

Modifications Included in APL64 Pre-Production Version (2020.0.3)

Table of Contents

Overview	3
Enhancements	3
New Row Button for Command Line	3
New EN-UK Keyboard Layout Definition.....	4
New Menu: Help > APL Language > System Variable	4
Minimum heights for Debugger/State Indicator, Editor and History Panes.....	5
Bug Fixes	6
!13 returned LIMIT ERROR	6
o /θ froze APL64 when executed.....	6
645 323 □dr 1 2+o0 returned (0 0) (1 1)	6
The Session > View Scale APL Text setting didn't stop at 500 with Ctrl + MouseWheel Up.....	7
AplUserTools not selected in the Import Settings dialogue	7
:TRYALL with the argument * hangs APL+Win (not APL64).....	7
Evaluators' Comments & APLNow's Responses	8
Remove the button labeled New Row in the session command line	8
The online documentation's search/find window remains visible.....	9
!13 returns LIMIT ERROR	9
o /θ freezes APL64 when executed	9
645 323 □dr 1 2+o0 returns (0 0) (1 1).....	10
String substitution to simply text formatting does not work	10
□wcall 'CharUpper' «APL64» aborted APL64	10
Different results for the examples for □nlength in the System Functions manual	11
The View Scale Entire GUI setting set to high at startup.....	11
How can I identify the new String data type?.....	12
APL64_SYSTEM_FUNCTION.pdf appears to be missing □a and □d.....	12
A block select can also be replaced and not just deleted.....	12
When Block Select works with 2-d numeric arrays in the editor window, how will the replace functionality work?	12
Please consider using the same jargon Block Select for Text Block in APL64.....	12
The description: The Shift + Cursor keystrokes select a contiguous, linear block of text is incorrect. The key combination is Alt + Shift + Cursor.	12

It's not possible to use 'EXT' ☐ NA with core NET DLLs such as System.DLL	12
Editor session causes abnormal end when closed with Ctrl + Q	13
Can menu Edit > Select All be greyed out when the window that has focus is empty?	13
Executed Statement History dialogue	13
The right-click menu should disable options that are unavailable within context.....	14
Installing APL64 message: Source file not found...	15
Clarification on the scope of :def ... :endef control structure statement.....	15
☐ wi forms run out of process	16
Should focus go to the GUI form every time the function runs from APL64?	16
Where is the I-Beam?.....	16
Focus bounces between two function editor windows.....	16
How do you execute a 'EXT' ... ☐ NA function that takes no arguments?	17
Issues with Clipboard with block select	17
The APL64 on-screen keyboard is NOT color-coded.....	18
Opening the keyboard Definition has the potential to lose active content	19

Overview

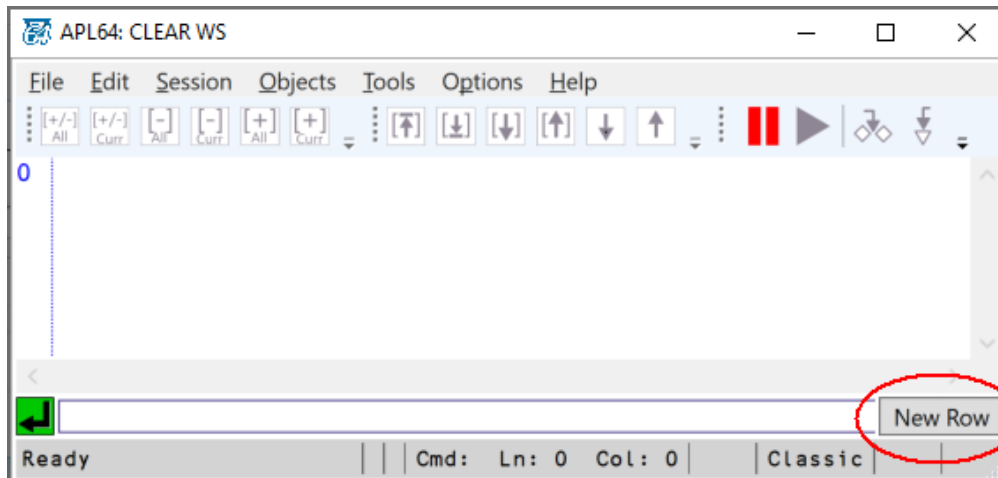
This document describes the enhancements and bug fixes incorporated into the third pre-production version of APL64 released in September 2021. This document also describes the evaluator comments and APLNow responses based on the prior pre-production versions of APL64.

Enhancements

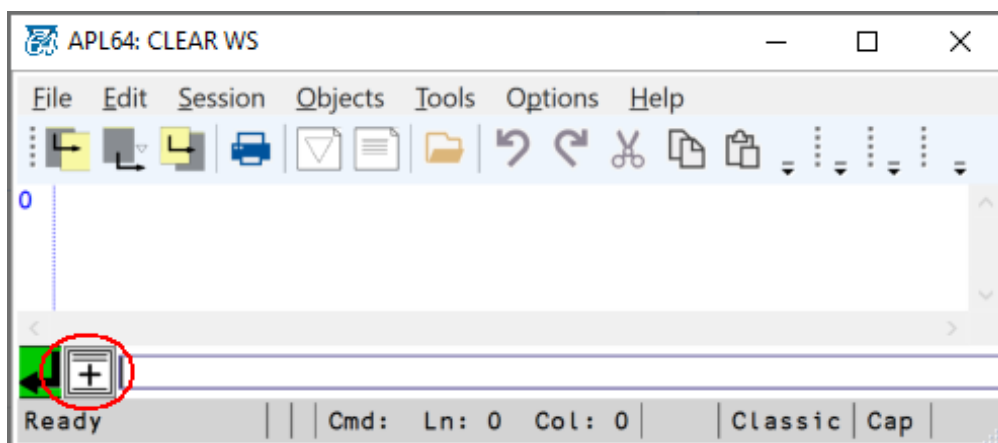
[New Row Button for Command Line](#)

The APL64 command line supports multi-line APL executable statements. The New Row button may be used to add a new line to the existing command line text.

The appearance and location for the New Row button in the session command line has been improved. Below was the previous location and appearance:



Below is the new location and appearance:



New EN-UK Keyboard Layout Definition

Support added for the EN-UK keyboard layout in the Keyboard Definition. Information on this is available in the menu topic: [Help > Developer Version GUI > Keyboard Definition](#)

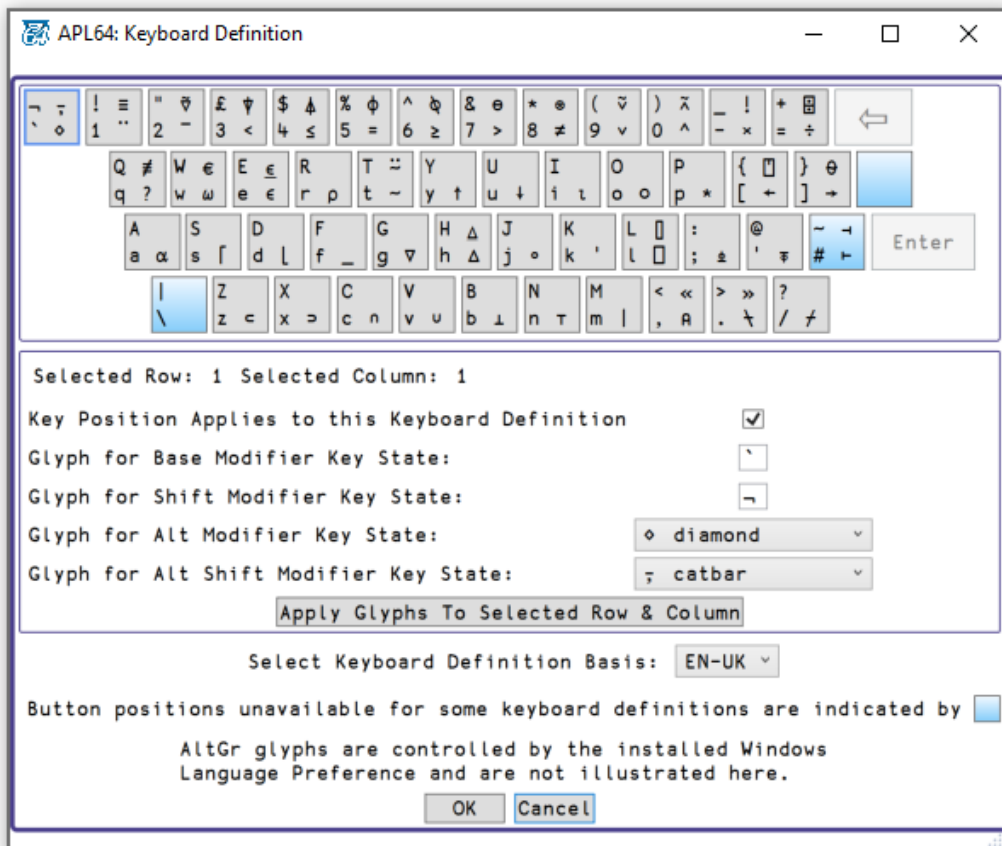
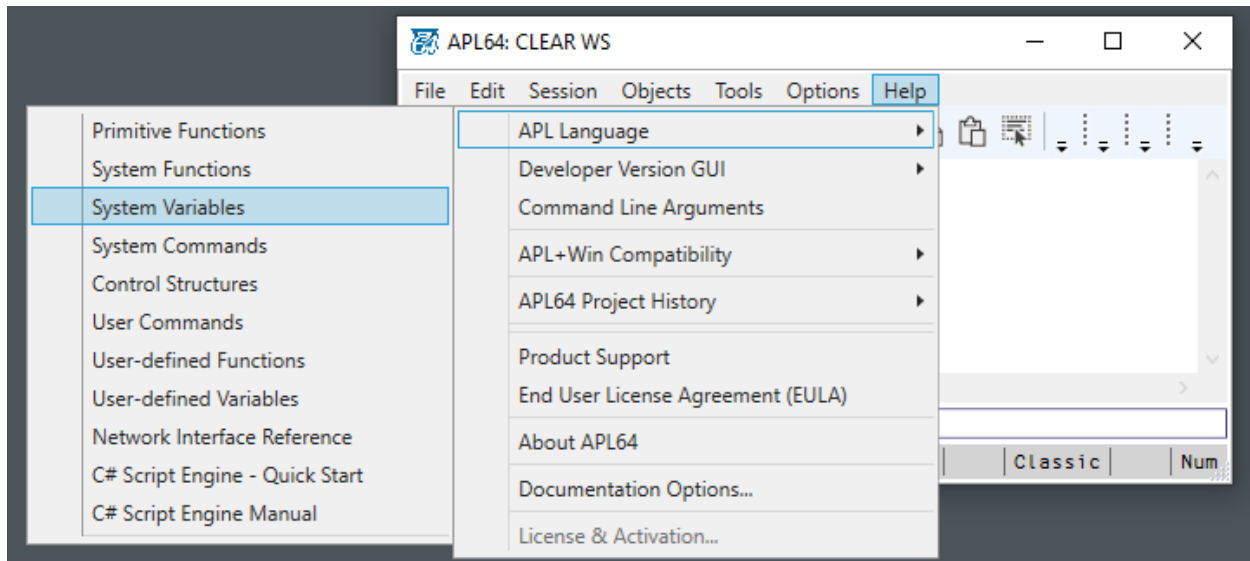


Figure. EN-UK keyboard layout

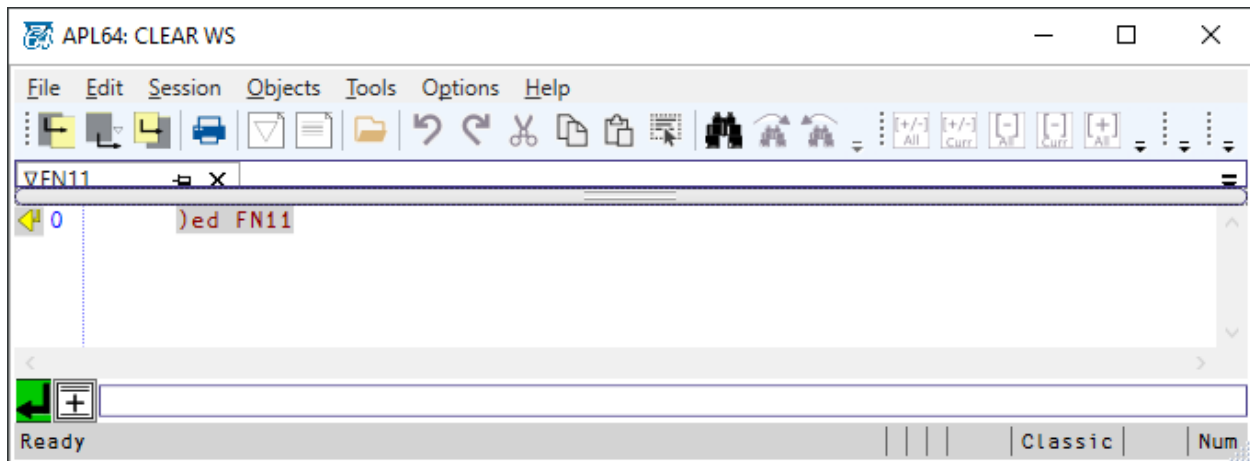
New Menu: [Help > APL Language > System Variable](#)

The System Variables documentation is available online:



Minimum heights for Debugger/State Indicator, Editor and History Panes

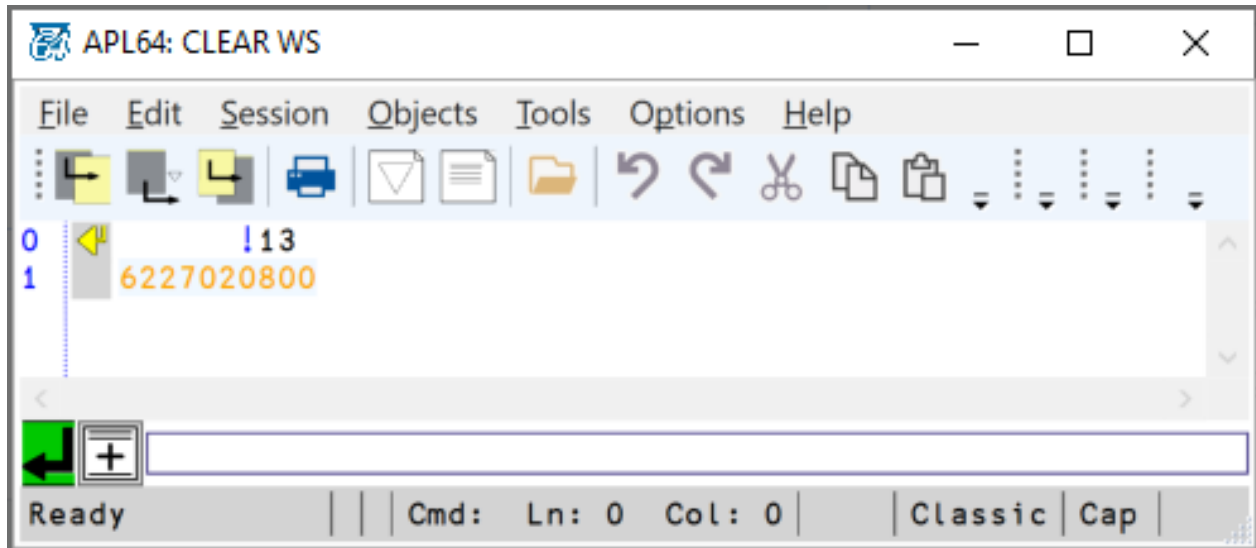
When these panes are made visible, their respective horizontal and vertical splitter bars will now remain visible to allow them to be adjusted. This will avoid the situation where a pane can be completely hidden beneath another pane.



Bug Fixes

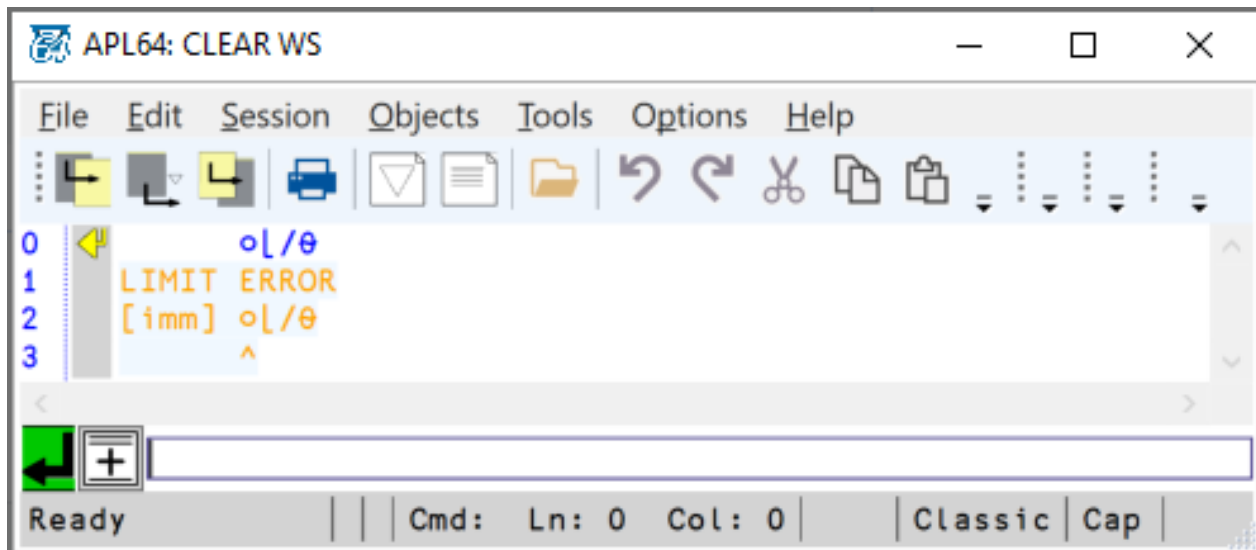
!13 returned LIMIT ERROR

This bug has been corrected to return the result below



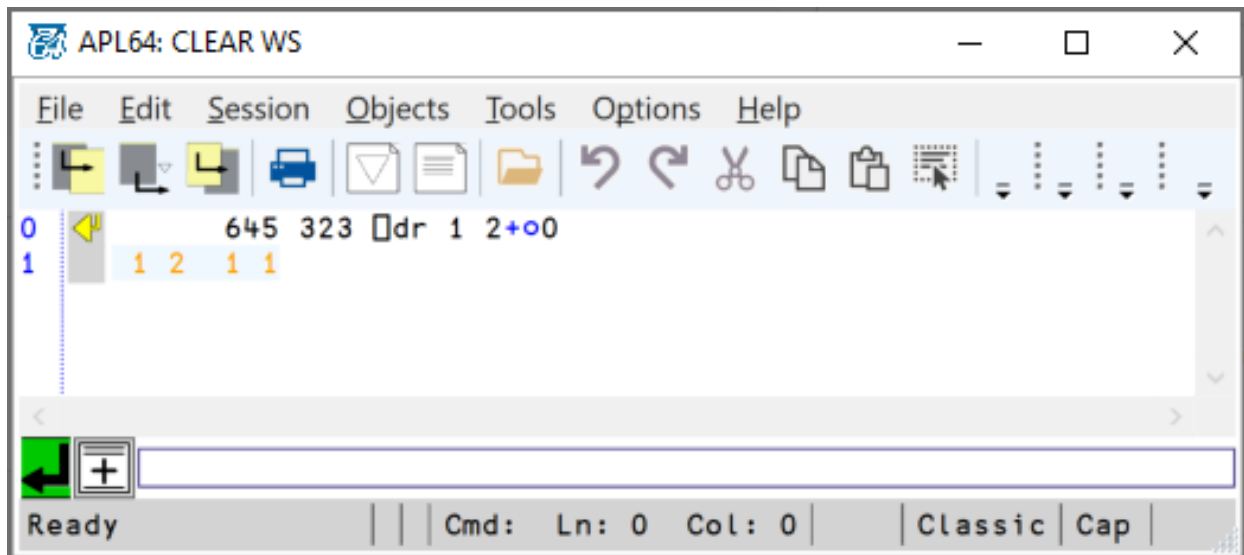
o|/θ froze APL64 when executed

This bug has been corrected to return the result below



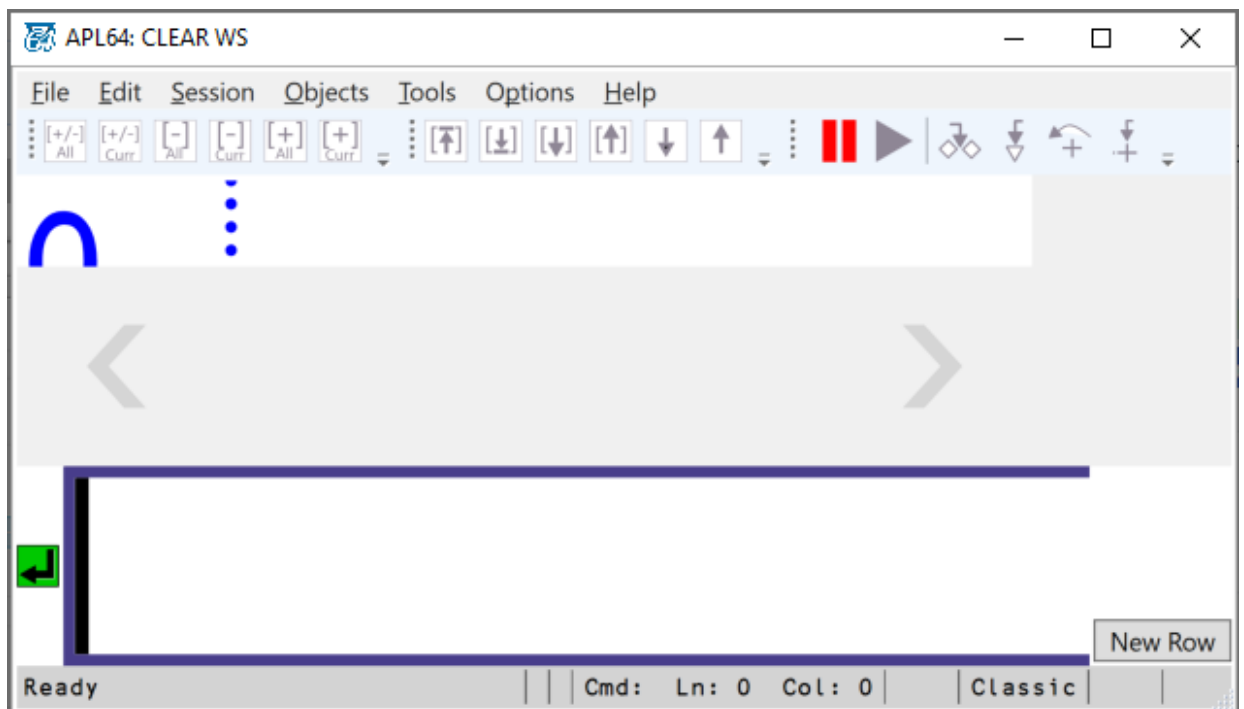
645 323 □dr 1 2+o0 returned (0 0) (1 1)

This bug has been corrected to return the result below



The Session > View Scale APL Text setting didn't stop at 500 with Ctrl + MouseWheel Up

The Session > View Scale APL Text setting didn't stop at the maximum limit of 500 when using the keyboard shortcut key Ctrl + MouseWheel Up. This caused the session history pane and the session command line to be scaled incorrectly, like below:



AplUserTools not selected in the Import Settings dialogue

The AplUserTools settings wasn't selected when the Select All button was clicked in the Import Settings dialogue.

:TRYALL with the argument * hangs APL+Win (not APL64)

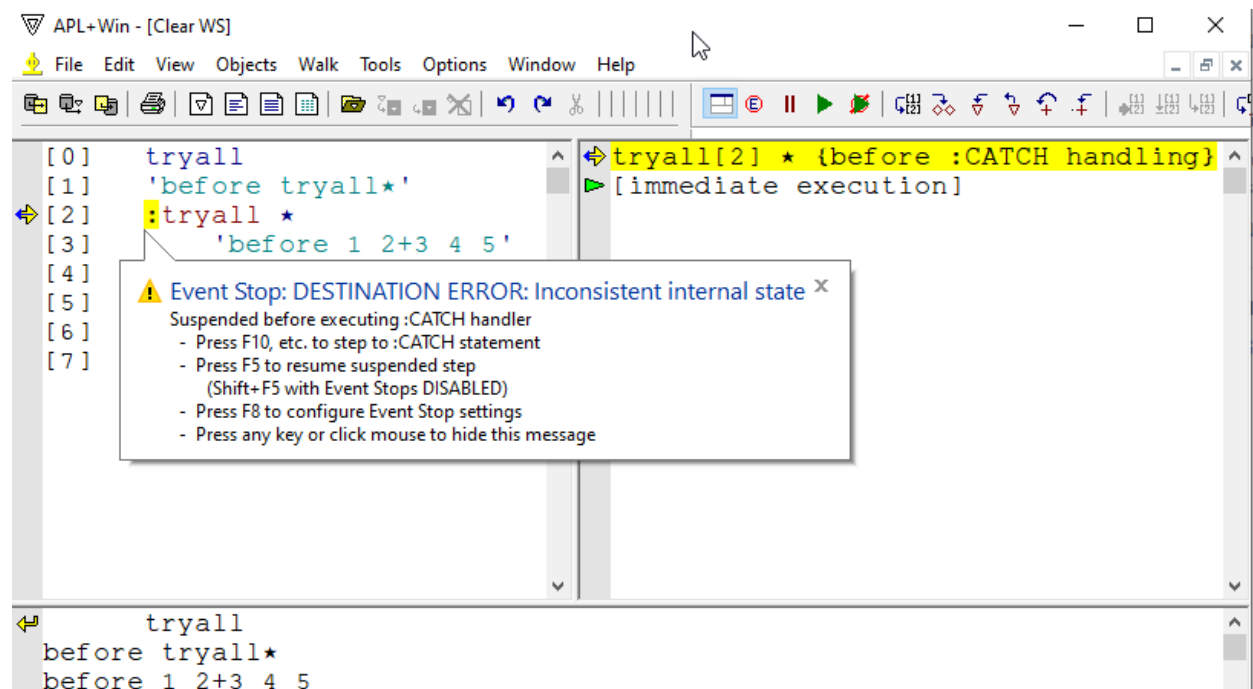
Consider the following function:

```

▽ tryall
[1] 'before tryall *'
[2] :tryall *
[3] 'before 1 2+3 4 5'
[4] 1 2+3 4 5
[5] 'after 1 2+3 4 5'
[6] :endtryall
[7] 'after tryall *'
▽

```

When you execute the tryall function without using the debugger, it hangs APL+Win. But when you step through the code in the debugger, you'll eventually end up here:



and each subsequent step you try to take replicates the previous DESTINATION ERROR (hence the hang). APL64 avoids this issue.

Evaluators' Comments & APLNow's Responses

This section contains the summary of the evaluators' comments (feedback) and APLNow's responses to the Pre-Production Version of APL64 (2020.0.2.19247). We would like to thank the evaluators for their contribution.

[Remove the button labeled New Row in the session command line](#)

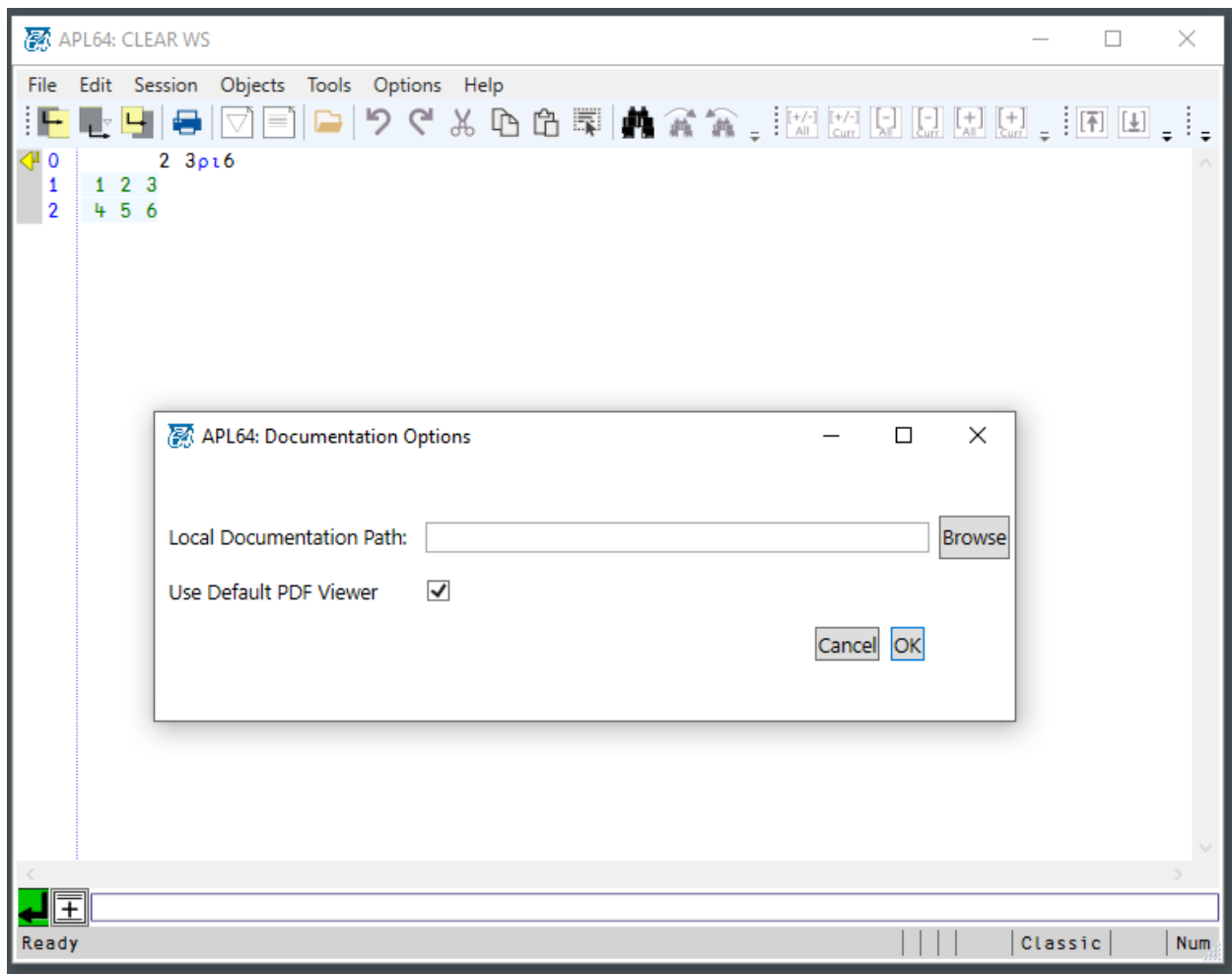
Thank you for your comment. There have been some internal discussions about this as well. We are considering some options on how to improve the layout of this button.

The online documentation's search/find window remains visible

The search/find window can remain visible when left opened when closing the online documentation window. This can also result in both the APL64 session and search/find window being unresponsive.

The matter has been reported to the developers of the Microsoft Edge WebView2 viewer that displays the online documentation viewer in APL64.

To use the default pdf viewer on the APL64 programmer's workstation, use the Help > Documentation Options dialogue and check the 'Use Default PDF Viewer' option in that dialogue. When this option is checked Microsoft Edge WebView2 viewer is not used by APL64.



!13 returns LIMIT ERROR

The fix will appear in the next pre-production version of APL64.

o[/θ freezes APL64 when executed

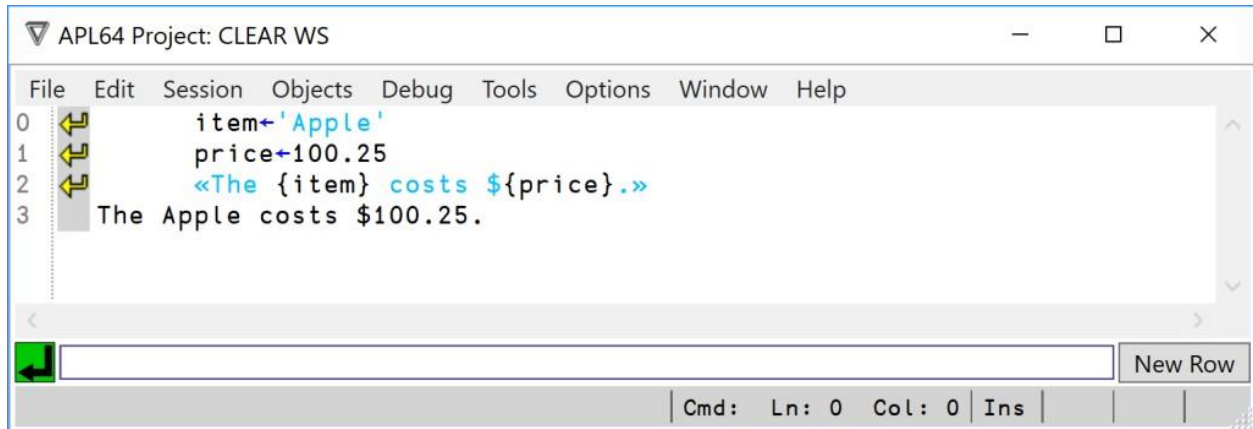
The fix will appear in the next pre-production version of APL64.

645 323 `⎕dr 1 2+00` returns (0 0) (1 1)

The fix will appear in the next pre-production version of APL64.

String substitution to simply text formatting does not work

The example on page 37 of APL64_PROJECT_TECHNICAL_DOCUMENT_SEPTEMBER_2018.pdf returned a SYNTAX ERROR instead of the result below:

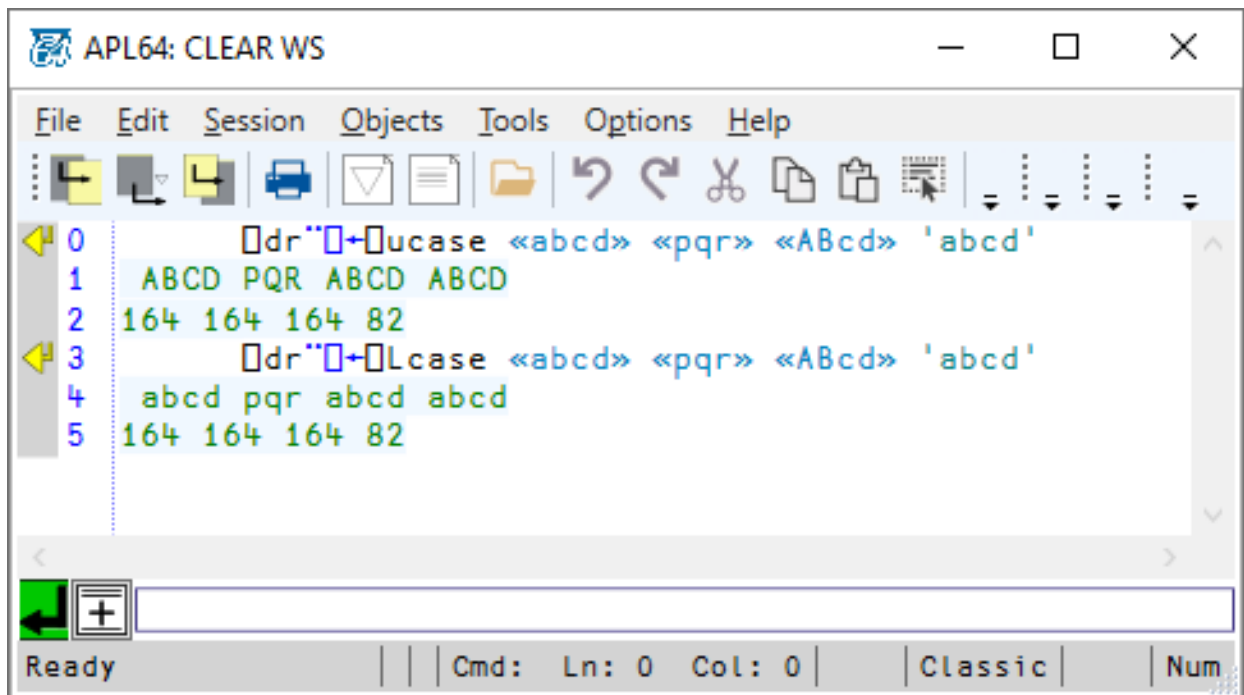


We are aware of it and will be addressed in a future APL64 pre-production version.

`⎕wcall 'CharUpper'` «APL64» aborted APL64

APL+Win does not support the string data type, so `⎕wcall 'CharUpper'`, which was specifically designed for APL+Win, does not support a string type argument. `⎕wcall` is based on Win32 technology, so in APL64, which is based on .Net technology, it is not anticipated that any modifications to `⎕wcall` will be made in APL64 except obviously avoiding a crash in this instance.

The new APL64 system functions `⎕ucase` and `⎕lcase` support character and string text:

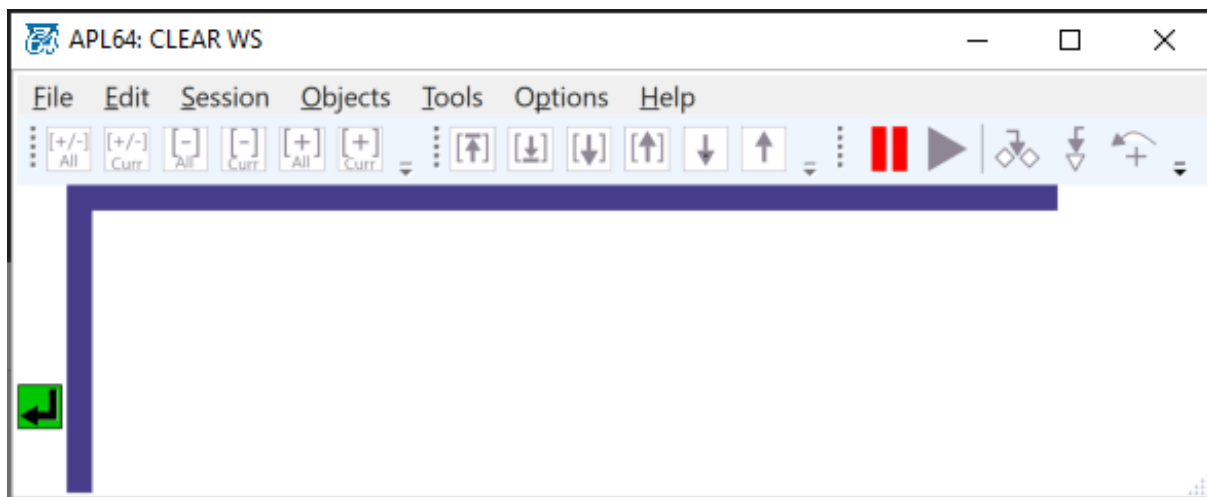


Different results for the examples for `⎕nlength` in the System Functions manual

We've compared the example(s) in the Adobe Acrobat file to the source material and have concluded that during the conversion all the leading and trailing blank spaces were replaced with a single blank space, unbeknownst to us. It will be investigated further and addressed in a future update.

The View Scale Entire GUI setting set to high at startup

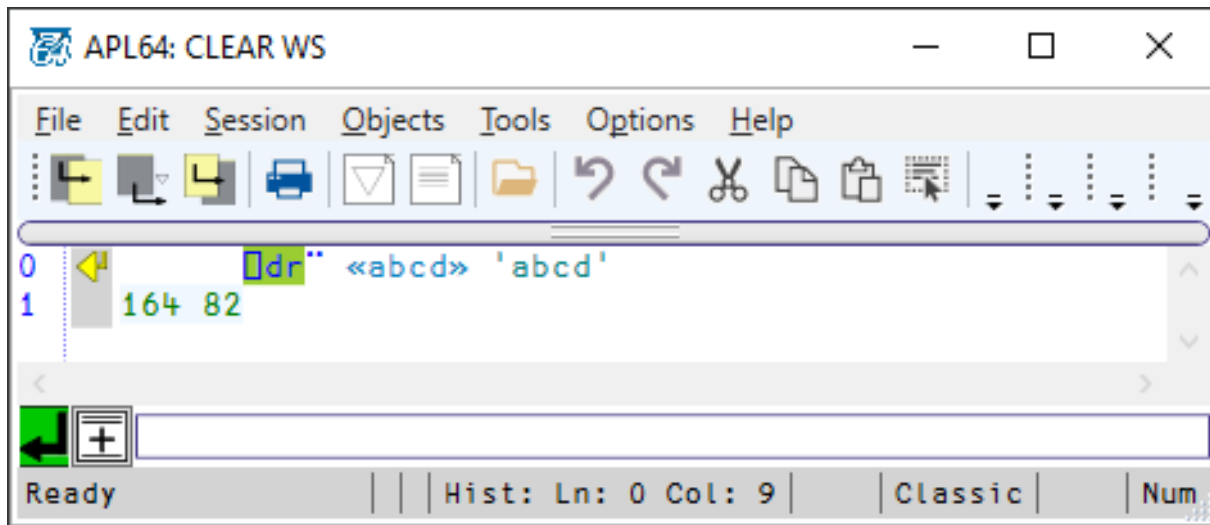
When the menu Session > View Scale Entire GUI setting was set too high at startup, APL64 displayed as followed:



This is addressed in the next pre-production version of APL64.

How can I identify the new String data type?

Use `⎕dr`. E.g.,



APL64_SYSTEM_FUNCTION.pdf appears to be missing `⎕a` and `⎕d`

Those system variables will appear in the System Variable document accessed from

Help > APL Language > System Variable in the next pre-production version.

A block select can also be replaced and not just deleted

The documentation has been updated to indicate that linear and rectangular selection within a rank 0, 1 or 2 text arrays also support replacement by paste or drag & drop.

When Block Select works with 2-d numeric arrays in the editor window, how will the replace functionality work?

Block select is unavailable with a numeric (nested or simple) array in the APL64 Editor window.

Please consider using the same jargon Block Select for Text Block in APL64

The documentation will be updated to indicate this. Furthermore, rectangular selections may also be made using Alt + Shift + MouseDrag and linear selections may also be made using Shift + MouseDrag.

The description: The Shift + Cursor keystrokes select a contiguous, linear block of text is incorrect. The key combination is Alt + Shift + Cursor.

The document is correct. There is a difference between a linear block of text and a rectangular block of text. The Alt + Shift + Cursor can select either type of text block, but Shift + Cursor can only select a linear block of text.

It's not possible to use 'EXT' `⎕NA` with core NET DLLs such as System.DLL

Correct. The Ext feature of `⎕NA` is designed so that an APL programmer may create a custom .Net Assembly class method and associate it with an APL64 user-defined function. The APL programmer created .Net Assembly class method can use any .Net methods, property, events within its definition.

A .Net method which is suitable for use with the Ext feature of □NA system function must have a specific signature:

```
public static Aval CustomMethodName(ExtContext context, Aval left, Aval right).
```

An overload .Net method may be APL programmer created to use a pre-existing .Net method which does not provide that signature. For example here is the text of such an overload method, `GetDateTimeNow()`, which provides the required Ext □NA compatible signature to use a pre-existing .Net method/property, `DateTime.Now`, which does not have the required signature:

```
using System;
```

```
namespace APLNext.DatetimeNow
```

```
{
    public class DateTimeNow
    {
        public static Aval GetDateTimeNow(ExtContext context, Aval left, Aval right)
        {
            var res = DateTime.Now;
            var aplCompatibleResult = new Int32[] { res.Year, res.Month, res.Day, res.Hour, res.Minute,
res.Second, res.Millisecond };
            var aplRes = new Aval(DT.Int, new Int32[] { 7 });
            aplRes.IntValues = aplCompatibleResult;
            return aplRes;
        }
    }
}
```

The EXT protocol provides for calling Monadic or Dyadic extension methods but at this time does not provide for a mechanism to call Niladic methods. So, when calling the APL □NA function, it would have to be called with a dummy argument such as $\emptyset$ or `⍬` or `⍬`.

[Editor session causes abnormal end when closed with Ctrl + Q](#)

This will be corrected in the next pre-production version of APL64.

[Can menu Edit > Select All be greyed out when the window that has focus is empty?](#)

The current behavior is by design and also consistent with other editors like in Notepad, Notepad++ and Visual Studio. So, it doesn't make sense practically for this to be changed.

[Executed Statement History dialogue](#)

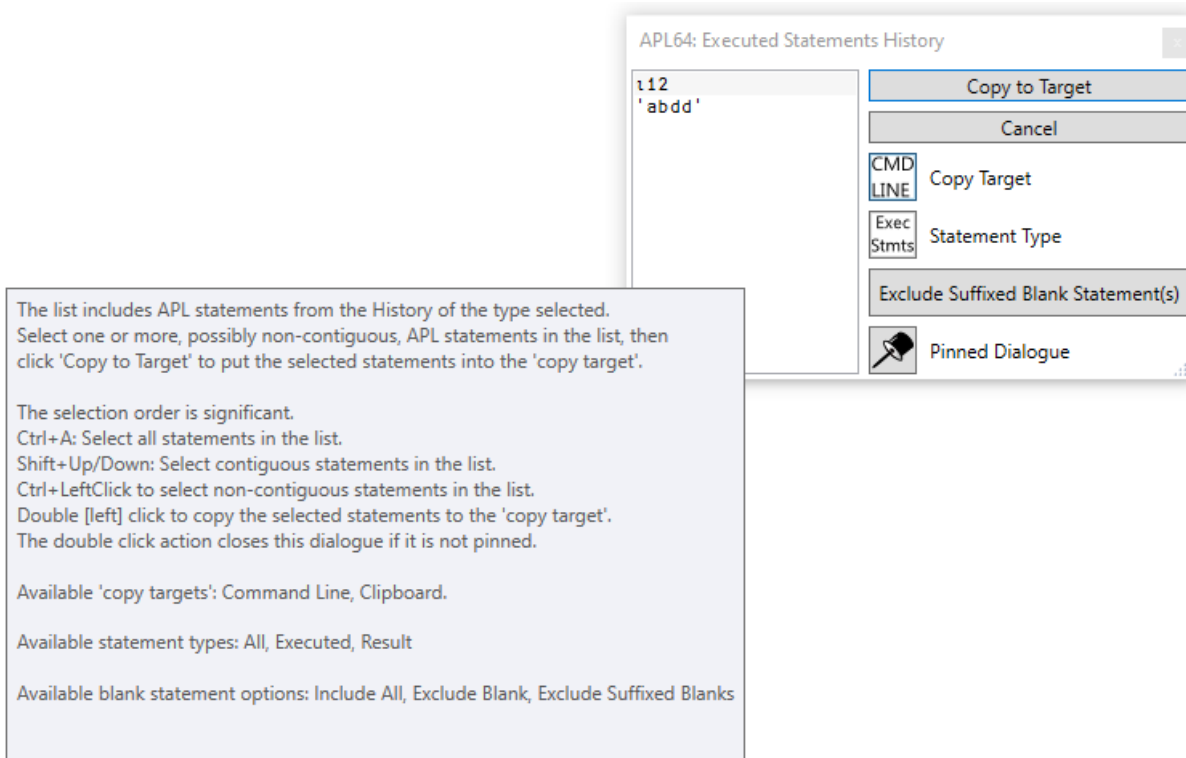
I understand Target to mean the window which has focus: this could be Debugger (paste does not make sense), or Editor or Command. But (1) pasting to debugger does not make sense and (2) Copy to Target always pastes to Command even when I have focus in Editor.

Response: Your understanding of the Copy Target is not correct. The Copy Target for this tool dialogue may be user-selected by clicking the toggle button to the left of the Copy Target label. The toggle button's caption will indicate the selected Copy Target as the command line or clipboard.

In this dialogue there are four toggle buttons:

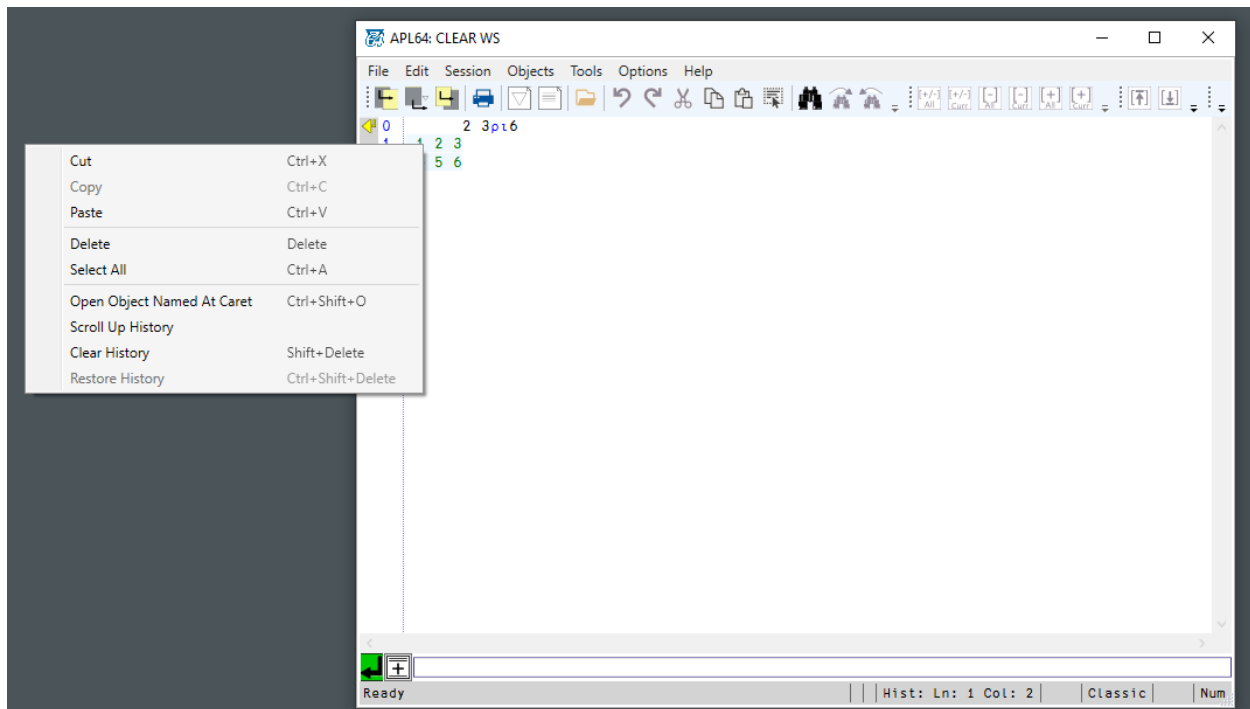
- Copy Target
- Statement Type
- Blank Statements
- Pinned Dialogue

Click each toggle button to review the options available to the user of this tool dialogue. The options for these toggle buttons are described in the tooltip for the statements portion of the dialogue:



The right-click menu should disable options that are unavailable within context

The Paste operation has been defined for the APL64 history pane, so that when the user pastes to the history pane, the result is inserted into the command line. It would be possible to provide an APL64 Session Option to disable this action as a user option.



Installing APL64 message: Source file not found...

That's a new one. Perhaps the installer was corrupted during or after it downloaded to your machine or maybe the antivirus/security software is intervening.

Clarification on the scope of :def ... :endef control structure statement

The function using :DEF is almost the same as this, using `⎕DEF` and localizing the same name as the outer function name:

```

⎕vr'ajay'
▽ ajay;ajay
[1]   + ⎕def 'r←ajay' 'r←10'
[2]   ajay
      ▽

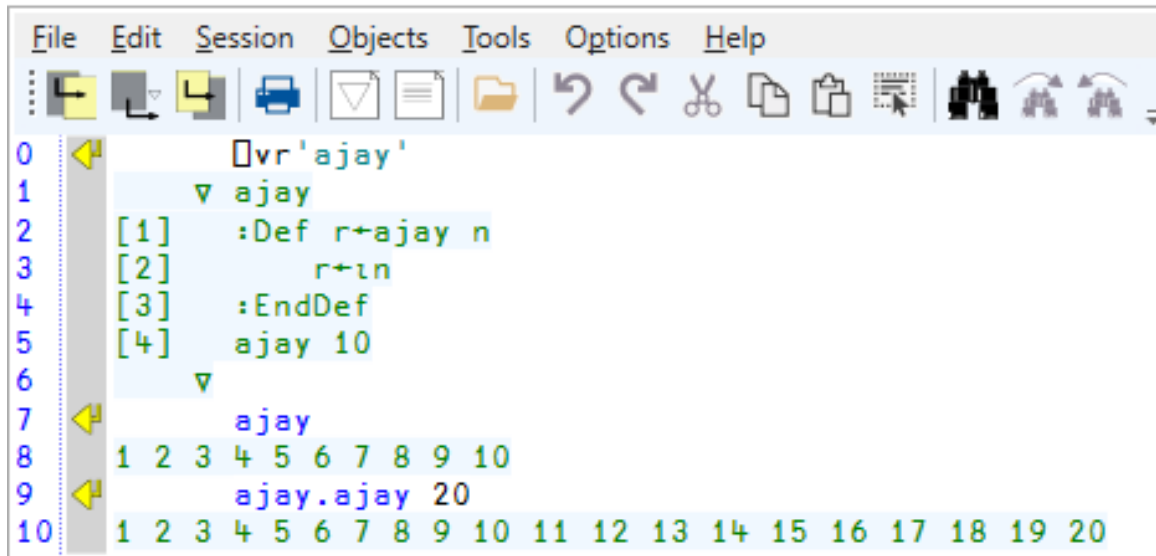
      ajay
1 2 3 4 5 6 7 8 9 10

```

Using :DEF creates a local function inside the containing function and implicitly localizes that name (so it doesn't need to appear in the locals list like we did in the APL+Win example above). So using :DEF doesn't have any special dangers that are different then localizing the containing function name and using `⎕DEF` or `⎕FX` to define a local function with the same name.

Now look at the :DEF version in APL64. In this case I modified the inner function to take an argument

APL64: CLEAR WS



```
0  ⍕vr 'ajay'
1      ∇ ajay
2  [1]  :Def r←ajay n
3  [2]      r←in
4  [3]  :EndDef
5  [4]  ajay 10
6      ∇
7  ⍕ajay
8  1 2 3 4 5 6 7 8 9 10
9  ⍕ajay.ajay 20
10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Note that we can call the parent function without an argument as before, but by using dot-notation we can also directly call the inner function with an argument.

If you neglect to localize the function name when you use `⍕DEF` to define a same-named inner function, that will overwrite the original function (big trouble). But with `:DEF` since the localization is implicit, you don't have that danger.

[⍕wi forms run out of process](#)

On running the function that creates the form, another icon, the APL+Win icon, appears on the taskbar. That suggests that with APL64, `⍕wi` forms run out of process. Is this correct?

Yes. The reason for this is that `⍕wi` forms are instantiated in `APLNow32.exe` that runs as an external process.

[Should focus go to the GUI form every time the function runs from APL64?](#)

Like in APL+Win, this depends on the method applied to the Windows form. With methods `Show` and `Wait`, the focus moves to the Windows form. But not with the method `Open`.

[Where is the I-Beam?](#)

Page 36 of `APL64_APL_PRIMITIVE.pdf` mentions I-Beam. Where is the symbol on the keyboard?

The I-Beam is unsupported in APL64. Therefore, it is being removed in the `APL64_APL_PRIMITIVE` document.

[Focus bounces between two function editor windows](#)

This is addressed in pre-production version 2020.0.3 of APL64.

How do you execute a 'EXT' ... □NA function that takes no arguments?

When the function does not require parameters (arguments), you must still supply an empty list enclosed in parentheses.

Issues with Clipboard with block select

Example:

1. End all APL64 Sessions
2. Start Notepad, type My Clipboard Content, Ctrl+A, Ctrl + V (OR you can select any text from anywhere).
3. Start APL64, press Ctrl + V (I get what I copied to the clipboard).
4. Execute (⌈12)∘.×⌈12 in APL64,, block select 2 3 4 6 (i.e. as 2 2p2 3 4 6) and press Ctrl + C
5. Press Ctrl + V (while you are still in the Session Window): What appears in the Command Window? (I do NOT get what I just copied!!)
6. Focus on the Command Window and clear its content using Backspace and press Ctrl + V (I get that I copied ie. what I block selected).

What you observed is the expected behavior. The APL64 action when pasting into the history pane is:

- (1) Capture the applicable history pane content
- (2) If there is no selection in the history pane, insert the clipboard content into the captured content at the caret position
- (3) If there is a selection in the history pane, replace the selection in the captured content with the clipboard content
- (4) Put the result into the command line, with consideration of the Session > Insert Text Into Command Line menu setting.

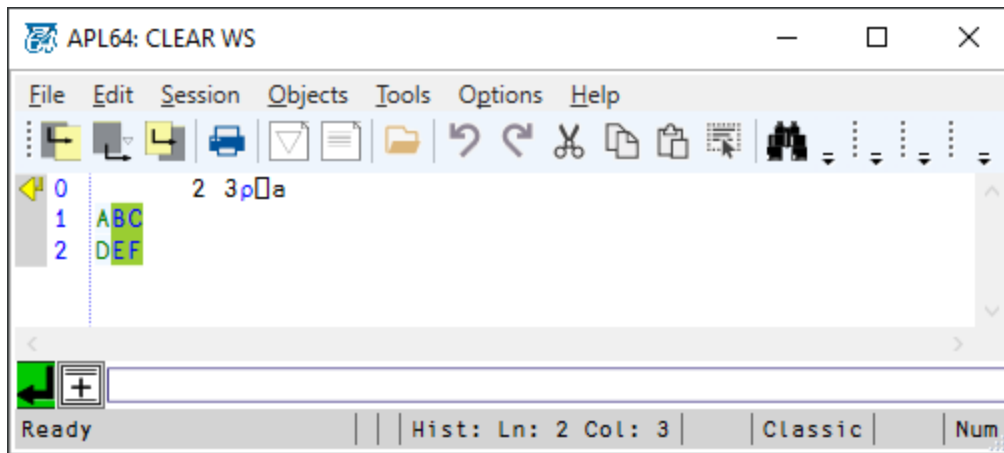
In your example:

- (a) You made a rectangular selection in the history pane
- (b) You copied the rectangular selection into the clipboard
- (c) The keyboard focus remained in the history pane, so the existing selection in the history pane continued to exist.
- (d) Now follow steps (1) - (4) above. to see that the clipboard content replaced the selection and the result was placed into the command line. Because the selection in the history pane was rectangular with two rows involved, there were two replacements of the clipboard content.

This is a simpler case where the clipboard content is only a single character:

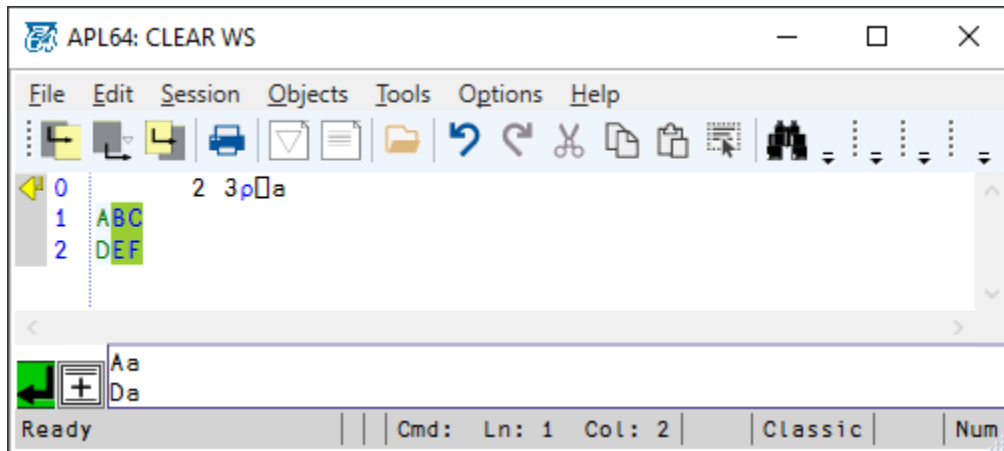
- (a) Put the character 'A' into the clipboard
- (b) Create the same history pane content as above
- (c) Make a rectangular selection in the history pane
- (d) Paste the clipboard content in to the history pane.

Before pasting 'a' into the rectangular selection:



After pasting 'a' into the rectangular selection:

- The history pane text is not modified
- The result of the paste is in the command line



The APL64 on-screen keyboard is NOT color-coded

This is addressed in pre-production version 2020.0.3. E.g.,



Additional information about this is available in the menu topic:
Help > Developer Version GUI > Keyboard Definition

Opening the keyboard Definition has the potential to lose active content

In pre-production version 2020.0.3, the following message box will appear prior to opening the APL64 Keyboard Definition option:

