

# APL64 Project – September 2020 Update

## Summary

The APL64 Project development team is continuing the development and testing of the new product. This document describes some of the APL64 Project progress since the [March 2020 Update](#).

- APL64 Project work is continuing with the full effort of the APL2000 team.
- Additional features needed for excellent compatibility with APL+Win have been implemented.
- New enhancements exclusive to the APL64 Project have been implemented.
- Testing of all project modifications are continuously performed.
- Testing by APL2000 personnel of the APL64 Project with existing APL+Win customer applications continues.

## Contents

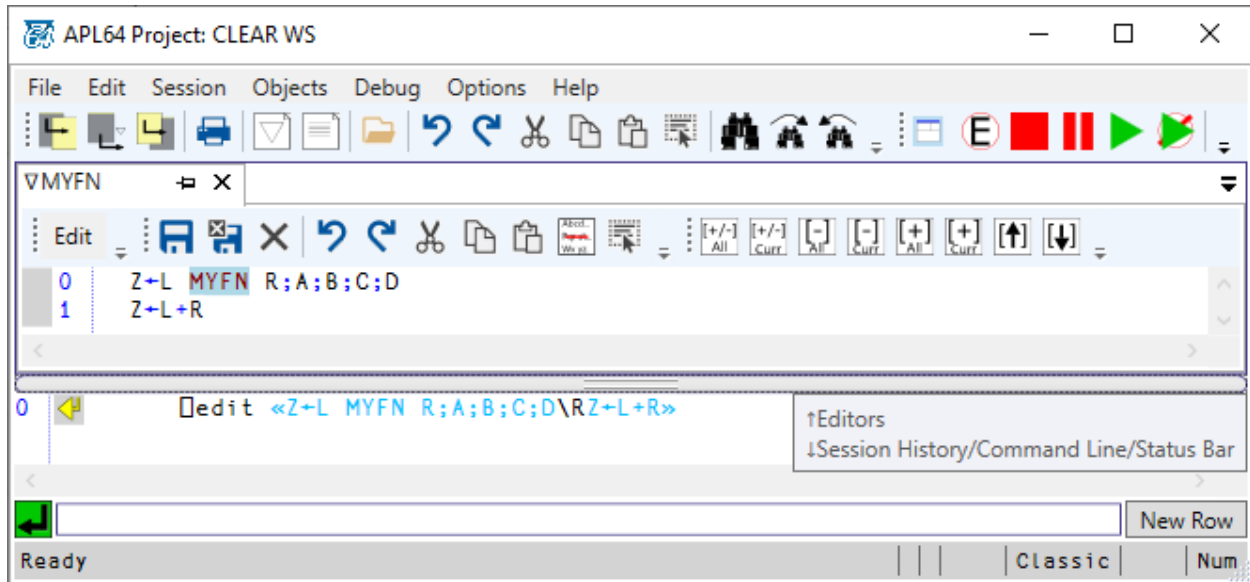
Summary .....	1
APL64 Project Session .....	3
Define Function using String Data Type with Embedded Escape Codes .....	3
Menu: Session > Scroll Up Session History .....	3
Menu: Objects > In Function Editor Indent New Line Based on Prior Line Indent .....	3
Objects Menu: Duplicate Editor Options .....	3
Prevent Duplicate Editors Option .....	3
Prompt Before Creating Duplicate Editor Session .....	3
Duplicate Function Editors Are Read Only .....	4
Session and Editor ⇄ Command Line Toggle .....	4
Updated Object Selector Dialogue: Filter Options: Exact, Contains, Begins With .....	4
Create Runtime Executable Utility .....	4
Debugged Function and State Indicator Panes are Independent of Main Window .....	6
Support for Silent Activation of APL64 Project Developer Version .....	6
New System Functions and Variables Exclusive to the APL64 Project .....	6
<input type="checkbox"/> DRIVEINFO System Function .....	6
<input type="checkbox"/> EDIT System Function .....	7
<input type="checkbox"/> EDITEVENTS System Function .....	8
<input type="checkbox"/> FINDINFILES System Function .....	8
<input type="checkbox"/> FN System Function .....	9

<input type="checkbox"/> L System Function .....	10
<input type="checkbox"/> OS System Function .....	10
<input type="checkbox"/> OVERV System Function.....	10
<input type="checkbox"/> SCOM System Function.....	11
<input type="checkbox"/> SKD System Function .....	12
<input type="checkbox"/> SPLASH .....	12
<input type="checkbox"/> STRING System Function.....	12
<input type="checkbox"/> SYSCONFIGFILE System Variable.....	12
<input type="checkbox"/> SYSINIT System Function.....	12
<input type="checkbox"/> TELPRINT System Function .....	13
<input type="checkbox"/> UCMDFILE System Variable.....	13
Updates to Existing System Functions and Variables .....	14
)CMDA System Command and <input type="checkbox"/> CMD System Function with Administrator Permissions .....	14
<input type="checkbox"/> SYS[33] System Variable Enhanced Control of <input type="checkbox"/> INBUF Clearing Behavior .....	14
<input type="checkbox"/> UCS Returns Unicode Code Points of Glyphs.....	15
<input type="checkbox"/> UCMD User Command Facility.....	15
<input type="checkbox"/> LIBD System Function.....	15

## APL64 Project Session

### Define Function using String Data Type with Embedded Escape Codes

The new string data type in the APL64 Project supports embedded escape codes designated with a backslash. For example, the `\R` in the example below is treated as a carriage return.



### Menu: Session > Scroll Up Session History

The result is analogous to the execution of the `⌈TCFF` system constant in the session in APL+Win. Also available in the context menu.

### Menu: Objects > In Function Editor Indent New Line Based on Prior Line Indent

When this option is checked, adding new lines to a function being edited will indent lines in a manner analogous to APL+Win.

### Objects Menu: Duplicate Editor Options

#### Prevent Duplicate Editors Option

When this menu item is checked, a duplicate editor will not be created and the existing editor will be selected. When this menu item is not checked, multiple editors of the same type and name can be created in an APL64 Project session.

#### Prompt Before Creating Duplicate Editor Session

This menu item is active only when 'Prevent Duplicate Editors' is not checked. If this menu item is checked, APL64 Project will prompt the user when an attempt is made to create a duplicate editor with the same name as an existing editor.

### Duplicate Function Editors Are Read Only

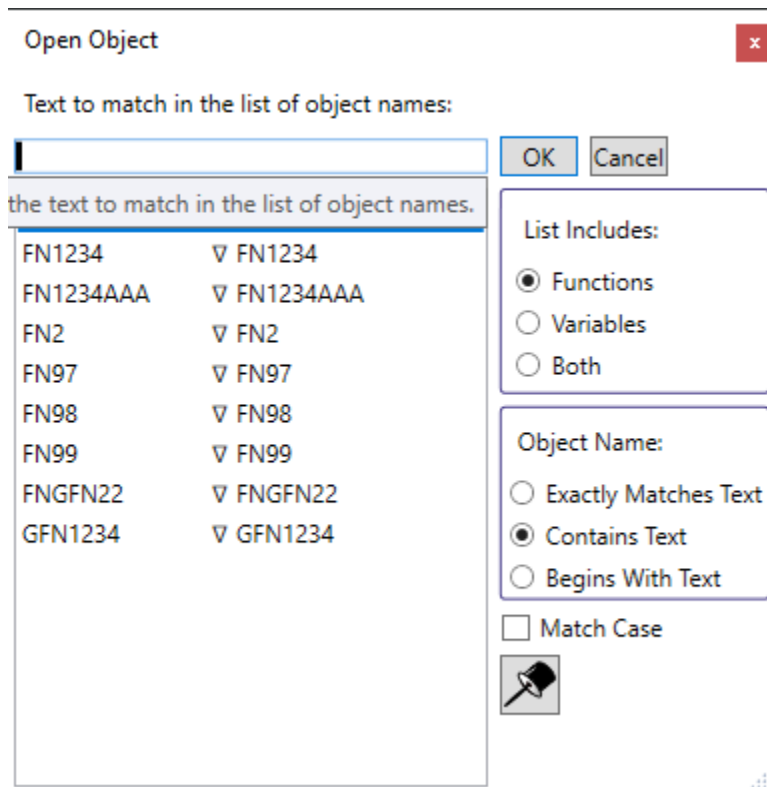
This menu item is active only when 'Prevent Duplicate Editors' is not checked. If this menu item is checked, duplicate function editors will be read only.

### Session and Editor ⇔ Command Line Toggle

The Ctrl+T keyboard shortcut has been implemented. This keyboard shortcut toggles between the Session and the recent editor session, or vice-versa.

### Updated Object Selector Dialogue: Filter Options: Exact, Contains, Begins With

The object selector dialogue is used by the Open Object (Ctrl+O) and the Fetch Object (Ctrl+G) dialogues.



### Create Runtime Executable Utility

There is no such thing as a runtime workspace in the APL64 Project, nor is there a runtime version of the APL64 Project in the same sense that they existed in APL+Win. Instead the system provides a way to package an application workspace and other supporting files into a custom stand-alone runtime executable (or Library) that runs your APL64 Project application.

The APL64 Project runtime feature is designed to make packaging and deployment of an APL64 Project - based Windows application easy for the programmer.

The runtime component of APL64 Project is bound to a runtime-ready APL64 Project workspace by this utility. The APL64 Project workspace ☐ LX value must contain the name of a user-defined APL64 Project function in the workspace that starts the APL programmer-developed application system.

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

A utility in the APL64 Project developer version is used to publish a single Windows operating system compatible .exe-format file containing:

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

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

This utility is accessed via the APL64 Project developer version Options > Create Runtime .Net Assembly > Create Runtime Executable... dialogue:

APL64 Project: Create Windows Runtime Executable

Runtime Workspace Filename:  Browse

EXECUTABLE CONTENT

Runtime Windows Executable Target Folder:  Browse ☒ Same as Workspace Folder

Runtime Executable Filename:  ☒ Same as Runtime Workspace Name

APL64 Config File:  Browse

☐ Include APLNow32 Configuration Components

Additional Files Required for the Application

Source Path	Base Target Path	Target Path Suffix	Overwrite
<input type="button" value="Add Required Files to Runtime Executable"/>			

ASSEMBLY META-DATA

File Description:

Product Name:

Copyright:

File Version:

Version:

CompanyName:

App Description:

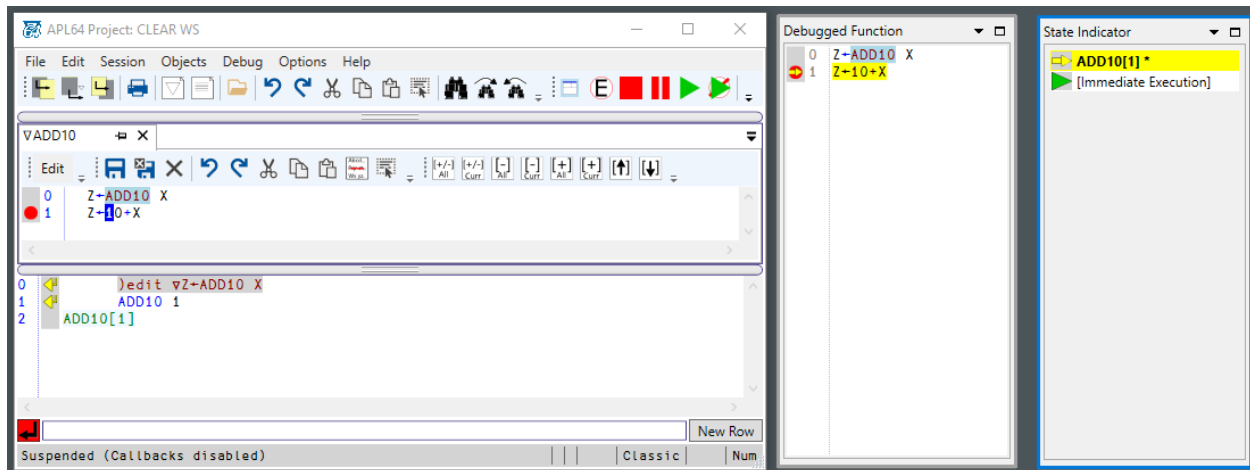
Assembly Version:

Neutral Language:

Description:

## Debugged Function and State Indicator Panes are Independent of Main Window

The debugged function pane and the state indicator panes may now be independently 'floated' separate from the main APL64 Project 'session' window:



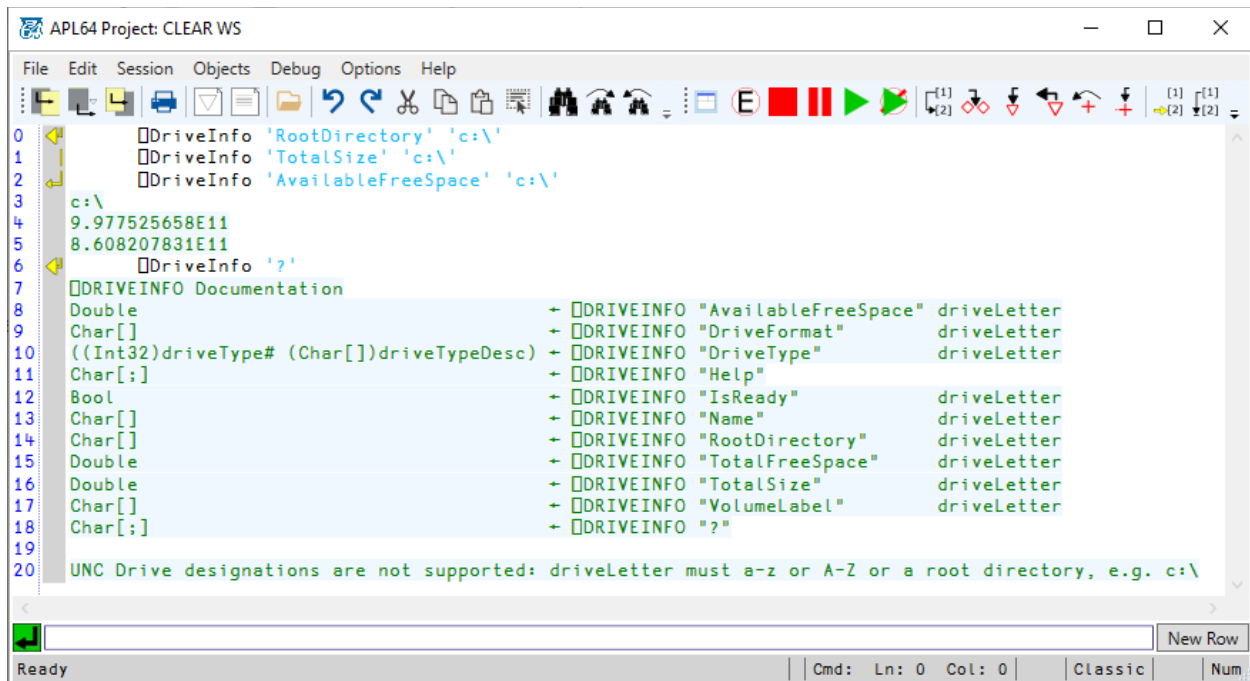
## Support for Silent Activation of APL64 Project Developer Version

The ACTIVATIONLICENSEID and ACTIVATIONLICENSEPASSWORD command line arguments, with abbreviations ALI and ALP, may be provided by an enterprise customer to silently activate the APL64 Project developer version.

## New System Functions and Variables Exclusive to the APL64 Project

### ☐ DRIVEINFO System Function

This system function returns information for local drives. It is available in the developer and runtime versions of the APL64 Project; UNC drives are not supported.



## ☐ EDIT System Function

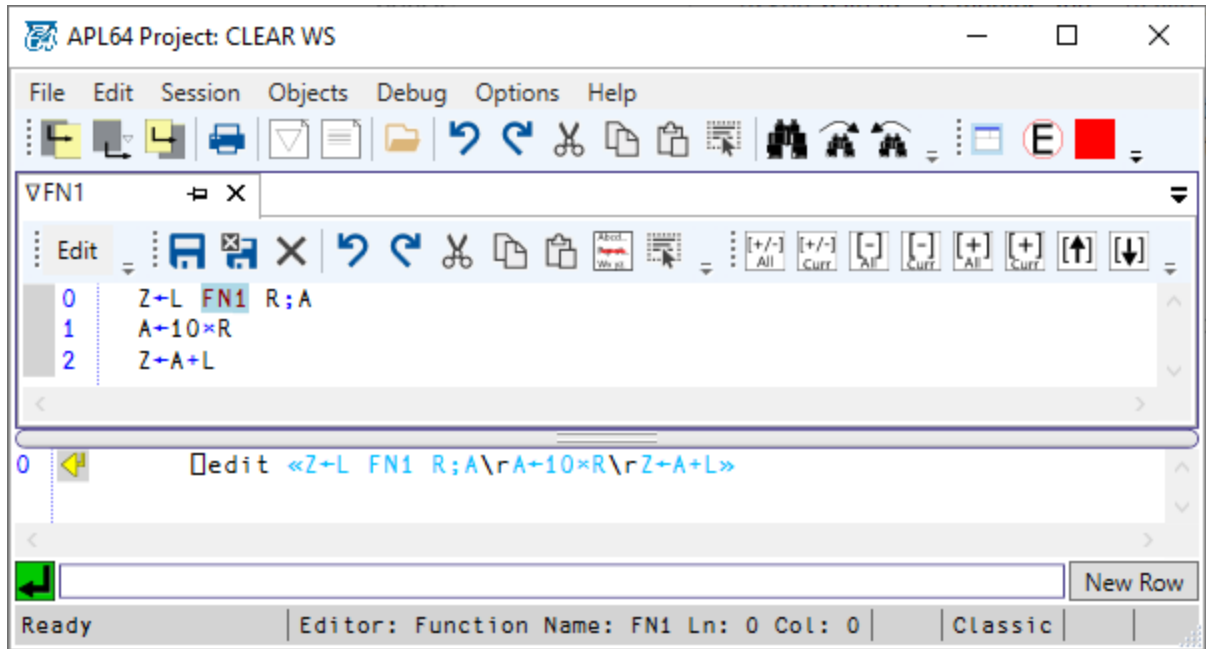
This system function invokes an edit session where you can modify a function, character vector, character matrix, or numeric variable. It is available only in the developer version of the APL64 Project as the runtime version of the APL64 Project does not provide a programmer GUI. [This is similar to the ☐ EDIT system function in both APLP\*LUS PC and APL\*PLUS II/386.]

☐ EDIT has no result.

Syntax: ☐ EDIT R

R: Right Argument

- 1<sup>st</sup> Element: Name of the object to be edited. The object need not exist.
- 2<sup>nd</sup> Element: Optional: Two element integer vector containing the initial caret position in a function editor containing an existing function: (row col) index origin 0.
- 3<sup>rd</sup> Element: Optional: Five-element modal editor specifications vector which, when present, will cause a modal editor window to be presented. If this element is not present, a non-modal editor pane will be presented.
  - Modal Editor Window Height
  - Modal Editor Window Width
  - Modal Editor Window Top
  - Modal Editor Window Left
  - Modal Editor Window Maximized: 0/No (Default) 1/Yes
- Since the APL64 Project string data type supports 'escaped' elements, new-line-separated, multi-line function definitions may be used as the 1<sup>st</sup> argument to the ☐ EDIT system function:



## □ EDITEVENTS System Function

This system function can be used to subscribe to specified events fired when selecting object names or editing objects.

□ EDITEVENTS has no result.

Syntax: subscribe □ EDITEVENTS eventName aplEhFnName

Left argument:

- Subscribe: 0/Unsubscribe 1/Subscribe: The unsubscribe action has no effect if the aplEhFnName has not subscribed to the eventName event.

Right Arguments:

- eventName: Selected from:
  - EditEnded: Fired when an editor is closed
  - EditSaved: Fired when an editor's content is saved.
  - EditStarted: Fired when an editor is opened.
  - ObjectNameSelected: Fired when an object name is selected in the Open or Fetch dialogue.
  - PreviewObjectNameSelection: Fired when the Open or Fetch dialogue is opened before an object name is selected
- aplEhFnName: name of the APL, user-defined function which will be executed when the event with eventName is fired.

## □ FINDINFILES System Function

This system function searches for specified text in filtered files in a folder and optionally files in its subfolders.

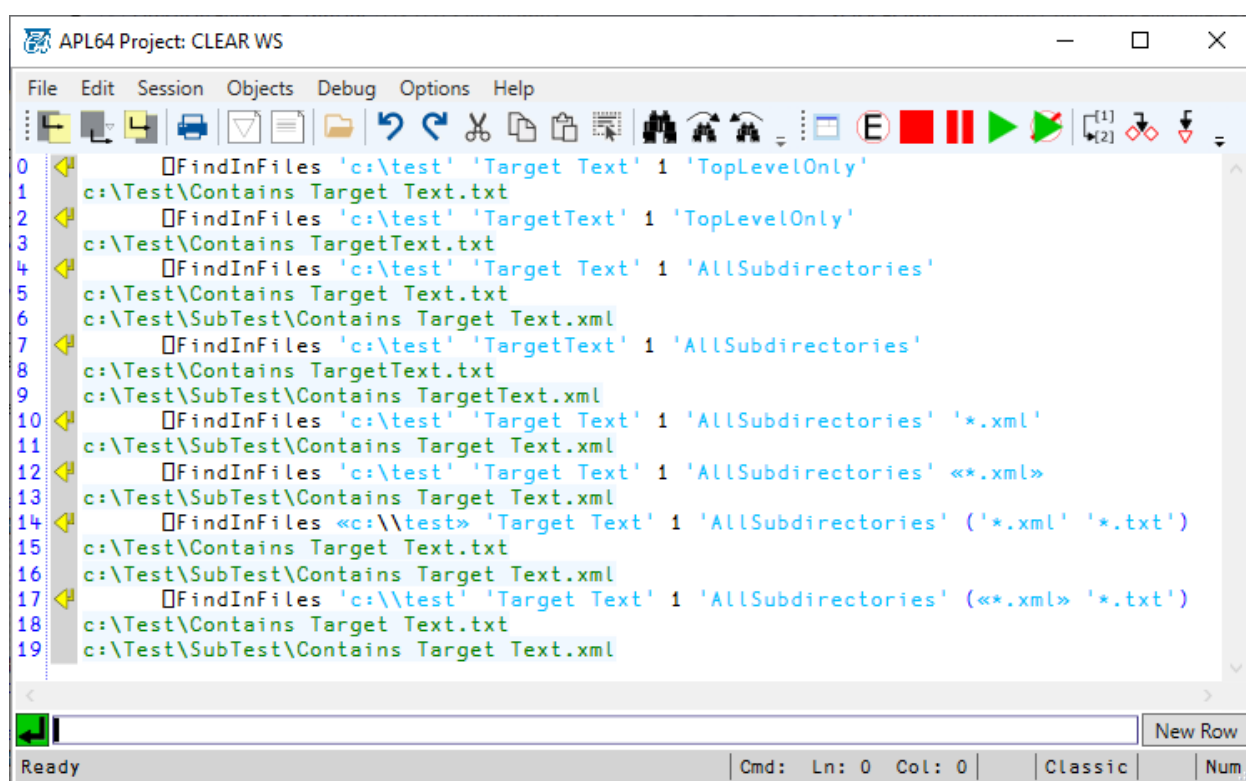
Syntax: □ FINDINFILES targetFolder targetText ignoreCase ignoreSubdirectories [fileWildCards]



Arguments:

- targetFolder: File system folder containing files to be considered in the search
- targetText: Text to find in the contents of the files considered in the search
- ignoreCase: 0/True 1/False
- ignoreSubdirectories: 0/1 or case-insensitive character vector or string scalar 'False'/'True'
- fileWildCards: Optional file name wild card patterns:
  - Character vector
  - String scalar or vector
  - Vector of character vectors and string scalars

Result: Character matrix containing the file names, if any, of files containing the targetText



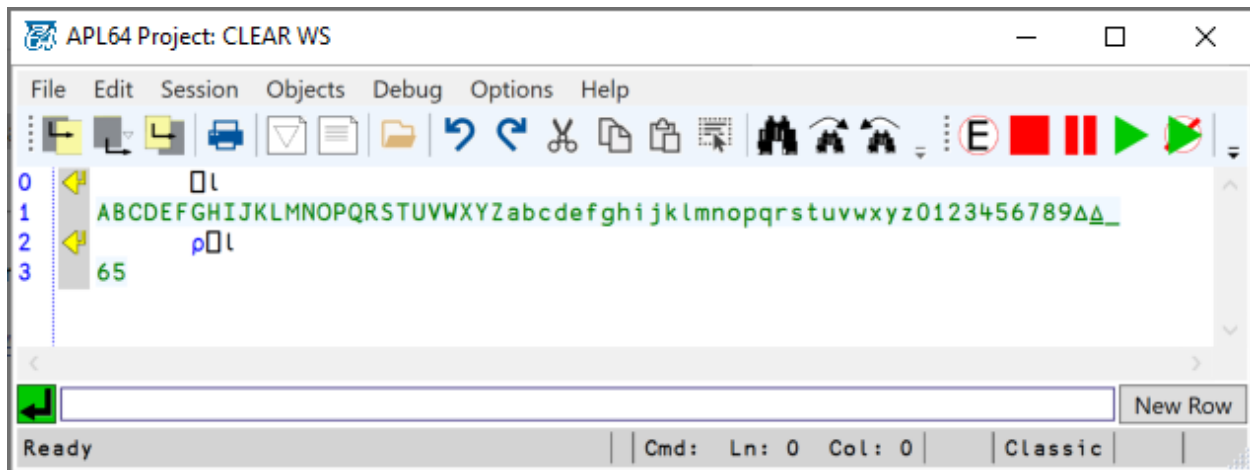
```
0  FindInFiles 'c:\test' 'Target Text' 1 'TopLevelOnly'
1  c:\Test\Contains Target Text.txt
2  FindInFiles 'c:\test' 'TargetText' 1 'TopLevelOnly'
3  c:\Test\Contains TargetText.txt
4  FindInFiles 'c:\test' 'Target Text' 1 'AllSubdirectories'
5  c:\Test\Contains Target Text.txt
6  c:\Test\SubTest\Contains Target Text.xml
7  FindInFiles 'c:\test' 'TargetText' 1 'AllSubdirectories'
8  c:\Test\Contains TargetText.txt
9  c:\Test\SubTest\Contains TargetText.xml
10 FindInFiles 'c:\test' 'Target Text' 1 'AllSubdirectories' '*.xml'
11 c:\Test\SubTest\Contains Target Text.xml
12 FindInFiles 'c:\test' 'Target Text' 1 'AllSubdirectories' «*.xml»
13 c:\Test\SubTest\Contains Target Text.xml
14 FindInFiles «c:\\test» 'Target Text' 1 'AllSubdirectories' ('*.xml' '*.txt')
15 c:\Test\Contains Target Text.txt
16 c:\Test\SubTest\Contains Target Text.xml
17 FindInFiles 'c:\\test' 'Target Text' 1 'AllSubdirectories' («*.xml» '*.txt')
18 c:\Test\Contains Target Text.txt
19 c:\Test\SubTest\Contains Target Text.xml
```

## FN System Function

This system function returns the name of the currently executing (user-defined) function. This is more efficient than querying the topmost function name `SI` stack (State Indicator). The result is empty for a clear `SI` stack.

## L System Function

This system function returns all characters that can be part of an APL identifier.

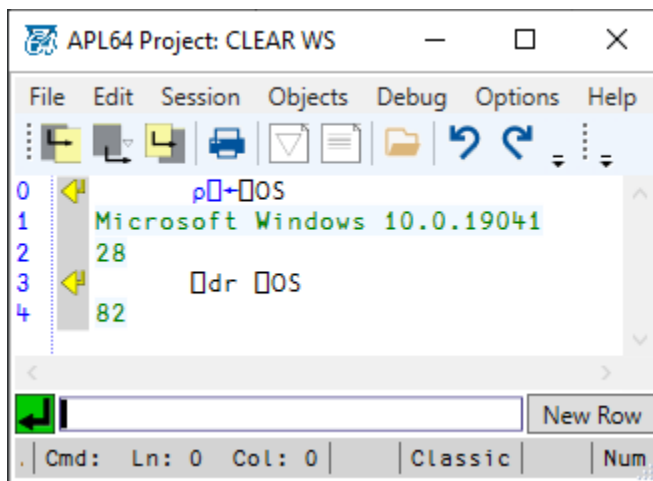


```
0  ⍵
1  ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_
2  ⍵
3  65
```

The screenshot shows the APL64 Project: CLEAR WS window. The code editor displays the L system function being executed. The result is a character vector containing all valid APL identifier characters: uppercase and lowercase letters, digits, and underscore. The result is shown on line 1, and the character count (65) is shown on line 3. The status bar at the bottom indicates 'Ready', 'Cmd: Ln: 0 Col: 0', and 'Classic' mode.

## OS System Function

This system function returns the name and version number of the host operating system.



```
0  ⍵
1  Microsoft Windows 10.0.19041
2  28
3  ⍵
4  82
```

The screenshot shows the APL64 Project: CLEAR WS window. The code editor displays the OS system function being executed. The result is a character vector containing the operating system name and version number: 'Microsoft Windows 10.0.19041'. The result is shown on line 1, and the character count (82) is shown on line 4. The status bar at the bottom indicates 'Cmd: Ln: 0 Col: 0', 'Classic' mode, and 'Num'.

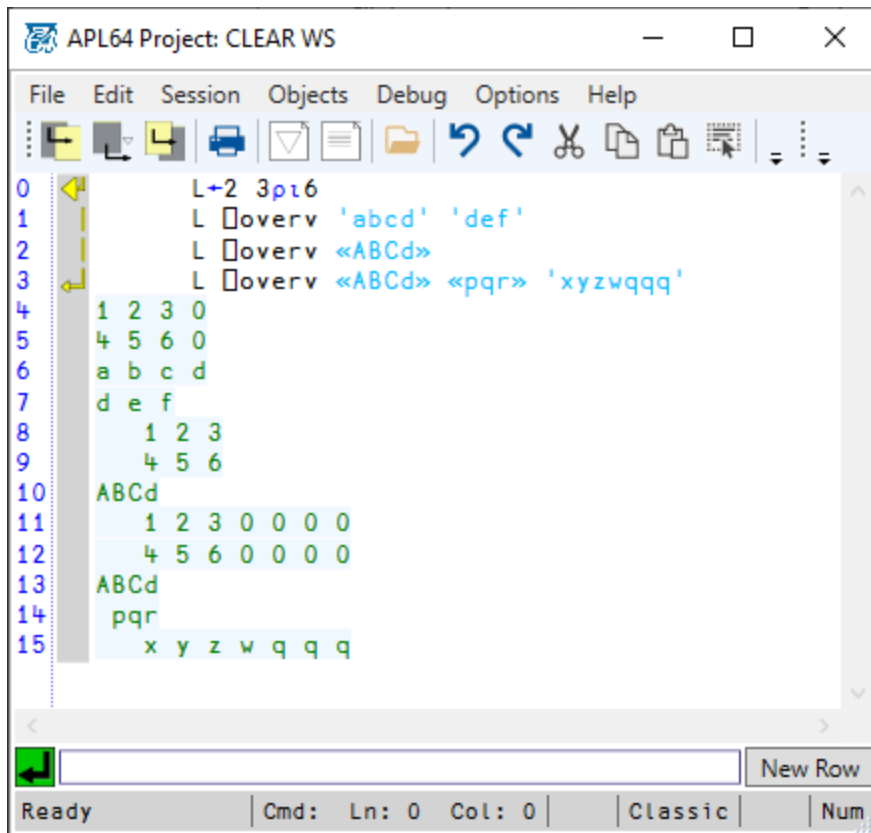
## OVERV System Function

This system function laminates the right argument under the left argument.

Syntax; L  OVERV R

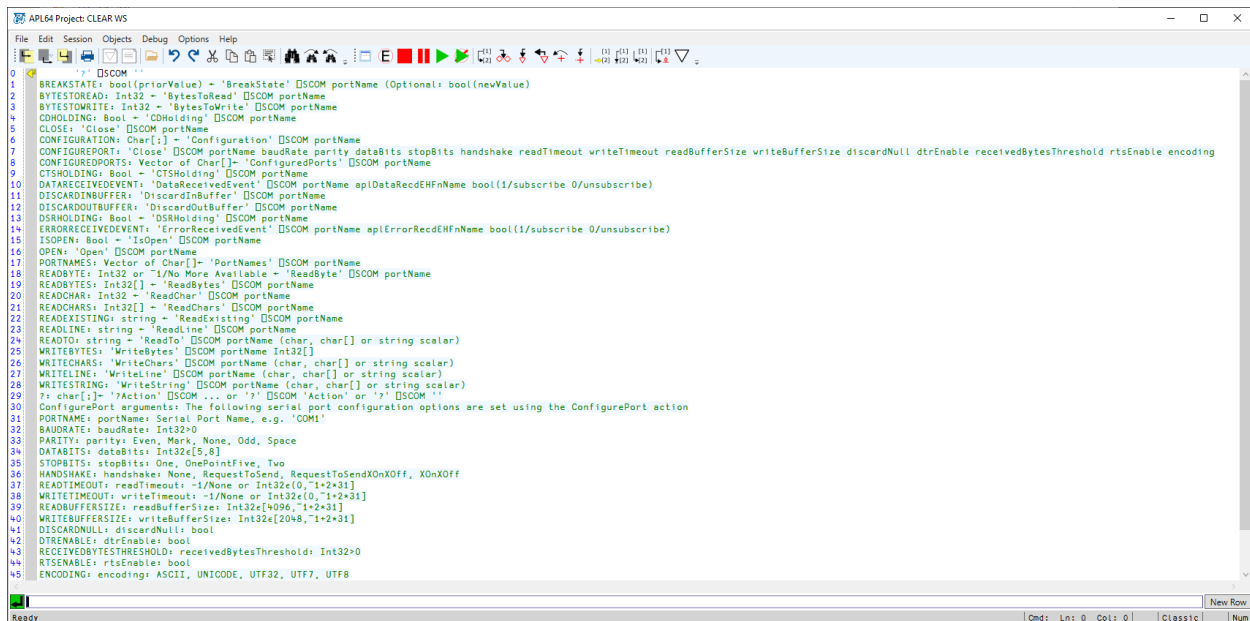
Left Argument (L): Any APL object or none

Right Argument (R): Any APL object of rank 0 or rank 1



## ☐ SCOM System Function

The ☐ SCOM system function enables [RS-232](#) transmission of data via the [.Net Serial Port class](#) in the APL64 Project. ☐ SCOM exposes the methods, properties and events of the .Net Serial Port class. ☐ SCOM is an enhanced version of the ☐ ARBIN system function in previous APL2000 products.



## □ SKD System Function

The □ SKD (string key dictionary) system function is used to create and use .Net dictionary instances in the APL64 Project. An □ SKD dictionary instance is a memory-based, keyed-array containing <key, value> pairs. A key is an APL64 Project string scalar. A value is an arbitrary APL64 Project object. Significant performance benefits may accrue when the same dictionary is to be searched multiple times in the same APL64 Project instance.

## □ SPLASH

This system function is used to close the user-provided splash screen (BMP, GIF, JPG) displayed at the start of the APL64 Project initiated from one of the command line arguments:

Command Line Argument (Abbrev.)	Data Type	Action
SplashImagePath (SIP)	Text	Path to splash image to display
SplashImageCenter (SIC)	Bool	Splash image centered on monitor ensemble
SplashImageTop (SIT)	Bool	Splash image is the topmost window

## □ STRING System Function

The .Net string data type is supported in the APL64 Project. The □ STRING system function provides access to the [.Net String class](#) properties and methods: COMPARE, COMPAREORDINAL, CONCAT, CONTAINS, ENDSWITH, EQUALS, FORMAT, GETHASHCODE, INDEXOF, INDEXOFANY, INSERT, ISNORMALIZED, ISNULLOREMPTY, ISNULLORWHITESPACE, JOIN, LASTINDEXOF, LASTINDEXOFANY, LENGTH, NORMALIZE, PADLEFT, PADRIGHT, REMOVE, REPLACE, SPLIT, STARTSWITH, SUBSTRING, TOCHARARRAY, TOLOWER, TOLOWERINVARIANT, TOUPPER, TOUPPERINVARIANT, TRIM, TRIMEND, TRIMSTART.

## □ SYSCONFIGFILE System Variable

This system variable returns the full path of the APL64 Project xml-format configuration file which was specified in the APL64 Project command line or if none, the default location of this configuration file. This system variable is the APL64 Project analogue of the APL32 □ WCALL 'W\_IniFile' method. The APL32 component of the APL64 Project provides support for □ WI features. □ WCALL 'W\_IniFile' returns the full path of the APL32 ini-format configuration file which was specified in the APL64 Project command line or if none, the default location of this configuration file.

## □ SYSINIT System Function

This niladic system function returns a vector with elements obtained from the .Net environment on which the APL64 Project runs. See below for a description of the elements:

Element#	Datatype	Description
0	Int32	Current process: Handle
1	Int32	Reserved
2	Char[]	Command line used to start APL64 Project executable
3	Int32	Reserved
4	Char[]	Current process: Name, e.g. "APL64Project"
5	Double	Current process: VirtualMemorySize64 (bytes)
6	Double	Current process: NonpagedSystemMemorySize64 (bytes)

7	Double	Current process: PrivateMemorySize64 (bytes)
8	Double	System.GC.GetTotalMemory (true)
9	String[]	Parsed command line arguments

### □ TELPRINT System Function

This system function reshapes a character matrix into multiple columns of characters. This serves as a replacement for the previous ∇TELPRINT function in APL+Win that executed the □CALL system function that is not supported in the APL64 Project.

Syntax: table←[maxwidth (Default □PW) □TELPRINT charmat

### □ UCMDFILE System Variable

This system variable returns the base path to the primary user command file. The path is set in the command line argument “UCMDFILE” or with the abbreviation “UCF”.

## Updates to Existing System Functions and Variables

### )CMDA System Command and □CMD System Function with Administrator Permissions

The new )CMDA system command was added to launch a DOS command window with elevated Administrator permissions (similar to Windows' "Run as administrator." option). Similarly, new left arguments, 4-7, were added to the □CMD system function to run with elevated Administrator permissions. See below for a description of the new values:

Values	Description
0	APL continues
1	DOS executes the command; APL waits until DOS completes
2	APL continues without waiting; DOS executes with no window
3	DOS executes with no window; APL waits until DOS finishes
4	APL continues (elevated Administrator permission). See Note.
5	DOS executes the command; APL waits until DOS completes (elevated Administrator permission) See Note.
6	APL continues without waiting; DOS executes with no window (elevated Administrator permission) See Note.
7	DOS executes with no window; APL waits until DOS finishes (elevated Administrator permission) See Note.

Note: You should see the UAC prompt asking for permission to run the command with administrator permissions provided UAC prompting isn't disabled in Windows' User Account Control Settings.

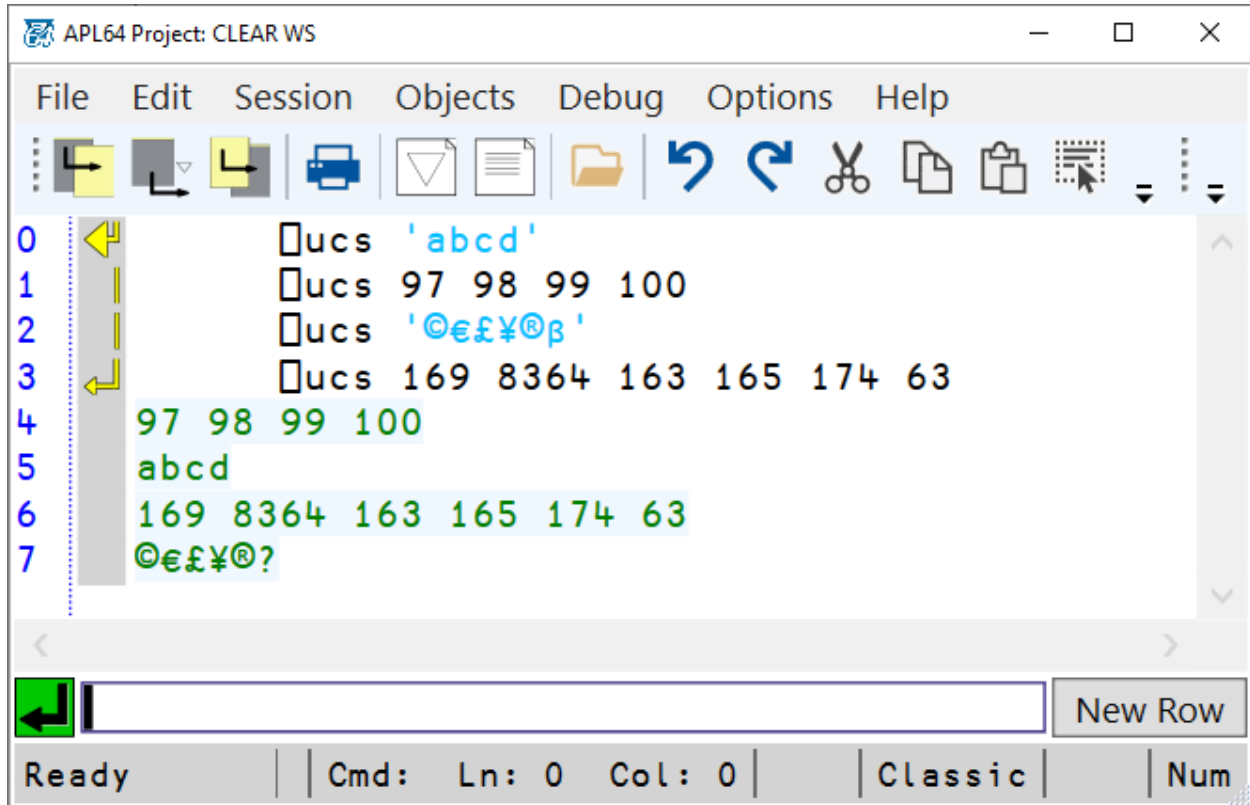
### □SYS[33] System Variable Enhanced Control of □INBUF Clearing Behavior

This system variable has a new element (33) to control error suspensions and □INBUF data clearing. See below for a description of the new values:

Value	□INBUF Clearing Behavior
0	APL+Win compatibility behavior: Do not clear due to exceptions or □ input evaluation errors
1	Clear due to exceptions or □ input evaluation errors
2	Similar to option '1', but when □INBUF is used in an □ELX or □ALX handler, does not clear if the handler suspends; so continuation of □INBUF actions are possible

### ☐ UCS Returns Unicode Code Points of Glyphs

This system function has been implemented and is compatible with APL+Win functionality. But new in the APL64 Project, the monadic case can now return the decimal Unicode code points of specified glyphs. As in APL+Win, the inverse operation is supported up to the limits of the current font.



### ☐ UCMD User Command Facility

The user command facility of the APL64 Project has been implemented and is compatible with APL+Win functionality. Note: Uses of the ☐CALL system function user command source code have been updated with the applicable new APL64 Project system functions. See the description for ☐UCMDFILE System Variable earlier in this document for additional information.

### ☐ LIBD System Function

This system function has been implemented and is compatible with APL+Win functionality.

APL2000 will continue to provide regular updates on the APL64 Project progress on the [APL2000 Forum](#). Contact [sales@apl2000.com](mailto:sales@apl2000.com) or call 301-208-7150 with questions or comments.