

Advantages of APL64

Contents

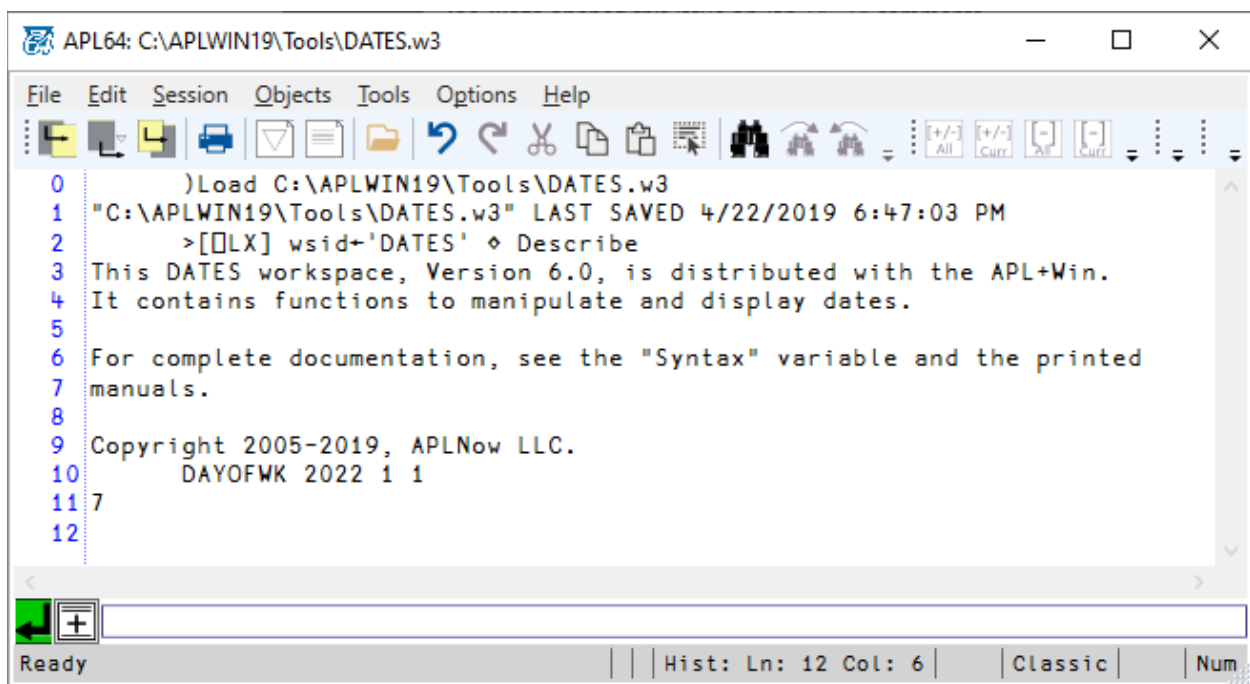
Excellent Compatibility with APL+Win.....	3
Increased Workspace Size Up To the Available Workstation Memory	3
Increased Variable Size	4
Performance Optimizations of APL Primitives and System Functions.....	4
Unified Variable Editor To Review & Edit Simple, Homogeneous, Heterogeneous, and Nested Variables .	4
Full Unicode Support.....	5
Multiple Formats to Display the History in the Developer Version	6
Classic.....	6
Sequential	7
Grouped	7
Separated.....	8
Additional Text Selection Options - Linear, Rectilinear and Row	8
Floating or Docked Panes – History, Function & Variable Editor, Debugger	10
Multi-row Executable APL Statements	11
Runtime Deployment Simplified – APL64 Application in a Single File	12
Context-Sensitive APL Developer Documentation	13
Optional Tooltips for Toolbars, Menus and Panes.....	15
New APL String Data Type Supported.....	17
String Definition	17
Defining a String Variable in APL64.....	17
New User-defined Tools Features.....	19
New Configure Host Error Mappings Dialog	20
Enhanced Find/Replace Dialog	21
☐ XL Exchange Data Between APL64 and Excel Worksheet Without Excel.exe.....	22
☐ EDSS Edit an APL Array in a Worksheet Editor.....	24
)EDSS Edit an APL Array in a Worksheet Editor	24
Find In Functions In Workspace Utility	24
Latent Function Stops	25
UTC Time Supported Universally in APL64	26

New menu Session > Configuration Settings > Apply Default Settings	26
☐ XML to Create, Edit, Read and Write XML-format Documents	28
APL64 Idioms Manager	28
Updates to Existing System Functions, Commands and Variables	30
Many New System Functions.....	31
Enhanced Dialogs	32
APL Statements History – More options than Gather action	32
History Pane – Optional Row Numbers	33
Colors Dialog	33
Library Definitions.....	34
History Log Enhanced.....	35
Keyboard Definition	36
Convert APL+Win Configuration File.....	37
APL64 Now Supports .NET Release 6.0 (x64).....	37
HttpClient Now Used	37
New ☐ Path System Function	37
New ☐ Pathcase System Variable.....	38
New Save As Xlxs Option in JEDSS/ ☐ EDSS.....	38
New APL64 System Highlights Feature in the Status Bar.....	39
Insert/Replace Keyboard Mode Supported in APL64	41
Find in Functions in Workspace Utility	41
'Last Fix Date' Column Added to the Filtered List of Functions	41
New Option is Provided to "Show Function Definition(s) for Selected Row(s)" in the Function List	42
☐ devcap Set to an Empty Vector.....	43
The Find/Replace Tool Replaces the Find/Replace Dialog.....	44
Rendering Large Output to History and ☐ xso System Variable.....	44
Insert APL Idiom Dialog is Now Not Modal	46
The Caption and hwndmain Properties for the ☐ wi System Object (#) are Deprecated	47
New menu: Objects Editors Pane Format Shrink Editor Area When All Editors Are Floated	47
New menu: Objects Editors Pane Format Shrink Editor Area When All Editors Are Floated	47
The ☐ wi Interface to the APLNow32.exe was Optimized to Provide Better Stability.....	47
New ☐ NFE 'SHA512Hash' Action	48

Session Log Timestamps Displayed.....	48
New Options Session Log Delete All Session Log Files Action	48
Variable and Function Names Can Begin With the Underscore Character	49
New System Functions: ☐ flock, ☐ fflush, ☐ nlock, ☐ nflush.....	49
New System Functions ☐ ask and ☐ askn	49
Options History Log History Log Options Menu Renamed	50
☐ EDITEVENTS/ ☐ EDITINFO and Documentation Updated.....	51
☐ SKD and Documentation Updated	51
Performance of ☐ Rowfind System Function is Improved	51
The Display of the MessageBox Warning is Now Avoided When Importing an Empty Session Log File....	51

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.



```

0      )Load C:\APLWIN19\Tools\DATES.w3
1      "C:\APLWIN19\Tools\DATES.w3" LAST SAVED 4/22/2019 6:47:03 PM
2      >[☐LX] wsid+'DATES' ♦ Describe
3      This DATES workspace, Version 6.0, is distributed with the APL+Win.
4      It contains functions to manipulate and display dates.
5
6      For complete documentation, see the "Syntax" variable and the printed
7      manuals.
8
9      Copyright 2005-2019, APLNow LLC.
10     DAYOFWK 2022 1 1
11 7
12

```

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

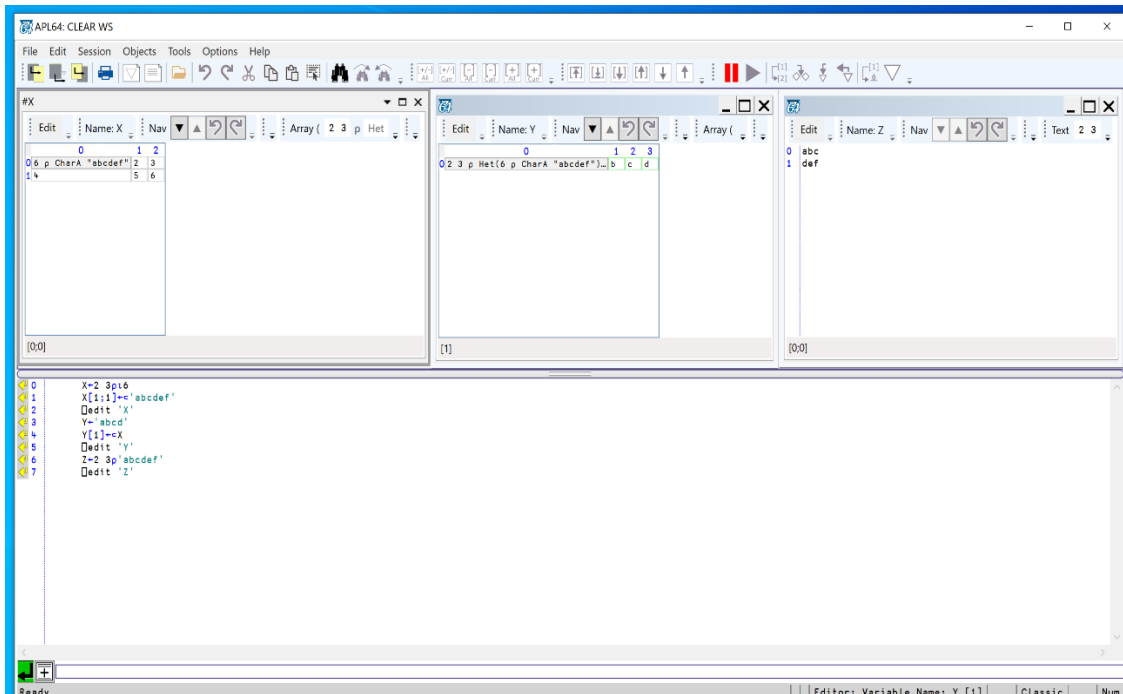
APLWin - Maximum homogeneous variable size less than 2 GB and in practice 500 MB or less.

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

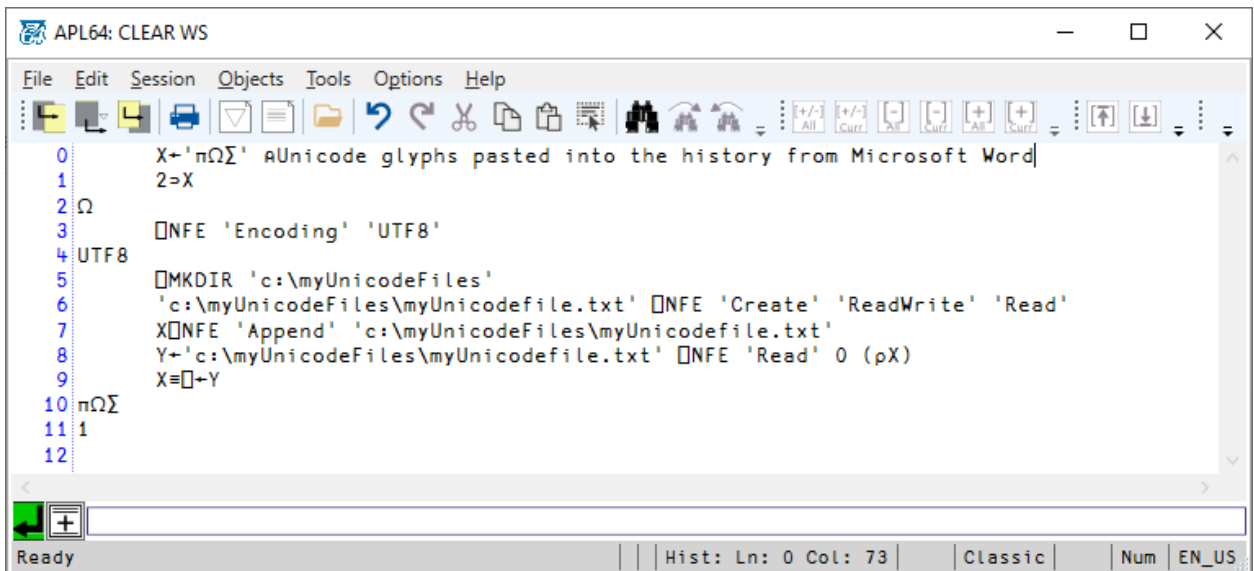
The new unified variable editor in APL64 can review and edit an APL variable of any type, including heterogeneous and nested variables. The APL+Win graphical editor is limited to simple text or simple numeric variable editing.



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.

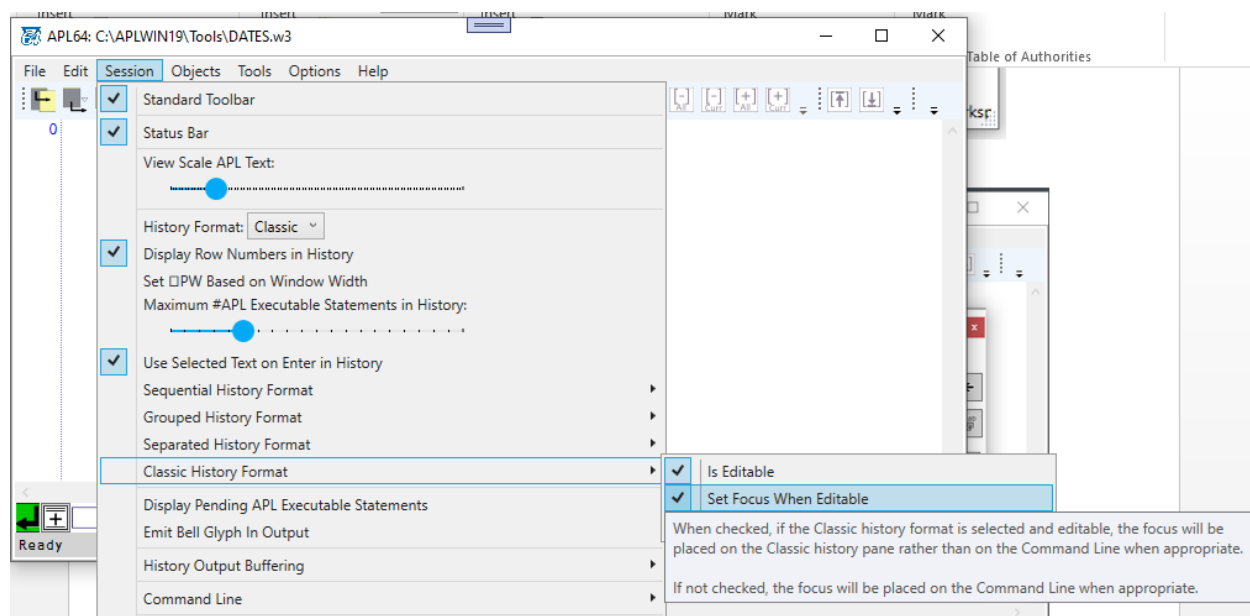
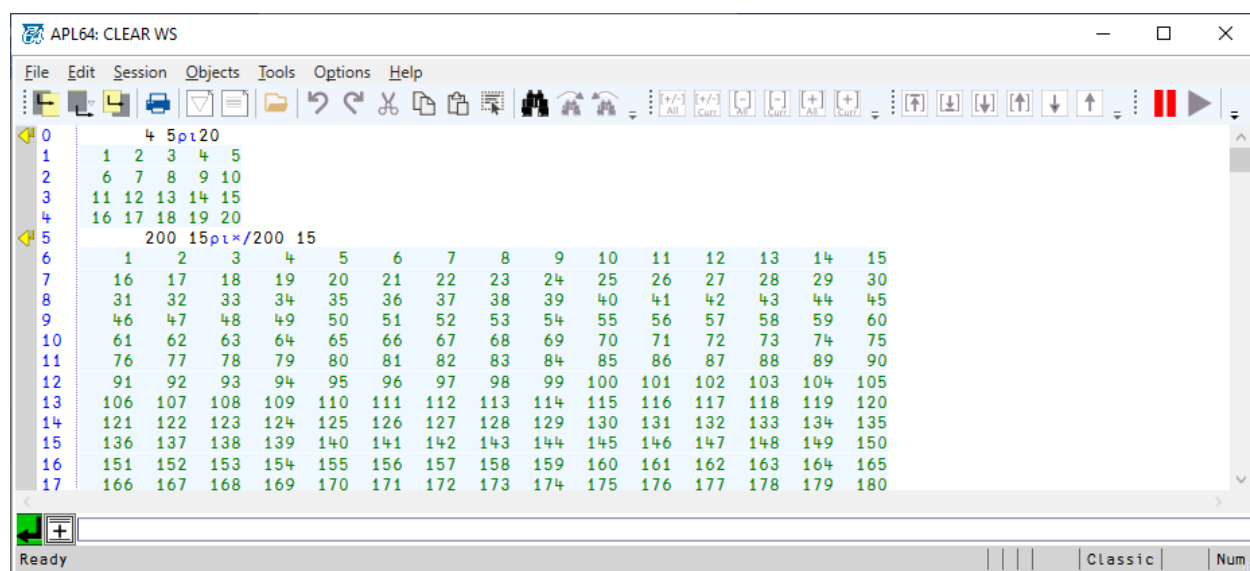
Unicode glyphs in APL64 variables or functions will be directly displayed in the APL64 Developer version. Use the ☐NFE system function to read/write Unicode data in native files.



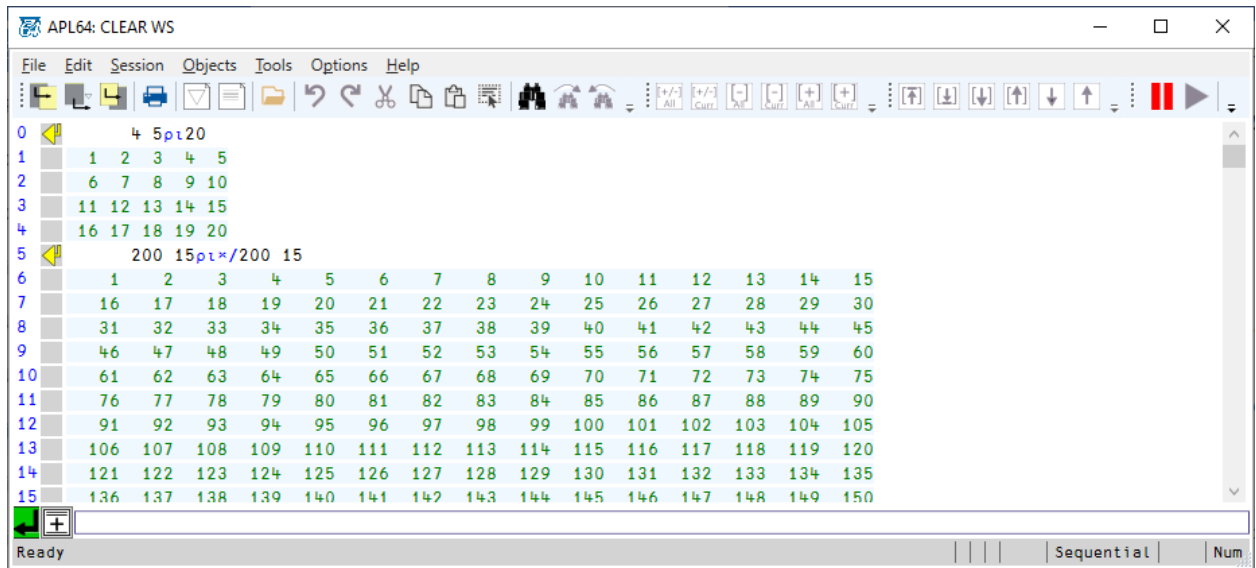
Multiple Formats to Display the History in the Developer Version

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.

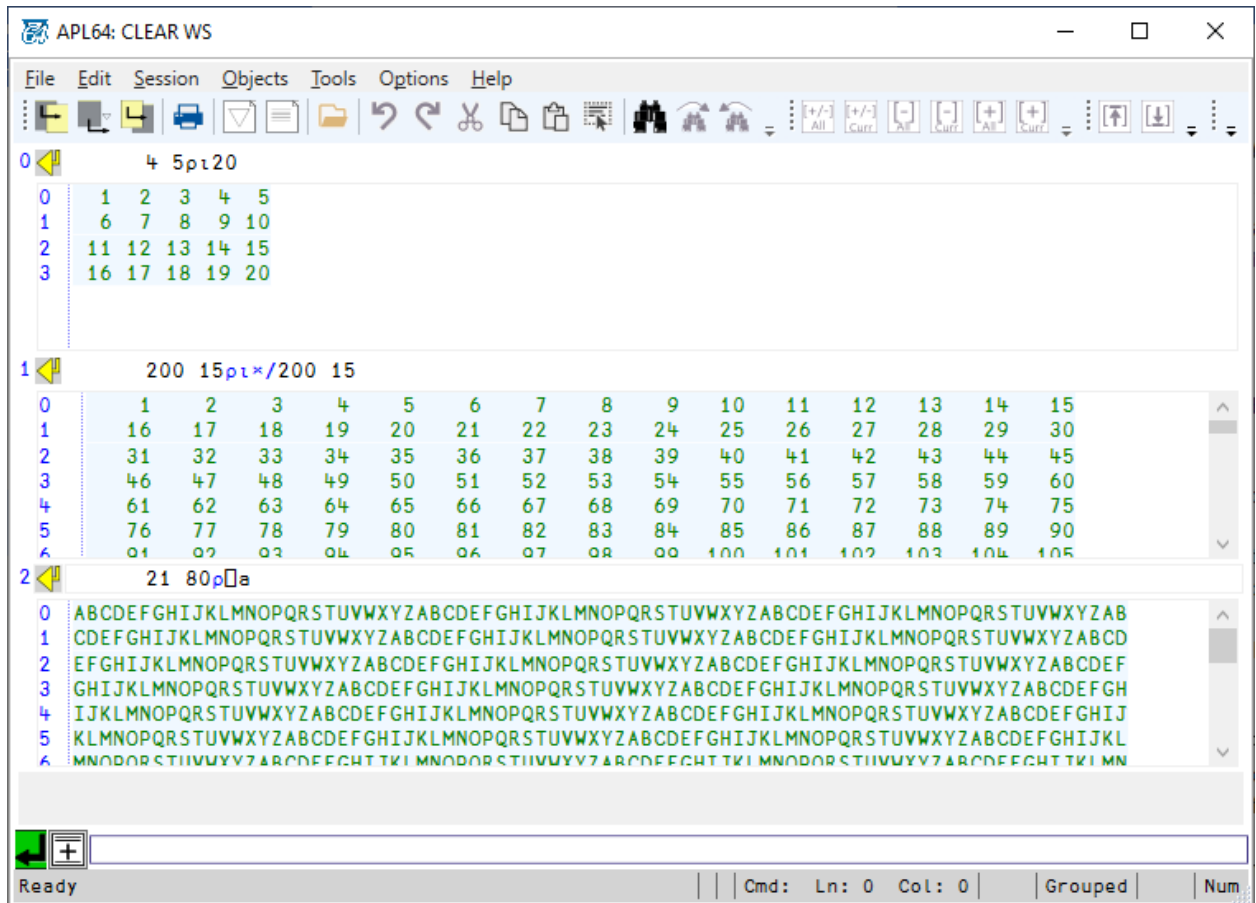
Classic – This format is analogous to the APL+Win ‘session’ document-style format with the information in execution order. The APL64 classic history format has an ‘Is Editable’ option. When Session > History Format: Classic, Session > Classic History Format > Is Editable is checked and Session > Classic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



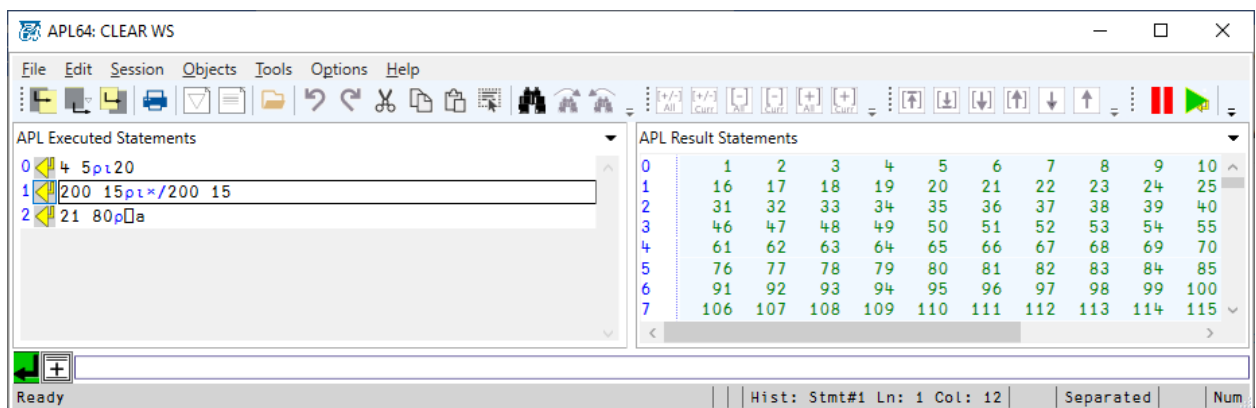
Sequential – Developer input and interpreter output is displayed as an array of rows.



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.



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.

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

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

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
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180

Ready | Hist: Ln: 9 Col: 60 | Classic | Num

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

4 5pt20

1 1 2 3 4 5

2 6 7 8 9 10

3 11 12 13 14 15

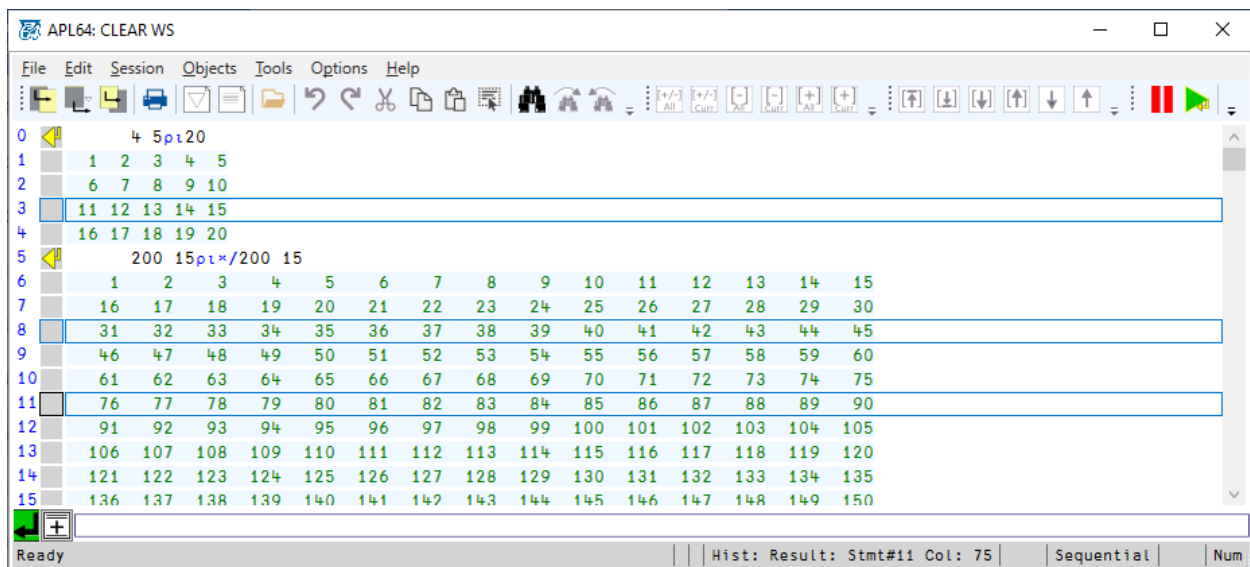
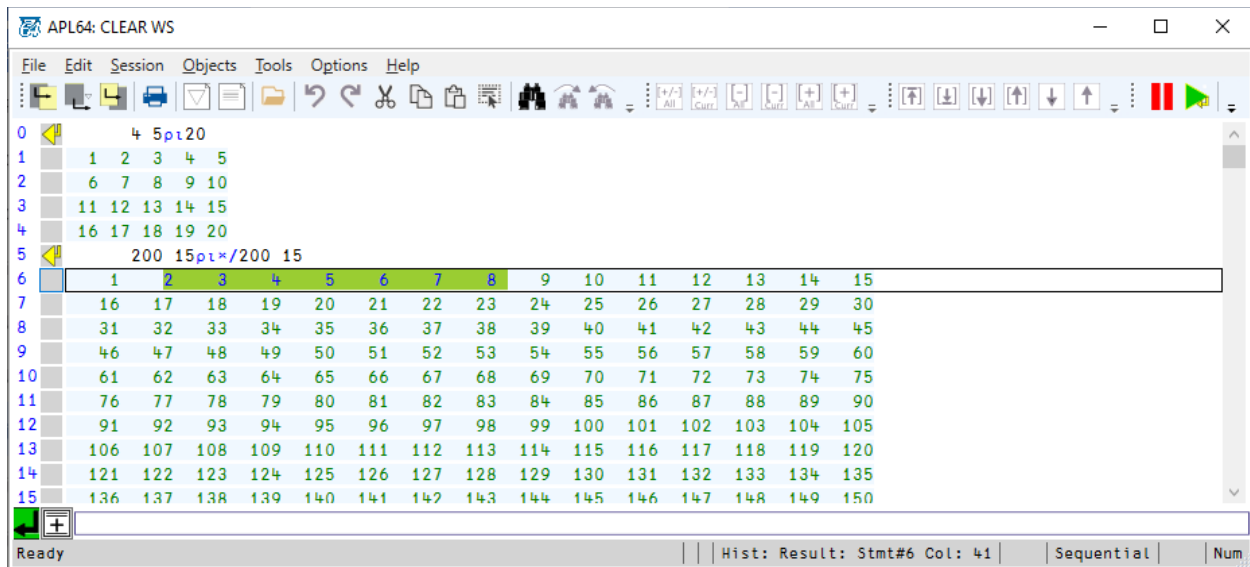
4 16 17 18 19 20

5 200 15pt*/200 15

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
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180

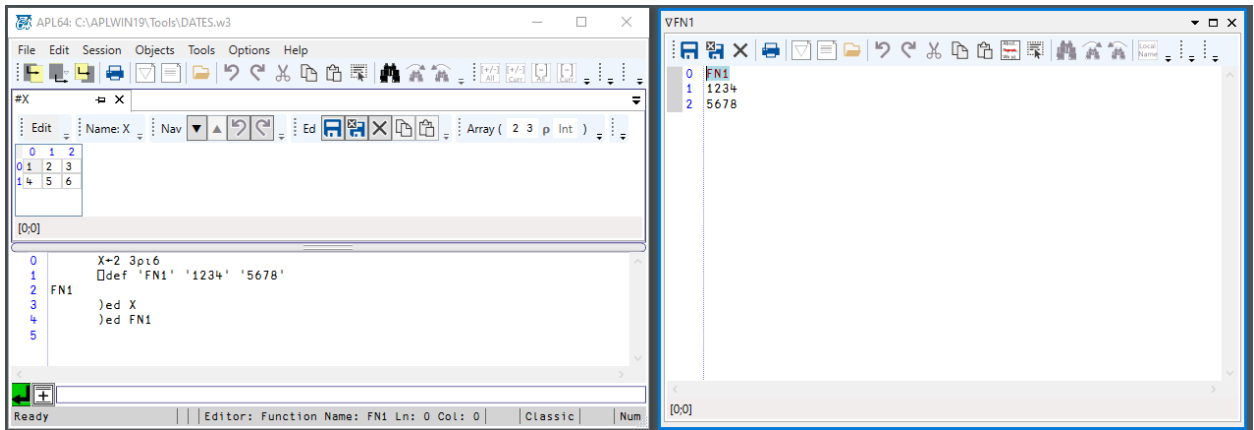
Ready | Hist: Ln: 3 Col: 12 | Classic | Num

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.

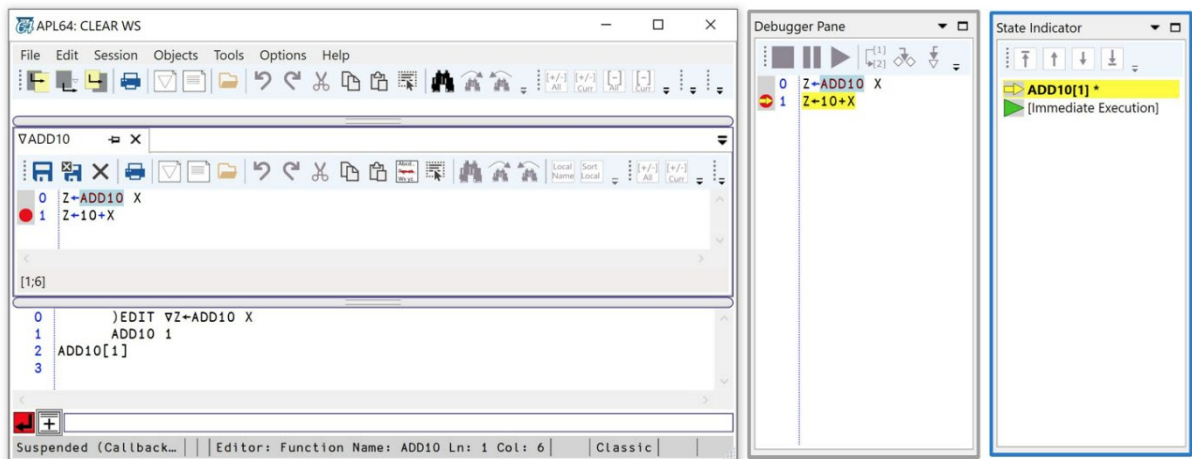


Floating or Docked Panes – History, Function & Variable Editor, Debugger

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



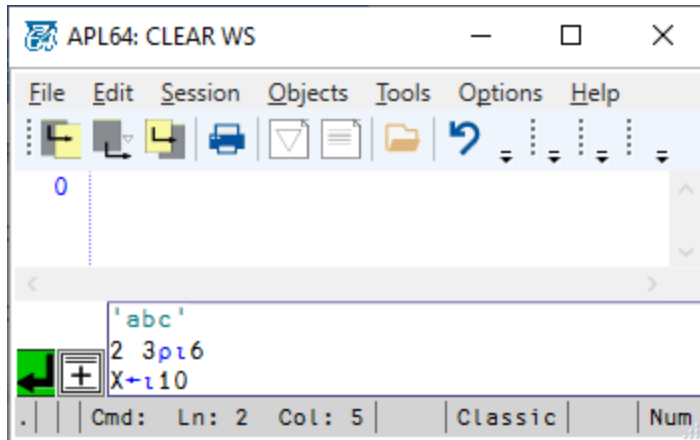
The debugged function pane and the state indicator panes may now be independently ‘floated’ separately from the main window:



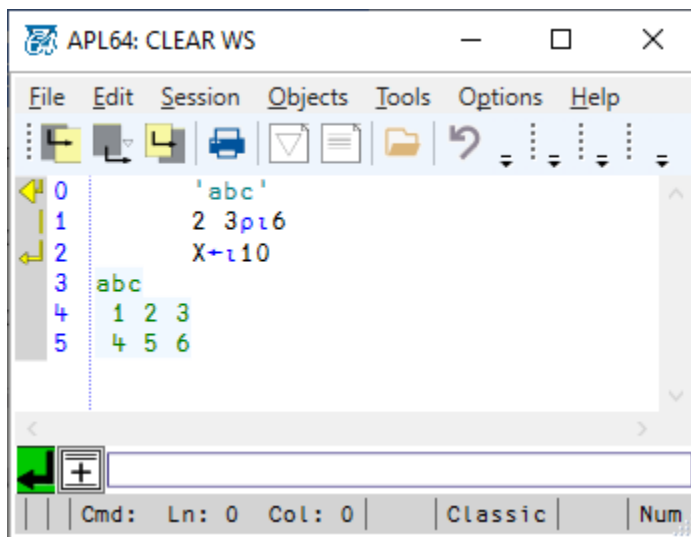
Multi-row Executable APL Statements

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.

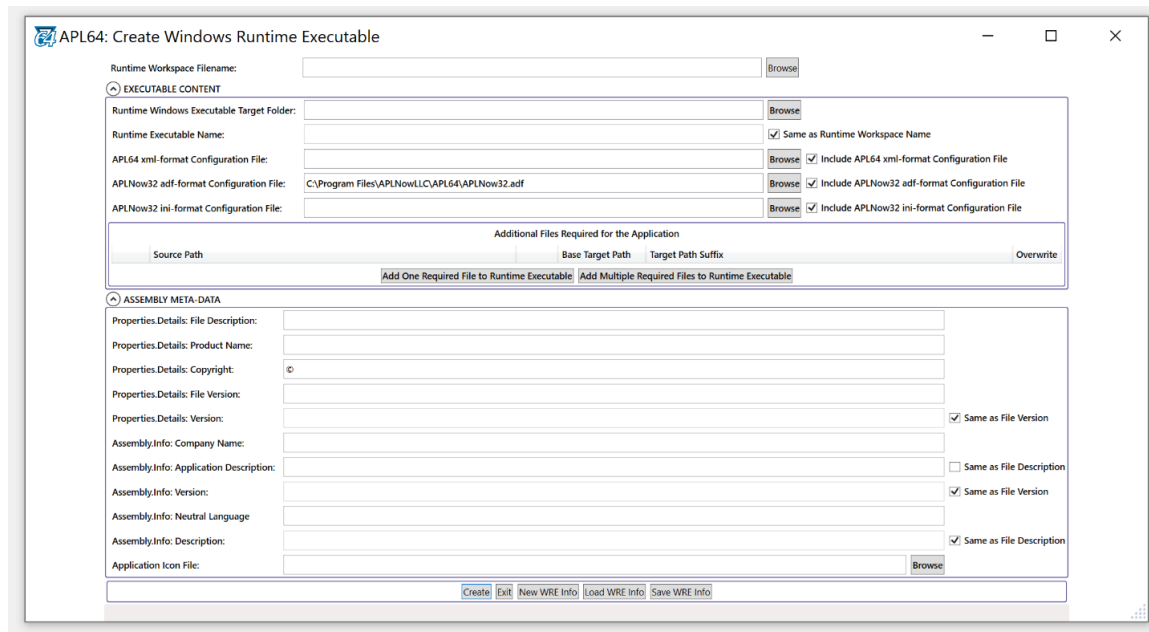
All features of APL64 which are available in the Windows operating system environment may be used by the APL64 programmer in the development of an APL64 runtime application.

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

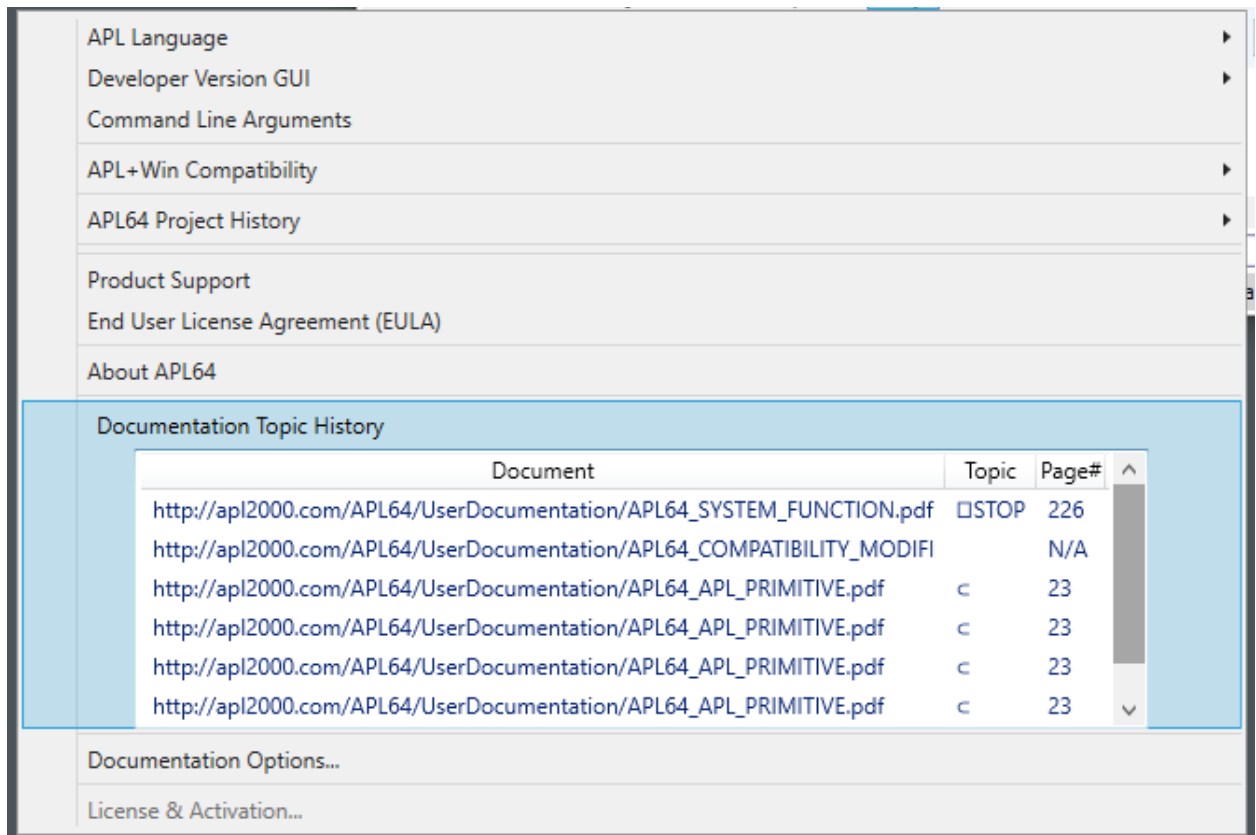
The published APL64 runtime executable is self-contained, including all of the necessary components of a complete APL64-based Windows application, so that it can be run via a double left mouse click or incorporated into a Windows program short-cut.



Context-Sensitive APL Developer Documentation

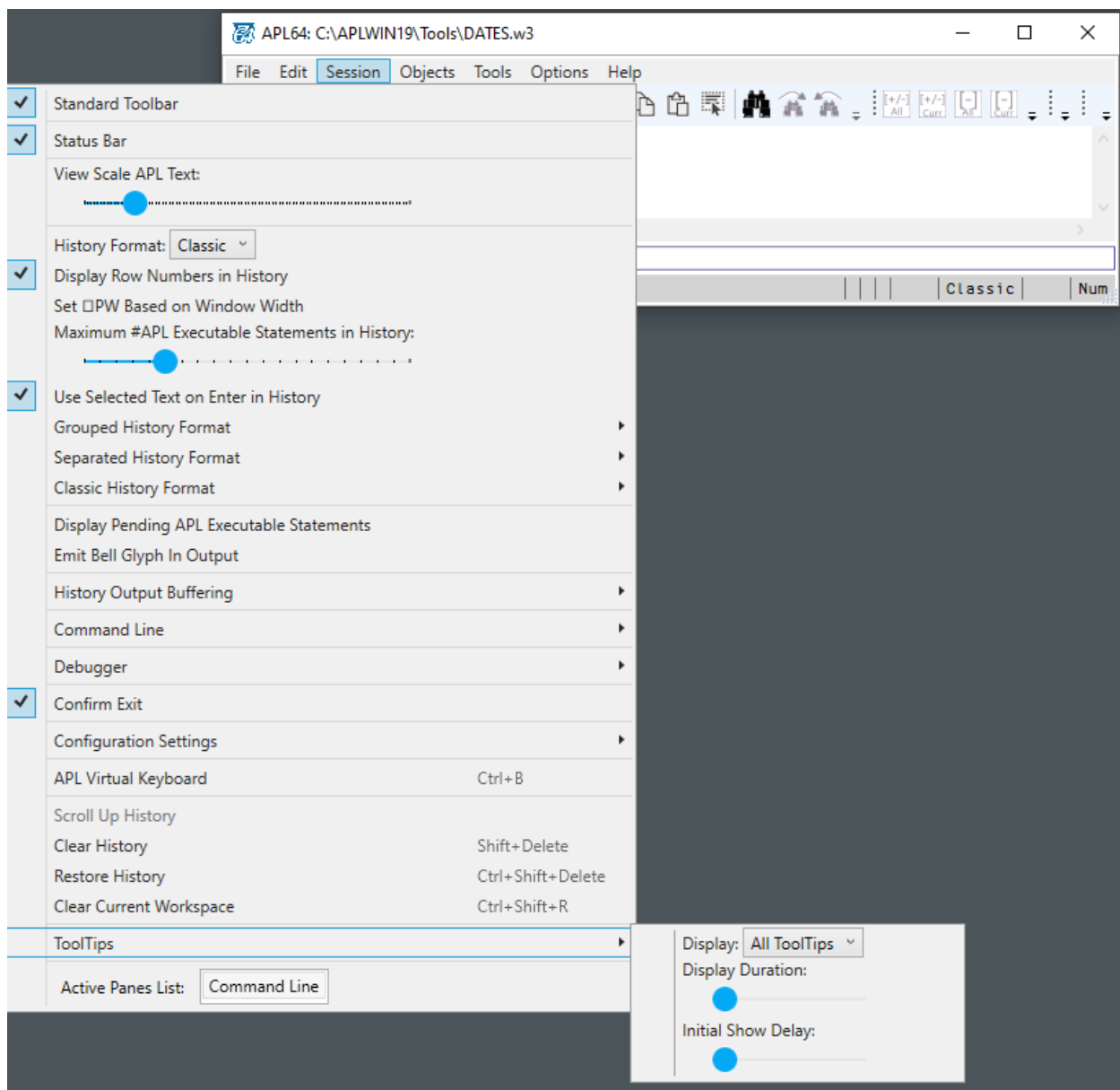
Context-sensitive documentation for APL primitive functions, system variables, functions and commands has been implemented in APL64.

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.

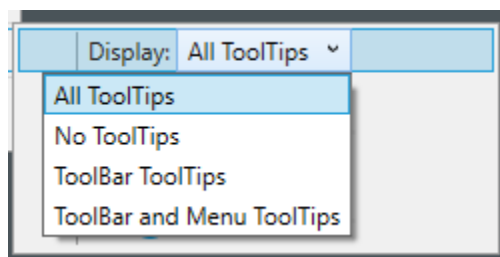


Optional Tooltips for Toolbars, Menus and Panes

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



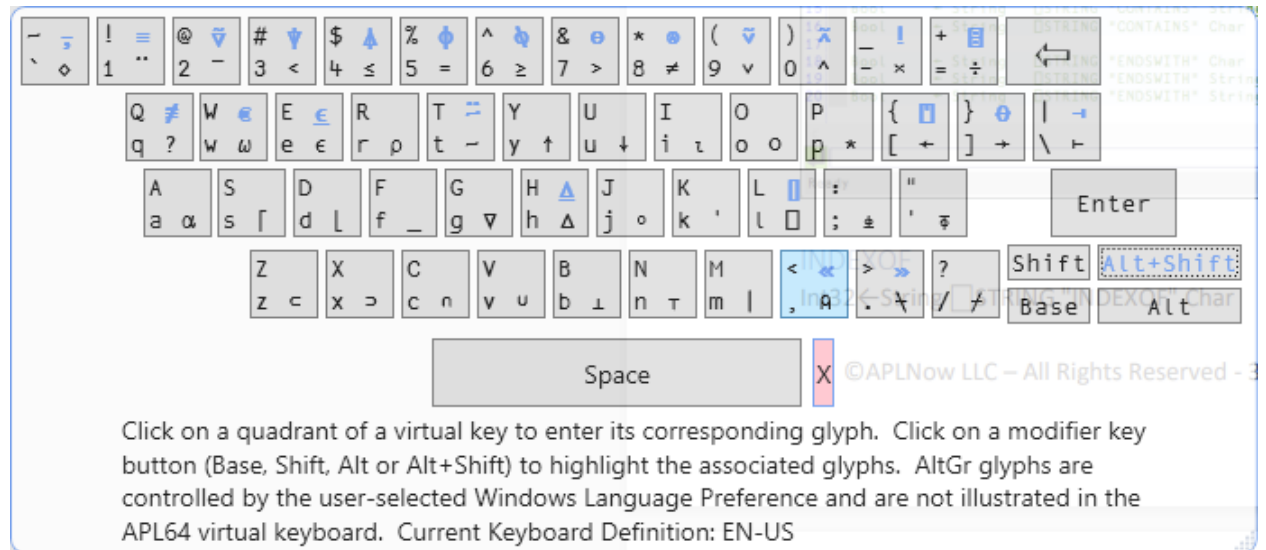
New APL String Data Type Supported

String Definition

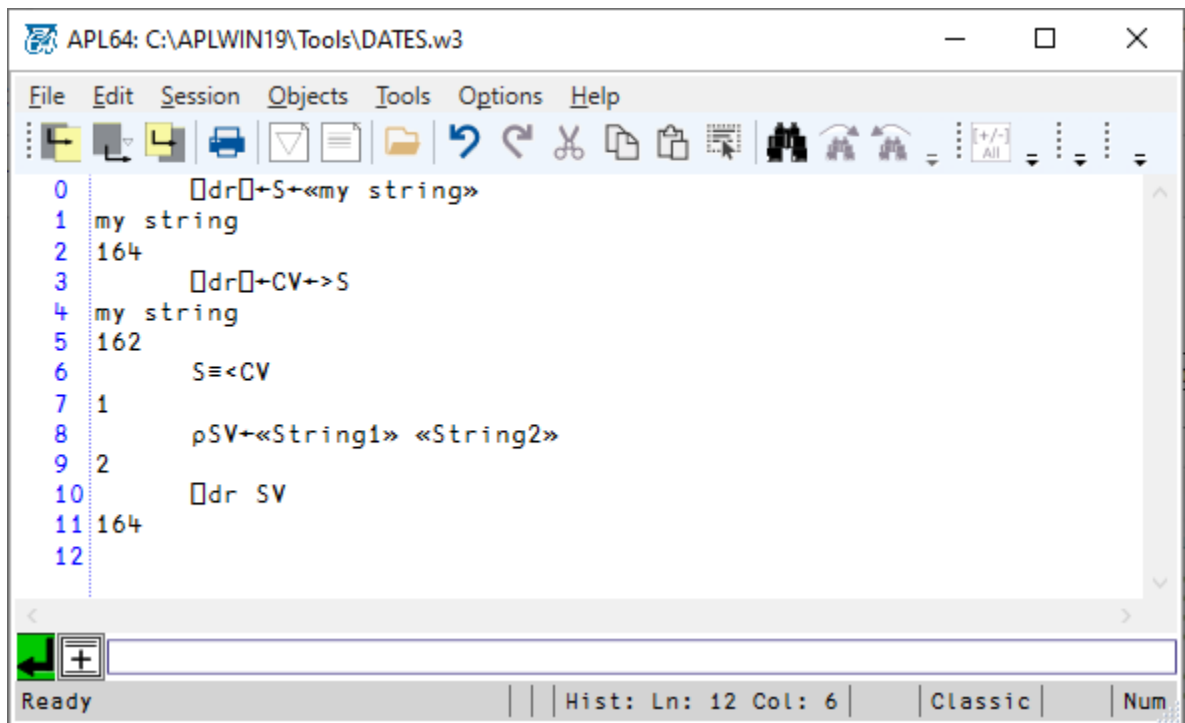
An APL string is an ordered group of characters treated as a scalar, with delimiters «...».

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.



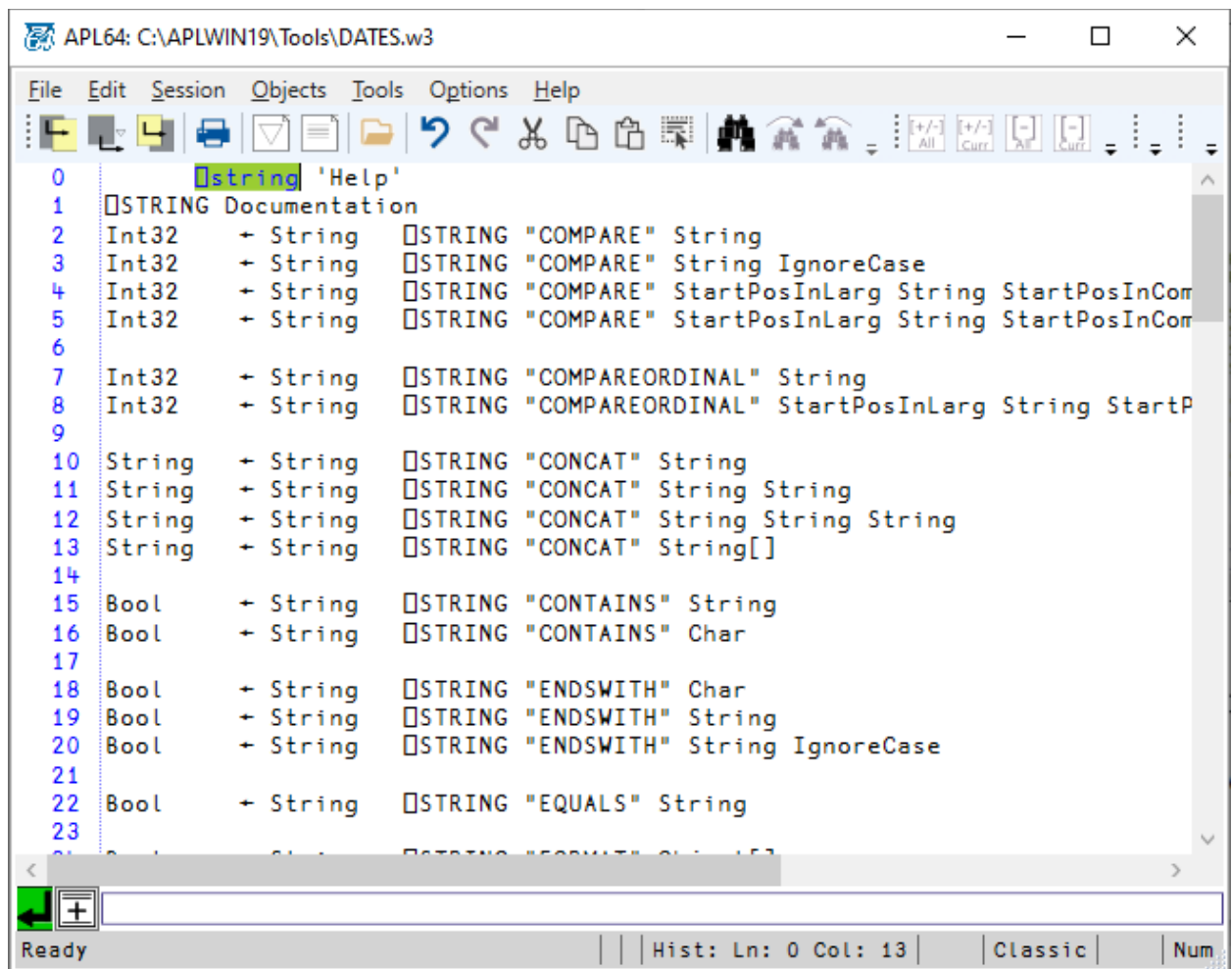
To define a string, enclose the text within the « and » glyphs. The APL64 enstring and destrung functions are available for conversion between the APL character vector and string formats. APL64 supports string scalars and arrays.



```
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
```

Ready | Hist: Ln: 12 Col: 6 | Classic | Num

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



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.

APL64: Configure User Tools

User Tool Definitions

Menu Text	Action Text	Key	Shift	Alt	Control	Windows
New User Tool 1	⌘Tool Action Text supports ▾ and ωOptionalArgumen	D	☒	☐	☒	☐

New Tool Delete Selected Tools Delete All Tools Move Selected Tool Up Move Selected Tool Down

User Tool Definition

Menu Text: New User Tool 1

Action Text: ⌘Tool Action Text supports ▾ and ωOptionalArgumentDesc

Keyboard Key:

Modifier Keys: ☒ Shift ☐ Alt ☒ Control ☐ Windows

User Tool Documentation

New User Tool Documentation Line1
New User Tool Documentation Line2

The Windows operating system, language, keyboard definition and installed software may affect available k

OK Cancel

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

If the user tool requires optional runtime arguments, include runtime argument names in the user tool action text.

A user tool runtime argument name must have a ω glyph prefix optionally followed by an argument name, e.g. ω or ωOptionalRuntimeArgName.

If the user tool requires the current text selection as a runtime argument, include the ▾ glyph among the User Tool arguments in the User Tool Action Text.

At runtime the current selection will be obtained from the APL64 Developer session GUI control which has the keyboard focus.

Multiple runtime tool arguments or tool argument names are supported.

Do not include "α" in the action text.

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.

APL64: User Tool Argument Values

User Tool Menu Text: New User Tool 1

User Tool Action Text: ⌘Tool Action Text supports ▾ and ωOptionalArgumentDesc

User Tool Documentation:

New User Tool Documentation Line1
New User Tool Documentation Line2

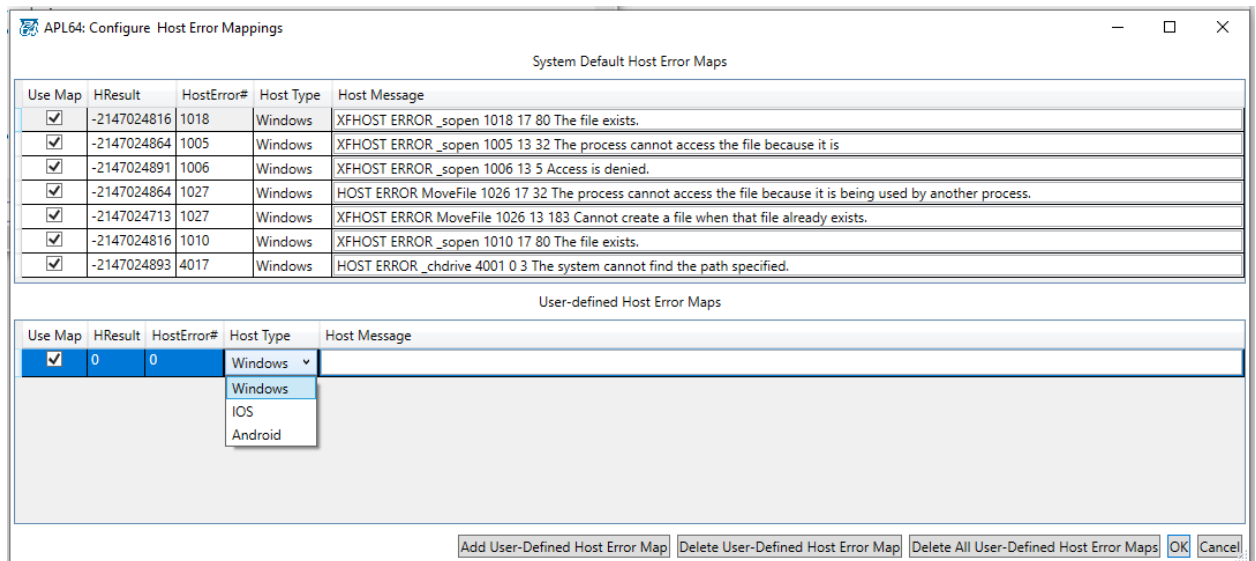
Current Selection:

Argument Name	Argument Value
Current Runtime Selection	Current Runtime Selection
ωOptionalArgumentDesc	

OK Cancel

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 Mappngs...* dialog.

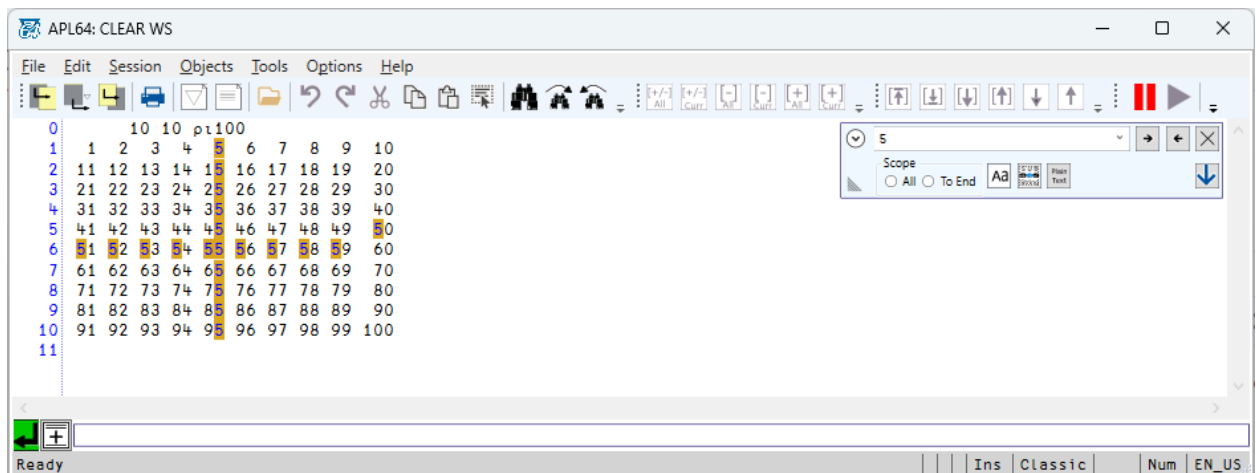


Enhanced Find/Replace Dialog

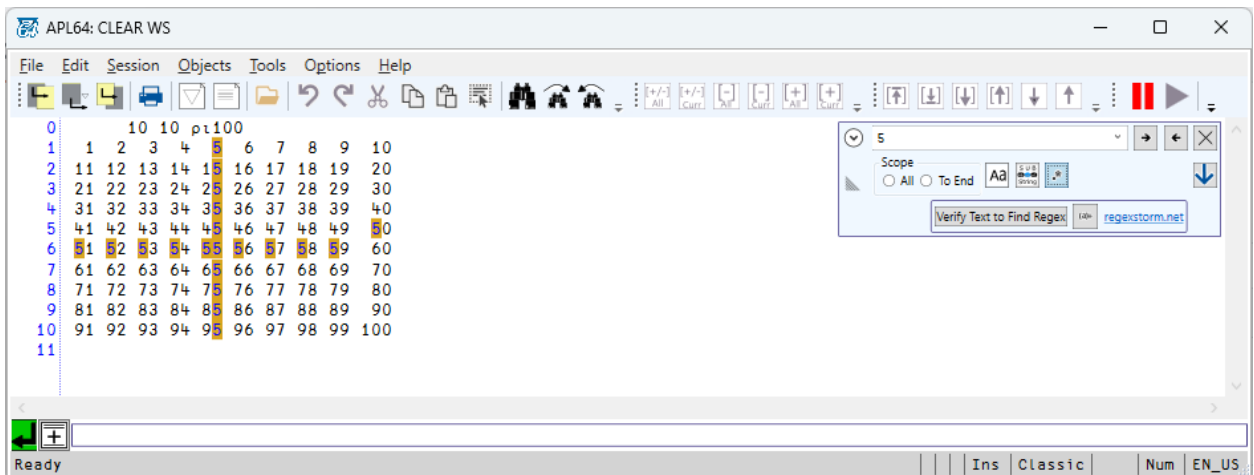
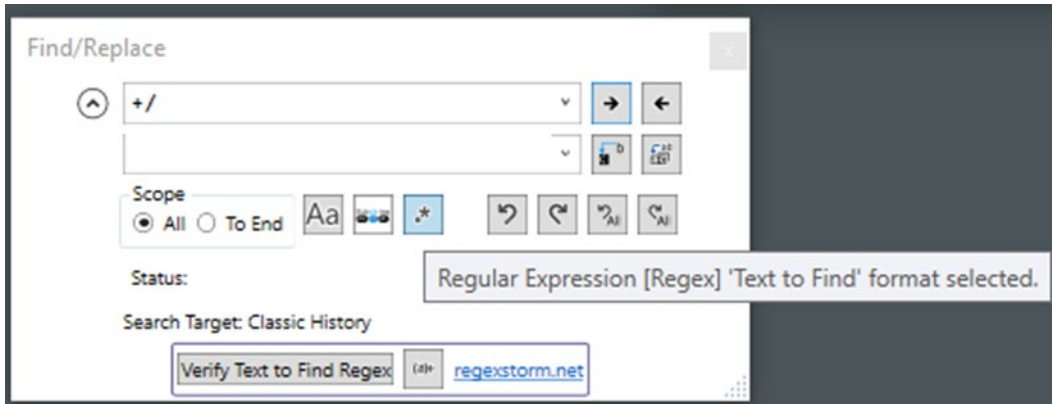
The APL64 Find/Replace dialog supports options for:

- Find and Replace
- Scope: All, From Current Cursor to End and for functions uncollapsed regions
- Undo, Redeo, 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:



□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 ← □XL Action [ActionArg1] ...

Actions (not case-sensitive):

Action	Description	Reqd. Action Args.
'?'	□XL Detailed syntax	None

'Help'	☐ 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	'AplString', 'AplChar'	'ToApl'
6	dateTimeConv	N	'AplDateTime', '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:

```

1  [?XL Documentation:
2  (bRes aRes)← [?XL 'FromApl' workbookPath worksheetName inclRows inclCols dateTimeConv sourceAplValues
3
4  bRes: 0/Fail, 1/OK
5  aRes: ErrMsg (APL64 character vector)/Fail, ''/OK
6  FromApl: [?XL Action code: Fill worksheet with an APL64 array
7           The source APL64 elements are accessed in ravel order
8  workbookPath: Full path of target Excel workbook. If the workbookPath is not found, it will be created
9  worksheetName: Full name of target worksheet. If worksheetName is not found, it will be created
10 inclRows, inclCols: Indices of rows/columns of worksheet to fill with APL64 array data. Order is significant. [?IO is significant.
11 dateTimeConv: 'AplDateTime': 7-elt APL [?TS date time will be converted to XL DateTime
12                'XLDateTime': Floating point value relative to midnight 12/30/1899 (effectively no conversion)
13
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: 'AplString': XL text converted to APL string
29                 'AplChar': XL text converted to APL char(s)
30                 Applies to Excel workbook cells with Value Type = Text.
31 dateTimeConv: 'AplDateTime': 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': Floating point value relative to midnight 12/30/1899 (effectively no conversion)
34                'DateTime cell determined by cell.ValueType (number) and cell.NumberFormat (date, datetime, time)
35 xlEmptyCellAplValue: Any APL64 object
36 xlCellErrorAplValue: Any APL64 object

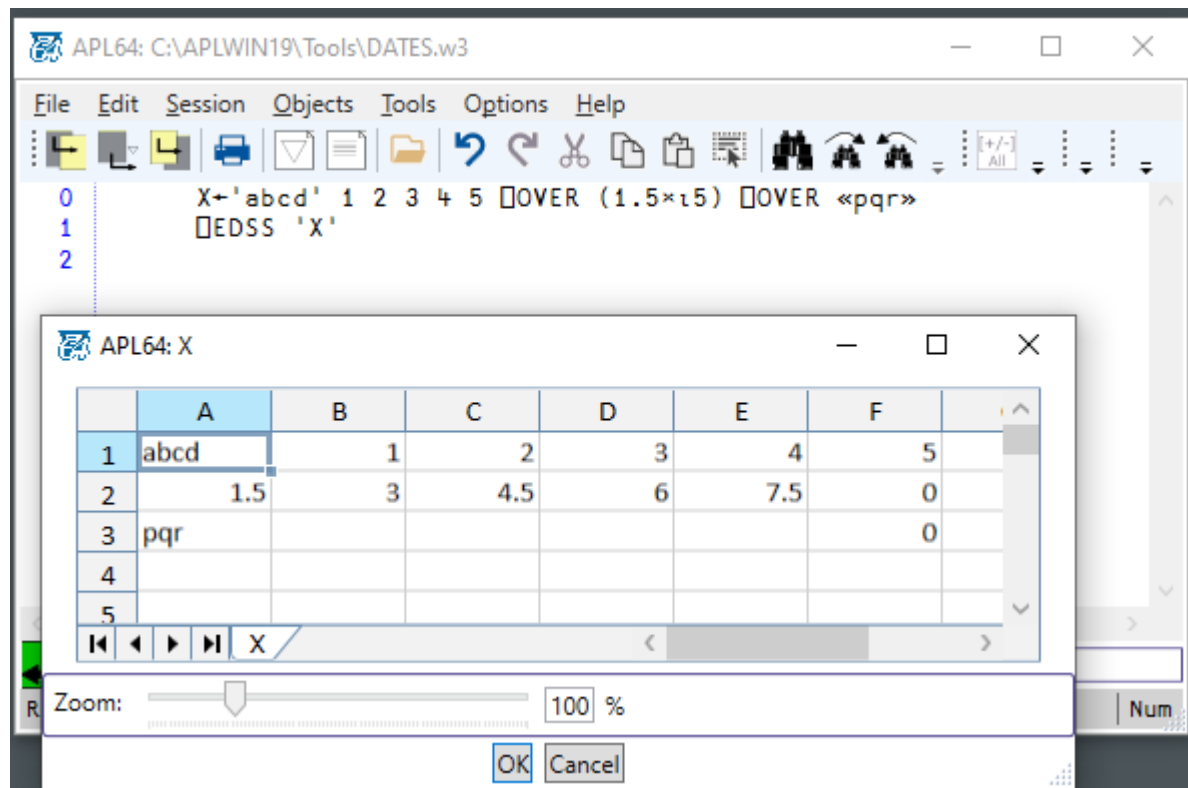
```

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.



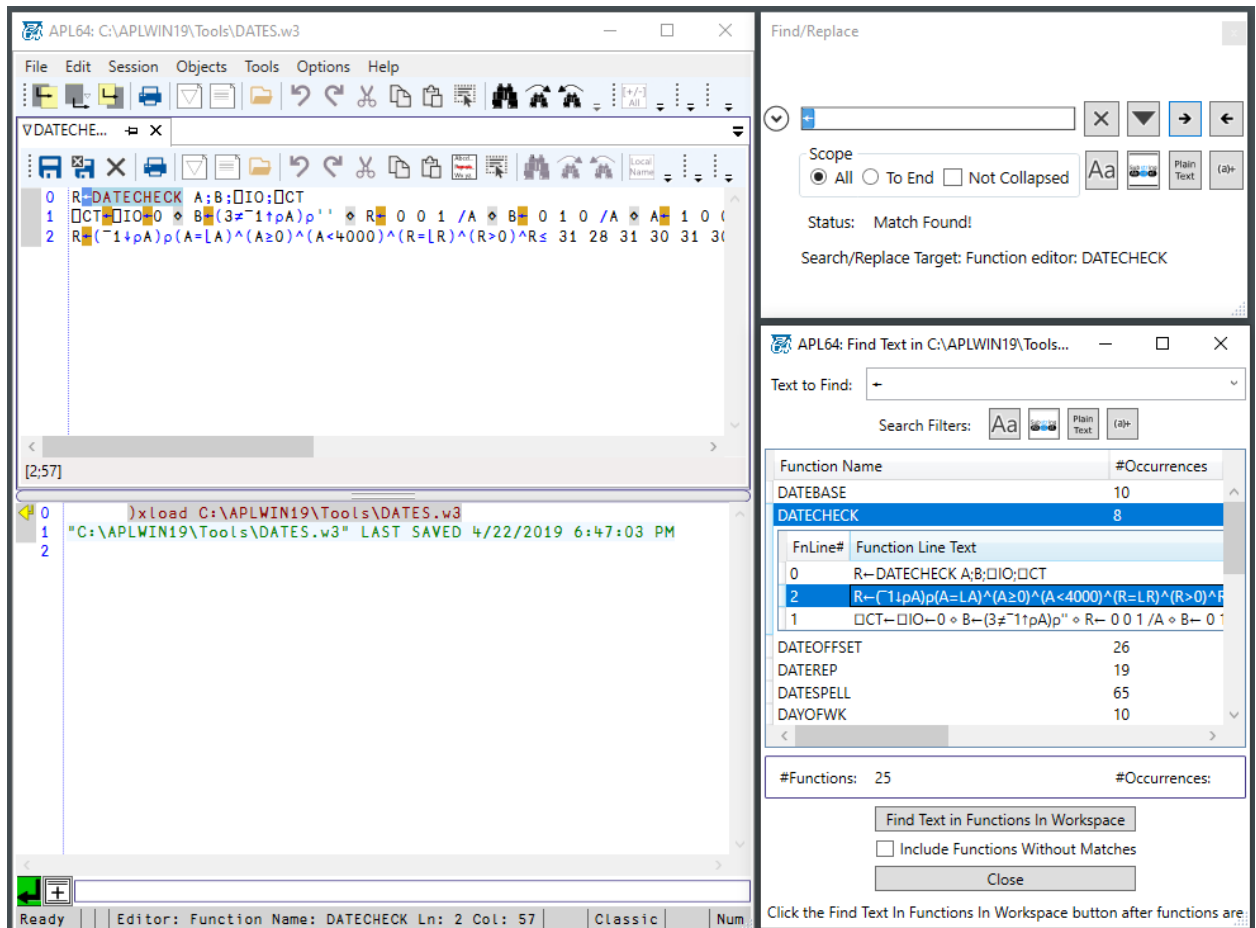
)EDSS Edit an APL Array in a Worksheet Editor

)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

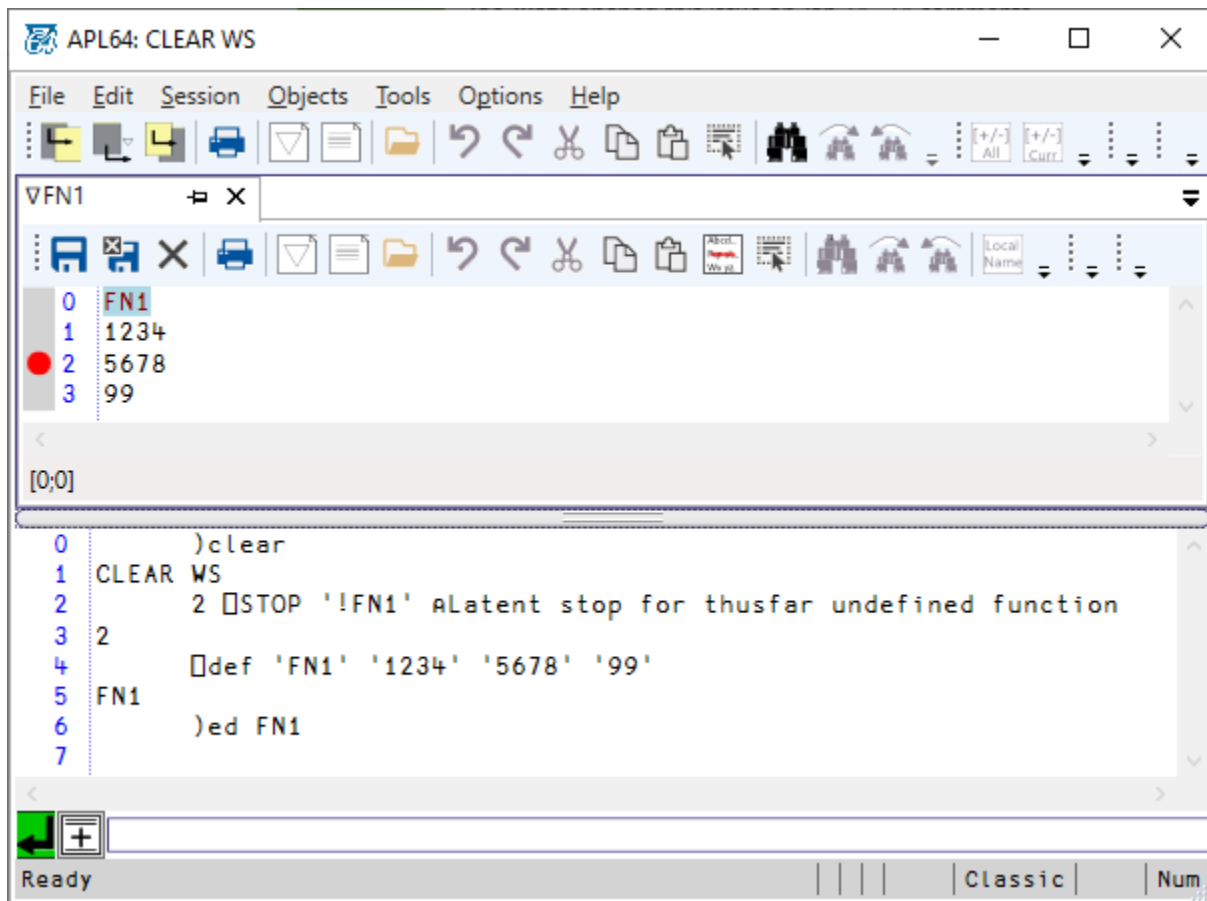
The Edit > Find in Functions in Workspace utility will identify the function lines of the functions in the workspace which match the developer-selected text. A double left click on a user-specified

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.

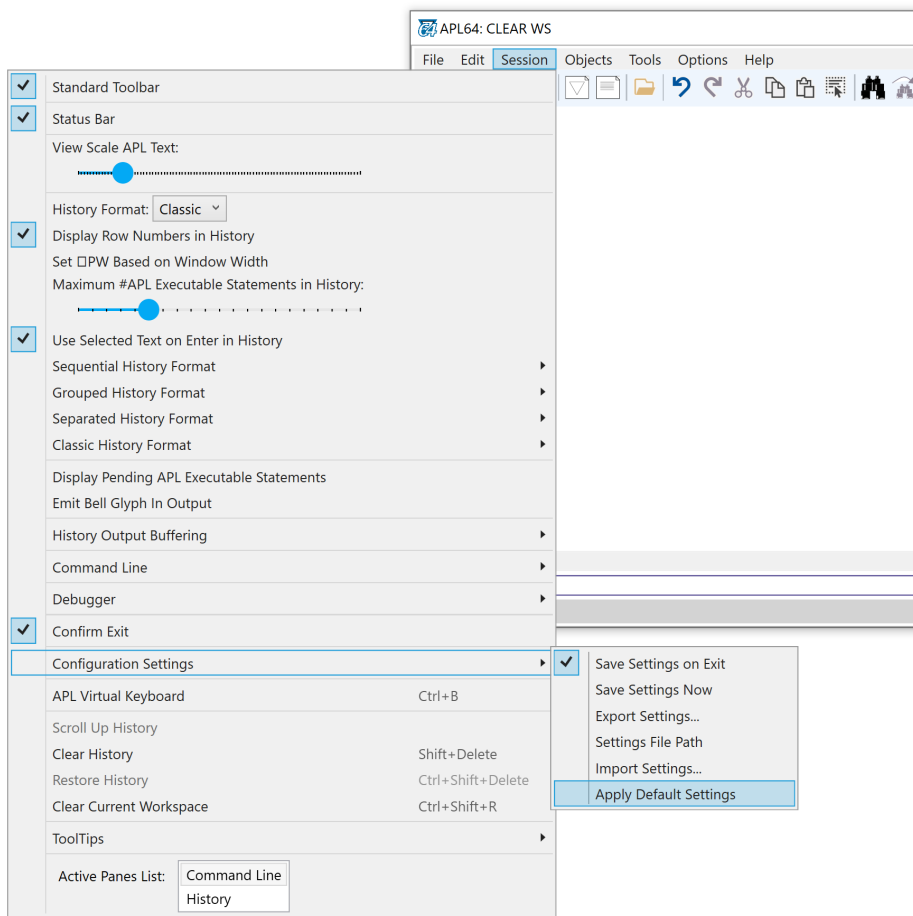


Fig: The menu location

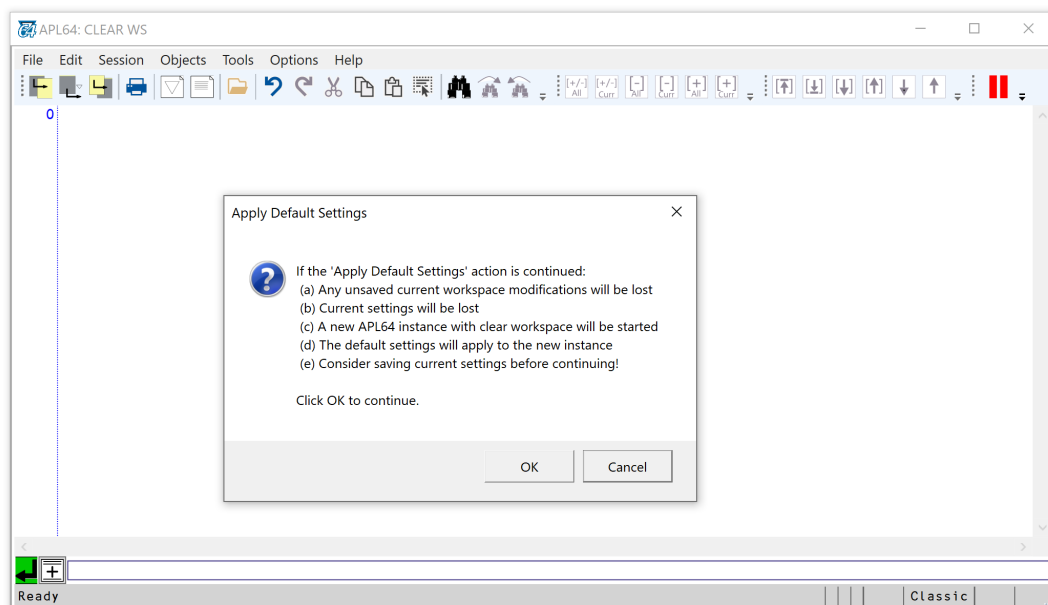


Fig: Warning message

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 '?'.

APL64: CLEAR WS

File Edit Session Objects Tools Options Help

0 XML '?'
 1 XML Documentation
 2 keyXAtt XML "AttCreate" xAttName xAttValue
 3 (eltName attValue) XML "AttGetVal" keyXAtt [parse(bool)]
 4 xAttXml(string) XML "AttGetXmlText" keyXAtt
 5 keyXData XML "CDataCreate" xCDataValue
 6 cDataValue XML "CDataGetVal" keyXCData
 7 xCDataXml(string) XML "CDataGetXmlText" keyXCData
 8 XML "CDataSetVal" keyXCData xCDataValue
 9 keyXCom XML "ComCreate" xComValue
 10 xComXml(string) XML "ComGetXmlText" keyXCom
 11 keyXDecl XML "DeclCreate" declVersion declEncoding declStandAlone
 12 xDeclXml(string) XML "DeclGetXmlText" keyXDecl
 13 XML "DictDiapObj" keyXObj[""/DiapAll]
 14 objInfo XML "DictObjInfo" keyXObj[""/All]
 15 keyXDoc XML "DocCreate" rootElementName
 16 keyXDecl XML "DocDeclGet" keyXDoc
 17 XML "DocDeclSet" keyXDoc keyXDecl
 18 xDocXml(string) XML "DocGetXmlText" keyXDoc
 19 XML "DocInsert" xKeyDoc xKeyElt xKeyObj InsertionOrder
 20 keyXDoc XML "DocLoad" xmlDocFilePath preserveWhiteSpace(bool)
 21 XML "DocRemove" keyXDoc keyXObj
 22 keyXRoot XML "DocRootGet" keyXDoc
 23 XML "DocSave" keyXDoc xDocFilePath disableFormatting(bool) omitDuplicateNamespaces(bool)
 24 XML "EltAdd" keyTgtXElT xKeyObj [location]
 25 keyXAtt XML "EltAttGet" keySrcXElT xAttName
 26 XML "EltAttRemove" keySrcXElT xAttName
 27 keyXElt XML "EltCreate" xEltName [namespaceInfo]
 28 keyXElt XML "EltGet" keyTgtXElT xEltTgtName
 29 keyXAtt XML "EltGetAttByIndex" keySrcXElT attIndex
 30 keyXElt XML "EltGetEltByIndex" keySrcXElT eltIndex
 31 (eltName eltValue) XML "EltGetVal" keyXElt [parse(bool)]
 32 xEltXml(string) XML "EltGetXmlText" keyXElt
 33 Int32 XML "EltNAtts" keyXElt
 34 Int32 XML "EltNElts" keyXElt
 35 bool XML "EltHasAtts" keyXElt
 36 bool XML "EltHasElts" keyXElt
 37 bool XML "EltIsEmpty" keyXElt
 38 keyXElt XML "EltLoad" xmlElementFilePath preserveWhiteSpace(bool)
 39 keyXElt XML "EltParse" xmlText preserveWhiteSpace(bool)
 40 XML "EltRemoveAll" keyXElt
 41 XML "EltRemoveAtts" keyXElt
 42 XML "EltSave" keyXElt xDocFilePath disableFormatting(bool) omitDuplicateNamespaces(bool)
 43 XML "EltSetAttVal" keyTgtXElT attName [attValue]
 44 XML "EltSetEltVal" keyTgtXElT childEltName [childEltVal]
 45 XML "EltSetVal" keyXElt eltVal
 46 xkeyXNs XML "NsCreate" nsText
 47 nsText(string) XML "NsGetXmlText" keyXNs
 48 keyXElt XML "XPathSelectElt" keySrcXElT xPathText
 49 Char[:] XML "HELP"
 50 Char[:] XML "?"
 51 Notes:
 52 attValue: Text
 53 attValue: Int32(scalar)
 54 attValue: Double(scalar)
 55 attValue: Bool(scalar)
 56 attValue: date(UTS-format, assumed UTC)
 57 key... : key to Xml object in the Xml objects dictionary
 58

Ready | Hist: Ln: 6 Col: 25 | Classic | Num EN_US

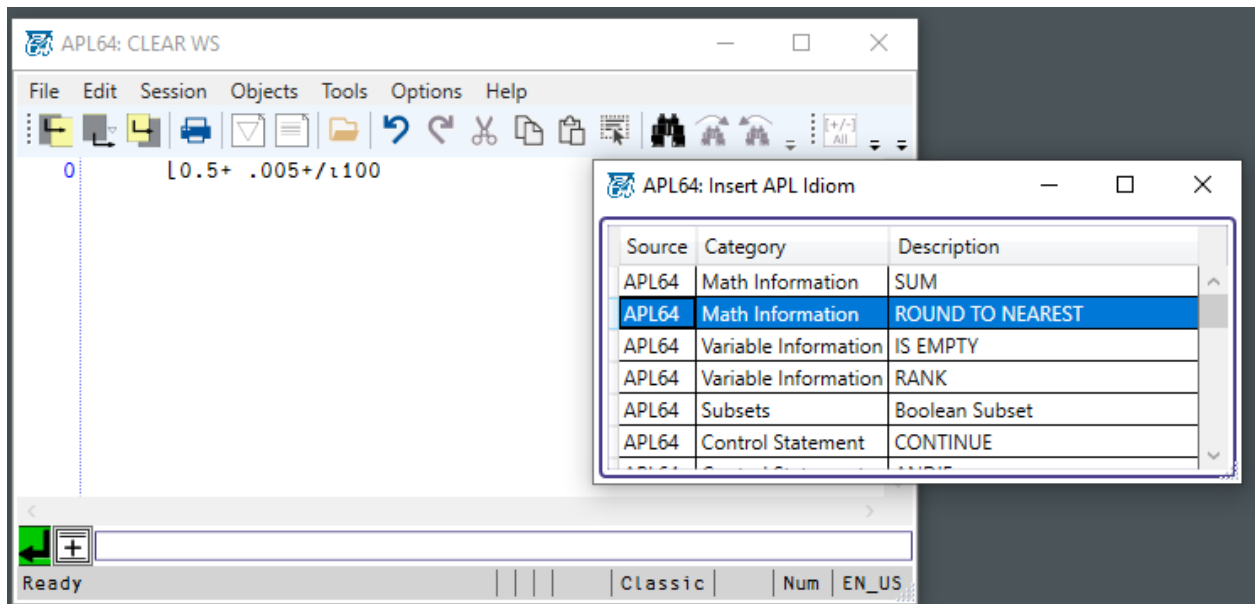
APL64 Idioms Manager

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:

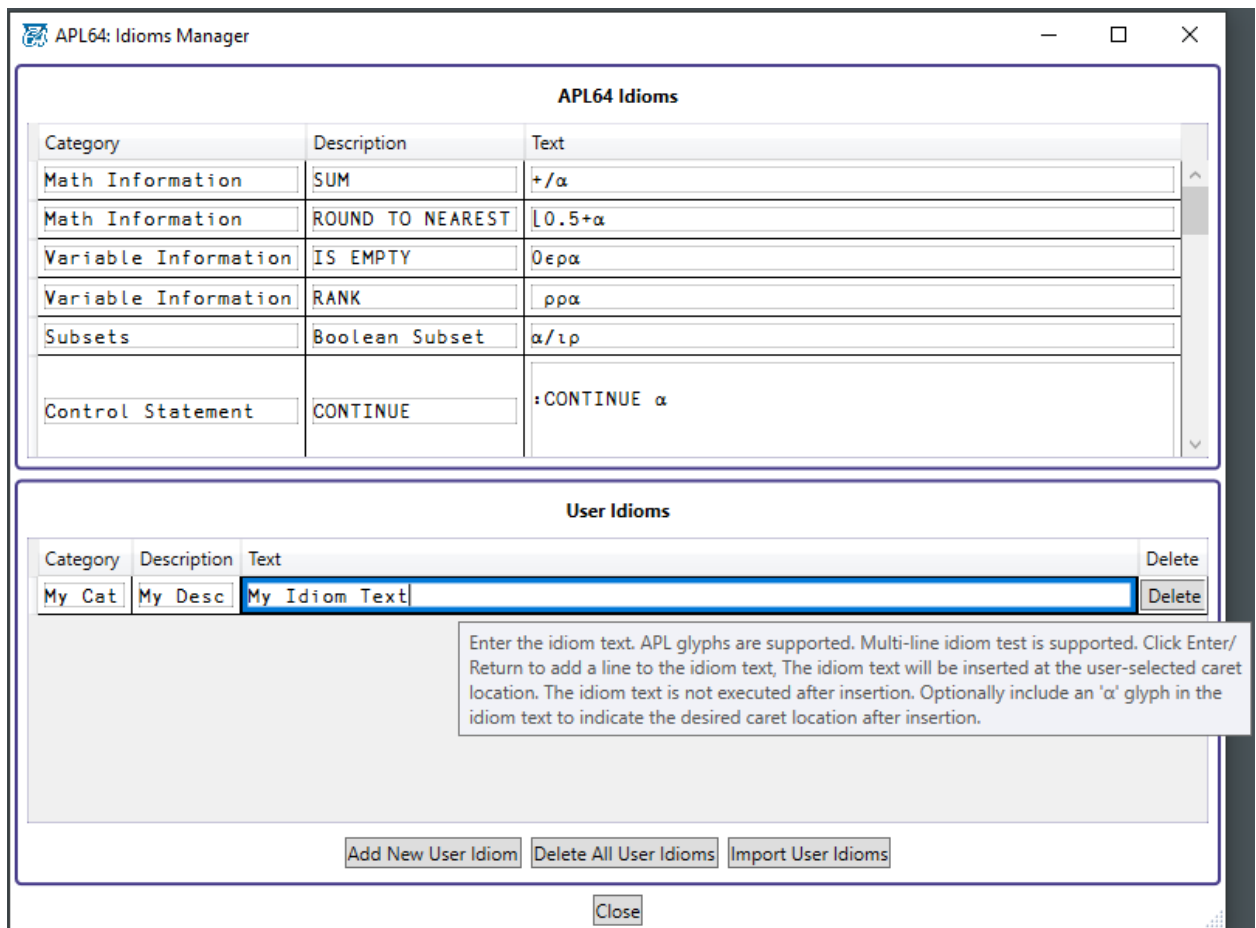
- Edit > APL Idioms > Insert An APL Idiom...(Ctrl+I)



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.



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

☐ LJUST, ☐ 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

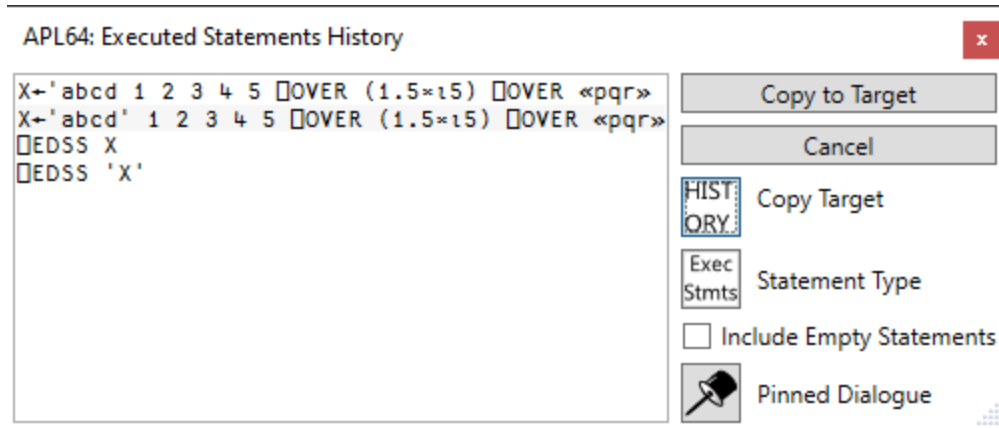
☐XL

☐XML

Enhanced Dialogs

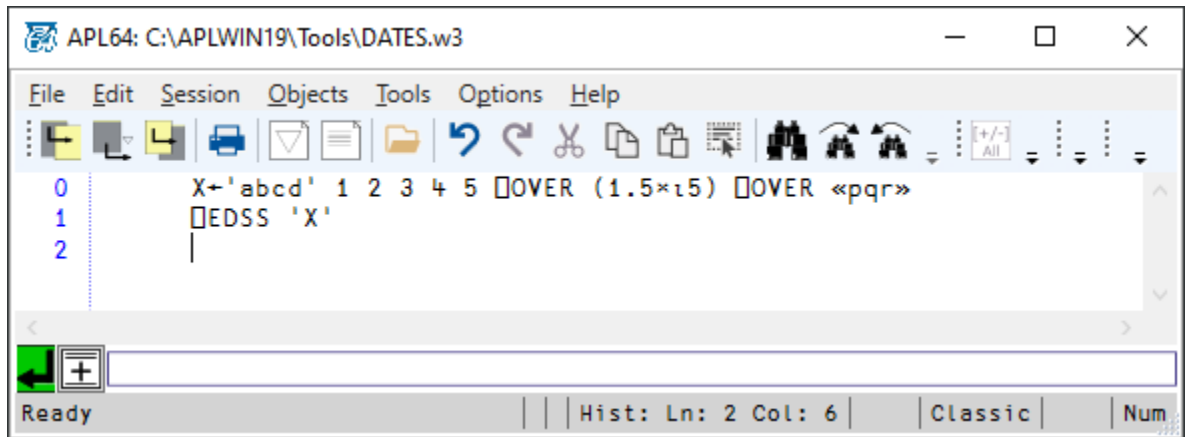
APL Statements History – More options than Gather action

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.



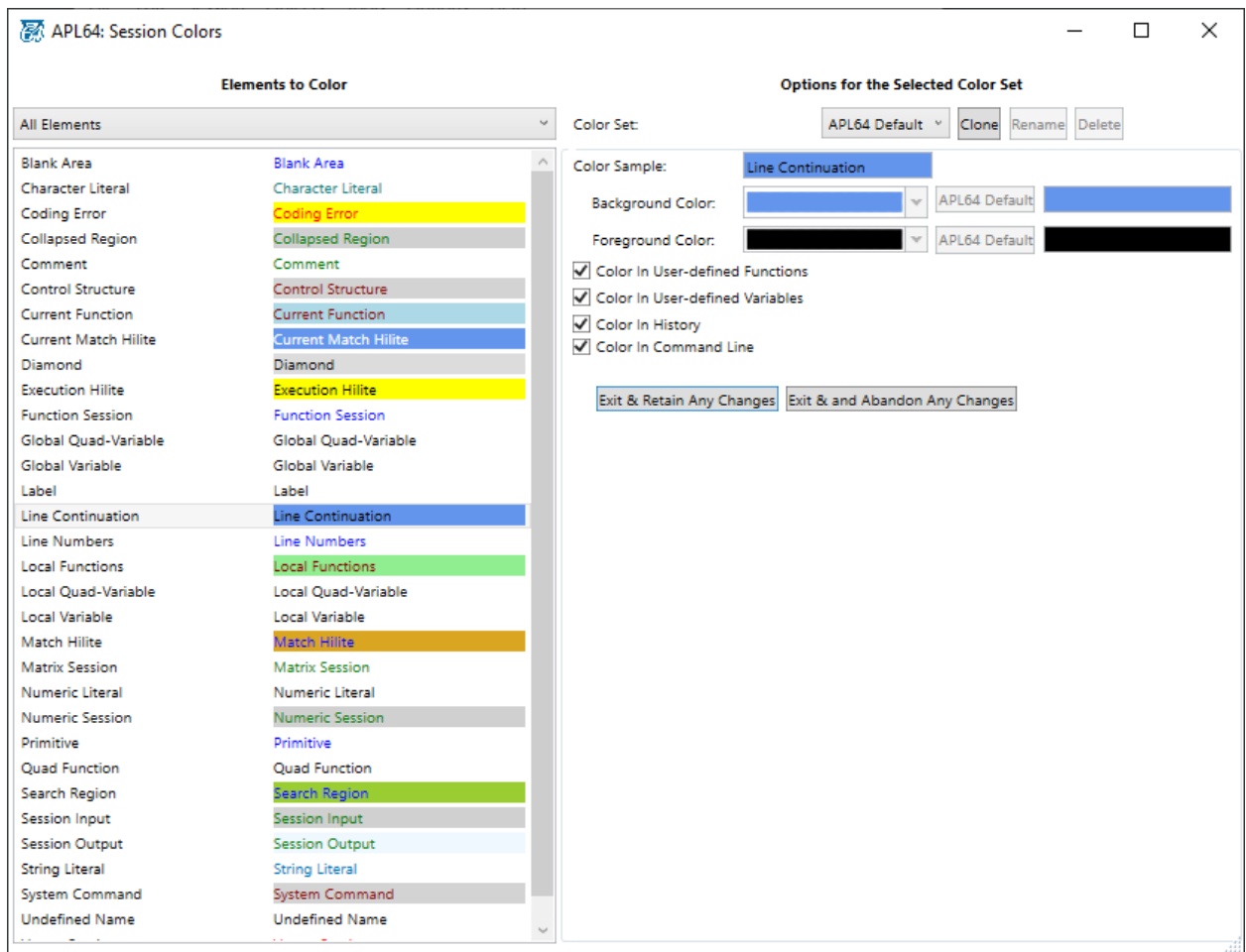
History Pane – Optional Row Numbers

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



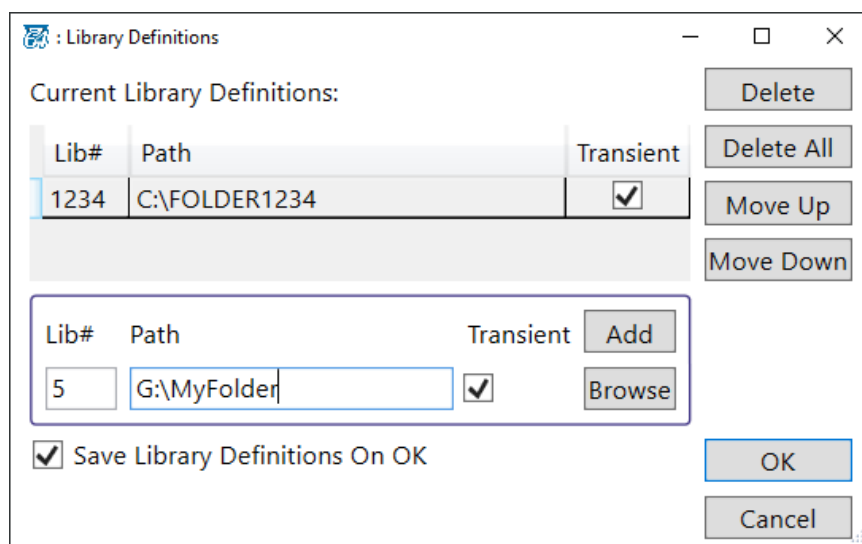
Colors Dialog

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: 20000

☒ Overwrite Log File

☐ Issue Alert when %Threshold is Exceeded

%Threshold for Alert: 90

Log File

Log File Prefix: AplSessionLog

Log File Path: C:\Users\Joe Blaze\AppData\Local

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

APL64: Keyboard Definition

Click on a Key Position to View Its Definition

Definition of the Selected Key Position ☐

Selected Row: 1 Selected Column: 6 Key Name: D5

Key Position Applies to this Keyboard Definition ☒

Modifier Key	Glyph	Unicode
None	5	53
Shift	%	37
Alt	= equal	61
Alt+Shift	ϕ column rotate	9021

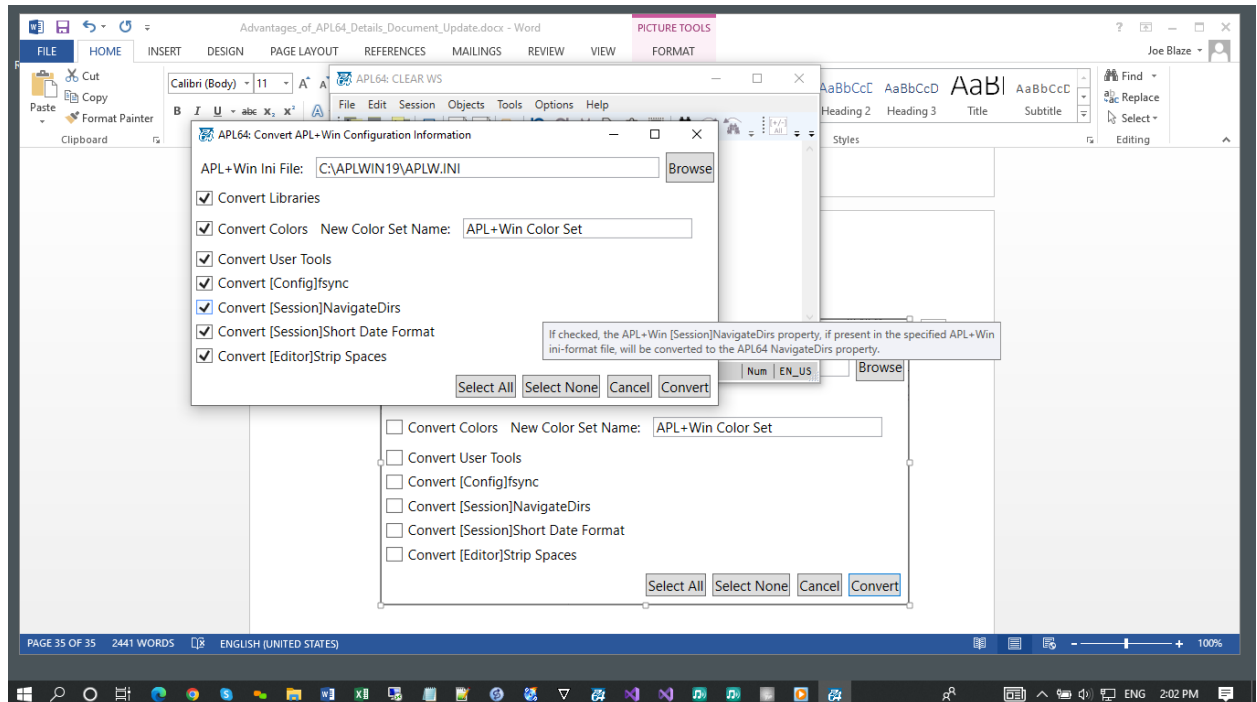
Select Keyboard Definition: EN-US

Button positions unavailable for some keyboard definitions are indicated by ☐

AltGr glyphs are controlled by the installed Windows keyboard layout. The rendering of decimal Unicode points to glyphs is controlled by the APL385 font.

OK Cancel

Convert APL+Win Configuration File



APL64 Now Supports .NET Release 6.0 (x64)

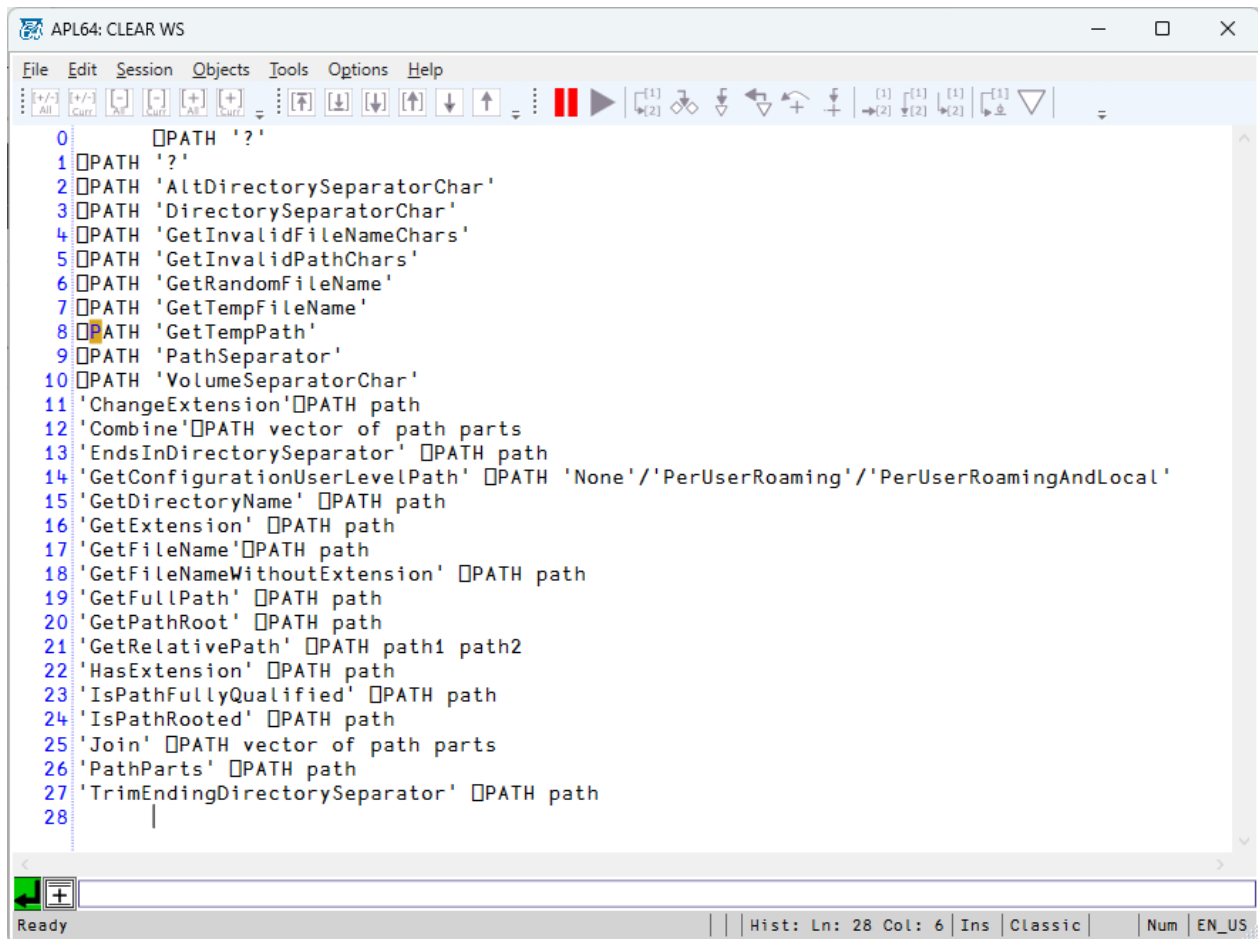
This second production release of APL64, 2022.0.2, requires the installation of the Microsoft .NET SDK 6.0.402. The installer for this version of APL64 will prompt you to download and install it.

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.



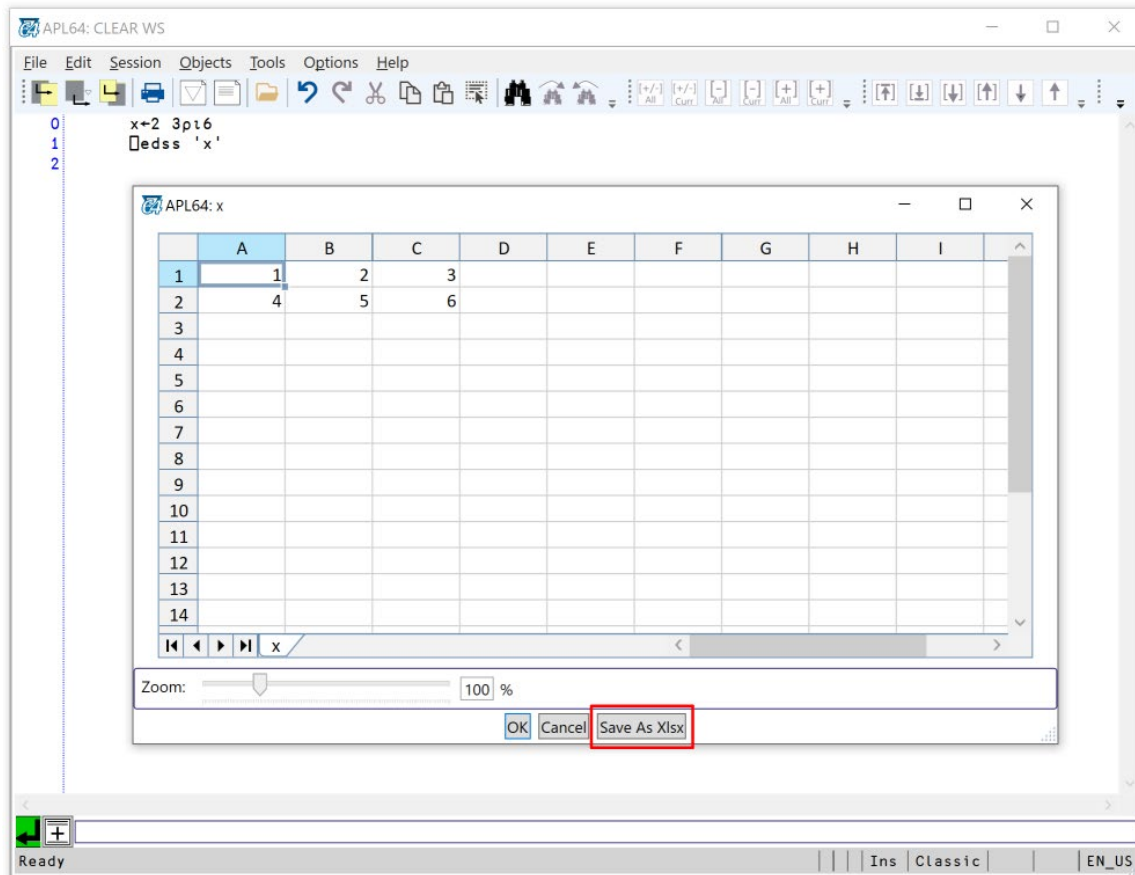
```
0  □PATH '?'
1  □PATH '?'
2  □PATH 'AltDirectorySeparatorChar'
3  □PATH 'DirectorySeparatorChar'
4  □PATH 'GetInvalidFileNameChars'
5  □PATH 'GetInvalidPathChars'
6  □PATH 'GetRandomFileName'
7  □PATH 'GetTempFileName'
8  □PATH 'GetTempPath'
9  □PATH 'PathSeparator'
10 □PATH 'VolumeSeparatorChar'
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
```

New ☐Pathcase System Variable

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

New Save As Xlxs Option in)EDSS/☐EDSS

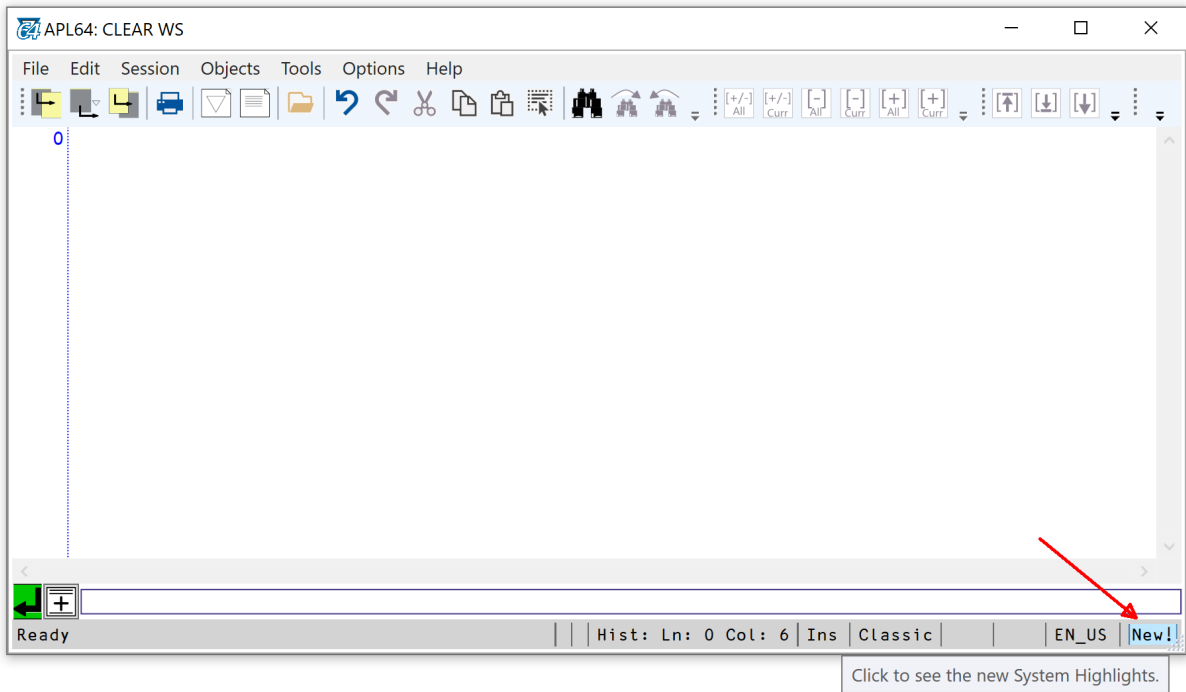
This option can be used to save the edit session information to a Microsoft Excel Open XML Spreadsheet (XLSX) file.



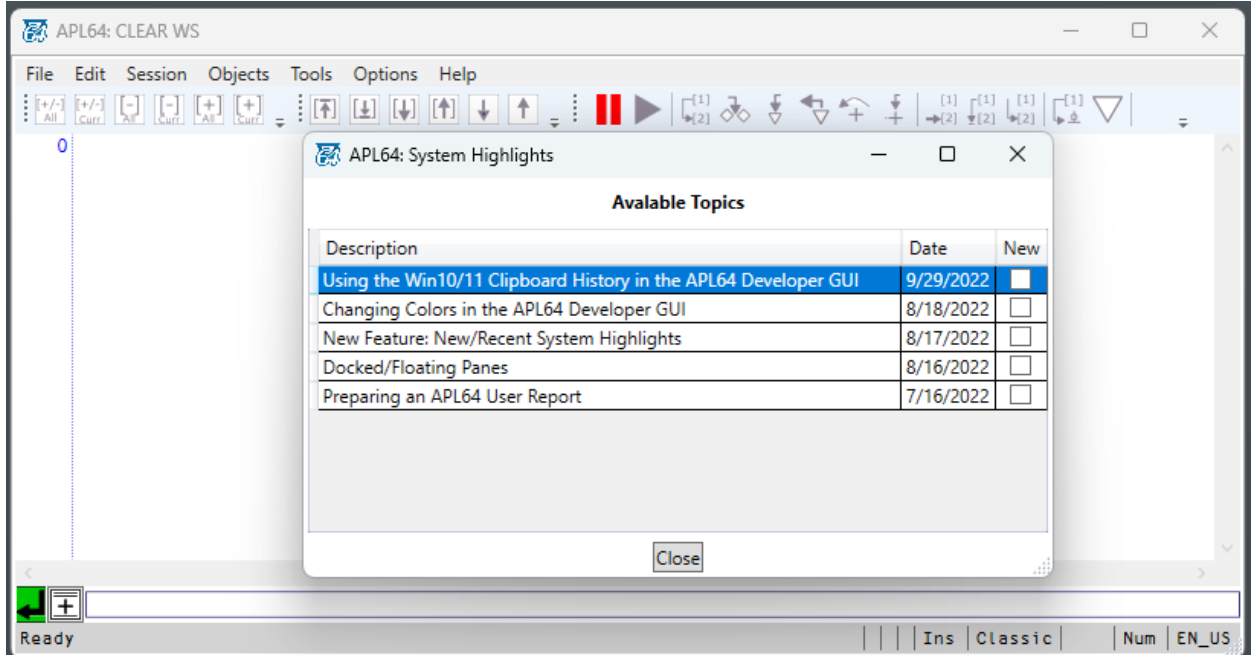
New APL64 System Highlights Feature in the Status Bar

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



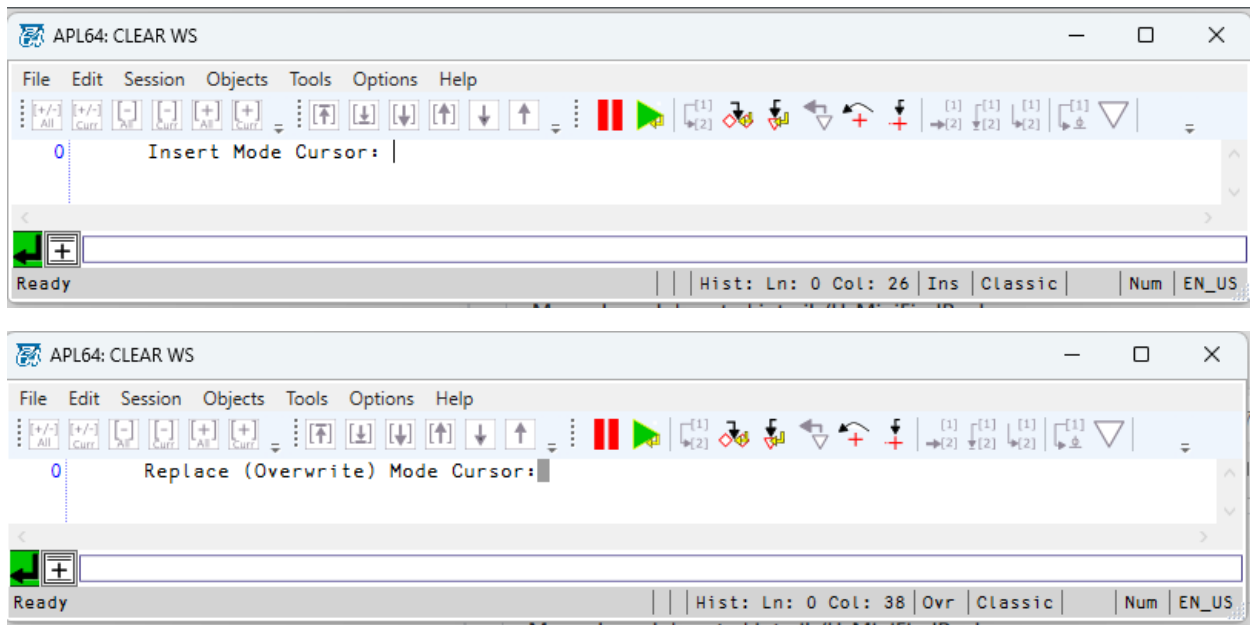
Clicking the notification area in the status bar or the **Help | System Highlights | Recent System Highlights** menu displays the **Available Topics** window shown below:



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

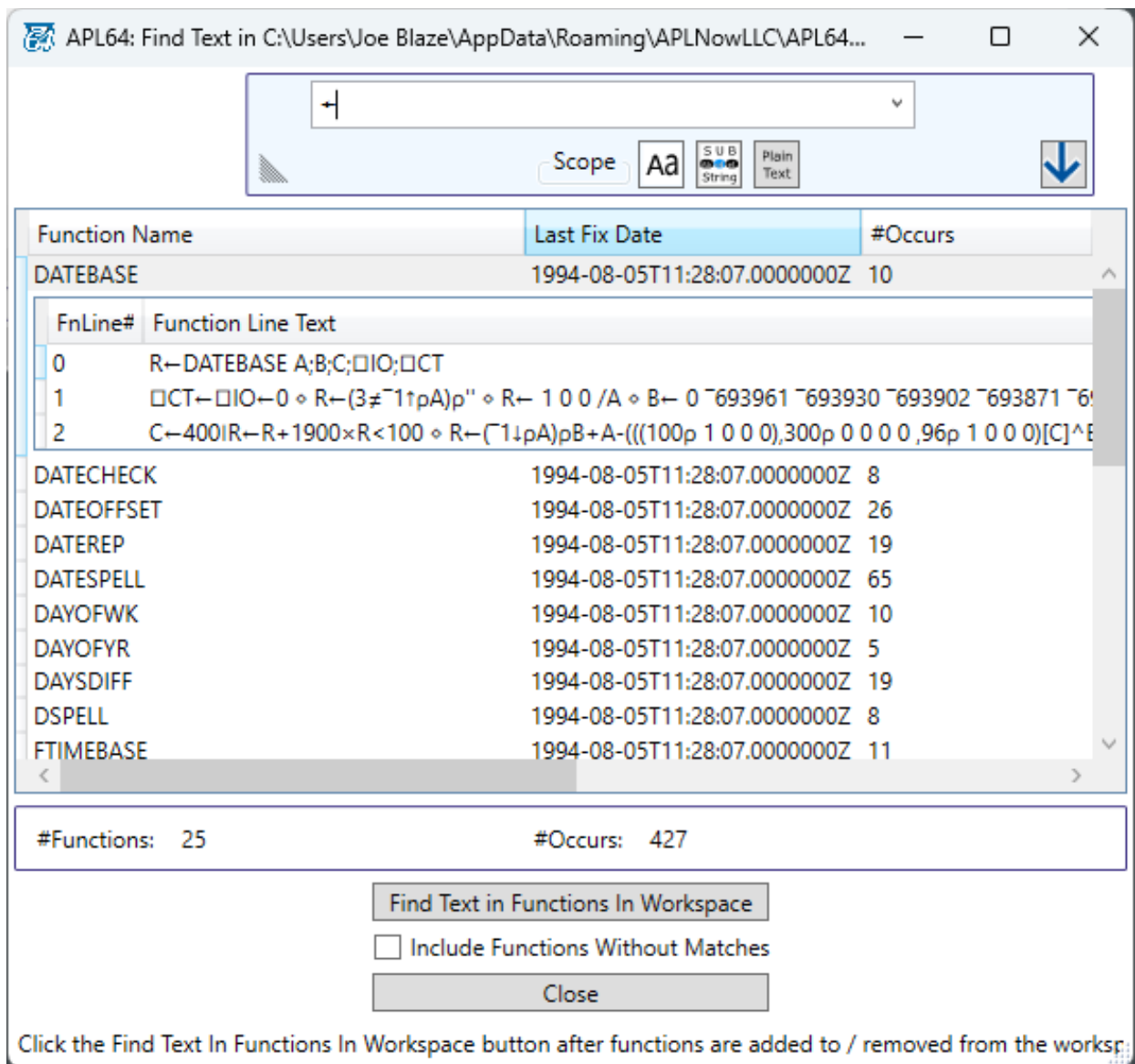
Pressing the Insert key on the keyboard toggles between Insert and Replace mode. Each mode has a distinctive caret; thin in Insert mode and thick in Replace (Overwrite) mode. The status line displays "Ins" when APL64 is in Insert mode and "Ovr" when in Replace mode.



Find in Functions in Workspace Utility

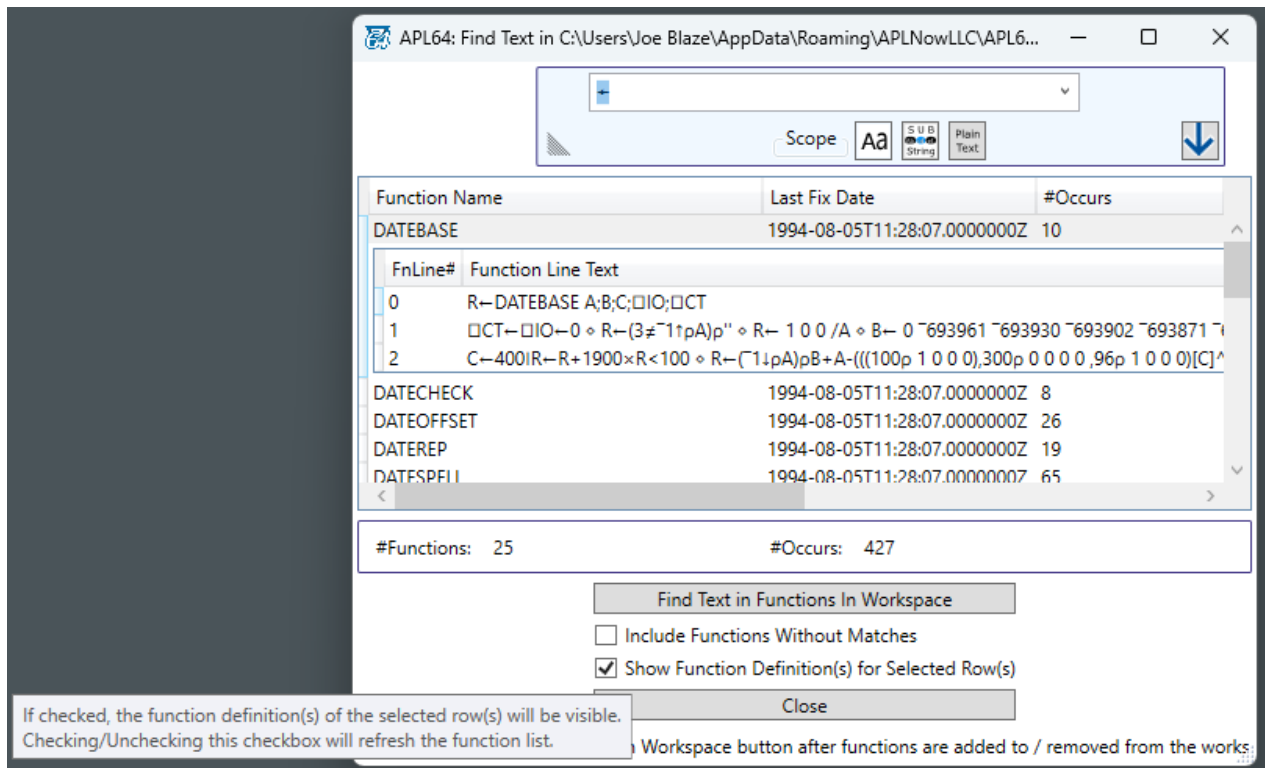
'Last Fix Date' Column Added to the Filtered List of Functions

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:



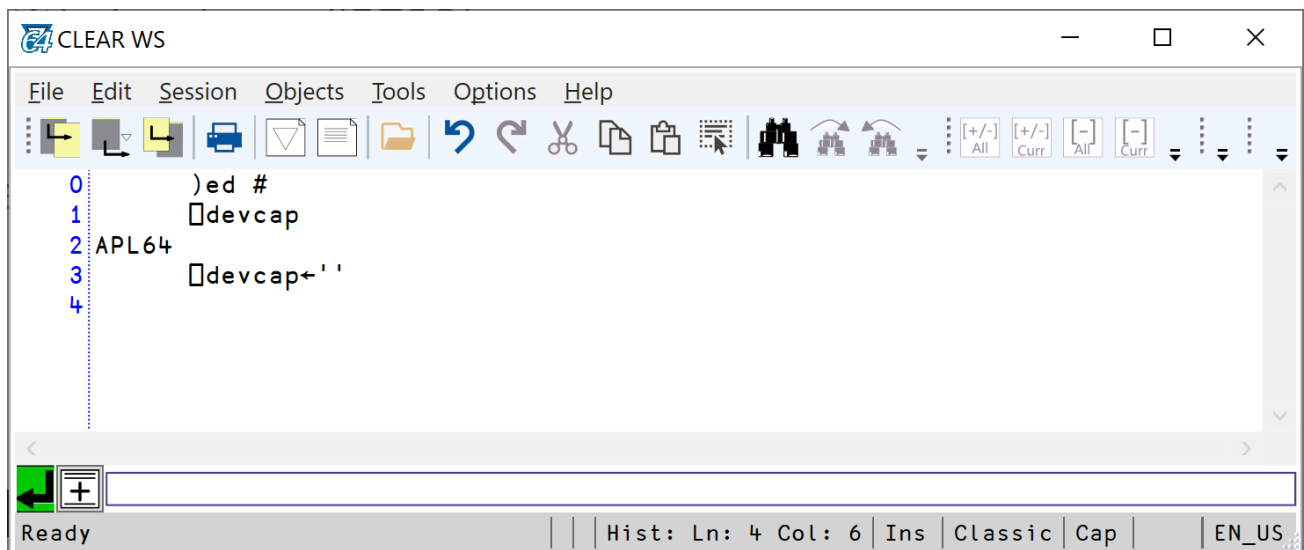
New Option is Provided to "Show Function Definition(s) for Selected Row(s)" in the Function List

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



□devcap Set to an Empty Vector

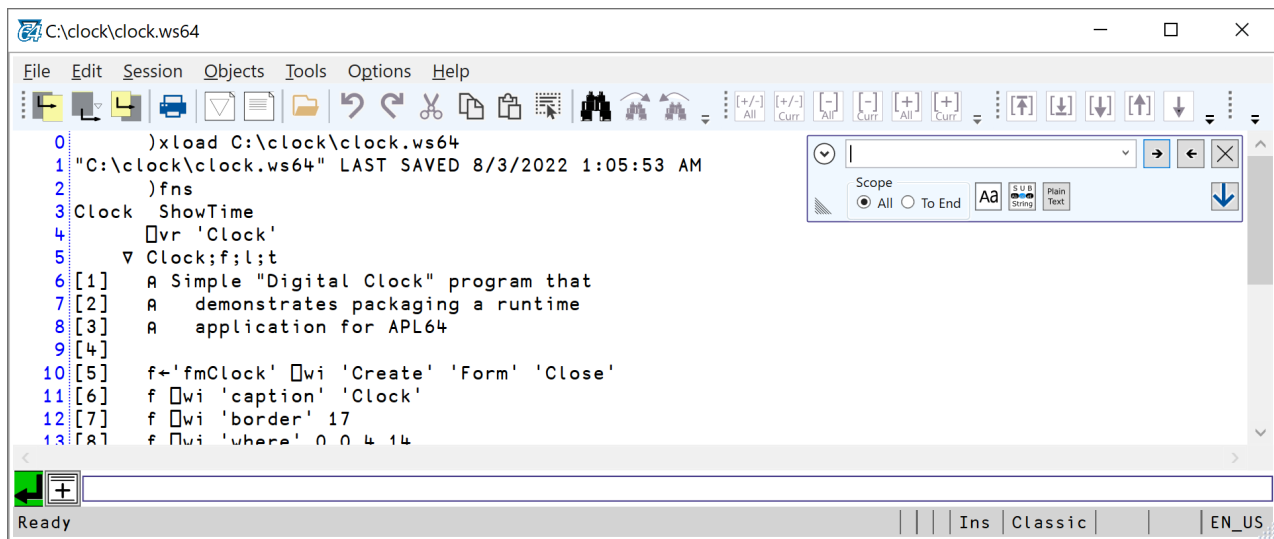
Assigning □devcap to an empty vector now displays just the name of the workspace in the APL64 session title bar.



The 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.

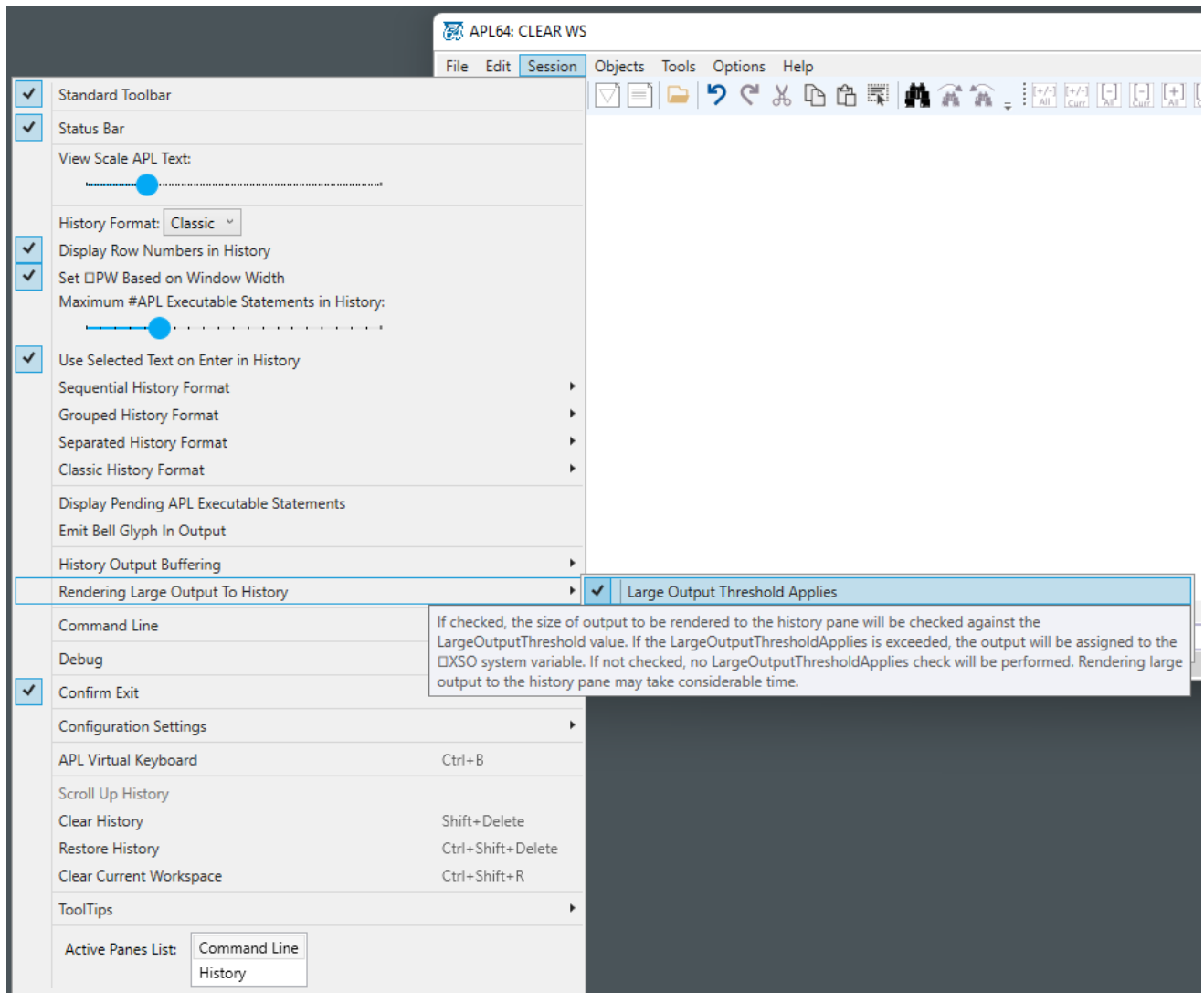
For example, in the editable classic history:



Rendering Large Output to History and □xso System Variable

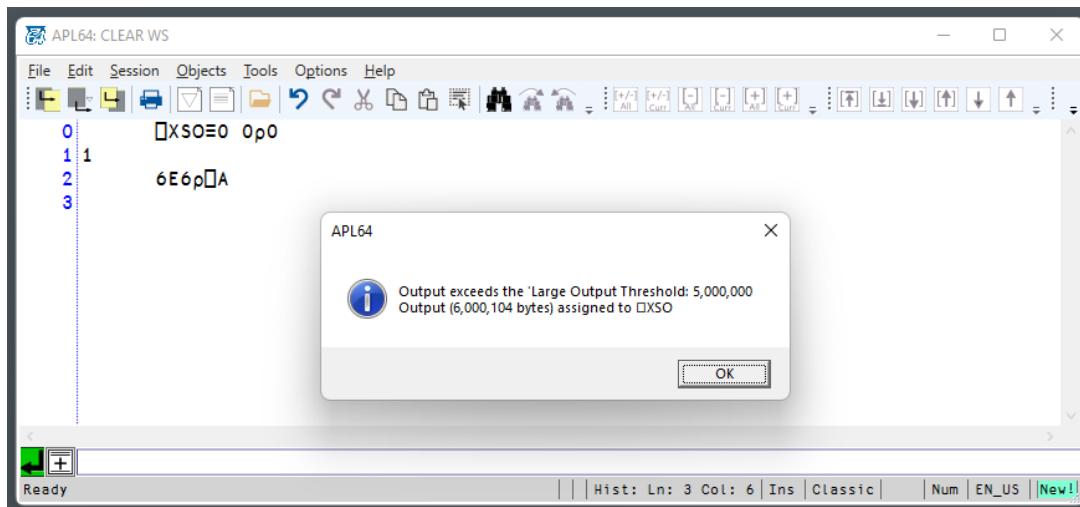
Very large variable sizes are possible in APL64 but formatting the values and rendering these values to the history pane may not be useful. A more practical approach is to assign the values to an APL64 variable and use the numerous APL features to reduce, format and render the data to the history pane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



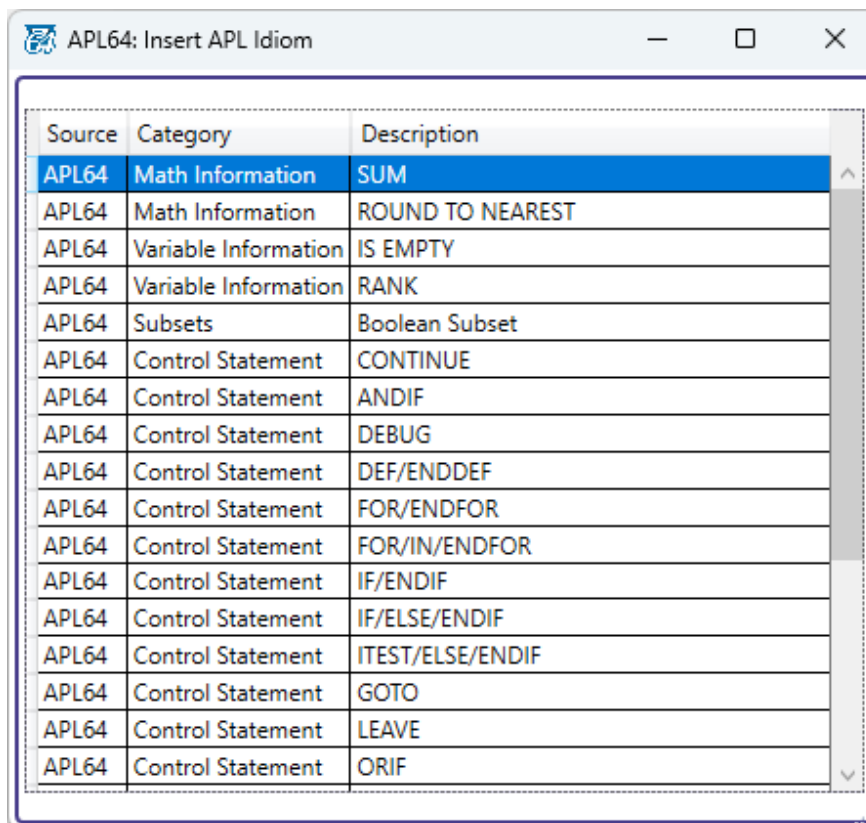
When the **Session | Rendering Large Output To History | Large Output Threshold Applies** is checked, the **Session | Rendering Large Output To History | Large Output Threshold** in units of 1E6 bytes will be used. If the size of the output to be formatted and rendered to the history pane exceeds the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



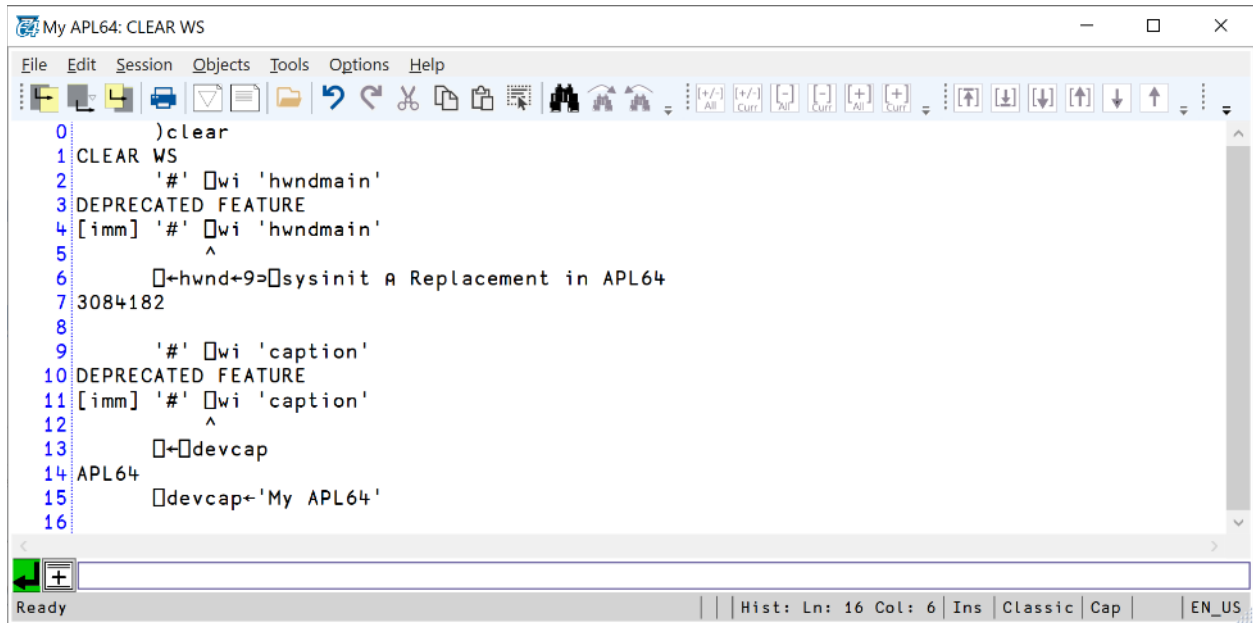
Insert APL Idiom Dialog is Now Not Modal

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 $9\Rightarrow$ ☐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.

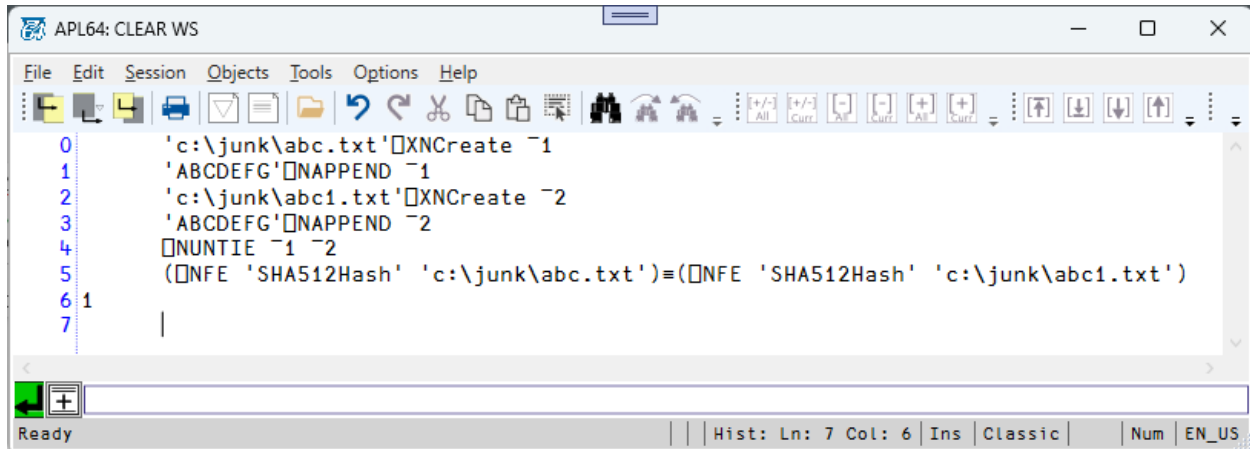
The ☐wi Interface to the APLNow32.exe was Optimized to Provide Better Stability.

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

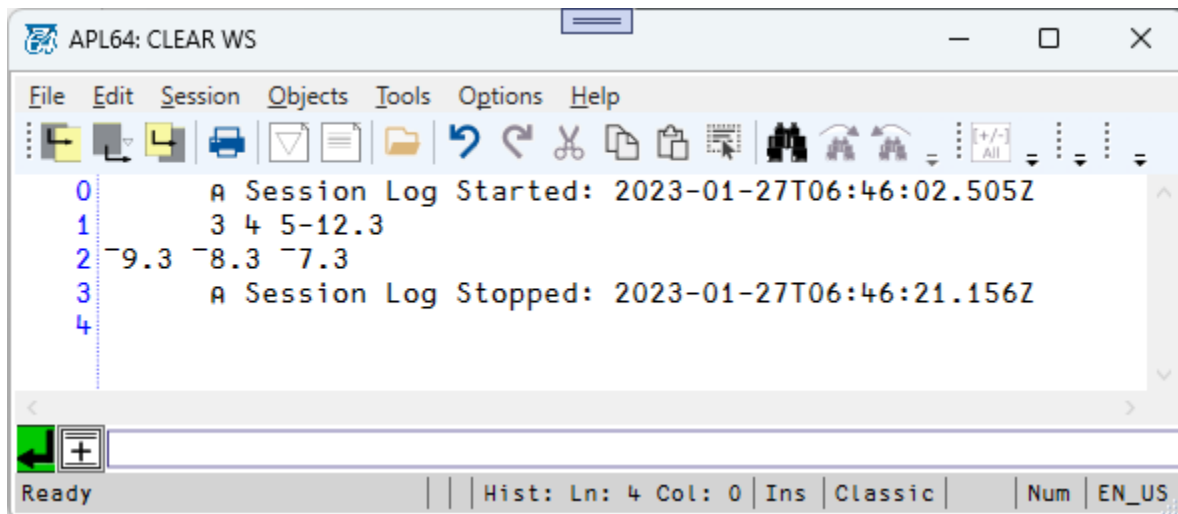
Purpose: To compare two file contents. Refer to [Help](#) | [APL Language](#) | [Using !\[\]\(77ac8013b2d3d61bfea37e205e2e87c2_img.jpg\) NFE](#) for the details.



```
0 'c:\junk\abc.txt'⎕XNCreate ~1
1 'ABCDEFG'⎕NAPPEND ~1
2 'c:\junk\abc1.txt'⎕XNCreate ~2
3 'ABCDEFG'⎕NAPPEND ~2
4 ⎕NUNTIE ~1 ~2
5 (⎕NFE 'SHA512Hash' 'c:\junk\abc.txt')=(⎕NFE 'SHA512Hash' 'c:\junk\abc1.txt')
6 1
7 |
```

Session Log Timestamps Displayed

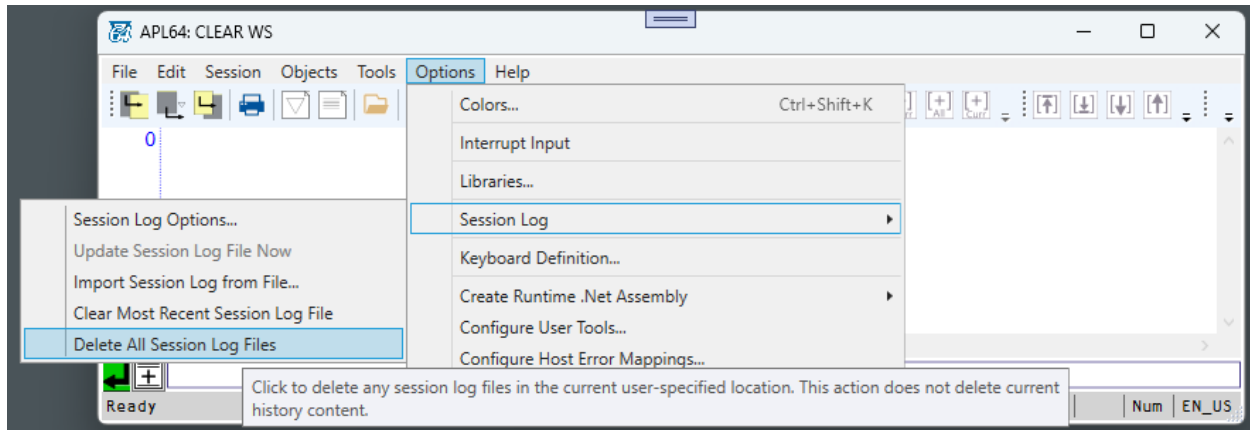
As in APL+Win, the session log start and end timestamps will display in the History log when the session log is enabled.



```
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
```

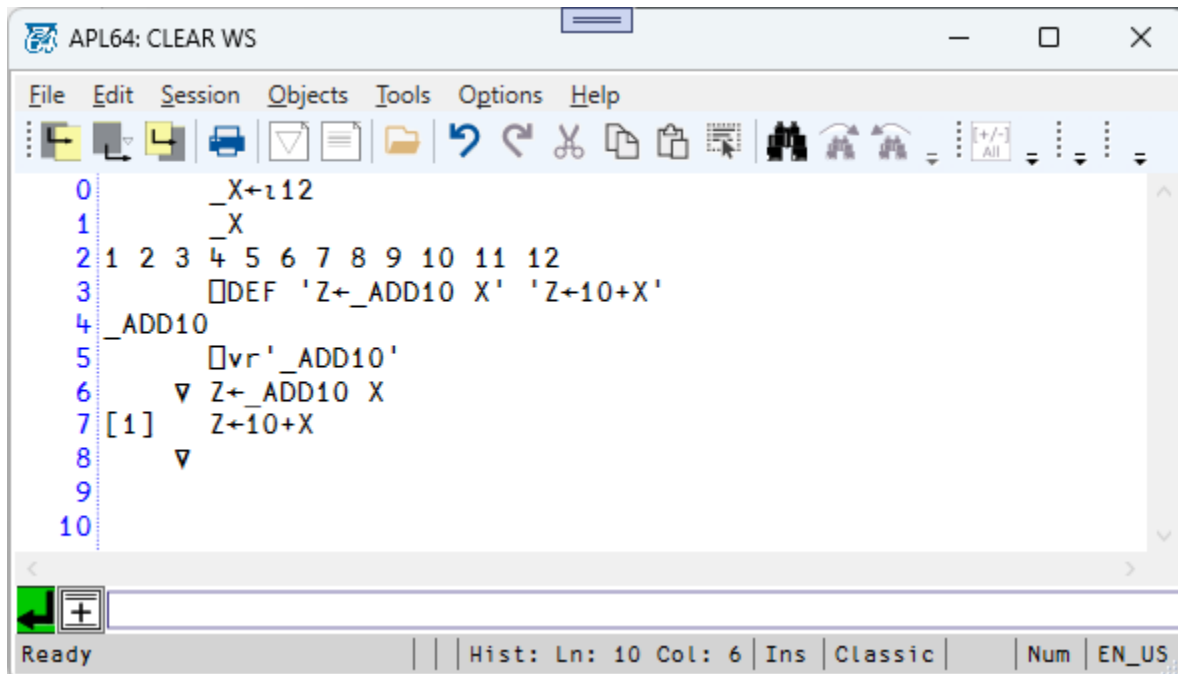
New Options | Session Log | Delete All Session Log Files Action

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.

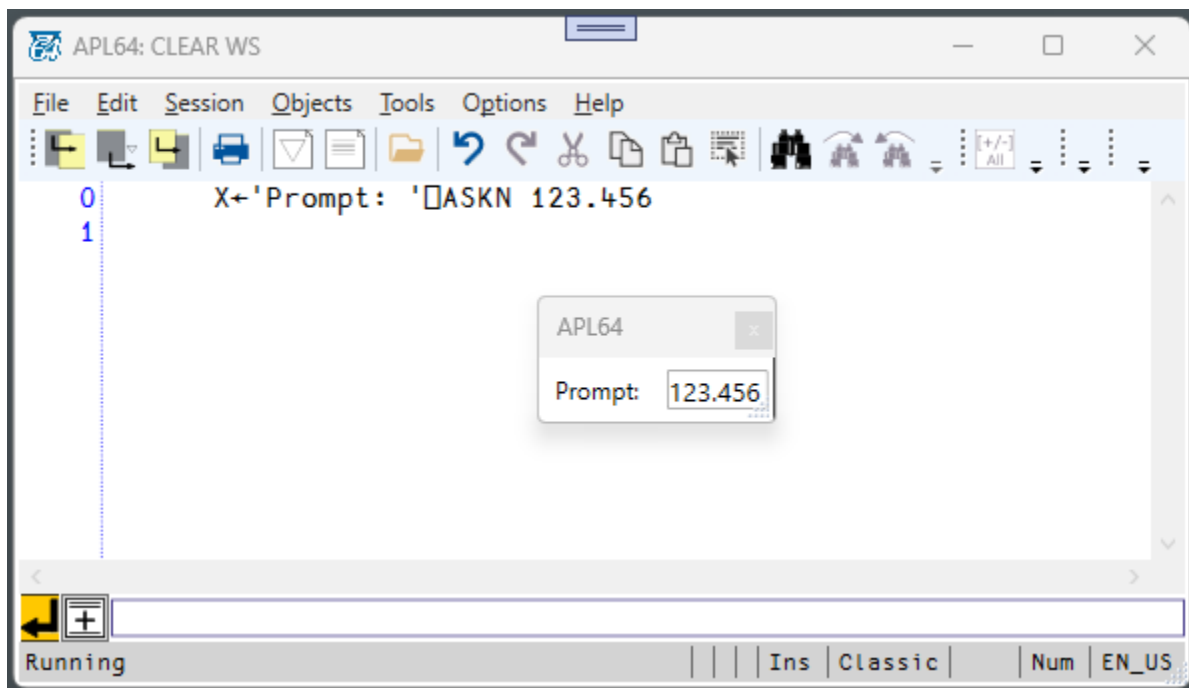
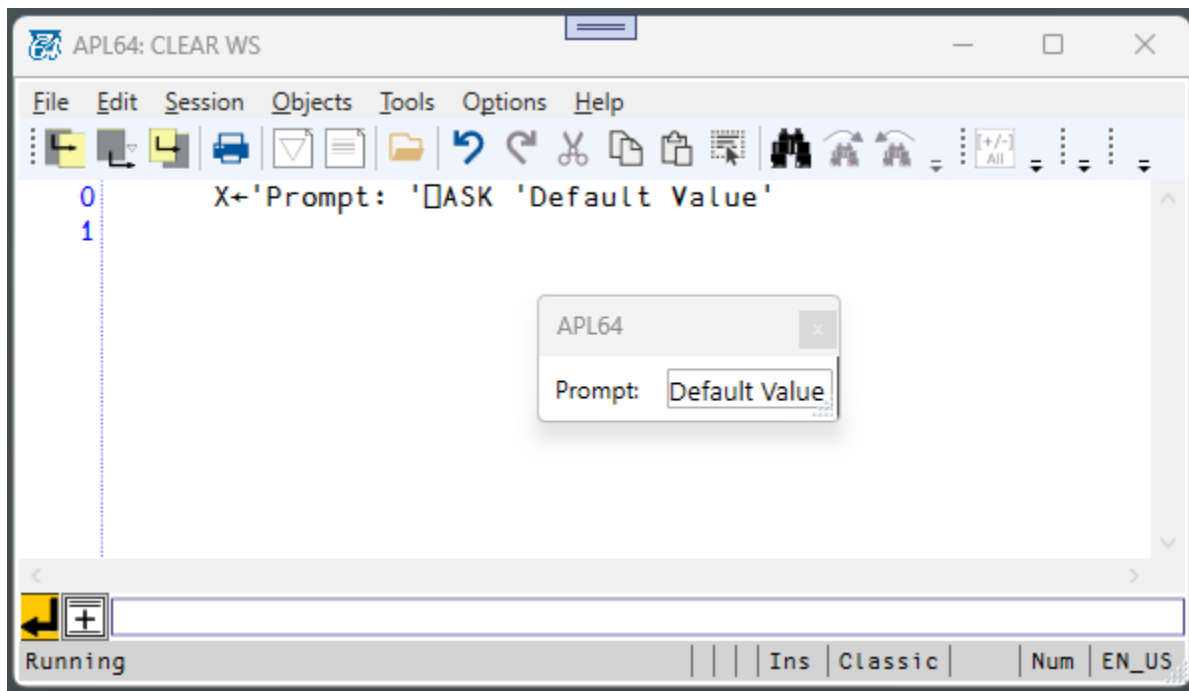


New System Functions: ☐flock, ☐fflush, ☐nlock, ☐nflush

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](#) the 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: ToFileXml and FromFileXml
- 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