

Modifications Included in APL64 Pre-Production Version (2020.0.2)

Table of Contents

Overview	7
Enhancements	7
Documentation Window Title	7
Open Object With Right Mouse Double Click	7
Clear/Restore Session History Short Cuts	8
Variable Editor Row Numbers	8
Options > Colors Dialogue	9
Renaming Cloned Color Set	9
Color for Blank Area	9
Color Picker Window	9
☐ PW Based on Window Width	9
Options > Configure User Tools	10
New Configure Host Error Mappings	11
APL+Win Configuration Conversion	11
Find/Replace dialog adds Undo/Redo/Undo all/Redo all and excluding collapsed regions	13
Bug Fixes	15
Rotate Primitive Function	15
:CATCHIF	15
☐ sysinit	15
Scientific Notation	15
Shared Network Drive Paths	15
☐ telprint	15
Switch Vertical Scroll Bar & Row Headers	15
☐ XLIB	15
Decode Primitive Function	15
The Decode primitive function example below produced an incorrect result	15
Concatenation	16
Sizing of Text	16
`2*31 reports LIMIT ERROR	16
)sinl system command not working	16

Evaluators Comments & APLNow Responses	16
When to Port Applications to APL64	16
Custom Installation Options	16
APL64 Installation impact on APL+Win.....	17
Installing APL64 on the Same Machine as APL+Win.....	17
APL64 Installation Target Drive.....	17
☐ new in APL64.....	17
Class inheritance in APL64	17
Class Definition in APL64.....	18
APL64 Documentation Availability.....	18
Preview documentation before installing APL64 software?	18
User Commands in APL64	18
User Command]DISPLAY	18
Output Display in APL64	18
Documentation Windows Navigation Toolbar	19
Assertion Failure	20
Invariant Process Handle	21
Variable Editor Line Numbers	21
Variable Editor Toolbars	22
Variable Editor: Non-homogeneous Element Types.....	22
CSE Exception.....	22
Developing a Cross-platform Application GUI	22
Context Menus in the APL64 Session.....	24
)COPY System Command	24
Persistent File Filter	25
☐ wslib	25
Double Right Click	25
Webview2 Documentation Browser	26
)LIBD System Command.....	26
ASMFNS.....	26
Rotate Primitive	27
APLGrid in APL64.....	27
APLDraw in APL64.....	27

ODBC Database Access in APL64	27
☐ WI Accessing 64-bit DLLs.....	28
Session Command Line	28
Right Mouse Double Click	28
Decode Primitive Function.....	28
Scientific Notation.....	29
When APL64 will be available?	29
What will happen to APL+Win?	29
Library Numbers	29
Showing Tooltips.....	29
Switch Position of Vertical Scroll Bar & Row Headers in Results Grid: Possible exception and behavior issues.....	30
Clear Session History Shortcut	30
Library Definition Dialogue	30
List of System Functions.....	30
Class Definition	30
MessageBox System Function	31
How to turn off Tooltips in the session?	32
Tooltip Location	32
Scroll Up Session History.....	32
Exception: given key 'e' was not present	32
Alternative to Session Commoand Line	33
)COPY System Function Pattern Exception	33
Storage of Scalars.....	33
Default Browser for Documentation Window	33
Pre-requisite Software for APL64	34
☐ TELPRINT System Function	34
Ctrl+Up/Down Shortcuts.....	35
F9 Shortcut Repeating.....	35
CSE in APL64.....	35
Library Definitions Dialogue Layout.....	35
French Keyboard.....	36
Concatenation Exception	36

Grouped Session History: Scrollbars & Row Headers	36
)LIB System Command Exception	36
Icon for APL64 Workspace	36
User Commands in APL64	37
Duplicate Editors Menu	37
Variable Editor Toolbars	38
Variable Editor: Inserting/Deleting Rows	38
Variable Editor: Line Numbers	38
Preferences Menu.....	39
Shouldn't most options in the Session menu be in a menu named Preferences?	39
Most Recent Object Names	39
)EDIT System Command.....	39
)SAVE System Command.....	40
Documentation Window: Search.....	40
APL+Win Development	40
Text Size in Session	40
Find/Replace Dialogue	41
Find All	42
Row Numbers in Session History	42
File > Copy Dialogue.....	43
Documentation Window: Back Button	43
First Executed APL Function.....	44
Documentation Window: Topic History	44
Documentation Window: Title.....	44
Documentation Window:.....	44
Documentation Window: Navigation Toolbar	45
)LOAD System Command Exception	47
CSE LoadAssembly Method.....	48
Separate Session History Format	48
Scroll Session History Content	49
User-defined Tools	49
Move Between Session History and Command Line	49
Scroll Up Session History.....	50

Session Font Size	52
Debugger Pane Toolbar	54
Copy Session History to Clipboard	54
Session Panes	55
Selection in the Grouped Session History Format	55
Font Size in Command Line	58
Font Alternatives.....	58
APL64 APL Font in ☐ WI.....	59
Sequential vs. Classic Session History Format	59
Additional Coloring Options.....	59
Scientific Notation.....	59
Pause Execution	60
Ctrl+Break Shortcut.....	60
Documentation Window: Go Back Option.....	60
CSE Dynamic Exception	60
Application System GUI Development.....	61
☐ WCALL & ☐ NA.....	62
Links to APL64 Documentation	62
Documentation Windows Location	63
Off-line APL64 Documentation	64
Session History Source of Executed Statement	64
☐ XLIB.....	65
Download APL64 Documentation.....	65
APL+Win Colors in APL64	65
Color Dialogue.....	65
)OFF System Command.....	65
Clear Session History.....	66
Variable Editor: Insert/Delete Rows	66
Variable Editor: Line Numbers	67
Depth Primitive Function	67
)SINL System Command.....	68
ADF File	68
Do I need a new ADF to make calls to windows?	68

Invalid Function Header	68
☐ sysinit	69
Status Bar	69
F1 Documentation Shortcut.....	69
Component Files	70
☐ Fcreate & ☐ XFcreate.....	70
CSE Results	70
APL+Win Text Editor	70
Rotate Primitive Function	71
)edit & ☐ edit	71
Documentation Window: Search	72
Most Recent Object Names	72
:CATCHIF.....	73
Next Pre-Production Version	73
Questions about the next pre-release version	73
APL64 Effect on APL+Win	73
Excel Interface.....	73
CSE Exception.....	73
Component File Exception	74
CSE: REPLSTR Property.....	74
CSE: Enumeration of Static Methods.....	76
CSE: DateTime Exception	76
CSE: ☐ wi 'Reset' Effect.....	76
CSE: Quick Start Document.....	76
CSE: Examples are now in CSE Manual	76
☐ wi 'Reset' Action.....	76
Octal Value Entry	76
Hour Glass Mouse Pointer	77
WebView2 Document Viewer.....	77
F10 Short Cut	77

Overview

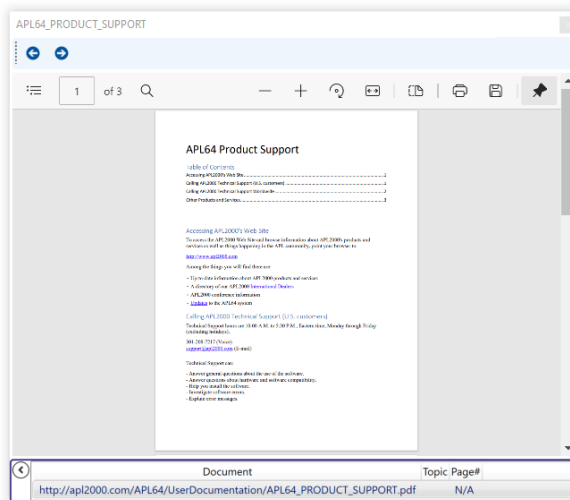
The first pre-production version of APL64 was made available to evaluators in May 2021. This document describes the enhancements, bug fixes incorporated into the second pre-production version of APL64 release in July 2021. This document also describes the evaluator comments and APLNow responses based on the first pre-production version of APL64.

Enhancements

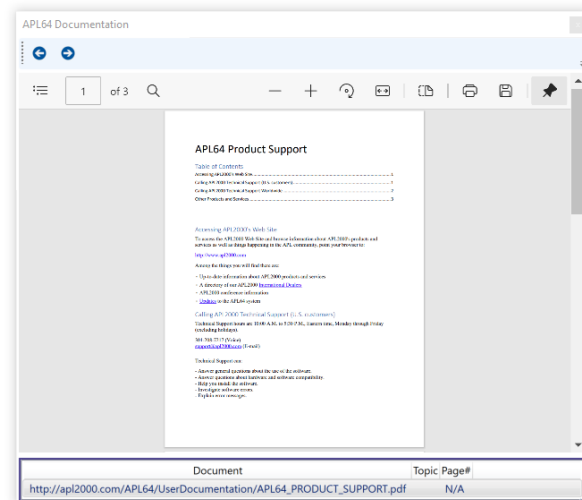
Documentation Window Title

The Document WebView help window now displays the title of the topic in place of the generic *APL64 Documentation* window title:

Now:



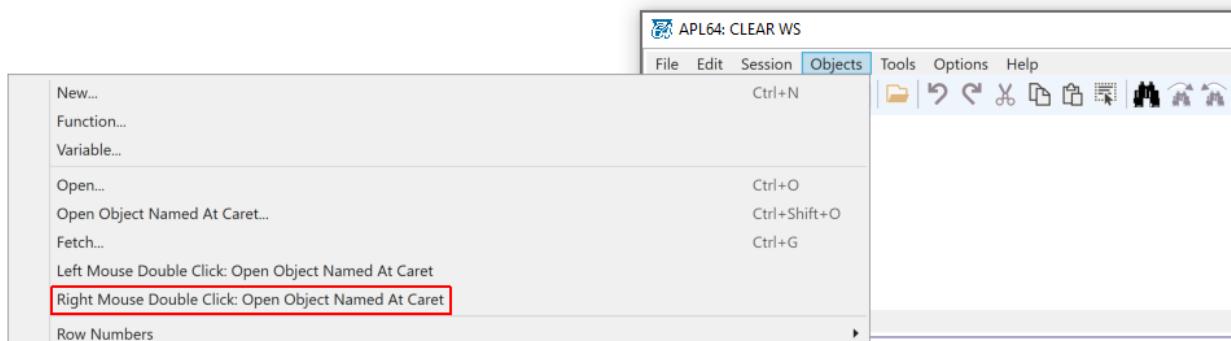
Previously:



Open Object With Right Mouse Double Click

New menu option Objects > Right Mouse Double Click Open Object Named At Caret

When checked, the Windows standard context menu will be suppressed and the APL+Win-style right mouse double click will open the object named at the caret:



Clear/Restore Session History Short Cuts

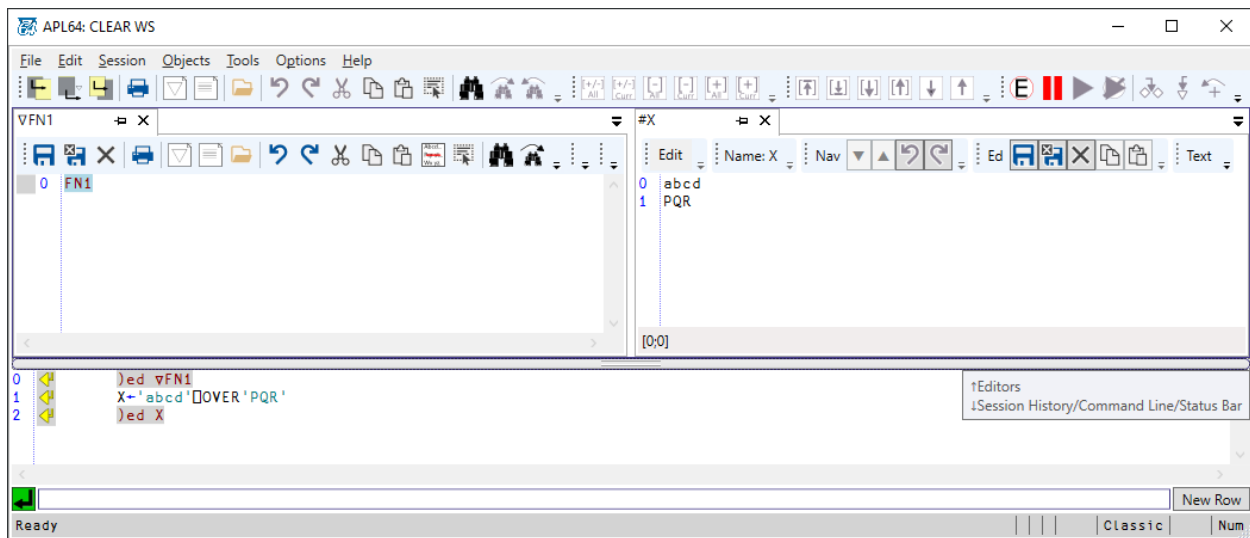
Keyboard shortcuts were added to the current Clear Session History and Restore Session History menu options:

- Shift + Delete to Clear Session History
- Ctrl + Shift + Delete to Restore Session History



Variable Editor Row Numbers

When Objects > Row Numbers > Display Row Numbers in Function Editor or Display Row Numbers in Variable Editor options are checked, the row numbers will begin with zero (0) which is consistent with the Session > Display Row Numbers in Session History option:



Options > Colors Dialogue

Renaming Cloned Color Set

When renaming a cloned color set: (a) The cloned color set name text is selected and (b) The Update Name button is the default button.

Color for Blank Area

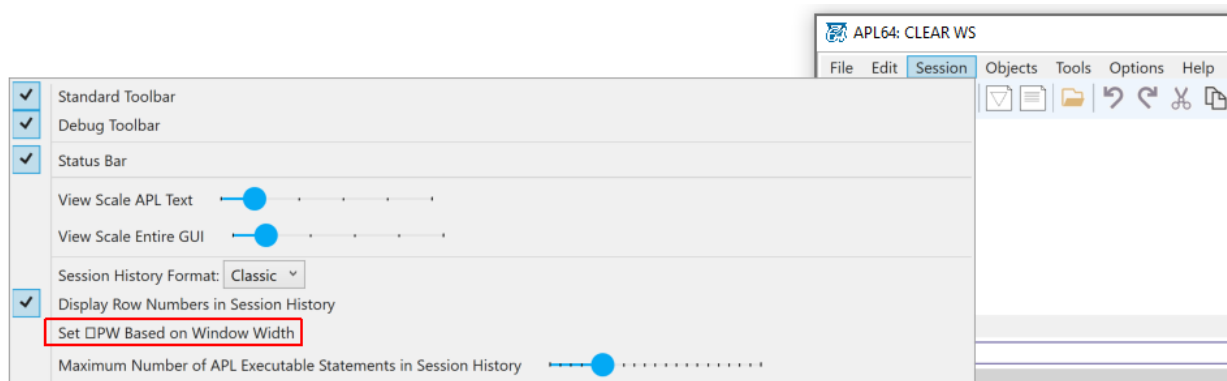
A color for the Blank Area color element now displays in the session for regions of an APL expression which do not include interpreter-provided colorization.

Color Picker Window

The color picker window no longer interferes with another dialogue when the keyboard focus is moved away from APL64

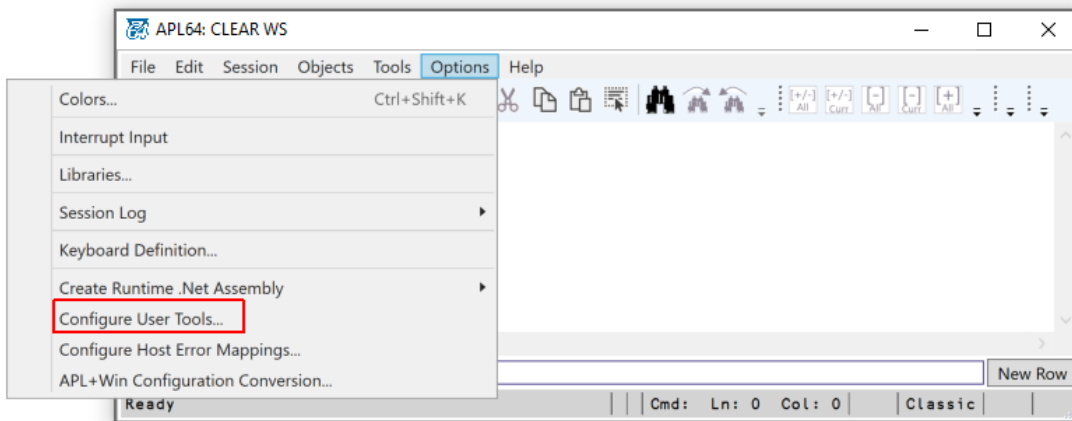
☐ PW Based on Window Width

When the menu option *Session > Set ☐ PW Based on Window Width* is enabled, the value of ☐ PW will consider the current session font size.

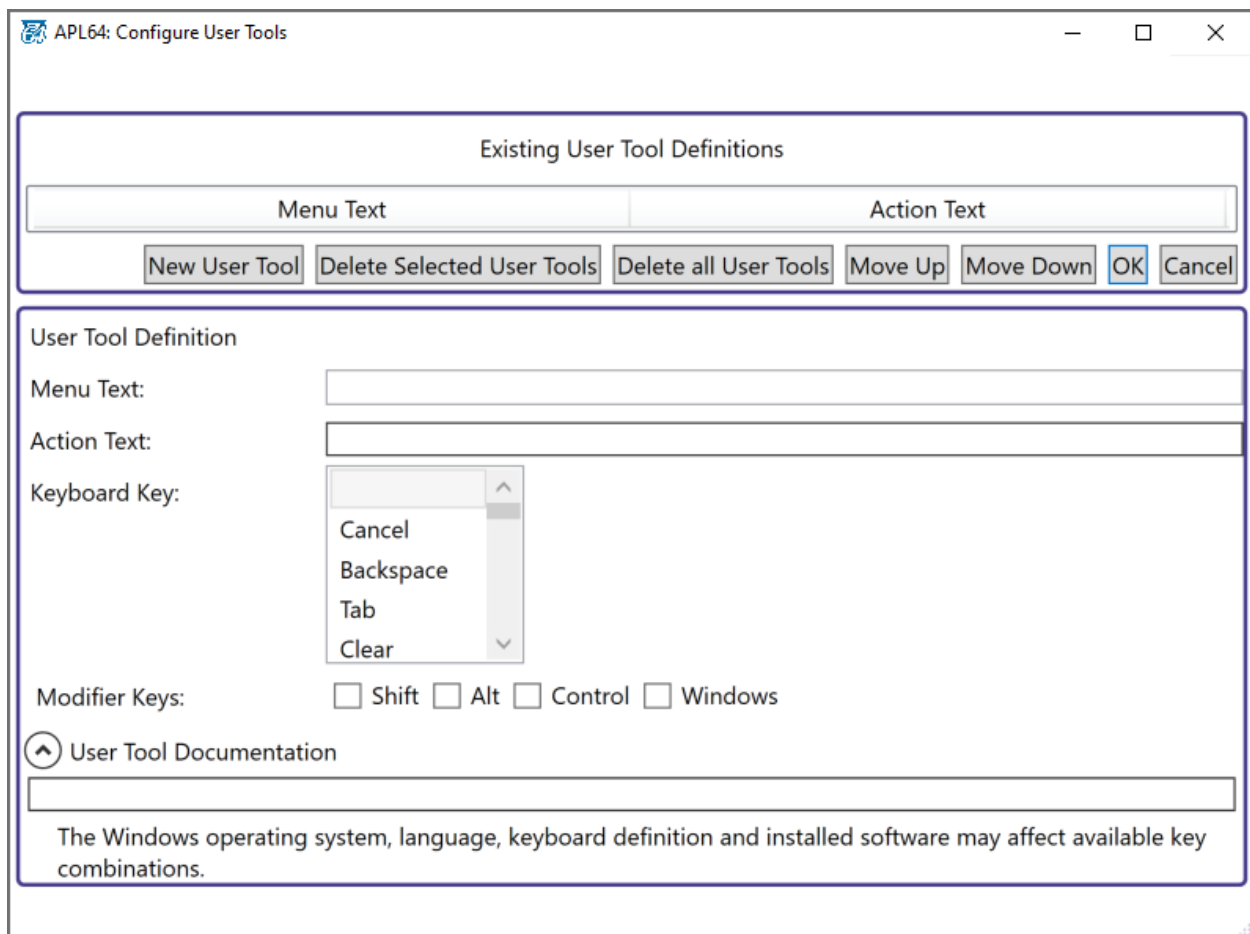


Options > Configure User Tools

New Configure User Tools (feature is similar to Define Tools in APL+Win) and is accessed with the menu option *Options > Configure User Tools...*



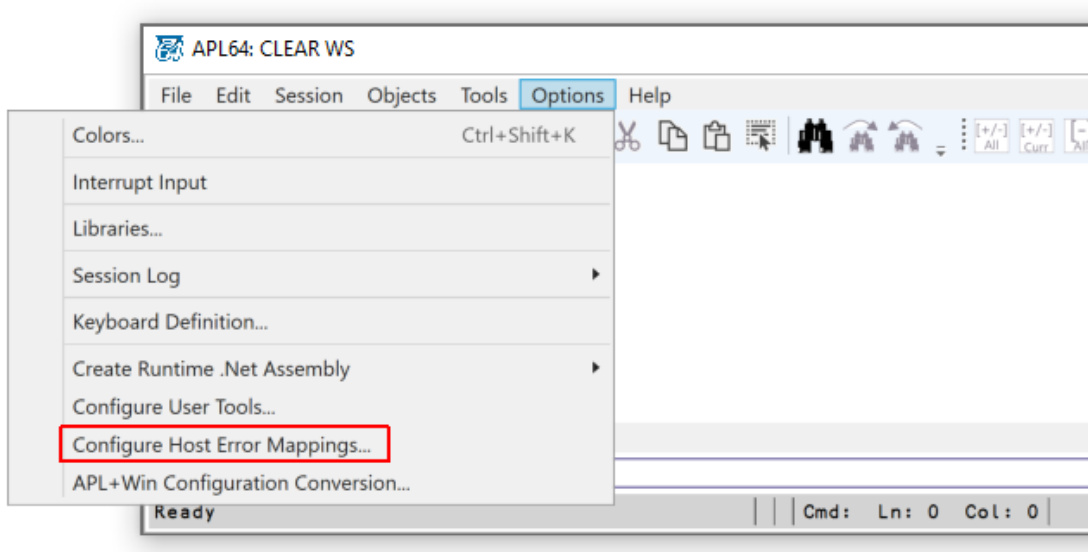
Documentation on the usage of the Configure User Tools feature is available online from the menu option *Help > Developer Version GUI > User Tools*



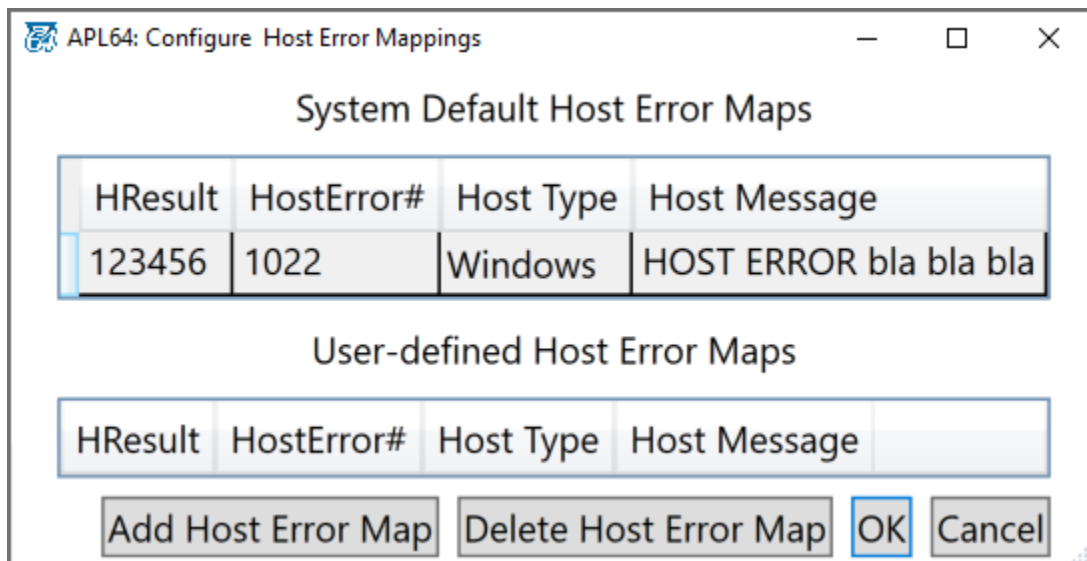
New Configure Host Error Mappings

The text of file system errors is not consistent between Win32 and .Net. In a future version of APL64, the APL64 programmer may use the 'Host Error Mappings' dialogue to coordinate exception handling in an APL+Win-based application with an APL64-based application system.

In this version of APL64, the *Options > Configure Host Error Mappings...* dialogue has been implemented.

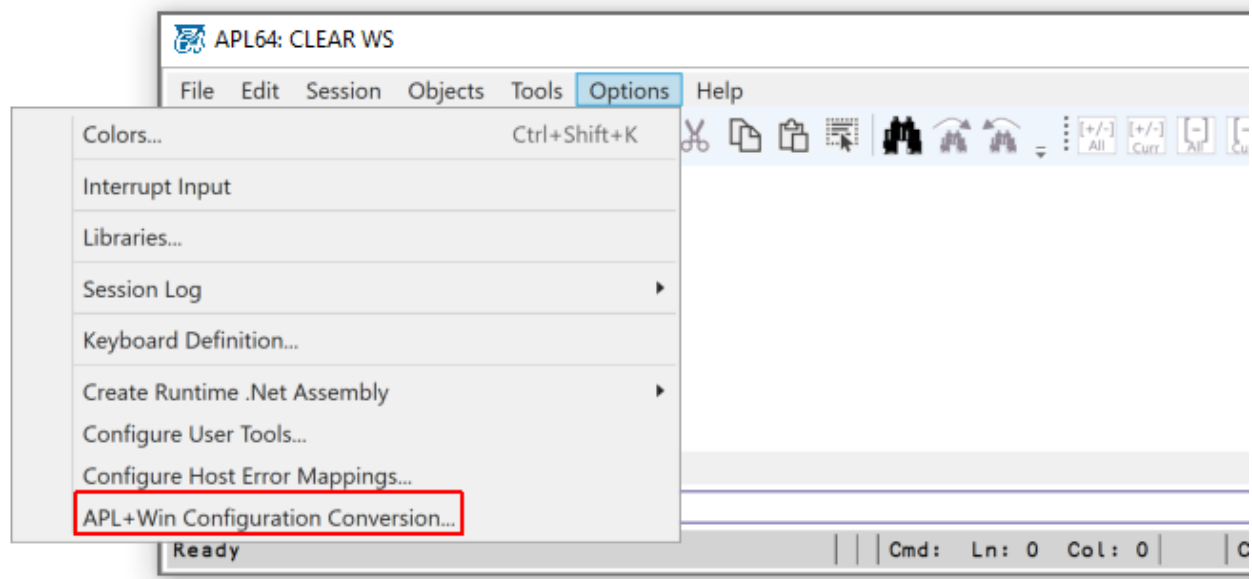


Apply changes to the host error messages in the Configure Host Error Mappings dialog below:

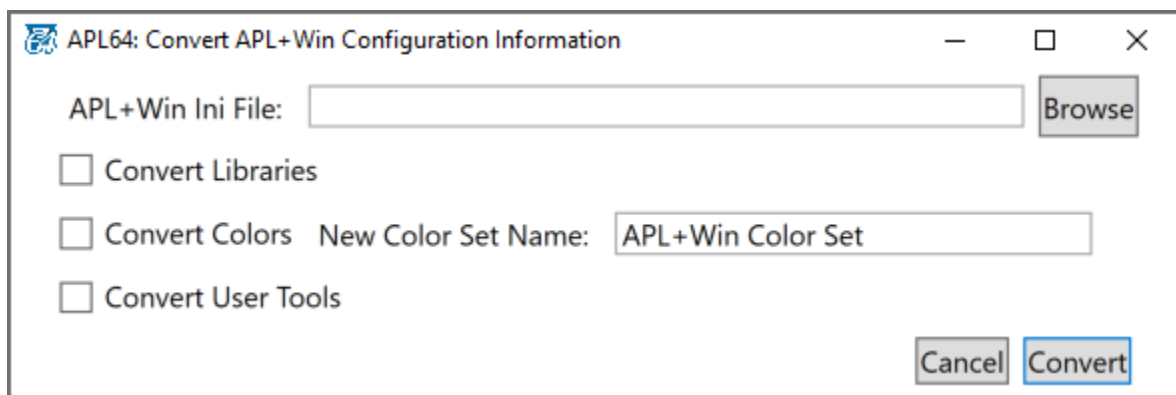


APL+Win Configuration Conversion

Use *Options > APL+Win Configuration Conversion...* dialogue to import APL+Win Libraries, Colors and User Tools to APL64:



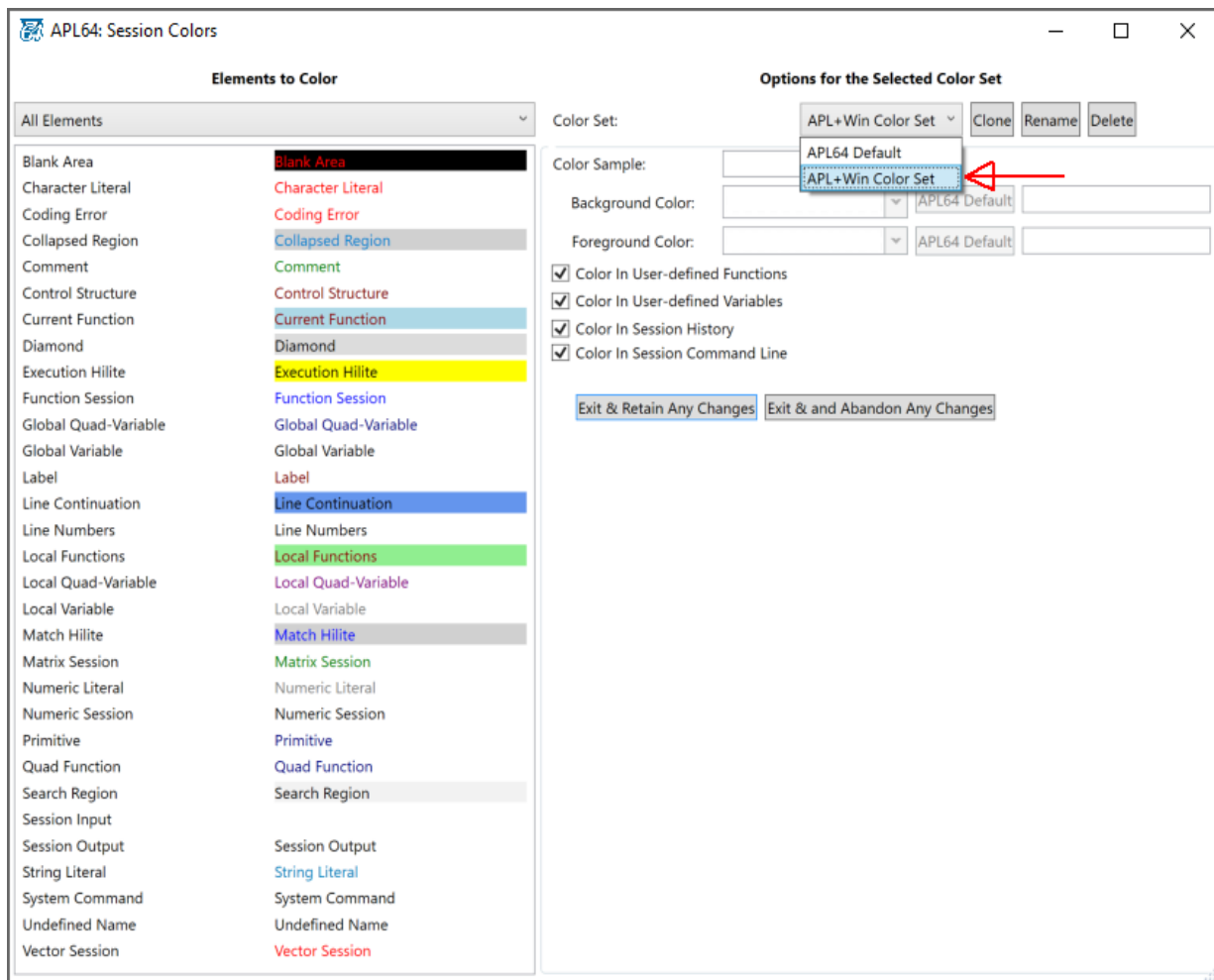
Specify the APL+Win configuration file and the information to import to APL64 from APL+Win



Converted User Tools will appear in APL64 under the Tools menu and are available immediately for use.

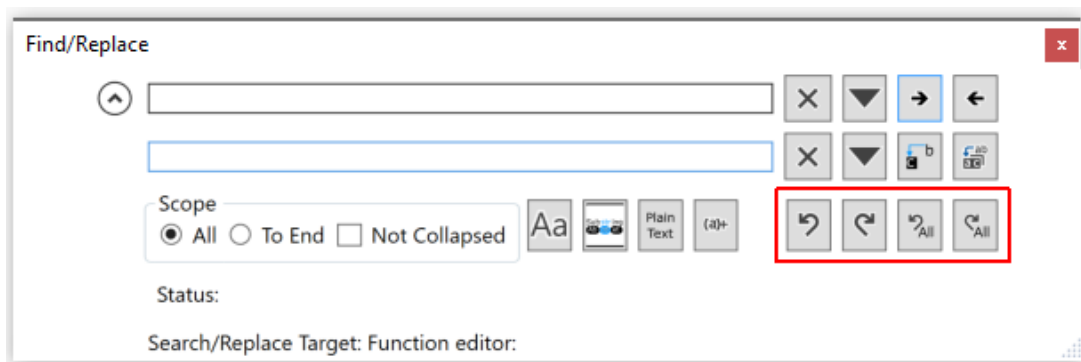
Converted Libraries will appear in APL64 under the menu option *Options > Libraries...* and are available immediately for use.

Converted Colors will be applied to the new color set name chosen. These colors will only be in effect when updating the Color Set in the Session Colors dialog with the menu option *Options > Colors...*

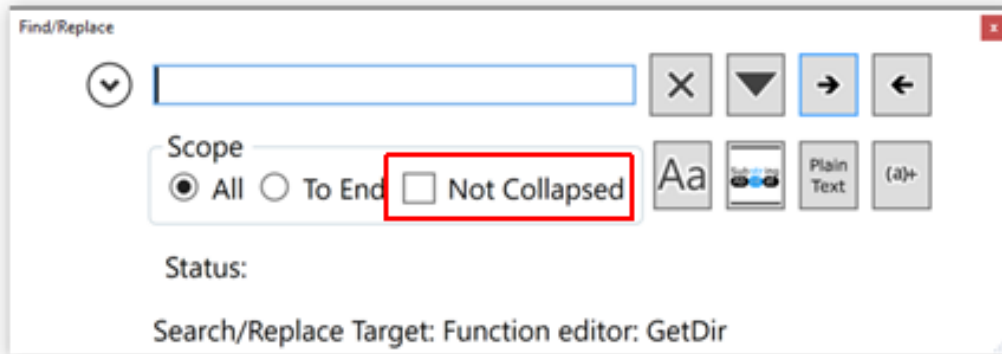


Find/Replace dialog adds Undo/Redo/Undo all/Redo all and excluding collapsed regions
In this version of APL64, the Find/Replace dialogue has these new features:

- Undo, Redo, “Undo all”, and “Redo all” previous changes(s)



- The new search scope option 'Not collapsed' to control when the search operation excludes collapsed regions. **Note:** This option only applies to the function editor.



Bug Fixes

Rotate Primitive Function

A bug in the rotate primitive function incorrectly reported a SYSTEM ERROR for example:

$Z \leftarrow (\epsilon''(1 \oplus Z, ''('', '')) \sim ''')$

:CATCHIF

A bug in the :CATCHIF control structure statement incorrectly reported a SYSTEM ERROR

☐ sysinit

⇒ ☐ sysinit incorrectly returned NONCE ERROR

Scientific Notation

Executing 4.0002E10 in the session command line incorrectly returned 4.0002E14

Shared Network Drive Paths

The File menus: Load, Xload, Copy, Pcopy, Save, SaveAs and Drop, failed to function correctly for workspaces on a shared network drive.

☐ telprint

The ☐ telprint system function may format results incorrectly in the session

Switch Vertical Scroll Bar & Row Headers

Setting the menu option Session > Grouped Session History Format > Switch Position of Vertical Scroll Bar & Row Headers in Results Grid, caused an internal error message resulting in APL64 exiting unexpectedly.

☐ XLIB

☐ XLIB 'test file.sf' incorrectly reported: "SYSTEM ERROR:Exception: The parameter is incorrect." Instead of XFHOST ERROR: CreateNew 1010 0 0 The file 'test file.sf' already exists.

Decode Primitive Function

The Decode primitive function example below produced an incorrect result

100 100 10000 100 100 ⊥ 28 5 2021 5 47
4.000204197E10

the correct result is 2.805202105E11

Concatenation

The statement `var,←` incorrectly returned NONCE ERROR: FIXME: Code.XCATINTO not yet implemented

Sizing of Text

The Ctrl + MouseWheel scrolling only applied to the sizing of the text in the session editors and not the session history.

When debugging, the size of the font in the State Indicator pane would appear larger than the other debugger panes in the session.

|^2*31 reports LIMIT ERROR

In the pre-production release, | ^2*31 incorrectly returned LIMIT ERROR instead of the value 2147483648.

)sinl system command not working

In the pre-production release, the)sinl system command reported the error message:

FIXME: FormatNameList() not done

Evaluators Comments & APLNow Responses

This section contains the summary of the evaluators' comments (feedback) and APLNow's responses to the Pre-Production Version of APL64 (2020.0.1.12467). We would like to thank the evaluators for their contribution.

When to Port Applications to APL64

Should developers port their 32-bit APL+Win applications to APL64 or to wait until APL64 has further stabilized?

APLNow LLC is not advising customers to port their production applications to the pre-production version of APL64 for production use. APLNow LLC is offering you the opportunity to investigate the potential to use APL64 in your environment. You may run your application in the pre-production version of APL64 within a test environment for evaluation purposes. APL2000 would be glad to hear the results of this testing.

Custom Installation Options

Will the installer allow "custom" option during prerequisites checking at startup?

The components which are installed with the APL64 developer version are required components of the application. There are no options available, as without these components, the APL64 developer version will not operate properly. Certain Microsoft components are required to be installed for proper operation of APL64 Developer version. The Microsoft installers for those components can be used to uninstall the component if desired. When viewing APL64 Developer version documentation, the user may select the default browser for pdf-format files.

APL64 Installation impact on APL+Win

Will the installation of APL64 impact the APL+Win installation?

No. The installation and operation of APL64 Developer version is independent of the installation and operation of existing installations of APL+Win.

Installing APL64 on the Same Machine as APL+Win

Is it alright to install APL64 on the same machine with APL+Win?

Yes. APL+Win and APL64 can be installed on the same machine and run independently of one another.

APL64 Installation Target Drive

Can APL64 be installed on a drive other than C:?

The suggested installation target for the APL64 developer version may be modified during the installation process by the APL64 programmer performing the installation.

Note that certain components required for proper operation of the APL64 developer version are Microsoft components and the installation target for those components is controlled by the associated Microsoft component installer.

Certain folders used by the APL64 Developer version are determined by the Microsoft Windows operating system, e.g. the path reported by `%userpath%`, which is selected because it assures read/write permissions on the APL64 programmer's workstation.

☐ new in APL64

☐ NEW is not yet operational and may not be included in the first production release of APL64.

Class inheritance in APL64

:CLASS is not yet operational and may not be included in the first production release of APL64.

Class Definition in APL64

How to know which classes have been defined?

The :CLASS construct in a function header is NOT yet operational. You CAN code :CLASS but at the present time it behaves like the :DEF keyword used in a function header. We may add this feature in a future release.

If this feature is added in the future, then there will be a new ☐NL category and a)CLASSES system command added to system to return the list of classes.

APL64 Documentation Availability

Preview documentation before installing APL64 software?

It is necessary to accept the EULA during the installation process and complete the activation procedure before any element of the APL64 developer version is available for use by an APL64 programmer.

User Commands in APL64

User Command Files are not included in the pre-production version

A future release of the Pre-Production version of APL64 will include updated user command files that are supported in APL64.

User Command]DISPLAY

]display not working

A future release of the Pre-Production version of APL64 will include updated user command files that are supported in APL64.

Output Display in APL64

Output display in APL session is slow

Please send support@apl2000 your workspace with 1000 functions so performance comparisons can be made.

For typically useful numbers of rows of text to display, the performance of APL64 is superior to APL+Win. Performance results will depend on:

- Target workstation hardware
- User-selected APL64 main window size
- Source data size
- Number of pre-existing rows of session history

For example: The average time to display the output of $X \leftarrow 1000p0.01 + i \times / 1000$, clearing session history after each trial:

System	Session History Format	Seconds
--------	------------------------	---------

APL+Win v19	Classic	>12
APL64 v2020.0.2.11531	Classic	<3
APL64 v2020.0.2.11531	Sequential	<8
APL64 v2020.0.2.11531	Grouped	<3
APL64 v2020.0.2.11531	Separated	<4

Documentation Windows Navigation Toolbar

The navigation toolbar is not visible on help files unless window is a certain size

The Documentation Window presentation of pdf-format files is based on the Microsoft WebView2 GUI component which is based upon the Google Chromium browsing engine. This engine does not currently support wrapping of the navigation toolbar. Instead, this engine reduces the buttons in the navigation toolbar as the window width is reduced by the user, eventually eliminating all the buttons. The workaround recommended by the developers of the Google Chromium browsing engine is to apply a minimum width to the window, so that the 'essential' navigation toolbar buttons remain visible.

In the next available APL64 update, a minimum width will be applied to the documentation window so the 'essential' navigation buttons will remain visible:

APL64_APL_PRIMITIVE

←

→

17

of 48

🔍

—

+

🔄

📄

⋮

📌

A←2 3pi4

A≡A

1

A≡'A'

0

Depth =

Syntax:

res ← ≡ arg

Description:

Levels of nesting in an array.

arg: any array.

res: the number of times you must use Pick (dyadic ⋈) to extract the most deeply nested simple scalar from arg.

Example:

≡3.3

0

≡1 2 3

1

≡"

1

≡2 3 4pi24

1

≡(1 2) (2 3) 'AB'

2

≡CCCCC12 12

6

∈ Member of

Syntax:

res ← larg ∈ rarg

Description:

Compare the contents of two arrays.

larg, rarg: any arrays.

res: same size as larg and contains a 1 if larg item is found anywhere in rarg; otherwise, 0.

ext: ⍵ct.

Example:

2 5 ∈ 1 2 3 4

1 0

Note: For monadic ∈, see Enlist in the "Structural Functions" section of this chapter.

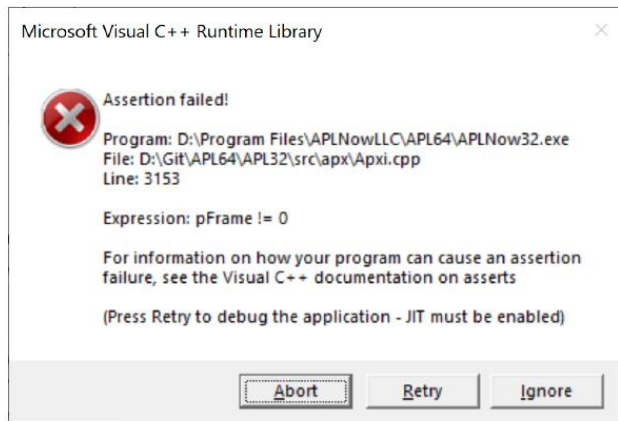
©APLNow LLC – All Rights Reserved - 5/11/2021 - Page: 17

Document	Topic	Page#
http://apl2000.com/APL64/UserDocumentation/APL64_SYSTEM_FUNCTION.pdf		N/A
http://apl2000.com/APL64/UserDocumentation/APL64_APL_PRIMITIVE.pdf	∈	17

Assertion Failure

APL64 reported an error report with the string: Assertion failed! Expression: ...in Axi.cpp[3153]...pFrame != 0

©APLNow LLC – All Rights Reserved - 7/18/2021 - Page#: 20



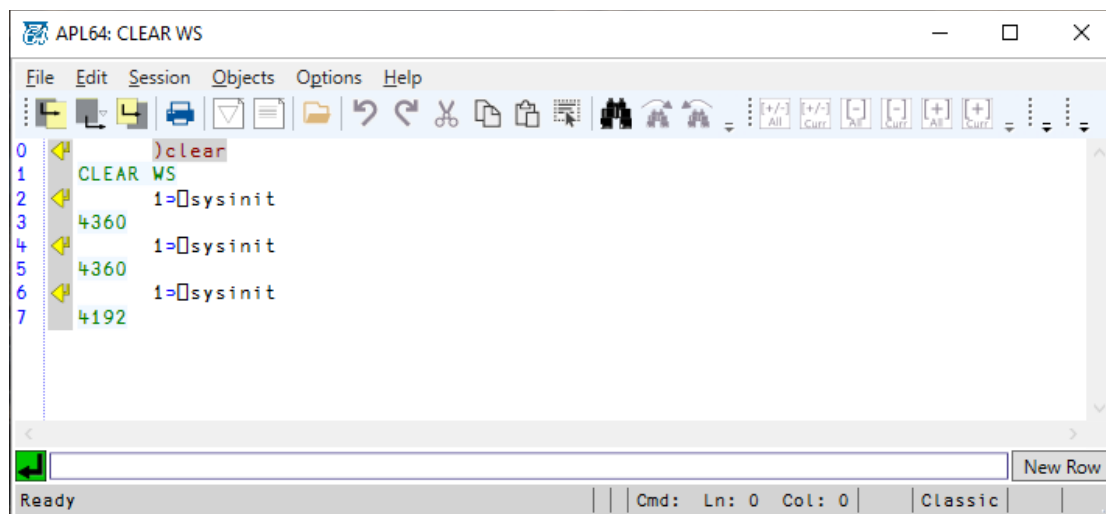
E.g.,

This issue will be addressed in the next available APL64 version.

Invariant Process Handle

Why would the process handle change within a session?

E.g.,



In the next available version of APL64 the first element of the result of `⎕SYSINIT` will be `Process.GetCurrentProcess().Id`. This static value will replace the prior result which was a handle to a .Net interface to the process which is not static.

Variable Editor Line Numbers

There are inconsistent line numbering starts with 0 everywhere except when editing numeric arrays

A future modification of APL64 will address this issue.

Variable Editor Toolbars

When **Objects > Toolbars > Display Toolbar** in **Variable Editor** menu is unchecked, how is anyone to find out that CTL + E, CTL + S, CTL + Q exists?

Once a user is familiar with the options of the variable editor, the user may uncheck the **Objects > Toolbars > Display Toolbar** if desired. This is analogous to learning GUI options via tooltips and then when experience is gained, the user can turn them off.

Variable Editor: Non-homogeneous Element Types

Why are non-numeric values allowed when editing a numeric variable?

The variable editor in APL64 is not restricted like the APL+Win variable editor. The APL64 unified variable editor can edit any APL64 variable. Features exclusive to APL64 variable editor include the ability to edit objects of rank > 2 and objects with elements of mixed data type, rank, and shape.

CSE Exception

What is the reason for the CSE ERROR: Message: (1,1): error CS0103: The name 'myCOMObj' does not exist in the current context

All of APL64 has been implemented in .Net Core 3.1 because that is necessary for cross-OS platform compatibility of the APL64 interpreter. The .Net Core 3.1 tools are a subset of the .Net Framework because some features of the .Net Framework are Windows-only technology.

Assemblies to be loaded into the application domain of the APL64 interpreter, e.g. via ☐ CSE, must be based on .Net Standard 2.0 (or greater) or .Net Core 3.1. This is probably the reason the assembly you are trying to load via ☐ CSE is not compatible with APL64. If there is a .Net Standard- or .Net Core-based version of your assembly, you may be able to reference it in a custom assembly you create with Visual Studio so that your custom assembly can be used in APL64.

Technology, such as ActiveX and Win32, are Windows-only technology that have been deprecated by Microsoft. ☐ WI is provided to APL64 programmers who are targeting only the Windows OS environment via a proxy to the APLNow32.exe component of APL64.

Developing a Cross-platform Application GUI

With APL64, what do you use to build a cross-platform GUI? I think ☐ cse will be unwieldy for this purpose.

The APL64 ☐ CSE cannot be used to develop a GUI. The APL64 ☐ CSE is implemented using Microsoft .Net Core so that it is cross-platform compatible. The Microsoft .Net Core design including the Roslyn .Net compiler prevents the APL64 ☐ CSE from interacting with WPF or System.Windows.Forms GUI controls.

The APL64 ☐ WI GUI tools cannot be used to develop a cross-platform compatible GUI because they are based on Microsoft-deprecated, Windows-only technology.

This situation is to be expected, considering information published by Microsoft about their future plans as well as new components and products they have developed or acquired. The role that Microsoft has outlined for its premier programming language, C#, is as the tool to support the calculation algorithm portion of an application. The other application components can be selected from a panoply of options, some provided by Microsoft, and used in application development via 'dependency injection' to create cross-platform applications. Numerous Nuget packages are available to supplement the basic features of C# to complete the application development framework.

Substitute APL64 for C# to understand that designing a cross-platform GUI requires fundamental design decisions on the part of the C#/APL64-application programmer:

- The application must be designed with appropriate separation of GUI, calculation algorithm, output preparation and data persistence. Separating the GUI component will permit the application to remain viable when the inevitable changes in GUI technology occur.
- Based on the target platform and the anticipated application user base, the application designer must choose one or more GUI tools to develop the application GUI.

A traditional APL programmer has had the benefit of a single APL-centric development environment which included support for GUI, calculation algorithm, output preparation and data persistence. There is no viable way to provide an all-inclusive APL environment which would provide cross-platform compatibility.

In a cross-platform design environment, APL64 can support elements of all application components except for the GUI component. This means that to develop a cross-platform compatible application the application programmer must incorporate development tools which include non-APL64 based tools for GUI development.

Here are some of the GUI development tool choices, noting that these are all continuously undergoing substantial enhancement and modification and may be mutually incompatible:

- HTML/javascript for a browser-based GUI
- ASP .Net for C# syntax which compiles to javascript for a browser-based GUI
- Xamarin which compiles to native Windows, IOS and Android GUI
- WPF and System.Windows.Forms for a Windows-only environment
- Blazor for browser-based applications where all the code (GUI, C#/APL64) run in the browser

See also:

https://en.wikipedia.org/wiki/List_of_platform-independent_GUI_libraries
<https://visualstudiomagazine.com/articles/2017/01/25/26-web-desktop-ui-tools.aspx>

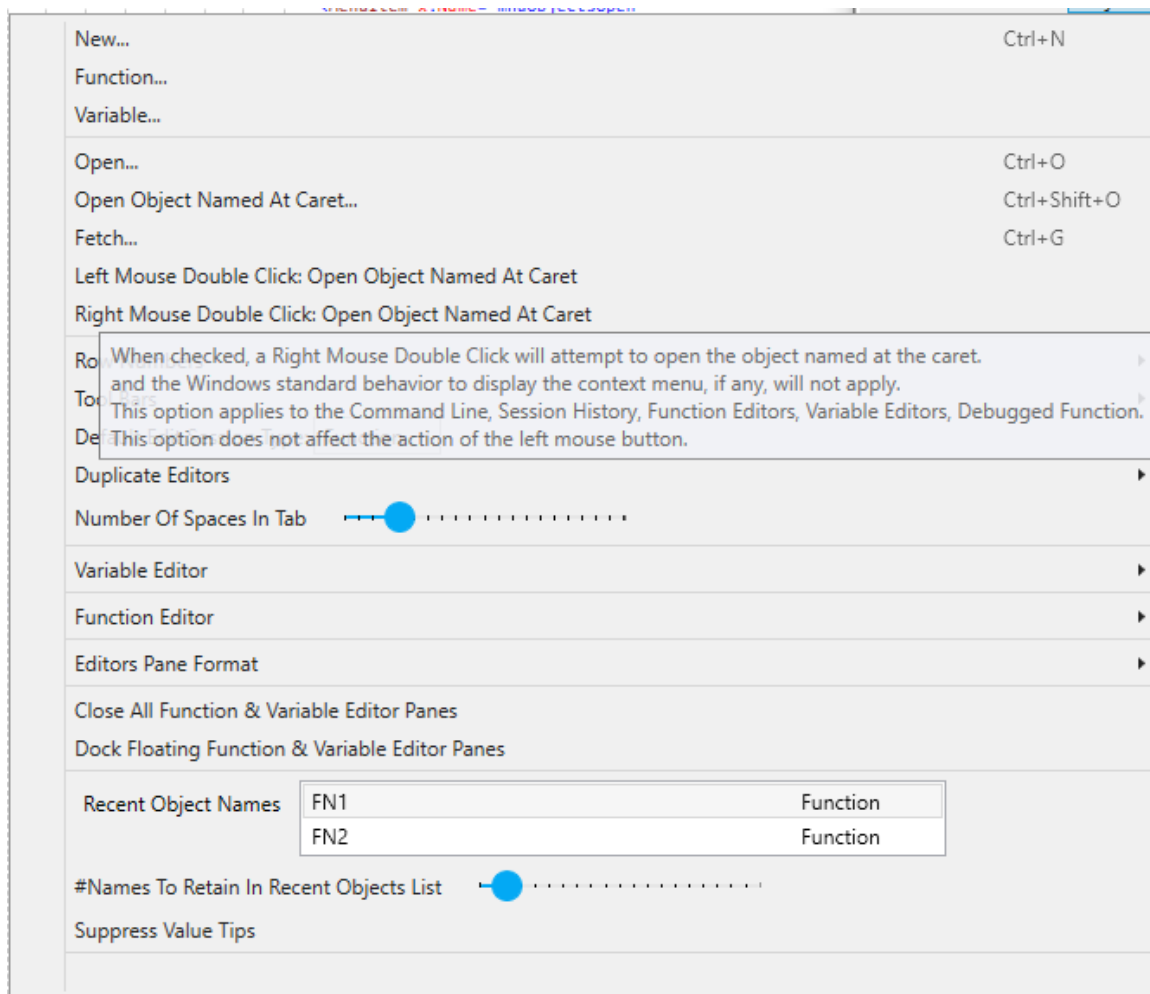
Overall the question you raise is fundamental and can be re-phrased to: How to encourage APL-centric application developers to consider APL as a component and not the entire application so that cross-platform applications can use the benefits of APL?

Context Menus in the APL64 Session

A right mouse click to open a function now has a menu, it was a very common gesture. Could that menu be optional?

The right mouse click in Windows desktop applications displays a context menu for the GUI control if the context menu is defined. The right mouse click in APL64 displays a context menu which includes the 'Open Object At Caret' action.

A future version of APL64 will provide an Objects menu option to block the Windows standard context menu presentation and provide the Open Object At Caret action for a right mouse double click:



)COPY System Command

The)copy system command can fail with spaces in the path name

When using the system commands `)copy` or `)pcopy` to copy from workspace file names that include one or more spaces, you can must delineate the workspace name from the object(s) using one of the two options listed below:

- append a semi-colon to the long file name (like in APL+Win)
- specify the long file name inside a pair of single or double quotes (new in APL64)

Below are some examples:

```
)copy "D:\copy test"
```

```
)copy "D:\copy test" a
```

```
)copy D:\copy test;
```

```
)copy D:\copy test; a
```

Persistent File Filter

Is it possible to have the workspace filter in the File > Load and Copy dialog boxes to 'stick' the next time they're opened?

We are considering your request to have the workspace filter remain persistent as an enhancement to APL64.

☐ wslib

Can ☐ wslib return both APL64 and APL+Win workspaces?

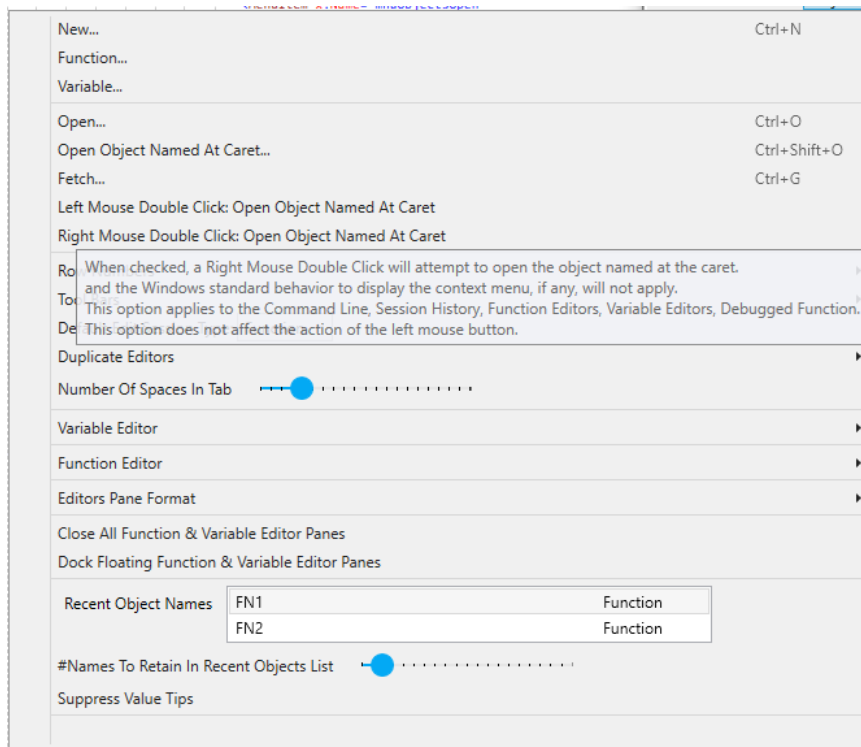
Your enhancement to allow ☐ wslib to also show workspaces from APL+Win with the `.w3` extension will be considered.

Double Right Click

I was expecting double right clicking `zzInit[281]` in the session to open `zzInit` in the Editor

The right mouse click in Windows desktop applications displays a context menu for the GUI control if the context menu is defined. The right mouse click in APL64 displays a context menu which includes the 'Open Object At Caret' action.

A future version of APL64 will provide an Objects menu option to block the Windows standard context menu presentation and provide the 'Open Object At Caret' action for a right mouse double click:



Webview2 Documentation Browser

It would be good that the WebView help window title not be just always: APL64 Documentation but also include the title of the topic

The next update to the pre-production release of APL64 will have a helpviewer window displaying the documentation filename (sans extension and folder) in the title bar in place of "APL64 Documentation".

)LIBD System Command

Is the)LIBD system command supported?

)LIBD is not a valid system command in any APL2000 APL system.

ASMFNS

Are ASMFNS Deprecated IN APL64?

ASM functions depend on ☐CALL. ☐CALL is dependent on Win32. APL64 is a 64-bit application, so using ☐CALL is not possible. This is a permanent situation for APL64.

There is another pathway to using unmanaged assembly language programs if they are 64-bit. It is possible using the enhanced 'EXT' option for ☐NA in APL64 to access a managed C++ dll which can access a 64-bit assembly language program. A programmer versed in managed C++ and 64-bit assembly language

programming could use this technology in APL64. It is not clear that performance gains would be achieved, so some experimentation and testing is necessary.

Currently in APL64 most of the ASM functions have APL64 system function analogues, e.g. `⎕DEB`, `⎕SSTOMAT`.

Rotate Primitive

`Z←(ε¨(1⊕¨Z,¨(' ',¨'))~¨' ' produced error message: SYSTEM ERROR:Aval.Categorize encountered an uncategorized child element at index=0`

This issue has been corrected and the fix will be available in the next APL64 Pre-Production version that we anticipate releasing within the week.

APLGrid in APL64

Beware that the APLGrid only supports APL+Win 19 and newer. A new "APLGrid" supporting APL64 will be released in the near future.

APLDraw in APL64

Replacements for ActiveX components APLDraw and APLGrid?

When an APL64 programmer is targeting the Windows operating system environment, ActiveX components, including APLDraw and APLGrid may be used. ActiveX is a Windows-only technology and has been deprecated by Microsoft, so ActiveX components will not be available when targeting the IOS or Android operating system environments.

Currently the APLGrid distributed with APL+Win doesn't operate in APL64. An APLGrid update will be available in a future APL64 version update which will be suitable for the Windows operating system environment as an ActiveX object in an APL64-based application system.

For an APL64 programmer targeting a cross-platform operating system environment, there are many commercial 'grid-style data editors' available, such as SpreadsheetGear for .Net Standard. When targeting a cross-platform operating system environment, APL64 can be used for application algorithm/calculation support, but for the GUI development of such an application system, the programmer has many choices available each with unique features and syntax.

ODBC Database Access in APL64

What will be the interface to ODBC databases in APL64?

The current APL+LinkPro, which depends on `⎕wcall` will continue to operate in APL64.

Using the APL64 `⎕WI`, the Active-X version of the Microsoft ADO technology can be utilized. Note that ActiveX technology is available only in the Microsoft Windows operating system environment.

Using the APL64 ☐CSE, the .Net version of the Microsoft ADO technology can be utilized.

☐WI Accessing 64-bit DLLs

APL64 will not access 64-bit DLLs with ☐wi, correct?

If the 64-bit dll is a .Net Assembly which was custom written by the APL64 programmer, the ☐NA system function using the new 'EXT' mode may be used to access the assembly in APL64.

If the 64-bit dll was not custom written by the APL64 programmer, e.g. a licensed dll, use the ☐CSE system function to access it in APL64.

☐WI may be used to access LocalServer executables that are either 32- or 64-bit.

☐WI may also be used to access ActiveX components.

Because ActiveX technology is not cross-platform, Microsoft has deprecated ActiveX technology in .Net Core, on which APL64 is based.

Session Command Line

Separating the APL Session command line below the session needs getting used to. At the moment, this is very unusual and circumstantial to me. I don't like it so far.

In APL64 the session history is not editable so that it accurately represents what occurred during an APL64 Developer version instance. This is not the case in APL+Win.

Right Mouse Double Click

Open an existing function in the editor should work with right mouse click (like in APL+Win)

The next APL64 version update will include a new Objects menu option to provide the behavior you requested. When checked, the Windows standard context menus will be suppressed and the right mouse double click will attempt to open object named at caret.

Decode Primitive Function

The Decode example below produces an incorrect result in APL64

E.g.,

```
100 100 10000 100 100 1 28 5 2021 5 47
```

4.000204197E10 A wrong, APL+Win gives me 2.805202105E11

The fix for it will be available in the next APL64 version update.

Scientific Notation

4.0002E10 executed in the session command line incorrectly shows 4.0002E14 as result in the session

We haven't been able to reproduce the issue where 4.0002E10 executed in the session returned the result 4.0002E14. However, this result is returned when executing 40002E10. So, is it possible that 40002E10 without the decimal point is what you had executed?

When APL64 will be available?

The pre-production version of APL64 is available now for those wishing to contribute towards the testing effort.

The production version of APL64 will be available once a suitable level of successful testing has been completed.

What will happen to APL+Win?

The future of APL+Win and APL64 will depend on the adoption of APL64 by the customer base. As noted in the APL64 Project forum documents, the ability to enhance APL+Win is limited because it is based on technology which has been deprecated by Microsoft. Bug fixes, if any, create a potential for future APL+Win updates. You can also view these documents from APL64: Help > APL64 Project History.

APL+Win is currently the production-ready implementation of APL available from APL2000. APL2000 will continue to provide support for this product for current APL+Win Subscription Licensees.

APL2000 has designed APL64 to be compatible with APL+Win so customers may easily load and run APL+Win applications in APL64. We hope customers will benefit from the many new features in APL64 and adopt this new product as the next generation APL language development tool.

Library Numbers

Does APL64 support library numbers? If so, how are they defined?

Yes. Library numbers are defined with the ☐ LIBD system function and in the menu: Options > Libraries.

Note that in APL64 libraries can be marked as transient or non-transient, indicating if they do not/ do persist between APL64 Developer version instances.

Showing Tooltips

How about not showing the tooltip when a line is "pulled down" (aka dragged down)

The cause of the example you provided cannot be resolved by APL64 as it is an underlying Windows operating system behavior.

Generally, the Tooltip window will not obscure the associated GUI control. However, the Windows operating system will modify the position of the Tooltip window relative to its associated GUI control under certain

scenarios so that the entire Tooltip window is visible on the current workstation display. Such scenarios involve:

- (a) The Tooltip text content
- (b) The location of the Tooltip-associated GUI control in the Windows containing that GUI control
- (c) The Size of the Window containing the Tooltip-associated GUI control
- (d) The position of the Window containing that GUI control, relative to the edges of display containing that Window.

Switch Position of Vertical Scroll Bar & Row Headers in Results Grid: Possible exception and behavior issues

The next APL64 pre-production version will resolve this issue: *Also, this option doesn't seem to be working properly when it's already turned on and a new scrollable list is displayed in the session - the scroll bar and line markers show up in their normal positions instead of reversed.* This too will be resolved in the next version update.

Clear Session History Shortcut

A shortcut for Session > Clear Session History is required to speed up interaction with APL64

These new keyboard shortcuts will appear in the next APL64 version update:

- Shift + Delete to Clear Session History
- Ctrl + Shift + Delete to Undo Clearing of Session History
-

Library Definition Dialogue

Will you reconsider the placement of the Add button in the Library Definitions dialogue?

Thank you for your suggestion. It's not a high priority. But we'll consider elevating it if there are more comments like it from customers.

List of System Functions

Is there a new feature for a listing of system functions?

Your enhancement request to return a complete list of the available APL64 system functions is under consideration.

Class Definition

How to know which classes have been defined

The :CLASS construct in a function header is not yet operational. You can code :CLASS but at the present time it behaves like the :DEF keyword used in a function header. We may add this feature in a future release.

If this feature is added in the future, then there will be a new ☐NL category and a)CLASSES system command added to system to return the list of classes.

MessageBox System Function

Is there a system function for display a standard .Net MessageBox?

No. There cannot be such a GUI control because the APL64 interpreter is designed to be operating system 'cross-platform' and GUI development is not cross-platform.

Doesn't every OS have the equivalent of a MessageBox that could be used?

The GUI development tools are a separate component which the application programmer must select. APL64 cannot do that because there are multiple GUI development environments in each operating system environment. The Cross-platform component of APL64 is basically a calculation engine, just like C#. That calculation engine can be employed by event handlers supporting events exposed by the application programmer's selected GUI environment.

For example with the Windows OS, there are several GUI development tools which use the APL64 interpreter to handle an event like a button click, including:

System.Windows.Forms

Windows Presentation Foundation (WPF)

ASP.Net

Xamarin

HTML/Javascript

Universal Windows Platform (UWP)

Win32 MFC

Based on the type of application you are developing, one or more of these GUI development environments may be suitable.

However, each of these have their own distinct and disparate GUI structure, syntax and features. It is up to you as the APL64 developer to select the GUI development environment and when appropriate use the APL64 interpreter, C#, or any one of the other available .Net programming languages to handle the calculation/algorithmic elements of the application.

The same situation applies to the other OS environments in that there are several ways to develop a GUI each with its proponents and detractors. The APL64 interpreter cannot make those decisions for you as the application developer.

As an application developer in the current environment, it is necessary to consider application development as a component selection endeavor. The days of one 'programming language' supporting all aspects of application development ended for most of the IT world in the late 1990's.

How to turn off Tooltips in the session?

The tooltips in the session are managed in the session menu: Session > ToolTips > Tooltips Enabled.

Tooltip Location

With the mouse still in the entry field causes the tooltip to pop up at just the location where the session output for that line is appearing, obscuring the result

Some cases cannot be resolved by APL64 as it is an underlying Windows operating system behavior.

Generally, the Tooltip window will not obscure the associated GUI control. However, the Windows operating system will modify the position of the Tooltip window relative to its associated GUI control under certain scenarios so that the entire Tooltip window is visible on the current workstation display. Such scenarios involve:

- (a) The Tooltip text content
- (b) The location of the Tooltip-associated GUI control in the Windows containing that GUI control
- (c) The Size of the Window containing the Tooltip-associated GUI control
- (d) The position of the Window containing that GUI control, relative to the edges of display containing that Window.

Scroll Up Session History

Do you suppose that scrolling by "Scroll Up Session History" could be done with the margin left blank?

The algorithm you suggest would not work for all session history formats.

Exception: [given key 'e' was not present ...](#)

Why did APL64 return the error message: WS FILE ERROR: Could not load workspace: The given key 'e' was not present in the dictionary?

We had observed error reports similar to this in other contexts during our internal testing. But they all went away after all the affected machines were updated recently with a Windows Update about eight days ago. I was the first one to notice this and concluded that it may have been the Windows 10, version 21H1 Update

(<https://support.microsoft.com/en-us/topic/may-11-2021-kb5003173-os-builds-19041-985-19042-985-and-19043-985-2824ace2-eabe-4c3c-8a49-06e249f52527>).

One way to quickly tell if your machine is "compromised" is by opening the DOS Command Prompt window and examine the case of the drive letter for drive E:. If it appears as a lowercase 'e', e.g., e:\ instead of E:\, then this explains why you are observing this issue. I recommend you check this before applying any Windows Update and then repeat afterwards.

Please report back to us with your results.

Alternative to Session Command Line

Is there no way to get the command-entry to happen within the session itself?

In APL64 the session history is not user-editable. This design keeps the session history accurately representative of what actually occurred during the session. This is not the case in APL+Win.

)COPY System Function Pattern Exception

The)copy system command of an APL+Win workspace returned
"PATTERN «library\lexwje.w3» ERROR: Invalid pattern 'library\lexwje.w3' at offset 9"

This has been fixed and the fix will be included in the next APL64 version update.

Storage of Scalars

Are scalars stored in the Symbol Table?

APL64 is a managed code, .Net application. Unlike APL+Win, it does not access physical memory, is assigned a virtual memory space by the operating system and most memory management is automatically performed by the operating system. The APL64 system functions ☐WA, ☐WSSIZE and ☐WU have been provided for compatibility to minimize application source code changes when transitioning to APL64 from APL+Win.

Default Browser for Documentation Window

Allow user to choose the default browser for displaying the online documentation?

The Microsoft WebView2 component installs the Chromium Engine which is required for proper operation of APL64 Developer version. The first time the documentation viewer is instantiated by an APL64 programmer, the Chromium Engine will add a prompt for the user to select the default browser. In some user-provided environments, it is necessary to enlarge the documentation window to view the entire prompt. Generally, responding to this prompt is not necessary as the Chromium Engine checks the Windows registry to determine the user-selected default browser.

WebView2 is not required for the runtime version of APL64.

Pre-requisite Software for APL64

Does the required pre-requisite software in APL64 completely remove itself when uninstalled?

Pre-requisites have to be met in order to install APL64. The install and un-install of the pre-requisites is solely on the provider. For example, components developed by APLNow LLC will be uninstalled by the APLNow LLC-provided installer and the uninstallation of Microsoft components required by APL64 is the responsibility of the Microsoft-provided installation tool.

What are the external applications required for the development system and which of them will also be required for a run-time environment?

The APL64 Developer version installer will check for the required prerequisites and inform the user of these required components. There are no optional software components of the APL64 Developer version. After installation and activation of the APL64 Developer version is complete, the Help > About window can be used to view the .Net assemblies used by the APL64 Developer version.

For an APL64 runtime application targeting the Windows operating system environment, there is only one application file needed. That file is the Windows runtime executable (.exe) which is created by the APL64 programmer who developed the application using the APL64 Developer version utility: Options > Create Runtime .Net Assembly > Create Windows Runtime Executable. For more information about the Windows Runtime Executable, please read the Help > Developer Version GUI > Create Windows Runtime Executable.

☐ TELPRINT System Function

☐ telprint system function returns wrong (display) result;

E.g.

```
aaa←'G<Test 9999>'⎕fmt 1100
```

```
⎕telprint aaa
```

```
Test 0001 Test 0010 Test 0019 Test 0028 Test 0037 Test 0046 Test 0055 Test 0064 Test 0073 Test 0082 Test 0091  
Test 0100
```

```
Test 0002 Test 0011 Test 0020 Test 0029 Test 0038 Test 0047 Test 0056 Test 0065 Test 0074 Test 0083 Test  
0092Test 0003
```

```
Test 0012 Test 0021 Test 0030 Test 0039 Test 0048 Test 0057 Test 0066 Test 0075 Test 0084 Test 0093Test 0004  
Test 0013
```

```
Test 0022 Test 0031 Test 0040 Test 0049 Test 0058 Test 0067 Test 0076 Test 0085 Test 0094Test 0005 Test 0014  
Test 0023
```

```
Test 0032 Test 0041 Test 0050 Test 0059 Test 0068 Test 0077 Test 0086 Test 0095Test 0006 Test 0015 Test 0024  
Test 0033
```

```
Test 0042 Test 0051 Test 0060 Test 0069 Test 0078 Test 0087 Test 0096Test 0007 Test 0016 Test 0025 Test 0034  
Test 0043
```

```
Test 0052 Test 0061 Test 0070 Test 0079 Test 0088 Test 0097Test 0008 Test 0017 Test 0026 Test 0035 Test 0044  
Test 0053
```

```
Test 0062 Test 0071 Test 0080 Test 0089 Test 0098Test 0009 Test 0018 Test 0027 Test 0036 Test 0045 Test 0054  
Test 0063
```

```
Test 0072 Test 0081 Test 0090 Test 0099
```

This has been fixed and the fix will be included in the next update.

Ctrl+Up/Down Shortcuts

Allow Ctrl+Up/Ctrl+Down to recall/navigate previous commands like F9 and Shift+F9?

This will not be possible because there are already plans for these keyboard shortcuts to scroll the session one row up and down, like in APL+Win.

F9 Shortcut Repeating

The F9 key repeats the previous line twice in the Command window

We are unable to duplicate the issue as reported. Please use Session > Configuration Settings > Export Settings... and send the resulting xml-format file to support@apl2000.com to investigate further.

CSE in APL64

What is the difference between the CSE in APL+Win and the CSE in APL64 as regards the LoadAssembly method?

In APL+Win, the CSE is an external process, i.e. the .Net side of the CSE exists in a separate application domain from the APL+Win application domain. The APL+Win uses LoadFrom, which locks the assembly until the application domain of the .Net side of the CSE is unloaded. In APL+Win this happens when the CSE is closed and the external CSE process is terminated.

In APL64 the CSE is not an external process, i.e. the .Net CSE runs in the same .Net application domain as APL64, providing significant performance benefits. To prevent locking an assembly, the APL64 CSE has its own AssemblyLoadContext. The AssemblyLoadContext loads the assembly using LoadFromStream, and tries to add references by using the MetadataReferenceProperties of the assembly the user wants to load.

However, in APL64 there is a list of forbidden assemblies that get removed from the references (any assembly with System.Windows in the name). When the developer tries to load an assembly with System.Windows, the load fails. However, if only a reference to the assembly the developer wants to load contains System.Windows, the load is reported as successful but the reference is not added.

For cross-platform compatibility, the APL64 ☐ CSE is implemented using the Microsoft .Net Core Roslyn compiler. This compiler emits dynamic assemblies to execute C# statements via the APL64 ☐ CSE. Since the underlying assemblies are dynamic, the thread which would be used to interact with GUI controls is not accessible by subsequent C# statements executed by the APL64 ☐ CSE. This Microsoft design prevents the APL64 ☐ CSE from effectively creating GUI components.

Therefore the APL64 ☐ CSE cannot load .Net assemblies which support GUI technology, such as WPF or System.Windows.Forms. These prohibited assemblies have the 'System.Windows' name prefix.

Library Definitions Dialogue Layout

I don't think the Add button should be above the Browse button in the Library Definitions dialogue. I think that putting it below or to the right of Browse would be less confusing.

This isn't a high priority. But we'll consider elevating it if there are more comments like it from customers.

French Keyboard

The French keyboard seems to work pretty well

Thank you for your comment on the operation of the French keyboard. Let us know if you discover any issues.

Concatenation Exception

"var,←" gave NONCE ERROR: FIXME: Code.XCATINTO not yet implemented

The fix for In-place assignment with catenation will be available in the next APL64 version update.

Grouped Session History: Scrollbars & Row Headers

Selecting "Session > Grouped Session History Format > Switch Position of Vertical Scroll Bar & Row Headers in Results Grid" caused an internal error message and APL64 shut down.

This bug will be fixed in the next pre-production version update.

)LIB System Command Exception

)lib 'fake' reports "SYSTEM ERROR:Exception: The filename, directory name, or volume label syntax is incorrect. : 'C:\Program Files\APLNowLLC\APL64'c:\aplwin library"

We have been unable to reproduce the SYSTEM ERROR:Exception error message with)lib 'fake'; meaning APL64 consistently returns the LIBRARY NOT FOUND message.

Please send us a reproducible case if you have one.

Icon for APL64 Workspace

Why not show the *.W3 [APL+Win] workspaces with the old icon and use the new icon for WS64 [APL64]?

The assumption is you're referring to the appearance of the icons for the workspace files in Windows Explorer, in which case I'm seeing the correct icons displayed for their respective APL systems. See below for an example:

A screenshot of a Windows File Explorer window. The address bar shows the path 'C:\Users\user\Ebaelen'. The search bar contains the text 'Search Ebaelen'. The file list has two columns: 'Name' and 'Date modified'. The 'Name' column has a checkmark icon on the left and a dropdown arrow on the right. The 'Date modified' column has a dropdown arrow on the left. The file list contains four items: 'FILE2NOTEPAD.w3' (Word document icon), 'File2Notepad.ws64' (Word document icon), 'File2Notepad.zip' (ZIP file icon), and 'my file.txt' (Text document icon). The 'FILE2NOTEPAD.w3' file is selected. The 'Date modified' column shows the following dates and times: '5/11/2021 11:29 PM' for 'FILE2NOTEPAD.w3', '5/11/2021 11:22 PM' for 'File2Notepad.ws64', '5/11/2021 11:32 PM' for 'File2Notepad.zip', and '5/11/2021 11:29 PM' for 'my file.txt'.

Please provide additional details if this is something else entirely.

User Commands in APL64

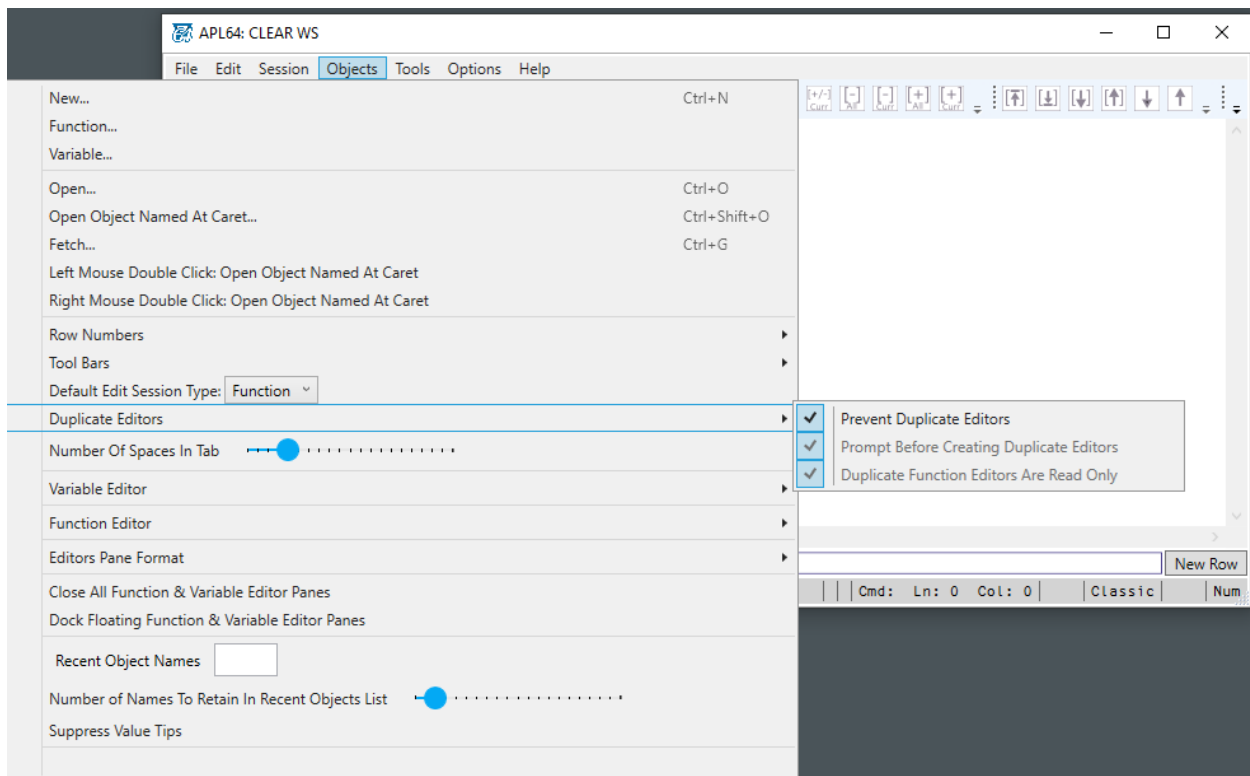
How can I use UCMD with APL64?

Available in the next pre-production version update.

Duplicate Editors Menu

In Objects > Duplicate Editors menu, if Prevent Duplicate Editors is checked then aren't the other 2 choices moot?

In the next available APL64 pre-production version If 'Prevent Duplicate Editors' is checked, the 'Prompt Before Creating Duplicate Editors' and 'Duplicate Function Editors Are Read Only' options will be disabled.



Variable Editor Toolbars

When **Objects > Toolbars > Display Toolbar in Variable Editor** menu is unchecked, how is anyone to find out that CTL + E, CTL + S, CTL + Q shortcuts exists?

Once a user is familiar with the options of the variable editor, the user may uncheck the **Objects > Toolbars > Display Toolbar** if desired. This is analogous to learning GUI options via tooltips and then when experience is gained, the user can turn them off.

Variable Editor: Inserting/Deleting Rows

Where are the clues for inserting/deleting rows from a numeric edit session?

The inserting/deleting of rows in the APL64 object editor is under development and will be available in a future production release of APL64.

Variable Editor: Line Numbers

Inconsistent line numbering starts with 0 everywhere except when editing numeric arrays.

A future APL64 version update will address the issue of inconsistent line numbering in editors.

Preferences Menu

Shouldn't most options in the Session menu be in a menu named Preferences?

Not exactly. Many of the options in the APL64 menu items apply to the current session and future sessions.

Most Recent Object Names

"Objects > Most Recent Object Names" opens an edit session when it does not exist in the active workspace

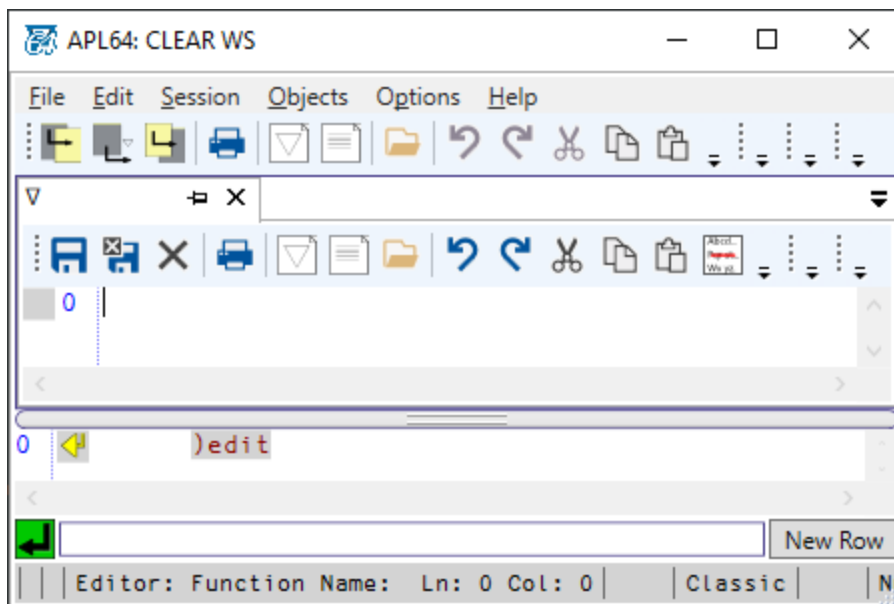
This is by design for consistency with APL+Win. Other than your comment, there have not been any negative comments related to this feature of APL. Please read the documentation: Help > APL64 Developer Version GUI > Instantiating an Object Editor.

)EDIT System Command

)edit in APL64 opens the editor and from the line numbering style it thinks I want a literal array:

- how does it know? I have Objects > Default Edit Session Type set to Function.
- on pressing CTL+E it closes the session. What happened to what I edited?
- ☐edit does not exist in APL+Win, With APL64, it fails.

If you have Objects > Default Edit Session Type > Function selected and you execute)edit with no right argument, the APL64 Developer version GUI will instantiate a function editor. Since you did not provide a right argument, there is no function header until you provide one. The ∇/# in the tab of an editor pane indicates that the object being edited is a function/variable.



In APL+Win only homogeneous variables, rank<2 with text or numeric elements can be edited. If APL+Win reports the exception 'Unable to edit object', that object falls outside the editable domain of the APL+Win)Edit system command.

The APL64 object editor does not have these limitations. The APL64 variable editor can edit any APL64 object except those which are zilde or contain enclose zilde.

In APL64 to see the options and features of)edit and ☐EDIT, please read the available documentation: Help > Developer Version GUI > Instantiating an Object Editor. Both)edit and ☐EDIT are operating according to specification and are subject to future enhancement.

[\)SAVE System Command](#)

The)Save system command failed to a mapped drive failed with the error message:

WS FILE ERROR: Could not save workspace : Could not find a part of the path 'o:\temp'.

There is no problem with APL64. What we've deduced is that this specific issue can arise when APL64 is run as Administrator and the mapping of the drives were done in a user mode instead of Administrator mode. So, the solution is to either run APL64 without Administrator mode or map the drives in Administrator mode.

[Documentation Window: Search](#)

There is no way to search in the Contents pane - all searches apply to the whole document. The behavior you note is by design of a pdf-format document. The pdf-format document technology was selected for APL64 documentation because its format and features is universally known worldwide.

[APL+Win Development](#)

Is the development of APL+Win being frozen and APL64 being seen as a superset product that includes APL+Win? Else, if APL+Win will continue to be maintained /enhanced, are you not creating the burden both products are released in parallel when APL+Win is changed?

The future of APL+Win and APL64 will depend on the adoption of APL64 by the customer base. As noted in the APL64 Project forum documents, the ability to enhance APL+Win is limited because it is based on technology which has been deprecated by Microsoft. Bug fixes, if any, create a potential for future APL+Win updates. You can also view these documents from APL64: Help > APL64 Project History.

[Text Size in Session](#)

I send up with multiple sizes on my screen at the same time

This issue will be addressed in the next APL64 pre-product version.

Find/Replace Dialogue

How do I do a replace [operation] in Find/Replace dialog?

The Ctrl+R keyboard shortcut has been deprecated in APL64. So, use Ctrl+H to open the Replace dialogue.

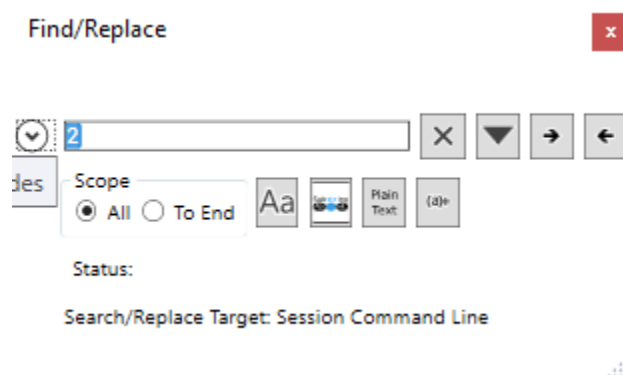
Fyi. The ability to replace will depend on the current focused pane in the APL64 Developer version. For example:

- (a) If the focus is in the session history, replace is not permitted because by design in APL64 the session history is not editable to maintain it as an accurate representation of what occurred during a particular APL64 Developer version instance.
- (b) If the focus is in the currently debugged function or the state indicator panes, replace is not permitted.
- (c) If the focus is in a function editor, replace is permitted
- (d) If the focus is in a variable editor plane of an APL64 variable which represents homogeneous text of rank<3, replace is permitted
- (e) If the focus is in the command line, replace is permitted

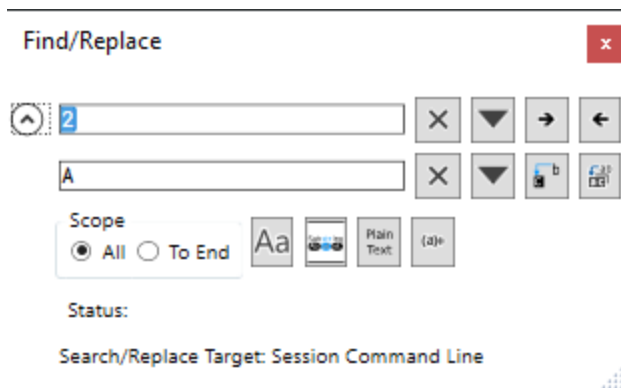
Note: The title of the Find/Replace dialogue will change indicating if Find or Find/Replace is permitted in a future APL64 version update.

Where replace is permitted, the 'expander' GUI control may be used to expose the replace input control. The expander control is a circular button containing V/^ depending on its collapsed/expanded state. The expander has a tooltip which explains its behavior.

Focus in command line, Find/Replace dialogue with expander collapsed:



Focus in command line, Find/Replace dialogue with expander expanded:



Find All

Is there a Find All in the Find dialogue, which just highlights the matches?

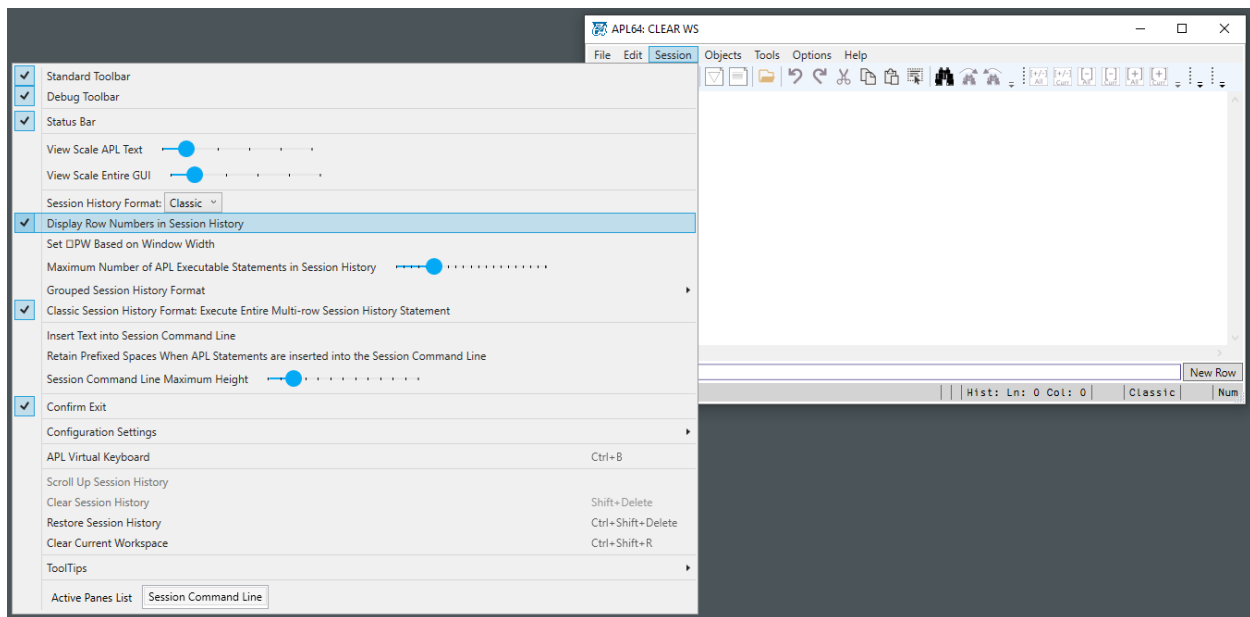
In the next available pre-production version of APL64, when the Find action is performed all matches in the document with the keyboard focus will be highlighted. The highlighting will vary depending on the Session > Session History Format which has been user-selected.



Row Numbers in Session History

Under Objects > Row Numbers menu, consider adding the session as a choice

This option is already available in the Session > Display Row Numbers in Session History menu item. The session history is not an object, so the menu item should not be in the Objects menu.



File > Copy Dialogue

When using the Copy dialogue from the menu, how do you select specific object for copy?

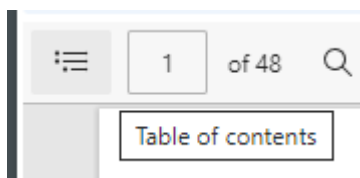
This action is not supported from the Copy dialogue. But it is being considered as a potential enhancement to APL64.

Documentation Window: Back Button

There isn't a back button to return to the TOC page in Document viewer

The APL64 documentation window presents pdf-format documents. When pdf-format documents are presented, the GUI options are generally based on the pdf-document standards initially implemented by Adobe. These options are generally beyond the control of APL64. Here are some options available to return to the table of content:

- (a) Use the Ctrl+Home short-cut to move the focus to the start of the pdf-format document which is usually the location for the table of contents
- (b) Assuming the user-selected documentation window width is sufficiently large, click on the table of content button to the left of the current page# in the pdf-format document viewer toolbar.



Additional Information:

The back/forward buttons on the APL64 documentation window will change the visible content only when the document URL would change, which is consistent with the behavior of current browser. For example:

(a) Displaying only one specific pdf-document in a browser and scrolling within that document does not constitute a url change, so the browser's forward/back buttons will have no effect on the current content of the browser.

(b) Displaying several different pdf-documents in a browser and using the browser's back/forward buttons will cause the browser content to change because the document URLs are different.

There is an option in the next APL64 pre-production version to present the APL64 documentation using an independent application the user has selected as the default PDF viewer. This option is available in Help > Documentation Options > Use Default PDF Viewer.

First Executed APL Function

First execution of a newly defined function takes some extra time, the next ones show "normal" performance

The scenario you describe is to be expected. A similar scenario occurs in APL+Win for a newly-defined function. The extent of the difference in execution time for the first execution and subsequent executions will depend on the user provided hardware and software environment.

Documentation Window: Topic History

APL64 Document viewer should have an option to rearrange where the topic history area is located

The position of the table of contents of a pdf-format document in the browser window is determined by the browser in use and not APL64. An expander button has been added to the documentation window so that the user may hide/show the documentation history list. This enhancement will be available in the next APL64 pre-production version.

Documentation Window: Title

The APL64 Documentation viewer window needs to display the help topic

This enhancement to the APL64 documentation viewer has been implemented so that the title of the window will reflect the current documentation file name. This will appear in the next APL64 version update.

Documentation Window:

The APL64 Documentation viewer window is missing the minimize and maximize icons

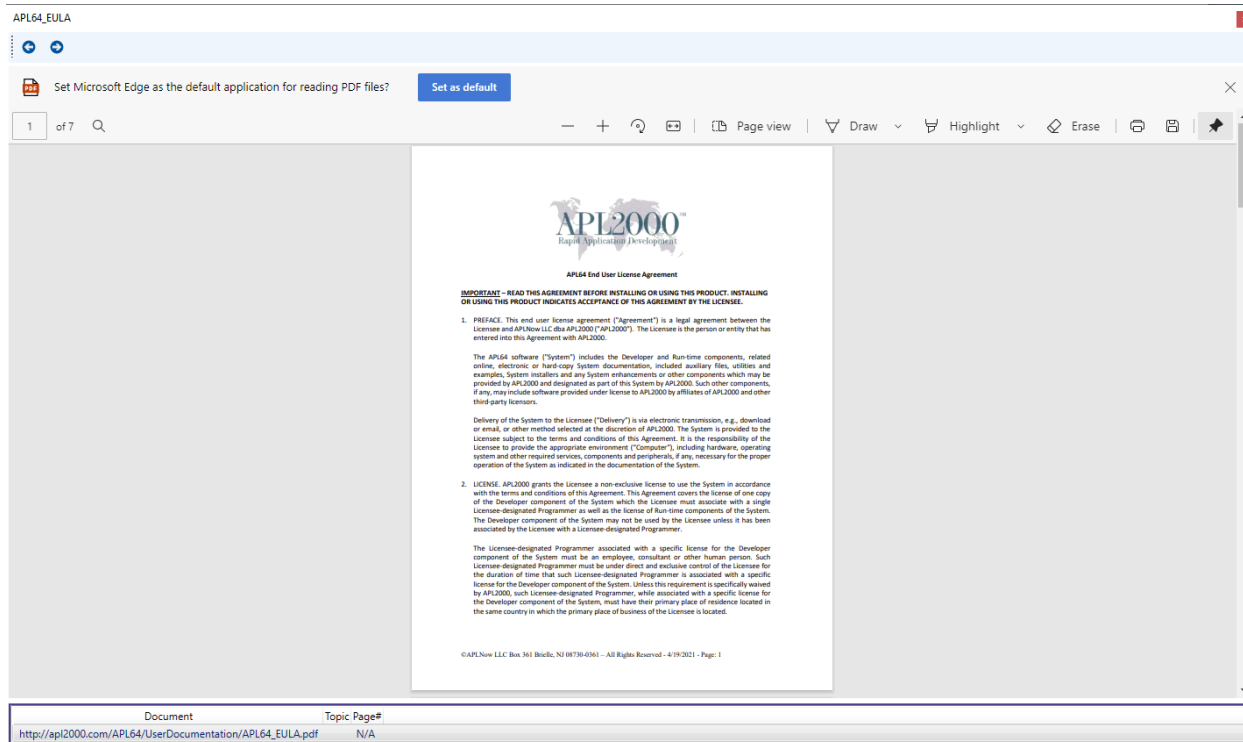
This is by design, so that the documentation can be viewed simultaneously while the APL64 programmer enters and executes APL statements. The APL64 documentation window has been implemented as a 'tool' window, so it does not have the minimize/maximize buttons. In a Windows desktop application the 'tool' window type is used when a window is non-modal like the documentation window, the find/replace window, the open object selector window, etc. Using the 'tool' window type is an indication to Microsoft Windows user that the window is non-modal.

Documentation Window: Navigation Toolbar

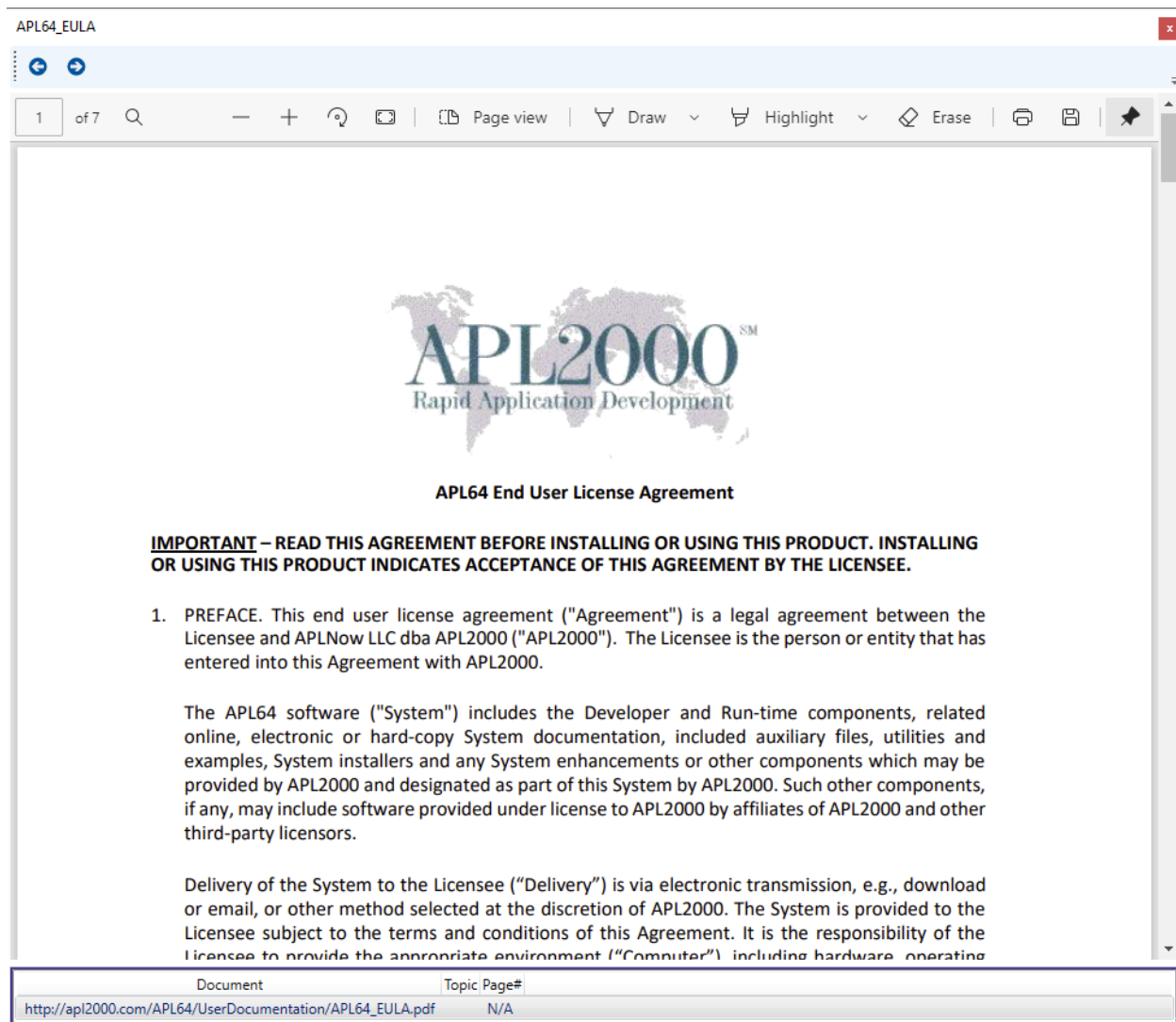
There is no toolbar for navigation [in APL64 Document viewer] and when the Table Of Content is missing

The APL64 documentation viewer uses an instance of the Chromium browser engine. The behavior and features of this engine depend on the user's selected browser environment, permissions, security features and hardware on the target workstation.

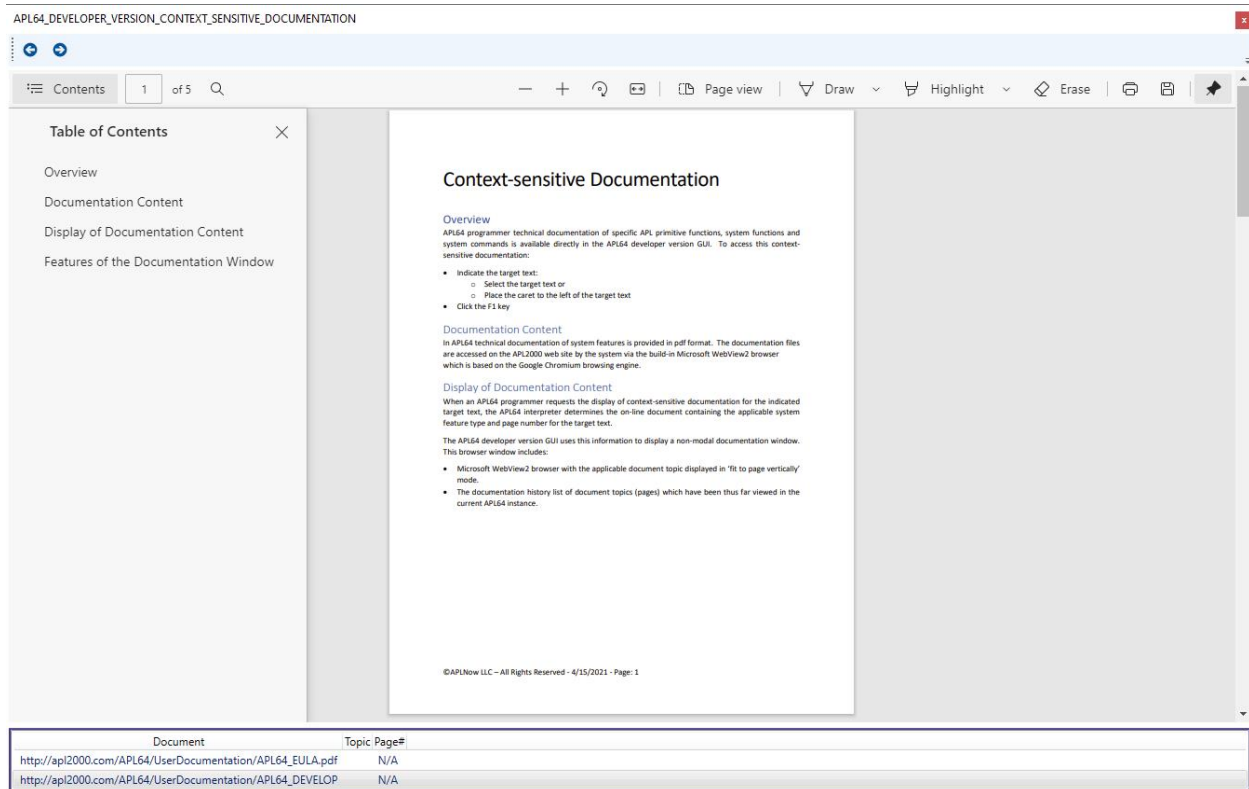
When first used, there may be a prompt which queries the user for the default browser for viewing pdf-format files. In some cases, administrator permission is required to change this setting.



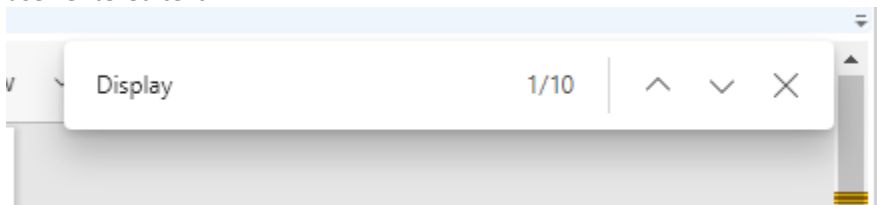
When the Microsoft Edge browser is the workstation user's default for viewing pdf format files, there are buttons for navigation, searching, printing, saving, highlighting, line drawing, table of content, etc. The capabilities of other user-selected browsers for pdf format files may be different.



When time permits, please review the APL64 > Help > Developer Version GUI > Developer Version Context-Sensitive Documentation information:



When the focus is on the APL64 documentation browser window, Ctrl+F will initiate a document search for user-entered text:



If you do not see these features in your environment:

- For some user-provided browser environments, try double clicking the help topic in the help topics list to refresh the browser window to see the tool bar, or
- Possibly the browser or workstation environment you have provided to APL64 is preventing these features in the APL64 documentation window.
-

LOAD System Command Exception

LOAD of an APL+Win workspace reported error message beginning with "WS FILE ERROR: Could not load workspace: Could not load file or assembly 'System.IO.Pipes,..."

Unfortunately, we have not been able to reproduce your issue here. You did install APL64 with all the pre-requisites in Windows 10 and Windows 10 is current with the latest updates?

If you can consistently reproduce this issue, see if the same happens in a new instance of APL64 with both APL+Win and APL64 workspaces in the same folder.

CSE LoadAssembly Method

What is the difference between the CSE in APL+Win and the CSE in APL64 as regards the LoadAssembly method?

- Internally, are they both using Assembly.LoadFile?
- Internally, are they both using Assembly.LoadFrom? My guess is that the APL+Win CSE is using LoadFrom (loads dependencies) and the APL64 CSE is using LoadFile (does NOT load dependencies).

In APL+Win, the CSE is an external process, i.e. the .Net side of the CSE exists in a separate application domain from the APL+Win application domain. The APL+Win uses LoadFrom, which locks the assembly until the application domain of the .Net side of the CSE is unloaded. In APL+Win this happens when the CSE is closed and the external CSE process is terminated.

In APL64 the CSE is not an external process, i.e. the .Net CSE runs in the same .Net application domain as APL64, providing significant performance benefits. To prevent locking an assembly, the APL64 CSE has its own AssemblyLoadContext. The AssemblyLoadContext loads the assembly using LoadFromStream, and tries to add references by using the MetadataReferenceProperties of the assembly the user wants to load.

However, in APL64 there is a list of forbidden assemblies that get removed from the references (any assembly with System.Windows in the name). When the developer tries to load an assembly with System.Windows, the loads fail. However, if only a reference to the assembly the developer wants to load contains System.Windows, the load is reported as successful but the reference is not added.

For cross-platform compatibility, the APL64 ☐ CSE is implemented using the Microsoft .Net Core Roslyn compiler. This compiler emits dynamic assemblies to execute C# statements via the APL64 ☐ CSE. Since the underlying assemblies are dynamic, the thread which would be used to interact with GUI controls is not accessible by subsequent C# statements executed by the APL64 ☐ CSE. This Microsoft design prevents the APL64 ☐ CSE from effectively using creating GUI components.

Therefore the APL64 ☐ CSE cannot load .Net assemblies which support GUI technology, such as WPF or System.Windows.Forms. These prohibited assemblies have the 'System.Windows' name prefix.

Separate Session History Format

In the Separated session format, output in the Results window that exceeds the width has a separate horizontal scroll bar attached to each and every line, rather than having one scroll bar to scroll the entire block.

The behavior observed is by design. The sequential, grouped and separated session history formats are row-style formats, so each row of the session history is a separate document container. When the user-set, available width of a pane is insufficient to contain the APL statement text, a horizontal scrollbar will be presented. Since each row is a separate container, each container will present a horizontal scroll bar.

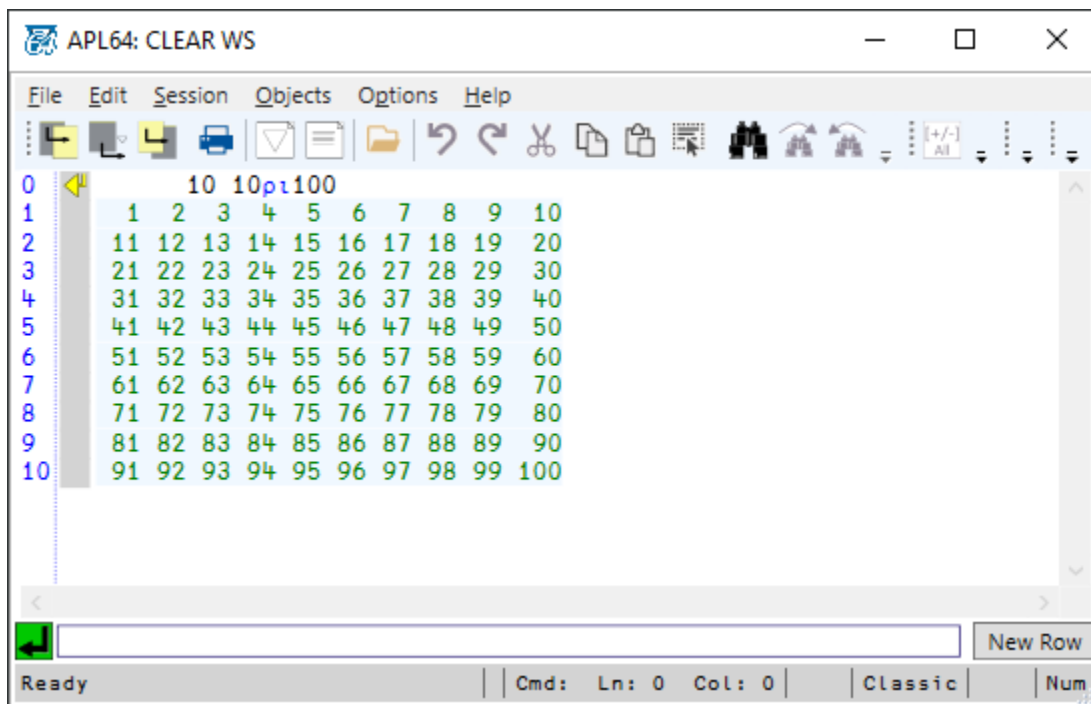
In many cases using the Session > Set ☐ PW Based on Window Width will cause the session history text to wrap and reduce the presentation of horizontal scroll bars.

The next APL64 pre-production version will include some improvements in the action of Session > Set ☐ PW Based on Window Width.

Scroll Session History Content

Might it be possible to allow the session to be scrolled with the mouse wheel?

When the mouse is hovered over the APL64 session history pane, or if the APL64 session history pane has the focus, the mouse scroll wheel may be used to scroll content of the session history pane. If all the output is contained in the available session history pane, the vertical scrollbar is disabled because there is nothing to scroll.



User-defined Tools

Are Tools available in APL+Win via the Tools menu missing in APL64?

We will investigate if it's possible to support the TOOLS facility in APL64 similar to that in APL+Win. So stay tuned for any announcement on this in the future.

Move Between Session History and Command Line

Is there was an easier way to end up in the command entry area when I first click back to APL64?

The APL64 session history pane contains output from APL executable statements and is not editable. APL executable statements must be user-entered into the command line.

The user-selected mouse position when clicked is used by the Windows operating system to put the focus in the GUI control containing the mouse position.

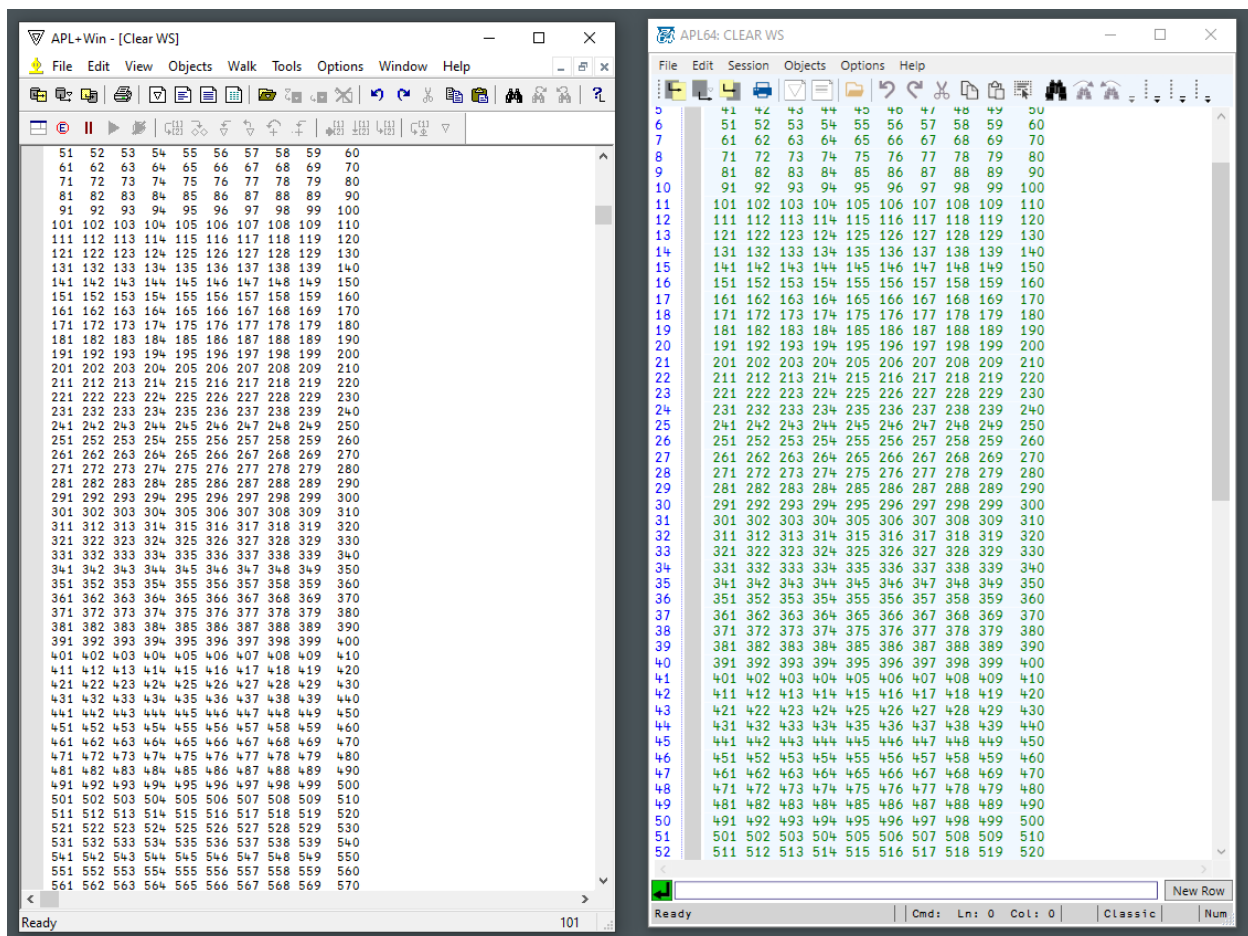
In the Windows environment Alt+Tab is an easy way to move the focus between independent applications without changing the focus relative to an application instance.

When the focus is in the APL64 session history pane, use Ctrl+End and then Key.Down (Down arrow) to put the focus into the APL64 command line.

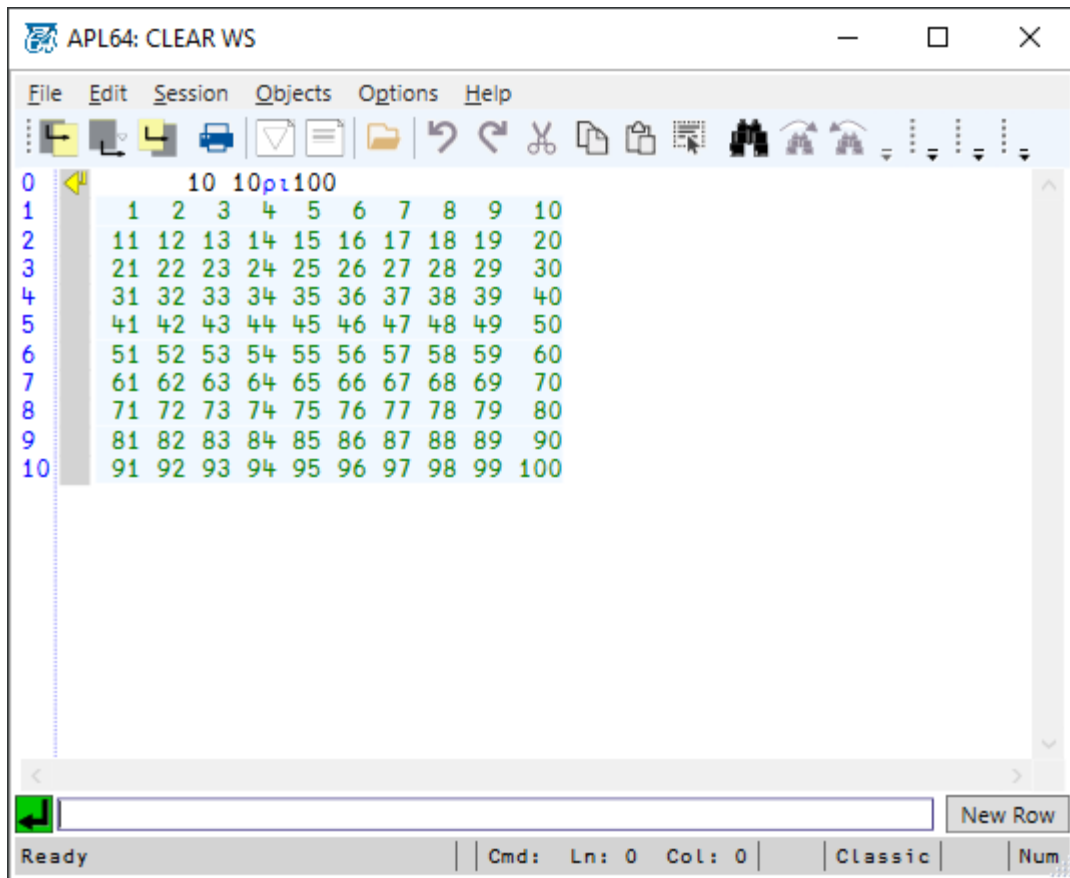
Scroll Up Session History

APL64 gives me no option to scroll up the session window by several lines and leave it there like in APL+Win

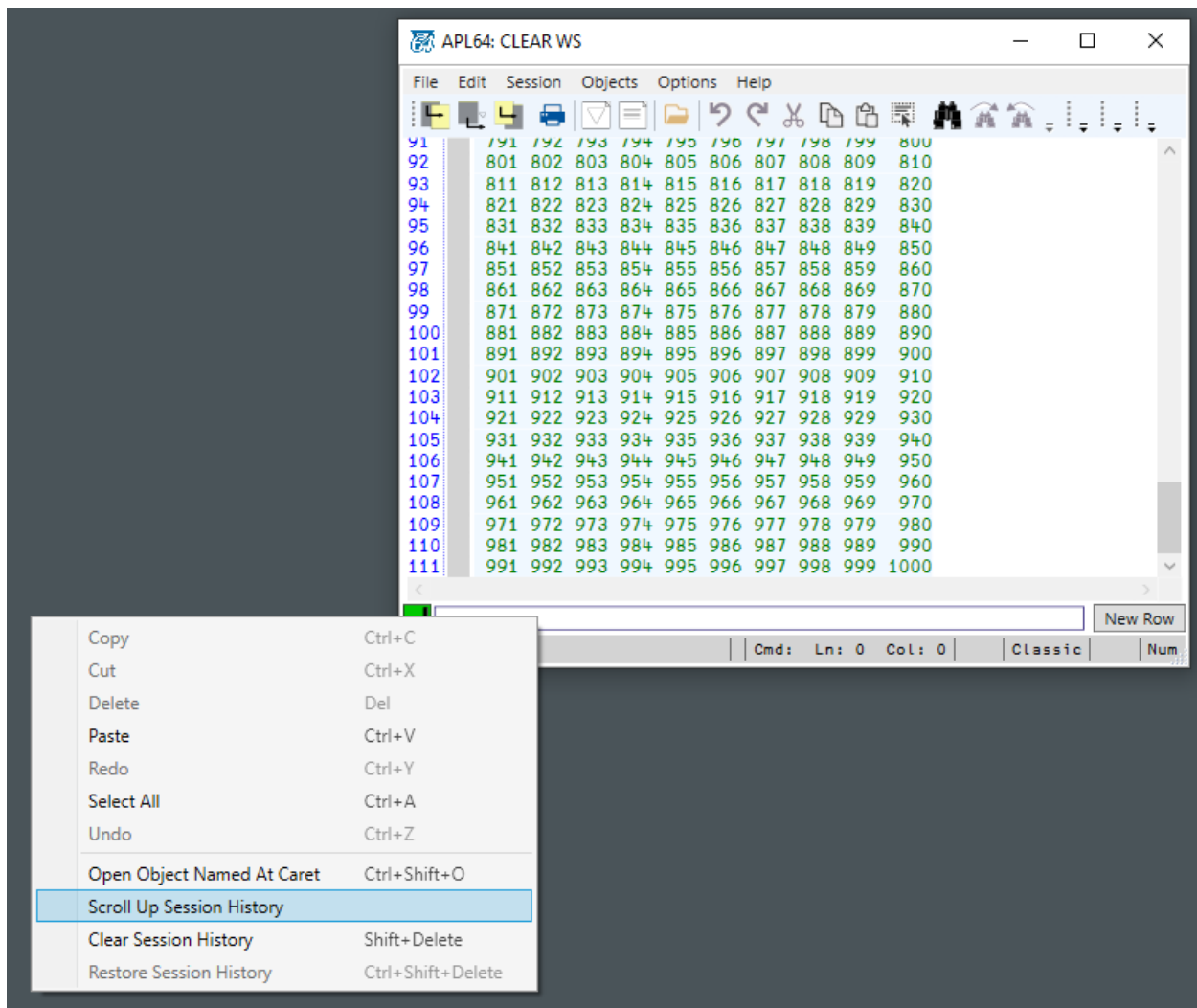
To use the mouse scroll wheel to scroll the session history pane content, hover the mouse over the session history pane and use the mouse scroll wheel. It is also possible to put the focus in the session history pane and use the mouse scroll wheel.



Note that when there is sufficient vertical space in the session history pane to contain all existing output, there is 'nothing to scroll' and the vertical scroll bar is disabled.



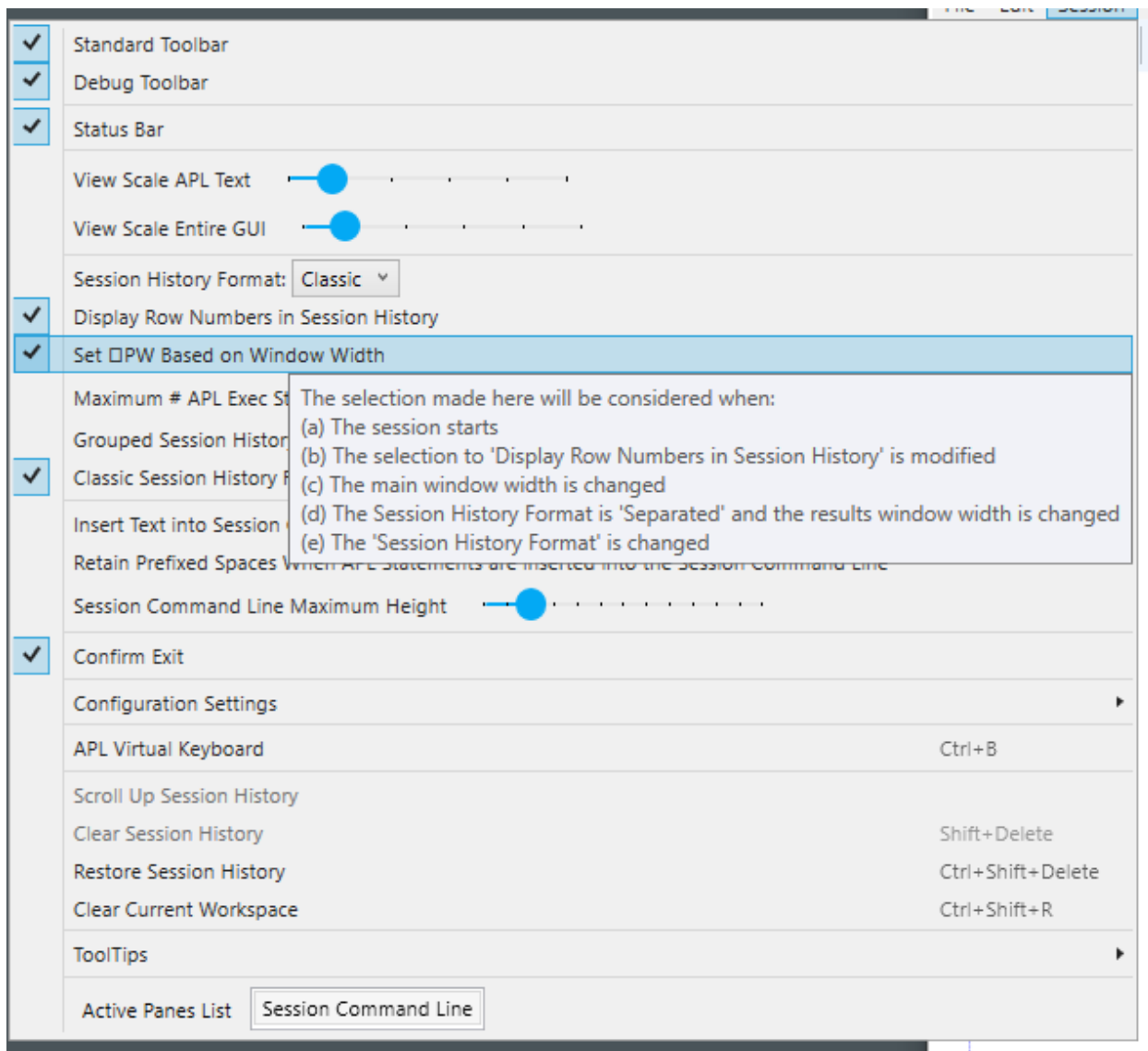
To 'fully scroll up' the APL64 session history pane content, use Session > Scroll-up Session History.



Session Font Size

Changing the session font size does not seem to affect the ☐PW setting

The issue has been corrected internally. In the next APL64 version update, when the menu option *Session > Set ☐PW Based on Window Width* is enabled, the value of ☐PW will consider the current session font size.



The Ctrl+Mouse Wheel only applies to the sizing of text in the editors in the pre-production release
Thank you for reporting this issue. The fix for this will be available in the next APL64 version update.

The size of the font in the State Indicator pane when debugging may appear larger than the other panes
in the session

This was a bug in the pre-production version 2020.0.1 that was corrected in the next pre-production version
update 2020.0.2.

Debugger Pane Toolbar

Why doesn't clicking on the Debugger Pane's icon toolbar and the tab title activate it?

Currently the debugged function must be selected to use the debugged function pane toolbar. A future APL64 pre-production version may change this behavior.

Copy Session History to Clipboard

Even after re-reading the instructions on highlighting session lines (for copying), I'm still having lots of difficulties with it. This needs improvement.

By design, there are differences in the selection behavior among the APL64 session history formats. Where possible other design ideas will certainly be considered.

Try this scenario in the sequential (row-style) session history format to observe how the context menu, selection and copy operate:

- (a) Execute several APL64 APL statements to develop some session history content.
- (b) Position the mouse in a specific sequential session history row.
- (d) Use the session history context menu and click 'Select all rows'.
- (d) Again use the session history context menu and click 'Copy'. The selection will transiently disappear.
- (e) Paste the selected rows to the APL64 command line or to an external application like Notepad.

Now try this scenario in the grouped (row-style) session history format. In this case you should observe that only the APL EXECUTABLE statements were selected and copied:

- (a) Execute several APL64 APL statements to develop some session history content with at least two APL executable statements each with some associated APL results statements.
- (b) Position the mouse in a specific APL EXECUTABLE statement row.
- (d) Use the session history context menu and click 'Select all rows'.
- (d) Again, use the session history context menu and click 'Copy'. The selection will transiently disappear.
- (e) Paste the selected rows to the APL64 command line or to an external application like Notepad.

Now try this scenario in the grouped (row-style) session history format. In this case you should observe that only the APL RESULT statements in that row were copied:

- (a) Execute several APL64 APL statements to develop some session history content with at least two APL executable statements each with some associated APL results statements.
- (b) Position the mouse in a specific APL RESULT statement row.

- (d) Use the session history context menu and click 'Select all rows'.
- (d) Again use the session history context menu and click 'Copy'. The selection will transiently disappear.
- (e) Paste the selected rows to the APL64 command line or to an external application like Notepad.

Session Panes

Session pane management is rather difficult in APL64

In the APL+Win Developer version GUI pane handling is based on technology which has been deprecated by Microsoft since the late 1990's. This technology was called Multiple Document Interface (MDI) and is supported only in the Win32 C++ GUI development environment. APL+Win panes cannot be floated beyond the MDI container.

In the APL64 Developer version GUI pane handling supports freely-floating panes and is based on the same technology as used by Microsoft Visual Studio and many other .Net application systems. This technology is commercially-available to a .Net programmer as the AvalonDock software for Windows Presentation Foundation (WPF) GUI development.

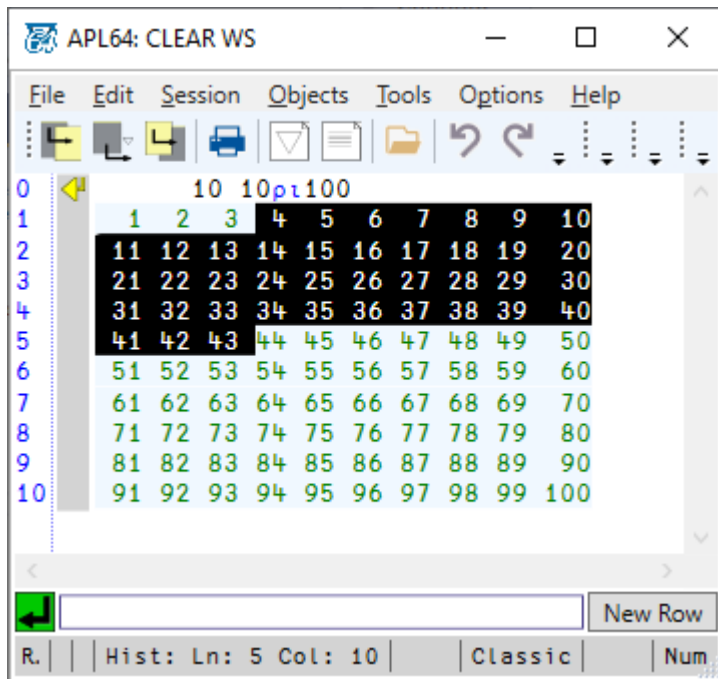
Search the Internet for "Using an AvalonDock GUI" and view an AvalonDock video such as: <https://www.youtube.com/watch?v=rKxzfdtdtRRl>. When time permits, an APL64 version of this video will be produced by the APL64 development team and included in the APL64 Developer version documentation.

Selection in the Grouped Session History Format

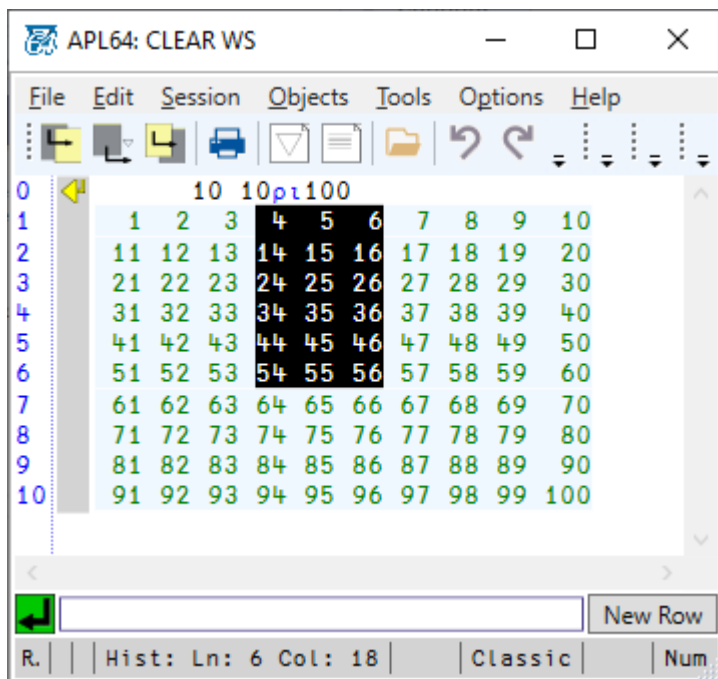
How am I supposed to highlight lines from the Grouped session so I can copy/paste them into an email to you?

The APL64 Help > Developer Version GUI > Session History Selection Options document provides the details of the differences in selection methods in the available session history formats. These differences are by design. There are two basic session history styles: document-based or row-based.

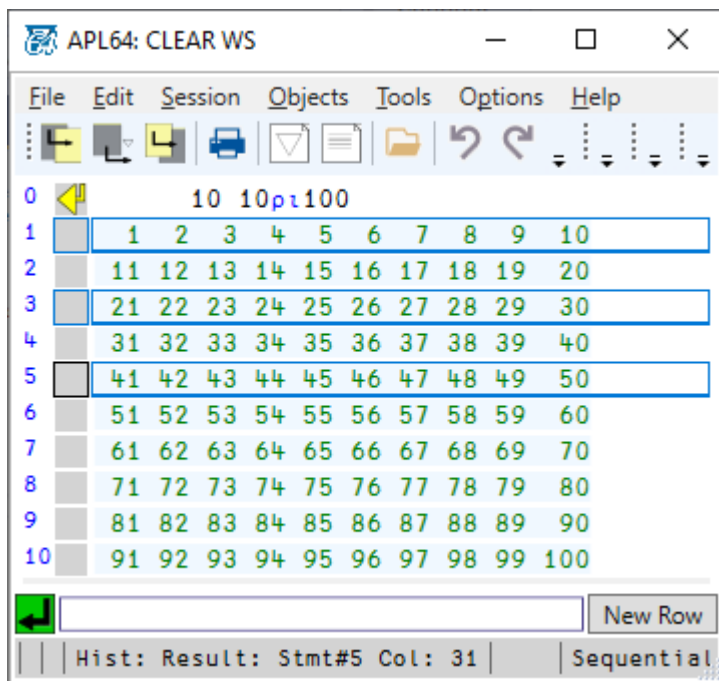
In APL+Win, only the document-based session history style is supported, so in APL64 the document-based session history style is called the classic session history format. In APL64, as in APL+Win, a linear selection, e.g. by 'dragging the mouse', is supported in the APL64 classic session history:



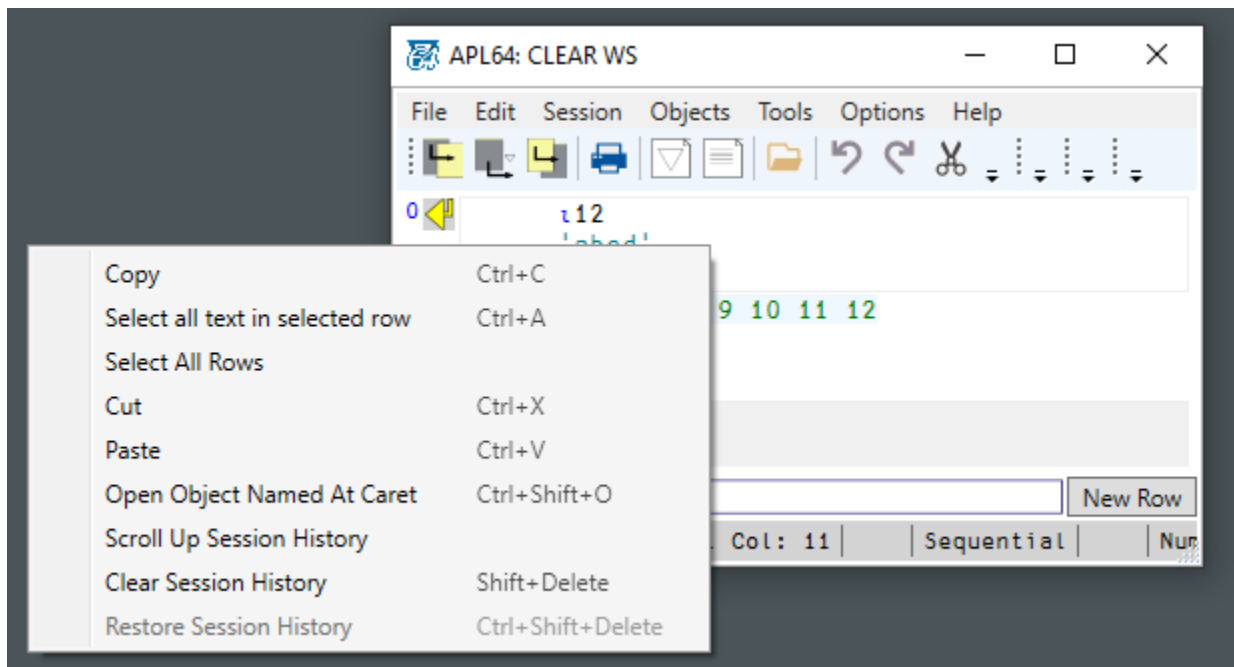
Exclusive to APL64 is a rectilinear selection in the classic session history:



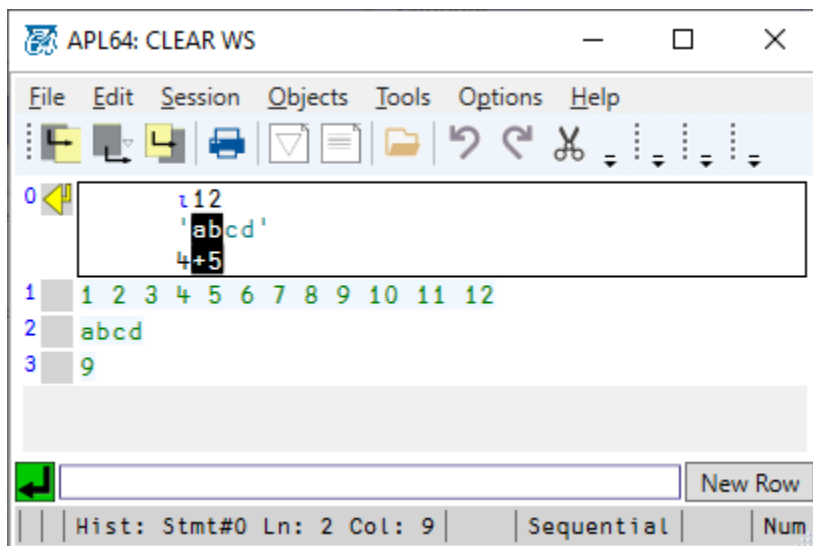
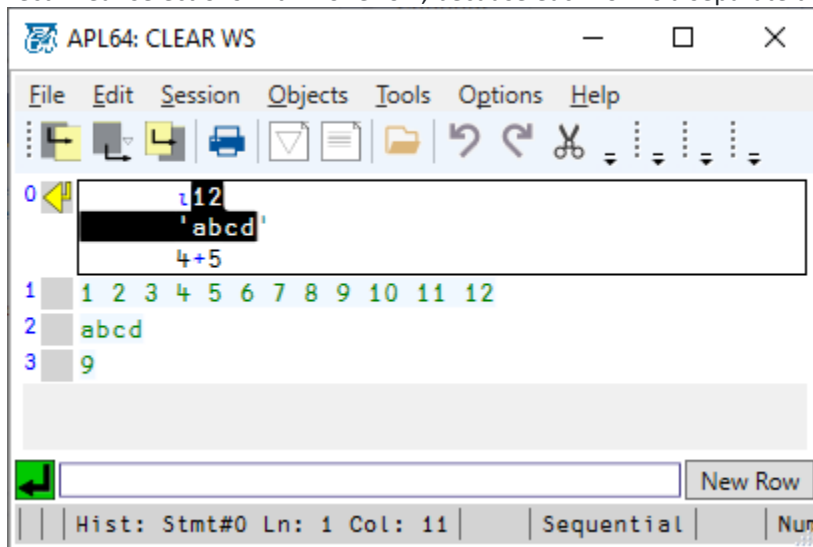
In the APL64 row-based session history formats (sequential, grouped and separated) the selection options are based on selected rows:



Assuming you have not blocked the new 'context menus' of APL64, the session history context menu for row-style session history formats provides additional selection options:



In the row-based session history formats, besides selecting entire rows, it is possible to create linear or rectilinear selections within one row, because each row is a separate document-style container:



Among the APL64 row-based session history formats there are additional variations in the selection options.

Font Size in Command Line

Could I request that the font size used in the command entry area be scaled to match that of the session history?

Thank you for observing this bug. It will be addressed in the next APL64 pre-production version.

Font Alternatives

Might there eventually be any options for changing the font in use in the session?

In areas of the APL64 Developer version where APL glyphs are presented, the APL385 font must be used. It is the only complete APL Unicode, fixed spacing (non-proportional) font available at this time.

APL64 APL Font in □WI

How do you use the APL64 font in □WI and runtime applications?

□WI is the same as □WI in APL+Win. □WI support in APL64 is based on the APLNow32.exe component of APL64 which is analogous to the APLW.exe component of APL+Win. The font(s) used by □WI GUI controls are specified by the □WI 'font' ... syntax and is not restricted by APL64.

The APL64 developer version uses two Unicode fonts, the default Windows font which is proportionally spaced and the APL385 font (APL385.TTF) which is a fixed spaced font. The default Windows font is used for GUI controls in the APL64 Developer version such as menu text, window titles, etc. It may be possible for a Windows user to change the default font. The APL385 font is used for GUI controls in the APL64 Developer version where APL glyphs will be displayed, e.g. the command line, the session history, etc. There is currently no alternative available for the APL385 font when a fixed-spaced, Unicode presentation containing APL glyphs is needed. The APL385 font is open sourced and available on-line.

If the runtime application is based on □WI, the first paragraph above provides the necessary information. If the runtime application is not □WI based, the ability to select the fonts will be based on the GUI tools used to develop that runtime application.

Sequential vs. Classic Session History Format

Can you tell me how the Sequential format is different than the Classic format?

Briefly, the sequential session history format is a row-style format and the classic session history format is a document-style format. This difference provides the opportunity for distinct options for display of information and selection of session history content.

Additional Coloring Options

In addition to setting foreground and background coloring for various text elements, what about the option to outline the area as well?

Thank you for your question. It is being considered as a future enhancement to APL64.

Scientific Notation

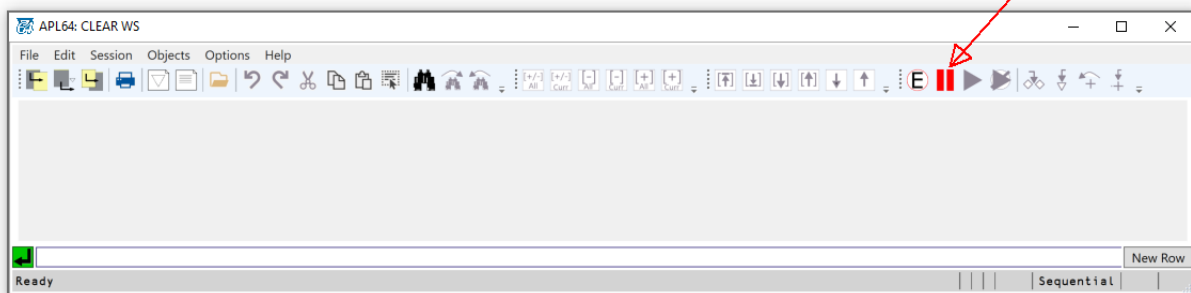
Executing 4.0002E10 in APL64 returned 4.0002E14

This was a bug in the pre-production version 2020.0.1 that was corrected in the next pre-production version update 2020.0.2.

Pause Execution

How do I interrupt a running process when my keyboard doesn't have key labeled Break?

You would normally use the Pause button on toolbar (double red vertical bars). See image below:



However, this sometimes works and sometimes doesn't. So, we're investigating it. But, you should try it in the interim.

Ctrl+Break Shortcut

APL64 does not respond to the Ctrl + Break shortcut key

The behavior of Ctrl+Break as well as the "Pause" button on the toolbar are planned but not yet completely functional at this time. We are aware of the issues you have discovered as well as other related issues and intend to resolve them in a subsequent pre-production build of the system.

Documentation Window: Go Back Option

Is there a way to go "back to the place you came from" in the online help Document Viewer?

The Chromium browser engine will respond to the back and forward actions (circular left and right arrow buttons) only when the pdf-format document changes, but not when the viewed location in the same document changes. This also appears to be the way that other browsers act upon pdf-format documents.

CSE Dynamic Exception

The CSE example below reports error message: "APL64_Message: (1,1): error CS8000: This language feature ('dynamic') is not yet implemented in Roslyn"

```
▽ cse
[1] □cself←'c' □cse 'Init' 'System'
[2] □cse 'ExecStmt' 'using System;'
[3] □cse 'ExecStmt' 'dynamic myVAR = 0;'
▽
    cse
0
-1
    □cse 'GetLastError'
```

Message: (1,1): error CS8000: This language feature ('dynamic') is not yet implemented in Roslyn.

We weren't able to reproduce the issue.

Application System GUI Development

How to encourage APL-centric application developers to consider APL as a component and not the entire application so that cross-platform applications can use the benefits of APL?

The APL64 ☐CSE cannot be used to develop a GUI. The APL64 ☐CSE is implemented using Microsoft .Net Core so that it is cross-platform compatible. The Microsoft .Net Core design including the Roslyn .Net compiler prevents the APL64 ☐CSE from interacting with WPF or System.Windows.Forms GUI controls.

The APL64 ☐WI GUI tools cannot be used to develop a cross-platform compatible GUI because they are based on Microsoft-deprecated, Windows-only technology.

This situation is to be expected, considering information published by Microsoft about their future plans as well as new components and products they have developed or acquired. The role that Microsoft has outlined for its premier programming language, C#, is as the tool to support the calculation algorithm portion of an application. The other application components can be selected from a panoply of options, some provided by Microsoft, and used in application development via 'dependency injection' to create cross-platform applications. Numerous Nuget packages are available to supplement the basic features of C# to complete the application development framework.

Simply substitute APL64 for C# to understand that designing a cross-platform GUI requires fundamental design decisions on the part of the C#/APL64-application programmer:

1. The application must be designed with appropriate separation of GUI, calculation algorithm, output preparation and data persistence. Separating the GUI component will permit the application to remain viable when the inevitable changes in GUI technology occur.
2. Based on the target platform and the anticipated application user base, the application designer must choose one or more GUI tools to develop the application GUI.

A traditional APL programmer has had the benefit of a single APL-centric development environment which included support for GUI, calculation algorithm, output preparation and data persistence. There is no viable way to provide an all-inclusive APL environment which would provide cross-platform compatibility.

In a cross-platform design environment, APL64 can support elements of all application components except for the GUI component. This means that to develop a cross-platform compatible application the application programmer must incorporate development tools which include non-APL64 based tools for GUI development. Here are some of the GUI development tool choices, noting that these are all continuously undergoing substantial enhancement and modification and may be mutually incompatible:

HTML/javascript for a browser-based GUI

ASP .Net for C# syntax which compiles to javascript for a browser-based GUI

Xamarin which compiles to native Windows, IOS and Android GUI

WPF and System.Windows.Forms for a Windows-only environment

Blazor for browser-based applications where all the code (GUI, C#/APL64) run in the browser

See also:

https://en.wikipedia.org/wiki/List_of_platform-independent_GUI_libraries

<https://visualstudiomagazine.com/articles/2017/01/25/26-web-desktop-ui-tools.aspx>

This question and exact answer have been previously provided.

□WCALL & □NA

Is APLFrame applicable in APL64?

The APLFrame user defined function in APL+Win doesn't work in APL64 because

'#' □wi 'hwndmain' returns the handle of the APL+Win main window for the development system but the APL+Win runtime system that doesn't have a main window is in APL64, hence it always returns the value 0.

Where does APL64 store □WCALL & □NA definitions?

Definitions for □NA functions are stored in the APL64 workspace. The first time you invoke a □NA function with a 32-bit (DLL) binding, the APL64 system instantiates that definition in the APLNow32 execution context and uses it to proxy the invocation in the 32-bit context subsequently.

Calls made via □WCALL are defined in the APLNow32 ADF file or INI file (just like in APL+Win). They may also be dynamically and temporarily defined via the □WCALL "W_Def" mechanism. Again, this is just like in APL+Win. These definitions are not saved with the workspace like □NA definitions.

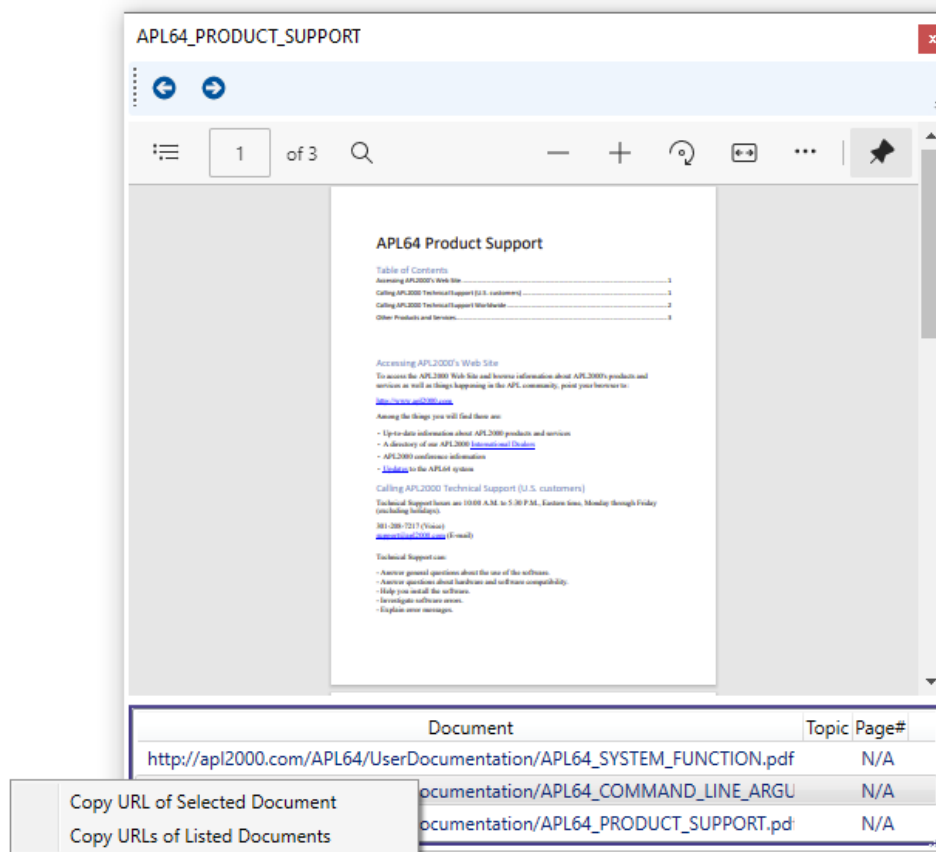
Is there a way to enumerate all the □wcall & □NA definitions that have been defined?

No, not at this time. The □NA mechanism in APL64 works similar to APL+Win which also did not support such an enumeration mechanism.

Links to APL64 Documentation

Since the APL64 documentation is hosted on Internet could you please send me the links to the various .pdf APL64 documentation files?

When a documentation topic is displayed in the APL64 Developer version documentation window, a right click on the topic in the list will display a context menu with an option to copy the documentation topic link(s) to the Windows Clipboard:



Documentation Windows Location

Can the APL64 Documentation viewer not always display on top of the session?

If the APL64 Developer Version GUI main window is presented full screen, there is no other place to display the APL64 documentation window except over the main window. If you want to share the available screen area with multiple windows:

(a) Change the presentation of the APL64 Developer Version GUI main window to windowed (sizable) and adjust its dimensions to the desired area.

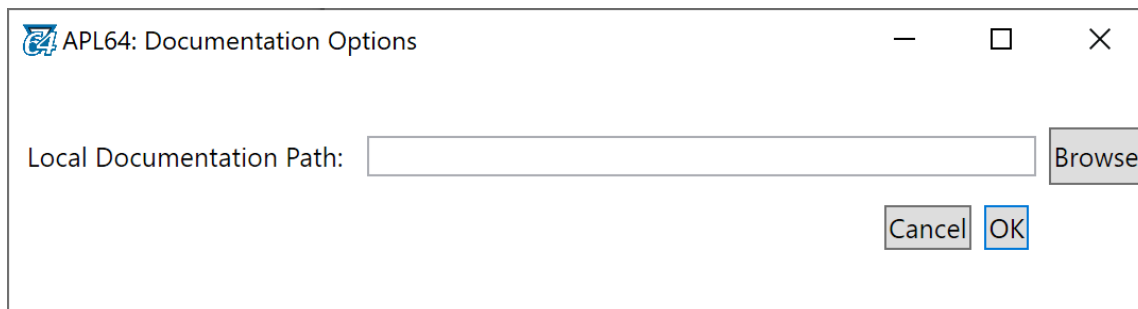
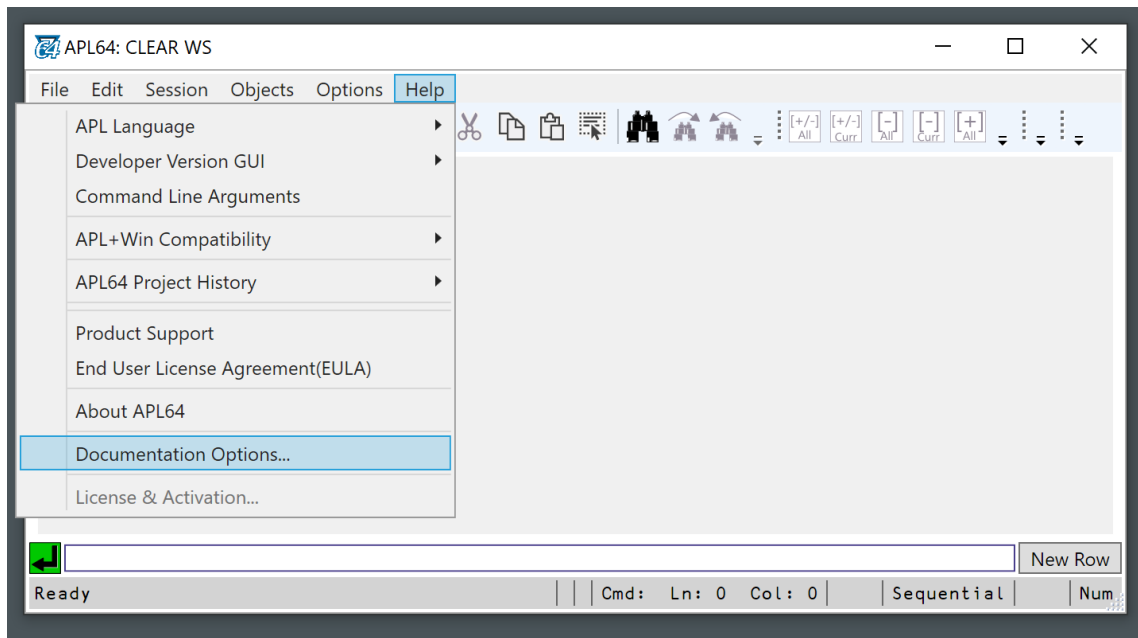
(b) When the APL64 documentation window is presented, adjust its position and size so that it does not overlap the APL64 Developer GUI main window.

(c) Use Session > Configuration Settings > Save Settings Now to preserve this layout for future APL64 Developer Version instances.

Off-line APL64 Documentation

Can the APL64 programmer view the documentation off-line?

An APL64 programmer can view the documentation off-line. When a documentation file is displayed in the APL64 documentation window, use the 'Save' button on the toolbar to save the file to the desired location. After the APL64 programmer has saved the desired documentation files locally, the APL64 documentation window can be pointed to the local folder using the Help > Documentation Options dialog. Using this option will fix the version of the documentation files to the last downloaded version and subsequent documentation updates would not be reflected until the user re-saves the documentation files.



Session History Source of Executed Statement

It's not possible to re-execute a previous error statement line with [imm] in the front

The [imm] is the 'source' element of the result of an APL64 executable statement. It may not be all that important in simple, single line, statements. But it could be when errors occur in multi-line statements or multi-statement groups of lines. With that said, we will consider implementing an option that will suppress the [imm] prefix from error reports.

XLIB

 XLIB 'test file.sf' reported "SYSTEM ERROR:Exception: The parameter is incorrect." Instead of XFHOST ERROR: CreateNew 1010 0 0 The file 'test file.sf' already exists.

Thank you for reporting this bug in the  xlib system function. It has been fixed and the fix will be included in the next update.

Download APL64 Documentation

Is it possible to download all the online documentation at once?

The APL64 developer version documentation is on-line, so that it is always up-to-date. Copying the documentation locally will inevitably result in it becoming out of date. It is not anticipated that an option to download all the documentation at once will be provided.

APL+Win Colors in APL64

Can I move my APLW.INI file to APL64 and have its color settings applied?

The APL64 > Options > Colors is operating according to specifications. By design the default colors cannot be modified by an APL64 programmer. Instead, the APL64 programmer can create any number of 'color sets'. From the Colors dialogue, click the Clone button use other elements of the dialogue to create your desired color set. Multiple color sets can be created and selected for a particular purpose.

For your information, we are exploring whether it is feasible to migrate the current session color settings in APL+Win to APL64.

Color Dialogue

The new name should be selected when renaming a color set in the Colors dialogue

>> This change will appear in the next APL64 version update.

The Update Name button should be the default button when renaming a color set in the Colors dialogue

>> This change will appear in the next APL64 version update.

The Blank Area color element in the Colors dialogue has no effect in the session

>> This will be corrected in the next APL64 version update.

Oddity in the Colors dialogue when moving the focus away from APL64, a different dialogue can appear below the colors picker window

>> This will be corrected in the next APL64 version update.

)OFF System Command

Glad to see)off yes [still] works without the prompt

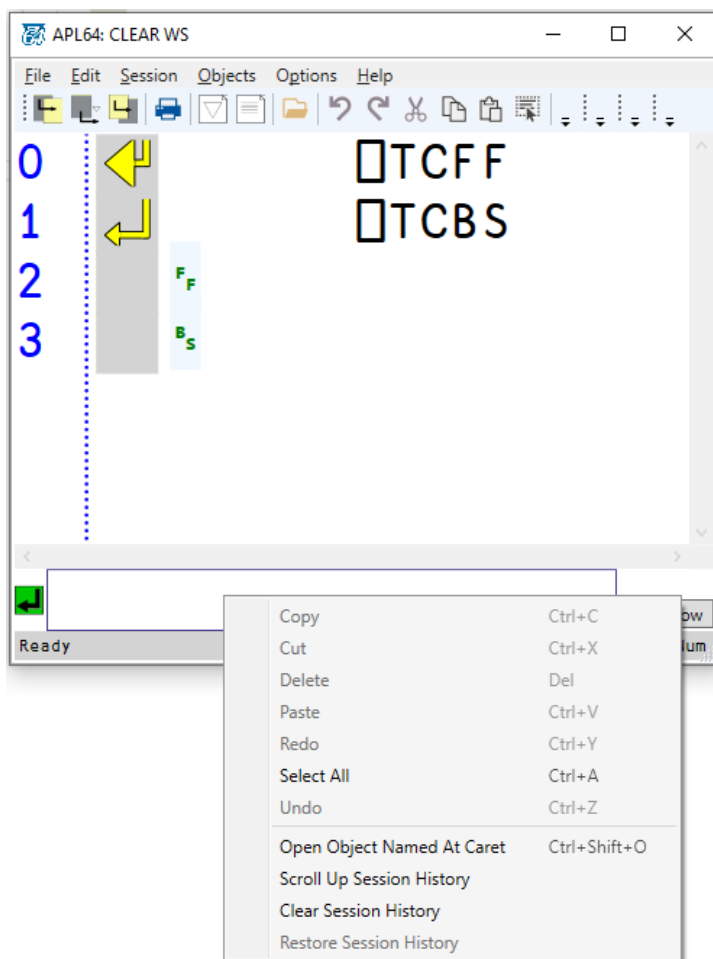
Clear Session History

I used to be able to clear the session screen with □TCFF, should that come back?

Recall that in the APL64 Developer version the session history is not editable except by executing an APL executable statement which generates output.

□TCFF is a Unicode character and not an action that generates output - in APL+Win it removes output. The same is true of □TCBS. In APL64 Unicode characters, except for □TCNL and □TCLF, are rendered to the session history using their associated glyphs.

In the APL64 Developer version, to simulate the APL+Win action of executing □TCFF, use Session > Scroll Up Session History. To clear the current session history use Session > Clear Session History. These actions are also available in the context menu of the session history or session command line.



Variable Editor: Insert/Delete Rows

Where are the clues for inserting/deleting rows from a numeric edit session?

The inserting/deleting of rows in the APL64 object editor is under development and will be available in a future production release of APL64.

Variable Editor: Line Numbers

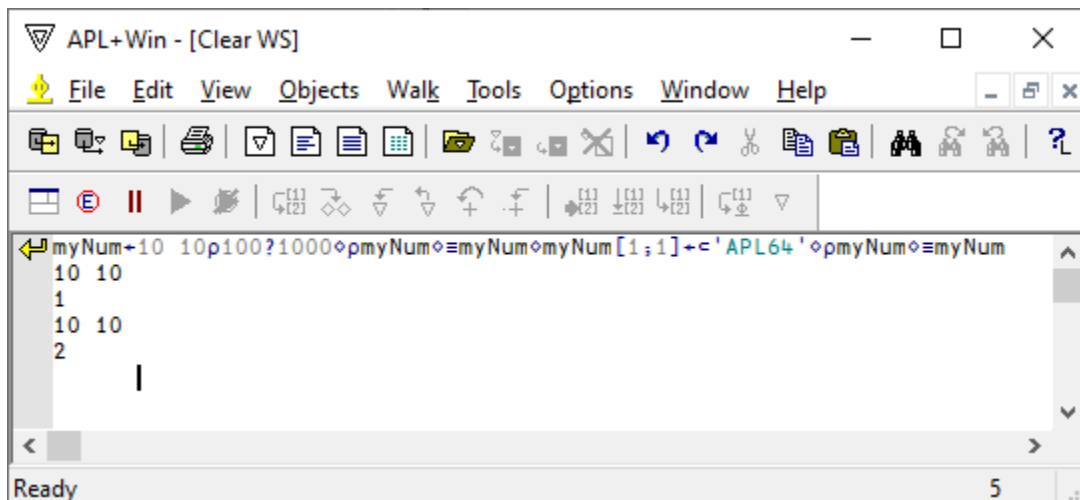
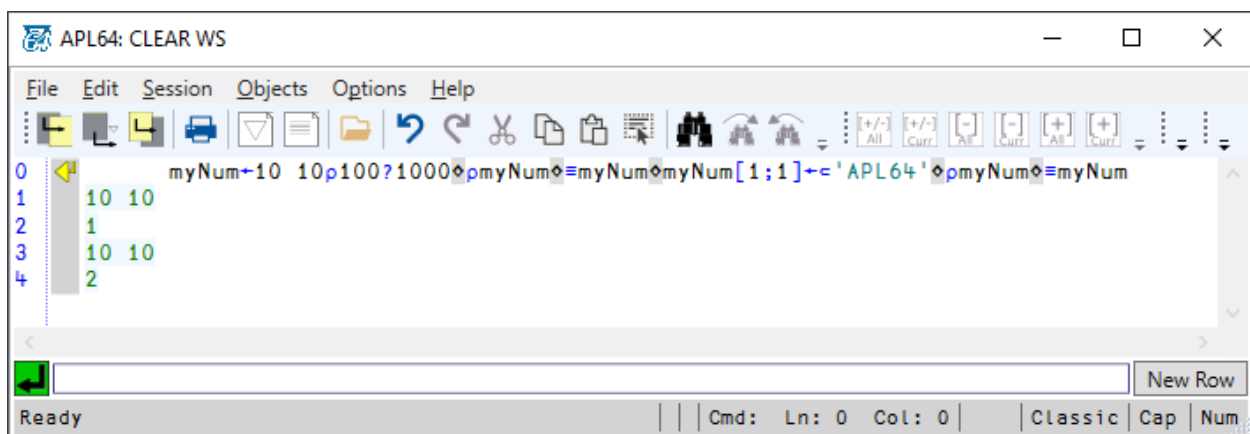
Line numbers show quite distinctly in a margin delimited by a vertical line except when editing numeric arrays

This is by design of the underlying GUI components used in APL64. Some APL64 objects, such as arrays of rank >2, arrays containing mixed data type elements, and pure numeric arrays, require a 'DataGrid' rather than a 'TextEditor' type of GUI control to present the data in the editor. These two GUI controls provide different methods to display and edit information, including line numbers.

Depth Primitive Function

Bug in depth primitive operator after inserting a text vector in a numeric array?

There is no error in APL64 and the issue is not related to using the APL64 variable editor. The APL64 unified variable editor handled the replacement of myNum[1;1] with a text vector just like an APL64 programmer would have manually done it. Observe the behavior of standard APL code in APL64 and APL+Win:



)SINL System Command

)sinl system command is not working

The)sinl system command currently reports the message: FIXME: FormatNameList() not done

It will be investigated and a fix will appear in a future APL64 version update.

ADF File

Do I need a new ADF to make calls to windows?

No. The `⎕WCALL` feature is fully operational in APL64. `⎕WCALL` actions may return results which are applicable to Win32 features of the Windows operating system. Such results may not be relevant to APL64 which runs as a .Net managed code application.

For example, as noted in the APL64 > Help > APL+Win Compatibility > APL64 Compatibility Modifications for APL+Win Applications.pdf:

`⎕wcall 'GetModuleFileName'` points to the APLNow32.exe component of APL64 and not the APL64 executable. Use `10⊃⎕SYSINIT` which returns the full path of the APL64 executable.

Invalid Function Header

How was this function Saved with Invalid header?

This is NOT a bug but rather just a misunderstanding regarding parsing rules. I will admit that SavedInvalidFn looks wrong:

```
⎕vr 'SavedInvalidFn'
▽ SavedInvalidFn⎕wself
[1]
[2]  A Header is invalid but but no error on CTL+E
[3]  110
▽
```

The intention may be to create a niladic function "SavedInvalidFn" with a local variable "`⎕wself`" but have forgotten to put a semicolon between the function name and locals list. However, that is NOT what's actually being declared by this function definition.

This is really a monadic function named "SavedInvalidFn" with a right-argument "`⎕wself`". Consider the following executed in APL+Win or APL64:

```
aaa←10
aaa⎕io
10 1
aaa ⎕io
10 1
```

The `aaa` and `⎕io` can be typed with or without a space between them. For better or worse, those are the parsing rules that apply to a user defined name contiguous with a quad name without spaces between them. This is historical and changing it would break existing code for some users.

This explains why we get a SYNTAX ERROR from the following:

```
      SavedInvalidFn
SYNTAX ERROR
      SavedInvalidFn
      ^
```

We are trying to invoke a monadic function without specifying an argument. But if we pass an argument that represents a valid `⌊wself` value such as `"` or `'` then it executes successfully:

```
      SavedInvalidFn ''
1 2 3 4 5 6 7 8 9 10
```

```
      SavedInvalidFn '#'
1 2 3 4 5 6 7 8 9 10
```

If we try passing in an argument that is not a currently defined `⌊WI` object name such as `"fmMain"` then we get an error when trying to set the `⌊wself` argument value:

```
      SavedInvalidFn 'fmMain'
⌊WI OBJECT ERROR: Object "fmMain" not found
      SavedInvalidFn 'fmMain'
      ^
```

`⌊sysinit`

The 10th element returned from `⌊sysinit` is a nested vector, not a character matrix

In the next available APL64 version, the documentation of the 10th element of the result of `⌊SYSINIT` will be corrected.

⇒ `⌊sysinit` returns NONCE ERROR

Thank you for reporting this issue. It will be investigated and a fix will appear in a future APL64 version update.

Status Bar

Session History Format appears in the status bar where APL+Win shows UNI when Enable Unicode Clipboard is on

Firstly, the Enable Unicode Clipboard option is deprecated in APL64 where Unicode is supported throughout the system. Secondly, APL64 displays the current session history format in the status bar. Its placement there should not be confusing to anyone familiar with APL+Win because APL+Win never displayed the APL keyboard selection in the status bar.

F1 Documentation Shortcut

The F1 key with a topic that does NOT exist under the cursor generates an Error 403:

A modification addressing this issue will be implemented in the next available APL64 version.

Component Files

Why is the maximum number of component files that can be created 10?

The same thing happens in APL+Win. With `⎕fcreate`, you can only create a classic component file that abides by the traditional representation (8.3) for file names. So here, appending '10' to the name 'newfile' exceeds the maximum 8 character limit for classic component file names.

The solution is to use `⎕xfcreate` or `⎕cfcreate`.

⎕Fcreate & ⎕XFcreate

`⎕Fcreate` & `⎕XFcreate` fail to add the .SF extension

The `⎕FCREATE` should append the .SF extension in APL64 just as it does in APL+Win. If this isn't happening for you, please send us a reproducible case. Unlike `⎕FCREATE`, `⎕XFCREATE` being an extended file function does not automatically append the .SF extension. Therefore, the extension must be added.

CSE Results

Results from CSE may not be the same in APL64 and APL+Win

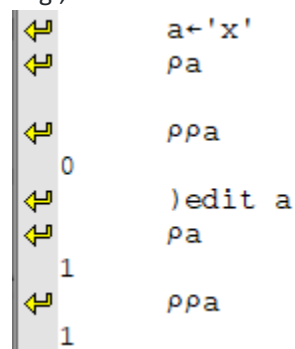
Recall that APL64 supports the .Net String data type and APL+Win does not. Your example requested String types from .Net, so the APL64 CSE GetValue method return String types in the result. The result is different, by design, compared to APL+Win. Be sure to read the documentation: Help > APL Language > C# Script Engine Manual > 'APL+Win CSE to APL64 CSE' section.

APL+Win Text Editor

Did APL64 uncover an issue in APL+Win's character editor?

Yes. APL+Win's character editor can unexpectedly convert a scalar 'x' to a vector , 'x' if you save the value without modifying it.

E.g.,



```
⎕←a←'x'  
⎕←ρa  
  
⎕←ρρa  
0  
  
⎕←)edit a  
⎕←ρa  
1  
  
⎕←ρρa  
1
```

Rotate Primitive Function

Bug in the rotate primitive function in APL64

E.g.,

```
SYSTEM ERROR:Aval.Categorize encountered an uncategorized child element at index=0
  at System.Environment.get_StackTrace()
  at APLNow.Data.SystemError.Message(Exception exception, String more, Object[] args)
  at APLNow.Data.Aval.Categorize()
  at APLNow.Data.Aval.IpDemote()
  at APLNow.Core.Code.EachBase.Exec(Apl apl)
  at APLNow.Core.Apl.Run()
  at APLNow.Core.Apl.Interpret(Who caller)
  at APLNow.Core.Apl.DoExecCall()
  at APLNow.Core.Apl.OnExec(Chars expr, Who who, ExStepFlags step)
  at APLNow.Core.AplExecTask.<>c__DisplayClass3_0.<.ctor>b__0()
  at APLNow.Core.Executor.ExecPending(Boolean wasWaiting)
  at APLNow.Core.Executor.Run()
ICS[3] Z←(€“(1⊕“Z,“(‘ ’,“”)~“” ’
      ^
```

This issue will be corrected in the next APL64 version update.

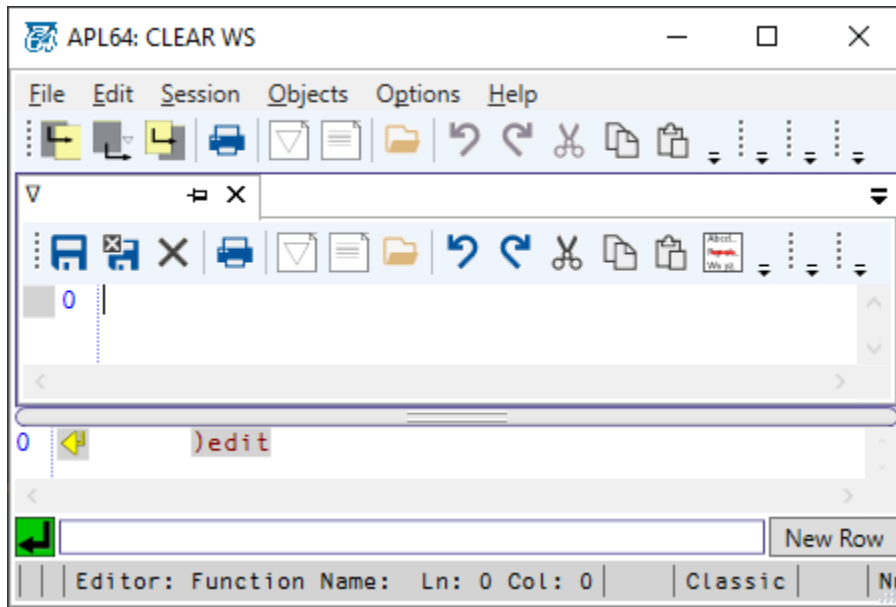
)edit & □edit

General comments and questions on)edit and □edit

)edit is a monadic system command that takes one argument. In APL+Win,)edit with no argument returns the message: Unable to edit object. Whereas, APL64 opens the editor (much like they both open the editor with ∇) and from the line numbering style it thinks I want a literal array:

- how does it know? I have Objects Default Edit Session Type set to Function.
- on pressing CTL+E it closes the session. What happened to what I edited?
- □edit does not exist in APL+Win, With APL64, it fails.

If you have Objects > Default Edit Session Type > Function selected and you execute)edit with no right argument, the APL64 Developer version GUI will instantiate a function editor. Since you did not provide a right argument, there is no function header until you provide one. The ∇/# in the tab of an editor pane indicates that the object being edited is a function/variable.



In APL+Win only homogeneous variables, rank<2 with text or numeric elements can be edited. If APL+Win reports the exception 'Unable to edit object', that object falls outside the editable domain of the APL+Win)Edit system command.

The APL64 object editor does not have these limitations. The APL64 variable editor can edit any APL64 object except those which are zilde or contain enclose zilde.

In APL64 to see the options and features of)edit and ☐EDIT, please read the available documentation: Help > Developer Version GUI > Instantiating an Object Editor. Both)edit and ☐EDIT are operating according to specification and are subject to future enhancement.

Documentation Window: Search

There is no way to search in Help's Contents pane - all searches apply to the whole document

The behavior you note is by design of a pdf-format document. The pdf-format document technology was selected for APL64 documentation because its format and features are universally known worldwide.

Most Recent Object Names

The "Objects > Most Recent Object Names" menu opens an edit session when it does not exist in the active workspace

This is by design for consistency with APL+Win. Other than your comment, there have not been any negative comments related to this feature of APL. Please read the documentation: Help > APL64 Developer Version GUI > Instantiating an Object Editor.

:CATCHIF

A bug in :CATCHIF statement reports the error "SYSTEM ERROR:Exception: Object reference not set to an instance of an object. at APLNow.Core.Apl.HandledByTryCatch()"

This issue will be addressed in the next APL64 version update.

Next Pre-Production Version

Questions about the next pre-release version

1 - Is the next pre-release version imminent?

2 - Will it include fixes for ☐wi & ☐cse?

There will be additional pre-production versions of APL64. It is anticipated that these versions will be made available to the current APL64 evaluators.

Based upon the excellent feedback received from the current APL64 evaluators, it is anticipated that future pre-production versions of APL64 will incorporate many enhancements and bug fixes.

A significant element of the APL64 development process is the creation of unit tests. The APL64 development team will certainly consider adding your test functions.

APL64 Effect on APL+Win

Effect APL64 has on the future of APL+Win?

The future of APL+Win and APL64 will depend on the adoption of APL64 by the customer base. As noted in the APL64 Project forum documents, the ability to enhance APL+Win is limited because it is based on technology which has been deprecated by Microsoft. Bug fixes, if any, create a potential for future APL+Win updates. You can also view these documents from APL64: Help > APL64 Project History.

Excel Interface

A customer application interfacing with Excel produced the error: ☐WI ERROR: 8000FFFF Catastrophic failure, followed by an Assertion failed!

This has been corrected and the fix will be available in the next APL64 Pre-Production version.

CSE Exception

I need clarification on the message: CSE ERROR: .Net data type, System.Int64, has no representation in APL64

The data type Int64 is not currently supported. Support for additional datatypes in APL64 will be considered for future versions.

Component File Exception

A FILE TIED error message was reported instead of FILE NOT FOUND when tying a non-existent component file

This issue will be addressed in the next APL64 version update.

CSE: REPLSTR Property

CSE: I could not find any reference to replstr (one of the properties) in the CSE manual. How it it meant to be used?

Use Help > APL Language > C# Script Engine Manual > CSE replstr property section to see the documentation and multiple examples for the CSE instance property 'replstr':

CSE: Enumeration of Static Methods

CSE: Doesn't enumerate methods in a *static* class and lambda functions in a non-static class

If this CSE enhancement is possible, it will be considered for inclusion in a future version of APL64.

CSE: DateTime Exception

CSE: Why the CSE ERROR: .Net data type, System.DateTime, has no representation in APL64 for DateTime columns?

The result in APL64 is as designed. The DateTime structure in .Net has fields, properties, methods and operators. APL has no representation for all of these features of the DateTime structure, so the result in APL64 is correct.

CSE: ☐wi 'Reset' Effect

CSE: Does '#' ☐wi 'Reset' have any impact on existing instances of C# Script Engine?

The '#' ☐WI 'Reset' operation does NOT have any affect on the C# Script Engine state.

CSE: Quick Start Document

The ☐cse_Quick_start_V1.0.0.0.pdf is missing its Table of Contents

This document has been updated with a Table of Contents and is available now to be previewed.

CSE: Examples are now in CSE Manual

Are there workspaces containing the functions in each of the CSE manuals?

The APL64 CSE user manual contains code sections which can be copied by the user directly from the document, so there is no longer a need for CSE workspaces. This assures that the sample code remains in synchronization with the documentation.

☐wi 'Reset' Action

Does '#' ☐wi 'Reset' do more in APL64 than in APL+Win?

No. '#' ☐WI 'Reset' does the exact same thing in APL64 as in APL+Win.

Octal Value Entry

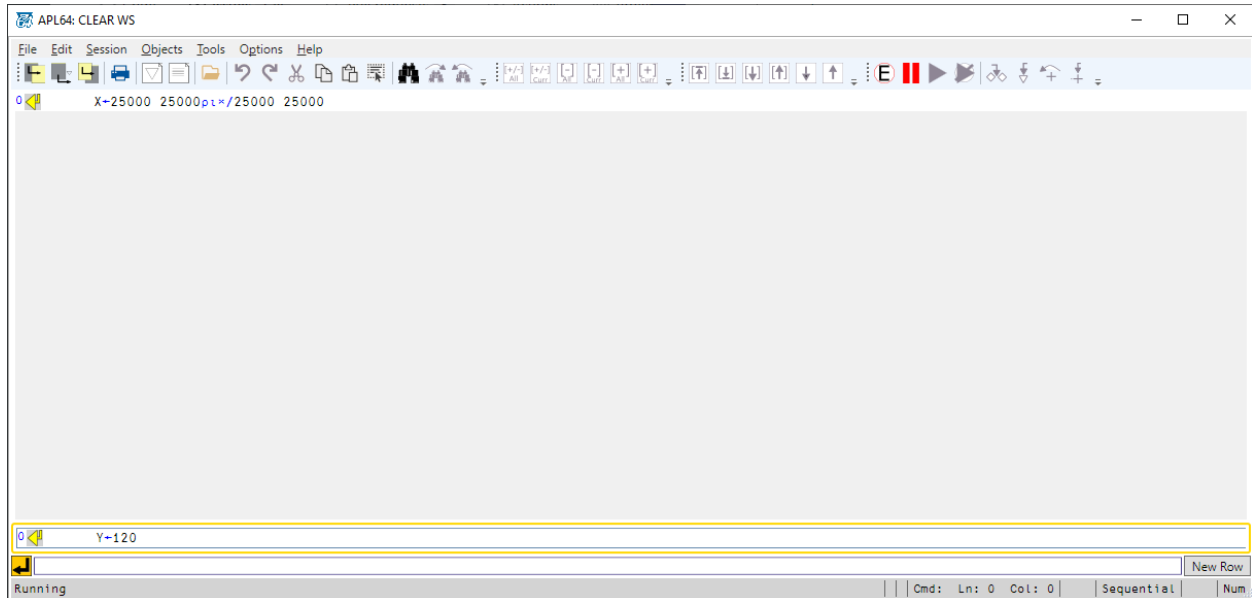
Does APL64 have support for Octal values?

No. This may be considered as an enhancement in a future APL64 version.

Hour Glass Mouse Pointer

The hour glass mouse pointer is missing during long execution cycle in APL64

The 'Running' text in the main window status bar and the yellow image to the left of the APL64 Command Line indicates that the APL64 system is 'busy'. The pointer icon cannot be changed to an 'hour glass' or 'spinning bagel', because, as noted in the APL64 Project History documents, the APL64 Command Line remains active and the user may enter additional APL executable statements while a long-running APL executable statement is being processed by the APL64 system. These additional APL executable statements will be placed in the Pending APL Statements data grid with a yellow border. For example:



WebView2 Document Viewer

The Find window in the online Webview2 Document Viewer can remain visible after closing the Document Viewer

This issue has been reported to the Microsoft WebView2 developers for investigation.

F10 Short Cut

Step to Next Line (SHIFT + F10) in the Debugger Pane terminates APL64.

This has been fixed and the fix will be included in the next update.