APL64 2025.0.9 Update

Overview

This document describes the enhancements and bug fixes in the APL64 2025.0.9 Update. Enjoy the new APL64 features. Thanks to the APL64 programmers who provide valuable feedback to the APL64 development team.

Interpreter Modifications

New ⎕XL Actions

RenameWorksheet action

The RenameWorksheet action renames the specified worksheet with the specified name.

AddHyperlink action

The AddHyperlink action removes any previous hyperlink in a specified cell and add a new hyperlink.

RemoveHyperlink action

The RemoveHyperlink action removes the hyperlink from a cell or removes the hyperlinks from a worksheet or workbook.

For more information on these new actions, refer to **Help | APL Language | Using ⎕XL**.

New ⎕PDIST System Function

The APL64 ⎕PDIST system function is an interface to the MathNet Probability and MathNet Random toolkits. The ⎕PDIST system function provides the Samples action to obtain sample values for many probability distributions using programmer-selected pseudo-random number generators. Refer to **Help | APL Language | Using ⎕PDIST** menu for more information.

Pseudo-random Number Generators

Several pseudo-random number generators are available including MersenneTwister, XorShift, Mcg###, Whichmann-Hill. Random 'seed' options are also available for these generators.

Continuous, Discrete and Multi-variate Distributions

Obtain distribution samples and distribution properties.

DBNull for ⎕SQL/⎕SQLite/⎕SQLMY/⎕SQLDB2: New DBNull action

DBNull returns a value used to specify null values for the ExecInsertQuery and ExecStoredProc actions. DBNull value will also be used as the APL64 value associated with a DBNull value returned from a GetRecord, GetAllRecords, or ExecStoredProc action.

New ⎕ROUND System Function

The ⎕round system function provides several .Net number rounding options. Refer to **Help | APL Language | System Functions** for more information.

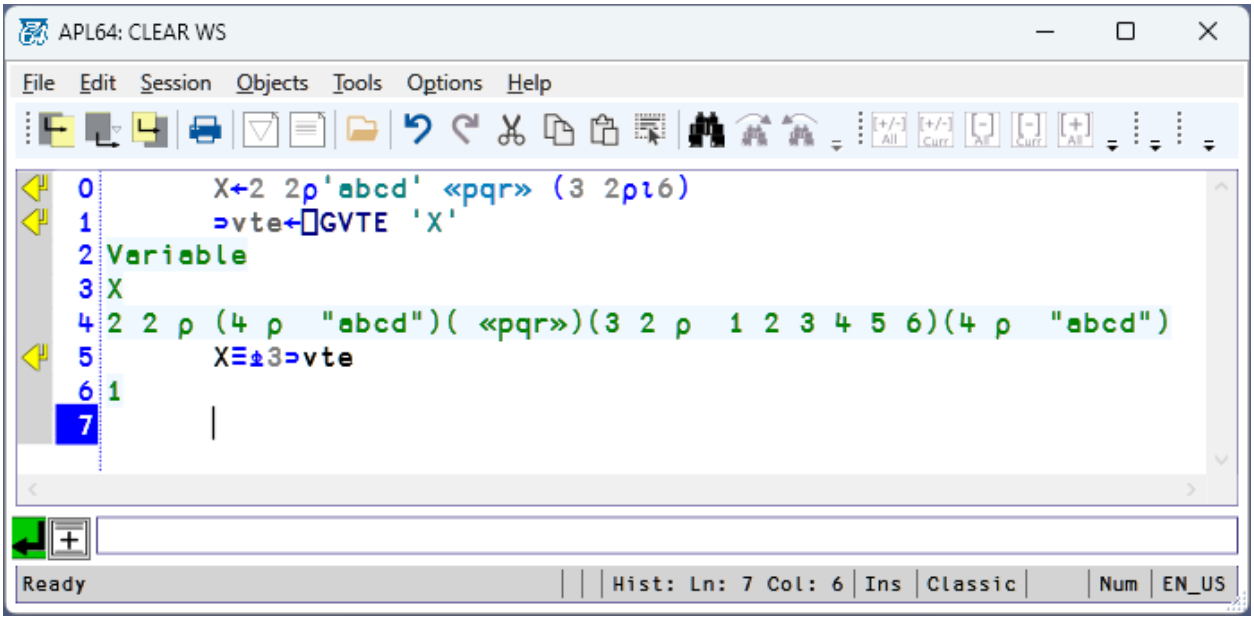
Rounding methods available include: MidpointRounding.ToEven, MidpointRounding.AwayFromZero, MidpointRounding.ToZero, MidpointRounding.ToNegativeInfinity, MidpointRounding.ToPositiveInfinity.

New ⎕CFLOCK and ⎕CFFLUSH system functions

The ⎕cfflush system function flushes the file buffers for a tied colossal component file. Refer to **Help | APL Language | System Functions** for more information.

The ⎕cflock system function performs a file lock and unlock for a colossal component file. Refer to **Help | APL Language | System Functions** for more information.

New: Use ⎕GVTE to obtain an executable 'recipe' for an APL64 variable



⎕NFE: Native files with Encoding

New ⎕NFE 'GetLength' action

The GetLength action returns the length of the specified file in bytes.

New ⎕NFE 'IsReadOnly' action

The IsReadOnly action will return a Boolean scalar indicating if the file or folder permission is read only or could not be found.

⎕NFE exception messages are now presented as APL-style error messages

⎕NFE 'Encoding' action documentation now indicates that it has a result

⎕NFE 'Create' action exception message improved with FileAccess or FileShare as 'rw'.

CSE: APL64 C# Script Engine

New ⎕CSE example: Create a .Net dll and use it with the ⎕CSE

In this example, a .Net Standard library assembly is created and used in APL64 via ⎕CSE, the C# Script Engine. A .Net Standard library assembly is inherently cross-platform. The APL64 ⎕CSE can be used to load and use such a library with APL64. Refer to the **Help | APL Language | ⎕CSE | C# Script Manual** menu item for the details.

New ⎕CSE example: Obtaining Financial Stock Information

In this example a financial company has made available a Web API which can be accessed as a Web API using ⎕cse. Refer to the **Help | APL Language | ⎕CSE | C# Script Manual** menu item for the details.

Improved Exception Message for ⎕cse 'SetValue'

If the argument to the ⎕cse 'SetValue' action is an APL64 zero-shape variable, the exception message will indicate that the variable value has 'No representation in .Net'.

Recreating a ⎕cse instance will now restore the ⎕cse default values

⎕cse 'properties' action returns property list in lower case

⎕cse visible and version properties are deprecated in .Net

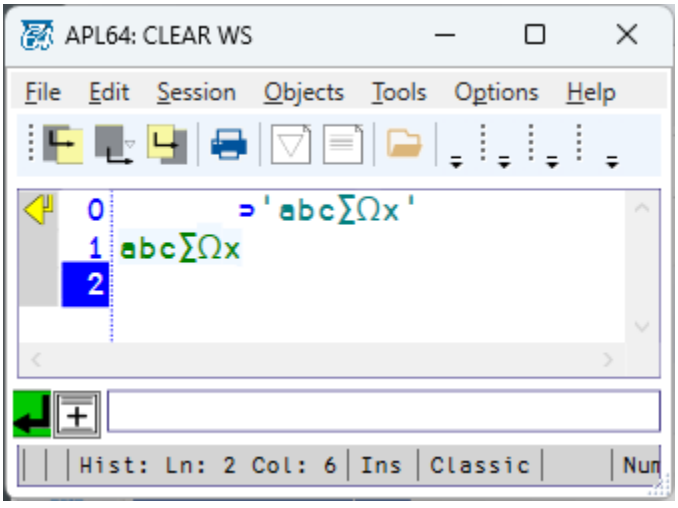
⎕deb and ⎕dltb exception messages improved for consistency

Branch primitive (→) now supports execution across the APL64 / APLNow32 interface

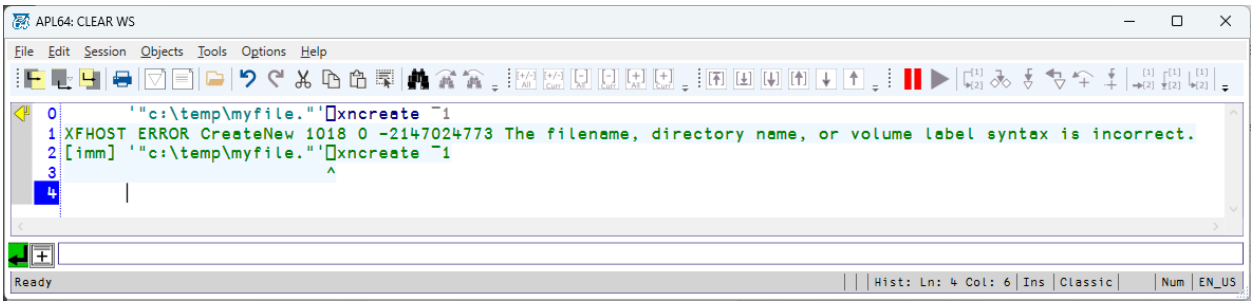
This modification supports ⎕wi exception handling across the 64/32-bit interface.

Disclose of an array containing Unicode characters is now supported

Instead of a NONCE ERROR, the expression can now be evaluated.



Improved Exception Message when Creating a File with an Invalid Name



Options | Configure Host Error Mappings Dialog

A user-defined Host Error Map may now be created, which can override a system default host error map. Items in the System Default Host Error Maps area are not user editable.

New ⎕WSE actions: AsyncCall, AsyncWait, AsyncFoFi, CallNow, and ⎕Progress

⎕WSE has been enhanced to support asynchronous, parallel operations using a pool of ⎕WSE server instances. The AsyncCall and AsyncWait actions are used to set up and control multiple ⎕WSE server instances to run APL functions asynchronously. The ⎕WSE CallNow action and the ⎕Progress system function are used to communicate between an APL client instance and ⎕WSE server instances. The AsyncFoFi action is used when the 'Fan Out / Fan In' programming structure is used in APL64 for asynchronous programming. Learn more about these APL64 enhancements from the **Help | APL Language | Using ⎕WSE** menu item.

⎕STRING with null left argument and 'Contains', 'Join' and 'Sortlindex' actions now return VALENCE ERROR

⎕EVAL "3 5⍴'"" returned a 3x5 array of Null's instead of failing with a LENGTH ERROR

⎕SIZE on a function included the size of the global variable when the object with the same name was localized in the function header

Developer Version GUI Modifications

Note: APL64 XML Configuration file

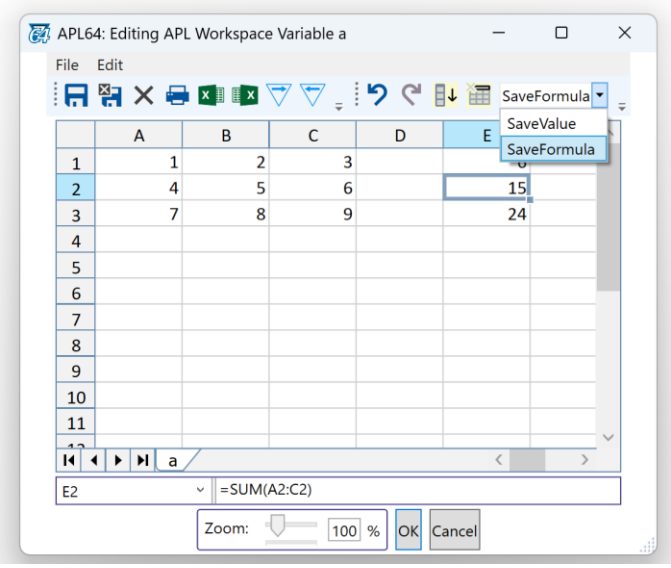
Your current APL64 XML configuration file will load properly in this update version. However, once the configuration file is saved in this version, the same APL64 XML configuration file will not load in prior versions.

EDSS Worksheet Editor

Cell formulas may be saved as the formula text or the evaluated value of the formula

Supported with the new SaveValue/SaveFormula combo box on the EDSSI dialog window.

When saving an EDSS worksheet to an APL variable, an option (SaveValue/SaveFormula) is provided to save the cell values which contain a formula as the evaluated value of the cell formula or the text representation of the cell formula.



EDSS Supports Hyperlinks in Worksheets

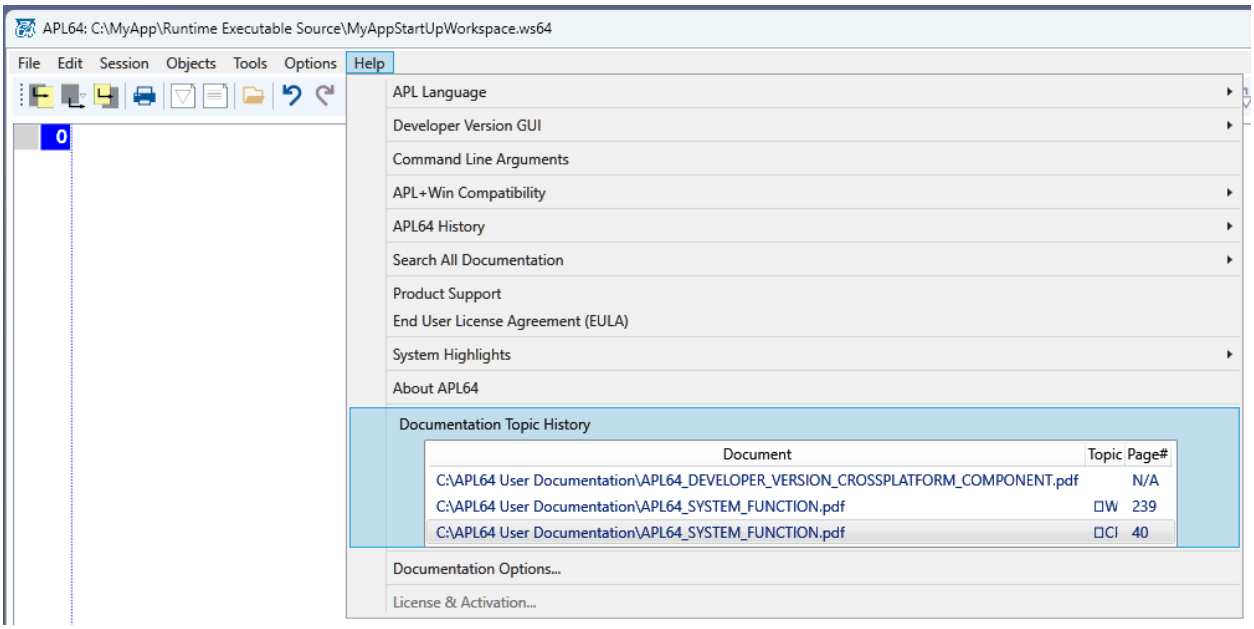
Hyperlinks in a worksheet imported to an EDSS editor are preserved. Hyperlinks may be created, edited and deleted in a worksheet using the EDSS editor. EDSS will export worksheets containing hyperlinks. APL variables support the hyperlink text but not hyperlink metadata.

Ctrl+MouseWheel will now update the zoom slider in an EDSS editor

Right, Left, Up, Dn, PgUp, PgDn keys will now update the zoom slider in an EDSS editor

Help | Documentation Improvements

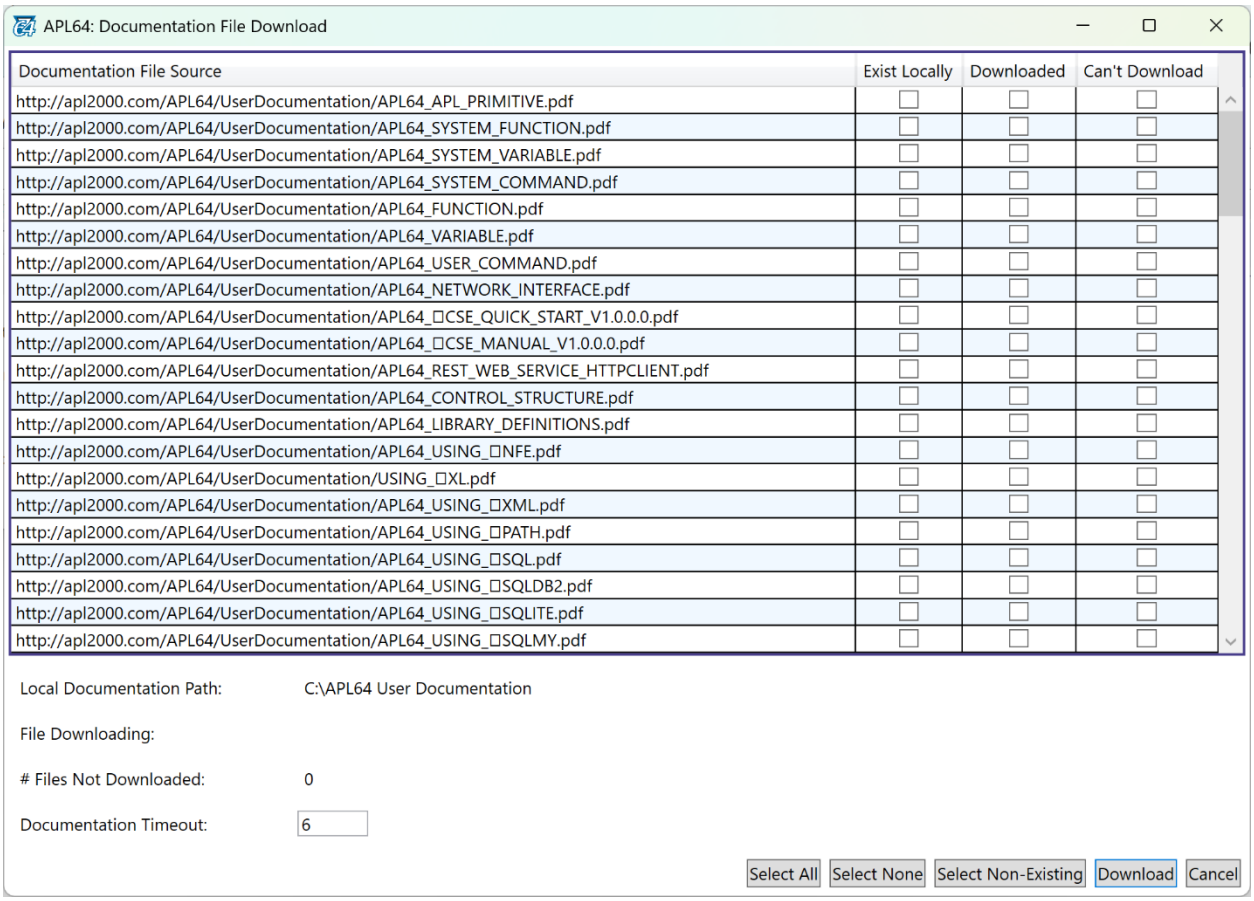
Help | Documentation Topic History improved format



Blank entries in the Documentation Topic History are no longer present.

Improvements for User Documentation Files Downloaded to the Workstation

New Download dialog provides information while downloading user documentation



Improved viewing of APL64 documentation stored locally on the workstation

By containing a local HTTP server within the APL64 Developer version instance, the default browser can now be used with local APL64 documentation files. The F1 key for context-sensitive documentation now displays the proper page for local documentation files, just like using the on-line documentation files.

The default browser is recommended for accessing local documentation files. Adobe Acrobat (Reader) is not recommended as the default viewer for local documentation files, because it does not properly open a local documentation file to the correct page unless Adobe Acrobat is closed after each use of the F1 key.

The new ‘Select Non-Existing' button in this dialog selects those documentation files that do not exist locally.

Access to System Highlights is now asynchronous to reduce the APL64 Developer start-up time

The start-up process for the APL64 developer version will no longer wait for the workstation to complete the check for a System Highlights update on the APL2K server.

Keyboard Short Cuts

The focus will remain in the floating editor window after executing a keyboard shortcut

This applies to the keyboard shortcuts: Ctrl+Shift+S (Save Workspace), Ctrl+Shift+C (Copy Workspace), Ctrl+Shift+L (Load Workspace) and Ctrl+Shift+R (Clear Current Workspace).

Shift+Tab didn't move tagged block of text to the left margin and outline glyphs disappeared

This could occur when pressing the Shift+Tab keyboard shortcut when the number of blank spaces remaining from the left margin didn't match the current value in the **Objects | Number Of Spaces In Tab** menu item.

Function / Variable Editors

Objects | Editors Pane Format | Minimize Editors Pane when all Editors are Floated

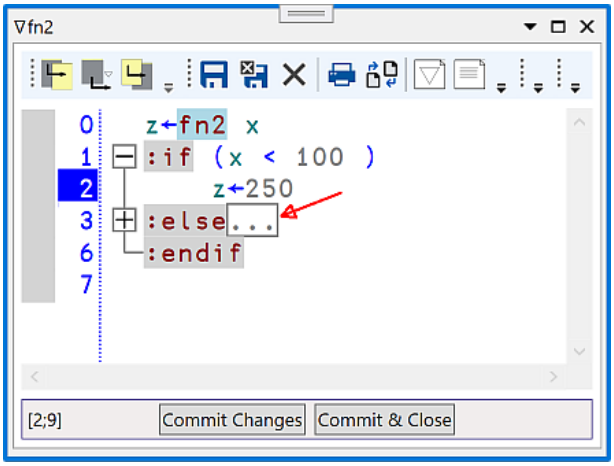
If this option is checked, the editors' pane area will be minimized (but not collapsed) when the only editors are floating editors. An editor pane does not entirely collapse so that the user can drag and manually dock a floating editor pane to the session.

Objects | Editors Pane Format |Maximize Editor or History Pane

This option enables maximizing the editor/History pane when it is selected in Active Panes (Ctrl+Tab). If checked, the editors (or history) pane will be maximized when a docked editor (or history) pane is selected by Ctrl+Tab. The editors pane will also be maximized when a new docked editor is opened.

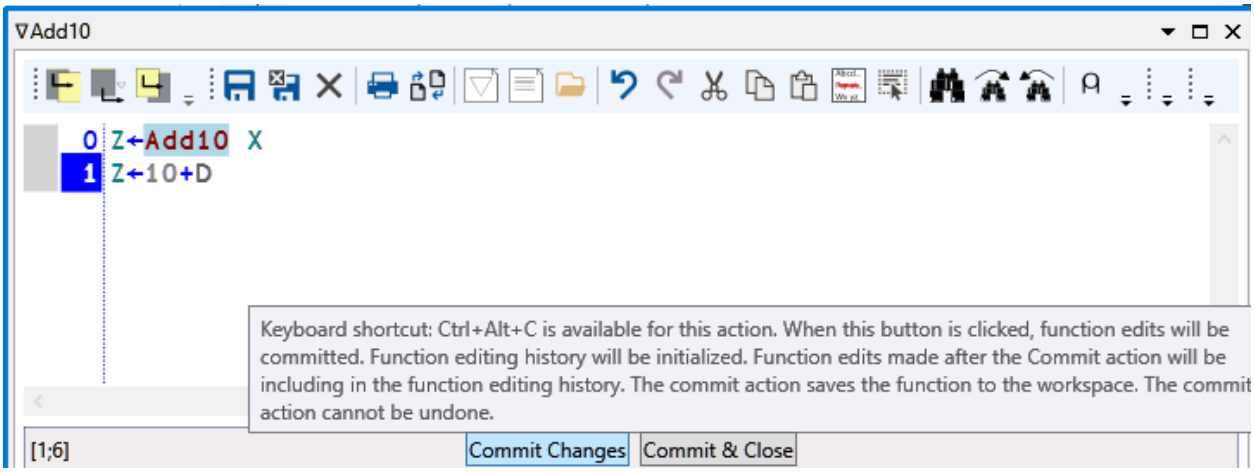
The function editor expands a collapsed region when code is entered on the collapsed line

The function editor now displays ellipsis to indicate a collapsed region:



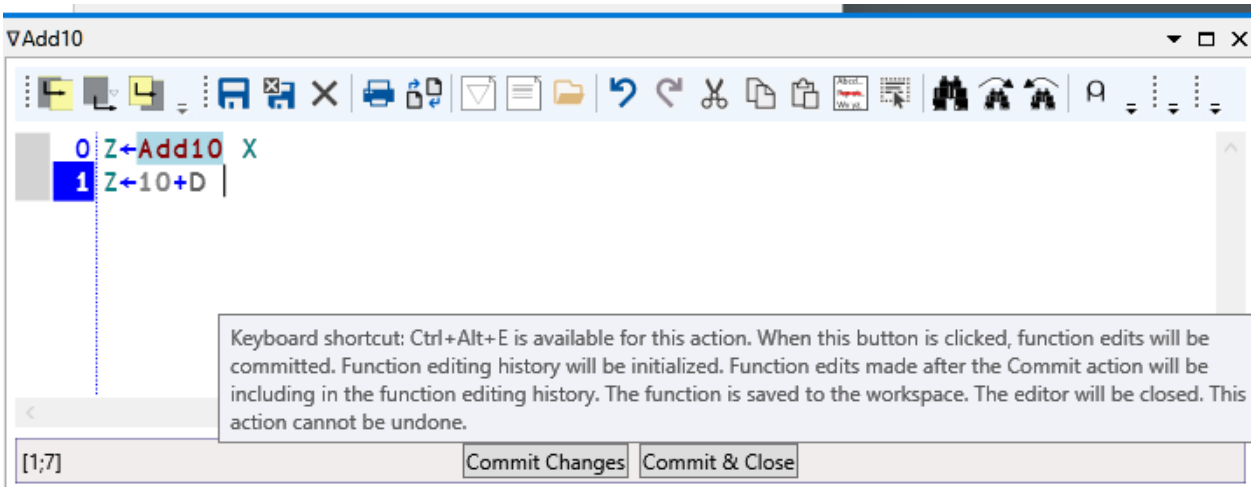
The toolbar height is now consistent when multiple editors are opened

Ctrl+Alt+C keyboard shortcut for Commit Changes button in the function editor

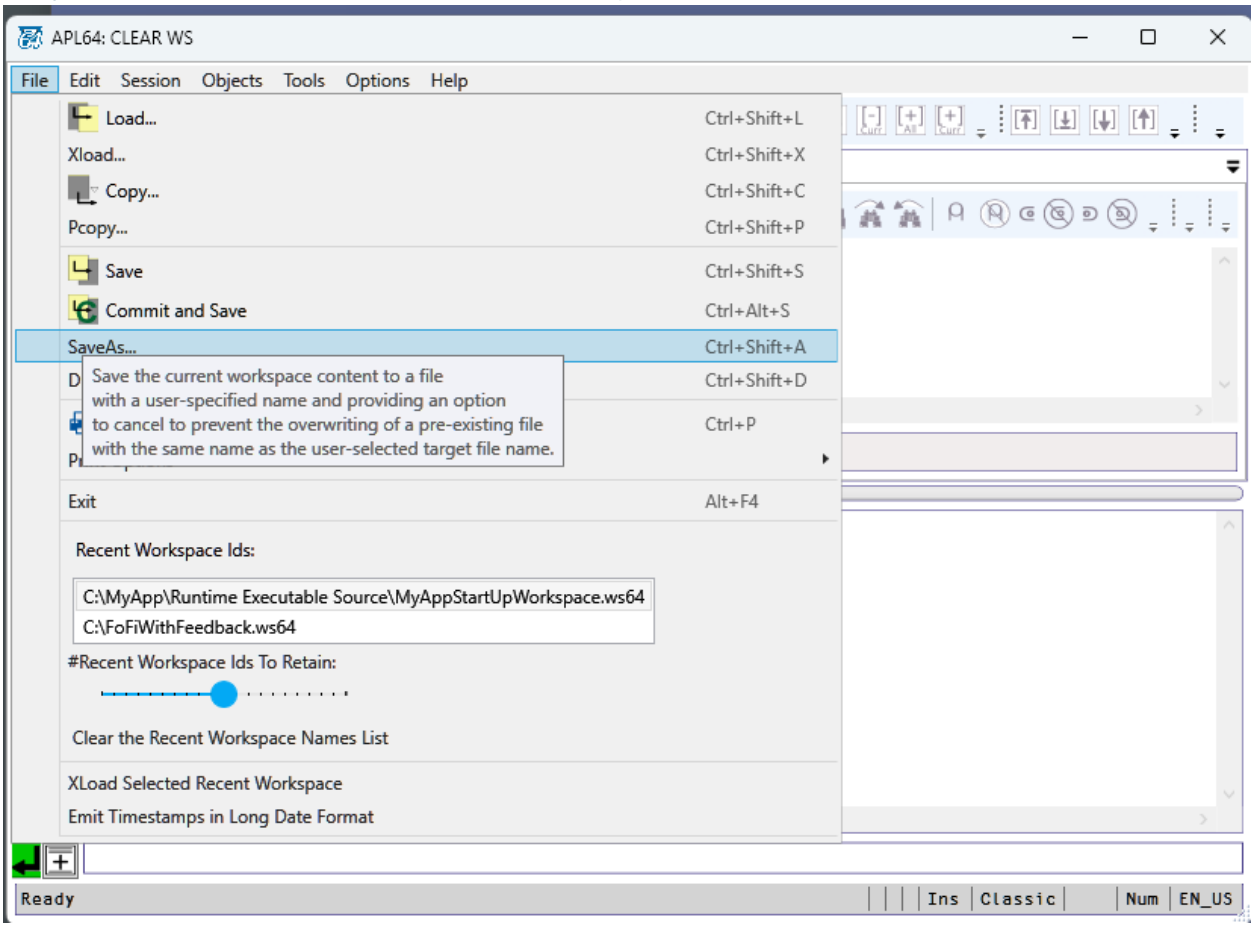


Function Editing History

Ctrl+Alt+E keyboard shortcut for the Commit & Close button in function editor



File | Commit and Save menu item and Ctrl+Alt+S keyboard shortcut



New ☐FnEdHist KBytes action

The result returned is the size, in kilobytes, of the function editing history for the function.

New)CSAVE system command

This system command commits existing function editing history records for functions in the current workspace and saves the workspace.

□def and □fx now initialize function editing history

When re-defining an APL64 function using these system functions, the pre-existing function editing history is initialized to minimize workspace memory used.

Ctrl+F12 keyboard short cut action corrected

When debugging, the Ctrl+F12 keyboard short cut will now open the suspended function in the editor on the suspended line

Editing Nested Variable: Edit Into Cell in 1st Column

When using the APL64 developer version variable editor on a nested variable, editing into a cell in the first column, will now point to the correct cell row.

File | Save (the current workspace content to a file)

When clicked, the workspace will be saved using the)save system command instead of)saveover.

Find/Replace Dialog

Replace dialog (Ctrl+H) Tab Key Order for TextToFind and ReplacementText Fields

When focus is in the 'Text to Find' field, the Tab key moves the focus to the 'Replacement Text' field.

The Find/Replace dialogs will not scale when the containing Pane/Editor is scaled

This modification will prevent the Find/Replace dialogs from obscuring parts of the containing editor/pane's content.

A match in Find/Replace is represented by the Search Region color element in the Colors dialog

When a match is found, the matched text is selected and the foreground and background colors used will be that of the Search Region color element in the Colors dialog.

The Click to find next key replaces the current match instead of the next match

Focus remains in the Find/Replace dialog using the Enter/Return key

When the Enter/Return key is pressed in the Find/Replace dialog, the next match, if any, will be selected with the focus remaining in the Find/Replace dialog.

Find/Replace with Existing Selection: Scope is now set to 'Selection'

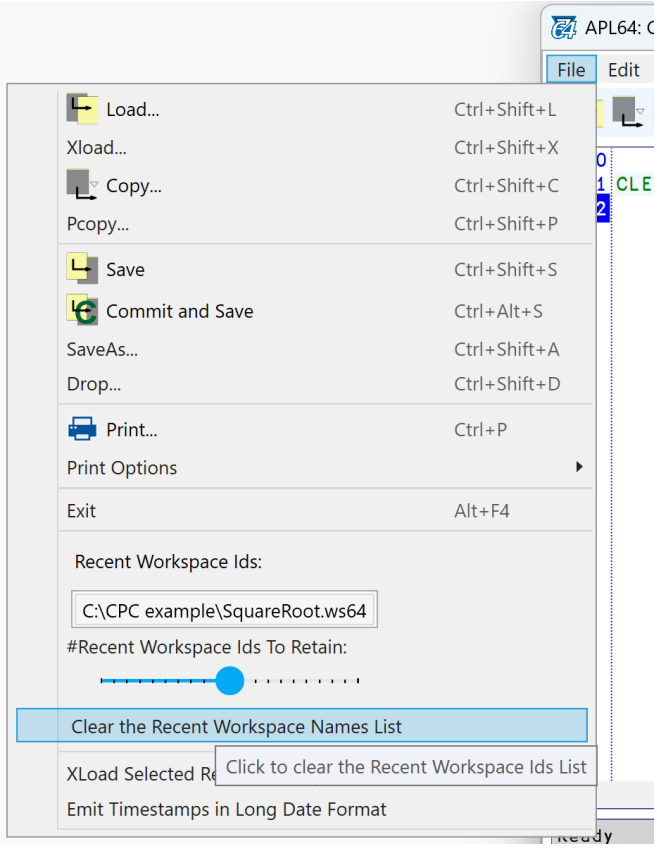
When the Find/Replace tool is presented and there is a current selection in the text to be searched/replaced, the Find/Replace Scope option will be set to 'Selection'.

Replace/Replace All action in the Replace dialog

When a Replace or Replace All action is performed and the Find/Replace dialog is closed, the focus will move to text in the editor after the last replacement was made.

File | Clear the Recent Workspace Names List menu item

When clicked, the workspace names in **File | Recent Workspace Ids** list will be cleared.



The output of a user-defined Tool is now directed to the pane with the keyboard focus

In prior versions, the output always appeared in the History pane.

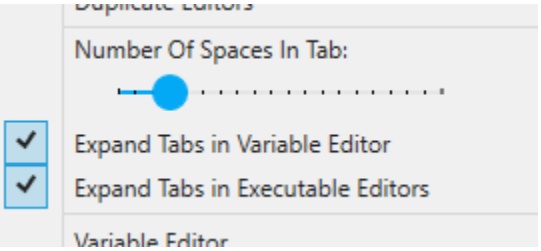
Text selections are clearly illustrated when no syntax coloring is chosen

When the Options | Colors dialog is used to select no syntax coloring, text selections are now clearly illustrated.

Options | Tabs menu section

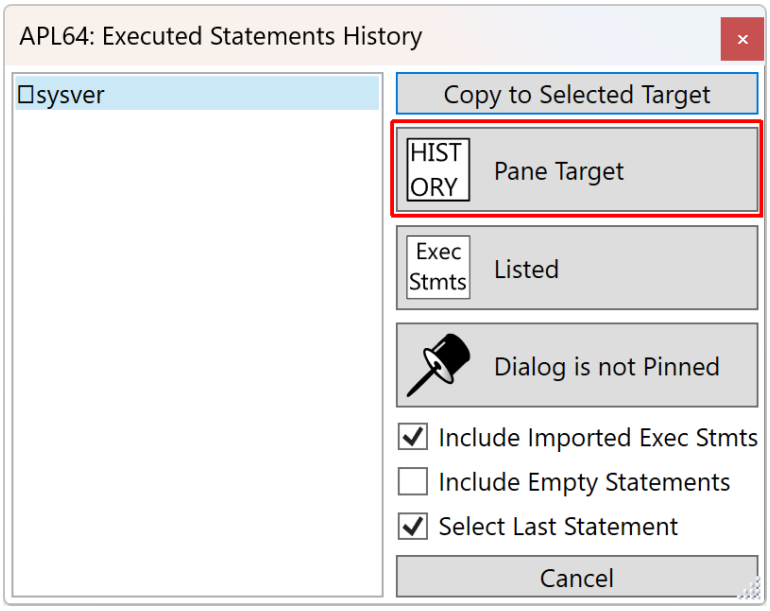
Improved Layout of Menu Options

These options control how tabs are inserted in the editors when the Tab key is clicked.



The Objects | Number of Spaces in minimum value is 1

Executed Statements History dialog: The default target set to "History" (pane) from "Clipboard"

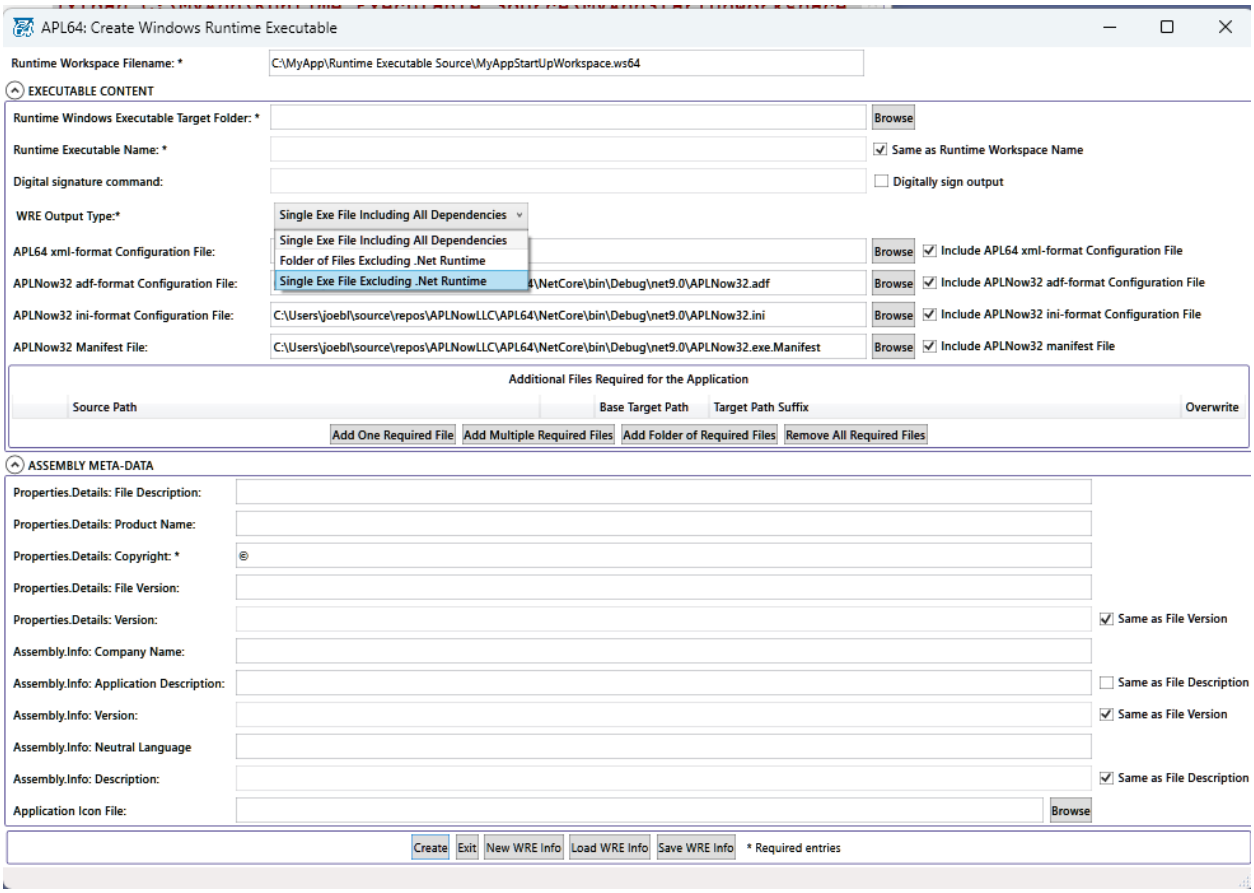


WRE and CPC: Deploying APL64 Apps

New WRE Output Type: Single Exe File Excluding .Net Runtime

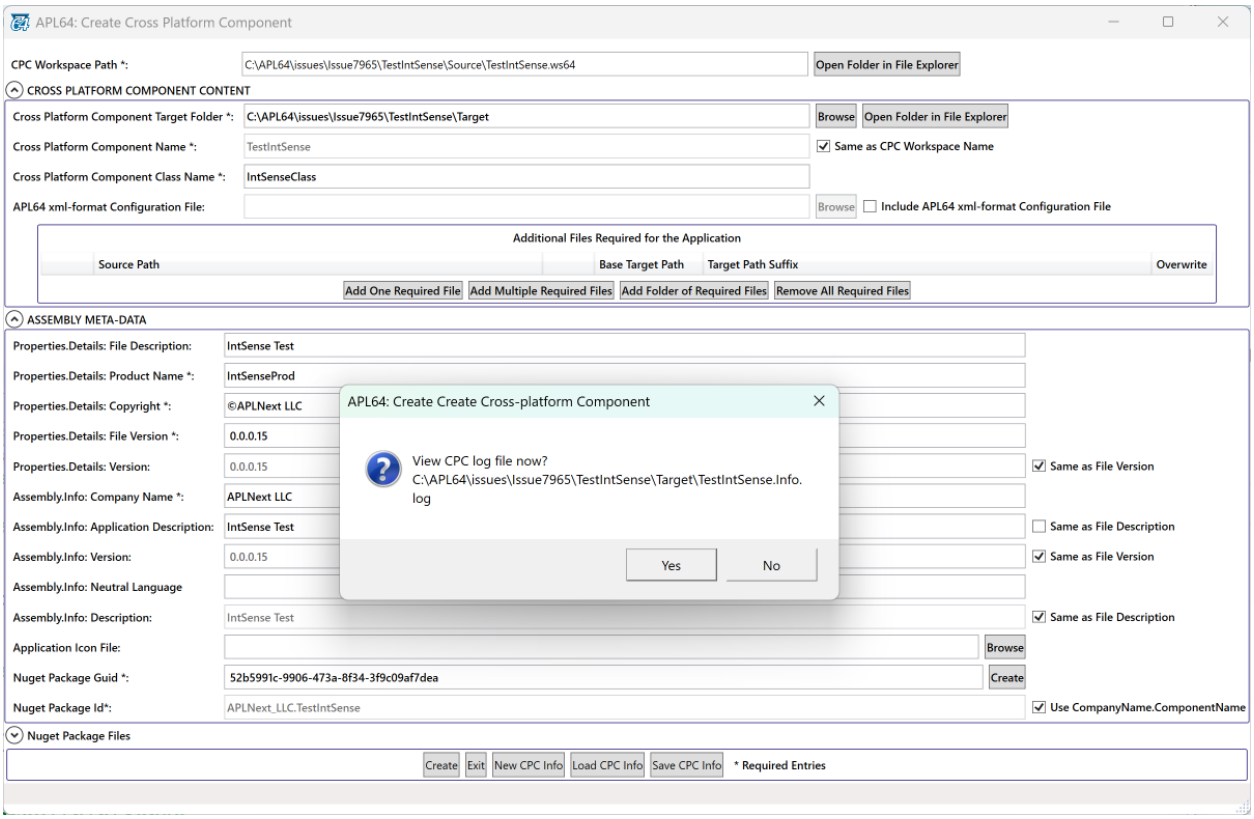
The output is a single, exe-format file containing the APL64 programmer-selected application specific

files and the APL64 runtime files, without the .Net Runtime files. When this WRE output is deployed to an end user workstation, that workstation must have the appropriate .Net runtime installed.



Option to view the log file created by the WRE and CPC Tools

After using the Create option in these tools, a prompt will be presented to indicate if the log file should be displayed.

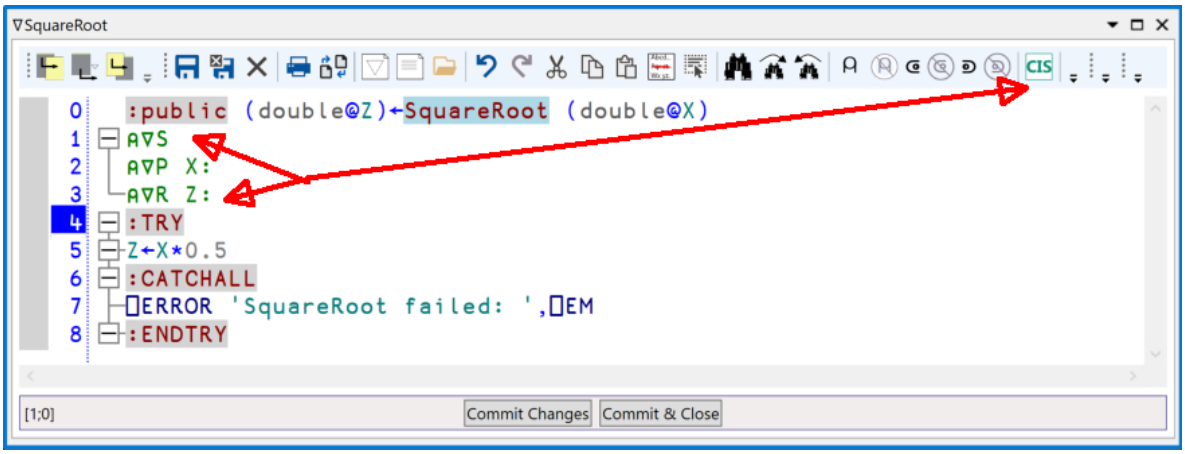


Load Workspace before Creating a WRE

The APL64 startup workspace must be loaded prior to starting the WRE creation tool. The 'Open Folder in File Explorer' button in the WRE creation tool has been removed, because the workspace has been loaded.

CPC Creation Tool can create Intellisense Documentation

The CPC creation tool will use public comments in public functions to create .Net Intellisense information for an APL64 cross-platform component. An 'Intellisense scaffold' of APL public comments can be created for a public function using the ⎕cpcis system function or the 'CIS' toolbar button in the function editor. The 'Intellisense scaffold' can be completed by the APL64 programmer based on the design of the public function.

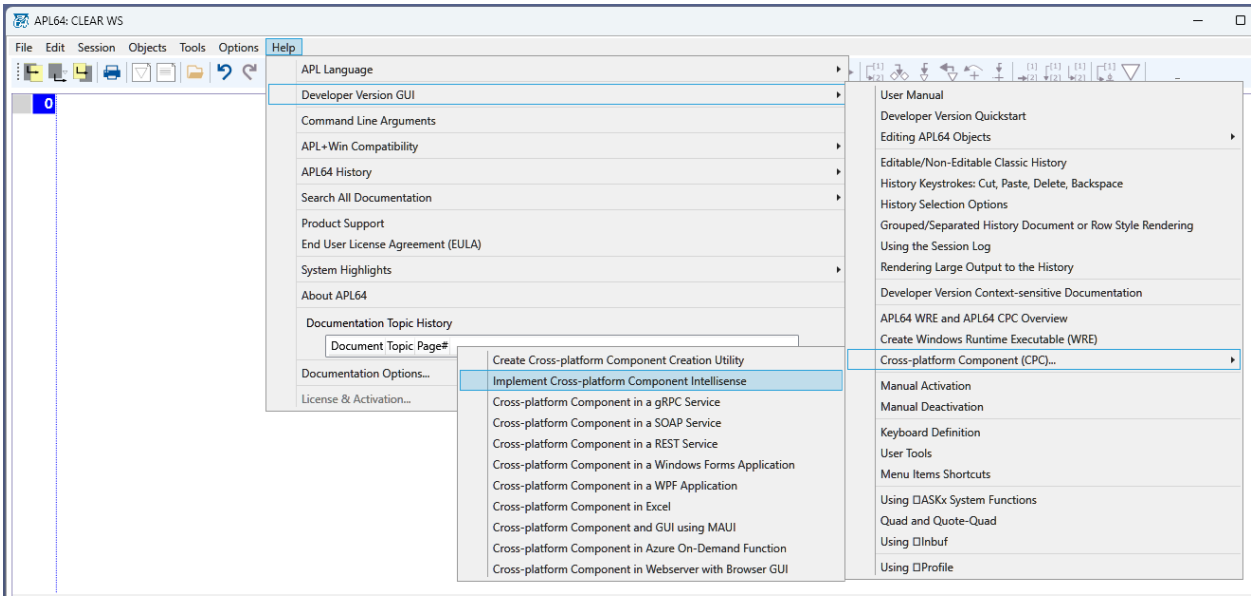


New □CSR and □CPCIS system functions

The □csr system function returns the name(s) or header(s) of the public functions in the current workspace.

The □cpcis system functions creates an IntelliSense scaffold to be used in public functions included in an APL64 cross-platform component (CPC).

For detailed documentation on these system functions, use the **Help | Developer Version GUI | Cross-Platform Component | Implement Cross-platform Component IntelliSense** menu item in the APL64 developer version.



New □WRE PropValue Action

The PropValue action will get and optionally set the properties in the WRE xml-format configuration file in the current APL64 instance. These properties correspond to the **Options | Create Runtime .Net Assembly | Create Windows Runtime Executable** dialog. The property names are not case sensitive. The □WRE New and PropValue actions may be used to create and modify the WRE xml-format configuration file in under program control.

New □CPC PropValue Action

The □CPC PropValue action will get, and optionally set, the properties in the CPC xml-format configuration file in the current APL64 instance. These properties correspond to the **Options | Create Runtime .Net Assembly | Create Cross-platform Component** dialog. The property names are not case sensitive. The □CPC New and PropValue actions may be used to create and modify the CPC xml-format configuration file under program control.

CPC Example Documentation Updated: CPC WebServer with an HTML Browser-based GUI

Refer to **Help | Developer Version GUI | Cross-platform Component (CPC)... | Cross-platform Component in Webserver with Browser GUI**

CPC Public functions: .Net data type specifications in the function header are case insensitive

APL64 Public functions: The .Net data type specifications are now case-insensitive. The APL64 function definition is not modified. The 'full' □csr fnName representation of the C# method associated with the APL64 function will use the appropriate case for the .Net data type.

This modification applies to □DEF, □FX and the APL64 function editor.