

Advantages of APL64

Contents

Excellent Compatibility with APL+Win.....	9
Increased Workspace Size Up To the Available Workstation Memory	9
Increased Variable Size	9
Performance Optimizations of APL Primitives and System Functions.....	10
Unified Variable Editor To Review & Edit Simple, Homogeneous, Heterogeneous, and Nested Variables	10
Full Unicode Support.....	10
Multiple Formats to Display the History in the Developer Version.....	11
Classic.....	11
Sequential	12
Grouped	13
Separated.....	14
Additional Text Selection Options - Linear, Rectilinear and Row	14
Floating or Docked Panes – History, Function & Variable Editor, Debugger.....	16
Multi-row Executable APL Statements	17
Runtime Deployment Simplified – APL64 Application in a Single File.....	18
Context-Sensitive APL Developer Documentation	19
Optional Tooltips for Toolbars, Menus and Panes.....	21
New APL String Data Type Supported.....	23
String Definition	23
Defining a String Variable in APL64.....	23
New User-defined Tools Features.....	25
New Configure Host Error Mappings Dialog.....	26
Enhanced Find/Replace Dialog	27
□XL Exchange Data Between APL64 and Excel Worksheet Without Excel.exe.....	28
□EDSS Edit an APL Array in a Worksheet Editor.....	30
)EDSS Edit an APL Array in a Worksheet Editor	30
Find In Functions In Workspace Utility	30
Latent Function Stops	31
UTC Time Supported Universally in APL64	32

New menu Session > Configuration Settings > Apply Default Settings	32
<input type="checkbox"/> XML to Create, Edit, Read and Write XML-format Documents	34
APL64 Idioms Manager	34
Updates to Existing System Functions, Commands and Variables	36
Many New System Functions.....	37
Enhanced Dialogs.....	38
APL Statements History – More options than Gather action	38
History Pane – Optional Row Numbers	39
Colors Dialog	39
Library Definitions.....	40
History Log Enhanced	41
Keyboard Definition	42
Convert APL+Win Configuration File	43
APL64 Now Supports .NET Release 6.0 (x64).....	43
HttpClient Now Used	43
New <input type="checkbox"/> Path System Function	43
New <input type="checkbox"/> Pathcase System Variable.....	44
New Save As Xlxs Option in)EDSS/ <input type="checkbox"/> EDSS.....	44
New APL64 System Highlights Feature in the Status Bar.....	45
Insert/Replace Keyboard Mode Supported in APL64	47
Find in Functions in Workspace Utility	47
'Last Fix Date' Column Added to the Filtered List of Functions	47
New Option is Provided to "Show Function Definition(s) for Selected Row(s)" in the Function List	48
<input type="checkbox"/> devcap Set to an Empty Vector.....	49
The Find/Replace Tool Replaces the Find/Replace Dialog.....	50
Rendering Large Output to History and <input type="checkbox"/> xso System Variable.....	50
Insert APL Idiom Dialog is Now Not Modal.....	52
The Caption and hwndmain Properties for the <input type="checkbox"/> wi System Object (#) are Deprecated	53
New menu: Objects Editors Pane Format Shrink Editor Area When All Editors Are Floated	53
New menu: Objects Editors Pane Format Shrink Editor Area When All Editors Are Floated	53
The <input type="checkbox"/> wi Interface to the APLNow32.exe was Optimized to Provide Better Stability.....	53
New <input type="checkbox"/> NFE 'SHA512Hash' Action	54

Session Log Timestamps Displayed.....	54
New Options Session Log Delete All Session Log Files Action	54
Variable and Function Names Can Begin With the Underscore Character	55
New System Functions: <input type="checkbox"/> flock, <input type="checkbox"/> fflush, <input type="checkbox"/> nlock, <input type="checkbox"/> nflush.....	55
New System Functions <input type="checkbox"/> ask and <input type="checkbox"/> askn	55
Options History Log History Log Options Menu Renamed	56
<input type="checkbox"/> EDITEVENTS/ <input type="checkbox"/> EDITINFO and Documentation Updated.....	57
<input type="checkbox"/> SKD and Documentation Updated.....	57
Performance of <input type="checkbox"/> Rowfind System Function is Improved	57
The Display of the MessageBox Warning is Now Avoided When Importing an Empty Session Log File....	57
Scalar operations have been optimized in APL64.....	57
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)	57
GUI Tools for Command Line Applications: <input type="checkbox"/> ASKx system functions.....	57
New <input type="checkbox"/> XML actions: Descendants, NDescendant, DescendantsAndSelf, and NDescendantAndSelf.....	57
New <input type="checkbox"/> XL actions: SortRange and UsedRange	58
Updates to <input type="checkbox"/> EDSS and)EDSS	58
Updates to SaveAs options for xlsx, xls, csv and Unicode formats:.....	58
New Sort Rows options for a user-selected range.....	58
New Print Option for the current worksheet selection	58
New Find & Replace Options	58
New <input type="checkbox"/> EDITINFO action: DeleteEditEventsSubscriptions.....	58
<input type="checkbox"/> EDITEVENTS 'PreviewObjectNameSelection' event simplified	58
Quad (<input type="checkbox"/>) input behavior improved	58
Option to exclude the CommandLine from the Ctrl+Tab list in editable Classic History:	58
Toolbar images were added to some menu items in the File, Edit and Objects drop down menus.....	58
The Find/Replace Tool now closes when pressing the Esc key in the window	59
New option added to download all online documents to a target folder on the local machine	59
The Outer Syntax Error message was improved for a case with an ill-formed :if clause without an argument in a local inner function.....	59
The description for the <input type="checkbox"/> trace system function was added to the System Functions document.....	59
New response code (-7) for <input type="checkbox"/> copy and <input type="checkbox"/> pcopy system functions.....	59

Additional information about creating and editing a user-defined function was added to the Help APL Language User-defined Functions document	59
The user-selected sizes and locations for the Open Object and Fetch Object dialogs are retained	59
Performance improvements for <input type="checkbox"/> DLB, <input type="checkbox"/> DLTB, <input type="checkbox"/> DTB, and <input type="checkbox"/> ROWFIND.....	60
APL64 Installer Enhancements	60
Create a Cross Platform Component Utility.....	60
New debugging with Watchpoints dialog.....	61
New <input type="checkbox"/> watchpoint System Variable	62
Syntax coloring and margin glyphs supported in editable classic history format	62
New Inline Comments: <input type="checkbox"/> <input type="checkbox"/>	63
New Ctrl+F3 keyboard shortcut to search for a token under the caret	65
[executed] statement history dialog.....	65
New Include Imported Exec Stmt option in the Executed Statements History dialog.....	65
New options available for searching the APL64 user documentation.....	66
Find/Replace dialog button tooltips updated to clarify their intended purpose	66
The Configure User Tools dialog Listview now supports Delete key actions.....	67
Restart immediately when the OK button is clicked in the Apply Defaults Settings dialog	67
Editing large functions performance improved.....	68
Session Log loading performance improved.....	68
)EDSS and <input type="checkbox"/> EDSS Improvements.....	68
New Find and Replace dialog.....	68
New Import worksheet and Export data to variable actions.....	69
New Undo and Redo actions.....	69
New context menu items implemented	70
Use Autofit for worksheet rows.....	71
EDSS Limits.....	71
Session Separated History Format menu: Tooltips updated	71
<input type="checkbox"/> SQLx system functions Updated	71
<input type="checkbox"/> NFE Enhanced	71
Improved exception messages for 'Read' and 'Write' actions.....	71
<input type="checkbox"/> MATHNET interface to MathNet linear algebra toolkit implemented	71
Successfully load previously damaged workspaces.....	71

Correctly reconstruct the SI stack when loading suspended or damaged workspaces.....	71
<input type="checkbox"/> wi Improvements	71
APL64 Colossal file functions were ported to APL64 from APLNow32.....	72
<input type="checkbox"/> tt32 System Function: High-Resolution Timer Tick Count for APLNow32 operations.....	72
<input type="checkbox"/> ZIP system function.....	72
System variable <input type="checkbox"/> SYS[21] value is 1 when the APL64 is runtime.....	72
)RSAVE system command now reports DEPRECATED FEATURE.....	72
<input type="checkbox"/> ROWFIND performance improved when the left and right arguments are in <input type="checkbox"/> AV.....	72
<input type="checkbox"/> SKD: ToFile/FromFile actions removed.....	72
<input type="checkbox"/> STRING updated.....	73
<input type="checkbox"/> WA operates faster.....	73
<input type="checkbox"/> ARBOUT and <input type="checkbox"/> PR system variables now report DEPRECATED FEATURE.....	73
<input type="checkbox"/> nfe 'delete' action exception message improved	74
Option to Hide APL Command Line	74
Option to suppress callback information in the history pane	76
Cancel option for Quad-input, Quote-quad input, and <input type="checkbox"/> ASKxx dialogs	76
Bookmarks are supported in the Editable Classic History Format.....	78
The editable classic History supports the Session Use Selected Text on Enter in History menu.....	78
'Find Next Bookmark' keyboard shortcut modified to Alt+F2 from F2.....	78
Line number display in the history pane and object editors have been enhanced.....	79
Find next/previous executed line options now supported in editable Classic History format.....	79
Executed Statements History (Ctrl+F9) window updated	80
Find tool and search keyboard shortcuts for Token and Word searches improved	80
Objects Row Numbers menus improved	81
Objects Tool Bars menus improved	82
Update to Fetch Object tool to support large matrices.....	82
New Session Suppress Unassigned History Output From Quote-Quad Output menu	82
New Session Classic History Format Refresh For Accurate History menu	83
<input type="checkbox"/> CSE documentation consolidated	83
Digital code signing available for an APL64 Windows Runtime Exe	83
Materialize additional files, if any, to the path where the WRE exe or CPC Nuget package is installed, rather than the path where the WRE or CPC dll is running	84

Materialize the APLNow32 ini and adf files, if included in the WRE, to the path where the WRE dll is running.....	84
The 'Add Folder of Required Files' button to add files in a folder and all sub-folders in the Additional Files Required for the Application area	84
The 'Remove All Required Files' button to remove all files listed in the Additional Files Required for the Application area	84
<input type="checkbox"/> CPC documentation consolidated.....	85
SCAN function updated.....	85
Arithmetic Functions Performance Improved for Small Arrays.....	85
New <input type="checkbox"/> AES system function: Encryption and Decryption	85
New <input type="checkbox"/> BROTLI system function: A better alternative to gzip inflate/deflate	86
New <input type="checkbox"/> XML methods: JsonToXml and XmlToJson methods	86
Performance Improved for Colossal Component Files	86
APL64 session responded to the Insert/Overwrite keyboard status changed in other applications	86
The CapsLock and NumLock keyboard status in the APL64 session could be out of sync with Windows .	86
Bookmarks (ctrl+F2) did not work in a new editable classic history session	86
Name validations for 'Runtime Executable Name', 'Company Name' and 'Copyright' fields.....	86
The Digital signature command and digital signature checkbox are on the same line in the utility.....	87
Entry added for the APLNow32.exe.manifest file*.....	88
Saving and Retrieving Unicode Data in APL64	88
Use an APL64 CPC in Excel	88
APL64 2025.0.1 Update	89
Interpreter Modifictions	89
Colossal File System: Significant performance improvements reading components.....	89
Colossal File System: New File Open Modes	89
XML Serialization Exception Message Format	89
<input type="checkbox"/> XL: New Actions to Read and Write CSV-format Files	89
<input type="checkbox"/> FnEdHist System Function.....	89
<input type="checkbox"/> NewLine and <input type="checkbox"/> SetNewLines System Functions.....	89
<input type="checkbox"/> LCASE and <input type="checkbox"/> UCASE system functions updated to support Unicode characters	89
<input type="checkbox"/> SYSINIT[2] (2nd element) returns the .Net Version targeted by APL64 instance	90
<input type="checkbox"/> STRING 'ToLower' and 'ToUpper' actions enhanced to support 'culture'	90
<input type="checkbox"/> VR system function enhanced to specify the format of the result.....	91

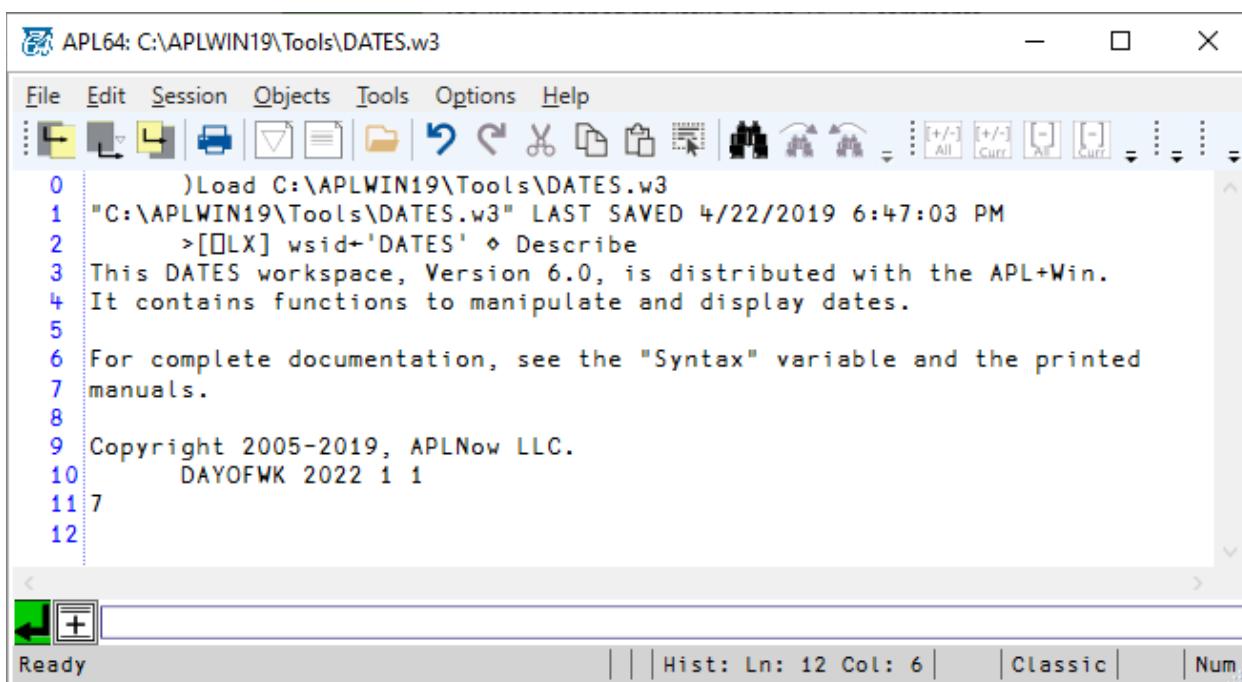
APL64 Developer Version GUI Modifications	92
New Function Editor Compare Format.....	92
New Function Editing Changes Records	93
Documentation of the Options Keyboard Definition Dialog Updated.....	94
Find/Replace Tool Updates.....	94
Set Initial Caret Marker Position when Editing a New Function.....	94
WRE and CPC Utility.....	94
New APL64 WRE and Overview documentation	95
Update to APL64 CPC in a REST Service Example (APL64CpcRestSvc.zip)	95
Update to APL64 CPC in WPF example targets .Net 8 Windows Presentation Foundation (WPF)	95
Update to APL64 CPC in Excel example targets .Net 8 Windows Presentation Foundation (WPF)	95
New APL64 CPC using .NET MAUI example	95
New APL64 CPC in an Azure Function example	95
Line Termination for Exception Messages.....	95
Miscellaneous	96
APL64 system and installer updated to support .Net 8	96
:IFTEST control structure can be closed with either :ENDIF or :ENDIFTEST or :END:	96
□XL,)EDSS and □EDSS: SpreadsheetGear Nuget package updated from v9.2.44 to 9.3.23.....	96
□SQLite: Microsoft.Data.Sqlite client software Nuget package updated from v8.0.6 to v8.0.8	96
Pdf document generation component updated: ceTe v12.23.0 => 12.25.0.....	96
APL64 2025.0.5 Update	96
Interpreter Modifications	96
Primitive functions with performance improvements	96
System functions with performance improvements and/or updated programmer documentation	97
New □TEXTREPL argument.....	97
New □XL actions: FromAPLToRange, ToAPLFromRange, Autofit and NumberFormatRanges	97
□SSASSIGN programmer documentation syntax and example updated	97
□SSCAT programmer documentation example updated	97
□SYSINIT Additional Values: EnvVars, MachineName and Processor count.....	97
APL64 Developer GUI Modifications.....	98
□EDSS: New Import and Export options from/to an APL variable	98
□EDSS: New option to print the Worksheet or Current Selection in the Worksheet	99

☐ EDSS: Active worksheet tab is more prominent with a white background	99
☐ EDSS: When there is only one worksheet, its tab will now be visible	100
Consistent Borders in the APL64 Developer GUI	100
Find/Replace Tool Performance Improvements.....	102
The Command Line, when hidden, is removed as an item in the Active Panes List.....	102
Function and Variable Editor Pane Tabs include Tool tips indicating the Object Name	102
The exception message was improved for a canceled print task	102
Print Updates	103
Print dialog: The Pages option in the Page Range section is enabled	103
A heading is printed on each printed page.....	104
New Documentation Timeout option in the Documentation Options window	104
WRE and CPC Utility	105
Cross-platform Component in Azure On-demand Function: documentation & example improved	105
New CPC Example: Using CPC in a Webserver with an HTML Browser-based GUI	105
Adding an entire folder of Additional Files for the Application is improved	105
Miscellaneous	105
Additional fields added to System Information document in Help About APL64 menu:.....	105
Programmer documentation Updated – APL64 Compatibility with APL+Win	106
APL64 2025.0.8 Update	106
Interpreter Modifications	106
☐ XL: New actions: AddWorksheet , CreateWorkbook, DeleteWorksheet & WsCount	106
☐ PROFILE: APL64 system version number and date/time (UTC) info added to the result	107
Developer Version GUI Modifications	107
Updates to ☐ EDSS and)EDSS: Formula Bar and related user documentation	107
Function Editor: Syntax coloring supported for lines with more than 9600 characters.....	108
Function Editor: Commit Changes and Commit & Close buttons added.....	108
Function Editor: Toolbar icon for the traditional editor format	109
Decimal Unicode 124 and 8739 are now available on EN-US and EN-UK keyboards.....	110
A statement executed in the History pane no longer moves the focus to the Debug Pane	111
Editors and Debug/SI pane horizontal splitter bar values are now saved.....	111
Main Window Toolbar Height no longer changes	111
Edit Add Inline Comment and Edit Delete Inline Comment Submenus	111

WRE and CPC Utility Modifications.....	111
WRE creation option to exclude the .Net Runtime	111
Option to commit function editing history within the CPC and WRE utilities.....	113
<input type="checkbox"/> CSR Extra new line character in the result removed	114
<input checked="" type="checkbox"/> CSR: 'XStmtEndMethod' output prefix removed for 'head' option.....	116

Excellent Compatibility with APL+Win

Load and run an APL+Win workspace in APL64 in many cases with minimal or no modification required. The Help > APL+Win Compatibility menu item describes this feature.



Increased Workspace Size Up To the Available Workstation Memory

APL64 – Maximum workspace size up to available workstation memory permitting the creation and processing of multiple large APL variables.

APL+Win - Maximum workspace size less than 3.7 GB.

Increased Variable Size

APL64 - Maximum homogeneous variable size up to 2,146,435,071 elements in a variable. Nested arrays may recursively contain up to this number of elements at each nesting level.

Type	Max GB
Boolean	2

Integer	8
Double	16
Unicode Char	4
APL+Win Char	2

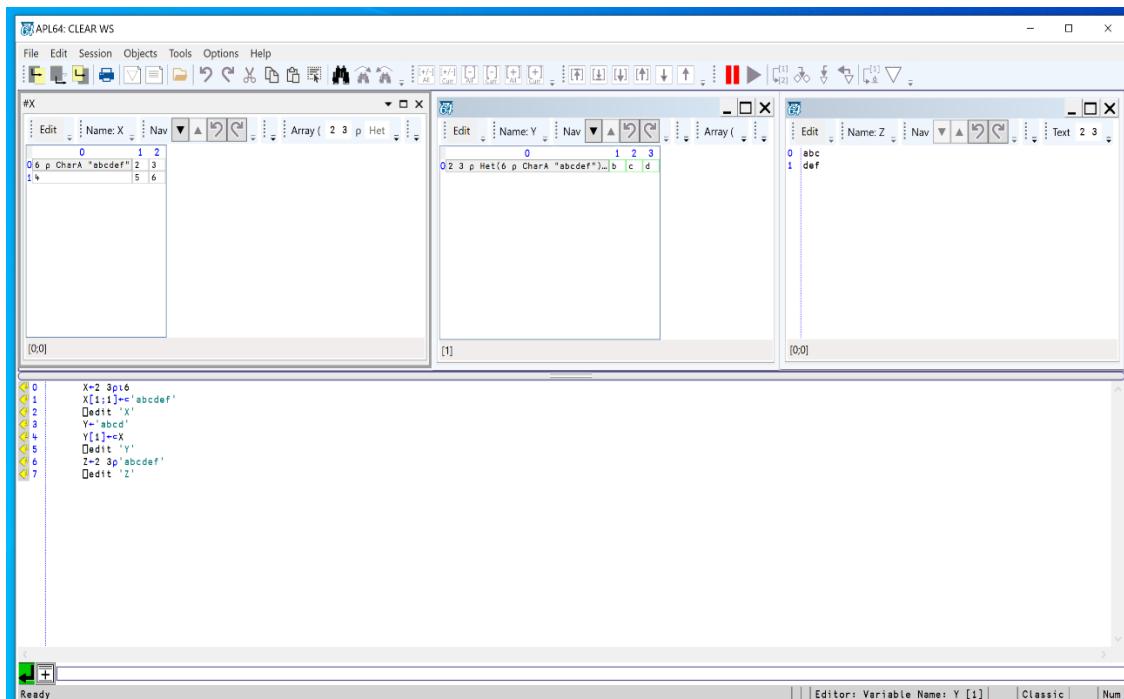
APLWin - Maximum homogeneous variable size less than 2 GB and in practice 500 MB or less.

Performance Optimizations of APL Primitives and System Functions

After an APL64 Project primitive or system function is implemented and tested, it is considered for performance optimization. The optimization process requires considerable effort and skill. These optimizations have yielded significant performance improvements. Click [here](#) to view graphs comparing APL64 performance to APL+Win in many areas. Performance optimizations are an ongoing process.

Unified Variable Editor To Review & Edit Simple, Homogeneous, Heterogeneous, and Nested Variables

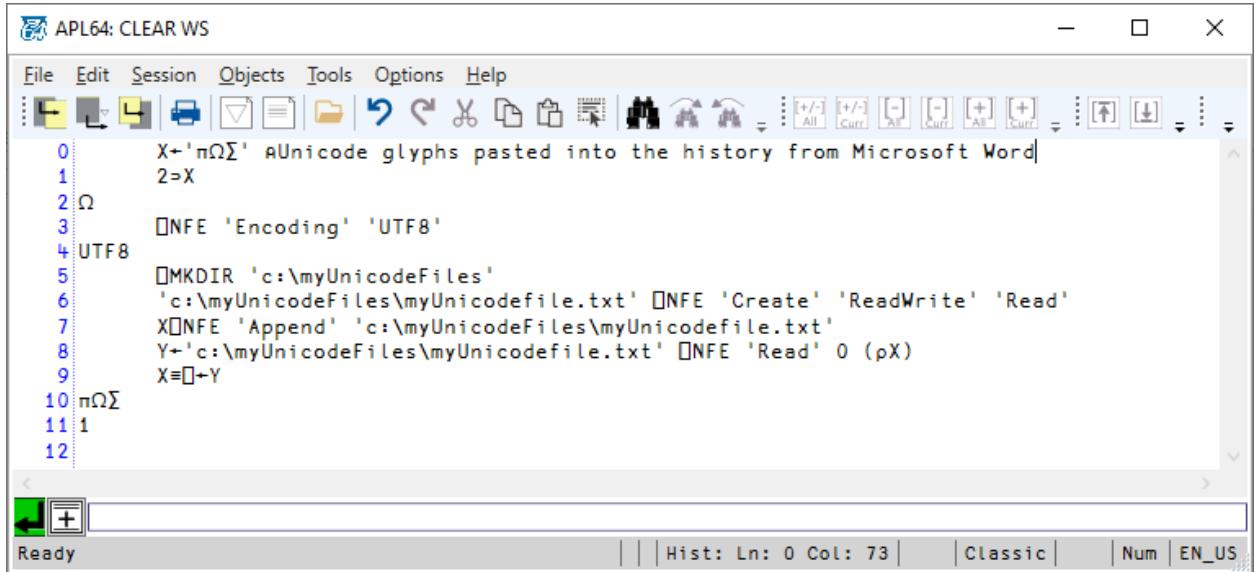
The new unified variable editor in APL64 can review and edit an APL variable of any type, including heterogeneous and nested variables. The APL+Win graphical editor is limited to simple text or simple numeric variable editing.



Full Unicode Support

Transparent support for Unicode characters in character scalars, vectors, matrices and arrays throughout the system. For compatibility AV has the same content as APL+Win.

Unicode glyphs in APL64 variables or functions will be directly displayed in the APL64 Developer version. Use the `⎕NFE` system function to read/write Unicode data in native files.



The screenshot shows the APL64 Developer application window titled "APL64: CLEAR WS". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Undo/Redo. The history pane displays the following APL session log:

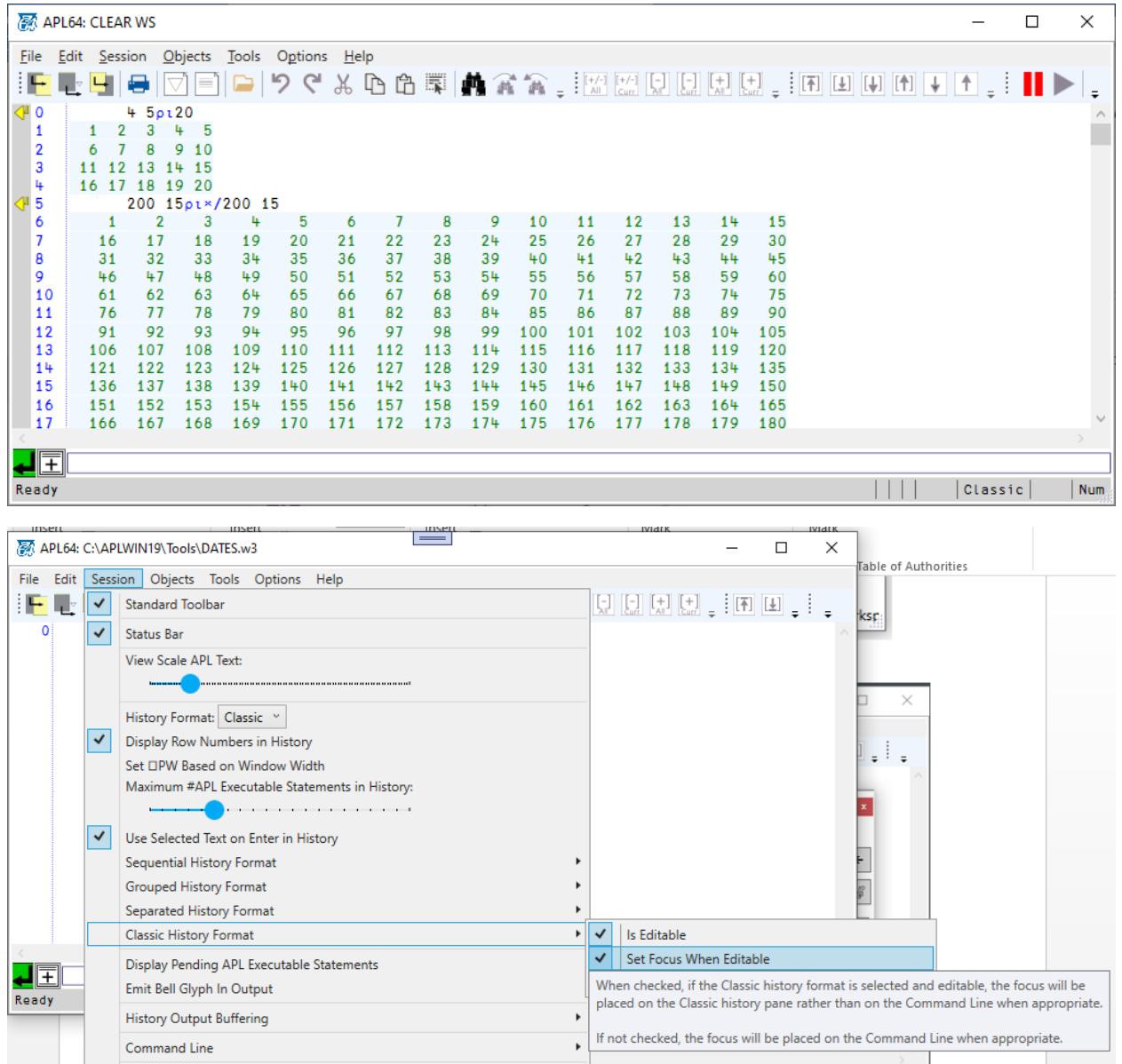
```
0 X←'πΩΣ' ⍝ Unicode glyphs pasted into the history from Microsoft Word
1 2=X
2 Ω
3 ⎕NFE 'Encoding' 'UTF8'
4 ⎕UTF8
5 ⎕MKDIR 'c:\myUnicodeFiles'
6 'c:\myUnicodeFiles\myUnicodefile.txt' ⎕NFE 'Create' 'ReadWrite' 'Read'
7 ⎕NFE 'Append' 'c:\myUnicodeFiles\myUnicodefile.txt'
8 Y←'c:\myUnicodeFiles\myUnicodefile.txt' ⎕NFE 'Read' 0 (⍴X)
9 X=⍴Y
10 πΩΣ
11 1
12
```

The status bar at the bottom shows "Ready", "Hist: Ln: 0 Col: 73", "Classic", "Num", and "EN_US".

Multiple Formats to Display the History in the Developer Version

Four formats are available for the history pane in the APL64 developer version which may be selected at any time during an APL64 instance to illustrate previously-executed APL statements and resulting interpreter output.

Classic – This format is analogous to the APL+Win ‘session’ document-style format with the information in execution order. The APL64 classic history format has an ‘Is Editable’ option. When Session > History Format: Classic, Session > Classic History Format > Is Editable is checked and Session > Classic History Format > Focus When Editable is checked, the Classic history format behavior will be very similar to the APL+Win history. This is the default setting.



Sequential – Developer input and interpreter output is displayed as an array of rows.

The screenshot shows a 15x15 matrix of sequential integers from 1 to 225, displayed in green text on a white background. The matrix is defined by the following APL code:

```
4 5 pt 20  
1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
200 15 pt ×/ 200 15  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30  
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45  
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60  
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75  
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
91 92 93 94 95 96 97 98 99 100 101 102 103 104 105  
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120  
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135  
136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
```

Grouped – The executed APL statements are displayed as an array of rows. The interpreter output can be displayed as an array of rows or in document-style. When the interpreter output is extensive, it is presented as a scrollable, user-specified-height text block. The results text block will be scrolled to the first or last result line depending on user preference.

Separated – The executed APL statements are displayed as an array of rows. The interpreter output for a user-specified executed APL statement can be displayed as an array of rows or in document-style.

	1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9	10
1	16	17	18	19	20	21	22	23	24	25
2	31	32	33	34	35	36	37	38	39	40
3	46	47	48	49	50	51	52	53	54	55
4	61	62	63	64	65	66	67	68	69	70
5	76	77	78	79	80	81	82	83	84	85
6	91	92	93	94	95	96	97	98	99	100
7	106	107	108	109	110	111	112	113	114	115

Additional Text Selection Options - Linear, Rectilinear and Row

The information in the function and variable editors, the debugged function and some user-selected history formats is presented in document style which supports linear, contiguous selection and rectilinear selection.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	5pt20													
1	1	2	3	4	5										
2	6	7	8	9	10										
3	11	12	13	14	15										
4	16	17	18	19	20										
5	200	15pt*x/200	15												
6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
8	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
9	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
10	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
11	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
12	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
13	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
14	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
15	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
16	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
17	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

0 4 5pt20
1 1 2 3 4 5
2 6 7 8 9 10
3 11 12 13 14 15
4 16 17 18 19 20
5 200 15pt×/200 15
6 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
8 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
9 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
10 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
11 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
12 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
13 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
14 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
15 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
16 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
17 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180

Ready | Hist: Ln: 3 Col: 12 | Classic | Num

Some user-selected history formats present the information in row-style supporting linear, contiguous selection and rectilinear selection within a row and non-contiguous whole row selections.

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

0 4 5pt20
1 1 2 3 4 5
2 6 7 8 9 10
3 11 12 13 14 15
4 16 17 18 19 20
5 200 15pt×/200 15
6 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
8 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
9 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
10 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
11 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
12 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
13 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
14 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
15 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150

Ready | Hist: Result: Stmt#6 Col: 41 | Sequential | Num

The screenshot shows the APL64: CLEAR WS application window. The main pane displays a 15x15 matrix of integers from 1 to 225. The matrix is defined by the following APL code at the top of the workspace:

```

0 4 5 p 20
1 1 2 3 4 5
2 6 7 8 9 10
3 11 12 13 14 15
4 16 17 18 19 20
5 200 15 p 15*200 15
6 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
7 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
8 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
9 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
10 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
11 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
12 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
13 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
14 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
15 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150

```

The status bar at the bottom indicates "Ready".

Floating or Docked Panes – History, Function & Variable Editor, Debugger

The variable editor and user-defined function editor panes may be floated separately from the main window.

The screenshot shows the APL64 interface with two floating panes. On the left is the "Function Editor" (FEN1) pane, which contains the following APL code:

```

#X
Edit Name: X Nav Ed Array ( 2 3 p Int )
0 1 2
0 1 2 3
1 4 5 6
[0:0]
0 X-2 3 p 6
1 ⌊def 'FN1' '1234' '5678'
2 FN1
3 )ed X
4 )ed FN1
5

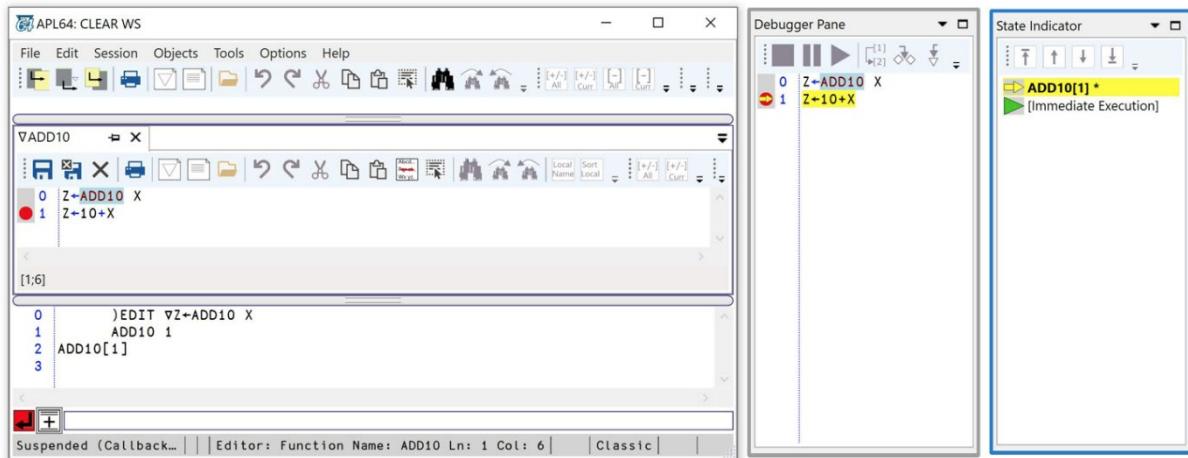
```

On the right is the "Variable Editor" (VFN1) pane, which shows the state of variable X:

Index	Value
0	FN1
1	1234
2	5678

The status bar at the bottom indicates "Editor: Function Name: FN1 Ln: 0 Col: 0".

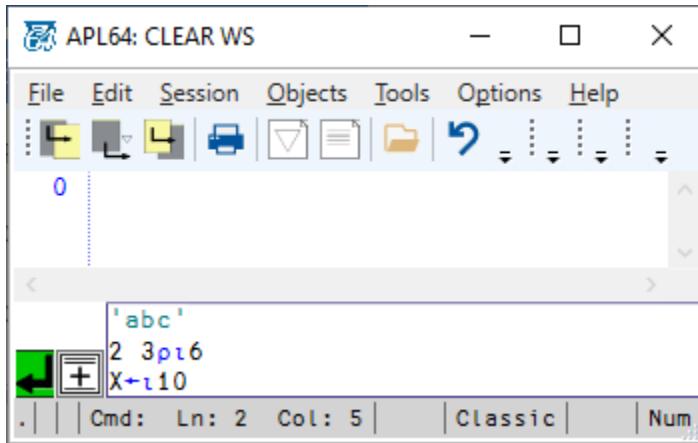
The debugged function pane and the state indicator panes may now be independently ‘floated’ separately from the main window:



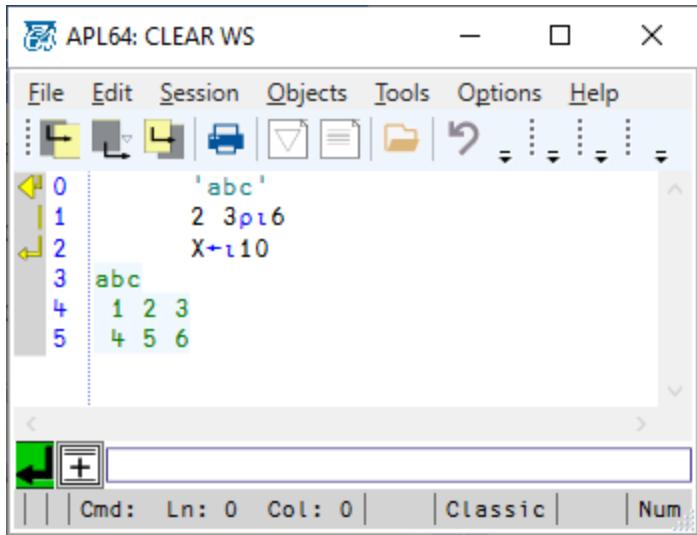
Multi-row Executable APL Statements

Often it is convenient to associate multiple APL executable statements into a ‘script’ which is executed *en masse*. A new line character is used to separate the rows of a multi-line executable APL statement to improve readability.

Multi-row APL executable statements may be manually entered or pasted into the APL64 Command Line.



Once executed, multi-row APL executable statements are identified in the history pane as a unit.



Runtime Deployment Simplified – APL64 Application in a Single File

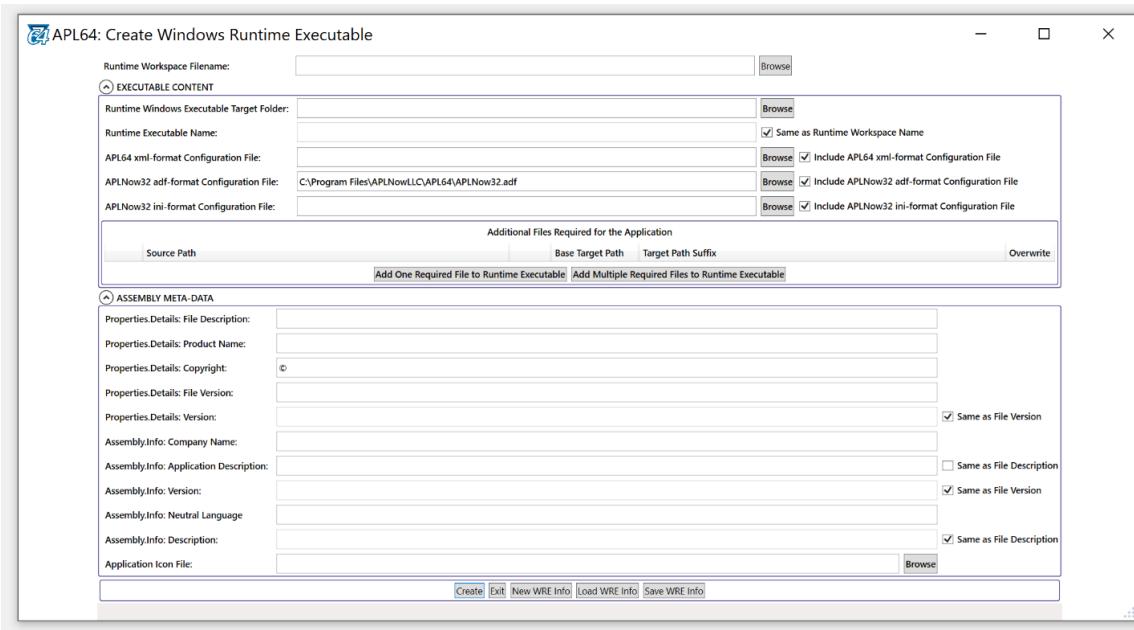
There is no need for an APL64 runtime version or workspace in APL64. In APL64 the Options | Create Runtime .Net Assembly | Create Windows Runtime Executable utility will package application workspaces and other supporting application files into a single executable file which includes the APL64 engine.

All features of APL64 which are available in the Windows operating system environment may be used by the APL64 programmer in the development of an APL64 runtime application.

The APL64 programmer creates the main APL64 workspace containing an `LX` to start the application. This APL64 workspace and other application-specific files are combined with APL64 engine into a single, easy to deploy ‘runtime’ file containing:

- APL64 runtime component
- APL64 xml-format configuration file
- APL programmer-created, runtime-ready APL64 workspace
- .Net components for the Windows operating system
- APLNow32 components to support the Win32 features of APL64 including `WI`, `WCALL`, `NI`
- Other APL programmer-provided application-specific files

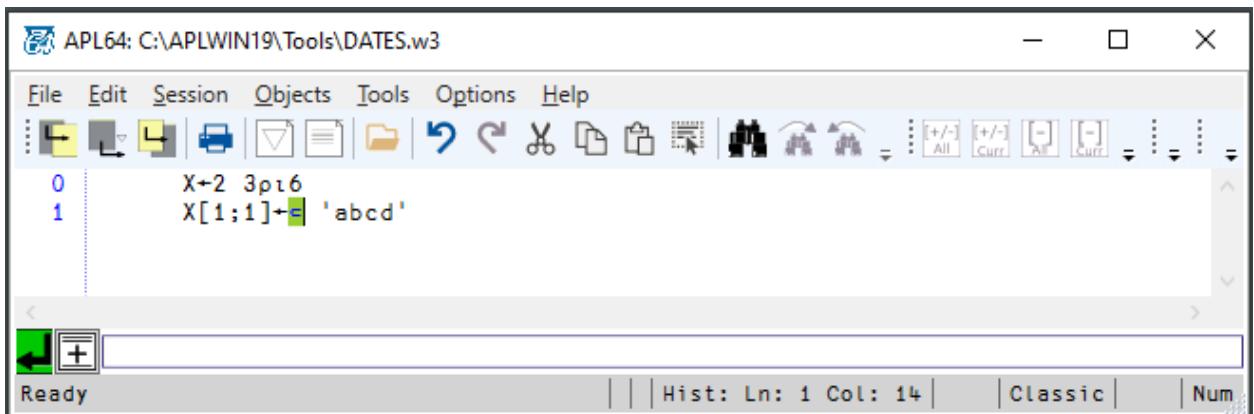
The published APL64 runtime executable is self-contained, including all of the necessary components of a complete APL64-based Windows application, so that it can be run via a double left mouse click or incorporated into a Windows program short-cut.



Context-Sensitive APL Developer Documentation

Context-sensitive documentation for APL primitive functions, system variables, functions and commands has been implemented in APL64.

When the cursor is placed on an element in the history pane, function editor or debugger and the F1 key is clicked, the applicable documentation for that element will be displayed in the user's default web browser. The pdf-format documentation supports bookmarks and table of contents containing APL glyphs and APL glyph names. This feature is new to APL64 and is not available in APL+Win.



The screenshot shows a web browser window with the title "APL64 API_PRIMITIVE.pdf". The URL is "api2000.com/APL64/UserDocumentation/APL64_API_PRIMITIVE.pdf#page=23". The page displays the "Table of Contents" on the left and the detailed documentation for the "Enclose" function on the right.

Table of Contents:

- Basic Information
- Numeric Functions
- Boolean Functions
- Structural Functions
 - Catenate
 - ⍳ Catenate
 - ⍣ Enclose
 - ⍣ Partition
 - ⍣ Disclose
 - ⍣ Pick
 - ↑ First
 - ↑ Take
 - ↓ Drop
 - ⍢ Unique
 - ⍢ Shape
 - ⍢ Reshape
 - ⍢ Transpose
 - ⍥ Horizontal Rotate
 - ⍥ Vertical Rotate
 - ⍋ Indexing
 - ⍋ Functions that Return an Index
 - ⍋ Miscellaneous Functions
 - ⍥ Operators
 - ⍥ Other Primitive Symbols
 - ⍥ Other Syntax Symbols
 - ⍥ Random Number Generator

Documentation for ⍣ Enclose:

Note 1: larg ⍴ rarg
larg ⍴[i] rarg (where i is pprarg and ⍵ io is 1)

Note 2: res ⍵← rarg res ⍵← res ⍴ rarg
The syntax is similar to the in-place operations in other programming languages like C++ and C#. However, in APL64, besides the shorter notation, this can also provide significant performance gains particularly in cases that involve repetitive catenations with large arrays such as inside an iterative control structure (a :for loop).

Syntax:
res ⍵← ⍣ arg
res ⍵← ⍣[i] arg

Description:
Create a nested scalar from any array that is not a simple scalar.
arg: any array.
i: non-negative integer scalar or vector that indicates the axes of the arguments to be enclosed.

When you use Enclose with axis, you can specify some or all of the axes of an array, yielding a result of lower rank than the right argument that contains the same data. Enclose with Axis is similar to ⍵ split but is more general in function.

Example:

```

C←'A' 'MM' 'SSS'
pC
3
C[2]←⍤2 2 p14
C
A 1 2 SSS
3 4
)DISPLAY C
-----
| .---. .---.
| A +1 2 | SSS |
| - | 3 +| '---'
| '---' |
| e-----|
-----
A←3 4p12
c[1]A
1 5 9 2 6 10 3 7 11 4 8 12
c[2]A
1 2 3 4 5 6 7 8 9 10 11 12
(c[1 2]A)≡cA
1
  
```

The Help > Documentation Topic History list is maintained during an APL64 instance. Double left click on a user-selected list item will open the associated document in the user's default web browser.

The screenshot shows the APL64 developer version interface. On the left is a vertical sidebar with several links:

- APL Language
- Developer Version GUI
- Command Line Arguments
- APL+Win Compatibility
- APL64 Project History
- Product Support
- End User License Agreement (EULA)
- About APL64

The main pane contains a table titled "Documentation Topic History". The table has columns for Document, Topic, and Page#. The data is as follows:

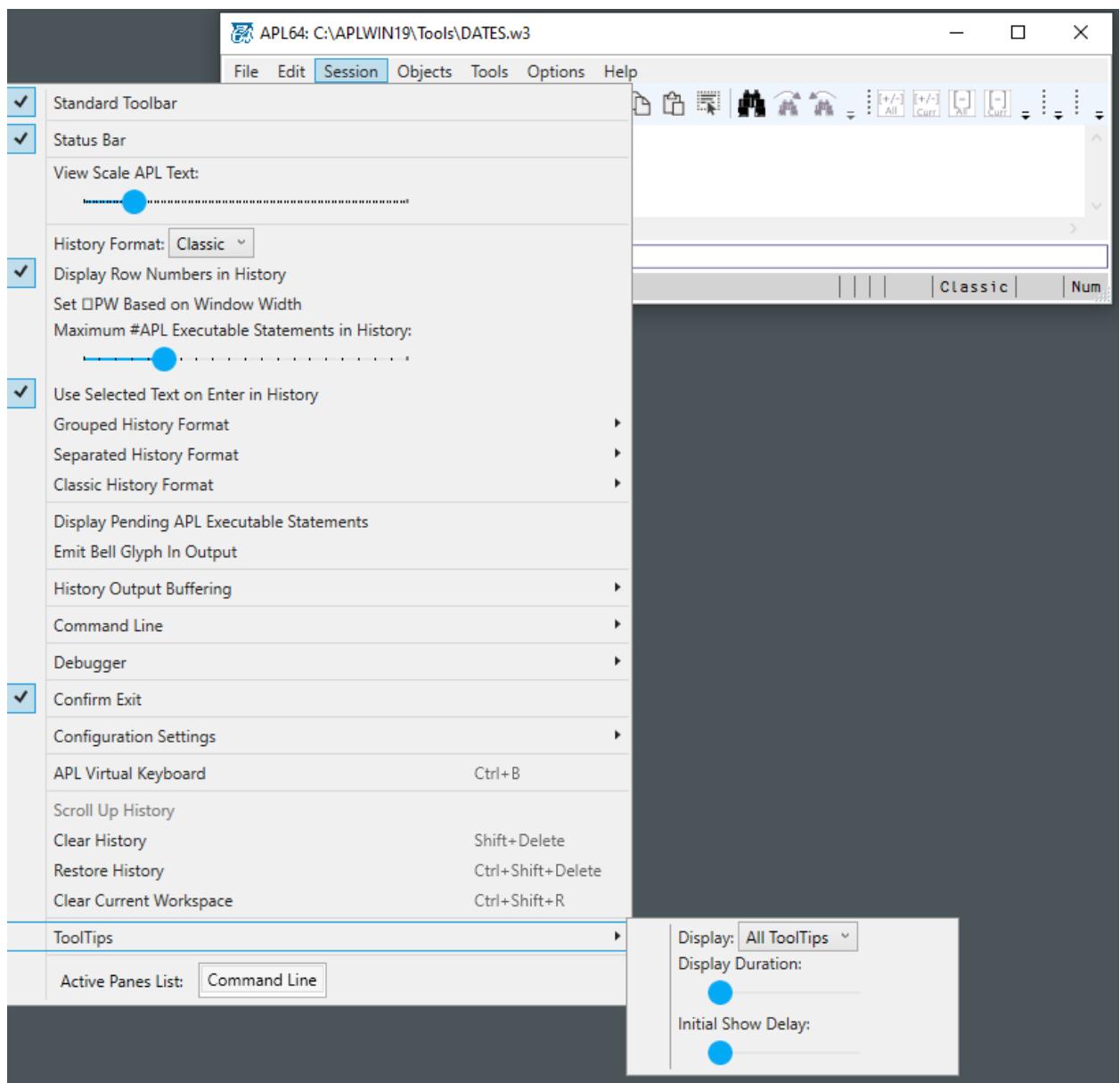
Document	Topic	Page#
http://api2000.com/APL64/UserDocumentation/APL64_SYSTEM_FUNCTION.pdf	□STOP	226
http://api2000.com/APL64/UserDocumentation/APL64_COMPATIBILITY_MODIFI		N/A
http://api2000.com/APL64/UserDocumentation/APL64_APL_PRIMITIVE.pdf	⌚	23

Below the table are two buttons:

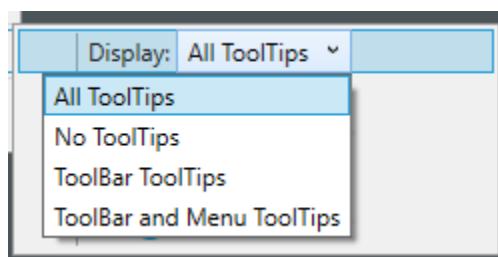
- Documentation Options...
- License & Activation...

Optional Tooltips for Toolbars, Menus and Panes

Depending on developer preference tooltips are presented when the mouse hovers over a toolbar button, menu item, input field or history, editor or debugger panes in an APL64 developer version instance.



- Tooltip display options



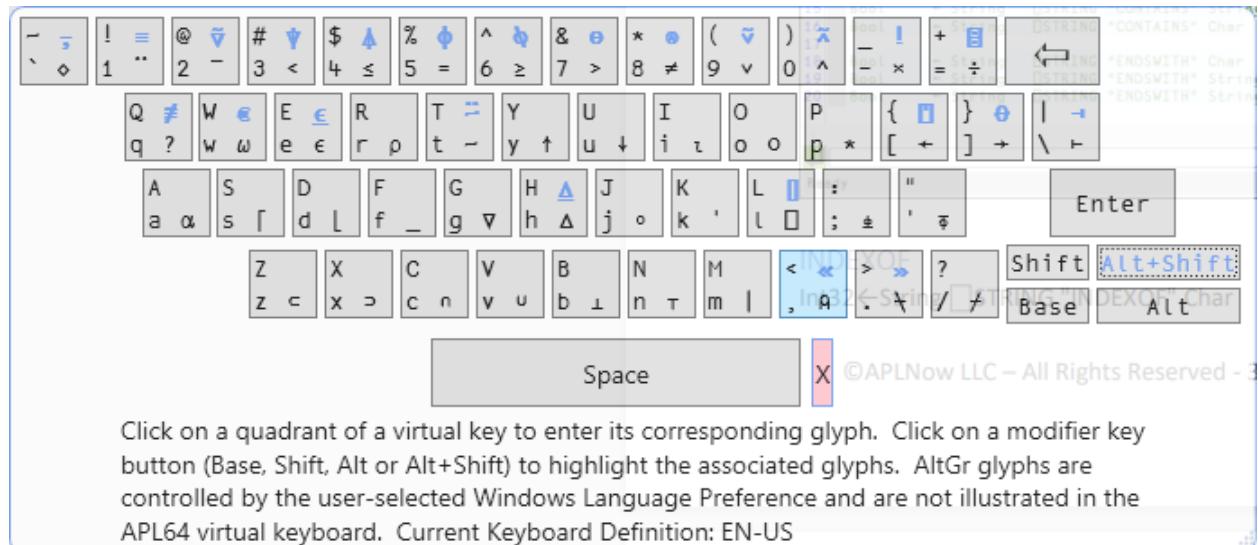
New APL String Data Type Supported

String Definition

An APL string is an ordered group of characters treated as a scalar, with delimiters «...».

Defining a String Variable in APL64

The ‘chevron’ string delimiters are available on the APL64 physical and virtual keyboards as Shift + Alt + < for « and Shift + Alt + > for » for the ‘EN-US’ keyboard definition. Locations of the chevrons may differ for other keyboard definitions.



Click on a quadrant of a virtual key to enter its corresponding glyph. Click on a modifier key button (Base, Shift, Alt or Alt+Shift) to highlight the associated glyphs. AltGr glyphs are controlled by the user-selected Windows Language Preference and are not illustrated in the APL64 virtual keyboard. Current Keyboard Definition: EN-US

To define a string, enclose the text within the « and » glyphs. The APL64 enstring and destring functions are available for conversion between the APL character vector and string formats. APL64 supports string scalars and arrays.

The screenshot shows the APL64 IDE interface. The title bar reads "APL64: C:\APLWIN19\Tools\DATES.w3". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. The main workspace displays the following APL code:

```
0      ⌊dr⌋←S←«my string»
1 my string
2 164
3      ⌊dr⌋←CV→>S
4 my string
5 162
6      S≡<CV
7 1
8 pSV+«String1» «String2»
9 2
10 ⌊dr SV
11 164
12
```

The status bar at the bottom shows "Ready" and "Hist: Ln: 12 Col: 6 | Classic | Num".

The ⌊STRING system function is an interface between the APL64 string datatype and the extensive features of the .Net String class.

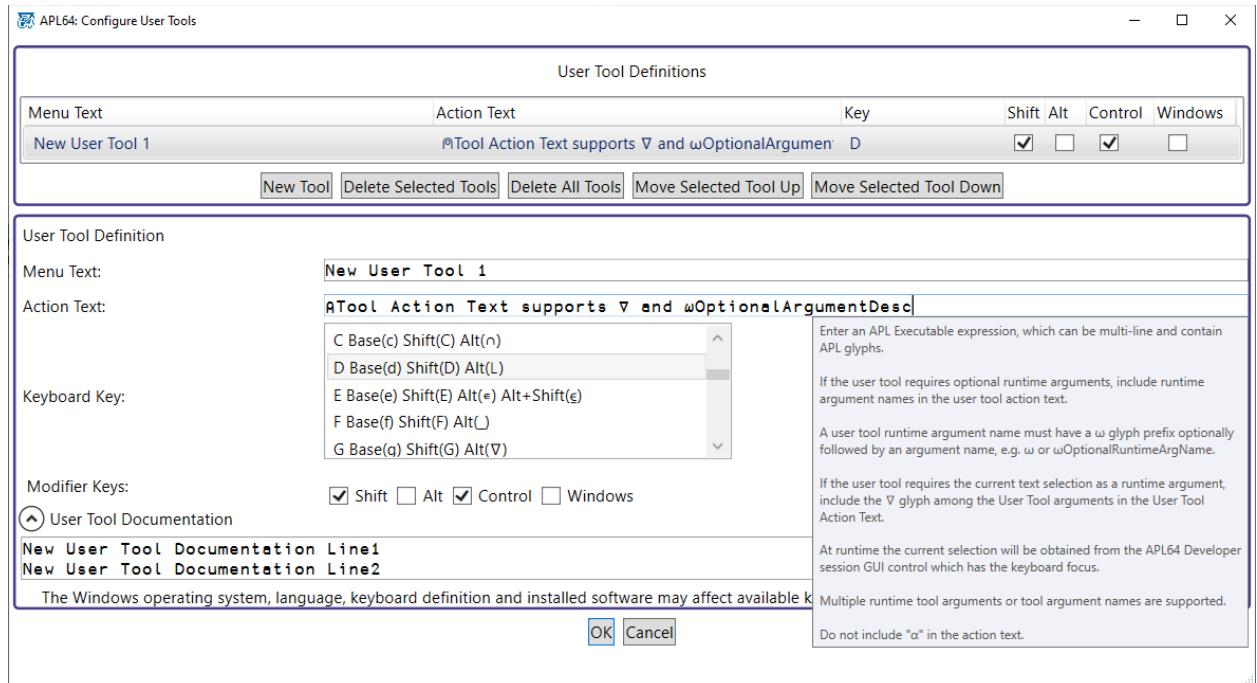
The screenshot shows the APLWIN19 Tools interface with the title bar "APL64: C:\APLWIN19\Tools\DATES.w3". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. The main pane displays a list of user-defined tools, each with a line number, a tool icon, a name, and a description. The tools listed are:

Line	Tool	Description
0	String	'Help'
1	STRING	Documentation
2	Int32	+ String STRING "COMPARE" String
3	Int32	+ String STRING "COMPARE" String IgnoreCase
4	Int32	+ String STRING "COMPARE" StartPosInLarg String StartPosInCom
5	Int32	+ String STRING "COMPARE" StartPosInLarg String StartPosInCom
6		
7	Int32	+ String STRING "COMPAREORDINAL" String
8	Int32	+ String STRING "COMPAREORDINAL" StartPosInLarg String StartP
9		
10	String	+ String STRING "CONCAT" String
11	String	+ String STRING "CONCAT" String String
12	String	+ String STRING "CONCAT" String String String
13	String	+ String STRING "CONCAT" String[]
14		
15	Bool	+ String STRING "CONTAINS" String
16	Bool	+ String STRING "CONTAINS" Char
17		
18	Bool	+ String STRING "ENDSWITH" Char
19	Bool	+ String STRING "ENDSWITH" String
20	Bool	+ String STRING "ENDSWITH" String IgnoreCase
21		
22	Bool	+ String STRING "EQUALS" String
23		

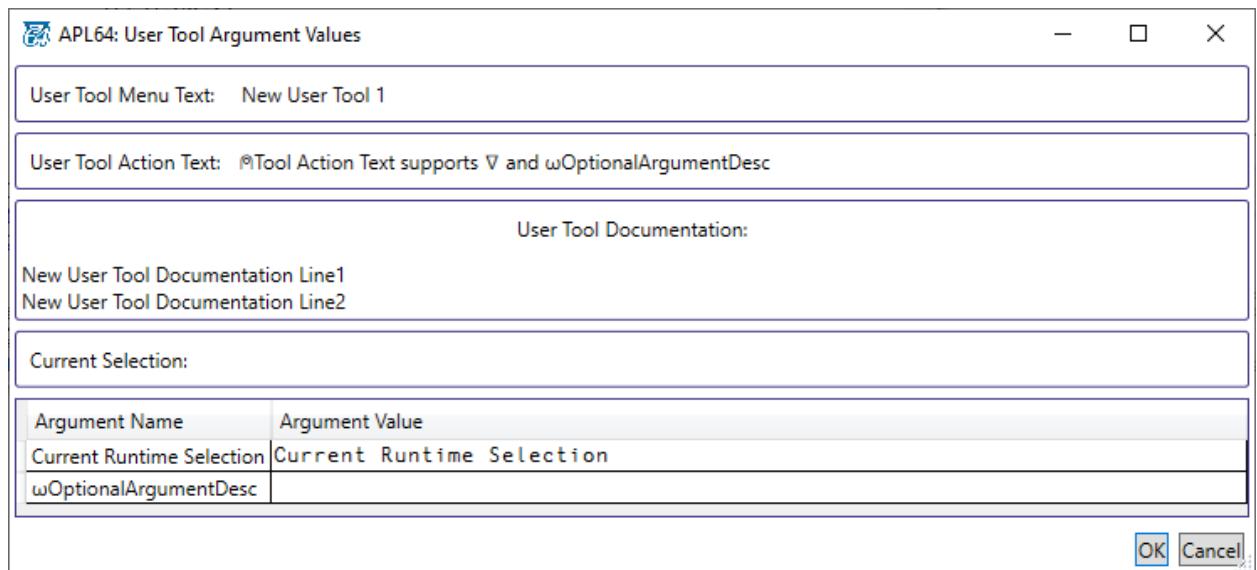
The status bar at the bottom shows "Ready", "Hist: Ln: 0 Col: 13", "Classic", and "Num".

New User-defined Tools Features

The Options > Configure User Tools dialog is enhanced including a user tool documentation multi-line field and optional user tool 'ω' arguments with suffixed user tool argument descriptions.

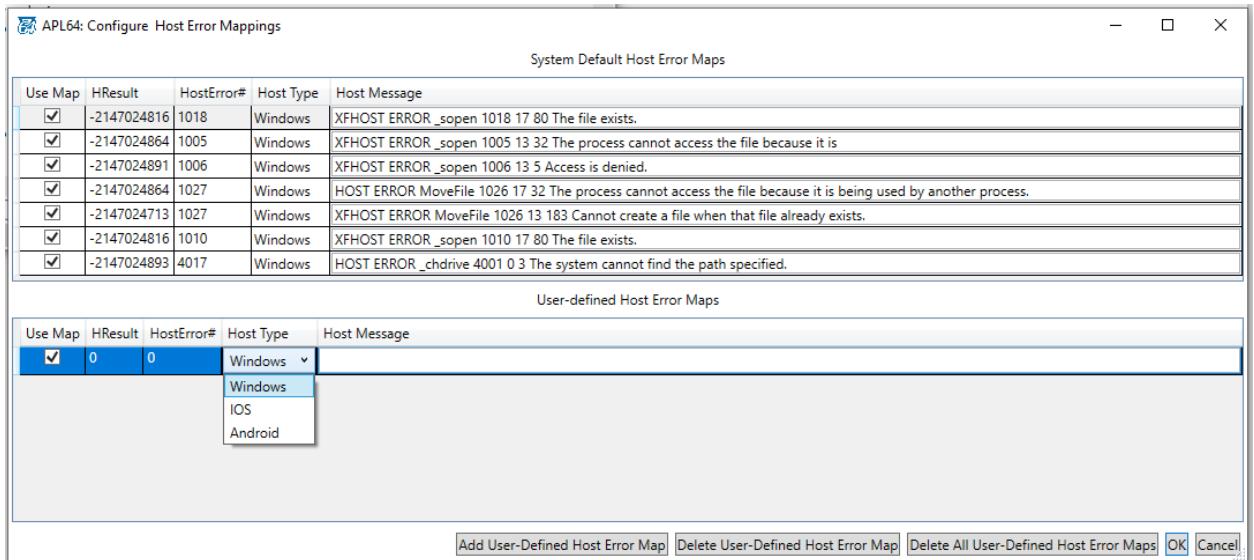


When a user-defined tool is used, the dialog to obtain the tool arguments is enhanced. Options to show/hide User Tool Action Text and User Tool Documentation are available when the user tools is defined.



New Configure Host Error Mappings Dialog

The text of file system errors is not consistent between Win32 and .Net. The APL64 programmer may use the 'Host Error Mappings' dialog to coordinate exception handling in an APL+Win-based application with an APL64-based application system using the *Options > Configure Host Error Mappings...* dialog.

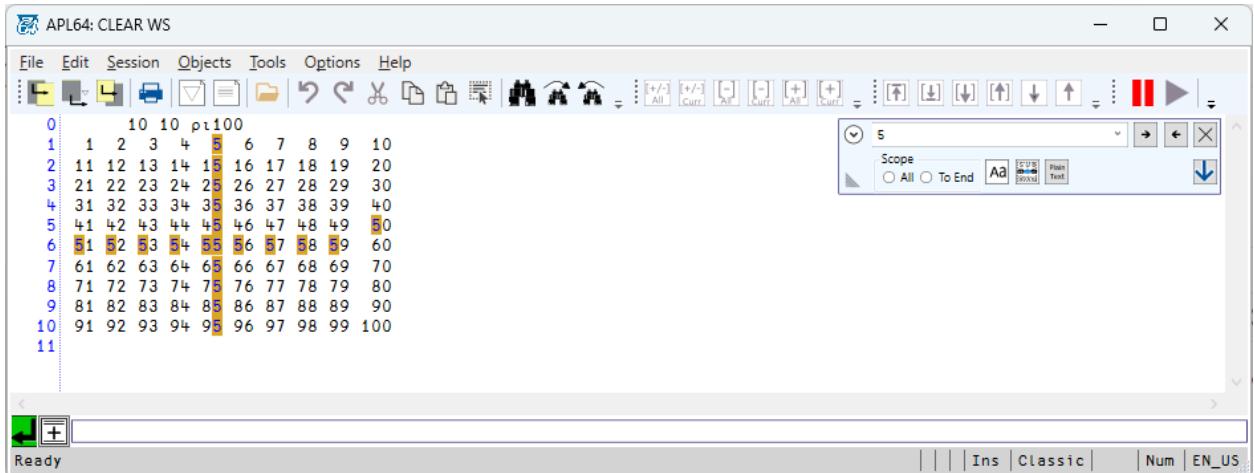


Enhanced Find/Replace Dialog

The APL64 Find/Replace dialog supports options for:

- Find and Replace
- Scope: All, From Current Cursor to End and for functions uncollapsed regions
- Undo, Redo, UndoAll, RedoAll
- Text, Wildcard or Regex searches
- Substring, whole word and APL token matching
- Case sensitive or not matching

The Find/Replace user control is embedded in each APL64 text pane: History, Function editor, Variable editor, Debugged code and Command line



When a Regex text to find entry is user-provided, the Find/Replace dialog has regex verification tools available:

The screenshot shows the APL64: CLEAR WS application window. At the top, there is a 'Find/Replace' dialog box with the following settings:

- Search Target: Classic History
- Text to Find: `+/-`
- Scope: All
- Status: Regular Expression [Regex] 'Text to Find' format selected.
- Buttons: Verify Text to Find Regex, regexstorm.net

Below the dialog, the main workspace displays a grid of numbers from 0 to 100. The numbers are arranged in a 10x10 grid. Some numbers are highlighted in yellow, such as 5, 15, 25, 35, 45, 55, 65, 75, 85, and 95. The APL64 interface includes a toolbar with various icons and a status bar at the bottom.

XL Exchange Data Between APL64 and Excel Worksheet Without Excel.exe

Data in an APL variable can be used to fill selected rows/columns of an Excel worksheet. Data in selected rows/columns of an Excel worksheet can be used to create an APL variable. The XL APL64 system function does not require the use of Excel.exe on the target workstation.

Syntax: Result \leftarrow XL Action [ActionArg1] ...

Actions (not case-sensitive):

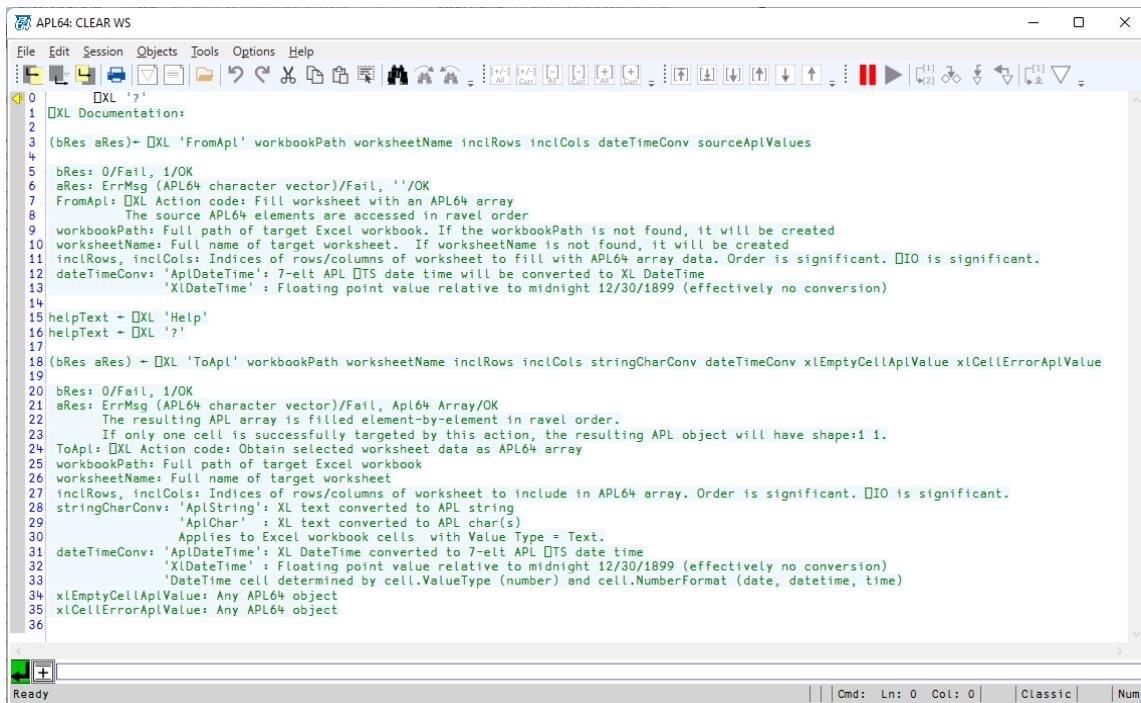
Action	Description	Reqd. Action Args.
'?'	<input type="checkbox"/> XL Detailed syntax	None

'Help'	<input type="checkbox"/> XL Detailed syntax	None
'ToApl'	Create APL variable from selected worksheet rows/columns	1, 2, 3, 4, 5, 6, 7, 8
'FromApl'	File selected rows/columns of worksheet from APL variable	1, 2, 3, 4, 6, 9

Action Arguments:

#	Action Argument Name	Case-Sensitive	Options	Applicable Action
1	workbookPath	OS-dependent	Full path to XL workbook	'ToApl', 'FromApl'
2	worksheetName	N	Name of target worksheet	'ToApl', 'FromApl'
3	inclRows	N/A	Integer vector	'ToApl', 'FromApl'
4	inclCols	N/A	Integer vector	'ToApl', 'FromApl'
5	stringCharConv	N	'ApString', 'ApChar'	'ToApl'
6	dateTimeConv	N	'ApDateTime', 'XLDateTime'	'ToApl', 'FromApl'
7	xlEmptyCellAplValue	N/A	APL variable	'ToApl'
8	xlCellErrorAplValue	N/A	APL variable	'ToApl'
9	sourceAplValues	N/A	APL variable	'FromApl'

'?' or 'Help': Provides detailed syntax text in the form of an APL character vector:



The screenshot shows the APL64: CLEAR WS application window. The title bar reads "APL64: CLEAR WS". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. The main window displays a text area with APL code and documentation. The text starts with "0 XL ?" and continues with "1 XL Documentation:" followed by several lines of APL code and explanatory text about the parameters and behavior of the XL action.

```

0 XL ?
1 XL Documentation:
2
3 (bRes aRes)-+ XL 'FromApl' workbookPath worksheetName inclRows inclCols dateTimeConv sourceAplValues
4
5 bRes: 0/Fail, 1/OK
6 aRes: ErrMsg (APL64 character vector)/Fail, ''/OK
7 FromApl: XL Action code: Fill worksheet with an APL64 array
8 The source APL64 elements are accessed in ravel order
9 workbookPath: Full path of target Excel workbook. If the workbookPath is not found, it will be created
10 worksheetName: Full name of target worksheet. If worksheetName is not found, it will be created
11 inclRows, inclCols: Indices of rows/columns of worksheet to fill with APL64 array data. Order is significant. ⌐IO is significant.
12 dateTimeConv: 'ApDateTime': 7-elt APL ⌐TS date time will be converted to XL DateTime
13 'XLDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)
14
15 helpText + XL 'Help'
16 helpText + XL '?'
17
18 (bRes aRes) -+ XL 'ToApl' workbookPath worksheetName inclRows inclCols stringCharConv dateTimeConv xlEmptyCellAplValue xlCellErrorAplValue
19
20 bRes: 0/Fail, 1/OK
21 aRes: ErrMsg (APL64 character vector)/Fail, APL64 Array/OK
22 The resulting APL array is filled element-by-element in ravel order.
23 If only one cell is successfully targeted by this action, the resulting APL object will have shape:1 1.
24 ToApl: XL Action code: Obtain selected worksheet data as APL64 array
25 workbookPath: Full path of target Excel workbook
26 worksheetName: Full name of target worksheet
27 inclRows, inclCols: Indices of rows/columns of worksheet to include in APL64 array. Order is significant. ⌐IO is significant.
28 stringCharConv: 'ApString': XL text converted to APL string
29 'ApChar': XL text converted to APL char(s)
30 Applied to Excel workbook cells with value Type = Text.
31 dateTimeConv: 'ApDateTime': XL DateTime converted to 7-elt APL ⌐TS date time
32 'XLDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)
33 DateTime cell determined by cell.ValueType (number) and cell.NumberFormat (date, datetime, time)
34 xlEmptyCellAplValue: Any APL64 object
35 xlCellErrorAplValue: Any APL64 object
36

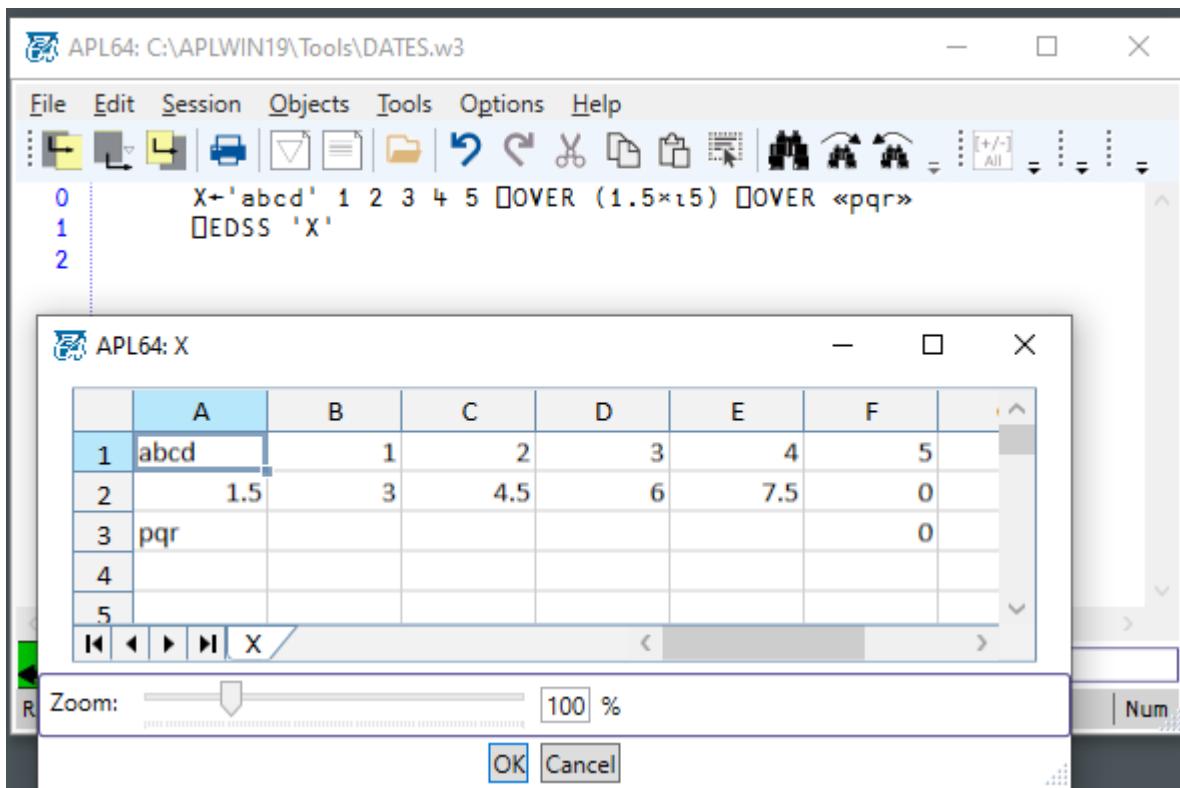
```

Additional □XL Actions:

GetA1AddrFromR1C1Addr, GetA1ColNameFromColNum, GetR1C1AddrFromAbsA1Addr, GetUsedRange and GetWorksheetNames. Descriptions and examples for all □XL actions are in the menu Help > APL Language > Using □XL.

□EDSS Edit an APL Array in a Worksheet Editor

□EDSS is a new system function for editing an APL array of rank 2 or less in a graphical worksheet editor. Data in an APL variable can be used to fill selected rows/columns of a worksheet. Data in selected rows/columns of a worksheet can be used to create/update an APL variable. A □EDSS editor is not modal, so multiple □EDSS editors may be created and data may be pasted between a □EDSS editor and other areas of the APL64 developer GUI including other □EDSS editors, the history pane and the command line as well as other Windows applications.



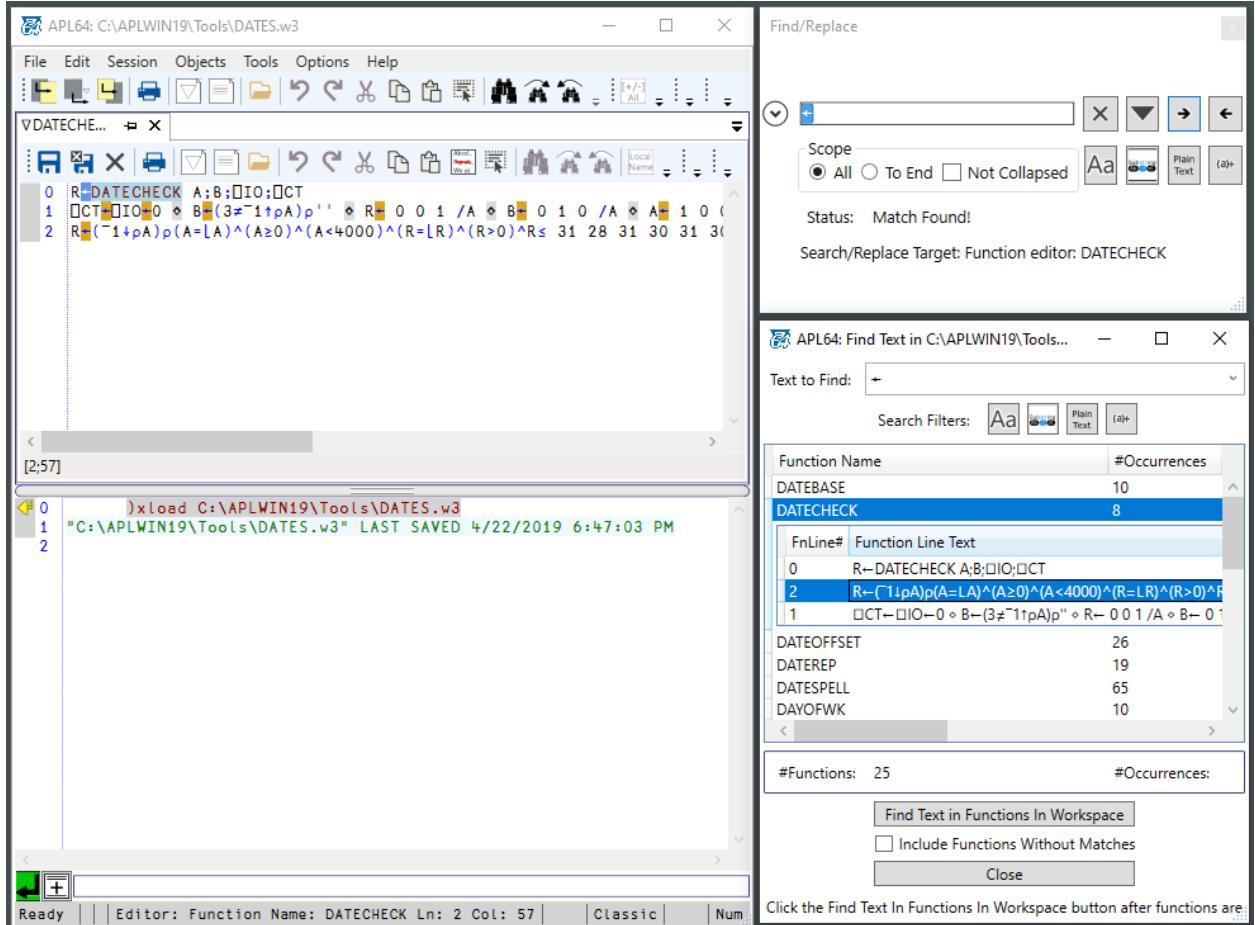
)EDSS Edit an APL Array in a Worksheet Editor

)EDSS is a new system command for editing an APL array of rank 2 or less in a graphical worksheet editor. This system command is analogous to the □EDSS system function.

Find In Functions In Workspace Utility

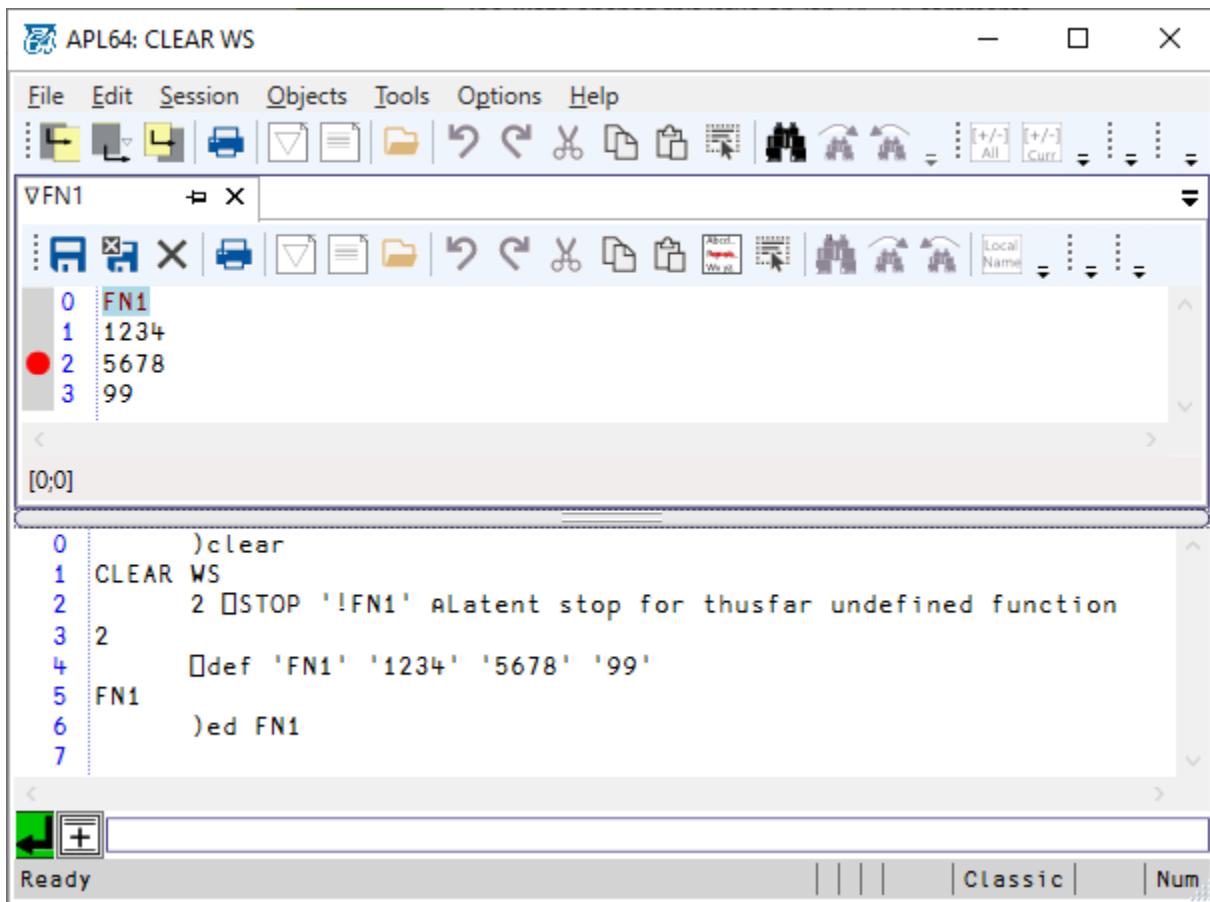
The Edit > Find in Functions in Workspace utility will identify the function lines of the functions in the workspace which match the developer-selected text. A double left click on a user-specified

row in the Find In Functions In Workspace dialog will present the selected function in a function editor.



Latent Function Stops

The `STOP` system function in APL64 supports latent stops which take effect when a function is defined in the current workspace. The enhanced syntax is: lines `STOP '!fnName'`.



UTC Time Supported Universally in APL64

The following system functions and commands were updated to now save and report in UTC time:
⎕AT, ⎕PSAVE, ⎕SAVE, ⎕TS, ⎕COPY, ⎕SAVE and Component file operations.

New menu Session > Configuration Settings > Apply Default Settings

This option allows APL64 to restart with all the default configuration settings.

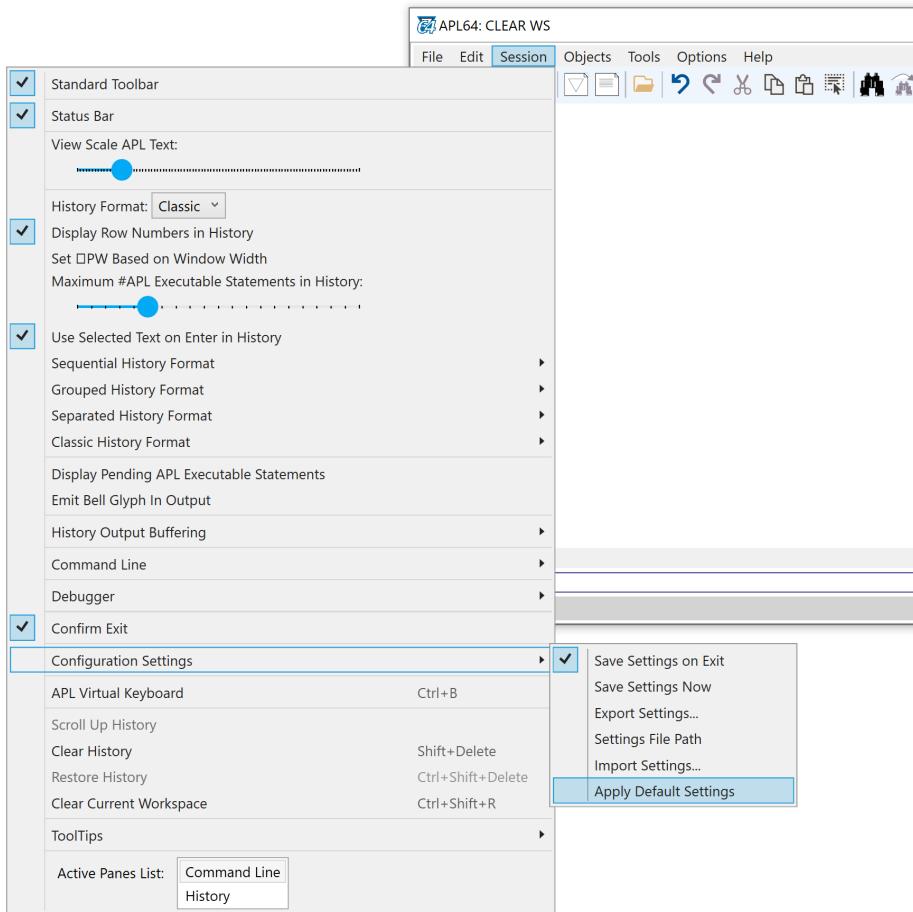


Fig: The menu location

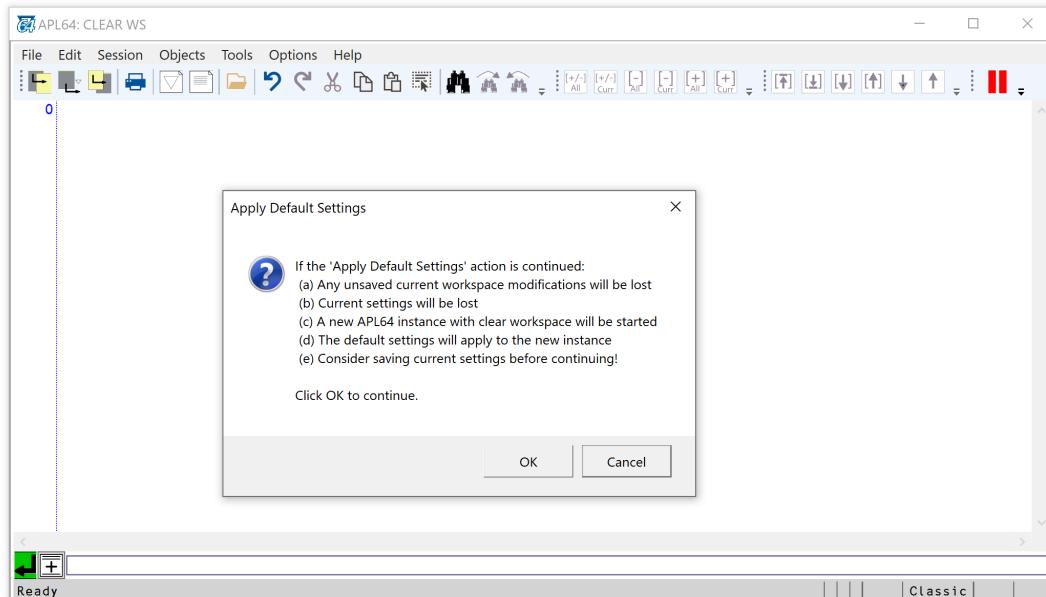
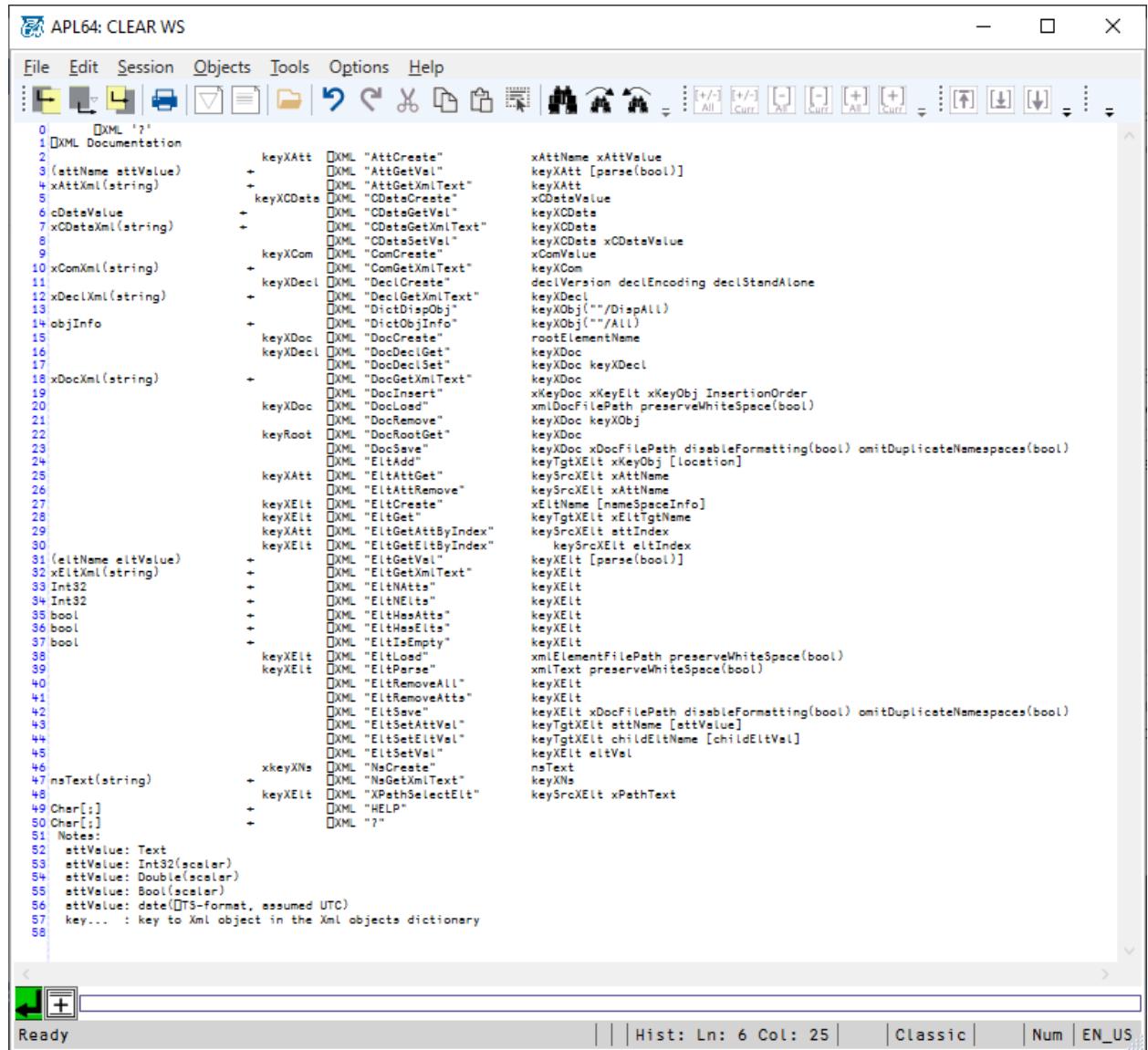


Fig: Warning message

XML to Create, Edit, Read and Write XML-format Documents

The new XML system function exposes the features of Microsoft Linq for XML. Using XML an APL programmer can create, edit, read and write XML-format documents. XML is a ubiquitous, platform-independent, self-descriptive text file format used to represent, receive and transmit hierarchical data.

Documentation is available in the menu Help > APL Language > Using XML. Summary documentation is available via XML '?'.



The screenshot shows the APL64: CLEAR WS IDE interface. The title bar reads "APL64: CLEAR WS". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. The main window displays the XML documentation for the XML system function. The code is organized into sections and subsections, with some lines starting with a plus sign (+) indicating further expansion. The code includes definitions for XML objects like XML, AttCreate, AttGetVal, AttGetXmlText, CDatCreate, CDatGetVal, CDatGetXmlText, CDatSetVal, ComCreate, ComGetXmlText, DeclCreate, DeclGetXmlText, DictDispObj, DictObjInfo, DocCreate, DocDecGet, DocDecGetSet, DocGetXmlText, DocInsert, DocClose, DocRemove, DocRootGet, DocSave, EltAdd, EltAttGet, EltAttRemove, EltCreate, EltGet, EltGetAttByIndex, EltGetElitByIndex, EltGetVal, EltGetXmlText, EltNatts, EltNElts, EltHasAtts, EltHasElts, EltIsEmpty, EltLoad, EltParse, EltRemoveAll, EltRemoveAtts, EltSave, EltSetAttVal, EltSetElitVal, EltSetVal, and XkeyXNs. It also includes definitions for XML objects like XML, AttCreate, AttGetVal, AttGetXmlText, CDatCreate, CDatGetVal, CDatGetXmlText, CDatSetVal, ComCreate, ComGetXmlText, DeclCreate, DeclGetXmlText, DictDispObj, DictObjInfo, DocCreate, DocDecGet, DocDecGetSet, DocGetXmlText, DocInsert, DocClose, DocRemove, DocRootGet, DocSave, EltAdd, EltAttGet, EltAttRemove, EltCreate, EltGet, EltGetAttByIndex, EltGetElitByIndex, EltGetVal, EltGetXmlText, EltNatts, EltNElts, EltHasAtts, EltHasElts, EltIsEmpty, EltLoad, EltParse, EltRemoveAll, EltRemoveAtts, EltSave, EltSetAttVal, EltSetElitVal, EltSetVal, and XkeyXNs. The code is annotated with comments explaining the parameters and return types for each method.

```
0  XML ?'
1 XML Documentation
2  (attName attValue)
3  + XML "AttCreate"
4  + XML "AttGetVal"
5  + XML "AttGetXmlText"
6  keyXData
7  + XML "CDatCreate"
8  + XML "CDatGetVal"
9  + XML "CDatGetXmlText"
10 + XML "CDatSetVal"
11 keyXCom
12 + XML "ComCreate"
13 + XML "ComGetXmlText"
14 keyXDecl
15 + XML "DeclCreate"
16 + XML "DeclGetXmlText"
17 keyXDoc
18 + XML "DocCreate"
19 + XML "DocDecGet"
20 + XML "DocDecGetSet"
21 + XML "DocGetXmlText"
22 keyXDoc
23 + XML "DocInsert"
24 + XML "DocClose"
25 + XML "DocRemove"
26 keyXRoot
27 + XML "DocRootGet"
28 + XML "DocSave"
29 keyXAtt
30 + XML "EltAdd"
31 + XML "EltAttGet"
32 + XML "EltAttRemove"
33 + XML "EltCreate"
34 + XML "EltGet"
35 + XML "EltGetAttByIndex"
36 + XML "EltGetElitByIndex"
37 + XML "EltGetVal"
38 keyXElit
39 + XML "EltGetXmlText"
40 + XML "EltNatts"
41 + XML "EltNElts"
42 + XML "EltHasAtts"
43 + XML "EltHasElts"
44 + XML "EltIsEmpty"
45 keyXElit
46 + XML "EltLoad"
47 nsText(string)
48 + XML "EltParse"
49 Char[]
50 Char[]
51 Notes:
52 attValue: Text
53 attValue: Int32(scalar)
54 attValue: Double(scalar)
55 attValue: Bool(scalar)
56 attValue: date[DT3-format, assumed UTC]
57 key... : key to Xml object in the Xml objects dictionary
58
```

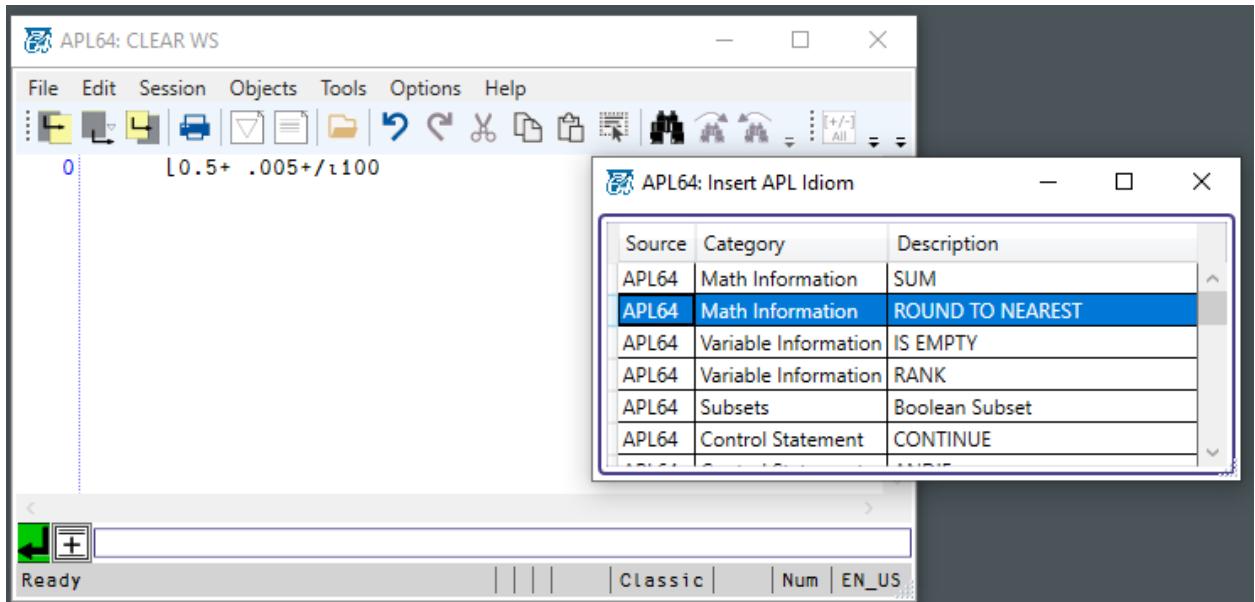
APL64 Idioms Manager

The APL64 Idioms Manager is a programmer productivity tool to enable the deployment of APL idioms into APL64-based application systems. An APL idiom is text which represents frequently

used APL statements. The APL idiom may contain complete APL executable statements or may require additional APL programmer input to become APL executable statements. An option is provided to specify the placement of the caret when the idiom text is inserted into a function editor, variable editor, command line or editable classic history.

The APL64: Insert APL Idiom dialog is accessible from the menus below:

- Edit > APL Idioms > Insert An APL Idiom...(Ctrl+I)



The APL64 Idioms Manager dialog is used to add or edit user-defined APL idioms:

- Edit > APL Idioms > Idioms Manager...(Ctrl+Shift+I)

The caret position within an APL64 idiom can be specified in the idiom text.

The screenshot shows the APL64 Idioms Manager application window. It has two main sections:

- APL64 Idioms Tab:** Displays a table of built-in idioms categorized by type (e.g., Math Information, Variable Information, Subsets, Control Statement) and their corresponding text representations.
- User Idioms Tab:** Displays a table for user-defined idioms with columns for Category, Description, and Text. A text input field below the table allows users to enter idiom text, with instructions about APL glyphs and multi-line support. Buttons for adding new idioms, deleting all, and importing are at the bottom.

Category	Description	Text
Math Information	SUM	+/ α
Math Information	ROUND TO NEAREST	L0.5+ α
Variable Information	IS EMPTY	0 $\epsilon\alpha$
Variable Information	RANK	p α
Subsets	Boolean Subset	$\alpha/\tau p$
Control Statement	CONTINUE	:CONTINUE α

Category	Description	Text	Delete
My Cat	My Desc	My Idiom Text	Delete

Enter the idiom text. APL glyphs are supported. Multi-line idiom test is supported. Click Enter/Return to add a line to the idiom text. The idiom text will be inserted at the user-selected caret location. The idiom text is not executed after insertion. Optionally include an ' α ' glyph in the idiom text to indicate the desired caret location after insertion.

Add New User Idiom | Delete All User Idioms | Import User Idioms | Close

Documentation is available in the menu Help >Developer Version GUI > Using the Idioms Manager.

Updates to Existing System Functions, Commands and Variables

Details of all APL64 System Functions can be found [here](#).

)CMDA System Command and CMD System Function with Administrator Permissions

SYS[33] System Variable Enhanced Control of INBUF Clearing Behavior

UCS Returns Unicode Points of Glyphs

UCASE, LCASE

UCMD User Command Facility

LIBD System Function

System Functions Now Fully-integrated into APL64

The NFE(Native File Access with Encoding) and CSE (C# Script Engine) system functions are now fully integrated into APL64 for superior performance compared to APL+Win.

Many New System Functions

Details of all APL64 System Functions can be found [here](#).

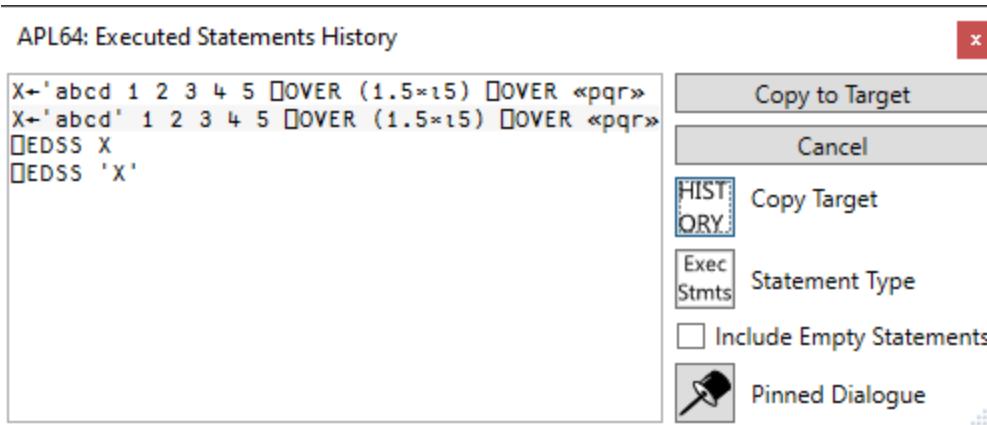
- A (char vector 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
- ACBD (GET System.AppContext.BaseDirectory)
- AL (char vector 'abcdefghijklmnopqrstuvwxyz')
- D (Char vector '0123456789')
- DEVCAP
- DIV
- DRIVEINFO
- DTB, □ DLB, □ DLTB, □ DEB
- DTBR
- EDIT
- EDITEVENTS
- EDSS
- EXEPATH
- FINDINFILES
- JUST, □ RJUST
- MATRIFY
- NAN
- NBLENGTH
- NEXTO
- OS
- OVER, □ OVERV
- PATH
- ROWFIND
- SSCAT, □ SSDEB, □ SSDLB, □ SSDLTB, □ SSDTB
- SCOM - Superior serial (RS-232) communication capabilities compared to □ ARBIN in APL+Win

- SKD
- SPLASH
- SSFIND, SSINDEX, SSLEN, SSSHAPE, SSTAKE, SSUNIQUE
- SSTOMAT, MATTOSS, SSASSIGN, SSDROP, SSCOMPRESS
- SPLASH
- STRING
- SYSCONFIGFILE
- SYSVERN
- TELPRINT
- TEXTREPL
- TRANSLATE
- UCASE, LCASE
- UCMDFILE
- USERPATH
- WHERE
- WORDREPL
- WRE
- XL
- XML

Enhanced Dialogs

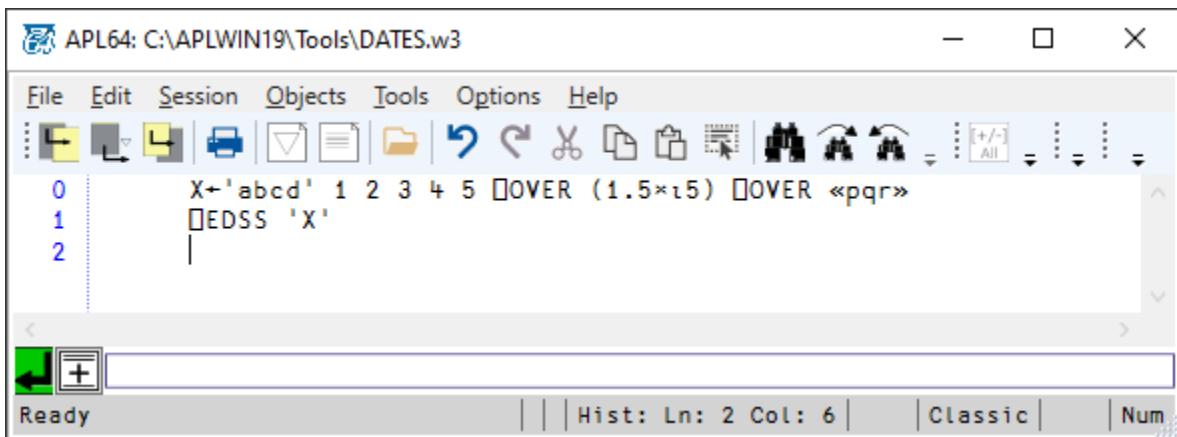
APL Statements History – More options than Gather action

Access this dialog via Edit | APL Statements from the History | APL Statements History... menu item. All, Executable or Result statements may be filtered. The copy target can be selected as History, Command Line or Windows Clipboard.



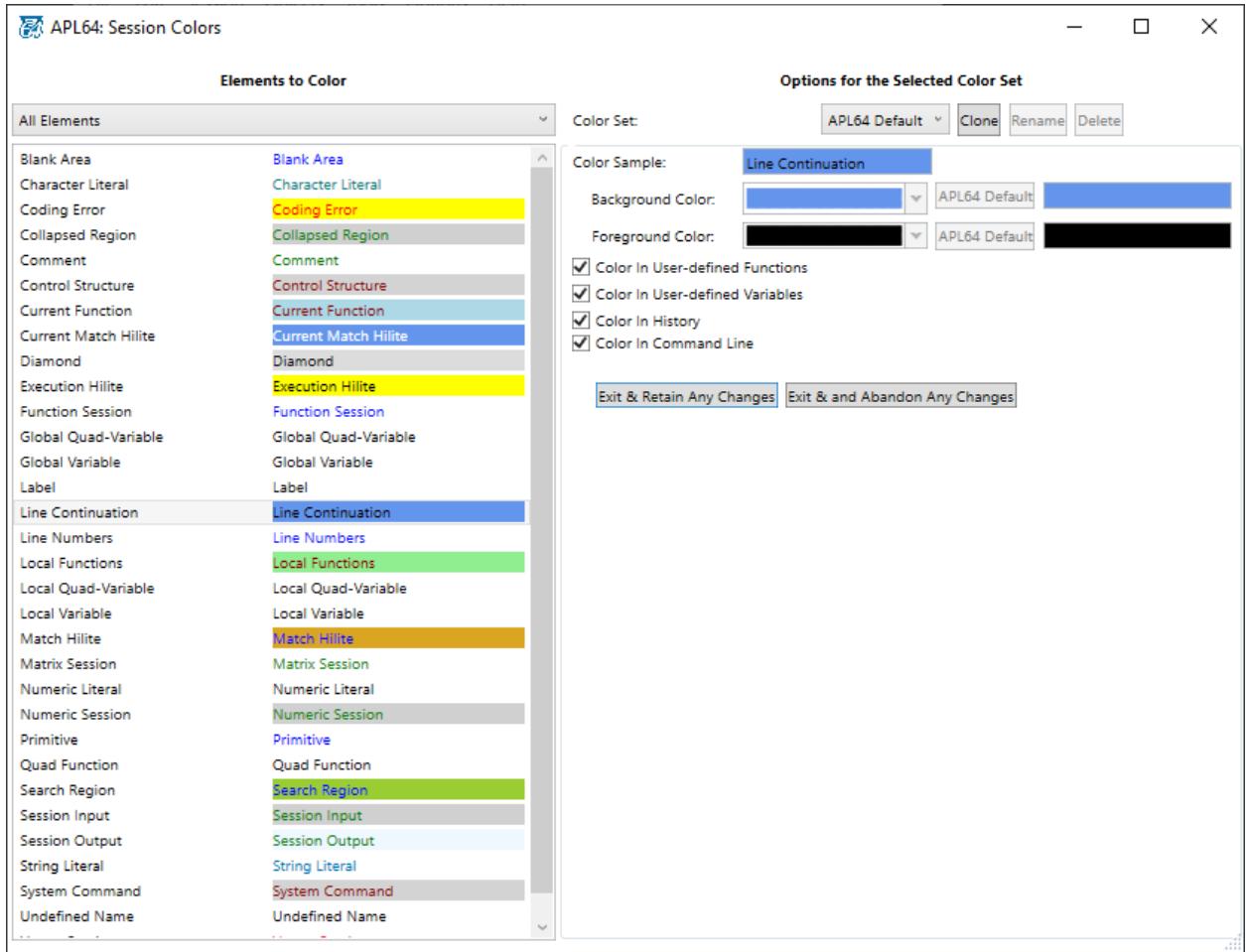
History Pane – Optional Row Numbers

Use the Session | Display Row Numbers In History menu item to control this option.



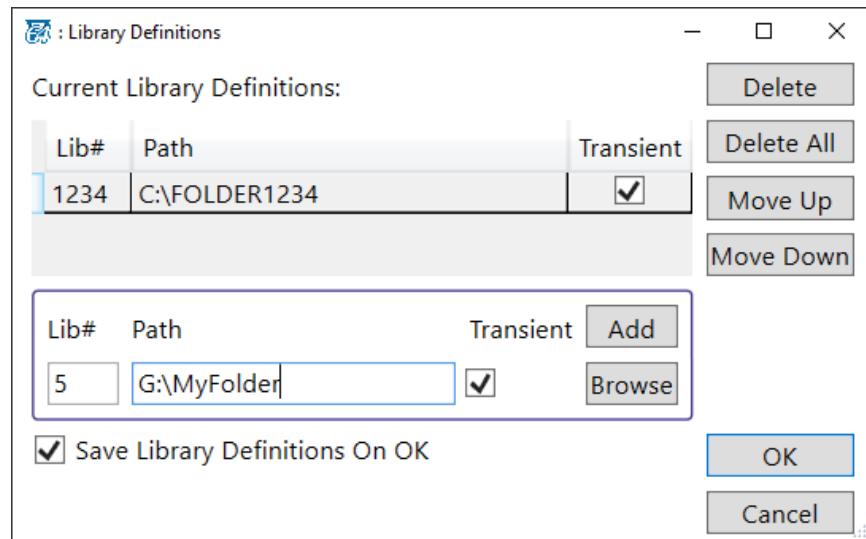
Colors Dialog

Multiple user-defined color sets may be created by cloning the system default color set.



Library Definitions

A library can be user-defined as transient or not, which indicates if a library definition will persist in the APL64 configuration file.



History Log Enhanced

APL64: APL History Log Options

Logging State

Currently Logging
 Commence Logging at Session Start

Logging Threshold

#Unlogged History Statements Threshold:

Overwrite Log File
 Issue Alert when %Threshold is Exceeded

%Threshold for Alert:

Log File

Log File Prefix:

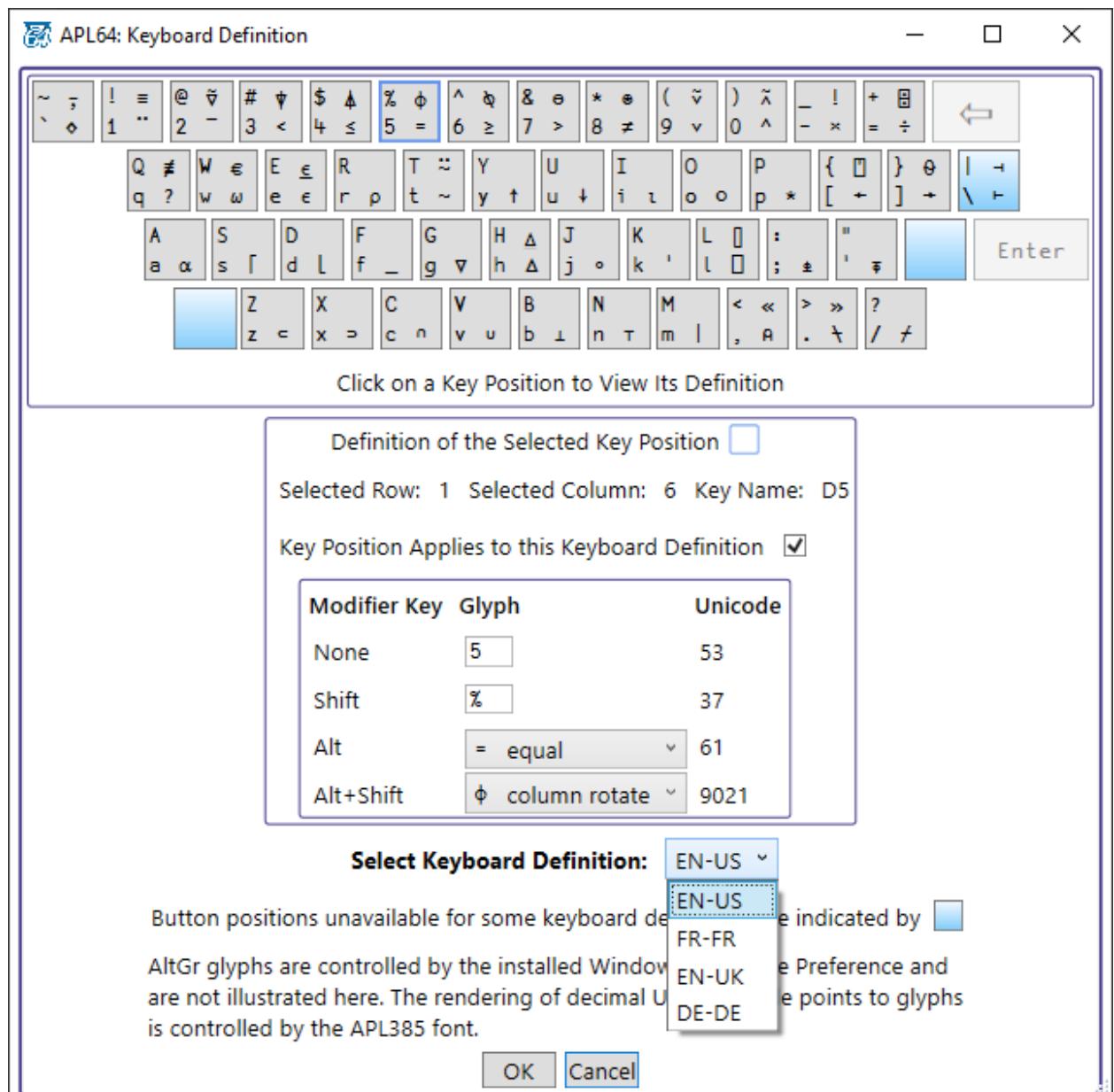
Log File Path:

Recent Log File: N/A

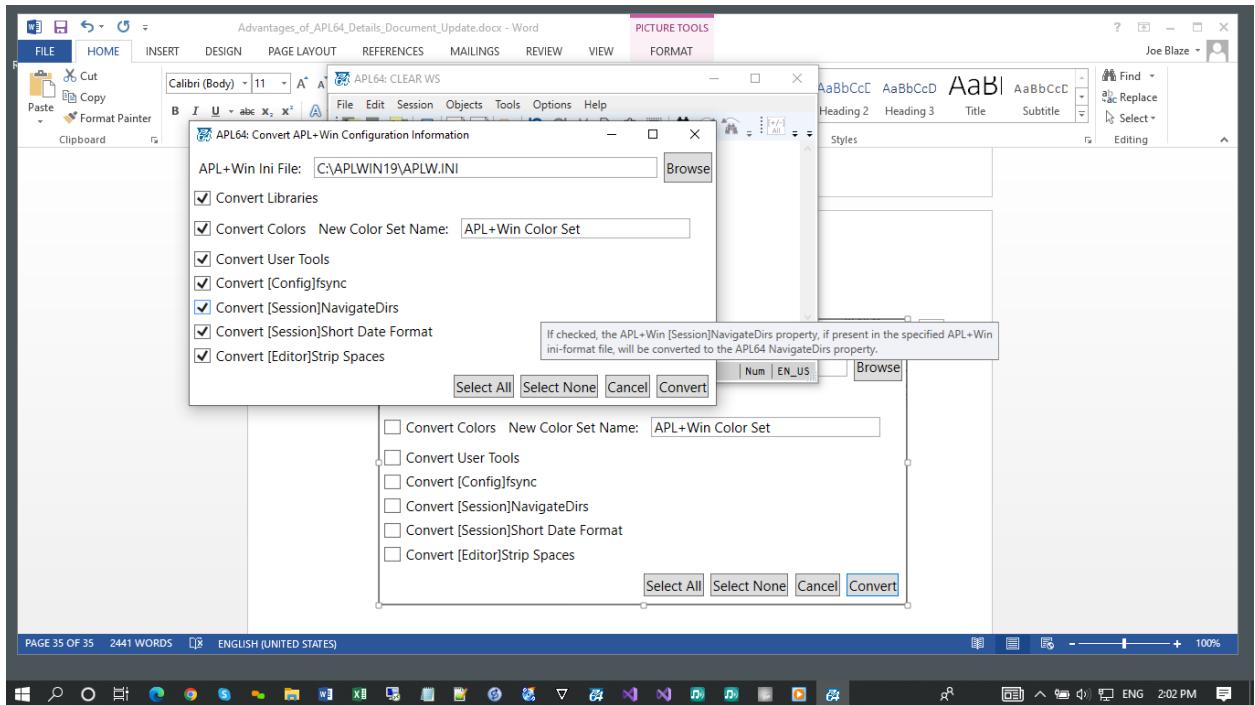
Log Import Options

Import Command Line Executed APL Statements
 Import Result Type APL Statements
 Import Callback Type APL Statements
 Import All Other APL Statement Types

Keyboard Definition



Convert APL+Win Configuration File



APL64 Now Supports .NET Release 6.0 (x64)

This second production release of APL64, 2022.0.2, requires the installation of the Microsoft .NET SDK 6.0.402. The installer for this version of APL64 will prompt you to download and install it.

HttpClient Now Used

The HttpClient technology replaces the Microsoft-deprecated WebClient technology used to display context-sensitive Help and System Highlights in APL64.

New Path System Function

The Path system function is an interface to the [.Net Path class methods](#) to perform file or directory path information operations.

The screenshot shows the APL64: CLEAR WS application window. The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and zoom. The main pane displays a list of APL functions, each preceded by a line number and a small icon. The functions listed are:

```
0 ⌊PATH '?'
1 ⌊PATH '?'
2 ⌊PATH 'AltDirectorySeparatorChar'
3 ⌊PATH 'DirectorySeparatorChar'
4 ⌊PATH 'GetInvalidFileNameChars'
5 ⌊PATH 'GetInvalidPathChars'
6 ⌊PATH 'GetRandomFileName'
7 ⌊PATH 'GetTempFileName'
8 ⌊PATH 'GetTempPath'
9 ⌊PATH 'PathSeparator'
10 ⌊PATH 'VolumeSeparatorChar'
11 'ChangeExtension' ⌊PATH path
12 'Combine' ⌊PATH vector of path parts
13 'EndsInDirectorySeparator' ⌊PATH path
14 'GetConfigurationUserLevelPath' ⌊PATH 'None' / 'PerUserRoaming' / 'PerUserRoamingAndLocal'
15 'GetDirectoryName' ⌊PATH path
16 'GetExtension' ⌊PATH path
17 'GetFileName' ⌊PATH path
18 'GetFileNameWithoutExtension' ⌊PATH path
19 'GetFullPath' ⌊PATH path
20 'GetPathRoot' ⌊PATH path
21 'GetRelativePath' ⌊PATH path1 path2
22 'HasExtension' ⌊PATH path
23 'IsPathFullyQualified' ⌊PATH path
24 'IsPathRooted' ⌊PATH path
25 'Join' ⌊PATH vector of path parts
26 'PathParts' ⌊PATH path
27 'TrimEndingDirectorySeparator' ⌊PATH path
28 |
```

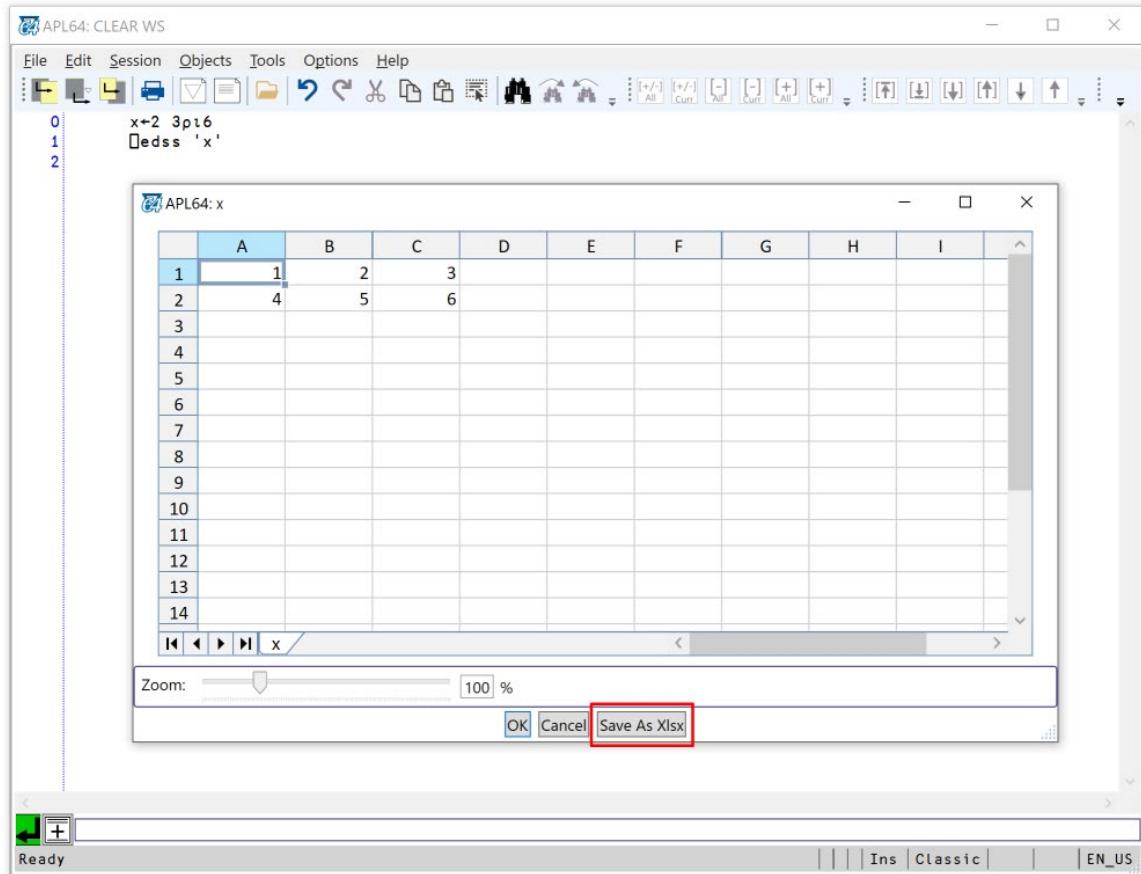
The status bar at the bottom shows "Ready" and "Hist: Ln: 28 Col: 6 Ins | Classic | Num EN_US".

New ⌊Pathcase System Variable

This workspace-related system variable specifies when the APL64 interpreter will compare file paths considering case.

New Save As Xlsx Option in)EDSS/⌊EDSS

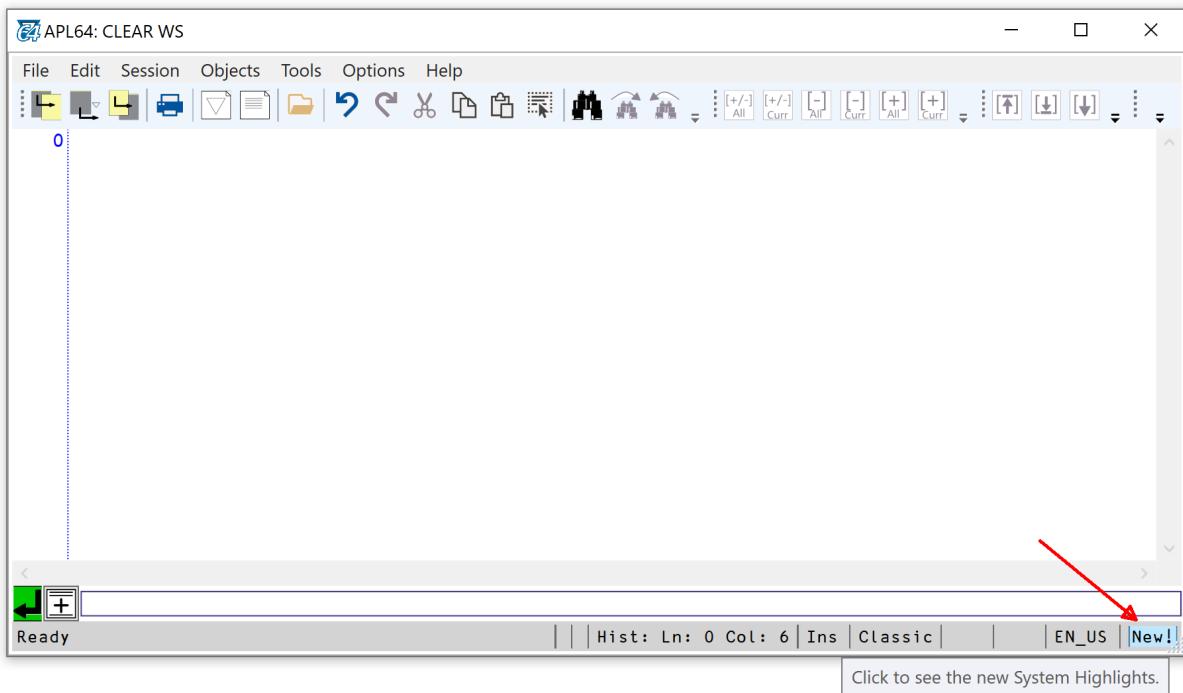
This option can be used to save the edit session information to a Microsoft Excel Open XML Spreadsheet (XLSX) file.



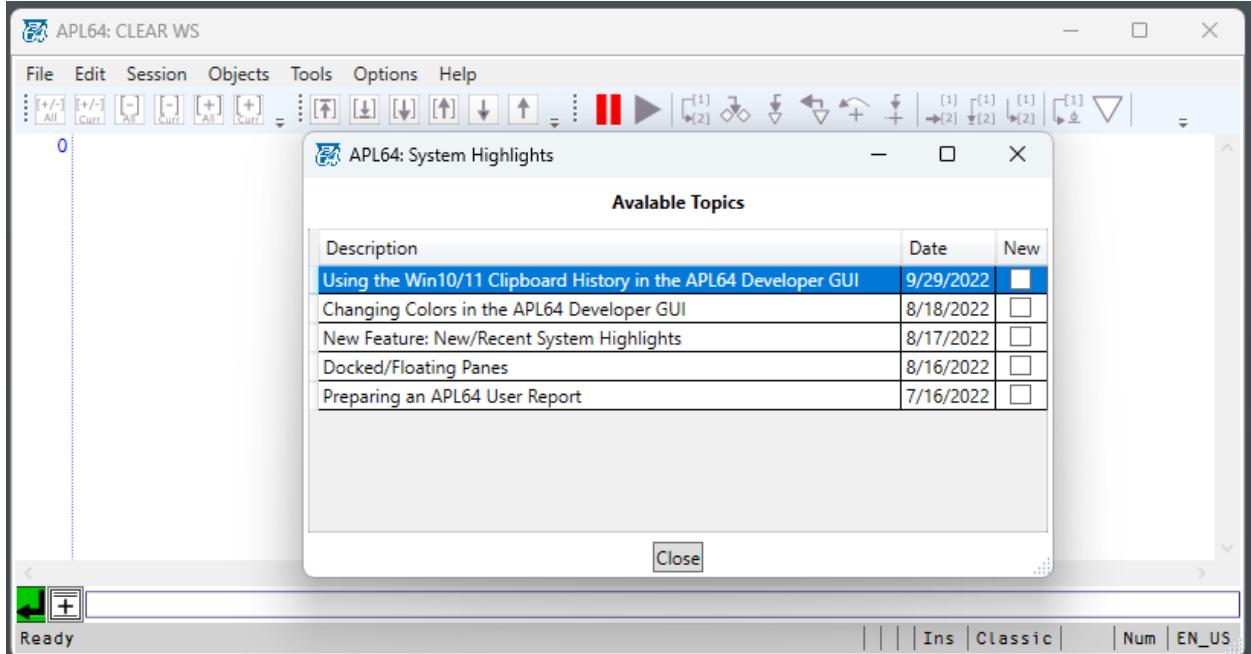
New APL64 System Highlights Feature in the Status Bar

The System Highlights displays custom notifications to APL64 developer version users which can be published by the APL64 Team without creating a system update. Potential uses:

- Warn user's when the license refresh requirement approaches
- Push notifications to customers for new features, updates, and video tutorials



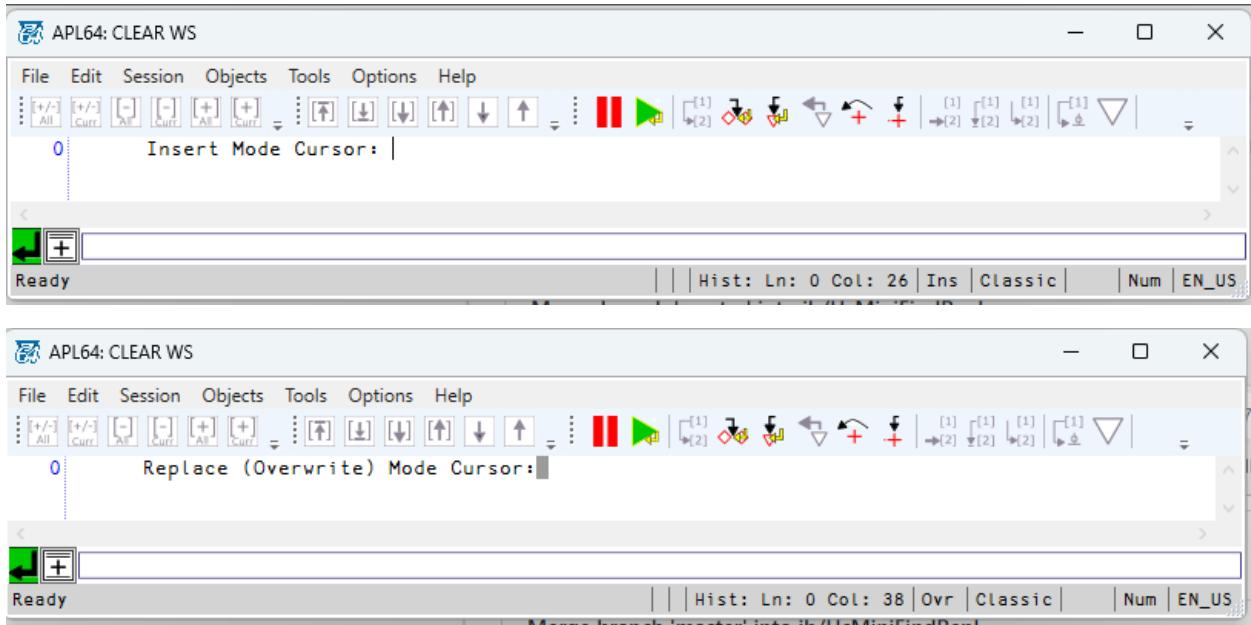
Clicking the notification area in the status bar or the **Help | System Highlights | Recent System Highlights** menu displays the **Available Topics** window shown below:



Note: This is an experimental feature in this release of APL64 and is subject to change or be removed in the future as we gain experience and feedback from users.

Insert/Replace Keyboard Mode Supported in APL64

Pressing the Insert key on the keyboard toggles between Insert and Replace mode. Each mode has a distinctive caret; thin in Insert mode and thick in Replace (Overwrite) mode. The status line displays "Ins" when APL64 is in Insert mode and "Ovr" when in Replace mode.



Find in Functions in Workspace Utility

'Last Fix Date' Column Added to the Filtered List of Functions

This new column as well as the pre-existing columns can be sorted (ascending/descending) by double clicking the column header in the **Edit | Find in Functions** in Workspace:

APL64: Find Text in C:\Users\Joe Blaze\AppData\Roaming\APLNowLLC\APL64...

Function Name	Last Fix Date	#Occurs
DATABASE	1994-08-05T11:28:07.000000Z	10
FnLine# Function Line Text		
0 R←DATABASE A;B;C;□IO;□CT		
1 □CT←□IO←0 ◦ R←(3≠~1↑pA)p" ◦ R← 1 0 0 /A ◦ B← 0 ~693961 ~693930 ~693902 ~693871 ~693841 ◦ C←400IR←R+1900×R<100 ◦ R←(~1↑pA)pB+A-(((100p 1 0 0 0),300p 0 0 0 ,96p 1 0 0 0)[C]^B		
DATECHECK	1994-08-05T11:28:07.000000Z	8
DATEOFFSET	1994-08-05T11:28:07.000000Z	26
DATEREP	1994-08-05T11:28:07.000000Z	19
DATESPELL	1994-08-05T11:28:07.000000Z	65
DAYOFWK	1994-08-05T11:28:07.000000Z	10
DAYOFYR	1994-08-05T11:28:07.000000Z	5
DAYSDIFF	1994-08-05T11:28:07.000000Z	19
DSPELL	1994-08-05T11:28:07.000000Z	8
FTIMEBASE	1994-08-05T11:28:07.000000Z	11

#Functions: 25 #Occurs: 427

Click the Find Text In Functions In Workspace button after functions are added to / removed from the workspace.

New Option is Provided to "Show Function Definition(s) for Selected Row(s)" in the Function List

For example, when this option is checked and the "DATABASE" row is selected, the DATABASE function definition will be displayed in a line-by-line list below the selected row:

The screenshot shows the APL64 Find Text dialog box. The title bar reads "APL64: Find Text in C:\Users\Joe Blaze\AppData\Roaming\APLNowLLC\APL6...". The main area displays a table of function definitions found in the workspace. The columns are "Function Name", "Last Fix Date", and "#Occurs". One row is selected, showing the definition for "DATABASE". The definition is a complex APL expression involving R-, CT-, and various operators. Below the table, it says "#Functions: 25" and "#Occurs: 427". At the bottom, there are checkboxes for "Find Text in Functions In Workspace" (unchecked), "Include Functions Without Matches" (unchecked), and "Show Function Definition(s) for Selected Row(s)" (checked). A note below says: "If checked, the function definition(s) of the selected row(s) will be visible. Checking/Unchecking this checkbox will refresh the function list." To the right, there is a "Workspace" button and a note: "Workspace button after functions are added to / removed from the works...".

Function Name	Last Fix Date	#Occurs
DATABASE	1994-08-05T11:28:07.0000000Z	10
FnLine# Function Line Text		
0 R←DATABASE A;B;C;IO;CT		
1 □CT←□IO←0 ◦ R←(3≠~1↑pA)p" ◦ R← 1 0 0 /A ◦ B← 0 ~693961 ~693930 ~693902 ~693871 ~		
2 C←400IR←R+1900×R<100 ◦ R←(~1↑pA)pB+A-(((100p 1 0 0 0),300p 0 0 0 0,96p 1 0 0 0)[C]^		
DATECHECK	1994-08-05T11:28:07.0000000Z	8
DATEOFFSET	1994-08-05T11:28:07.0000000Z	26
DATEREP	1994-08-05T11:28:07.0000000Z	19
DATESPFL	1994-08-05T11:28:07.0000000Z	65

#Functions: 25 #Occurs: 427

Find Text in Functions In Workspace
 Include Functions Without Matches
 Show Function Definition(s) for Selected Row(s)

If checked, the function definition(s) of the selected row(s) will be visible. Checking/Unchecking this checkbox will refresh the function list.

Workspace button after functions are added to / removed from the works...

devcap Set to an Empty Vector

Assigning `devcap` to an empty vector now displays just the name of the workspace in the APL64 session title bar.

The screenshot shows the APL64 session window. The title bar reads "CLEAR WS". The menu bar includes File, Edit, Session, Objects, Tools, Options, Help. The toolbar has various icons for file operations. The workspace area shows the following APL code:

```

0 )ed #
1 □devcap
2 APL64
3 □devcap← ''
4

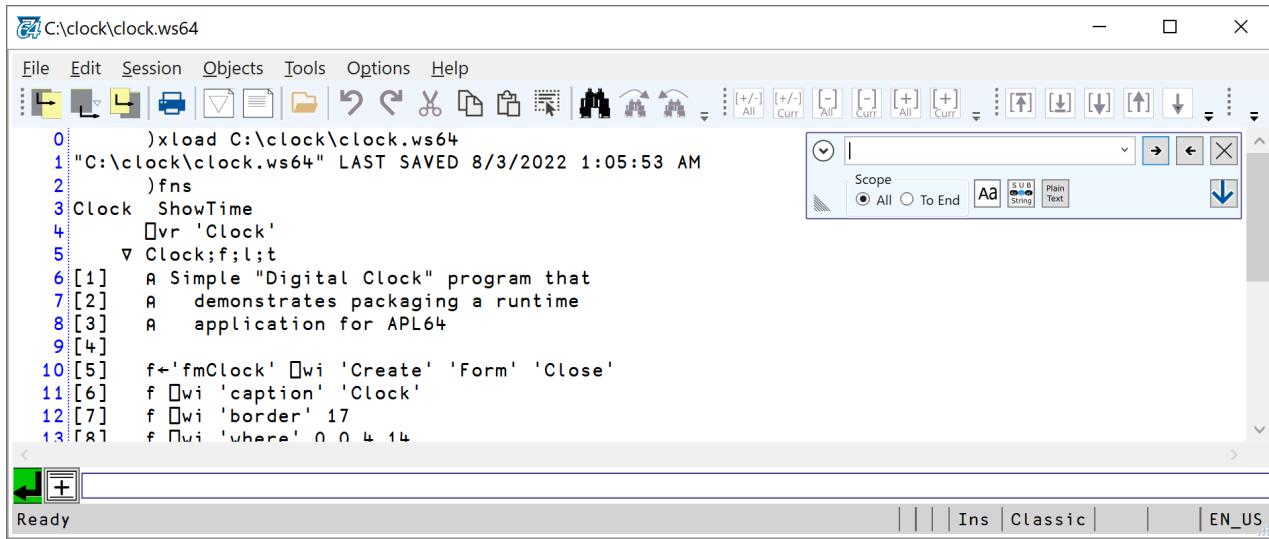
```

The status bar at the bottom shows "Ready", "Hist: Ln: 4 Col: 6", and buttons for Ins, Classic, Cap, and EN_US.

The Find/Replace Tool Replaces the Find/Replace Dialog

The Find/Replace tool is now embedded in the History/Editor pane and no longer a separate floating window as in the previous version of APL64. The arrow located in the bottom right corner of the Find/Replace tool optionally places the tool in the upper right or lower right corner of the History/Editor pane. This modification enables the simultaneous display of the Find/Replace tool and the content of any docked or floating pane.

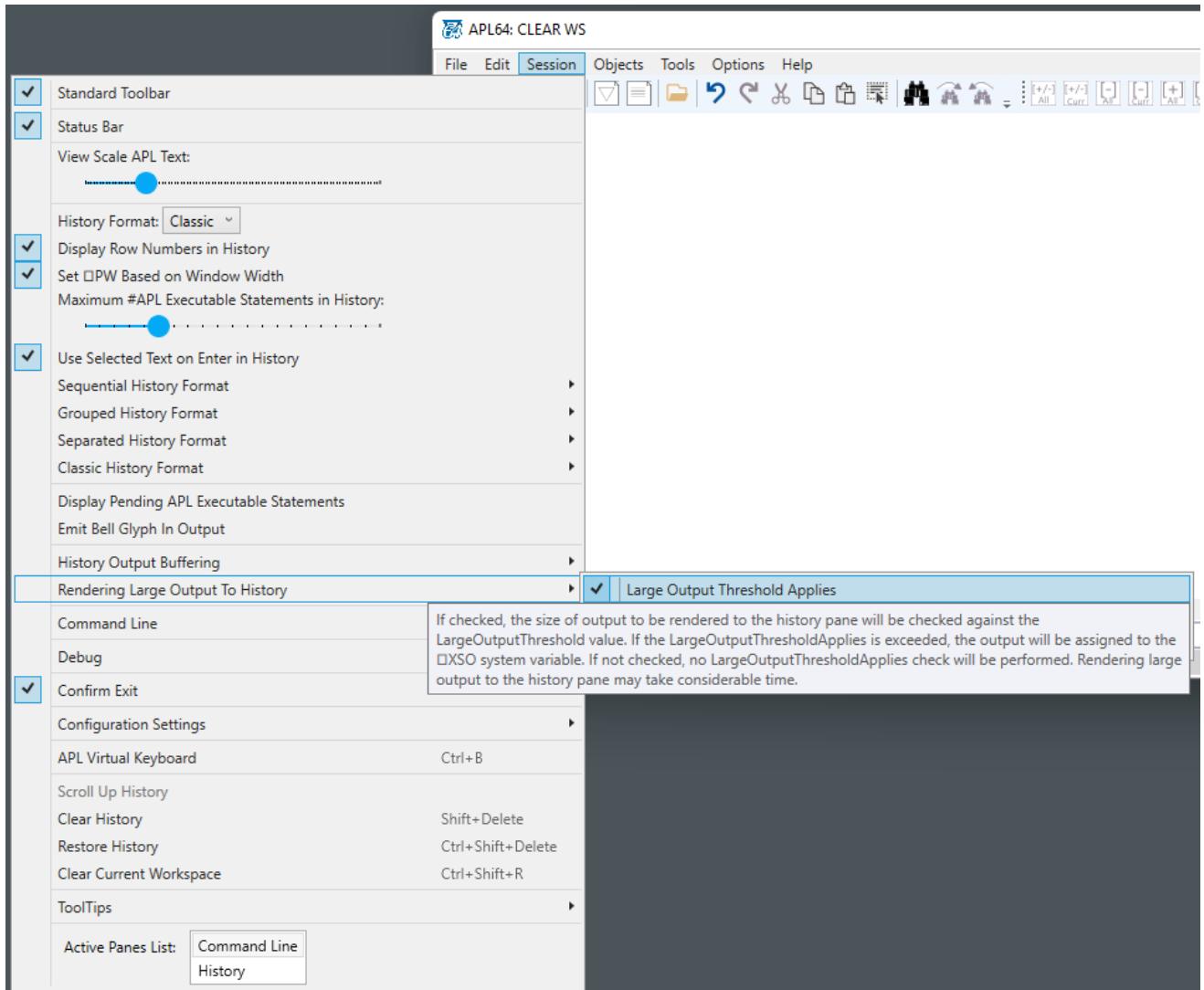
For example, in the editable classic history:



Rendering Large Output to History and xso System Variable

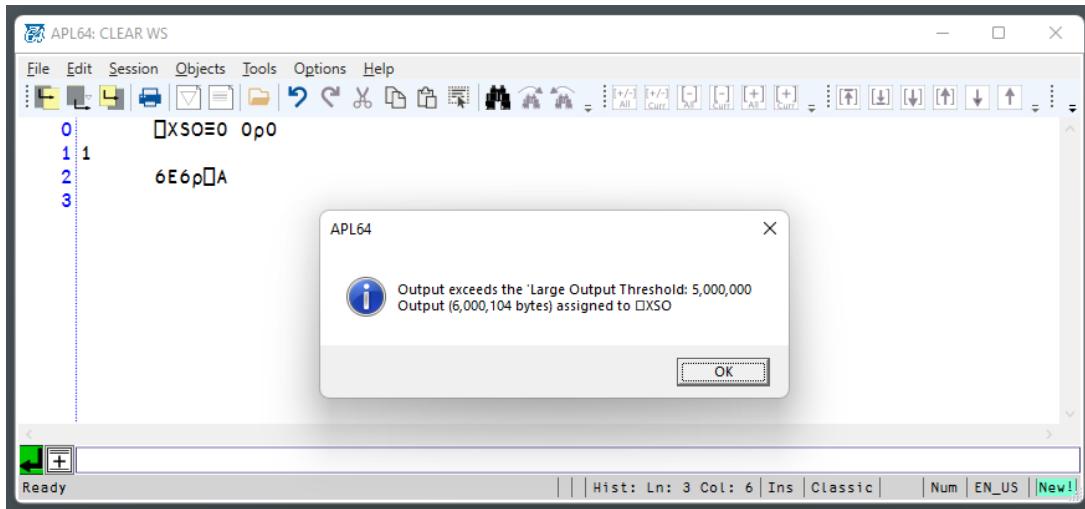
Very large variable sizes are possible in APL64 but formatting the values and rendering these values to the history pane may not be useful. A more practical approach is to assign the values to an APL64 variable and use the numerous APL features to reduce, format and render the data to the history pane in a form conducive to human analysis.

It is easy to forget that large data being prepared in APL64 would take considerable time to format and render to the history pane. If desired, to avoid such a delay, the APL64 Developer version user can take advantage of the **Session | Rendering Large Output to History** menu options:



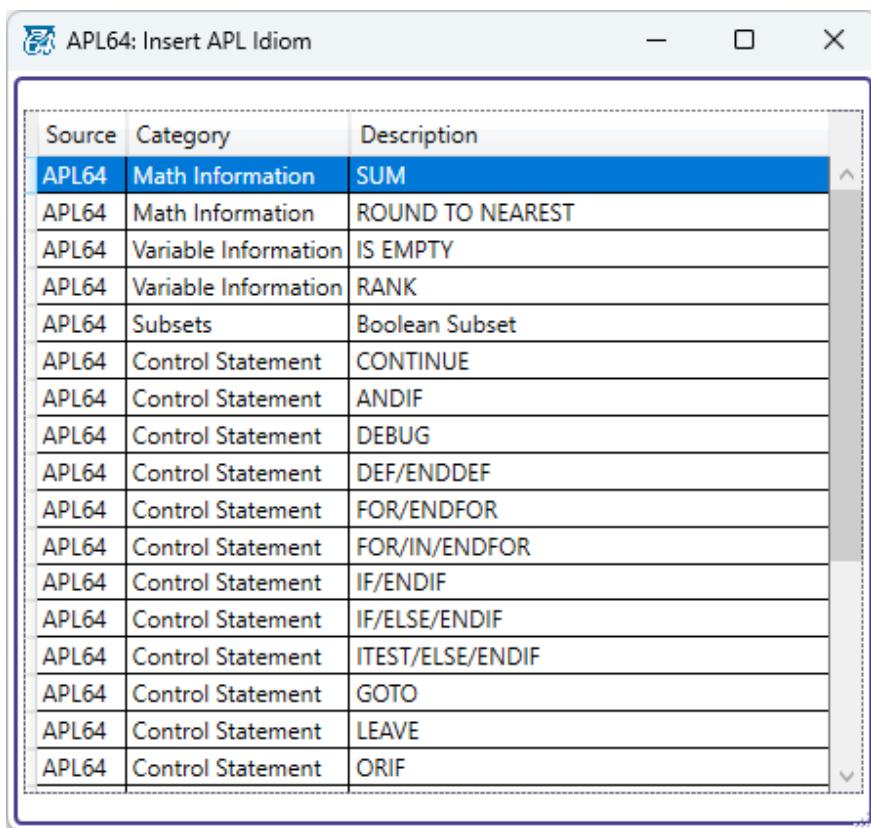
When the **Session | Rendering Large Output To History | Large Output Threshold Applies** is checked, the **Session | Rendering Large Output To History | Large Output Threshold** in units of 1E6 bytes will be used. If the size of the output to be formatted and rendered to the history pane exceeds the Large Output Threshold:

- A message will be presented indicating that the Large Output Threshold has been exceeded
- The output, prior to formatting, will be assigned to the □xso system variable



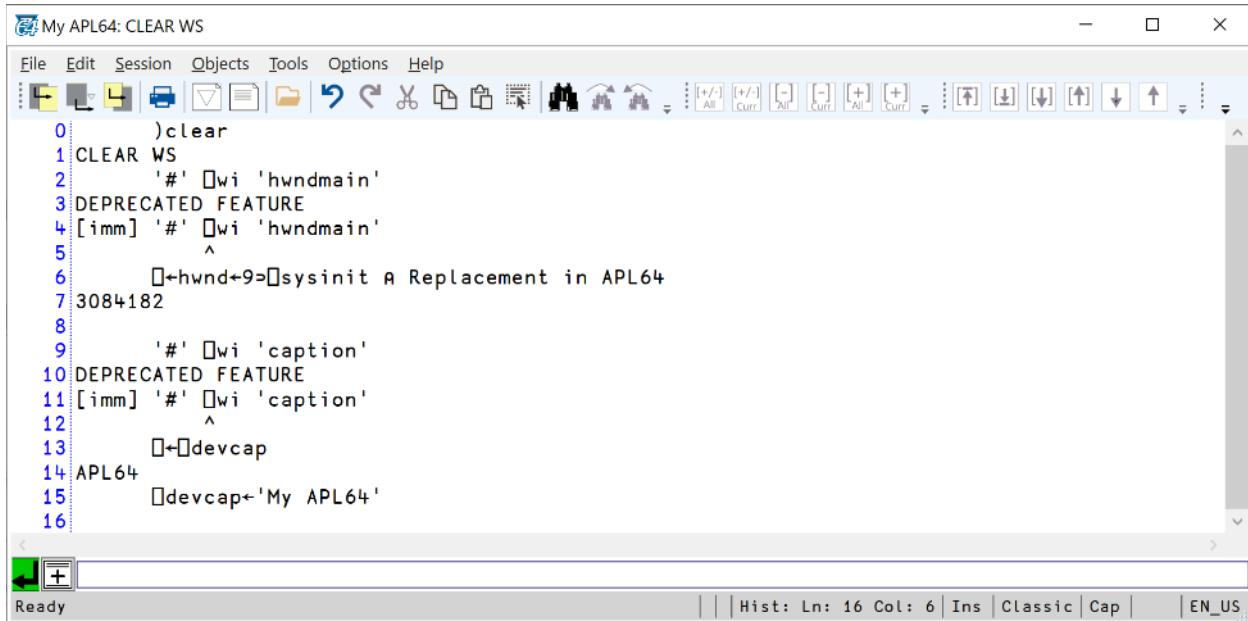
Insert APL Idiom Dialog is Now Not Modal

Use **Edit | APL Idioms | Insert an APL Idiom...** to present this dialog:



The Caption and hwndmain Properties for the □wi System Object (#) are Deprecated

The result is that they now return the message **DEPRECATED FEATURE**. Refer to the property names in the [Help | APL+Win Compatibility | Windows GUI Reference](#) menu for the recommended replacement system functions, □devcap and 9⇒□sysinit.



```
0 )clear
1 CLEAR WS
2 '#' □wi 'hwndmain'
3 DEPRECATED FEATURE
4 [imm] '#' □wi 'hwndmain'
5 ^
6 □←hwnd+9⇒□sysinit A Replacement in APL64
7 3084182
8
9 '#' □wi 'caption'
10 DEPRECATED FEATURE
11 [imm] '#' □wi 'caption'
12 ^
13 □←□devcap
14 APL64
15 □devcap←'My APL64'
16
```

New menu: Objects | Editors Pane Format | Shrink Editor Area When All Editors Are Floated

When this option is checked, the editor's area will shrink to increase the main window area available to the history pane when all editors are floated.

New menu: Objects | Editors Pane Format | Shrink Editor Area When All Editors Are Floated

When this option is checked, the debug area will shrink to increase the main window area available to the history pane when both the Debug and SI Panes are floated.

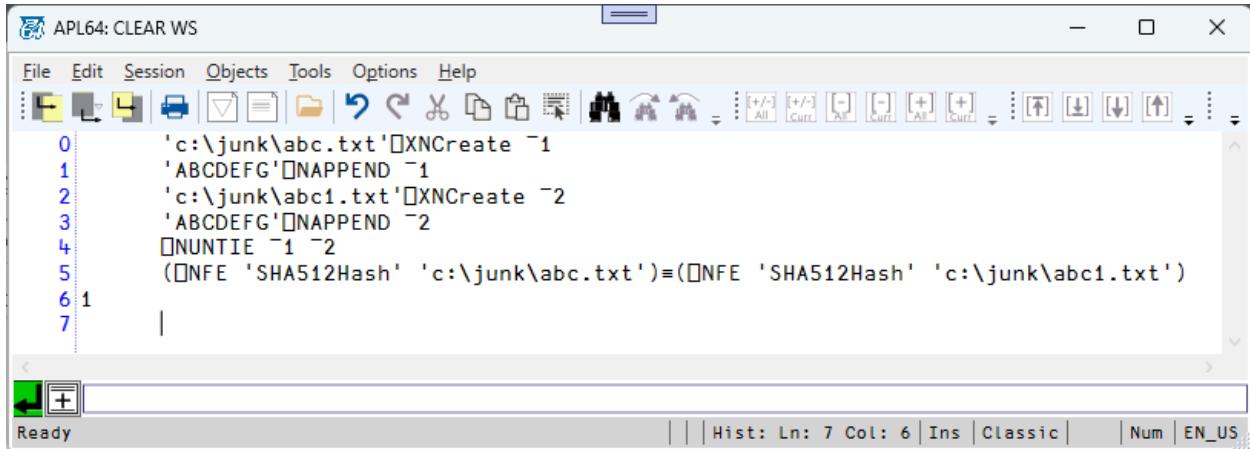
The □wi Interface to the APLNow32.exe was Optimized to Provide Better Stability.

For some □wi GUIs, with this optimization, the display time of the GUI will be less than or equal to APL+Win.

□wi GUIs which use User Defined Controls, are now operating properly.

New □NFE 'SHA512Hash' Action

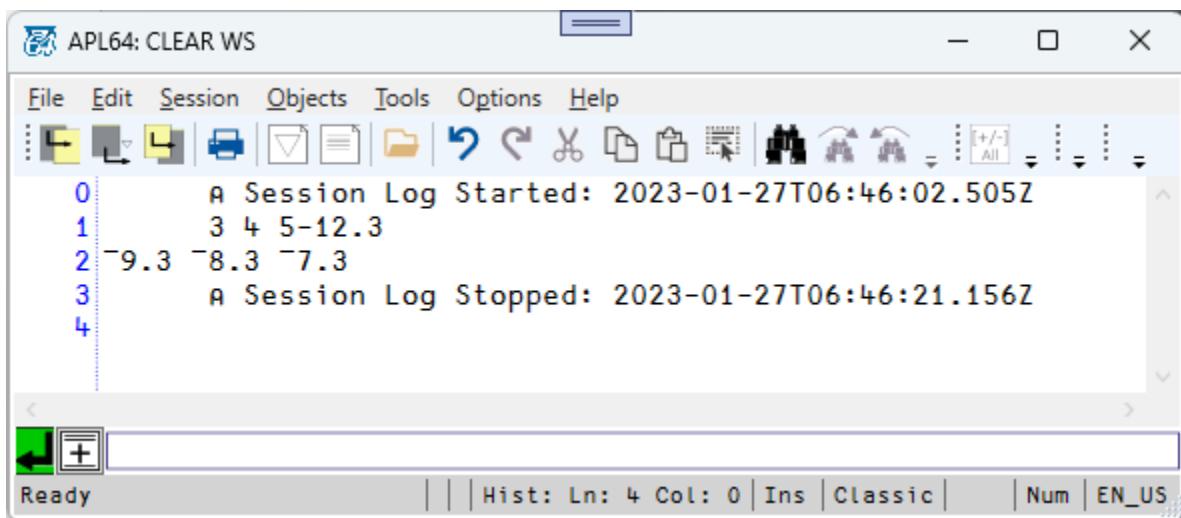
Purpose: To compare two file contents. Refer to [Help | APL Language | Using □NFE](#) for the details.



```
APL64: CLEAR WS
File Edit Session Objects Tools Options Help
[Toolbar]
0   'c:\junk\abc.txt'□XNCREATE ▒1
1   'ABCDEFG'□NAPPEND ▒1
2   'c:\junk\abc1.txt'□XNCREATE ▒2
3   'ABCDEFG'□NAPPEND ▒2
4   □UNTIE ▒1 ▒2
5   (□NFE 'SHA512Hash' 'c:\junk\abc.txt')=(□NFE 'SHA512Hash' 'c:\junk\abc1.txt')
6   1
7   |
< [+] Ready ||| Hist: Ln: 7 Col: 6 Ins | Classic | Num | EN_US
```

Session Log Timestamps Displayed

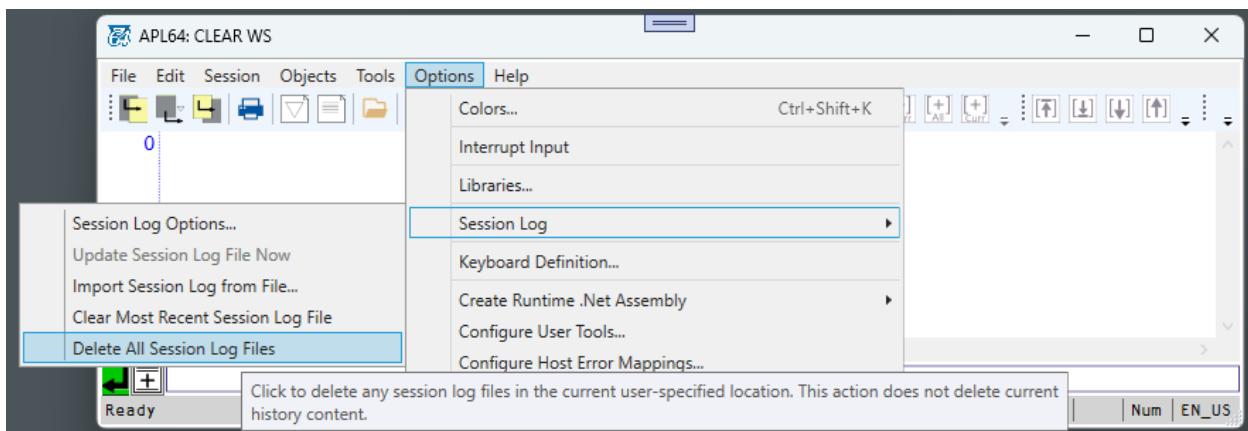
As in APL+Win, the session log start and end timestamps will display in the History log when the session log is enabled.



```
APL64: CLEAR WS
File Edit Session Objects Tools Options Help
[Toolbar]
0   A Session Log Started: 2023-01-27T06:46:02.505Z
1   3 4 5-12.3
2   ▒9.3 ▒8.3 ▒7.3
3   A Session Log Stopped: 2023-01-27T06:46:21.156Z
4
< [+] Ready ||| Hist: Ln: 4 Col: 0 Ins | Classic | Num | EN_US
```

New Options | Session Log | Delete All Session Log Files Action

This new menu will delete all session log files in the current user-specified log file location.



Variable and Function Names Can Begin With the Underscore Character

The underscore character (`_`) is permitted as the first character in the name of the object, which extends the available object names compared to APL+Win.

A screenshot of the APL64: CLEAR WS software window showing a session log. The log contains the following code:

```
0      _X+12
1      _X
2 1 2 3 4 5 6 7 8 9 10 11 12
3      ⌊DEF 'Z←_ADD10 X' 'Z←10+X'
4 _ADD10
5      ⌋vr '_ADD10'
6      ▽ Z←_ADD10 X
7 [1]   Z←10+X
8      ▽
9
10
```

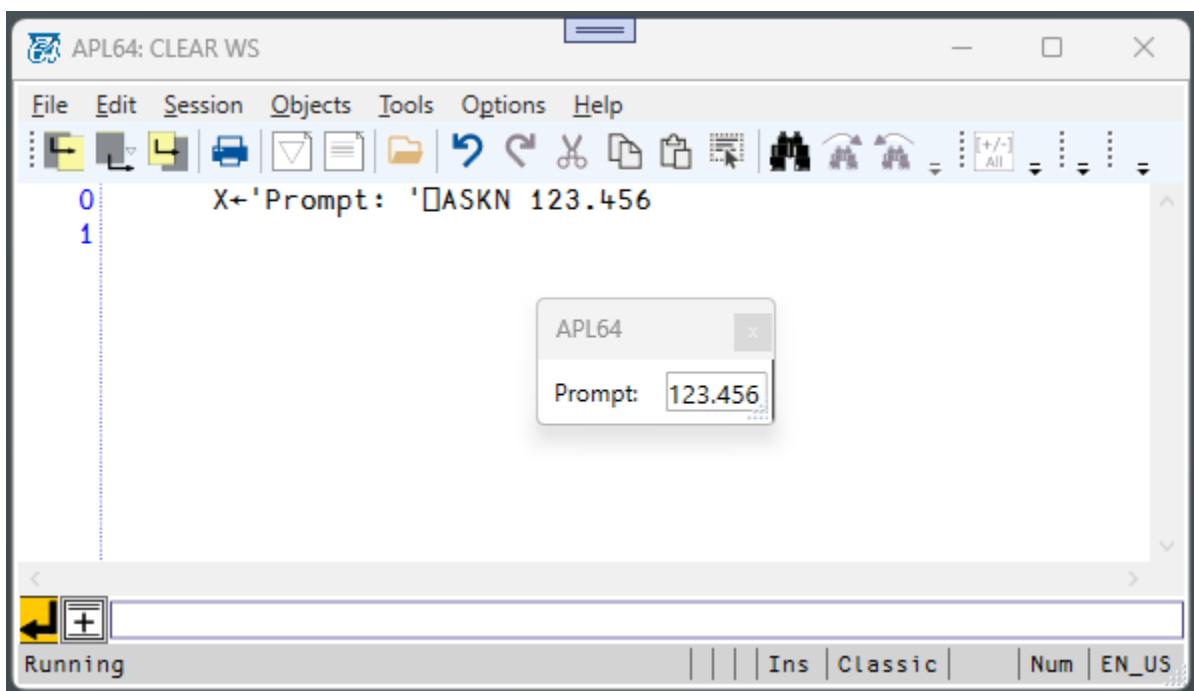
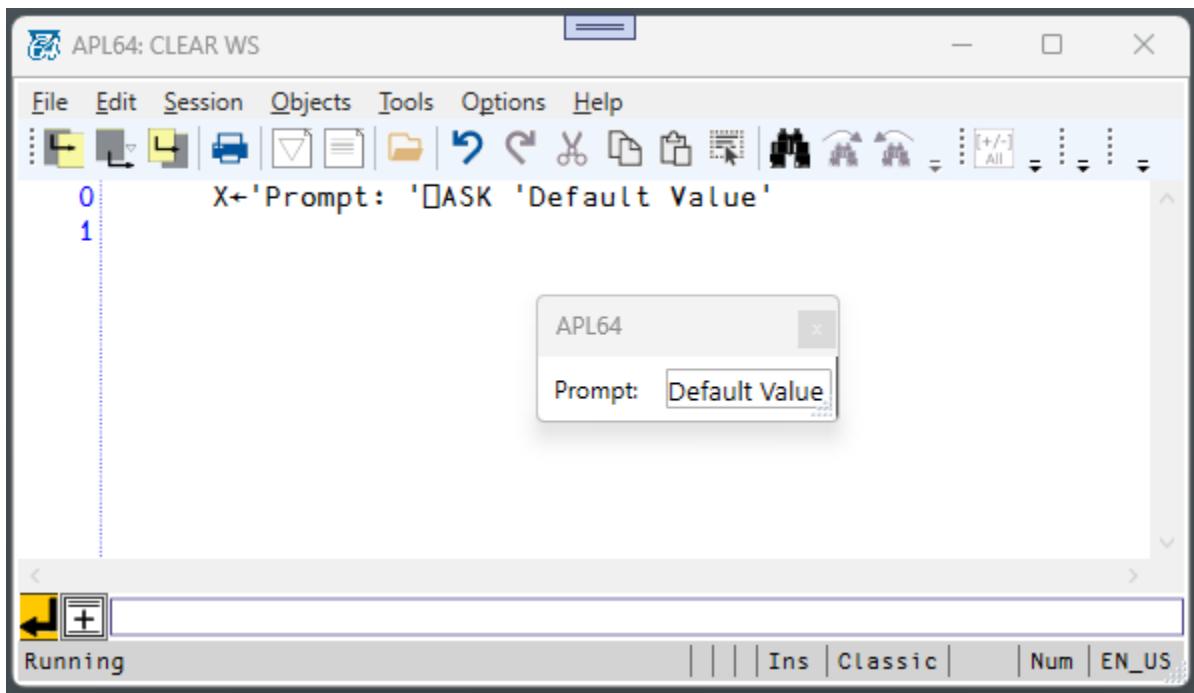
The status bar at the bottom shows "Ready" and "Hist: Ln: 10 Col: 6 Ins | Classic | Num EN_US".

New System Functions: `fflock`, `fflush`, `fnlock`, `fnflush`

Refer to the [Help | APL Language | System Functions](#) for the details.

New System Functions `ask` and `askn`

Optional replacements for the Quote-Quad input (`⌊`) and Quad input (`⌋`) system functions. Refer to the [Help | APL Language | Using ⌂ASK and ⌂ASKN](#) for the details.



Options | History Log | History Log Options Menu Renamed

This menu is now named Options | Session Log | Session Log Options... to more accurately reflects the action of the facility.

EDITEVENTS/ EDITINFO and Documentation Updated

New EditInfo action: GetEditEventsSubscriptions to obtain the subscriptions created using the EditEvents system function. Refer to [Help | APL Language | Using EditEvents and EditInfo](#) for detailed documentation.

SKD and Documentation Updated

- Resolved spurious exceptions when XML deserializing a SKD instance
- New SKD actions: ToFileDialog and FromFileDialog
- New menu: [Help | APL Language | Using SKD](#) for detailed documentation.

Performance of Rowfind System Function is Improved

The Display of the MessageBox Warning is Now Avoided When Importing an Empty Session Log File

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.

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](#)

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

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

New \square XL actions: SortRange and UsedRange

Refer to the Help | APL Language | Using \square XL for the details.

Updates to \square EDSS and)EDSS

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

New Sort Rows options for a user-selected range

New Print Option for the current worksheet selection

New Find & Replace Options

New \square EDITINFO action: DeleteEditEventsSubscriptions

\square EDITEVENTS 'PreviewObjectNameSelection' event simplified

Quad (\square) 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.

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

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.

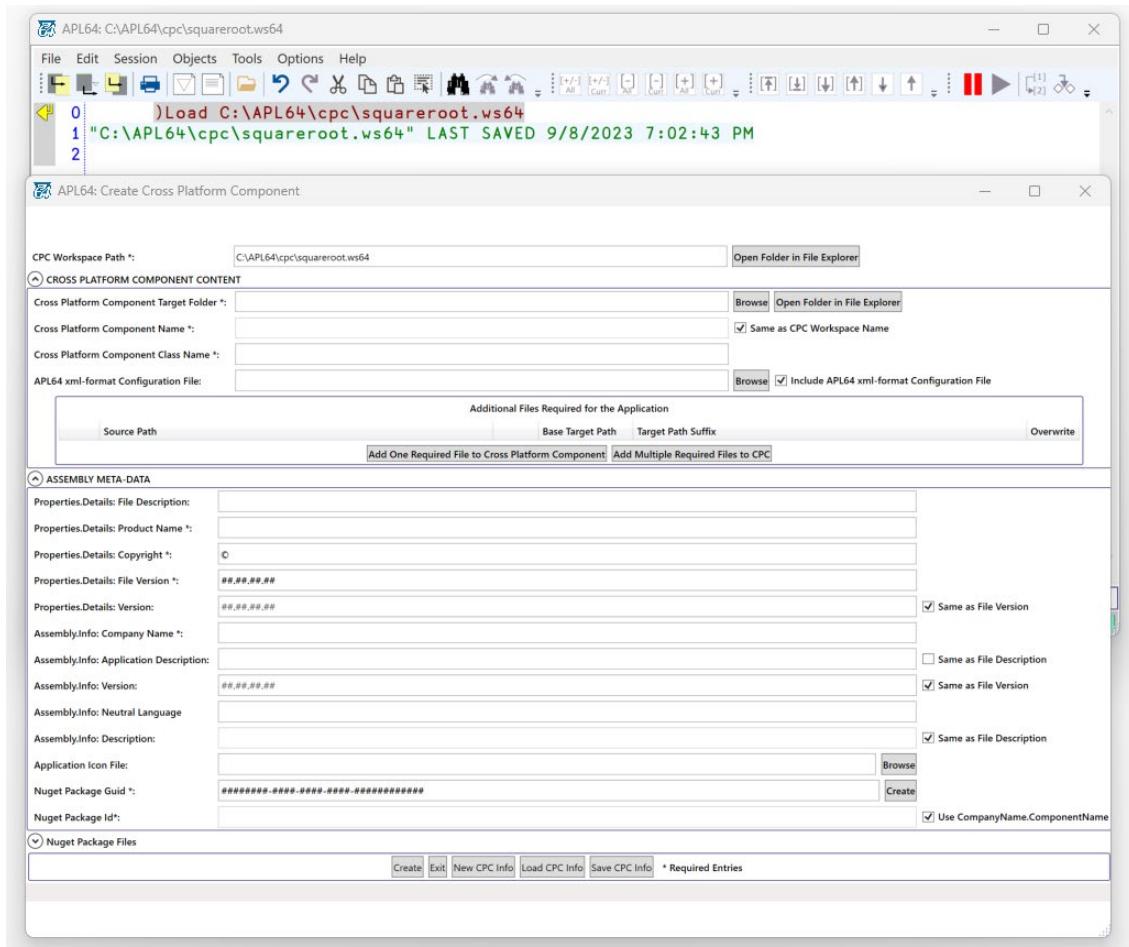
APL64 Installer Enhancements

The APL64 installer now installs the APL+Win fonts

Create a Cross Platform Component Utility

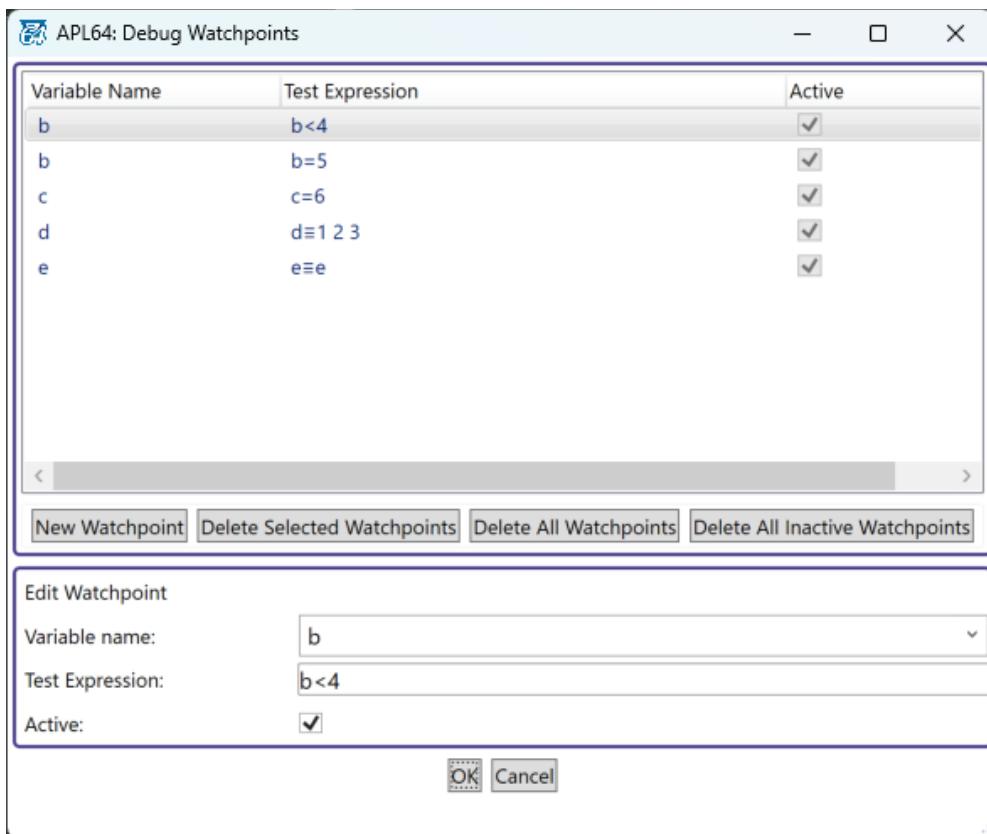
An APL64 cross-platform component is a [Nuget package](#) containing a [.Net Core assembly](#) which exposes one or more APL64 programmer-defined ‘public’ functions in the workspace. This package may be used as part of a containing .Net application to support the APL business rules or algorithms of the application. Any .Net programming language may be used to access the APL64 cross-platform component in a hardware and operating system environment which supports Microsoft .Net Standard v2.1.

In APL64, refer to the menu **Help | Developer Version GUI | Create Cross-Platform Component** for the documentation and the menu **Options | Create .Net Runtime Assembly | Create Cross Platform Component** for the GUI to create the cross platform component.



New debugging with Watchpoints dialog

The feature is functionally like APL+Win but with an updated modernized GUI interface that is accessed with the menu **Session | Debug | Edit Watchpoints (Ctrl+W)**. Refer to the **Debugging with Watchpoints** in the APL64 User Manual for the details.



New □watchpoint System Variable

Watch points are maintained in a system variable □watchpoints. Refer to the **Debugging with Watchpoints** in the APL64 User Manual for the details.

Syntax coloring and margin glyphs supported in editable classic history format

Syntax coloring and left margin glyphs are optionally available in the classic history pane. The new left margin delta glyph (Δ) indicates if a line in the classic history has been user-edited, but not executed. The enhancement is configured with the menu **Session | Classic History Format | Show Syntax Coloring When Editable**.

Note:

- The delta glyph will be displayed in the editable classic history for a blank line inserted with the Ctrl+Enter shortcut.
- The delta glyph will not be displayed in the editable classic history on the last line of the history if that line contains none or only space characters.
- The delta glyph will not be displayed in the editable classic history if a history line is user-edited and immediately executed and the line was not previously a delta line.

- Example: User editing of the output of an APL executable expression

User executes: `2 3⍴6` [Enter] in the classic history pane:

```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
[Icons]
0 2 3⍴6
1 2 3
2 4 5 6
3
Ready | Hist: Ln: 3 Col: 6 Ins Classic | EN_US | New!

```

User modifies rows 1 (replace 3 with 4) and 2 (replace 6 with 9). The left margin delta glyph marks rows 1 and 2 as user-edited rows:

```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
[Icons]
0 2 3⍴6
1 2 4
2 4 5 9
3
Ready | Hist: Ln: 2 Col: 6 Ins Classic | EN_US | New!

```

New Inline Comments: ☒ ☓

Inline comments are useful for “commenting out” multi-line blocks or inserted annotations in the middle of executable code. Inline comments begin and end with glyphs that look somewhat like sideways lamps: ☒ ☓

These glyphs are typed on the APL64 US keyboard as:

ALT+SHIFT+Z: ☉

ALT+SHIFT+X: ☊

Note: The keyboard short-cuts described in this document pertain to the EN-US keyboard definition in APL64. When using a different keyboard definition, some keyboard short-cuts may be different.

Wherever possible the APL glyph locations are the same for all APL64 keyboard definitions.

Unlike traditional comments, inline comments can optionally be terminated before the end of line using a closing comment “⊸” glyph.

If a single opening comment glyph is used without a closing comment glyph, it behaves like a traditional comment and runs to the end of line:

```
10 20 ☉ 30 40 50 60
```

```
10 20
```

But if you terminate the comment it stops before the end of the line:

```
10 20 ☉ 30 40 ⊸ 50 60
```

```
10 20 50 60
```

You can also use inline comments to temporarily change part of an APL statement when experimenting or debugging code.

For example, consider the following statement:

```
(X+Y) Foo A B C D E
```

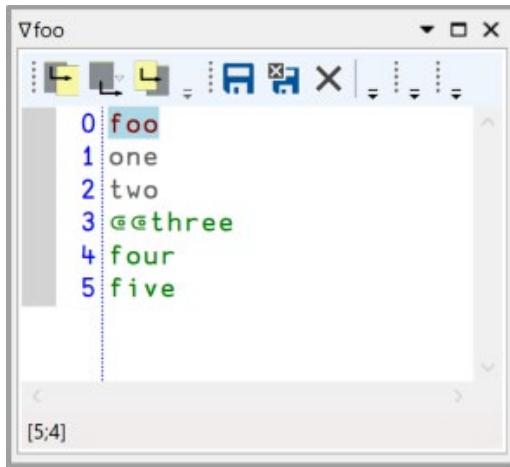
You might want to temporarily change this to something like the following

```
(X+Y) Foo A B T⊸C⊸ D E
```

Where argument C has been temporarily changed to T, but without discarding the original C character in the executable expression.

If you use two or more opening comment glyphs, the comment block can span multiple lines.

If you don't terminate the comment block, it runs to the end of the function, like this:



New Ctrl+F3 keyboard shortcut to search for a token under the caret

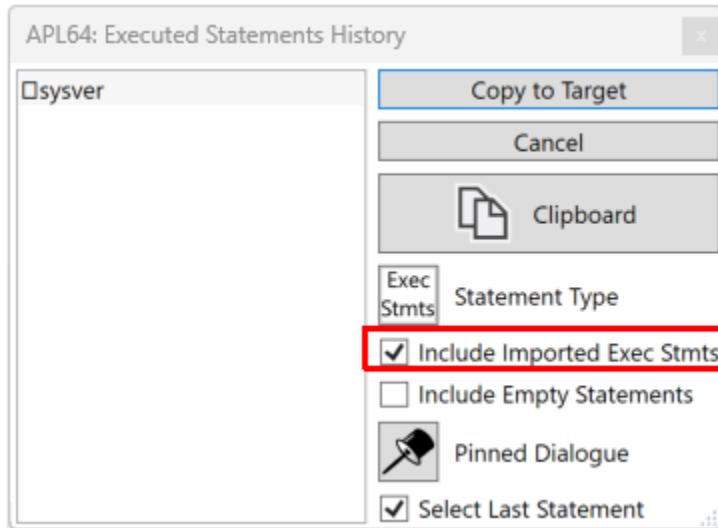
This feature is similar to APL+Win where the search is performed on the selection and the token under the caret when there is no selection.

[executed] statement history dialog

The [executed] statement history is now retained between APL64 sessions

New Include Imported Exec Stmts option in the Executed Statements History dialog

Check this option to include executed statements imported from the log file, if any, when the **Exec Stmt**s statement type is selected.



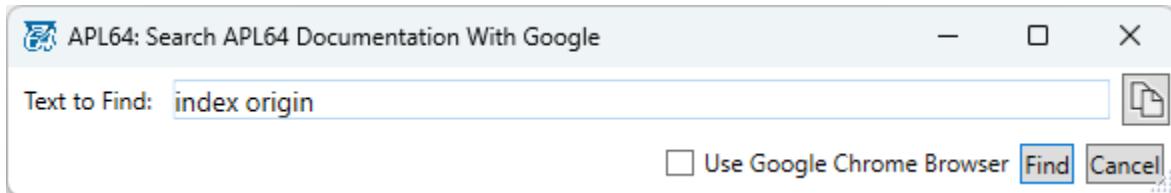
New options available for searching the APL64 user documentation

Menu Help / Search All Documentation with Adobe

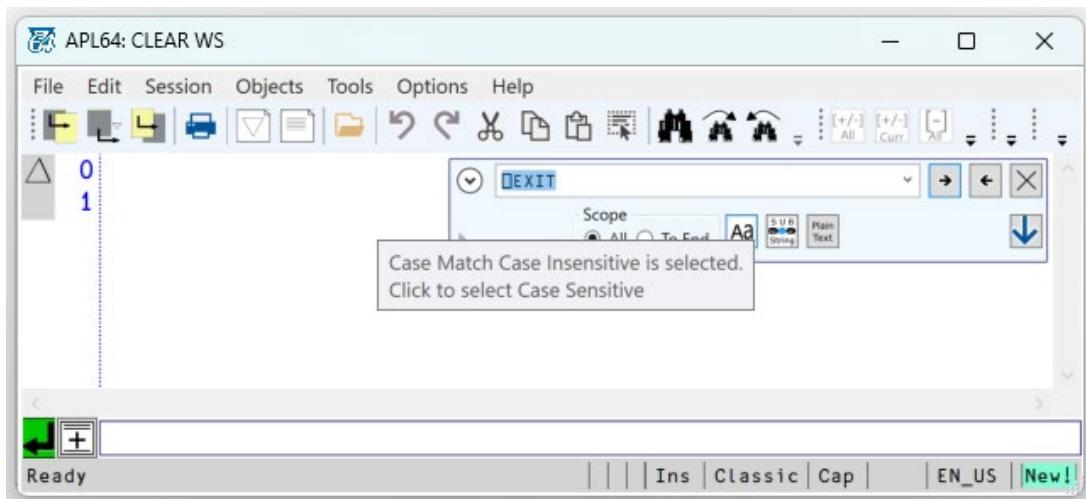
Presents the online document “APL64 Developer Version User Documentation Wide Scope Search” with instructions on using the Advanced Search option in the Adobe Acrobat Reader to search within multiple Adobe documents that are stored on your machine.

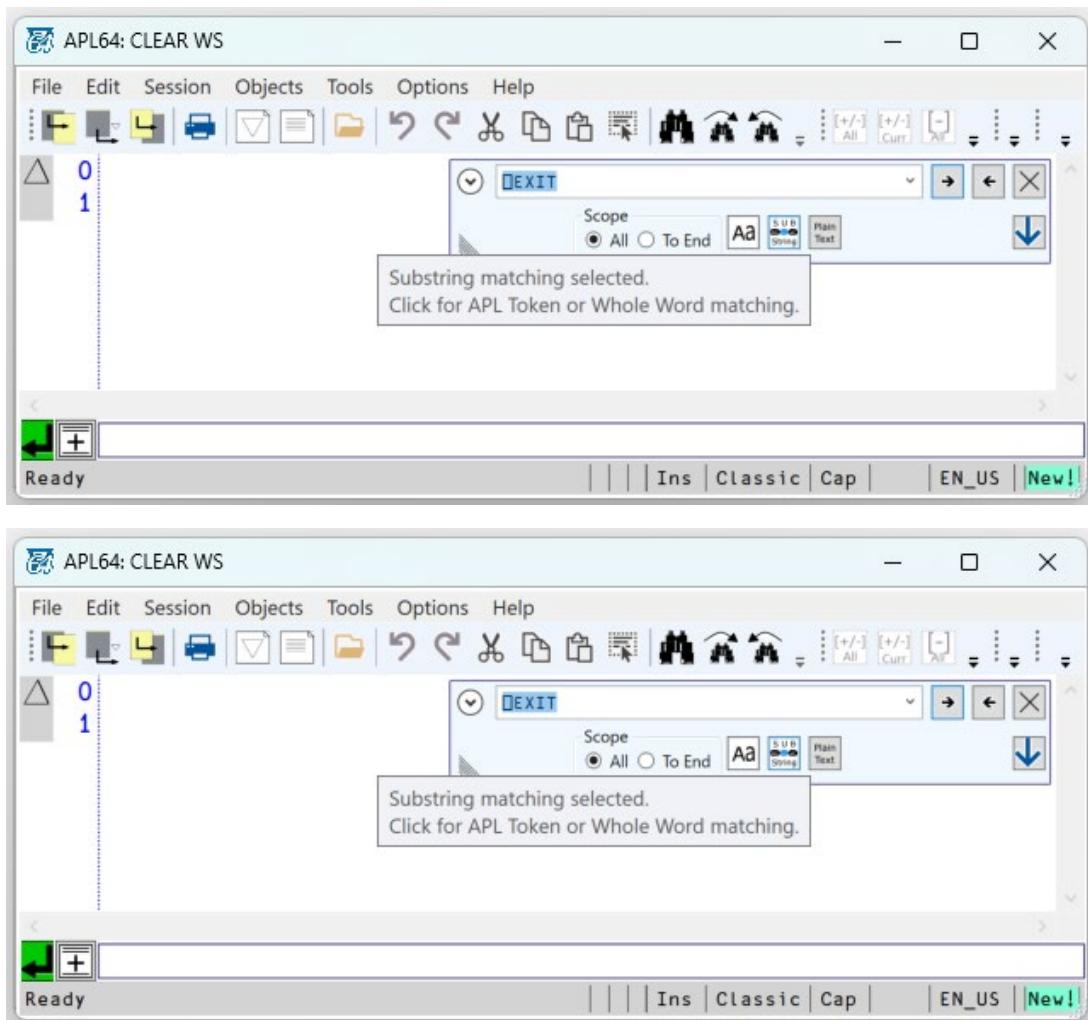
Menu Help / Search All Documentation with Google

Uses the Google search engine to find user-selected text in the APL64 Developer version documentation on the APL64 server:



Find/Replace dialog button tooltips updated to clarify their intended purpose





The Configure User Tools dialog Listview now supports Delete key actions

Restart immediately when the OK button is clicked in the Apply Defaults Settings dialog

When selecting menu **Session | Configuration Settings | Apply Default Settings**, and clicking OK in the Apply Defaults Dialog, APL64 will immediately restart even with the menu **Session | Confirm Exit** setting is checked.

Editing large functions performance improved

Performance improved in function editor for large functions with thousands of rows of code.

Session Log loading performance improved

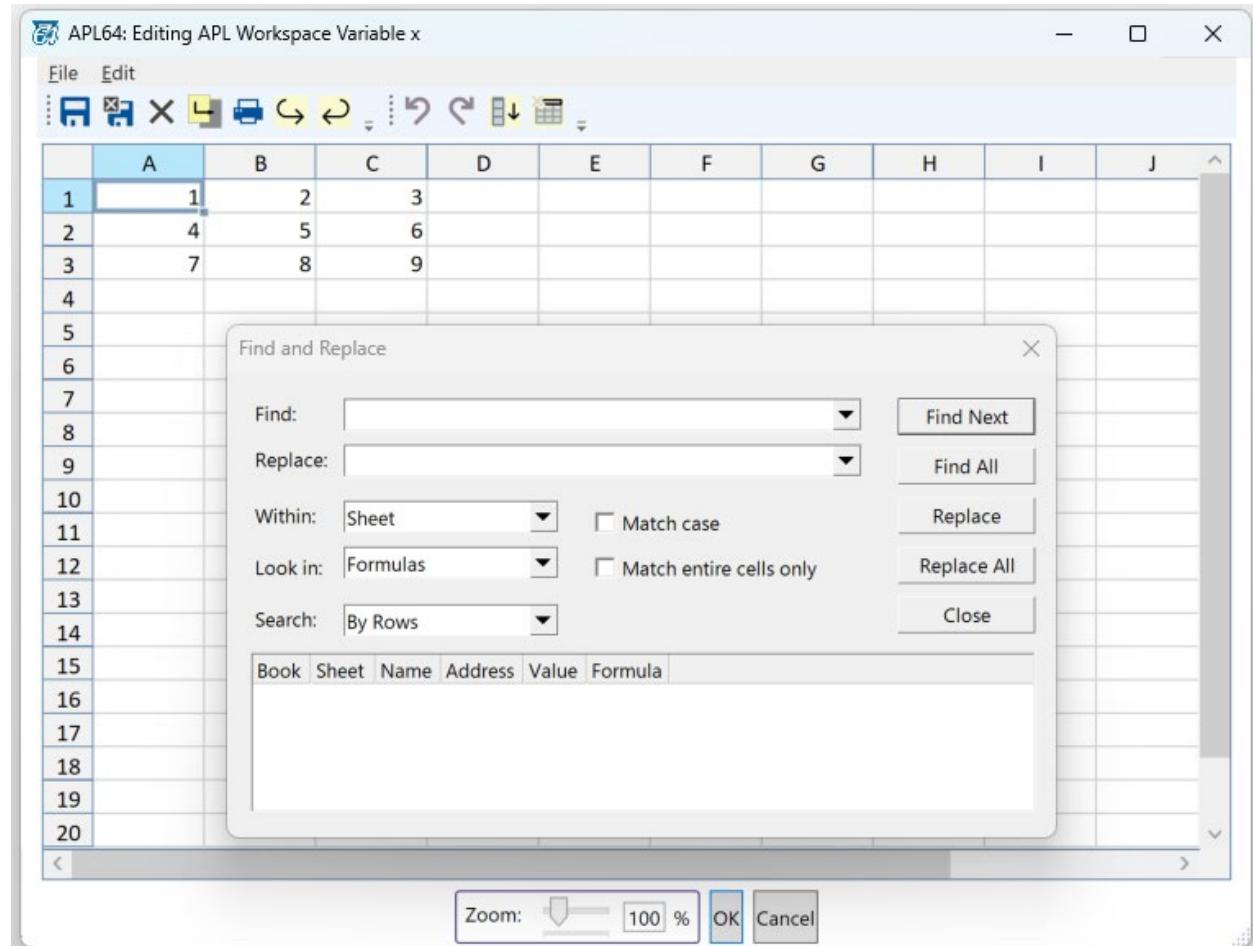
Performance improved in the history display at startup when loading a very large session log.

)EDSS and □EDSS Improvements

New Find and Replace dialog

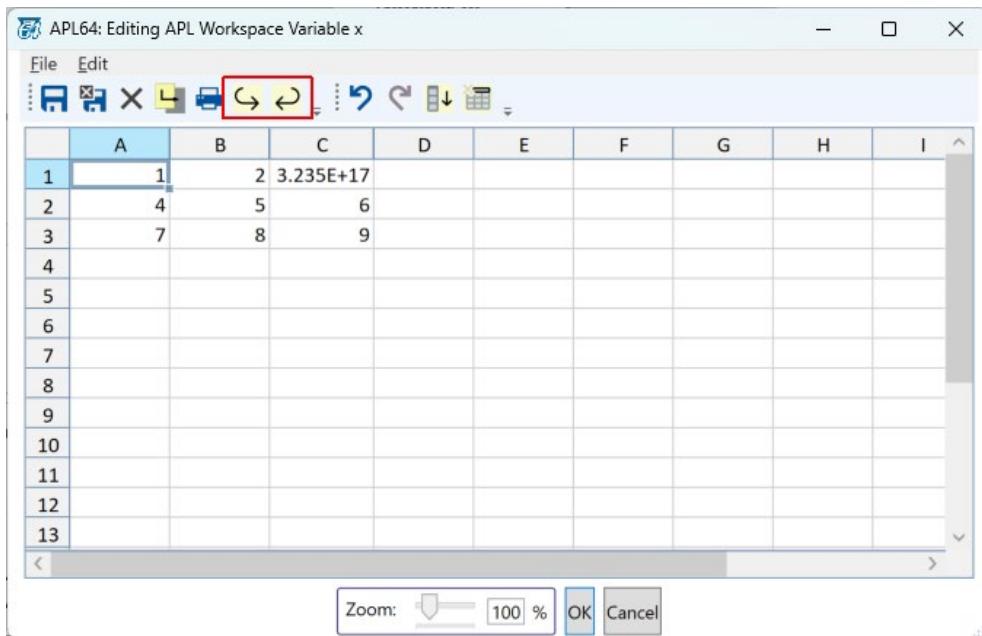
The Ctrl+F keyboard shortcut now displays the Excel-like Find and Replace dialog of SpreadsheetGear.

Note: As with Microsoft Excel, F3 and shift+F3 do not perform the actions find next and find previous, respectively.



New Import worksheet and Export data to variable actions

New Import worksheet and Export data to variable actions

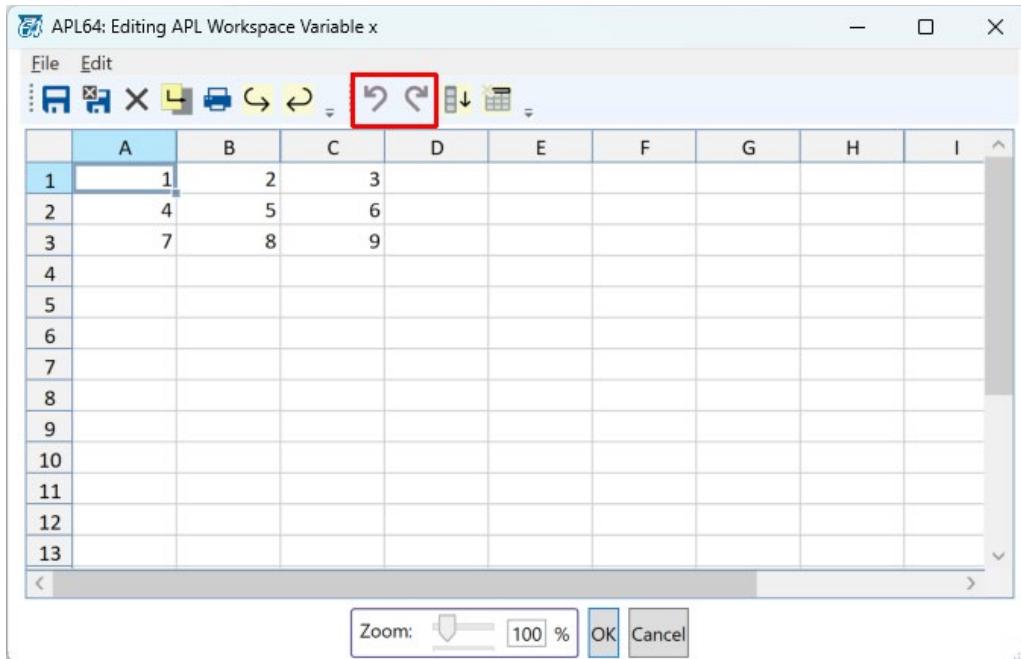


↳ This toolbar button selects the worksheet to import from an existing workbook.

↳ This toolbar button exports the data in the active worksheet as an APL variable to the workspace.

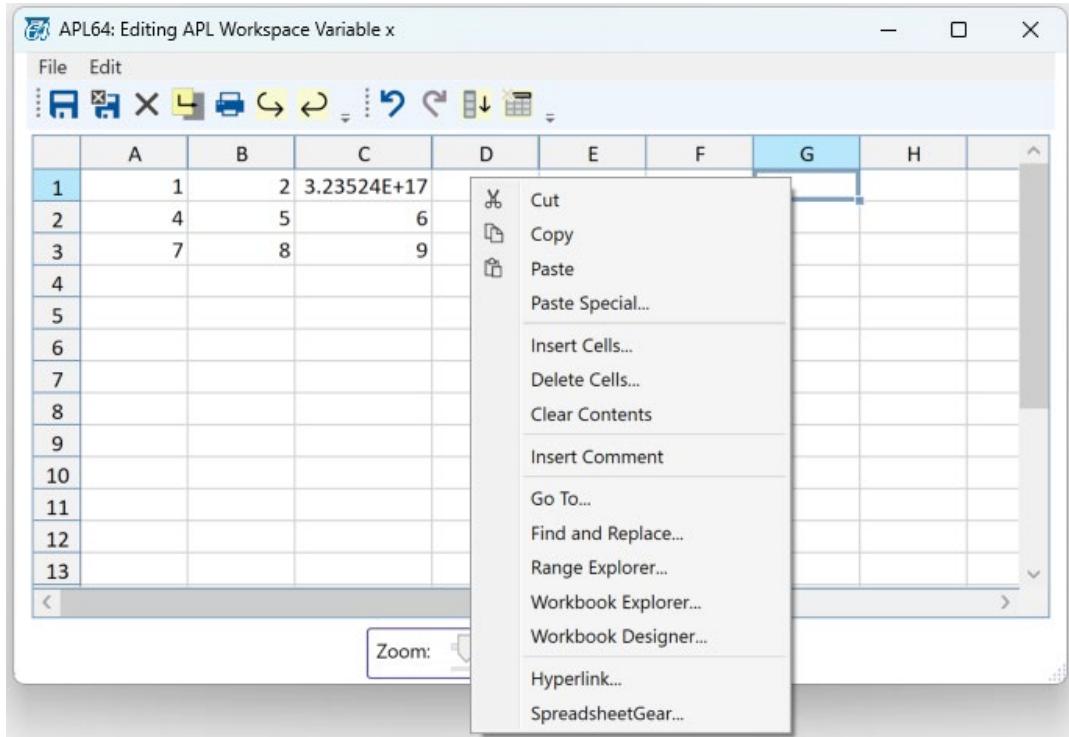
New Undo and Redo actions

The undo and redo actions for the most recent cell edits in the active workbook are supported. They're accessible from the toolbar buttons, identified in the red rectangle in the image below, and keyboard shortcut keys, Ctrl+Z and Ctrl+Y, respectively.



New context menu items implemented

- Go To...
- Find and Replace...
- Workbook Designer...



Use Autofit for worksheet rows

EDSS Limits

The maximum number of worksheets, columns and rows are the same as Microsoft Excel.

Session | Separated History Format menu: Tooltips updated

SQLx system functions Updated

- SQL Synonym datatype 'Int32' for 'Integer' added for the ExecInsertQuery' action
- SQL User documentation updated with additional examples
- SQLMicrosoft.Data.SqlClient for SQL .Net component updated to v5.1.1
- SQLMy: MySql.Data for SQLMy .Net component updated to v8.0.33
- SQLite: Microsoft.Data.Sqlite for SQLite .Net component update to v7.0.5

NFE Enhanced

New actions: 'ReadAllLines', 'ReadAllText', 'WriteAllLines', 'WriteAllText'

Improved exception messages for 'Read' and 'Write' actions

MATHNET interface to MathNet linear algebra toolkit implemented

Refer to Help | API Language | Using MATHNET for details

Successfully load previously damaged workspaces.

Correctly reconstruct the SI stack when loading suspended or damaged workspaces

wi Improvements

This especially pertains to workspaces that included WI, NI, WCALL callouts or callbacks suspended on the SI stack.

APL64 Colossal file functions were ported to APL64 from APNow32

Benefits of this modification are:

- Use native data serialization to support new APL64 data types such as string and char
- Add support for case sensitive file systems in cross-platform-targeted applications.

□tt32 System Function: High-Resolution Timer Tick Count for APNow32 operations

Purpose: Returns the number of ticks for all APNow32 operations (□wi and □wcall) since the start of the APL64 session to the time the value of □tt32 was obtained.

Refer to the topic □tt32 in the Help | APL Language | APL64 System Functions in APL64 for the details.

□ZIP system function

The □ZIP APL64 system function is an interface to the cross-platform compatible .Net System.IO.Compression tools. The □ZIP system function actions involve zip-format files. Since file operations are involved, the user's workstation credentials must be appropriate for the actions to be performed by □ZIP.

Refer to the topic □zip in Help | APL Language | Using □ZIP in APL64 for the details.

System variable □SYS[21] value is 1 when the APL64 is runtime

When APL64 is running as a WRE (Windows runtime executable) or a CPC (Cross platform), the value of □SYS[21] is 1.

)RSAVE system command now reports DEPRECATED FEATURE

)RSAVE system command is deprecated in APL64. This message is more descriptive than INCORRECT COMMAND. Use the Options | Create .Net Runtime Assembly | Create Windows Runtime Executable utility instead.

□ROWFIND performance improved when the left and right arguments are in □AV

□SKD: ToFile/FromFile actions removed

Microsoft has deprecated these actions, so they're no longer supported in □SKD. Use the FromFileDialog and ToFileDialog □SKD actions instead. Before updating APL64, use the □SKD 'FromFile' action followed by the □SKD 'ToFile' action to convert any stored □SKD dictionaries to the XML format.

STRING updated

- String '?' documentation summary corrected
- A String action which accepts a String left argument will also accept a rank 0 or 1 character value left argument, which is converted to a String.
- A String action which accepts a String[] vector left argument will also accept a vector left argument with elements which can be a string scalar or rank 0 or 1, character values which are converted to String.

WA operates faster

The enhancement speeds up the performance of wa to estimate available workspace memory.

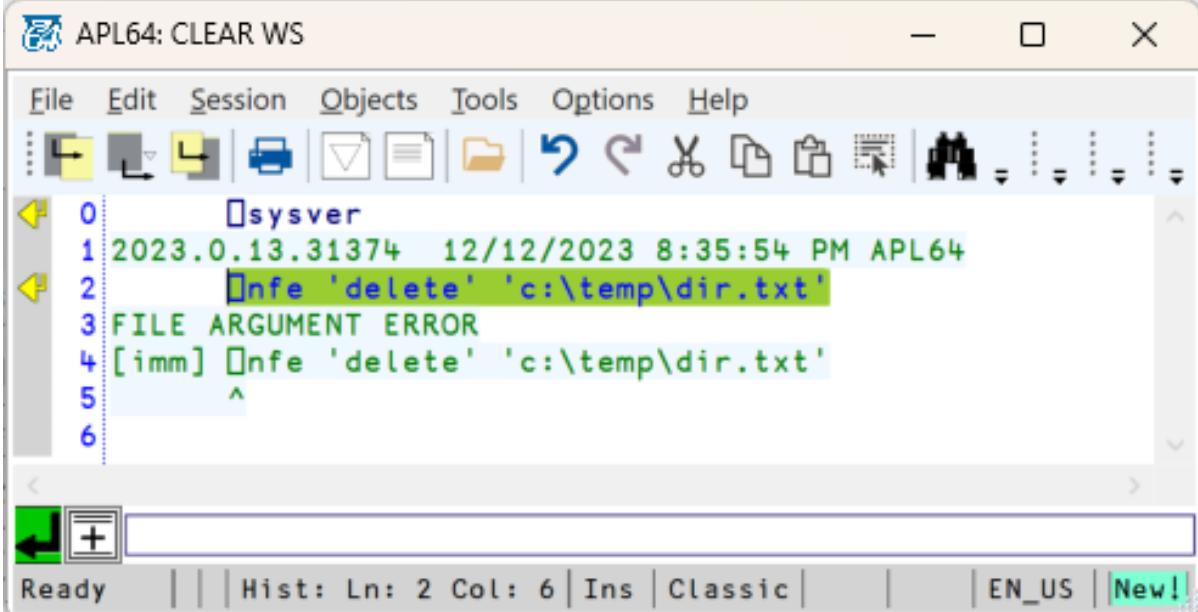
Enhanced wa to avoid doing garbage collection (workspace compaction) to estimate available workspace memory

ARBOUT and PR system variables now report DEPRECATED FEATURE

arbout and pr are legacy system variables are deprecated in APL64. In APL64 the result of Quote-Quad query or input never includes the programmer-defined prompt text.

nfe 'delete' action exception message improved

Behavior in the prior version:

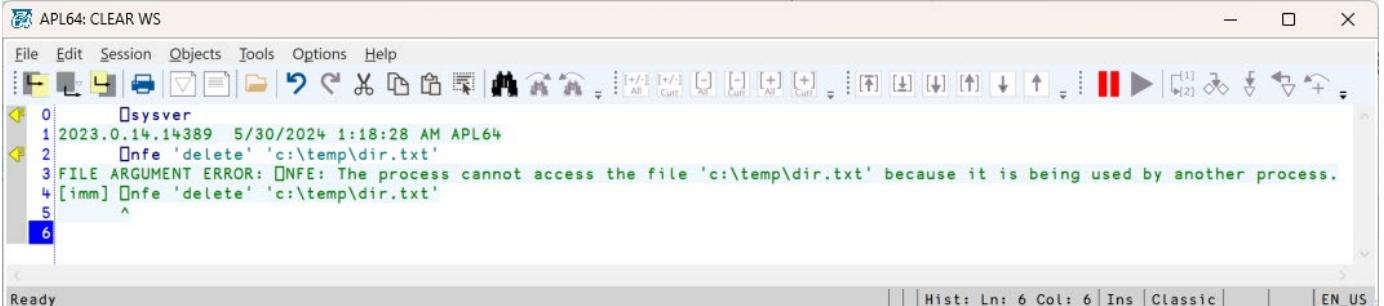


The screenshot shows the APL64: CLEAR WS application window. The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations. The main pane displays the following log entries:

```
0 sysver
1 2023.0.13.31374 12/12/2023 8:35:54 PM APL64
2 nfe 'delete' 'c:\temp\dir.txt'
3 FILE ARGUMENT ERROR
4 [imm] nfe 'delete' 'c:\temp\dir.txt'
5 ^
6
```

The entry at line 2 is highlighted in blue, indicating it is the current selection.

New behavior in this version:



The screenshot shows the APL64: CLEAR WS application window. The menu bar, toolbar, and main pane are similar to the previous version. The log entries are as follows:

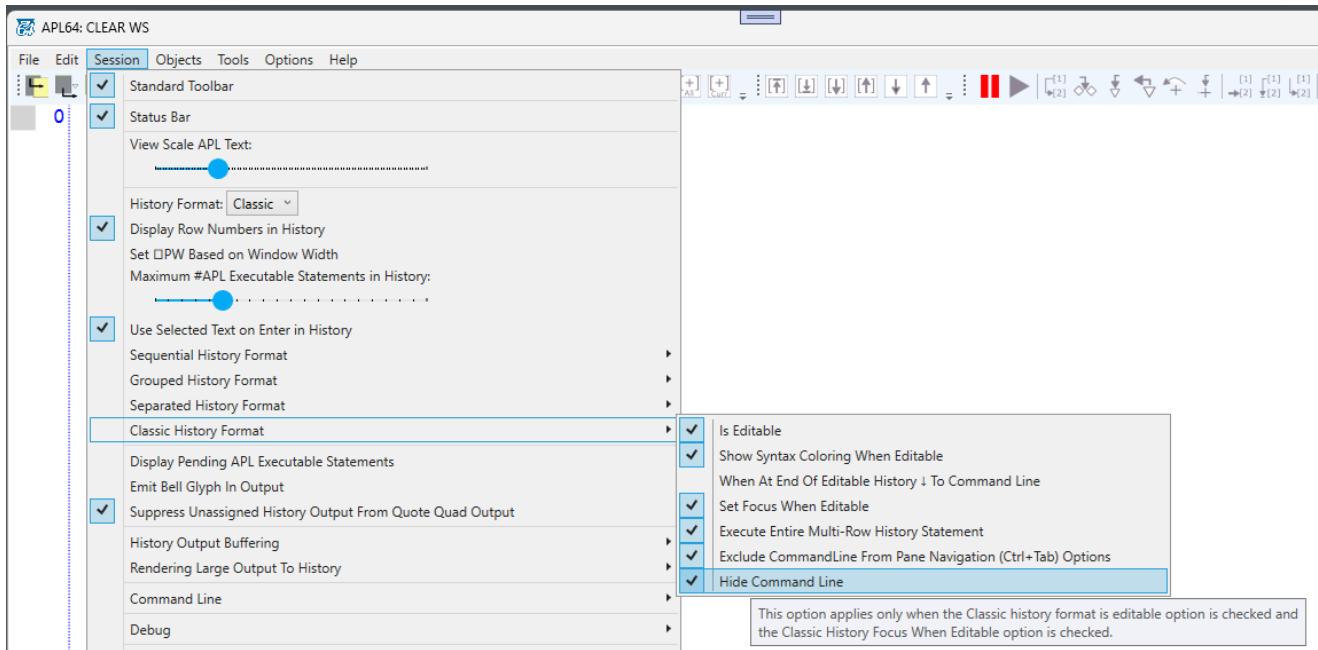
```
0 sysver
1 2023.0.14.14389 5/30/2024 1:18:28 AM APL64
2 nfe 'delete' 'c:\temp\dir.txt'
3 FILE ARGUMENT ERROR: NFE: The process cannot access the file 'c:\temp\dir.txt' because it is being used by another process.
4 [imm] nfe 'delete' 'c:\temp\dir.txt'
5 ^
6
```

The entry at line 3 is highlighted in blue, indicating it is the current selection. The error message is displayed in red text.

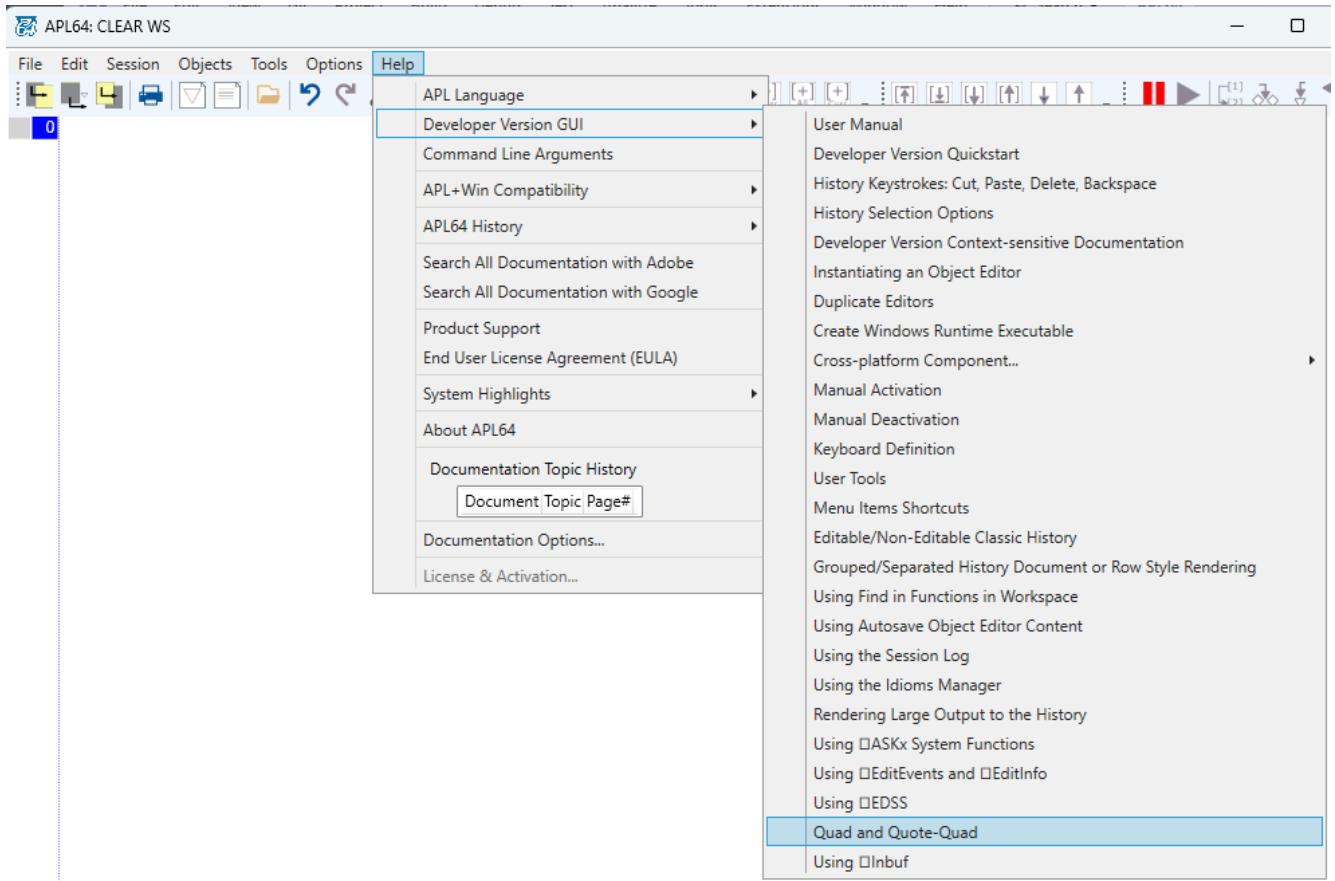
Option to Hide APL Command Line

Initial versions of the APL64 developer version Quad and Quote-Quad input were dependent on the Command Line in the APL64 Developer version. This dependency has been eliminated, so that the APL64

programmer may hide the Command Line (only when the editable Classic history format is used) by enabling the menu item **Session | Classic History Format | Hide Command Line**.



Providing an option to suppress the command line required modifications to the behavior of quad and quote-quad in APL64. Review the **Help | Developer Version GUI | Quad and Quote-Quad** programmer documentation for important information and examples.

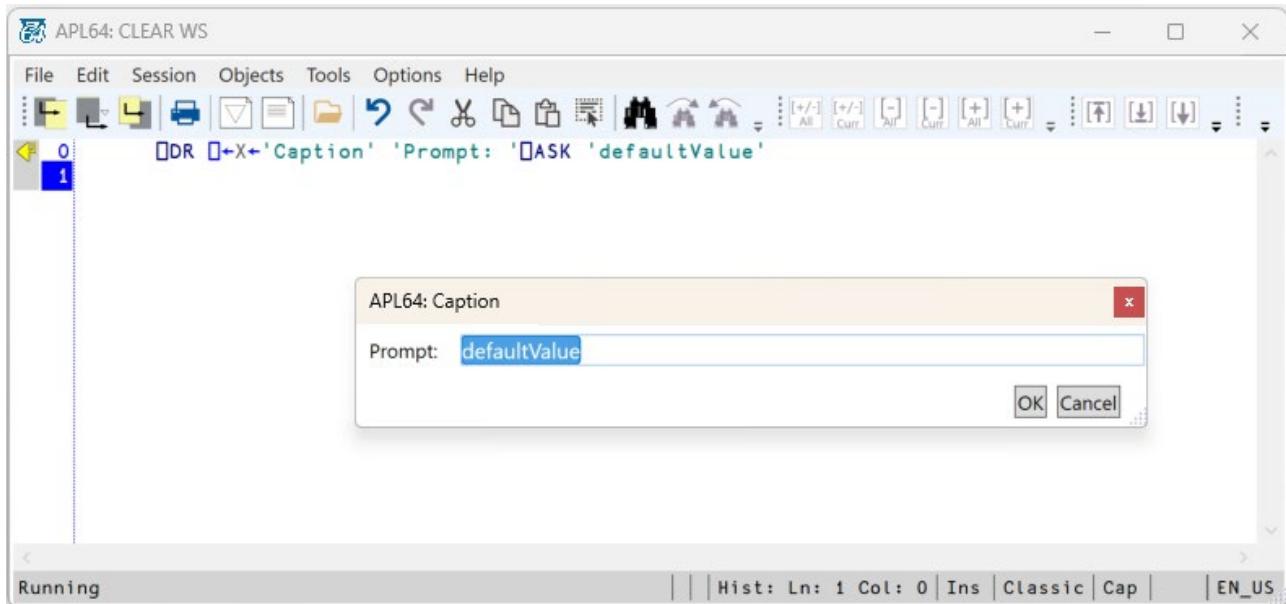


Option to suppress callback information in the history pane

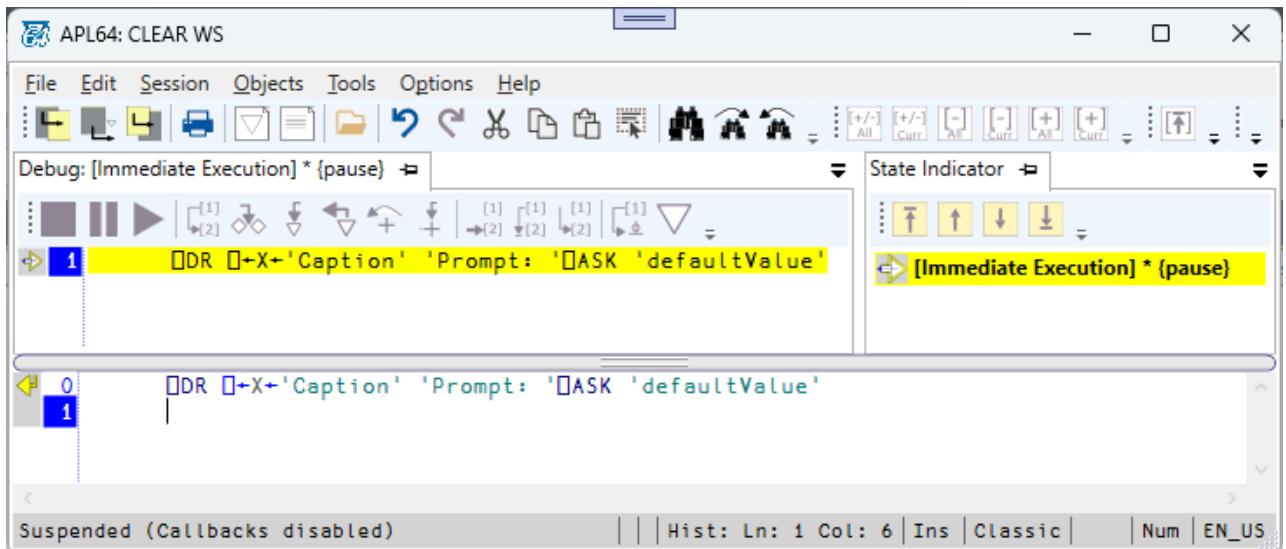
Use the APL64 system command 'output nocallback' to suppress callback information in the APL64 developer version history pane. In APL64 callbacks generate information with an `>[object;event]` prefix on the executable statement which caused the callback. Displaying this cause-and-effect information maintains an accurate history of what occurred in an APL64 instance. Callbacks include execution of a user-entered APL executable statement, interpreter callbacks, invocation of a user-defined tool and `□wi` events.

Cancel option for Quad-input, Quote-quad input, and `□ASKxx` dialogs
`□DR □←X←'Caption' 'Prompt: '□ASK 'defaultValue'`

Before the user closes the dialog:



After the user closes the dialog using the Cancel button:



Bookmarks are supported in the Editable Classic History Format

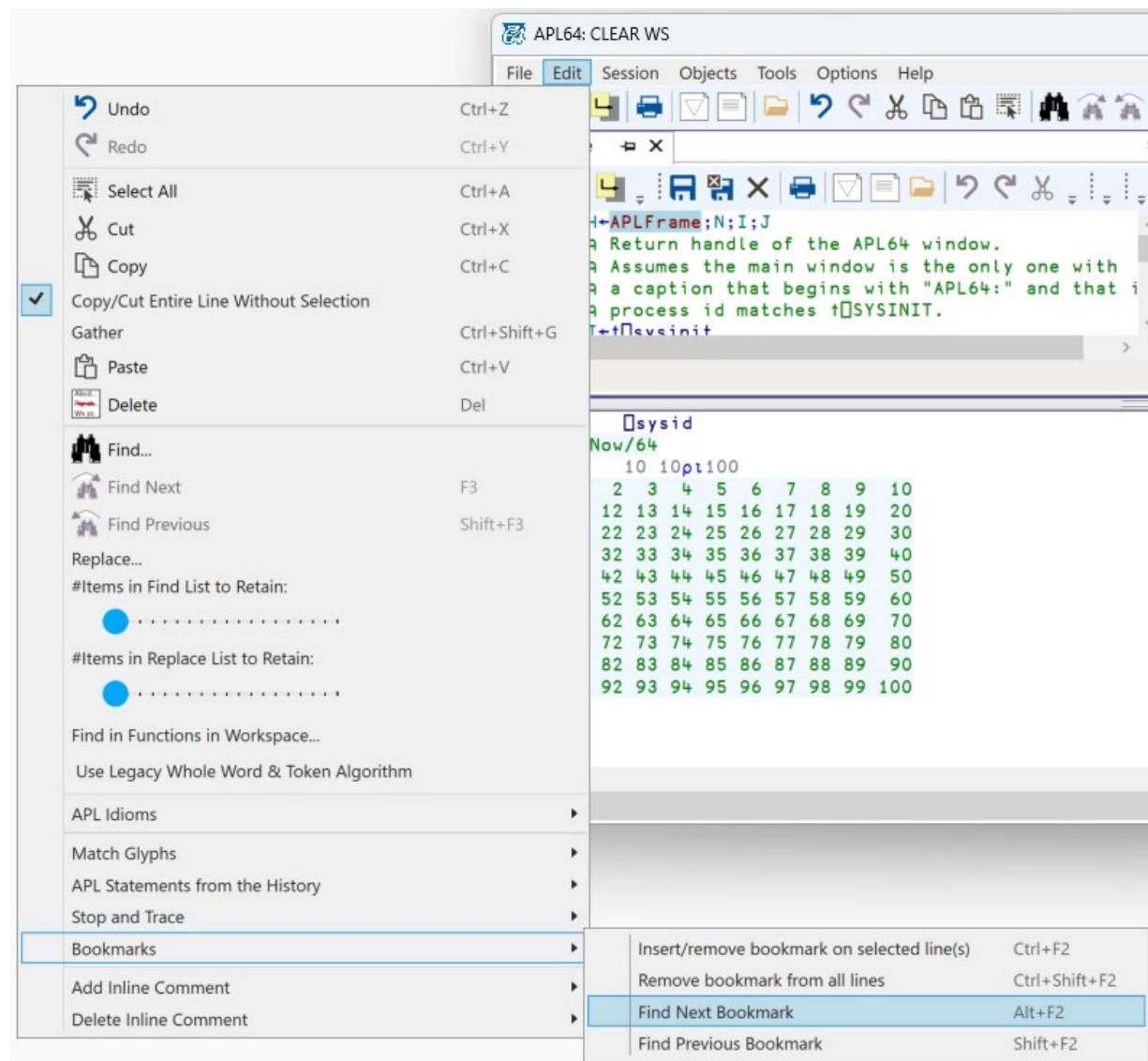
This feature is available when enabling the menu item **Session | Classic History Format | Show Syntax Coloring When Editable**.

The editable classic History supports the Session | Use Selected Text on Enter in History menu

This setting controls the behavior for selected and unselected text in the editable classic history when Enter or Return is pressed.

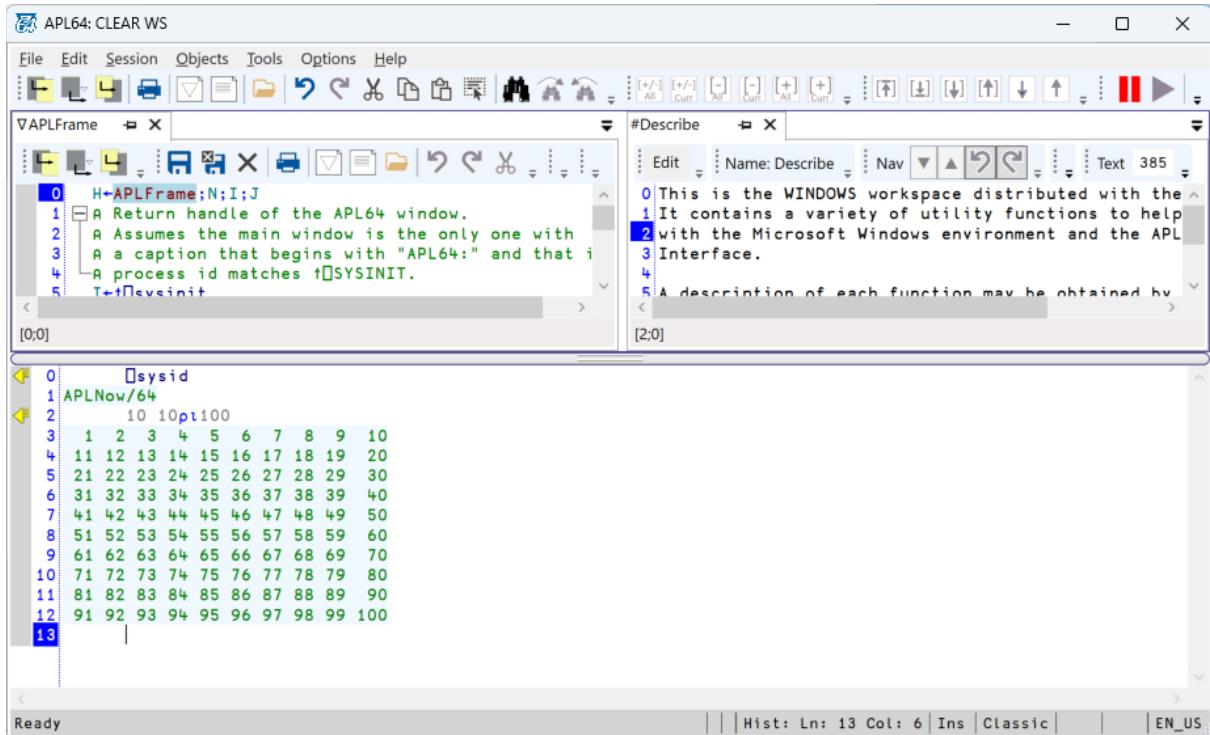
'Find Next Bookmark' keyboard shortcut modified to Alt+F2 from F2

The purpose of this change was to support the keyboard shortcut in the Sequential history format.



Line number display in the history pane and object editors have been enhanced

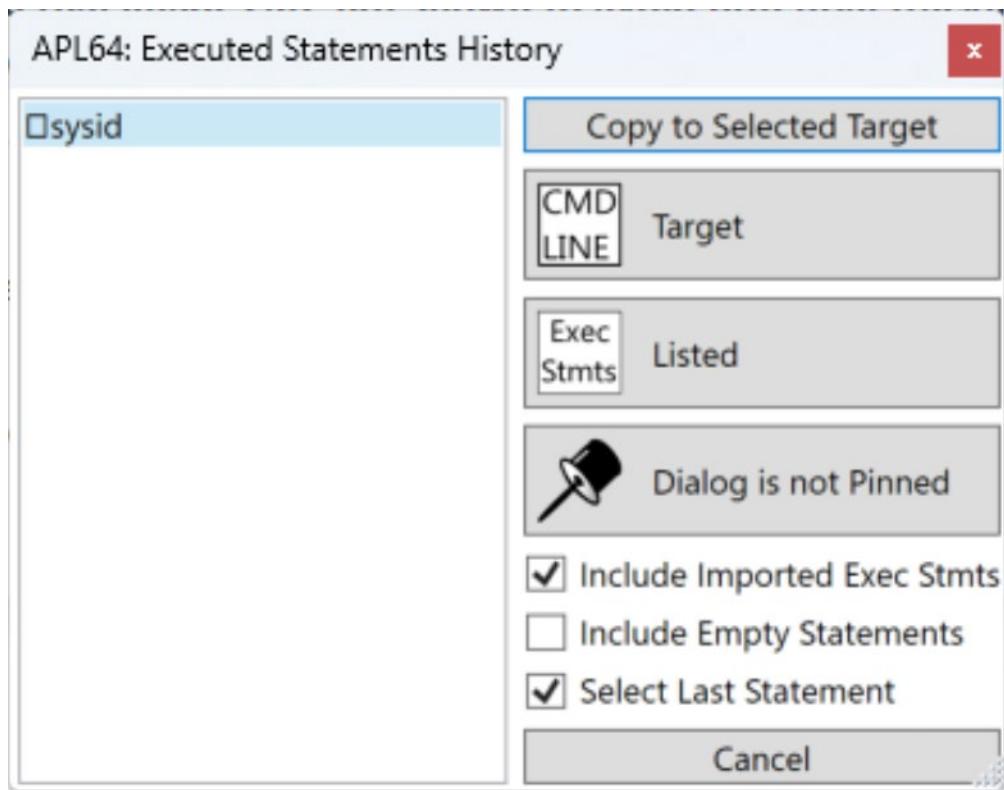
The user-selected foreground and background colors for line numbers are reversed for the line containing the cursor to provide a visual cue indicating the position of the cursor.



Find next/previous executed line options now supported in editable Classic History format

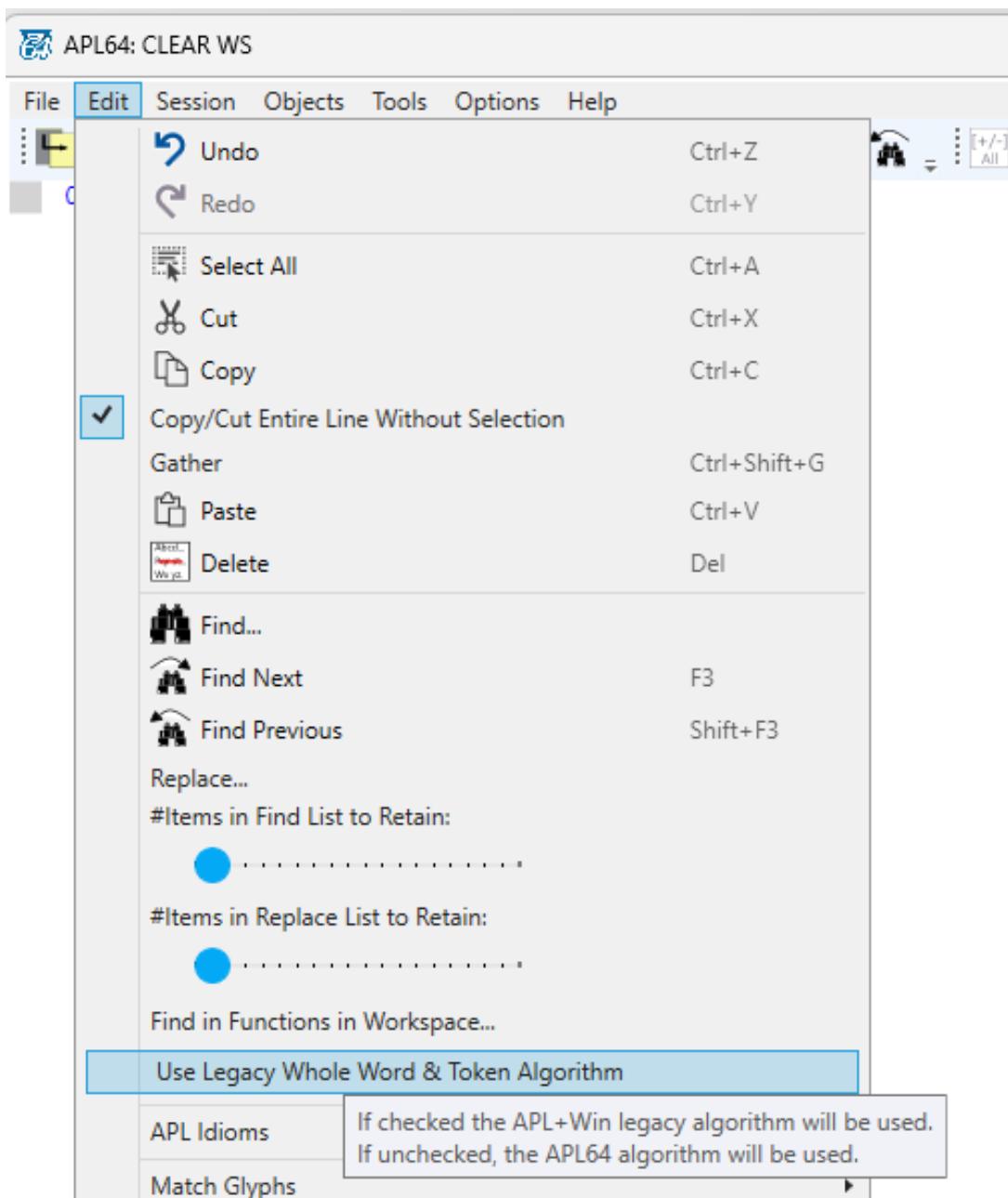
The Find next and previous executed line menu options and their corresponding keyboard shortcuts, Alt+F9 and Shift+Alt+F9, respectively, are now supported when the Classic History format is editable.

Executed Statements History (Ctrl+F9) window updated



Find tool and search keyboard shortcuts for Token and Word searches improved

For better APL+Win compatibility the menu item **Edit | Use Legacy Word & Token Algorithm** is enabled by default in APL64. This option applies to the Find tool, Ctrl+F3 and Ctrl+Shift+F3 keyboard shortcuts with and without a selection. Uncheck this menu item to use the APL64 algorithm.



Objects | Row Numbers menus improved

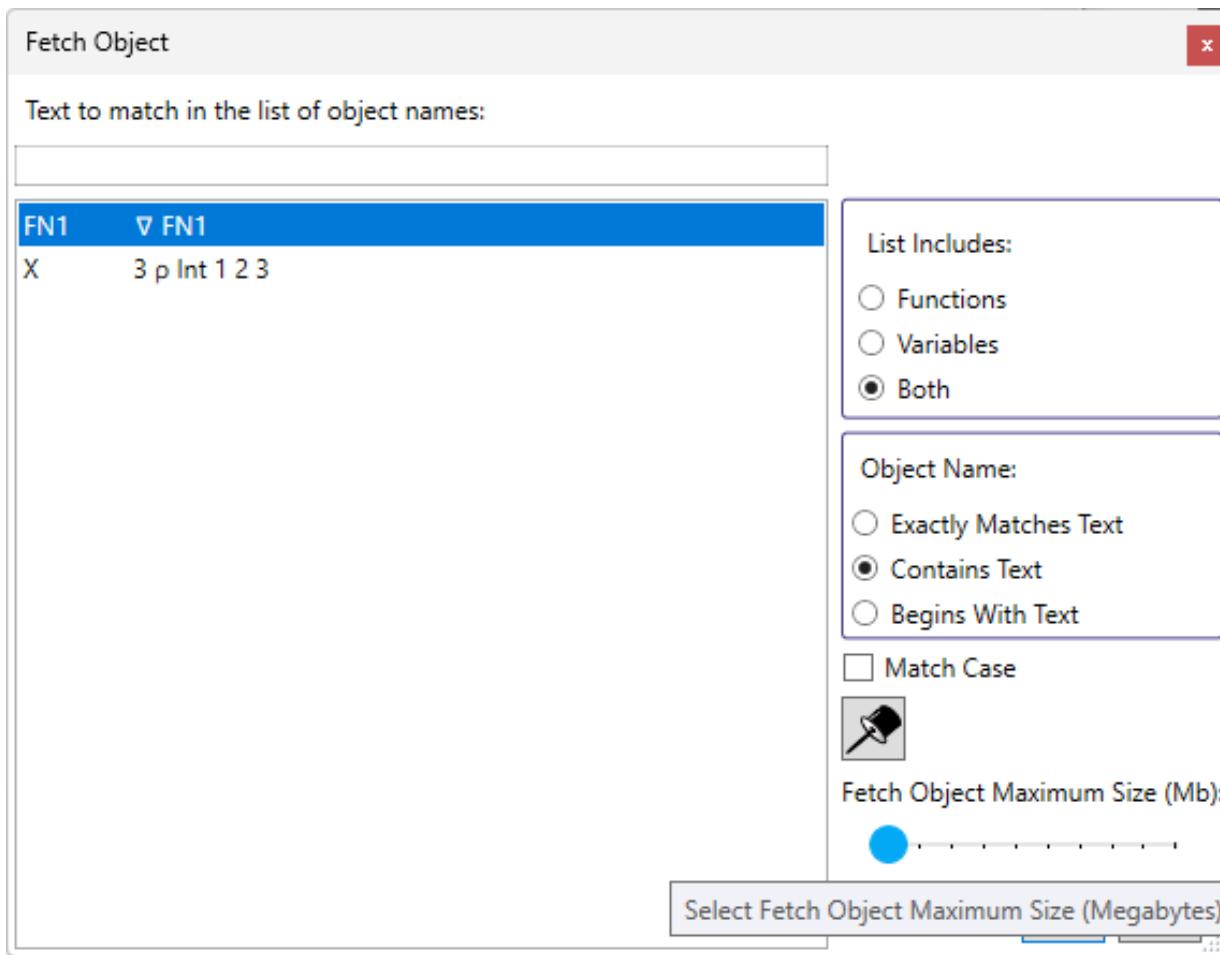
The **Objects | Row Numbers | Display Row Numbers in Function Editor** and **Objects | Row Numbers | Display Row Numbers in Variable Editor** submenus were moved into their respective **Objects | Variable Editor** and **Objects | Function Editor** menus since the row numbers are optional features in the variable and function editors.

Objects | Tool Bars menus improved

The **Objects | Tool Bars** submenus were moved to the **Objects | Function Editor** and **Objects | Variable Editor** menus.

Update to Fetch Object tool to support large matrices

The new **Fetch Object Maximum Size** option may be used to specify the maximum size of an object that can be fetched. The default value is 100Mb.



New Session | Suppress Unassigned History Output From Quote-Quad Output menu

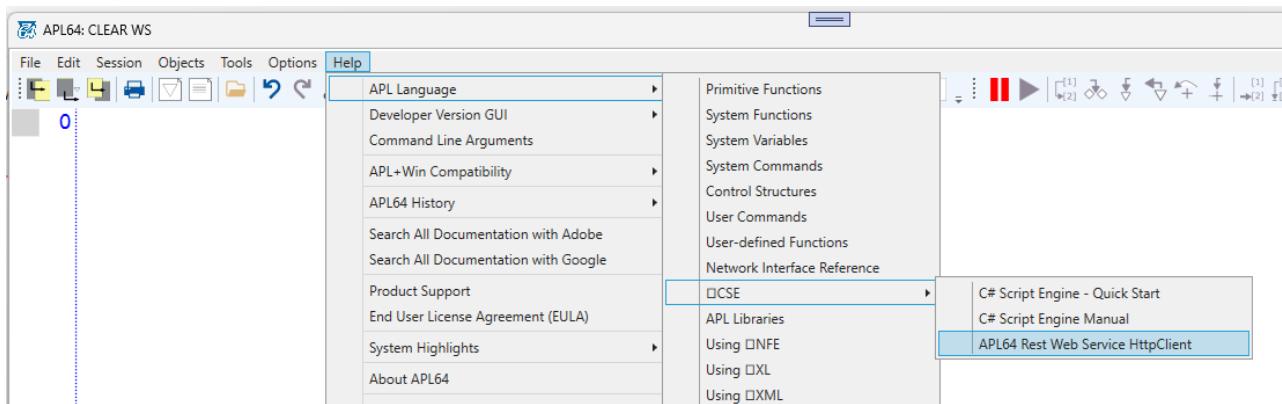
This option specifies when to display the quote-quad output value in the history pane when assignment is used. Quote-quad output can always be assigned. The APL+Win behavior (unchecked) is that even if quote-quad output is assigned, the quote-quad output value is rendered to the history. For some uses of quad input, the suppression (checked) behavior is appropriate.

New Session | Classic History Format | Refresh For Accurate History menu

This menu option will refresh the editable classic history pane by removing any programmer scratch-pad text that has not been executed by the interpreter.

CSE documentation consolidated

The menu item **Help | APL Language | CSE** now contains all CSE programmer documentation. The new **APL64 Rest Web Service HttpClient** topic illustrates how to use the CSE system function to access a REST web service from APL64 CSE and use .Net assemblies which are published in a Nuget package.



Digital code signing available for an APL64 Windows Runtime Exe

An option is provided to digitally sign an APL64 WRE (Windows runtime executable) using an APL64 programmer-provided digital signature certificate. An application should be digitally signed to indicate that the application as delivered to the end user came from the application publisher (e.g. APL programmer). Use the optional digital signature section of the Create Windows Runtime Executable dialog for this purpose. Refer to the document in **Help | Developer Version Gui | Create Windows Runtime Executable** for the details.

APL64: Create Windows Runtime Executable

Runtime Workspace Filename: *

EXECUTABLE CONTENT

Runtime Windows Executable Target Folder: *

Runtime Executable Name: * Same as Runtime Workspace Name

Digitally sign output
 Digital signature command

APL64 xml-format Configuration File: Include APL64 xml-format Configuration File

APLNow32 adf-format Configuration File: Include APLNow32 adf-format Configuration File

APLNow32 ini-format Configuration File: Include APLNow32 ini-format Configuration File

Additional Files Required for the Application		
Source Path	Base Target Path	Target Path Suffix
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Add One Required File"/> <input type="button" value="Add Multiple Required Files"/> <input type="button" value="Add Folder of Required Files"/> <input type="button" value="Remove All Required Files"/>		
<input type="checkbox"/> Overwrite		

ASSEMBLY META-DATA

Properties.Details: File Description:

Properties.Details: Product Name:

Properties.Details: Copyright: *

Properties.Details: File Version:

Properties.Details: Version: Same as File Version

Assembly.Info: Company Name:

Assembly.Info: Application Description: Same as File Description

Assembly.Info: Version: Same as File Version

Assembly.Info: Neutral Language:

Assembly.Info: Description: Same as File Description

Application Icon File:

* Required entries

Materialize additional files, if any, to the path where the WRE exe or CPC Nuget package is installed, rather than the path where the WRE or CPC dll is running

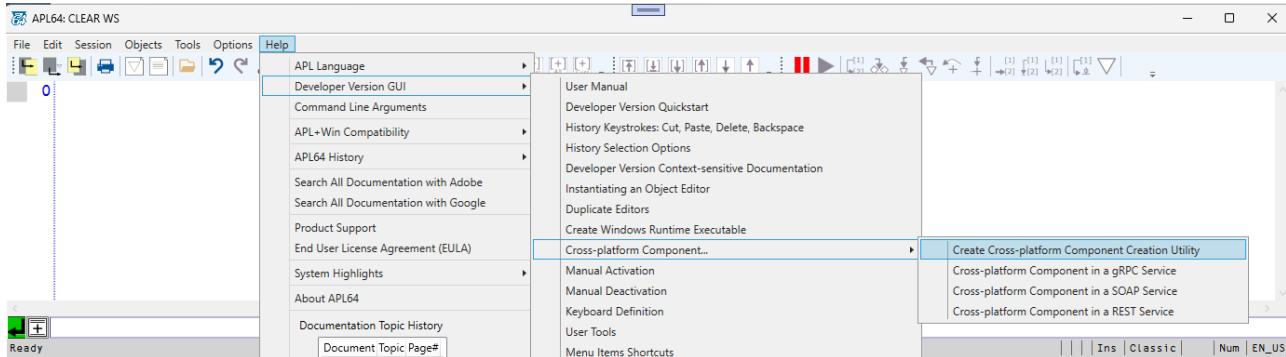
Materialize the APLNow32 ini and adf files, if included in the WRE, to the path where the WRE dll is running

The ‘Add Folder of Required Files’ button to add files in a folder and all sub-folders in the Additional Files Required for the Application area

The ‘Remove All Required Files’ button to remove all files listed in the Additional Files Required for the Application area

CPC documentation consolidated

The menu item **Help | Developer Version GUI | Cross-Platform Component** contains all CPC programmer documentation. Examples are provided for using an APL64 cross-platform component in a gRPC Service, SOAP Service and REST Service.



SCAN function updated

If your workspace contains the SCAN function from past versions of APL64 or APL+Win, you will need to update it with the updated version in the TOOLS\UTILITY.ws64 workspace prior to using it in this version of APL64.

Arithmetic Functions Performance Improved for Small Arrays

The arithmetic functions (+ - × ÷) are improved for small arrays so that the performance of APL64 equals or may exceed that of APL+Win.

The APL64 interpreter identifies small arrays using the IT64 'TuneDS' system function with a default threshold value of 100 elements. An APL64 programmer may query or modify this system function value to optimize system performance in an application system.

Since 2023 the analogous performance of APL64 for large arrays exceeds that of APL+Win. Since results may vary, application-specific testing is important.

Syntax:

```
existingSetting ←  IT64 'TuneDS'    Ⓜ query dyadic scalar primitive tuning parameter  
 IT64 'TuneDS' newSetting      Ⓜ set dyadic scalar primitive tuning parameter
```

Refer to **Help | APL Language | Using WSE** for additional information.

New AES system function: Encryption and Decryption

The AES system function supports the encryption and decryption of text using the AES cryptographic algorithm, implemented in the Microsoft System.Security.Cryptography toolkit.

AES supports the encryption of text which includes Unicode code points not in AV.

Refer to [AES](#) in Help | APL Language | System Function for additional information.

New [BROTLI](#) system function: A better alternative to gzip inflate/deflate

Brotli is a lossless data compression algorithm developed by Google. It uses a combination of the general-purpose LZ77 lossless compression algorithm, Huffman coding and 2nd-order context modelling. Brotli is primarily used by web servers and content delivery networks to compress HTTP content, making internet websites load faster. A successor to gzip, it is supported by all major web browsers and has become increasingly popular, as it provides better compression than gzip.

[BROTLI](#) supports the compression of text which includes Unicode code points not in [AV](#).

Refer to [BROTLI](#) in Help | APL Language | System Function for additional information.

New [XML](#) methods: JsonToXml and XmlToJson methods

These methods are helpful when accessing web services which use json format in APL64.

Refer to [XML](#) in Help | APL Language | Using [XML](#) for additional information.

Performance Improved for Colossal Component Files

By using explicit layout structures, rather than BitConverter logic, the performance of colossal component files is improved. Previously created colossal component files remain compatible.

APL64 session responded to the Insert/Overwrite keyboard status changed in other applications

The CapsLock and NumLock keyboard status in the APL64 session could be out of sync with Windows

Bookmarks (ctrl+F2) did not work in a new editable classic history session

Name validations for 'Runtime Executable Name', 'Company Name' and 'Copyright' fields

The Digital signature command and digital signature checkbox are on the same line in the utility

APL64: Create Windows Runtime Executable

Runtime Workspace Filename: *

EXECUTABLE CONTENT

Runtime Windows Executable Target Folder: *

Runtime Executable Name: * Same as Runtime Workspace Name

Digital signature command: Digitally sign output

APL64 xml-format Configuration File: Include APL64 xml-format Configuration File

APLNow32 adf-format Configuration File: C:\Program Files\APLNowLLC\APL64\APLNow32.adff Include APLNow32 adf-format Configuration File

APLNow32 ini-format Configuration File: C:\Program Files\APLNowLLC\APL64\APLNow32.ini Include APLNow32 ini-format Configuration File

APLNow32 Manifest File: C:\Program Files\APLNowLLC\APL64\APLNow32.exe.Manifest Include APLNow32 manifest File

Additional Files Required for the Application

Source Path	Base Target Path	Target Path Suffix	Overwrite
<input type="button" value="Add One Required File"/>	<input type="button" value="Add Multiple Required Files"/>	<input type="button" value="Add Folder of Required Files"/>	<input type="button" value="Remove All Required Files"/>

ASSEMBLY META-DATA

Properties.Details: File Description:

Properties.Details: Product Name:

Properties.Details: Copyright: *

Properties.Details: File Version: Same as File Version

Properties.Details: Version: Same as File Version

Assembly.Info: Company Name:

Assembly.Info: Application Description: Same as File Description

Assembly.Info: Version: Same as File Version

Assembly.Info: Neutral Language:

Assembly.Info: Description: Same as File Description

Application Icon File:

* Required entries

Entry added for the APLNow32.exe.manifest file*

The screenshot shows the 'APL64: Create Windows Runtime Executable' dialog box. In the 'EXECUTABLE CONTENT' section, the 'APLNow32 Manifest File' field is highlighted with a red border. It contains the path 'C:\Program Files\APLNowLLC\APL64\APLNow32.exe.Manifest'. To the right of this field are two checkboxes: 'Include APLNow32 manifest File' (checked) and 'Include APLNow32 executable manifest File' (unchecked). Below this section is a table for 'Additional Files Required for the Application' with columns for 'Source Path', 'Base Target Path', 'Target Path Suffix', and 'Overwrite'. Buttons at the bottom include 'Create', 'Exit', 'New WRE Info', 'Load WRE Info', and 'Save WRE Info'. A note at the bottom right says '* Required entries'.

*The APLNow32.exe.manifest file should be included in an APL64 WRE when some wi controls like CommandBars and CommandButtons are used.

Saving and Retrieving Unicode Data in APL64

Refer to new Help | APL Language | NFE | Save Unicode Data menu for information on handling application-specific text data that may contain Unicode code points which are not in AV.

Use an APL64 CPC in Excel

Refer to Help | Developer Version GUI | Cross-platform Component | Cross-platform Component in Excel for information on creating an APL64 cross-platform component (CPC) that can support application-specific Excel functions, which are transparently available for use in Excel worksheet cells or macros.

APL64 2025.0.1 Update

Interpreter Modifications

Colossal File System: Significant performance improvements reading components

This is accomplished when a colossal file is tied exclusively (open mode = 0+16 or 2+16) in update 2025.0.1. Refer to the user documentation of the new open modes for `□cftie` for additional information.

Colossal File System: New File Open Modes

`□cfstie` now supports file tying modes like the modes of `□xntie`. This allows files to be explicitly opened in read-only mode, providing shared, read-only access by multiple users or processes. Refer to the user documentation of the new open modes for `□cftie` for additional information.

XML Serialization Exception Message Format

If an XML serialization exception occurs, the formatting of a multi-line exception message is improved.

□XL: New Actions to Read and Write CSV-format Files

The `□XL SaveWorkSheetAsCsv` and `ToAplFromCsv` actions have been implemented to read and write 'comma-separated' format files. An option is provided to specify the field separator.

□FnEdHist System Function

Function editing history can be captured as an APL nested array or an XML document using the new `□FnEdHist` system function. Refer to the function editing change records features described below for additional information.

□NewLine and □SetNewLines System Functions

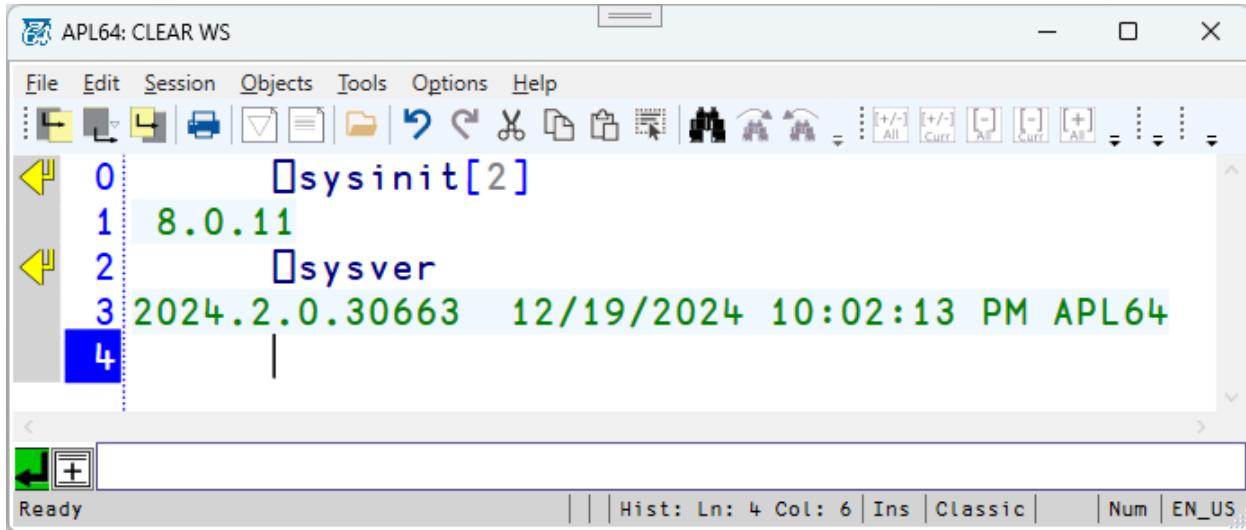
These new system functions are useful when an APL64 cross-platform component generates multi-line text, so that line termination character(s) which are appropriate for the operating system are used.

□LCASE and □UCASE system functions updated to support Unicode characters

These system functions now support Unicode characters which are not included in `□av`.

SYSINIT[2] (2nd element) returns the .Net Version targeted by APL64 instance

This information is useful when an APL64 cross-platform component is incorporated into a larger .Net application, or when using the `cse` system function. Refer to [Environment.Version Property \(System\)](#) | Microsoft Learn for additional information.



The screenshot shows the APL64: CLEAR WS window with the following output:

```
0  ⌊sysinit[2]
1  8.0.11
2  ⌊sysver
3  2024.2.0.30663  12/19/2024 10:02:13 PM APL64
4
```

STRING 'ToLower' and 'ToUpper' actions enhanced to support 'culture'

string 'ToLower' and 'ToUpper' actions have been enhanced to provide an optional argument specifying the 'culture' to use for these actions. Use the new `string 'GetCultureInfo'` action to obtain the appropriate .Net culture specification, e.g. 'en-US', 'de-DE'. E.g.,

The screenshot shows the APL64: CLEAR WS application window. The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Undo/Redo. Below the toolbar is a code history pane with numbered steps:

- 0 ⌂ucs X←'i'
- 1 105
- 2 ⌂ucs⌊←X⌊String 'ToUpper' 'en-US'
- 3 I
- 4 73
- 5 ⌂ucs⌊←X⌊String 'ToUpper' 'tr-TR'
- 6 İ
- 7 304
- 8

The status bar at the bottom shows "Ready" and "Hist: Ln: 8 Col: 6 Ins | Classic | Num EN_US".

☐ VR system function enhanced to specify the format of the result

☐ vr system function has been enhanced to provide an optional right argument to specify the format of the result. Without the optional right argument (`☐vr 'myFn'`) the traditional format is the result. The 'InclLFD' optional right argument (e.g., `☐ vr 'myFn' 'InclLFD'`) includes the 'last fix date' timestamp of the function in the result. The 'xml' and 'json' optional right arguments (e.g., `☐ vr 'myFn' 'xml'` or e.g., `☐ vr 'myFn' 'json'`) include the function editing history in the result of `☐ vr`. The `☐def` system function accepts the three new formats of `☐ vr 'myFn'`.

The screenshot shows the APL64: CLEAR WS application window. The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Undo/Redo. The main code editor area displays the following APL code:

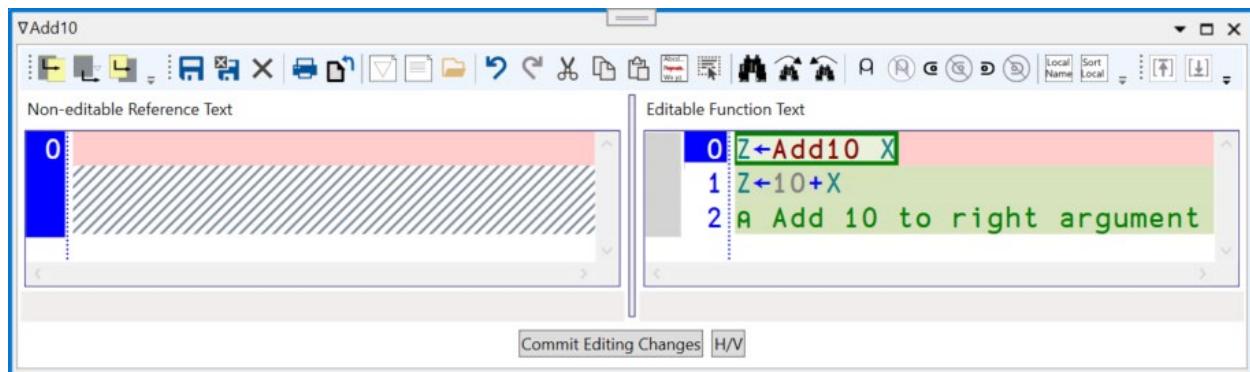
```
0 def 'FN1' '123' '456'
1 FN1
2 ⌂vr 'FN1' 'InclLFd'
3   FN1
4 [1] 123
5 [2] 456
6 
7 Last Fix Date: 2024 12 9 6 46 7 620
8
9
```

The status bar at the bottom shows "Ready", "Hist: Ln: 9 Col: 6", and "EN_US".

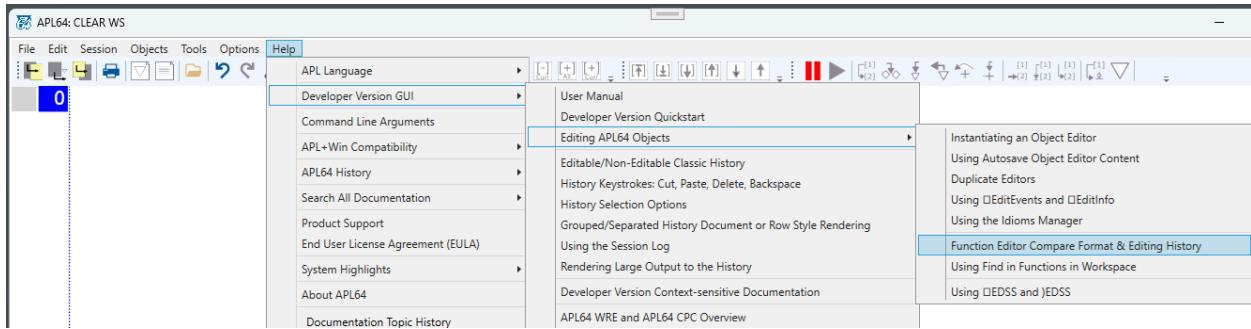
APL64 Developer Version GUI Modifications

New Function Editor Compare Format

The function editor compare format illustrates a reference text (left) and the current edited text (right). Editing changes are identified graphically using background colors, and rectangles. The traditional format of the function editor has not been modified.



To learn more about the new function editor compare format access the user documentation here:



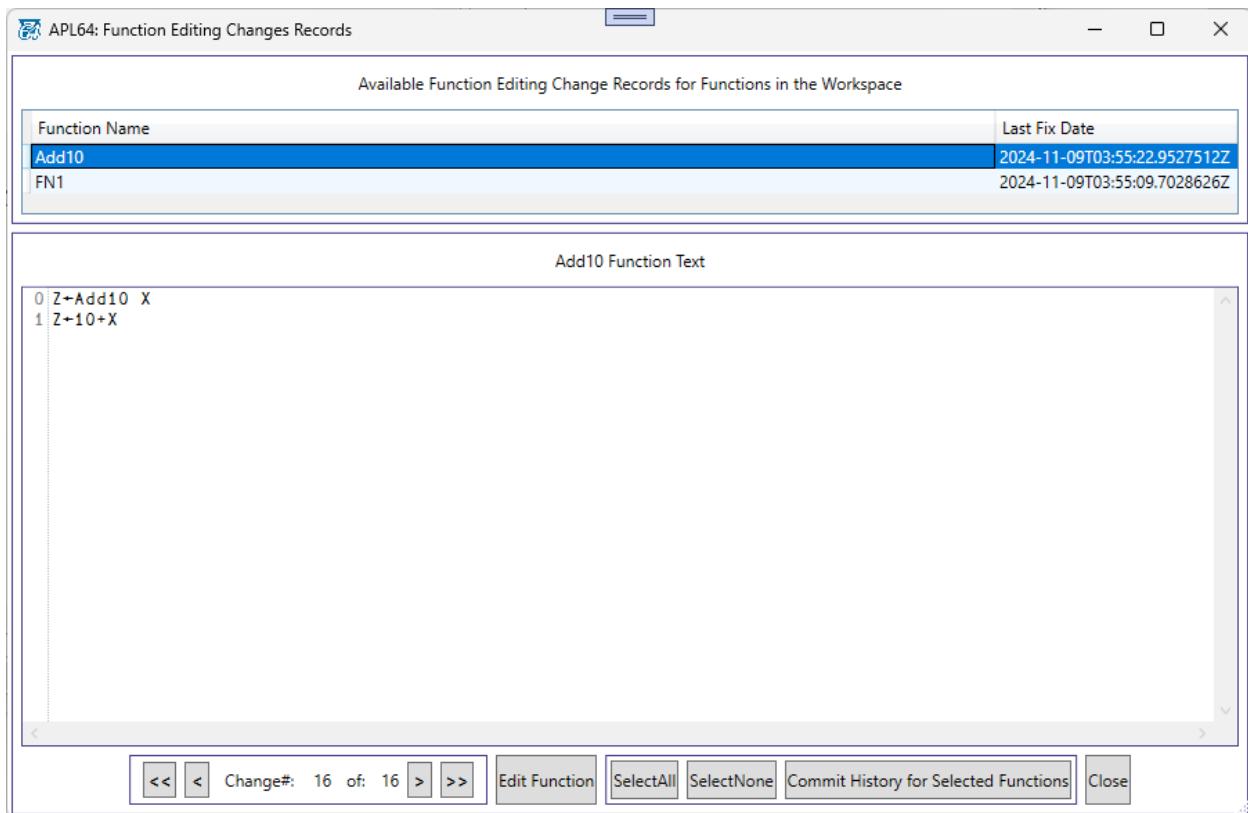
New Function Editing Changes Records

An option has been implemented to record function editing changes within the function definition so that they may be undone/redone in subsequent function editing instances, even in subsequent APL64 developer version instances. Select this option using the **Objects | Function Editor | Function Editing Change Records | Consider Function Editing Change Records** check box.

Function editing change records are graphically shown when the new function editor compare format is selected.

Function editing change records for user-defined functions in the workspace may be reviewed using the **Objects | Function Editor | Function Editing Change Records | Review Function Editing Change Record** dialog.

Refer to the **Help | Editing APL64 Objects | Function Editor Compare Format & Editing History** document for details.



Documentation of the Options | Keyboard Definition Dialog Updated

Refer to the **Help | Developer Version GUI | Keyboard Definition** document for details.

Find/Replace Tool Updates

1. The Find/Replace tool now reopens in the same location it was closed. This behavior applies between text panes in the same APL64 developer version instance, and between APL64 developer version instances
2. The Find action will now select the next found instance, if any, and put the keyboard focus on the searched text so that the user may conveniently enter a replacement text.

Set Initial Caret Marker Position when Editing a New Function

This is configurable in the **Objects | Function Editor | Caret At End of Header For New Function** menu item.

WRE and CPC Utility

APL64 provides two runtime options. The APL64 Windows Runtime Executable (WRE) option is analogous to the traditional APL+Win Windows desktop runtime. The APL64 Cross-platform Component (CPC)

option creates a .Net assembly which can be incorporated into a larger .Net application that can be deployed to the Windows, Android, Linux and ios environments.

New APL64 WRE and Overview documentation

Read this user documentation to find out about the APL64 runtime options. Refer to **Help | Developer Version GUI | Cross-platform Component | APL64 WRE and APL64 CPC Overview** for more information.

Update to APL64 CPC in a REST Service Example (APL64CpcRestSvc.zip)

This example shows how to use an APL64 cross-platform component (CPC) to support the server side of a [REST web service](#). See **Help | Developer Version GUI | Cross-platform Component | Cross-platform Component in a REST Service** for more information.

Update to APL64 CPC in WPF example targets .Net 8 Windows Presentation Foundation (WPF)

This example shows how to use an APL64 cross-platform component (CPC) to support calculations and algorithms exposed by an open-source, [Windows Presentation Foundation](#) (WPF) desktop GUI. Refer to **Help | Developer Version GUI | Cross-platform Component | Cross-platform Component in a WPF Application** for more information.

Update to APL64 CPC in Excel example targets .Net 8 Windows Presentation Foundation (WPF)

This example shows how expose an APL64 cross-platform component (CPC) as a custom Excel function. Refer to **Help | Developer Version GUI | Cross-platform Component | Cross-platform Component in Excel** for more information.

New APL64 CPC using .NET MAUI example

The APL64 CPC using [.NET MAUI](#) example, illustrates how APL64 programmer-defined functions can be included in a cross-platform GUI. Only one source code base is required to create a GUI for the Windows, Android and Apple operating system environments. Refer to **Help | Developer Version GUI | Cross-platform Component | Cross-platform and GUI using MAUI** for more information.

New APL64 CPC in an Azure Function example

The APL64 CPC in an [Azure Function](#) application example illustrates how to use APL64 to create a 'server-less' application on the Microsoft Azure cloud platform which does not require a '24-7' web server and accessible on-demand from a client browser. Refer to **Help | Developer Version GUI | Cross-platform Component | Cross-platform in Azure On-Demand Function** for more information.

Line Termination for Exception Messages

The line termination character(s) in a multi-line APL64 interpreter exception message will be determined by the host operating system environment.

Miscellaneous

APL64 system and installer updated to support .Net 8

APL64 has been updated to Microsoft .Net 8. The APL64 installer for the developer version requires that the target workstation have the appropriate Microsoft .Net 8 SDK and Windows desktop runtime installed.

:IFTEST control structure can be closed with either :ENDIF or :ENDIFTEST or :END:

The :ENDIFTEST and :END options are now documented in the Control Structure and User Manual documents.

XL,)EDSS and EDSS: SpreadsheetGear Nuget package updated from v9.2.44 to 9.3.23.

SQLite: Microsoft.Data.Sqlite client software Nuget package updated from v8.0.6 to v8.0.8

Pdf document generation component updated: ceTe v12.23.0 => 12.25.0.

This component supports the display of user documentation and printing in the APL64 developer version.

APL64 2025.0.5 Update

Interpreter Modifications

Primitive functions with performance improvements

- Power (*)
- Binomial (!)
- Decode (\perp)
- Catenate (τ) for in place assignment, e.g. $X, \leftarrow Y$
- Epsilon (\in) when arguments are a character datatype

System functions with performance improvements and/or updated programmer documentation

- TEXTREPL: Performance and programmer documentation
- TELPRINT: Performance and programmer documentation
- WORDREPL: Performance
- MATRIFY: Performance

New TEXTREPL argument

Added an optional Boolean scalar argument option to the right argument to indicate when overlapping replacement is permitted for successive replacements. The default behavior is to prevent overlapping replacement.

New XL actions: FromAPLToRange, ToAPLFromRange, Autofit and NumberFormatRanges

- FromAPLToRange action inserts the elements of an APL64 variable into an Excel worksheet range.
- ToAPLFromRange action fills an APL64 array with the values of an Excel worksheet range.
- Autofit action specified ranges.
- NumberFormatRanges sets the number format of selected ranges in an Excel-format worksheet.

Refer to the **Help | APL Language | Using XL** menu for additional details.

SSASSIGN programmer documentation syntax and example updated

SSCAT programmer documentation example updated

SYSINIT Additional Values: EnvVars, MachineName and Processor count

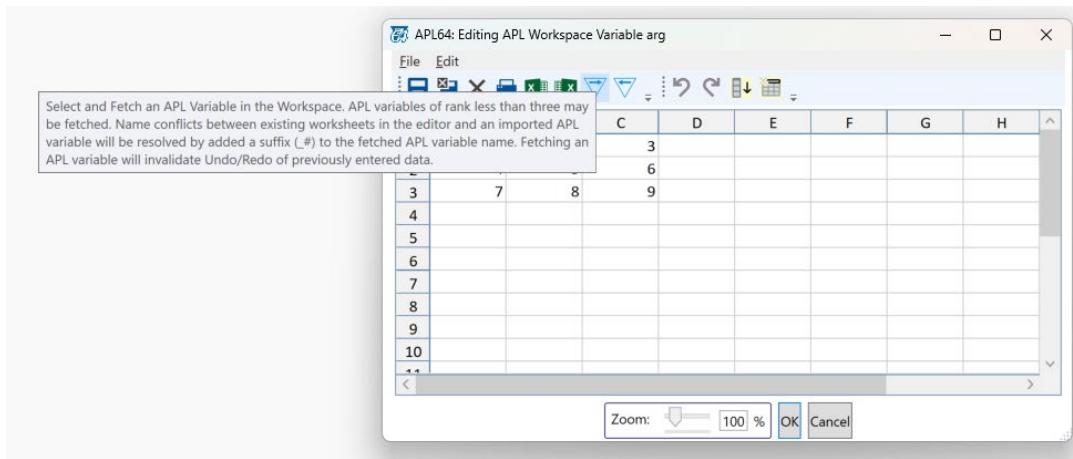
- SYSINIT[4]: Environment variables array
- SYSINIT[6]: Machine name
- SYSINIT[7]: Processor count

APL64 Developer GUI Modifications

EDSS: New Import and Export options from/to an APL variable

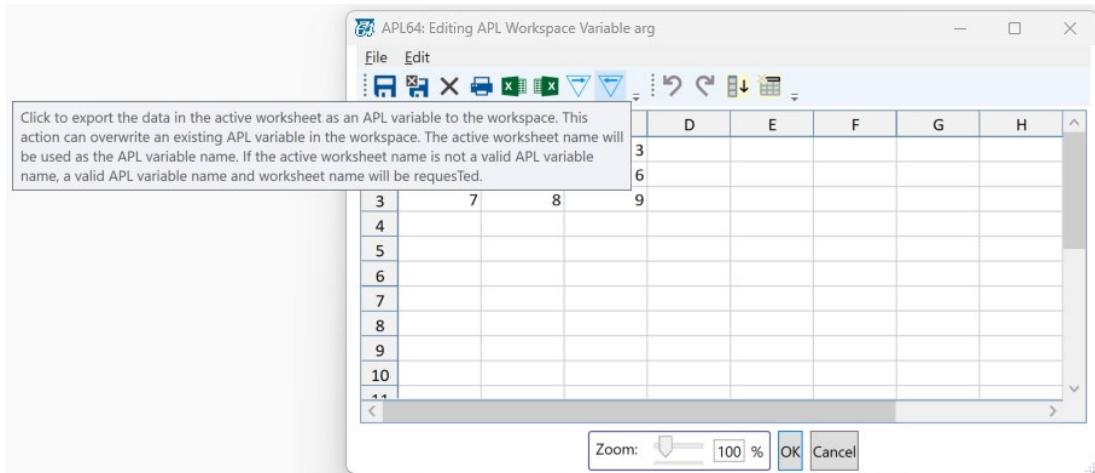
- Import an APL Variable from the Workspace

Use the EDSS editor to edit an existing APL variable, and click the **File | Import APL Variable to Worksheet** menu item or toolbar button  to select the workspace variable to import to the EDSS editor:

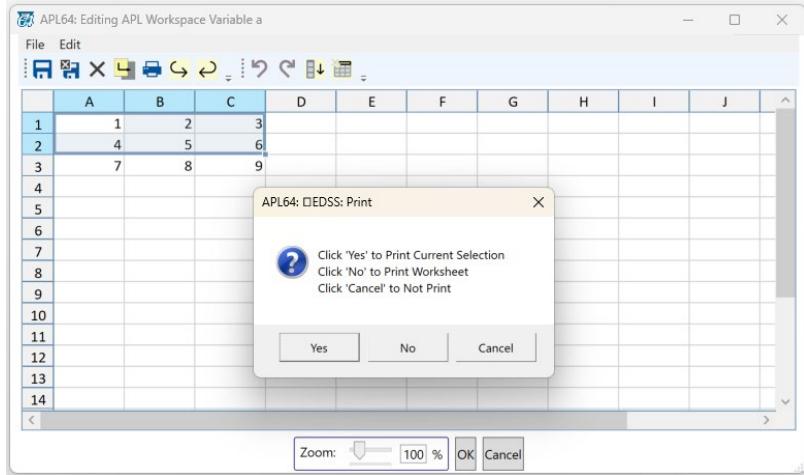


- Export Active Worksheet Data to an APL Variable

The EDSS dialog supports saving the data in the active worksheet to the workspace. Click the **File | Export Active Worksheet Data As APL Variable** menu item or the toolbar button.

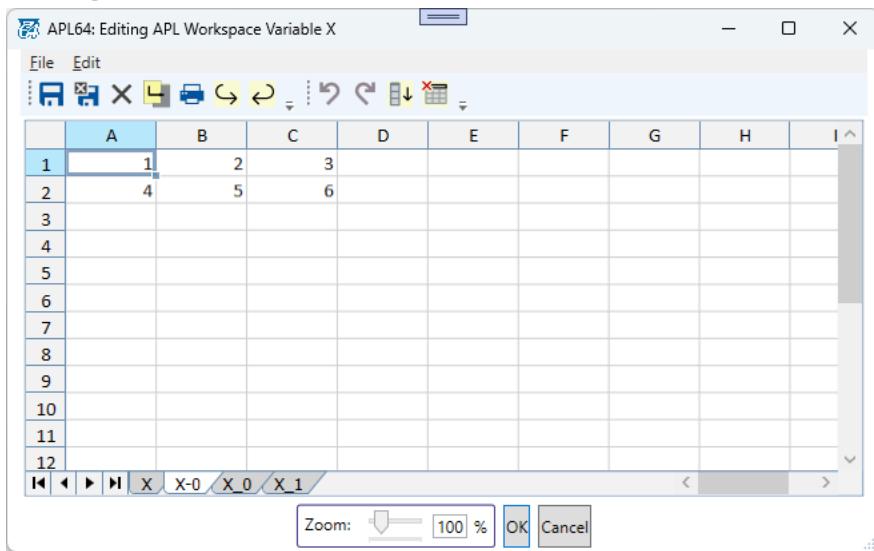


EDSS: New option to print the Worksheet or Current Selection in the Worksheet

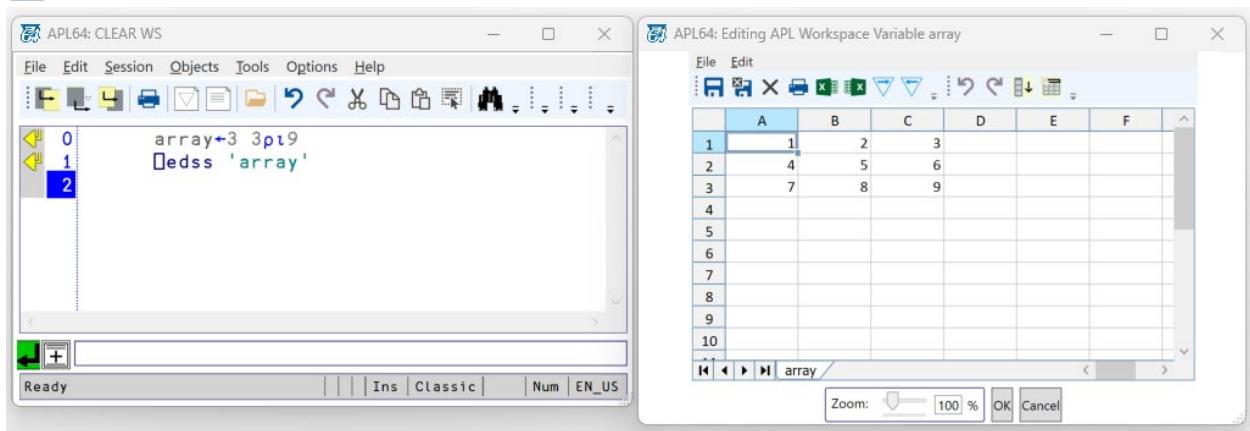


Note: When the *Current Selection* option is selected, the selection will be printed, however the printer dialog option button for Selection is not checked. This behavior has been reported to the third-party vendor.

EDSS: Active worksheet tab is more prominent with a white background



EDSS: When there is only one worksheet, its tab will now be visible



Consistent Borders in the APL64 Developer GUI

The same border thickness was applied to the following:

- Status bars for variable editor, function editor
- Classic, Sequential, Grouped and Separated history panes
- Editors docking area
- Debug docking area
- Gridsplitter

Prior version:

```

APL64: C:\APLWIN19\Tools\WINDOWS.w3
File Edit Session Objects Tools Options Help
CLIPPaste > X
[0:0]
0 R=CLIPPaste:hClipMemory:hSize:lpGlobalMemory:hApI
1 AVR=CLIPPaste -- Get CF_TEXT data from the Windows clipboard
2 R=''
3 A Get the handle for APL
4 hApI ← APFLame
5 A Open the clipboard.
6 →(↓wcall 'OpenClipboard' hApI)↓0
7 A First see if the clipboard contains any data.
8 →(↓wcall 'IsClipboardFormatAvailable' 'CF_TEXT')pi+1↑LC ◊ 0 0 p↓wcall 'CloseClipboard' ◊ →0
9 A Get a handle to the clipboard data.
10 hClipMemory ←↓wcall 'GetClipboardData' 'CF_TEXT'
11 →(hClipMemory)pi+1↑LC ◊ 0 0 p↓wcall 'CloseClipboard' ◊ →0
12 A Get the size of the data.
13 hSize ←↓wcall 'GlobalSize' hClipMemory
14 o Get a far pointer to the memory

)Load C:\APLWIN19\Tools\WINDOWS.w3
1 "C:\APLWIN19\Tools\WINDOWS.w3" LAST SAVED 4/22/2019 6:54:16 PM
2 >[DLX] Describe
3 This is the WINDOWS workspace distributed with the APL+Win.
4 It contains a variety of utility functions to help you interact
5 with the Microsoft Windows environment and the APL+Win Windows
6 Interface.
7
8 A description of each function may be obtained by entering
9 Explain 'fname'. Run "Summary" for a synopsis of all of the
10 functions in the workspace.
11
12 Copyright 2005-2019, APLNow LLC.
13 jed CLIPPaste
14

```

Ready ||| Editor: Function Name: CLIPPaste Ln: 0 Col: 0 Ins | Classic | Num | EN_US

Updated version:

```

APL64: C:\APLWIN19\Tools\WINDOWS.w3
File Edit Session Objects Tools Options Help
CLIPPaste > X
[0:0]
0 R=CLIPPaste:hClipMemory:hSize:lpGlobalMemory:hApI
1 AVR=CLIPPaste -- Get CF_TEXT data from the Windows clipboard
2 R=''
3 A Get the handle for APL
4 hApI ← APFLame
5 A Open the clipboard.
6 →(↓wcall 'OpenClipboard' hApI)↓0
7 A First see if the clipboard contains any data.
8 →(↓wcall 'IsClipboardFormatAvailable' 'CF_TEXT')pi+1↑LC ◊ 0 0 p↓wcall 'CloseClipboard' ◊ →0
9 A Get a handle to the clipboard data.
10 hClipMemory ←↓wcall 'GetClipboardData' 'CF_TEXT'
11 →(hClipMemory)pi+1↑LC ◊ 0 0 p↓wcall 'CloseClipboard' ◊ →0
12 A Get the size of the data.
13 hSize ←↓wcall 'GlobalSize' hClipMemory
14 o Get a far pointer to the memory

)Load C:\APLWIN19\Tools\WINDOWS.w3
1 "C:\APLWIN19\Tools\WINDOWS.w3" LAST SAVED 4/22/2019 6:54:16 PM
2 >[DLX] Describe
3 This is the WINDOWS workspace distributed with the APL+Win.
4 It contains a variety of utility functions to help you interact
5 with the Microsoft Windows environment and the APL+Win Windows
6 Interface.
7
8 A description of each function may be obtained by entering
9 Explain 'fname'. Run "Summary" for a synopsis of all of the
10 functions in the workspace.
11
12 Copyright 2005-2019, APLNow LLC.
13 jed CLIPPaste
14

```

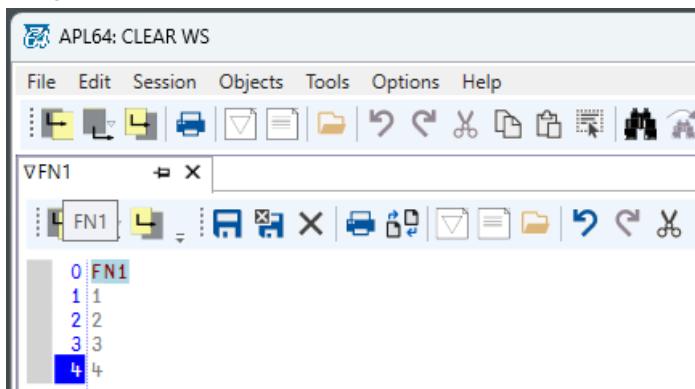
Ready ||| Editor: Function Name: CLIPPaste Ln: 0 Col: 0 Ins | Classic | Num | EN_US

Find/Replace Tool Performance Improvements

The performance of the Find Next/Previous action is significantly improved for large source texts. The option, **Edit | Use Legacy Find Match Highlighting**, is provided to use the legacy behavior so that only the current match is highlighted.

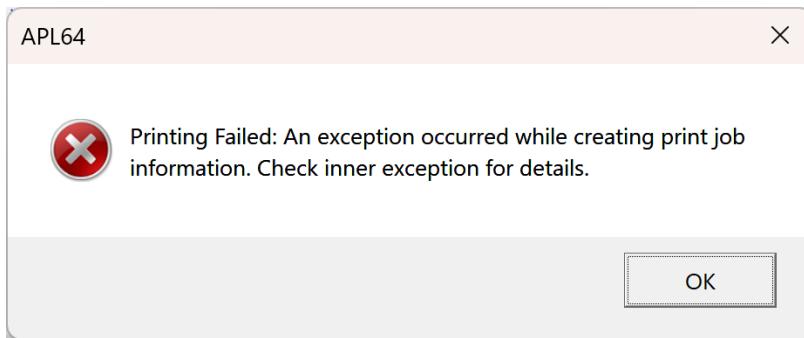
The Command Line, when hidden, is removed as an item in the Active Panes List

Function and Variable Editor Pane Tabs include Tool tips indicating the Object Name

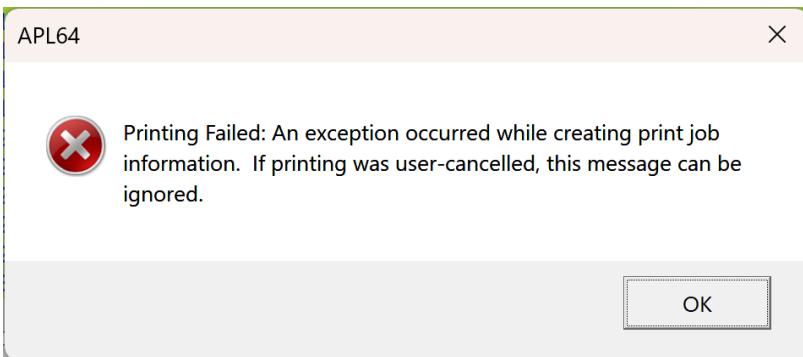


The exception message was improved for a canceled print task

Prior:

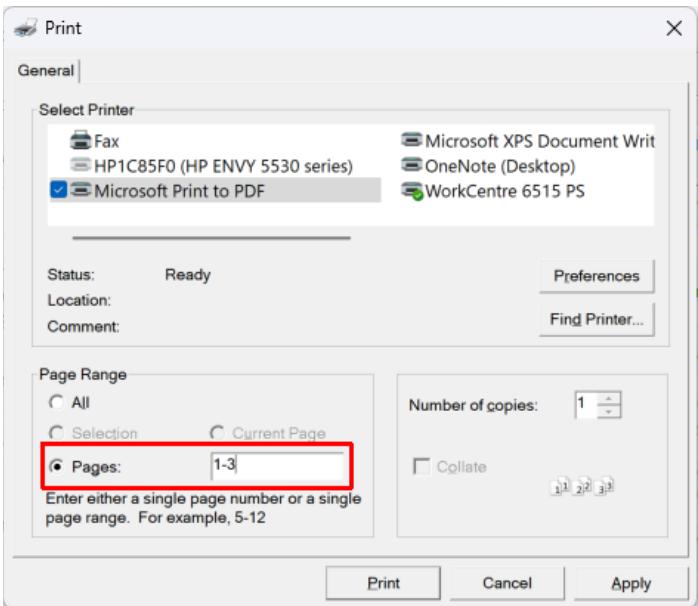


Updated:



Print Updates

Print dialog: The Pages option in the Page Range section is enabled



A heading is printed on each printed page

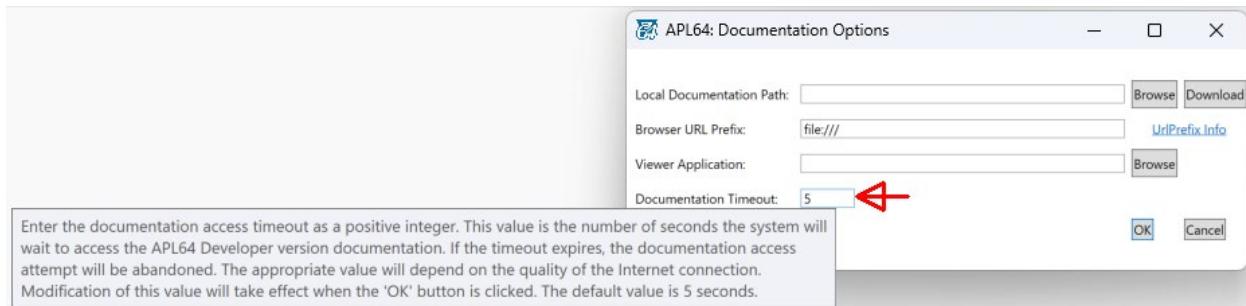
- Object type prefix: #/Variable or !/Function
- Object name
- Page # in the format "Page N of M"

Example:

```
▼APLFrame: Page 1 of 1
[0] H+APLFrame;N;I;J
[1] A Return handle of the APL64 window.
[2] A Assumes the main window is the only one with
[3] A a caption that begins with "APL64:" and that its
[4] A process id matches t$SYSINIT.
[5] I+t$sysinit
[6] H+□wcall 'FindWindow' 0 0
[7] +(xH+□wcall 'GetWindow' H 'GW_HWNDFIRST')+0
[8] :Repeat
[9]     (N J)+□wcall 'GetWindowThreadProcessId' H (,0)
[10]    :If I=J
[11]        N+tt/□wcall 'GetWindowText' H (64p□TCNUL) 64
[12]        +(1tN OSS 'APL64:')/0
[13]    :EndIf
[14] :Until 0=H+□wcall 'GetWindow' H 'GW_HWNDNEXT'
[15]
```

New Documentation Timeout option in the Documentation Options window

This value is the number of seconds the system will wait to access the online APL64 Developer version documentation.



WRE and CPC Utility

Cross-platform Component in Azure On-demand Function: documentation & example improved

The documentation and example for an APL64 CPC in a Serverless Application has been improved. The menu item pointing to this information in APL64 is **Help | Developer Version GUI | Cross-platform Component | Cross-platform Component in Azure On-demand Function**.

New CPC Example: Using CPC in a Webserver with an HTML Browser-based GUI

The documentation and example for an APL64 CPC in a Webserver Application is available from the **Help | Developer Version GUI | Cross-platform Component | Cross-Platform Component in Webserver with Browser GUI** menu in APL64.

Adding an entire folder of Additional Files for the Application is improved

The selected folder is compressed in the WRE or CPC container. The WRE and CPC documentation is updated to illustrate this improvement.

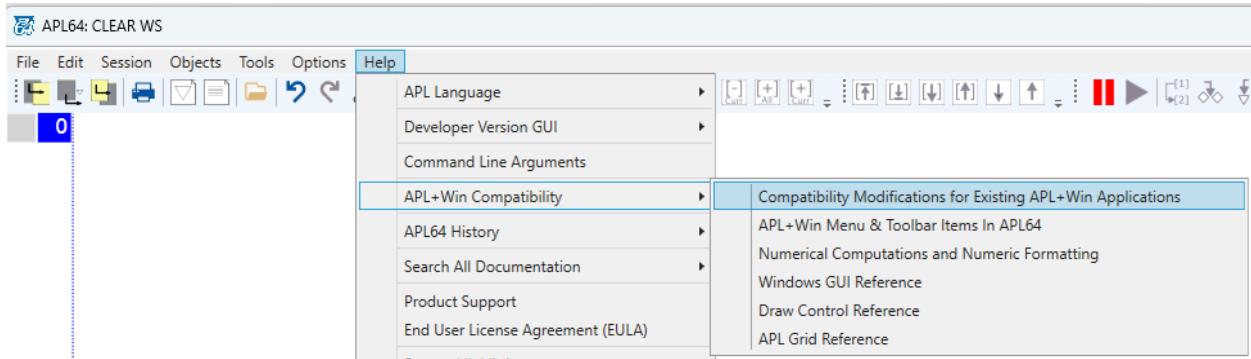
Miscellaneous

Additional fields added to System Information document in **Help | About APL64** menu:

These fields were added:

<input type="checkbox"/> ACBD	<input type="checkbox"/> SYSINIT
<input type="checkbox"/> EXEPATH	<input type="checkbox"/> SYSVERN
<input type="checkbox"/> CT	<input type="checkbox"/> USERID
<input type="checkbox"/> SYSCONFIGFILE	<input type="checkbox"/> WA
<input type="checkbox"/> SYSID	<input type="checkbox"/> WSSIZE

Programmer documentation Updated – APL64 Compatibility with APL+Win



Improved double clicking the mouse selecting a whole word containing delta (Δ) and delta-underscore ($\underline{\Delta}$) character

APL64 2025.0.8 Update

Interpreter Modifications

☐XL: New actions: AddWorksheet , CreateWorkbook, DeleteWorksheet & WsCount

- ☐xl 'AddWorksheet': Adds the specified worksheet to the specified Excel workbook
- ☐xl 'CreateWorkbook': Creates a new Excel workbook
- ☐xl 'DeleteWorksheet': Deletes the specified worksheet from the specified Excel workbook
- ☐xl 'GetWorksheetCount': Returns the number of worksheets in the specified Excel workbook

PROFILE: APL64 system version number and date/time (UTC) info added to the result

APL64 2025.0.5 APL64

2025.0.8 Update

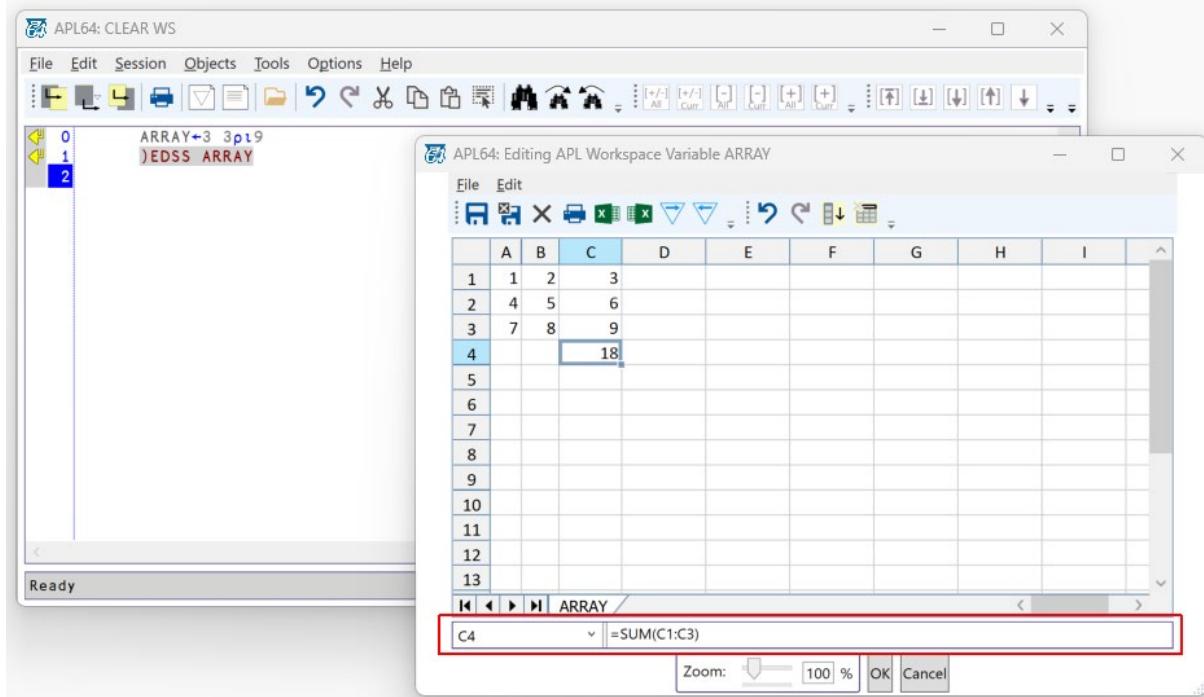
```

APL64 2025.0.5 APL64
File Edit Session Objects Tools Options Help
File Edit Session Objects Tools Options Help
80 )profile get text
81 Total 100.00% 55921.4512ms #852157012 (GC: 6532.959ms)
82 XASPD(Applied dyadic scalar primitive) 23.40% 13087.9467ms #119471339
83 + (Dyadic) 7.63% #268.2715ms #39188107
84 IntScalar + BoolScalar 7.15% 3995.6051ms #36137024
85 DoubleScalar + DoubleScalar 0.47% 261.44ms #3000014
86 IntScalar + IntVector 0.02% 9.897ms #49002
87 BoolScalar + BoolScalar 0.00% 1.0765ms #2062
88 BoolScalar + DoubleScalar 0.00% 0.2489ms #5
89 < (Dyadic) 7.48% #185.0197ms #36001067
90 IntScalar < IntScalar 7.48% 4185.0197ms #36001067
91 + (Dyadic) 6.17% #3452.6188ms #37001055
92 T+Scalar & DoubleScalar 5.70% 3187.2452ms #34000000
Ready
File Edit Session Objects Tools Options Help
File Edit Session Objects Tools Options Help
84 )profile get text
85 Total 100.00% 53384.9987ms #852157012 (GC: 5311.012ms)
86 APL64 Version: 2025.0.8.1887
87 2025-05-30T15:12:53.056Z
88 XASPD(Applied dyadic scalar primitive) 22.17% 11832.8948ms #119471339
89 + (Dyadic) 7.11% 3794.3688ms #39188107
90 IntScalar + BoolScalar 6.61% 3530.673ms #36137024
91 DoubleScalar + DoubleScalar 0.48% 254.9855ms #3000014
92 IntScalar + IntVector 0.01% 7.9783ms #49002
93 BoolScalar + BoolScalar 0.00% 0.4966ms #2062
94 BoolScalar + DoubleScalar 0.00% 0.2354ms #5
95 < (Dyadic) 7.04% 3758.4173ms #36001067
Ready

```

Developer Version GUI Modifications

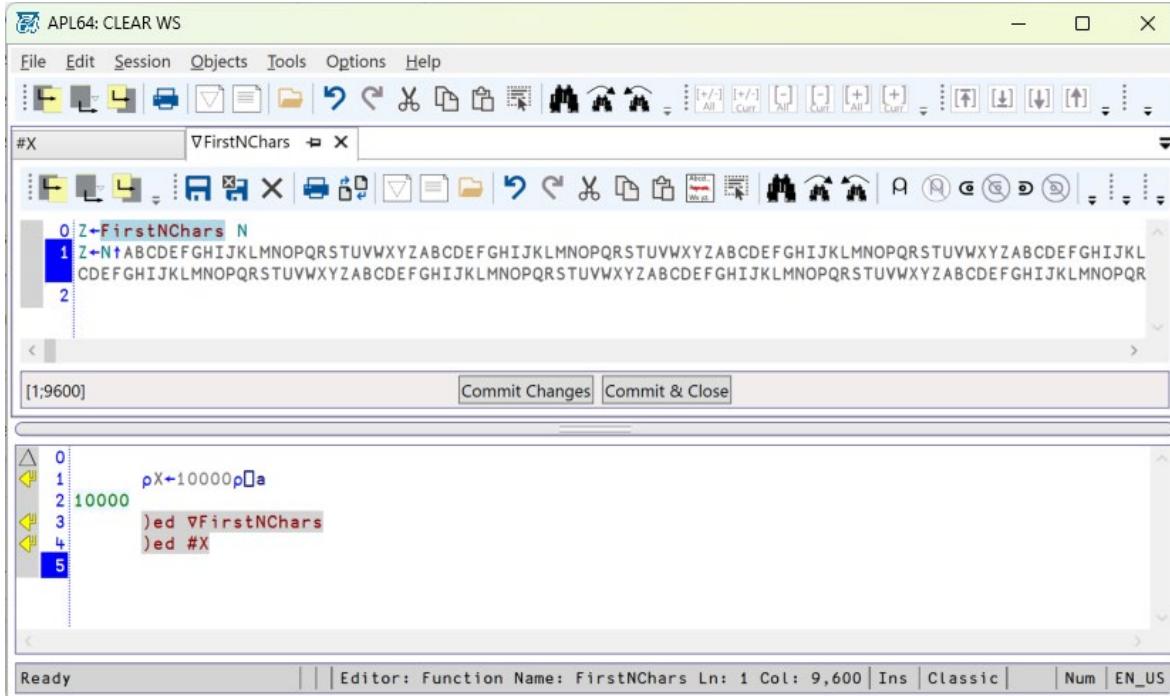
Updates to EDSS and)EDSS: Formula Bar and related user documentation



Documentation was added describing an EDSS worksheet cell can contain a user-entered Excel formula.

The cell formula may be entered directly into the desired cell or entered using the new Formula Bar.

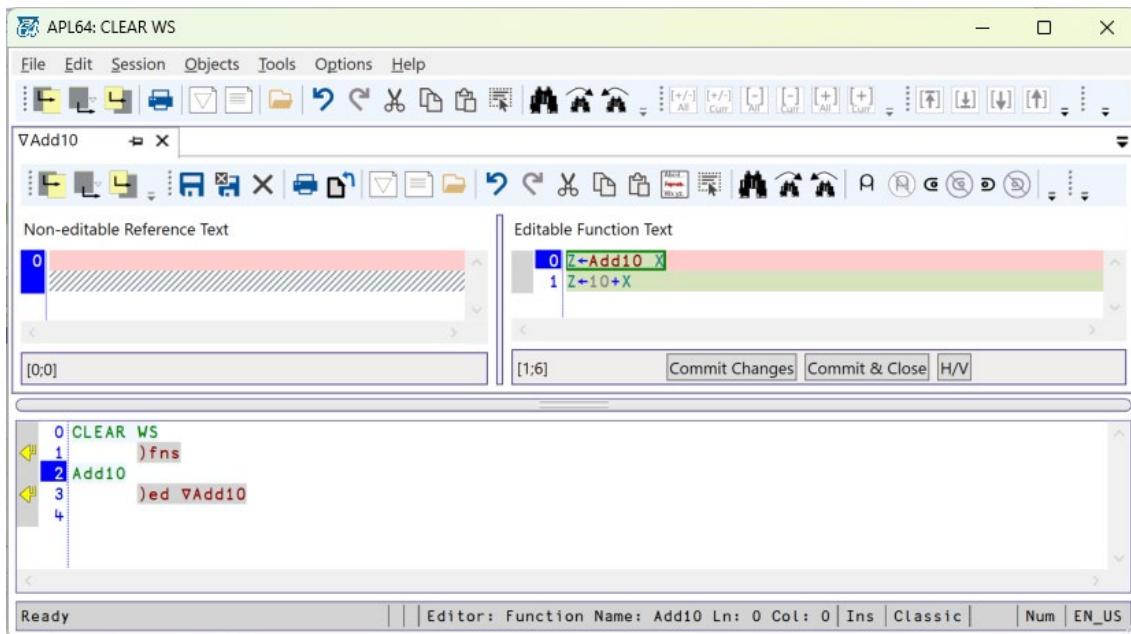
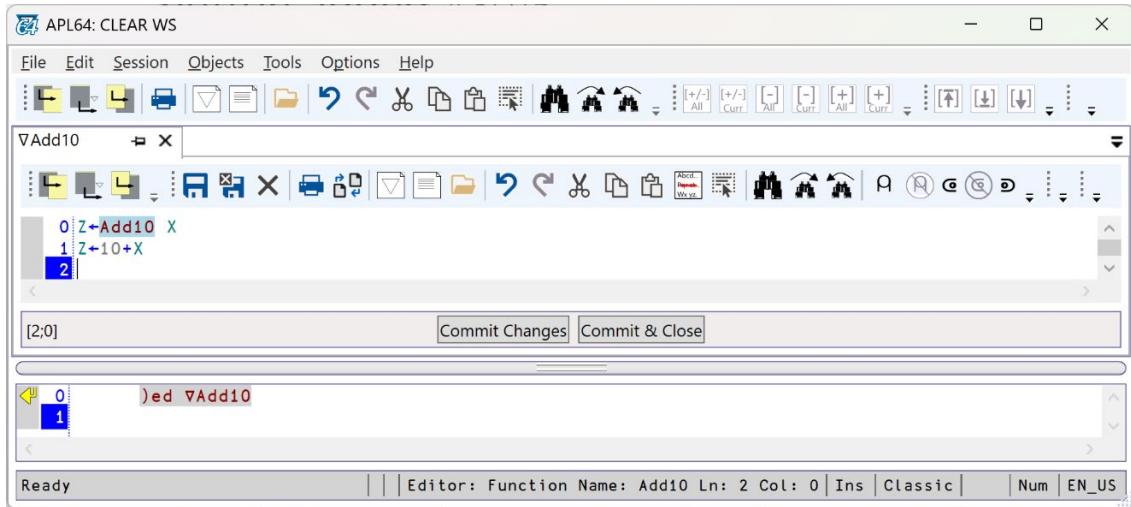
Function Editor: Syntax coloring supported for lines with more than 9600 characters



Function Editor: Commit Changes and Commit & Close buttons added

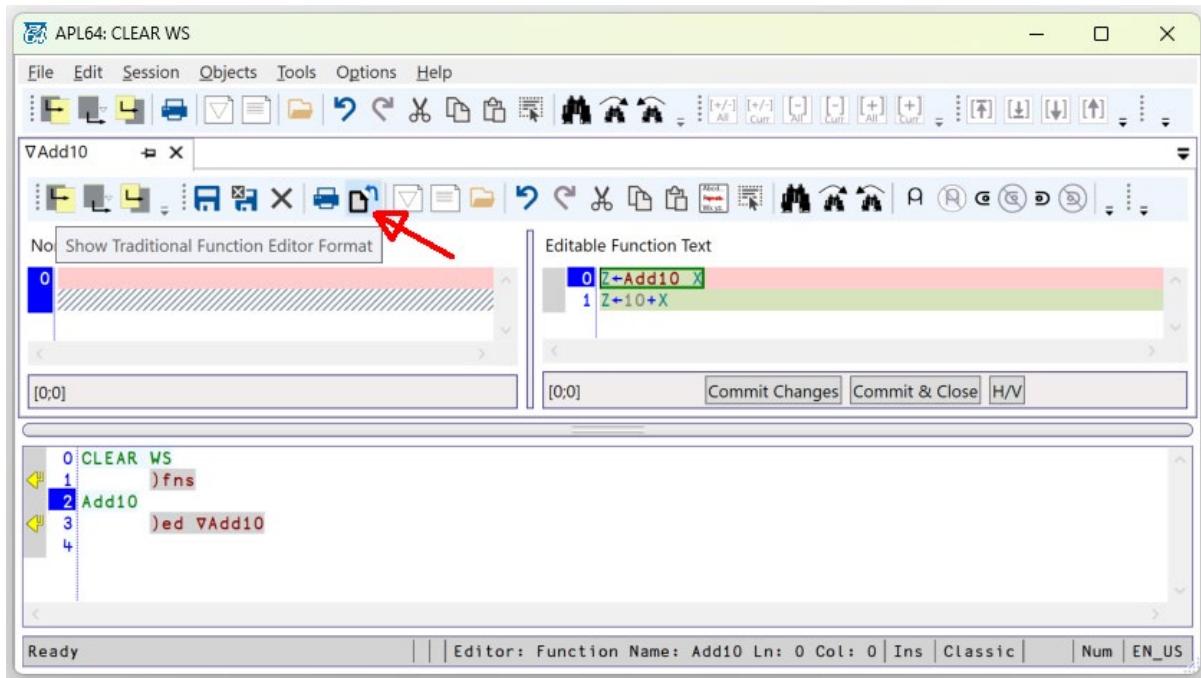
These buttons have been added to the traditional and compare formats of the function editor.

- Commit Changes: When this button is clicked, function edits will be committed.
 - Commit & Close: When this button is clicked, function edits will be committed and the editor will close. The keyboard shortcut Ctrl+Shift+E is available for this action.



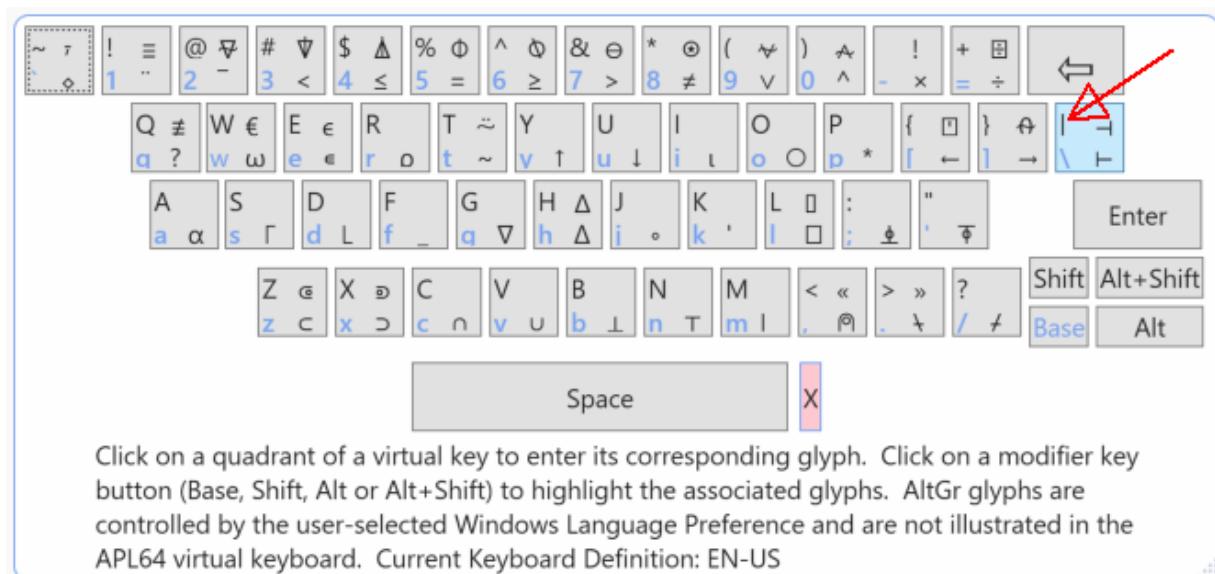
Function Editor: Toolbar icon for the traditional editor format

The toolbar icon to select the function editor traditional format is now distinct from the toolbar icon to select the function editor compare format:



Decimal Unicode 124 and 8739 are now available on EN-US and EN-UK keyboards

For the EN-US and EN-UK keyboard definitions, because APL64 is Unicode aware, APL64 distinguishes between $\Box\text{av}[124+\Box\text{io}]$ (decimal Unicode 124) and $\Box\text{av}[254+\Box\text{io}]$ (decimal Unicode 8739). Decimal Unicode 8739 is the Unicode glyph specified for the APL Magnitude and Residue primitive functions. Therefore, unlike APL+Win, the glyph for Magnitude and Residue is available only by using Alt+M. For the EN-US and EN-UK keyboards Shift+\ generates the decimal Unicode 124 glyph. In the APL64 font, the decimal Unicode 124 and 8739 glyphs appear graphically identical.



In prior versions for the EN-US and EN-UK keyboard definitions, the Magnitude and Residue glyph resulted from the Alt+M and Shift+\ keystrokes. Therefore, the Virtual Keyboard has been modified so that it is consistent with the physical keyboard operation.

A statement executed in the History pane no longer moves the focus to the Debug Pane

When the interpreter state is suspended and the Debug pane is visible, executing an APL statement, or clicking Enter/Return in the History pane will not move the focus to the Debug while that APL statement is being processed by the interpreter.

A statement executed in the History pane no longer transiently hides the Debug pane

When the interpreter state is suspended and the Debug pane is visible, executing an APL statement, or clicking Enter/Return in the History pane will cause the Debug/SI panes to be transiently hidden. The Session | Debug | Automatically Show Debug/SI Panes when Suspended option now operates as expected.

Editors and Debug/SI pane horizontal splitter bar values are now saved

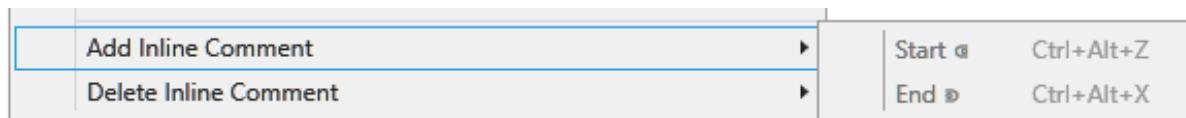
When the horizontal splitter bars for docked editors or Debug/SI panes are moved by the programmer, these values are now saved in the APL64 configuration file.

Main Window Toolbar Height no longer changes

The height of the main window toolbar no longer changes when the focus moves to a floating editor pane while in debug mode.

Edit | Add Inline Comment and Edit | Delete Inline Comment Submenus

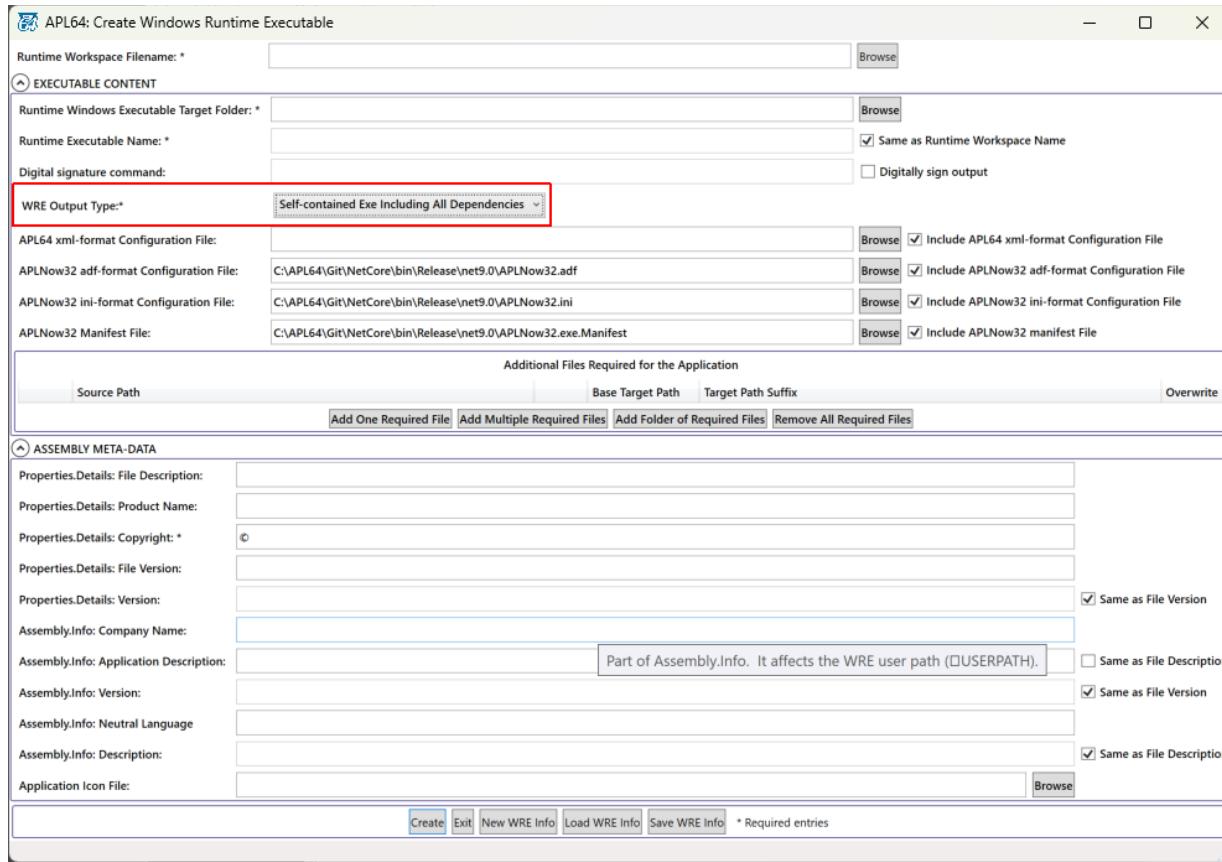
These menu items now use the inline comment glyphs:



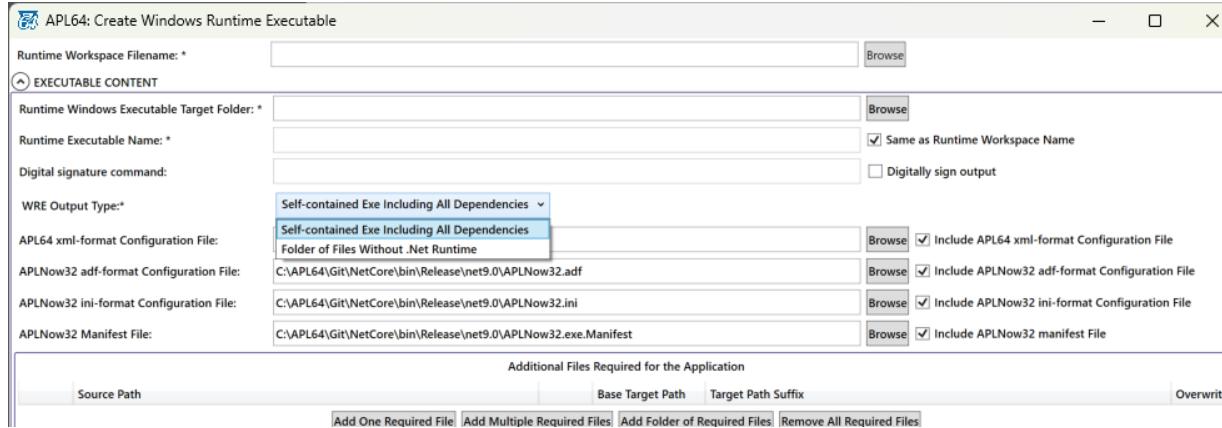
WRE and CPC Utility Modifications

WRE creation option to exclude the .Net Runtime

The WRE creation dialog provides a new option, 'WRE Output Type', to include/exclude the .Net Runtime from the resulting WRE:



Available WRE .Net Runtime options:



Description of the WRE .Net Runtime options:

Self-contained Exe Including All Dependencies

The output is a single Windows exe-format file which is ready to run on the target workstation. The exe-format file contains the APL64 runtime files and the appropriate .Net Runtime libraries. Because this

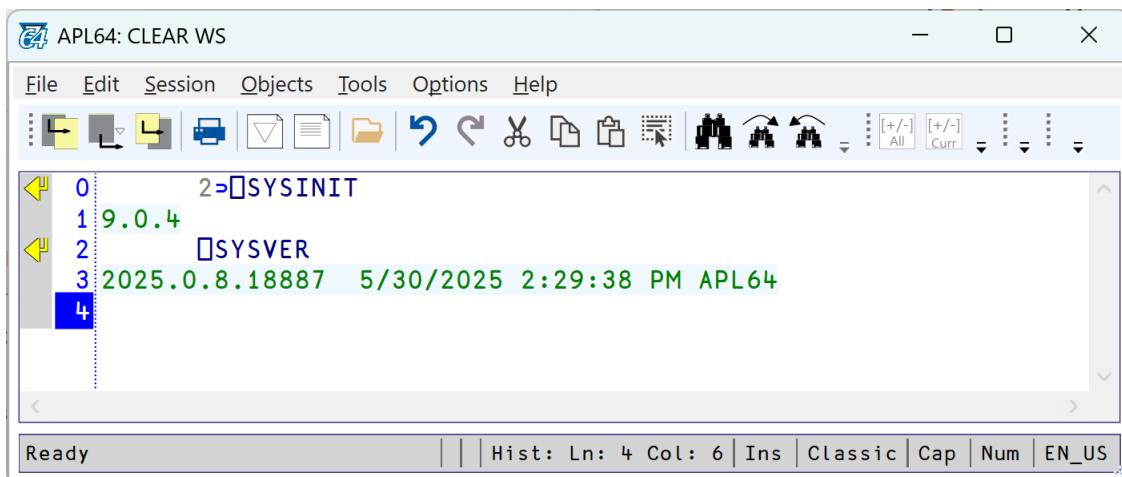
option produces a ready-to-run application, no additional components need to be installed on the target workstation by the end user of the WRE.

Folder of Files Without .Net Runtime

This new option creates a folder of files which the APL64 programmer may use to deploy the WRE application without including the .NET Runtime libraries, reducing the resulting WRE by approximately 150 MB.

Using this option requires that the appropriate version of the .NET Runtime libraries is installed on the target workstation by the WRE end user. When the WRE application is run, missing .NET Runtime libraries will be detected, and a prompt to download and install them will be presented.

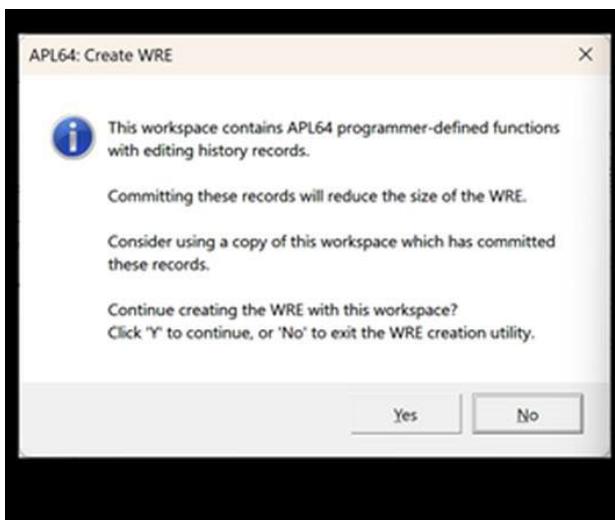
The output is a folder named ‘Publish’ containing the Windows exe-format file and the APL64 runtimes files necessary for the application. Because this option does not include the .Net runtime, the deployed application will not run unless the appropriate version of the .Net Runtime is installed on the target workstation. The APL64 programmer may determine the appropriate version of the .Net Runtime from the second element of SYSINIT:



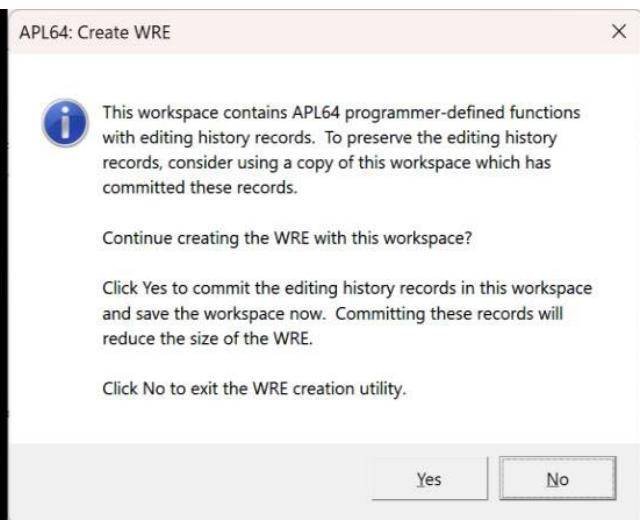
Option to commit function editing history within the CPC and WRE utilities

When the WRE or CPC creation dialog is presented, the functions in the loaded workspace will be checked for APL64 programmer editing modifications which have not yet been committed. These uncommitted editing modifications increase the size of those functions. The message box which alerts the APL64 programmer now contains an option to commit these function editing changes, save the workspace and minimize the size of those functions in the WRE or CPC runtime application.

APL64 2025.0.5



APL64 2025.0.8



CSR Extra new line character in the result removed

Prior version with extra new line character:

Screenshot of the APL64 IDE showing a workspace named 'SquareRoot.ws64'. The code editor displays the following APL code:

```
0      sysver
1 2025.0.5.18789 4/10/2025 2:26:25 PM APL64
2 )fns
3 SquareRoot  SquareRoot1  SquareRoot2  SquareRoot3  SquareRoot4
4 )csr '*'
5 XStmtEndMethod : :public (double@Z)+SquareRoot (double@X)
6 XStmtEndMethod : :public (double@Z)+SquareRoot1 (double@X)
7 XStmtEndMethod : :public (double@Z)+SquareRoot2 (double@X)
8 XStmtEndMethod : :public (double@Z)+SquareRoot3 (double@X)
9
10 XStmtEndMethod : :public (double@Z)+SquareRoot4 (double@X)
11
12
13
14
15 |
```

The code editor interface includes a toolbar, a menu bar with File, Edit, Session, Objects, Tools, Options, Help, and a status bar at the bottom.

Update version without extra new line character:

The screenshot shows the APL64 IDE interface. The title bar reads "APL64: C:\CPC example\SquareRoot.ws64". The menu bar includes File, Edit, Session, Objects, Tools, Options, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find. The main workspace displays the following APL code:

```
0      sysver
1 2025.0.6.22480 5/16/2025 4:29:26 PM APL64
2 )Load C:\CPC example\SquareRoot.ws64
3 "C:\CPC example\SquareRoot.ws64" LAST SAVED 5/16/2025 6:30:48 PM
4 )fns
5 SquareRoot  SquareRoot1  SquareRoot2  SquareRoot3  SquareRoot4
6   )csr '*'
7 XStmtEndMethod : :public (double@Z)+SquareRoot (double@X)
8 XStmtEndMethod : :public (double@Z)+SquareRoot1 (double@X)
9 XStmtEndMethod : :public (double@Z)+SquareRoot2 (double@X)
10 XStmtEndMethod : :public (double@Z)+SquareRoot3 (double@X)
11 XStmtEndMethod : :public (double@Z)+SquareRoot4 (double@X)
12 XStmtEndMethod : :public (double@Z)+SquareRoot4 (double@X)
13 |
```

The status bar at the bottom shows "Ready", "Hist: Ln: 13 Col: 6", and "Classic" mode.

CSR: 'XStmtEndMethod' output prefix removed for 'head' option

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

Toolbar icons: Open, Save, Print, Find, Copy, Paste, Undo, Redo, Cut, Copy, Paste, Find, Replace, Select All, New Window, Close, Minimize, Maximize, Exit.

```
0   ⌂vr 'SquareRoot'
1   ▽ :public double@Z+SquareRoot double@X
2 [1] Z+X*0.5
3   ▽
4
5   'head'⌂CSR 'SquareRoot'
6 XStmtEndMethod : :public double@Z+SquareRoot double@X
7   ⌂sysver
8 2025.0.5.28606 4/10/2025 7:55:36 PM APL64
9
```

Buttons: New, Open, Save, Print, Find, Copy, Paste, Undo, Redo, Cut, Copy, Paste, Find, Replace, Select All, New Window, Close, Minimize, Maximize, Exit.

Ready | Hist: Ln: 6 Col: 17 | Ins | Classic | Num | EN_US

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

Toolbar icons: Open, Save, Print, Find, Copy, Paste, Undo, Redo, Cut, Copy, Paste, Find, Replace, Select All, New Window, Close, Minimize, Maximize, Exit.

```
0   ⌂vr 'SquareRoot'
1   ▽ :public double@Z+SquareRoot double@X
2 [1] Z+X*0.5
3   ▽
4
5   'head'⌂CSR 'SquareRoot'
6 :public double@Z+SquareRoot double@X
7   ⌂sysver
8 2025.0.6.23565 5/24/2025 5:05:36 PM APL64
9
```

Buttons: New, Open, Save, Print, Find, Copy, Paste, Undo, Redo, Cut, Copy, Paste, Find, Replace, Select All, New Window, Close, Minimize, Maximize, Exit.

Ready | Hist: Ln: 6 Col: 0 | Ins | Classic | Num | EN_US