

Modifications Included in APL64 Pre-Production Version (2020.0.8)

Table of Contents

Overview	4
Enhancements	4
New System Function: ☐ XML	4
APL64 Idioms Manager	4
New menu item: Objects Variable Editor Minimum Column Width in DataGrid for specifying the minimum column width for objects displayed in the variable editor DataGrid	4
The current APL64 keyboard definition added to the APL64 StatusBar	5
Bug Fixes	6
Saving a function with thousands of lines in the editor with all regions collapsed may take an extraordinarily long time to complete	6
An APL function could suspend unexpectedly when branching (→) to a label appearing at the end of the executed statement.....	6
The keyboard shortcut keys that included functions keys did not successfully execute when the focus was in the Debug or State Indicator panes	6
Incorrect token highlighted when mouse double clicking.....	6
A mouse double click may include the leading single and double quotes in the selection.....	6
A library number, instead of the folder it was linked to, was displayed in the Recent Workspace Ids ...	7
The caret (focus) didn't remain in the editable history for the duration of debugging a function	7
The navigation cursor keys did not move the caret in the Debug pane	7
The Session menu remained highlighted after a new selection was made in the menu: Session ToolTips Display	7
APL64 crashed when copying a workspace with a function that had the same argument name.....	7
A selection from the APL64 Active Pane list with Ctrl+Tab did not update the line and col numbers information in the StatusBar.....	7
The Quad and Quote-Quad input did not set the focus on the APL Command Line while simultaneously stopped in the debugger	7
There could be problems with rendering of APL executable statements in the editable History while suspended in a function due to events firing out of order	7
The editable History was not accurate after using the backspace key to modify a statement on an earlier line then executing it	7
A right or left mouse double click in the empty APL command line or History opened the default editor when the caret was positioned at the beginning of an empty line	7

)lib,)flib,)wslib and □lib with an invalid directory returned error message: SYSTEM ERROR:Exception: The filename, directory name, or volume label syntax is incorrect...	8
Evaluators' Comments & APLNow's Responses	9
Word or Object Under Caret	9
Is it possible/viable to show the Keyboard on the status bar of APL64?	9
Is it possible/viable to have the installer select the appropriate APL64 keyboard at installation? (case #5480)	10
The User Tool echoes the executed code to the session (case #5477)	10
With APL64, Gather is more enhanced and NOT intuitive as in APL+Win (case #5479)	10
With APL64, Windows Key and V does NOT update following Ctrl+A followed by Ctrl + Shift + G and Copy to Target (case #5478)	11
APL64 should always store most recent workspace used WITHOUT library numbers! (case #5487)....	11
Adding a first argument to the three □XL methods arguments, GetR1C1AddrFromA1Addr, GetR1C1AddrFromAbsA1Addr, and GetA1ColNameFromColNum, if they should work for a .XLS file or .XLSX file (case #5488)	12
I pressed Ctrl + Shift + O with the cursor on quadXL and)ed ▽quadXL appears in the session window (case #5506)	12
Why is the depth of the data variable 1 returned from □xl when I expected 2?...	12
Does □XL ever return C# type char as char (meaning with shape ⍉) instead of a character vector of length 1? (case #5501)	13
.NET Multi-platform App UI (MAUI) and □cse (case #5512)	13
APL64 did not run after installing 2020.0.7 (case #5476)	13
APL files with accented characters in the name cannot be created or opened (case #5539)	13
□xferase returned "FILE ACCESS ERROR" (case #5540)	13
The APL+Win Windows Reference Manual does not show the APL chars properly (case #5542)	14
I cannot find how to change an object type in the editor (e.g., from function to variable or from matrix to vector) (case #5524)	14
□SI describes □SINL, but there is no entry for □SINL (case #5528)	14
DO NOT USE multi-statements in documentation -- confusing for user... (case #5530)	14
□ROWFIND documentation confusing and too elaborate (case #5529)	14
□SIZE rejects invalid names with DOMAIN ERROR; does not return 0 (case #5531)	14
□SIZE example wrong (returns 3p112, not 3p32) (case #5533)	14
Why did calling Word.Application launch Word in read-only mode? (case #5541)	14
Would it be possible to have (or be able to set) a minimum column width for the Variable editor? Currently, a variable with small integers displays like this in the editor: (case #5244)	15

Is it possible to select more than one cell, for example if the user wants to copy/paste part or all of the edited variable to Excel? (case #5245)	15
Are you voluntarily favoring copy/pasting from the Variable Editor to Excel over copy/pasting to APL? (case #5246)	15
Ctrl+Sh+O from within a function session opens wrong object; it seems to open most recently opened object (case #5310-1)	16
Ctrl+Sh+O from within the history does not put the focus on the newly opened object (case #5310-2)	16
APL64 crashes when copying with  copy a workspace with a function named SCAN (case #5436)...	16
 nfe'GetLastAccessTime' returning false information? (case #5486)	16
My dyadic iota example throws this SYSTEM ERROR (below):... (case #5288)	17
10 `2147483641 engenders LIMIT ERROR. Modulus seems to have trouble with large negative integers. (case #5525)	17
`2147483648↓[1]2 3p16 returned LIMIT ERROR (case #5526)	17
 2p /θ produces a floating point indefinite (displays as 0.0E291) (case #5450)	18
 LIB gives voluminous error message for invalid directory (case #5453)	18
' '^.= 1 2 3 returned "SYSTEM ERROR:Exception: CS0103: The name 'rfcs' does not exist in the current context" (case #5538)	18

Overview

This document describes the enhancements and bug fixes incorporated into the eighth pre-production version of APL64 released in 2020.0.8. This document also describes the evaluator comments and APLNow responses based on the prior pre-production versions of APL64.

Enhancements

New System Function: [□XML](#)

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

Documentation is available in the menu **Help | APL Language | Using [□XML](#)**.

[APL64 Idioms Manager](#)

The APL64 Idioms Manager is a programmer productivity tool to enable the deployment of APL idioms into APL64-based application systems. An APL idiom is text which represents frequently used APL statements. The APL idiom may contain complete APL executable statements or may require additional APL programmer input to become APL executable statements. An option is provided to specify the placement of the caret when the idiom text is inserted into a function editor, variable editor, command line or editable classic history.

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

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

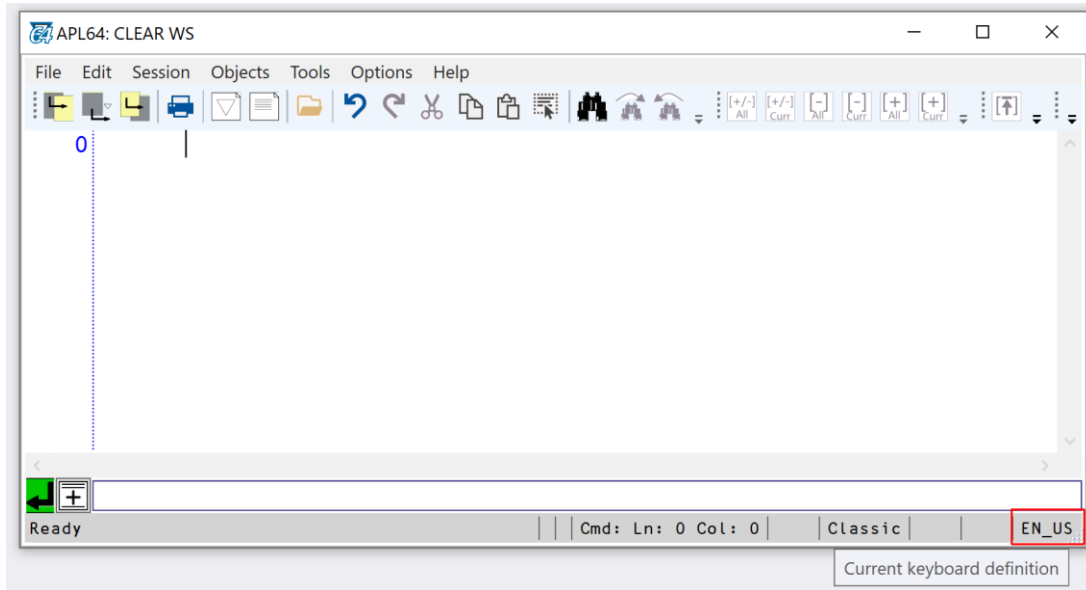
The APL64 Idioms Manager dialog is used to add or edit user-defined APL idioms:

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

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

New menu item: [Objects | Variable Editor | Minimum Column Width in DataGrid](#) for specifying the minimum column width for objects displayed in the variable editor DataGrid

The current APL64 keyboard definition added to the APL64 StatusBar



Bug Fixes

Saving a function with thousands of lines in the editor with all regions collapsed may take an extraordinarily long time to complete

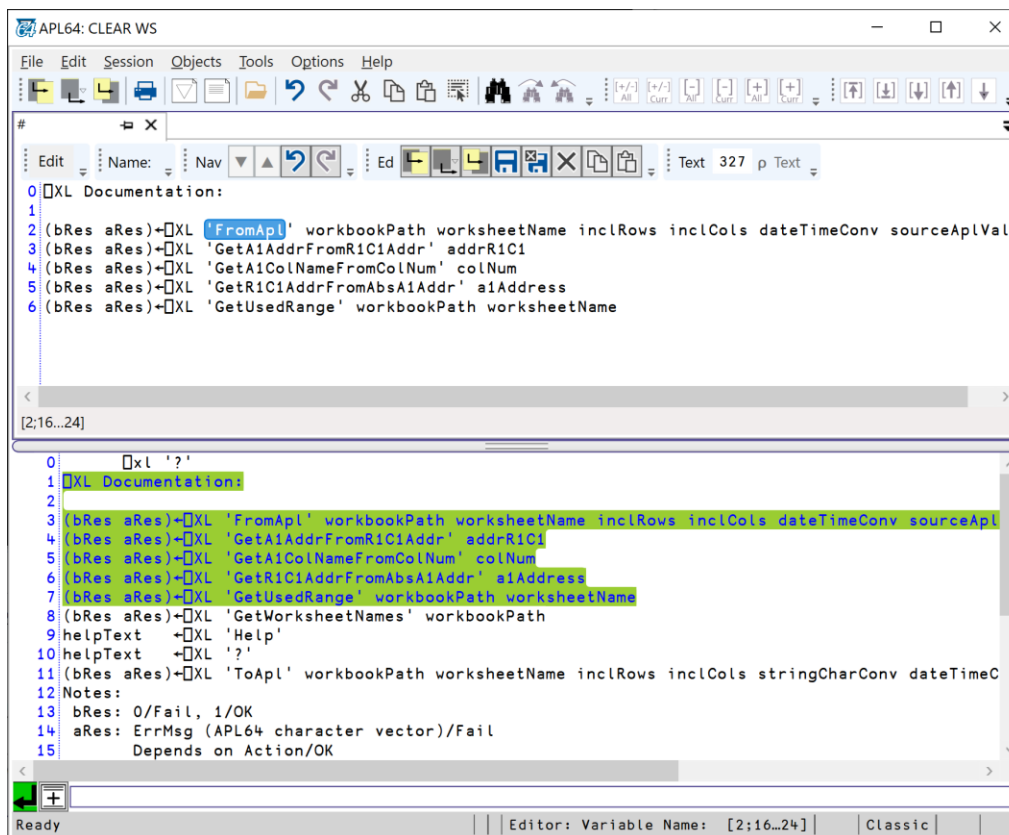
An APL function could suspend unexpectedly when branching (→) to a label appearing at the end of the executed statement

The keyboard shortcut keys that included functions keys did not successfully execute when the focus was in the Debug or State Indicator panes

Incorrect token highlighted when mouse double clicking

The first word in a string would be highlighted when mouse double clicking on a subsequent word beginning with an uppercase alphanumeric letter.

A mouse double click may include the leading single and double quotes in the selection
e.g.



A library number, instead of the folder it was linked to, was displayed in the Recent Workspace Ids

The caret (focus) didn't remain in the editable history for the duration of debugging a function

The caret was incorrectly placed in the APL command line at the start and end of the debugging session for an editable history.

The navigation cursor keys did not move the caret in the Debug pane

The Session menu remained highlighted after a new selection was made in the menu:
Session | ToolTips | Display

APL64 crashed when copying a workspace with a function that had the same argument name

A selection from the APL64 Active Pane list with Ctrl+Tab did not update the line and col numbers information in the StatusBar

The Quad and Quote-Quad input did not set the focus on the APL Command Line while simultaneously stopped in the debugger

There could be problems with rendering of APL executable statements in the editable History while suspended in a function due to events firing out of order

The editable History was not accurate after using the backspace key to modify a statement on an earlier line then executing it

A right or left mouse double click in the empty APL command line or History opened the default editor when the caret was positioned at the beginning of an empty line

Note: When menu **Object | Right (Left) Mouse Double Click** was enabled.

Pressing the Enter key in a docked/floated variable editor with multiline text (rank<3) scrolled to the last line of the text

)lib,)flib,)wslib and □lib with an invalid directory returned error message: **SYSTEM ERROR:Exception: The filename, directory name, or volume label syntax is incorrect...**

Evaluators' Comments & APLNow's Responses

This section contains the summary of the evaluators' comments (preceded with ">") and APLNow's responses to the Pre-Production Version of APL64 (APL64_2020.0.8. 18996). We would like to thank the evaluators for their contribution.

Word or Object Under Caret

[User] For Ctrl + C, it is the whole line.

>For Ctrl + Shift + O, it is the word delimited by space at either end.

>For both of the above, the line or word does NOT have to be selected/highlighted.

Your comment on Ctrl+C is true only when the menu **Edit | Copy/Cut Entire Line Without Selections** is checked.

[User] What is considered to be under the caret for ∇ in the context of user defined tools?

- i. Is it the word with space at either end? What if that word includes punctuation such as : and \ > in a fully qualified filename?*
- ii. Does the word always need to be selected/highlighted?*

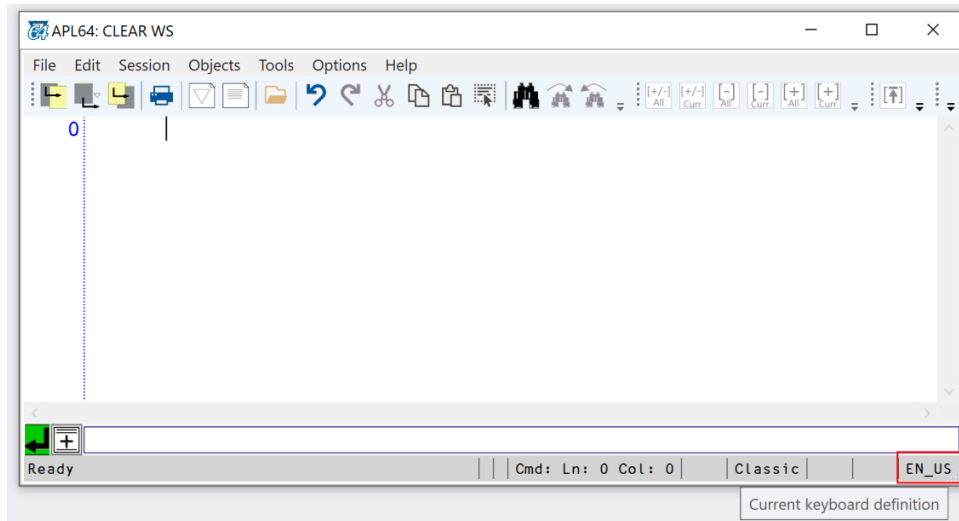
Per the documentation, the del (∇) is replaced with any text that is highlighted in the current window or with the object that is closest to the cursor in the current window if you did not highlight anything before invoking the tool.

[User] iii. Is the word read as ∇ cached? That is, is it ever re-used internally?

No.

Is it possible/viable to show the Keyboard on the status bar of APL64?

In the next available APL64 pre-production version user-selected APL64 keyboard definition will be illustrated in the APL64 StatusBar. (Ver: 2020.0.8)



Is it possible/viable to have the installer select the appropriate APL64 keyboard at installation? (case #5480)

We would not recommend automatically selecting the APL64 Keyboard definition based on the user's workstation 'region' during installation because 'region' may differ from the user-selected 'Language Preference'.

The User Tool echoes the executed code to the session (case #5477)

Your observation is correct, and this is by design of APL64. In APL64 the history pane is designed to accurately reflect what occurred during an instance of the APL64 Developer version. To maintain this accurate history, the cause and effect of an action are displayed in the APL64 history pane. Examples of this are a user-entered APL executable statement, interpreter callbacks, invocation of a user-defined tool, etc. In each case the cause and the result, if any, are displayed in the APL64 history.

In APL+Win the history was often inaccurate since the cause of an action was not always displayed in the history.

For more on this issue, please refer to **APL64 | Help | Developer Version GUI | Editable/Non-editable Classic History**.

With APL64, Gather is more enhanced and NOT intuitive as in APL+Win (case #5479)

User: Simply using Ctrl+A followed by Ctrl + Shift + G and simply clicking Copy to Target does NOT copy the executed lines to the clipboard

Your observation is correct if the Windows Clipboard has not been selected as the copy target. Click the toggle button to the left of the Copy Target label in the dialog to change the target. The tooltip will change depending on the target.

User: Why Copy to Target? I am guessing that Target is either the Editor or Session or Command window – whichever has focus, Besides, nothing get copied NEITHER to the Editor or Session or Command windows NOR to the clipboard

There is no need to guess. At any time, there are two possible targets. Click the toggle button to the left of the Copy Target label in the dialog to change the target. The tooltip will change depending on the target. The first target is the Windows Clipboard. The second target is either the editable history pane or the command line. If the user-selected history format is not 'Classic' or if it is 'Classic' (but it is not editable) the second target is the command line. If the user-selected history format is 'Classic' (and it is editable) the second target is the editable history pane.

User: The Statement Type option (highlighted in blue in the screenshot) toggles three options: Exec/Result/All Stmts: the toggle has an unpleasant side effect, namely, it causes the dialogue to resize or to shrink and expand

Your observation is correct, and the display depends on the user-selected width for the dialog.

With APL64, Windows Key and V does NOT update following Ctrl+A followed by Ctrl + Shift + G and Copy to Target (case #5478)

[User]: With APL64, Windows Key and V does NOT update following Ctrl+A followed by Ctrl + Shift + G and Copy to Target.

Your observation is correct. Currently APL64 is based on .Net 5. When in a future version APL64 is updated to .Net 6, the Win+V can be implemented in APL64.

[User]: The number of entries in the Windows (Windows Key and V) clipboard is restricted not least because other applications can pin entries which reduce the number of available lines even further. Therefore, I am ambivalent about this feature: on the one hand, it is useful to have APL content available across sessions but on the other hand, what can be available is rather limited.

Using the Windows Clipboard as the target is not mandatory, so its use is determined by the APL programmer.

APL64 should always store most recent workspace used WITHOUT library numbers! (case #5487)

Actually, the same issue did occur in the previous pre-production release. We agree library numbers should not appear in the most recent workspace ids and will investigate correcting this in a future pre-production release (ver: 2020.0.8).

Adding a first argument to the three `□XL` methods arguments, `GetR1C1AddrFromA1Addr`, `GetR1C1AddrFromAbsA1Addr`, and `GetA1ColNameFromColNum`, if they should work for a .XLS file or .XLSX file (case #5488)
Your suggestion to add a code argument to these methods is 'old style' Win32/APL+Win design. The limitations of .xls workbooks are based on the proprietary format of Excel 97-2003, which Microsoft has deprecated long ago.

However, your request could be accomplished if the methods in question have an optional last argument specifying a particular physical workbook file path or a text vector ("Excel8"/"OpenWb"), which would not disrupt the existing syntax and handle future versions of Excel, if any:

1. If the optional argument is present, an exception will be thrown by the method if the requested value is not valid for the specific workbook.
2. If not present, the values returned will be for the most recent version of Excel which is supported by the tool used to implement `□XL` in APL64.

Your request to enhance these methods will be considered for a future version of APL64.

I pressed Ctrl + Shift + O with the cursor on `quadXL` and `∇quadXL` appears in the session window (case #5506)

This behavior in 2020.0.7 is by design and documented so in the last bullet item in the Help | APL+Win Compatibility | Compatibility Modifications for Existing APL+Win Applications:

- To maintain an accurate history in the APL64 Developer version, any modal editor initiated from the APL64 Developer version will update the history pane with the appropriate APL executable statement.

The fact that APL+Win didn't do this was an oversight.

Why is the depth of the data variable 1 returned from `□xl` when I expected 2?...

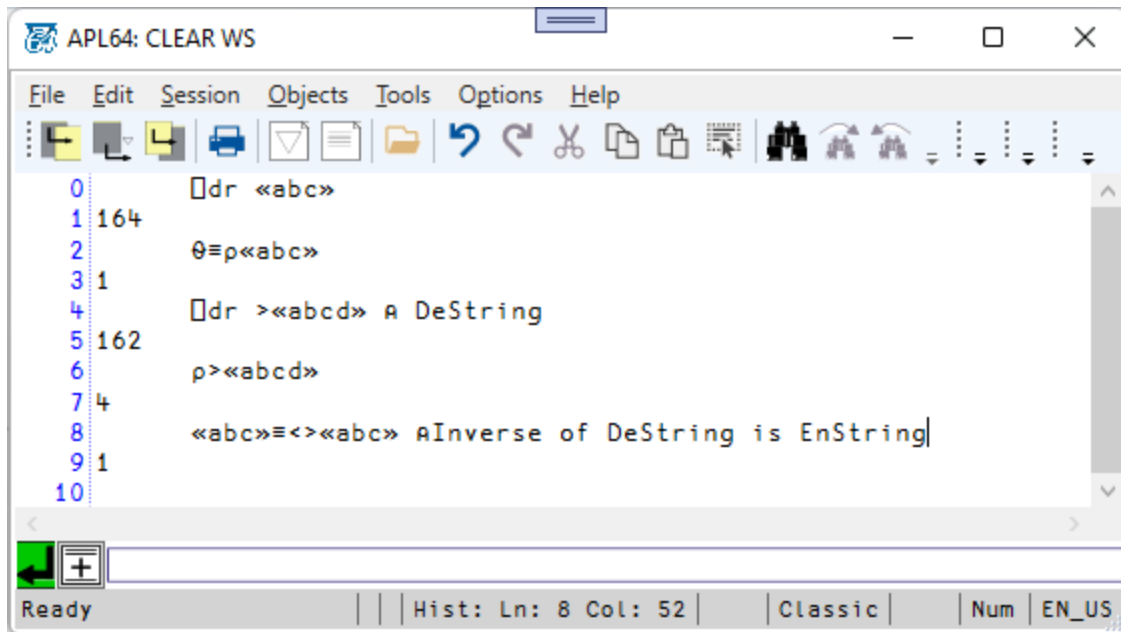
...Why can't I get the shape of each element of data? It comes across as empty? Can data be cast as APL+Win data? If not, is this compatibility failure? (case #5501)

Without knowing what was in your worksheet, the following are guesses:

- The data returned is valid APL64 data, but some data elements are scalars.
- You selected 'APLSTRING', so character data is returned as APL64 string objects.

Try 'AplChar' instead. APL+Win does not have a string data type, its limited to character data.

An APL64 string object is a scalar, so its shape is zilbe. Try this:



Does ☐XL ever return C# type char as char (meaning with shape θ) instead of a character vector of length 1? (case #5501)

No. When the user requests APL character data to be returned, any Excel cell text value is returned as an APL64 character vector, even if the Excel text value is a single character.

.NET Multi-platform App UI (MAUI) and ☐cse (case #5512)

There are no reports of APL2000 customers using .NET MAUI.

Based on the Microsoft documentation, MAUI requires .Net 6. When APL64 is updated to .Net 6 and Visual Studio support for MAUI is finalized, the availability of MAUI in ☐CSE can be determined.

APL64 did not run after installing 2020.0.7 (case #5476)

Try installing the .NET Desktop Runtime 5.0.17 (x64) at <https://dotnet.microsoft.com/en-us/download/dotnet/5.0>.

APL files with accented characters in the name cannot be created or opened (case #5539)

This is normal behavior with the traditional component file system. You will want to use the extended component file functions instead, e.g., 'versé' ☐xfcreate 1

☐xferase returned "FILE ACCESS ERROR" (case #5540)

This is normal behavior. The reason is that the file must be exclusively tied with ☐xftie for it to be erased with ☐xfcreate.

The APL+Win Windows Reference Manual does not show the APL chars properly (case #5542)

The evaluator's reply to request for specific examples:

That one is now solved after restarting Win10.

It seems that my PDF reader needs a MS-Win restart to make even embedded fonts available. [It reminds me of the same problem with another software.] Anyway, I see APL chars now.

I cannot find how to change an object type in the editor (e.g., from function to variable or from matrix to vector) (case #5524)

The issue will be investigated.

☐SI describes ☐SINL, but there is no entry for ☐SINL (case #5528)

The error in the description for ☐SI will be corrected. ☐SINL is a system variable and therefore described in System Variables.

DO NOT USE multi-statements in documentation -- confusing for user... (case #5530)

...Some examples of confusing multi-line statements in documentation:

`[]DEB []DIV []ROWFIND []OVERV []SKD`

This will be reviewed and if any updates will be considered.

☐ROWFIND documentation confusing and too elaborate (case #5529)

The current documentation will be reviewed, and any edits deemed necessary will be considered.

☐SIZE rejects invalid names with DOMAIN ERROR; does not return 0 (case #5531)

Actually, it's a bug in APL+Win that it returned DOMAIN ERROR instead of zero, as documented, for an invalid name. So, APL64 is behaving correctly here except that we noticed that a zero wasn't returned for all invalid names and have reported this to the developers to investigate.

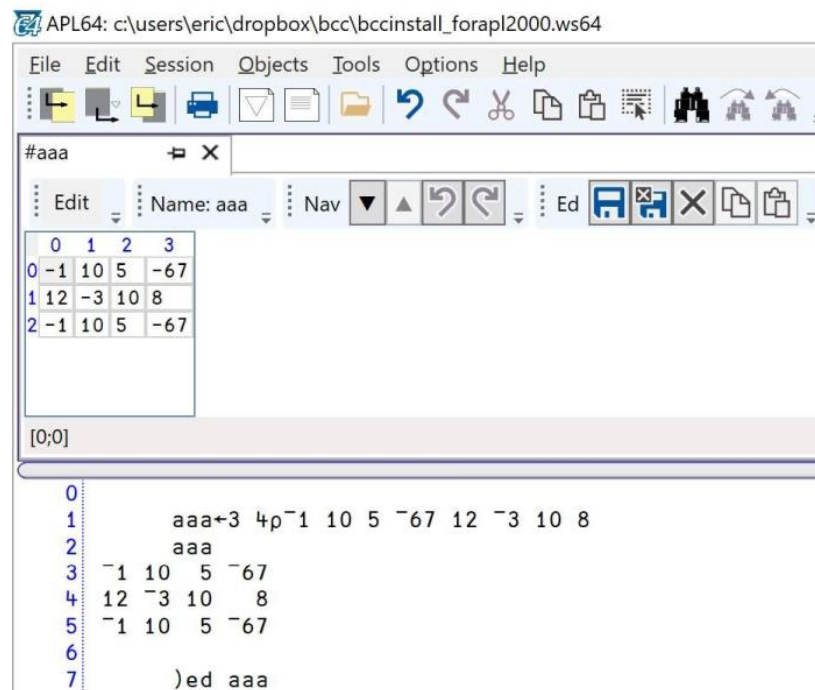
☐SIZE example wrong (returns 3p112, not 3p32) (case #5533)

The example in the System Functions manual was corrected.

Why did calling Word.Application launch Word in read-only mode? (case #5541)

Calling external ActiveX components like Word should operate similarly in APL64 as in APL+Win. Was the same read-only mode for Word observed when instantiated from APL+Win on the same machine? I suspect this is the case, in which case there is no issue in APL64.

Would it be possible to have (or be able to set) a minimum column width for the Variable editor? Currently, a variable with small integers displays like this in the editor: (case #5244)



This enhancement has been implemented in this release. Refer to the new menu item:
Objects | Variable Editor | Minimum Column Width in DataGrid.

Is it possible to select more than one cell, for example if the user wants to copy/paste part or all of the edited variable to Excel? (case #5245)

The variable editor is in an early stage of development, but yes copying/pasting blocks of cells is planned. In the meantime, use)EDSS or ☐ EDSS to put the APL64 variable data into an Excel worksheet.

Are you voluntarily favoring copy/pasting from the Variable Editor to Excel over copy/pasting to APL? (case #5246)

When the current decimal separator on the workstation is comma, the behavior in APL64 is expected and by design. Although it may be possible to change the formatting of numbers in the APL64 editor so that they would use the period decimal separator, that would eliminate the possibility of conveniently copying from APL64 to Excel which you point out in your report.

Instead, when exclusively working in APL64, use the Windows utility intl.cpl (Additional Settings) to change the decimal separator from comma to period (and any other settings in that dialog deemed necessary) in which case the APL64 editor will provide values which are convenient to copy from APL64 to APL64.

It will not be possible to simultaneously support both options in APL64.

Ctrl+Sh+O from within a function session opens wrong object; it seems to open most recently opened object (case #5310-1)

The fix for this issue below will be available in the next release of APL64. (Ver: 2020.0.8).

Ctrl+Sh+O from within the history does not put the focus on the newly opened object (case #5310-2)

We haven't been able to reproduce the issue below. But it's possible that the same fix (Ver: 2020.0.8) will also address this too. If this issue persists for you, please provide us with a reproducible scenario.

APL64 crashes when copying with `⎕copy` a workspace with a function named SCAN (case #5436)

This occurred because the function had a right argument also named SCAN. The fix for this issue below will be available in the next release of APL64. (Ver: 2020.0.8).

`⎕nfe'GetLastAccessTime'` returning false information? (case #5486)

It seems to me that `⎕nfe'GetLastAccessTime'` sometimes return a wrong result: for example, I have not used the following 4 files in a very long time, but `⎕nfe'GetLastAccessTime'` returns a last access time of today at 2:36:03!

e.g.,

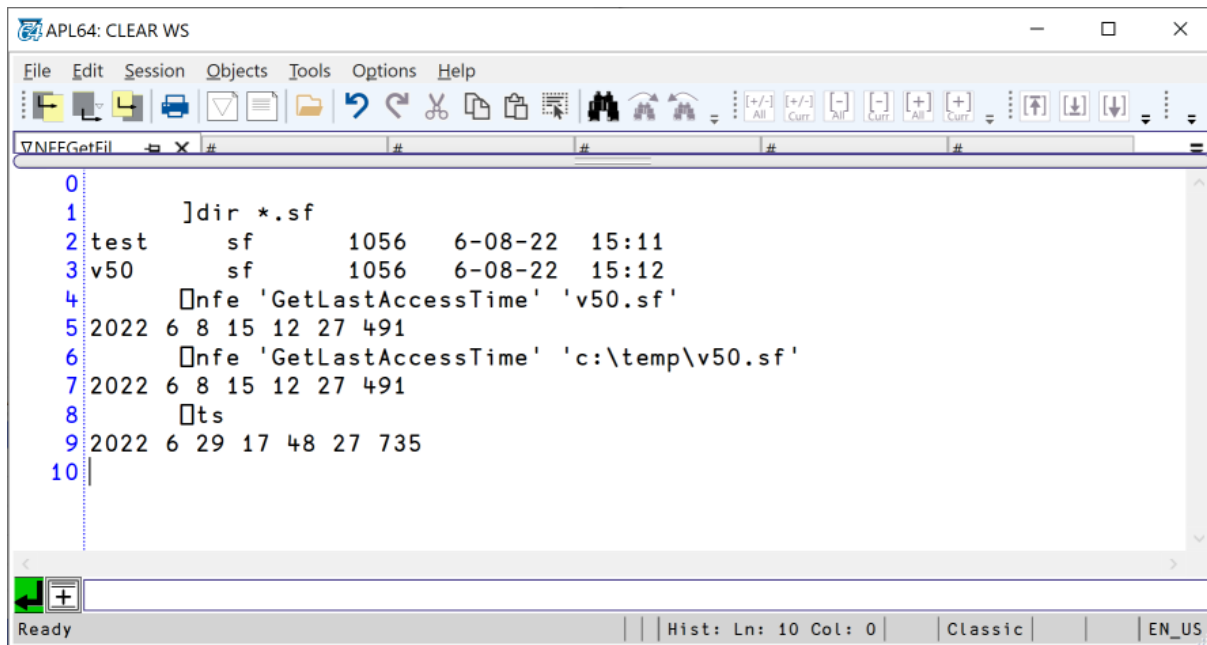
```
⎕nfe'GetLastAccessTime' 'c:\temp\test1.sf'  
2022 6 29 2 36 3 704
```

```
⎕nfe'GetLastAccessTime' 'c:\temp\test2.sf'  
2022 6 29 2 36 3 709
```

```
⎕nfe'GetLastAccessTime' 'c:\temp\test3.sf'  
2022 6 29 2 36 3 720
```

```
⎕nfe'GetLastAccessTime' 'c:\temp\test4.sf'  
2022 6 29 2 36 3 720
```

We can't replicate your issue as I've demonstrated below:



```
0
1      ]dir *.sf
2 test      sf      1056   6-08-22   15:11
3 v50       sf      1056   6-08-22   15:12
4      [nfe 'GetLastAccessTime' 'v50.sf'
5 2022 6 8 15 12 27 491
6      [nfe 'GetLastAccessTime' 'c:\temp\v50.sf'
7 2022 6 8 15 12 27 491
8      [ts
9 2022 6 29 17 48 27 735
10
```

If possible, please provide to support@apl2000.com a specific scenario which can duplicate your issue.

[My dyadic iota example throws this SYSTEM ERROR \(below\):... \(case #5288\)](#)

```
-----
SYSTEM ERROR:Exception: Index was outside the bounds of the array.
at APLNow.Data.Aval.Hash()
at APLNow.Core.Prim.hpeifast(Apl apl, Aval l, Aval r, Boolean opEpsilon)
at APLNow.Core.Prim.hpepsiota(Apl apl, Aval l, Aval r, Char op)
at APLNow.Core.Code.XAPD.Exec(Apl apl)
at APLNow.Core.Apl.Run()
[imm] ...
-----
```

The fix for this issue below will be available in the next release of APL64. (Ver: 2020.0.8).

[10|`2147483641 engenders LIMIT ERROR. Modulus seems to have trouble with large negative integers. \(case #5525\)](#)

The issue will be investigated, and any correction will be in a future release of APL64. (Ver: 2020.0.8)

[`2147483648↓\[1\]2 3p16 returned LIMIT ERROR \(case #5526\)](#)

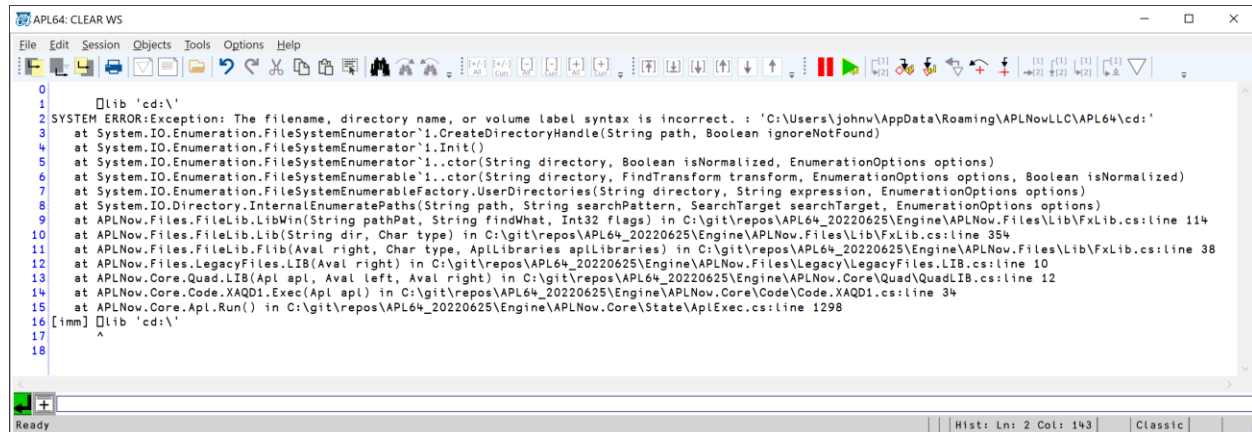
The issue will be investigated, and any correction will be in a future release of APL64. (ver 2020.0.8)

 2p|/θ produces a floating point indefinite (displays as 0.0E291) (case #5450)

The issue will be corrected to return LIMIT ERROR in a future APL64 release. (Ver: 2020.0.8)

 LIB gives voluminous error message for invalid directory (case #5453)

e.g.,



```
0
1      [Lib 'cd:\']
2 SYSTEM ERROR:Exception: The filename, directory name, or volume label syntax is incorrect. : 'C:\Users\johnv\AppData\Roaming\APLNowLLC\APL64\cd:'
3   at System.IO.Enumeration.FileSystemEnumerator`1.CreateDirectoryHandle(String path, Boolean ignoreNotFound)
4   at System.IO.Enumeration.FileSystemEnumerator`1.Init()
5   at System.IO.Enumeration.FileSystemEnumerator`1..ctor(String directory, Boolean isNormalized, EnumerationOptions options)
6   at System.IO.Enumeration.FileSystemEnumerable`1..ctor(String directory, FindTransform transform, EnumerationOptions options, Boolean isNormalized)
7   at System.IO.Enumeration.FileSystemEnumerableFactory.UserDirectories(String directory, String expression, EnumerationOptions options)
8   at System.IO.Directory.InternalEnumeratePaths(String path, String searchPattern, SearchTarget searchTarget, EnumerationOptions options)
9   at APLNow.Files.FileLib.LibWin(String pathPat, String findWhat, Int32 flags) in C:\git\repos\APL64_20220625\Engine\APLNow.Files\Lib\FxLib.cs:line 114
10  at APLNow.Files.FileLib.Lib(String dir, Char type) in C:\git\repos\APL64_20220625\Engine\APLNow.Files\Lib\FxLib.cs:line 354
11  at APLNow.Files.FileLib.Flib(Aval right, Char type, APLLibraries aplLibraries) in C:\git\repos\APL64_20220625\Engine\APLNow.Files\Lib\FxLib.cs:line 38
12  at APLNow.Files.LegacyFiles.LIB(Aval right) in C:\git\repos\APL64_20220625\Engine\APLNow.Files\LegacyFiles.LIB.cs:line 10
13  at APLNow.Core.Quad.LIB(Apl apl, Aval left, Aval right) in C:\git\repos\APL64_20220625\Engine\APLNow.Core\Quad\QuadLIB.cs:line 12
14  at APLNow.Core.Code.XAQD1.Exec(Apl apl) in C:\git\repos\APL64_20220625\Engine\APLNow.Core\Code\Code.XAQD1.cs:line 34
15  at APLNow.Core.Apl.Run() in C:\git\repos\APL64_20220625\Engine\APLNow.Core\State\AplExec.cs:line 1298
16 [imm] [Lib 'cd:\']
17      ^
18
```

The issue will be addressed in a future release of APL64. (Ver: 2020.0.8)

' '^.= 1 2 3 returned "SYSTEM ERROR:Exception: CS0103: The name 'rfcs' does not exist in the current context" (case #5538)

The issue will be investigated, and a correction will be available in a future release of APL64.