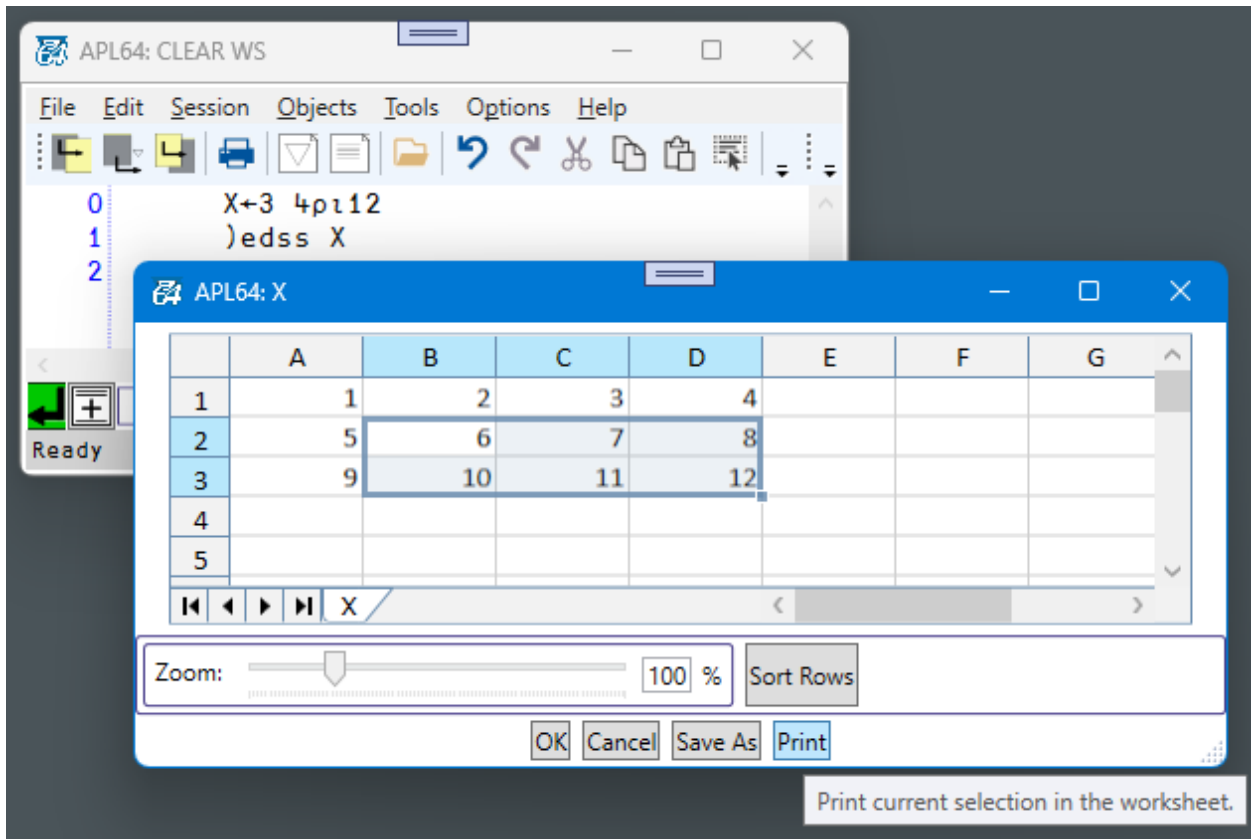


New Sort Rows options for a user-selected range

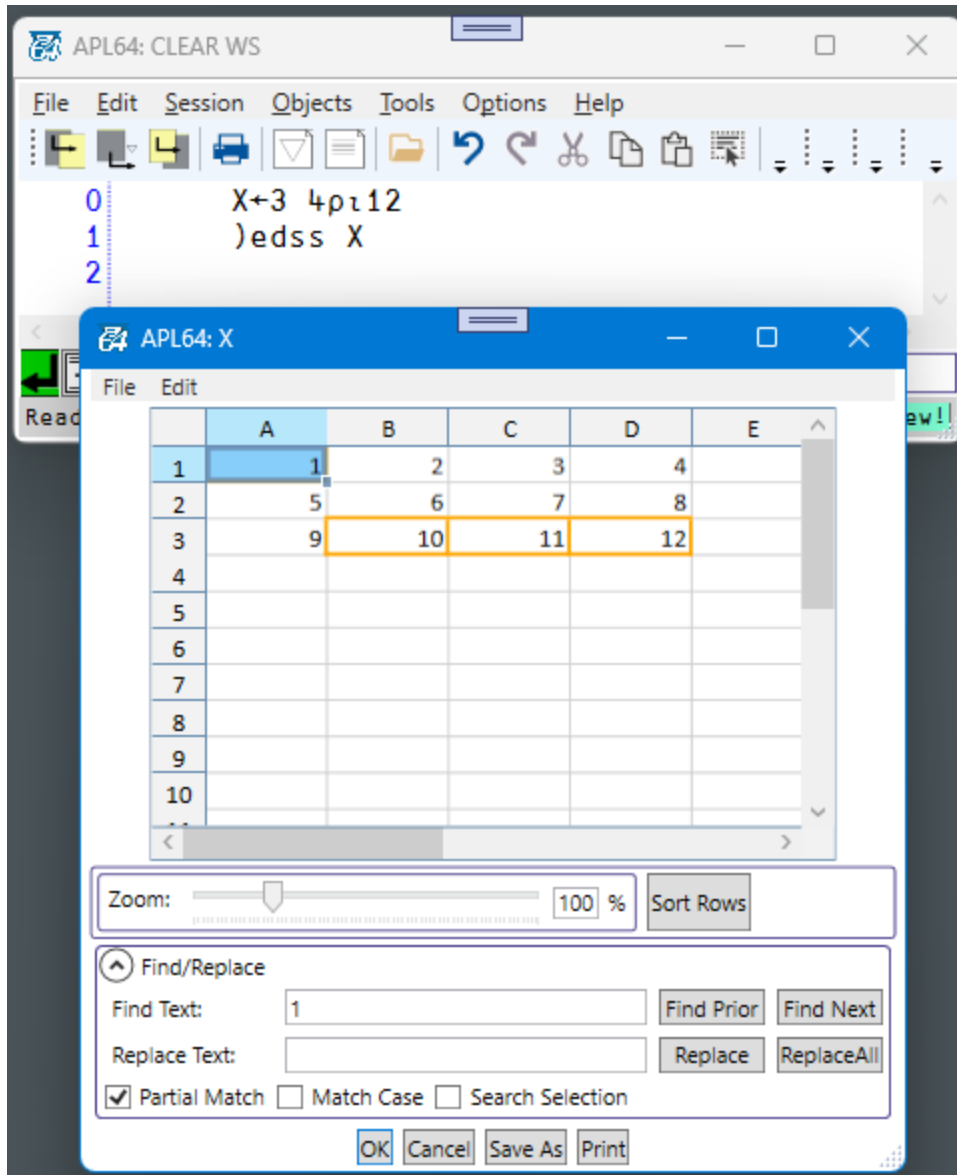
Select a single contiguous range, or the entire used range, in the worksheet and click the 'Sort Rows' button to be prompted for the sort levels:



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

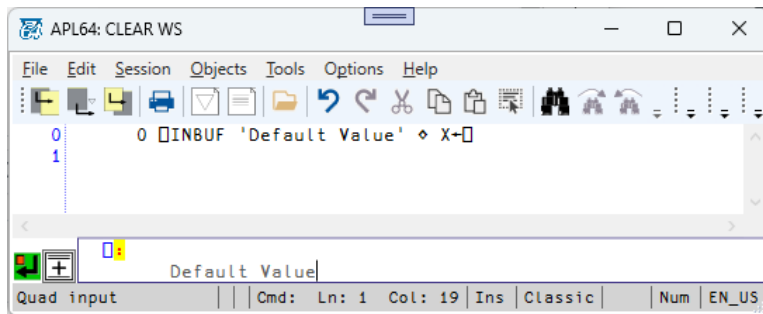
The purpose of a PreviewObjectNameSelection handler is to instantiate or delete APL64 variables or user-defined functions in the current workspace. When the PreviewObjectNameSelection handler is executed, the specified objects are instantiated in the current workspace and the object selector dialog will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

eventName: 'PreviewObjectNameSelection'

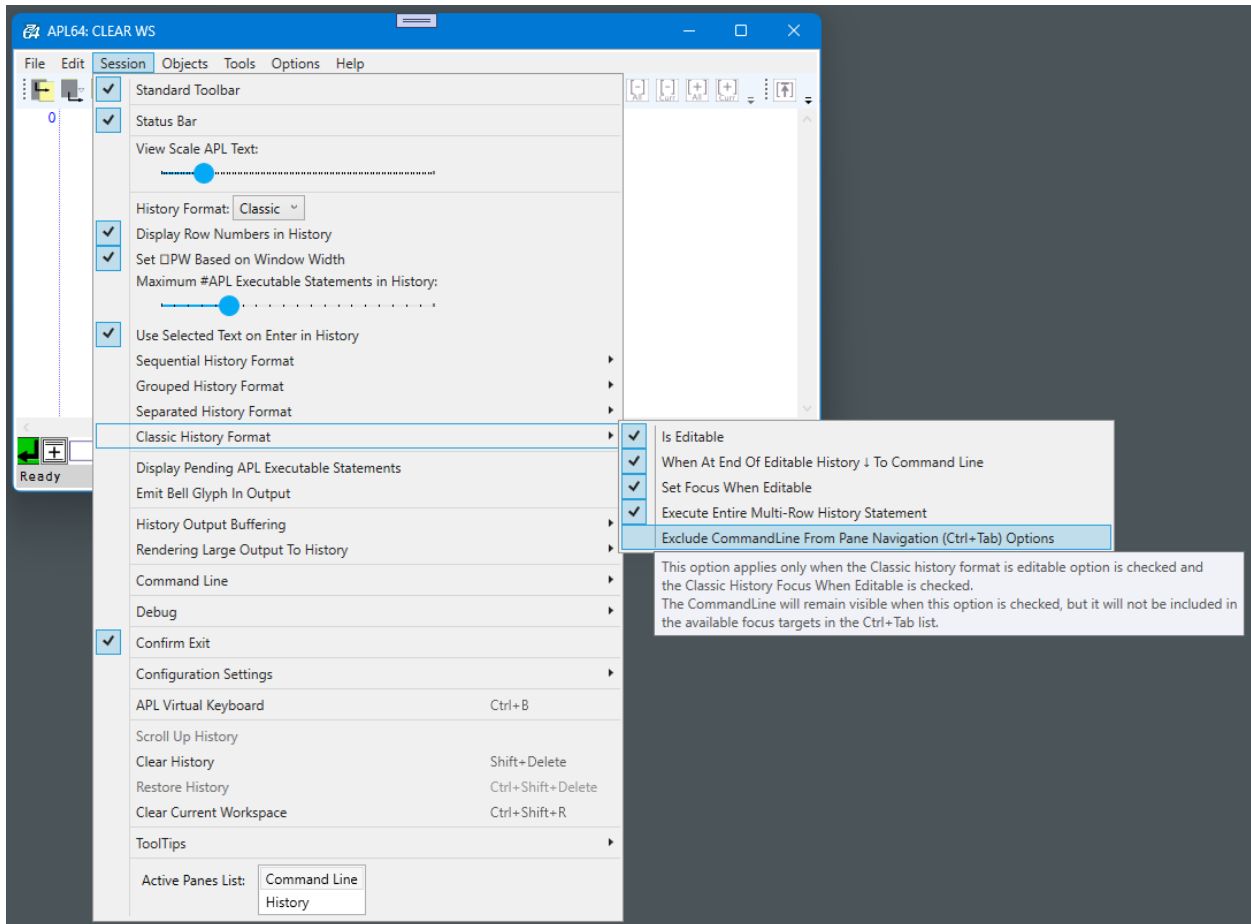
ehText: The PreviewObjectNameSelection event handler can be an APL64 user-defined function or an APL executable statement:

- 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

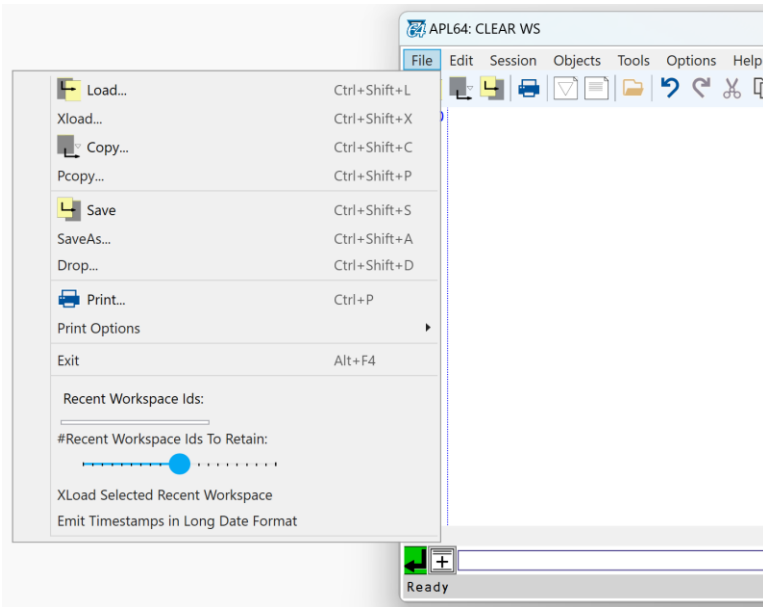


Option to exclude the CommandLine from the Ctrl+Tab list in editable Classic History:

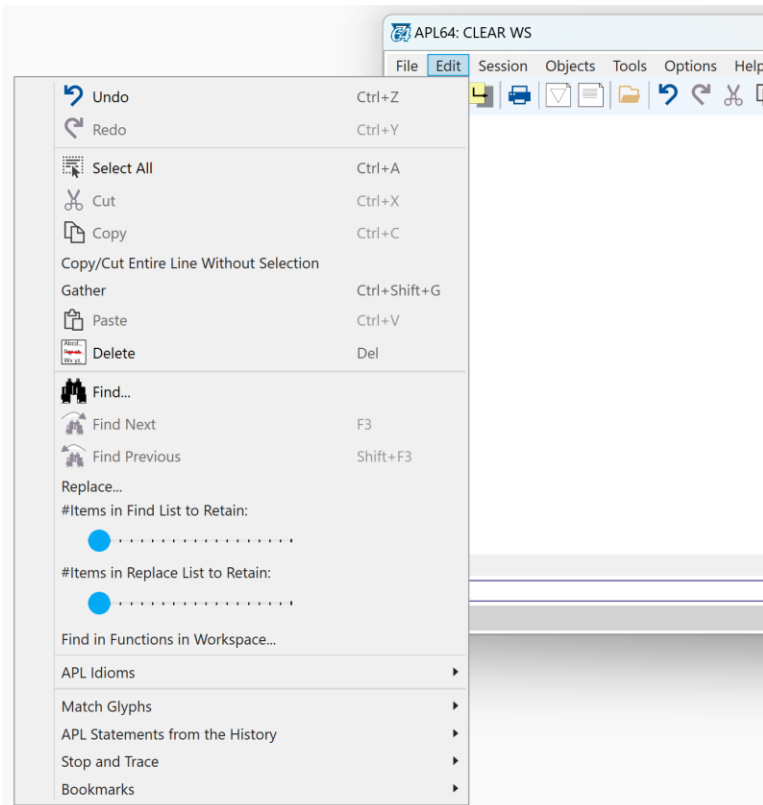


Toolbar images were added to some menu items in the File, Edit and Objects drop down menus

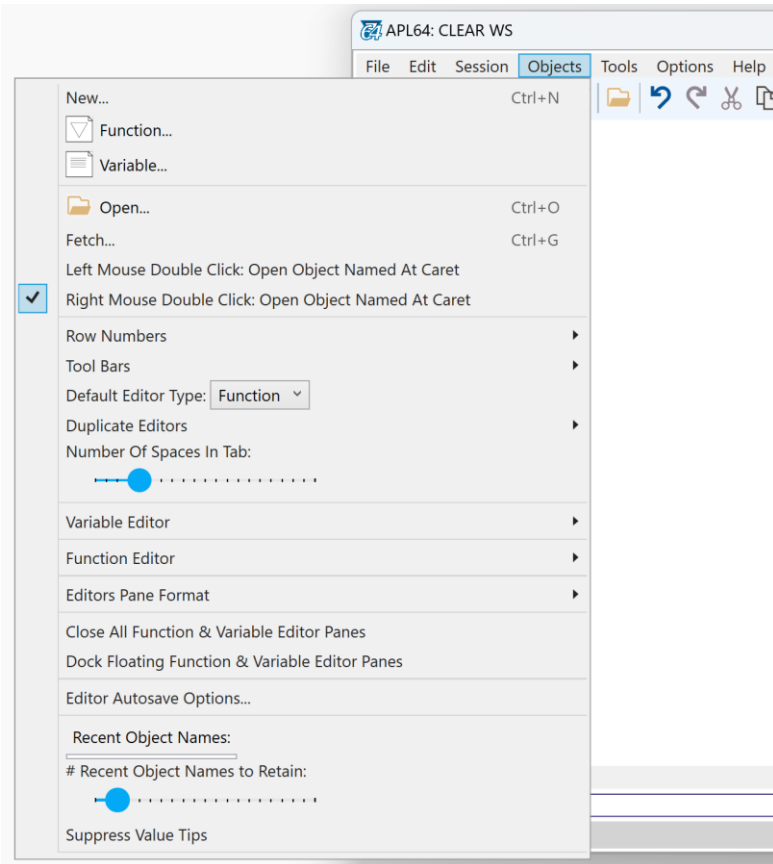
Images on Load, Copy, Save and Print on File menu:



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



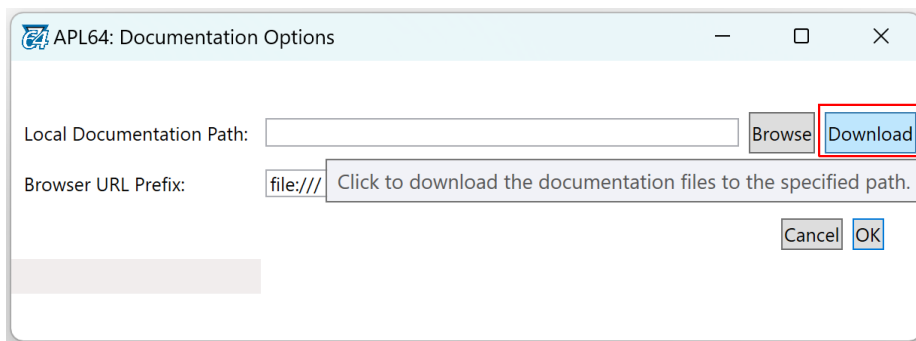
Images Function, Variable and Open menu items on Objects menu:



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

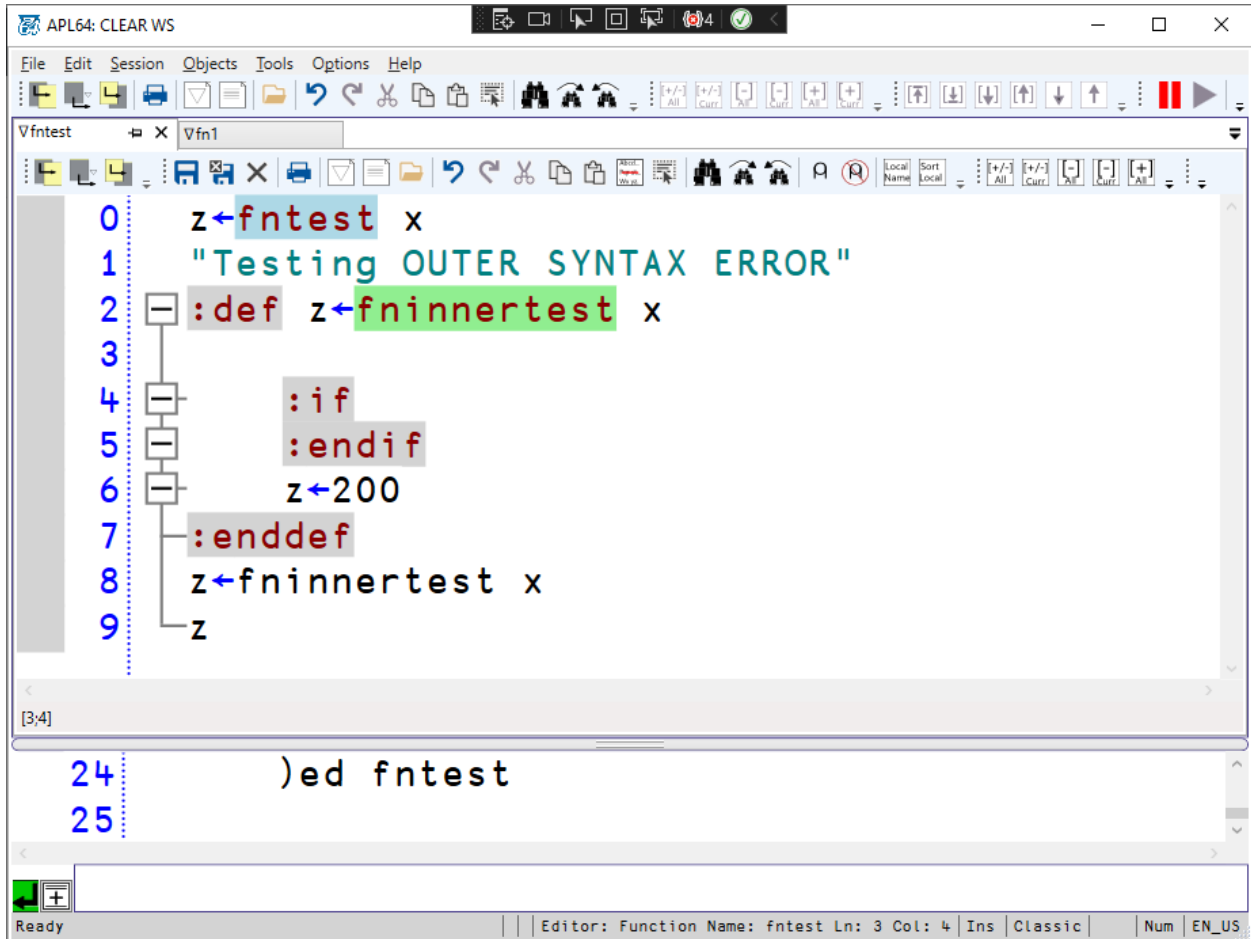
Click Download to download all the documentation files related to this specific version to the specified path.



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.

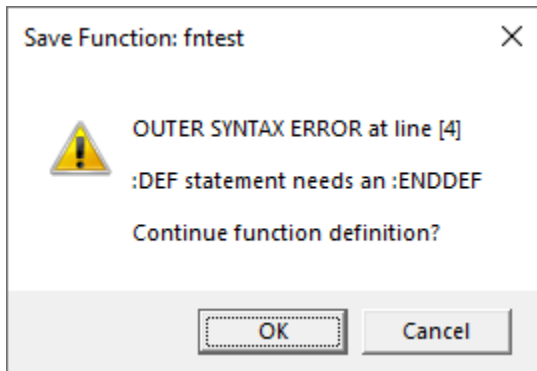
The Outer Syntax Error message was improved for a case with an ill-formed :if clause without an argument in a local inner function

E.g.,

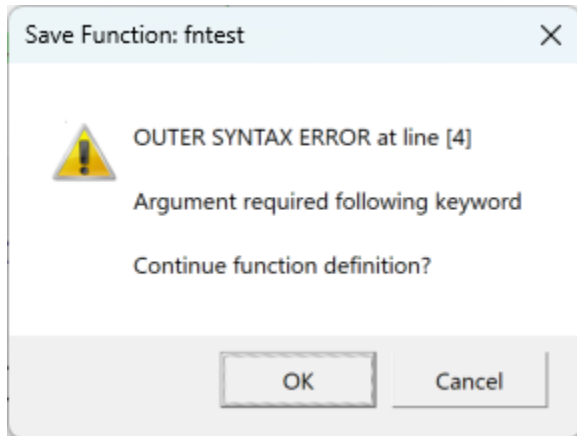


```
0  z←fntest x
1  "Testing OUTER SYNTAX ERROR"
2  :def z←fninnertest x
3  |
4  |   :if
5  |   :endif
6  |   z←200
7  | :enddef
8  | z←fninnertest x
9  | z
24  )ed fntest
25
```

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

There are several key notable differences in the output in APL64 from APL+Win:

1. Output from any traced statement will always be displayed on the next line following the *function[line#]* prefix
2. Branch, scalar, and vector outputs will not be indented
3. Rather than displaying diamonds as prefixes for secondary statements on diamondized lines, the output will instead display as *function[line#.part]* such as `GO[3.1]`

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

The user-selected sizes and locations for the Open Object and Fetch Object dialogs are retained

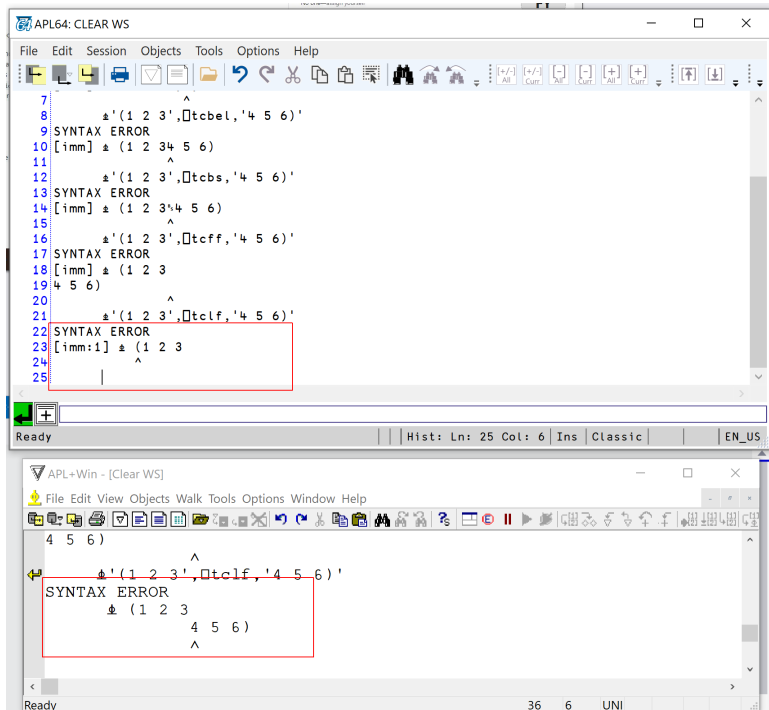
APL64 now records the user-selected display sizes and locations when these dialogs are closed to be reused the next time they're opened.

Performance improvements for `⎕DLB`, `⎕DLTB`, `⎕DTB`, and `⎕ROWFIND`

These system functions were updated to provide improved performance in APL64.

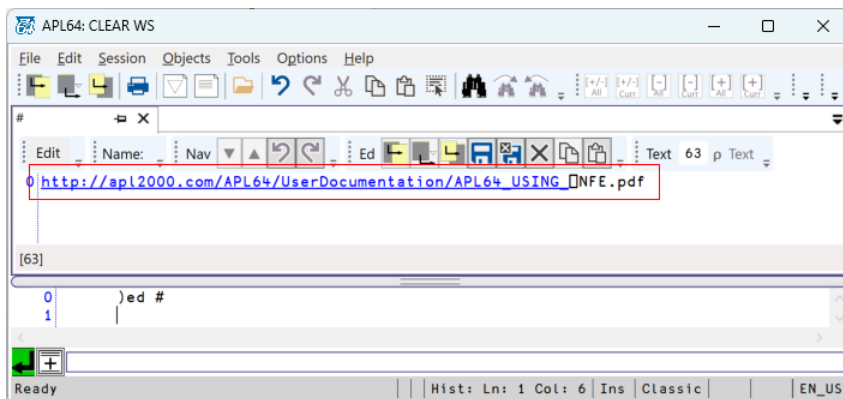
Bug Fixes

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



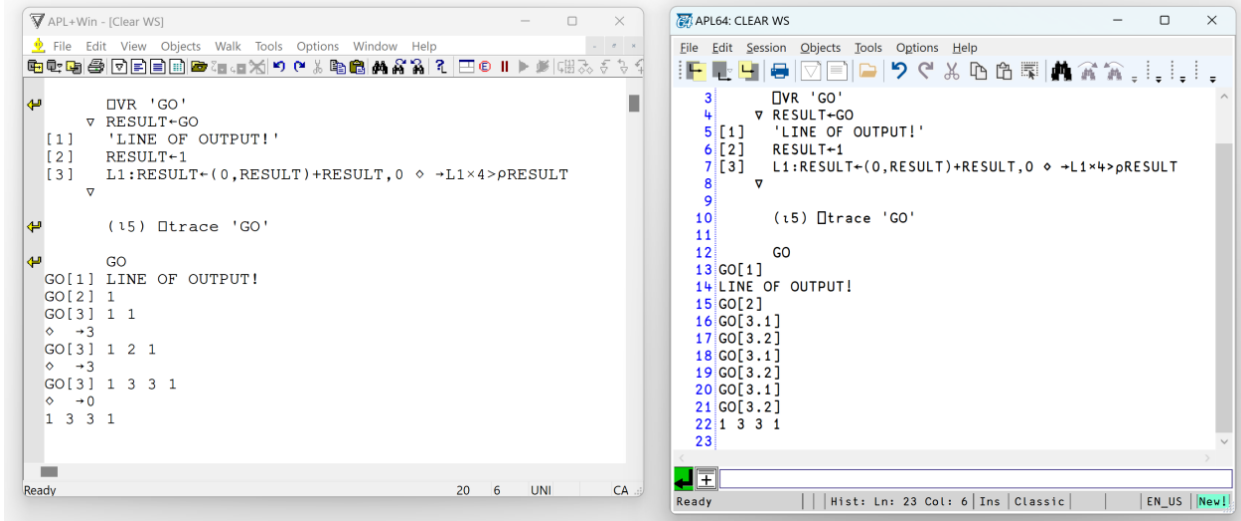
The auto underlining of strings with leading `http` in the editor stopped at the Quad (`⊞`) character

E.g.,



The `TRACE` output in APL64 was different and didn't display the value of the calculation on the traced lines like APL+Win

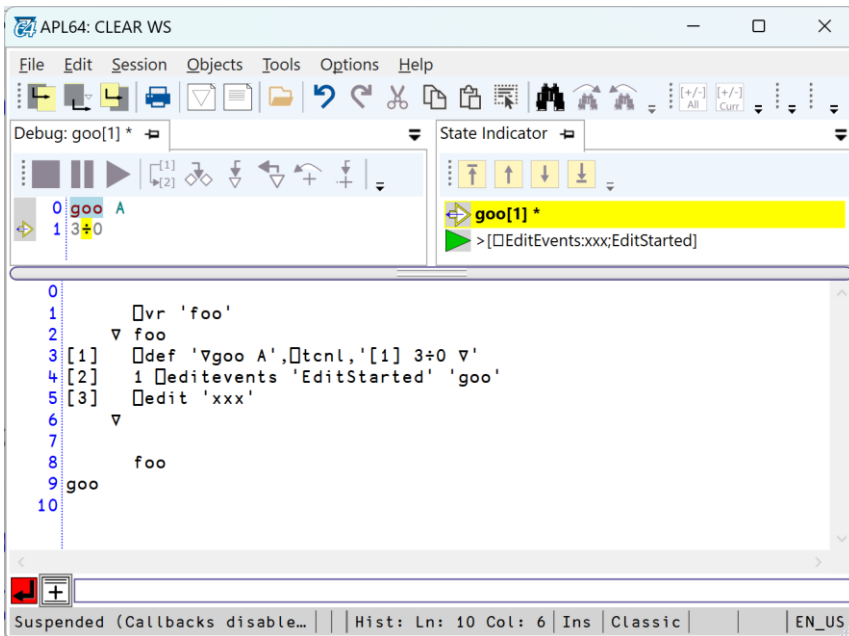
E.g.,



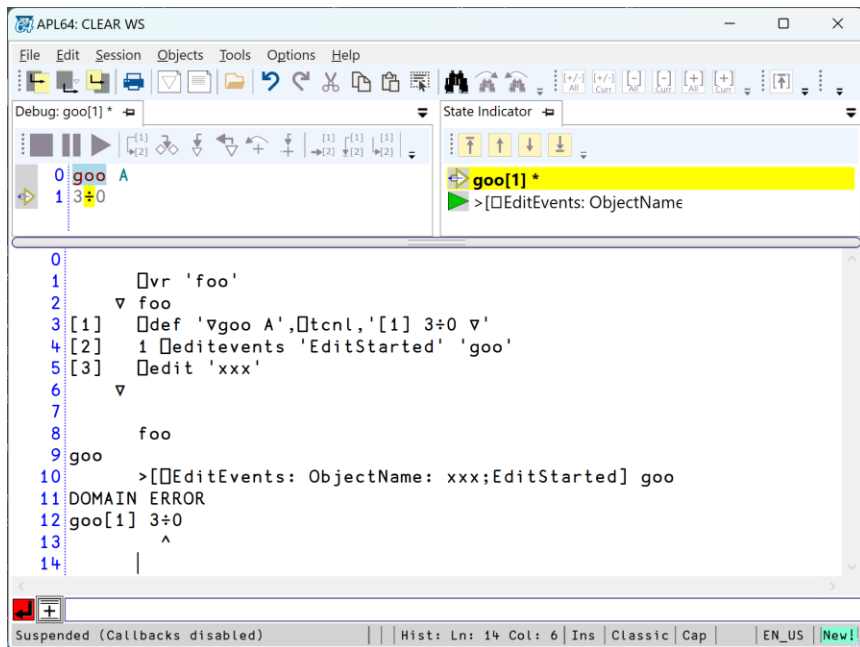
A Quote-quad input could be repeated in the history when in the editable classic history

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

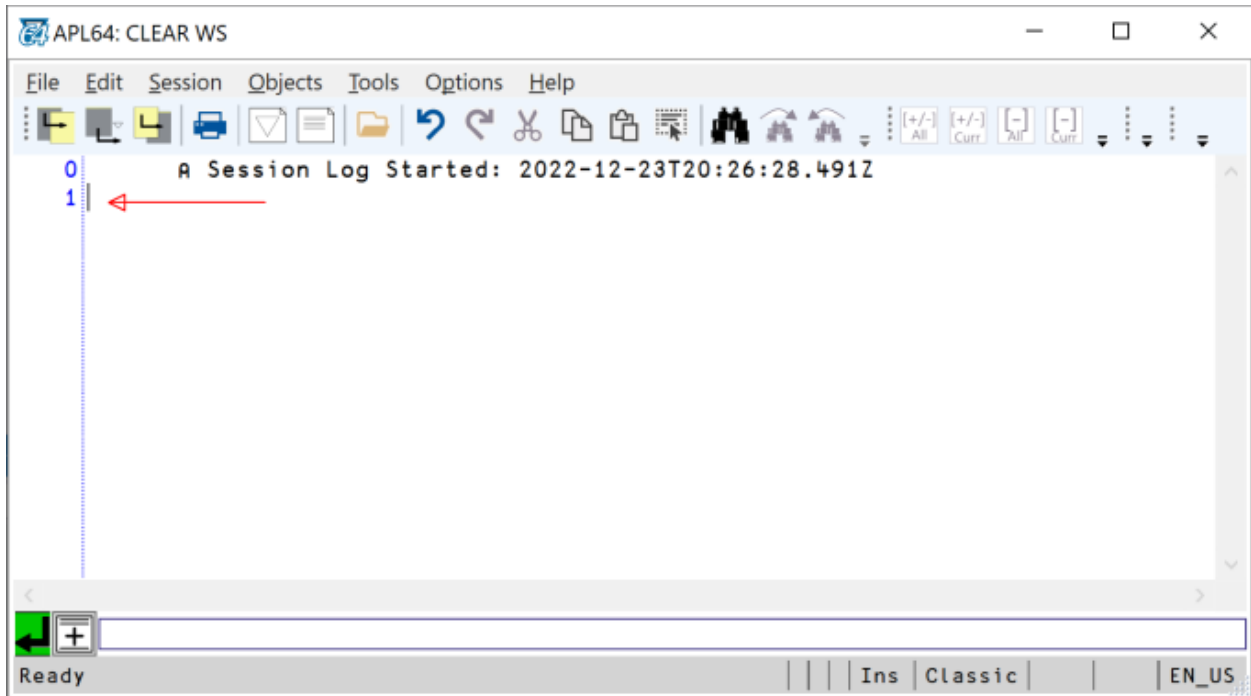
With the **Session | Set PW based on Window Width** menu enabled, resizing the session window then querying the value of PW did report the new value.

)reset or)sic failed to reset the stack (state indicator) after suspending the workspace

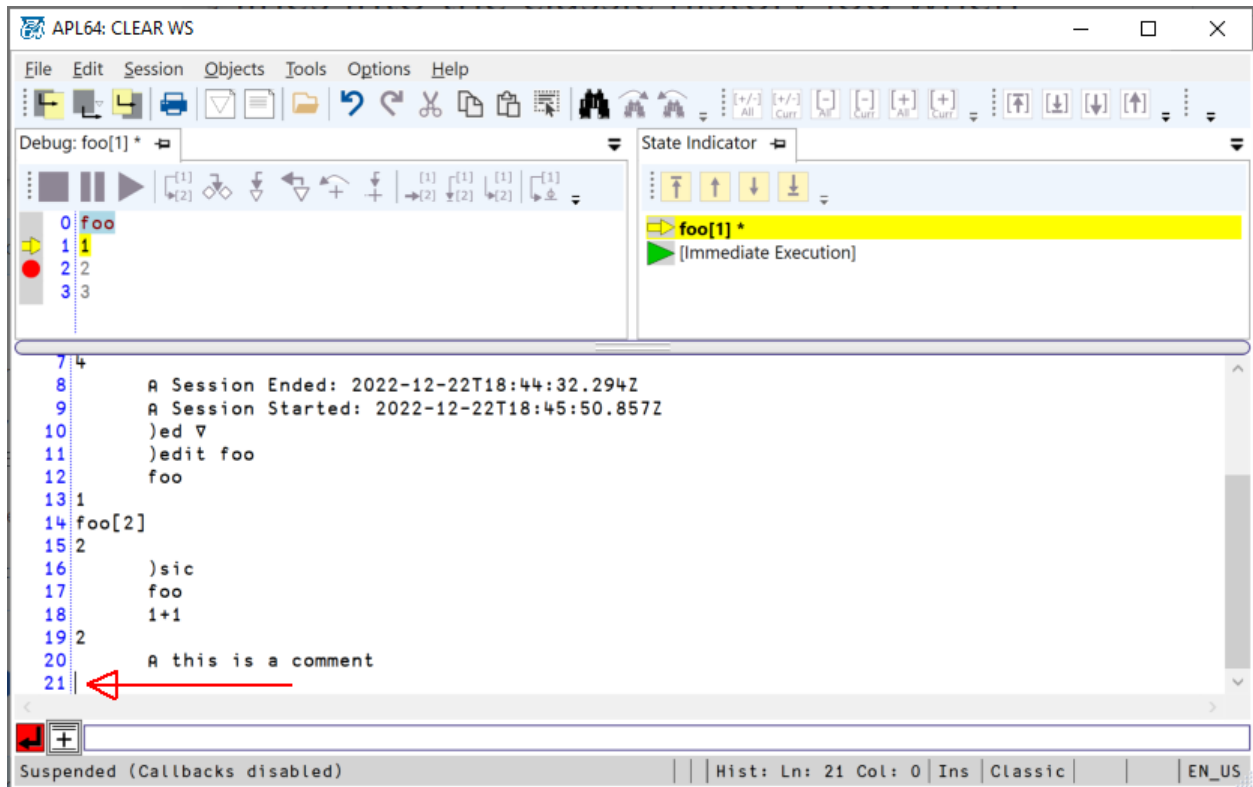
Addressed multiple scenarios that would prevent the)reset and)sic commands from clearing the state indicator after a debug suspension in the workspace that resulted in wi forms from responding.

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 has 64 or more rows

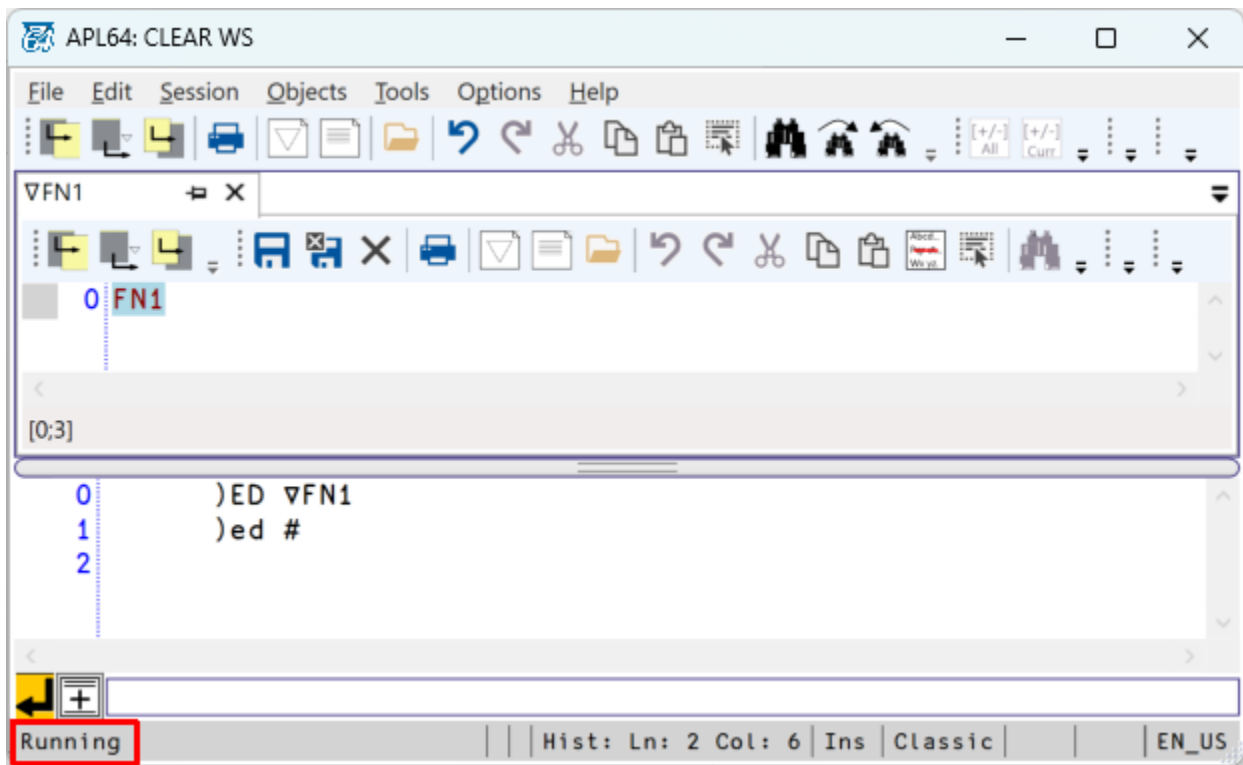
Grade-up () and grade-down () with Booleans returned incorrect results

MF of a function with a :returnif clause in a :case block returned a SYSTEM ERROR

Saving a function with system functions in the header row didn't report an error on line 0

ρ NA 'xxx' returned the result 0 0 instead of 0

The interpreter status displayed "Running" when the initial function editor was created in a new APL64 session



The interpreter status displayed "Running" when loading some APL+Win workspaces from the **File | Load** menu

Every load of an APL+Win workspace caused an **APL32Now32 Runtime System (32-bit)** process to accumulate in the Windows Task Manager