



APL+Win Version 16.0
Copyright (c) 2016 APLNow LLC.
All Rights Reserved
Apr 13, 2016

The APL+Win v16.0 release is available and recommended for all current APL+Win subscribers. To obtain this release, visit <http://www.apl2000.com/software.php> and click [APL+Win v16.0 Installer](#).

APL+Win 16.0 Updates:

New "Control Structure Keywords" Menu Options

The new "Control Structure Keywords" menu item in the Edit and context sensitive menus contain the *Jump to Top Keyword in Current Control Structure Block*, *Jump to Bottom Keyword in Current Control Structure Block*, *Jump to Next Keyword in Current Control Structure Block*, *Jump to Previous Keyword in Current Control Structure Block*, *Jump to Start of Next Control Structure Block*, and *Jump to Start of Previous Control Structure Block* options. Their purpose is to be able to efficiently navigate between control structure keywords in the function editor. The menu items and their corresponding keyboard shortcut keys are listed below:

Operation -----	Keybd Shortcut*
Jump to Top Keyword in Current Control Structure Block	Ctrl+M, Ctrl+T
Jump to Bottom Keyword in Current Control Structure Block	Ctrl+M, Ctrl+B
Jump to Next Keyword in Current Control Structure Block	Ctrl+M, Ctrl+N
Jump to Previous Keyword in Current Control Structure Block	Ctrl+M, Ctrl+P
Jump to Start of Next Control Structure Block	Ctrl+M, Ctrl+D
Jump to Start of Previous Control Structure Block	Ctrl+M, Ctrl+U

*This keyboard shortcut is entered by typing the prefix key (Ctrl+M), releasing the M key and then typing and releasing the second lettered key.

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150

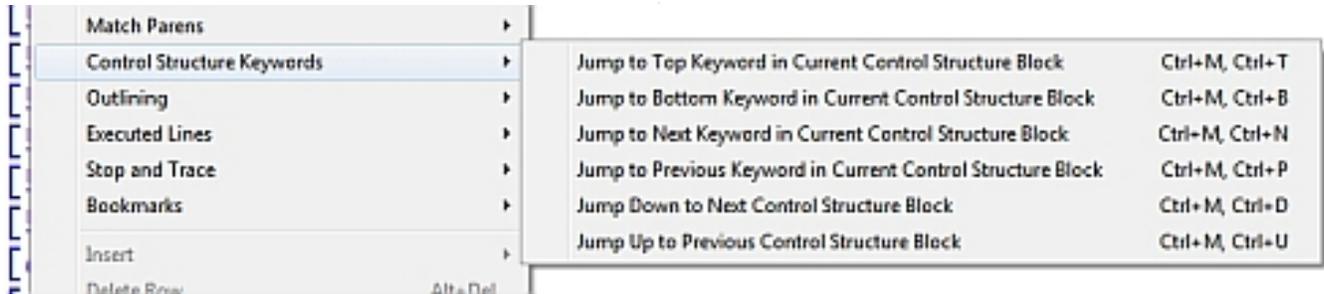


Figure 1: New keyboard shortcut keys for navigating control structure keywords

New "Outlining" Menu Option

The new "Outlining" menu item in the Edit and context sensitive menus contain the *Toggle All Regions*, *Toggle Current Region*, *Collapse All Regions*, *Collapse Current Region*, *Expand All Regions*, and *Expand Current* options. Their purpose is to expand and collapse regions (comment and control structure elements) in the function editor. The menu items and their corresponding keyboard shortcut keys are listed below:

Operation	Keyboard Shortcut*
Toggle All Regions	Ctrl+M, Ctrl+L
Toggle Current Region	Ctrl+M, Ctrl+M
Collapse All Regions	Ctrl+M, Ctrl+A
Collapse Current Region	Ctrl+M, Ctrl+C
Expand All Regions	Ctrl+M, Ctrl+X
Expand Current Region	Ctrl+M, Ctrl+E

*This keyboard shortcut is entered by typing the prefix key (Ctrl+M), releasing the M key and then typing and releasing the second lettered key.

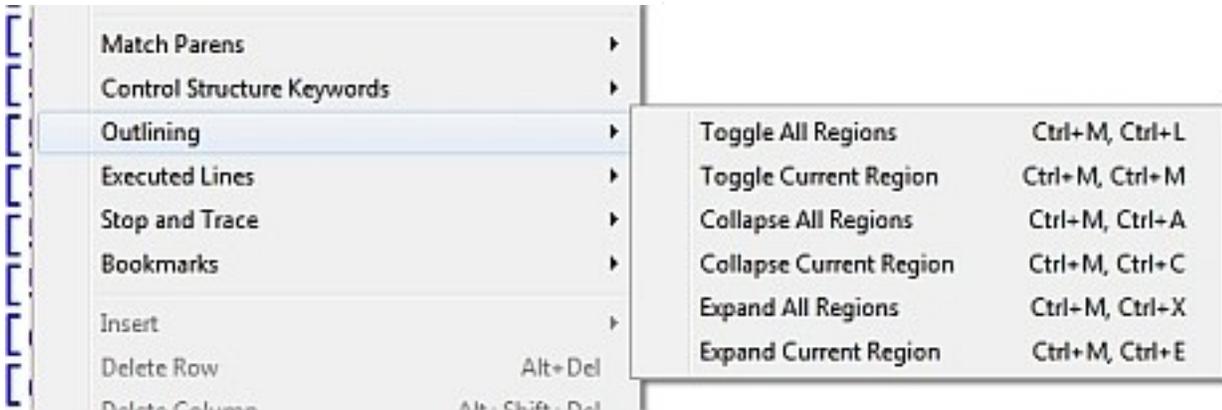


Figure 2: New keyboard shortcut keys to collapse and expand regions

Update to Unique primitive function (U)

The Unique primitive function (U) has been enhanced to support a scalar argument instead of reporting a LENGTH ERROR error message.

Update to “Match Pares” and “Match and Tag Pares” Menu Options

The keyboard shortcut keys for “Match Pares” and “Match and Tag Pares” in the Match Pares menu item in the Edit menu has changed from Ctrl+M and Ctrl+Shift+M, respectively, to the following:

Operation	Keyboard Shortcut	Old Keyboard Shortcut*
Match Pares	Ctrl+M, Ctrl+J	Ctrl+M
Match and Tag Pares	Ctrl+Shift+M, Ctrl+Shift+J	Ctrl+Shift+M

*This keyboard shortcut is entered by typing the prefix key (Ctrl+M), releasing the M key and then typing and releasing the second lettered key.

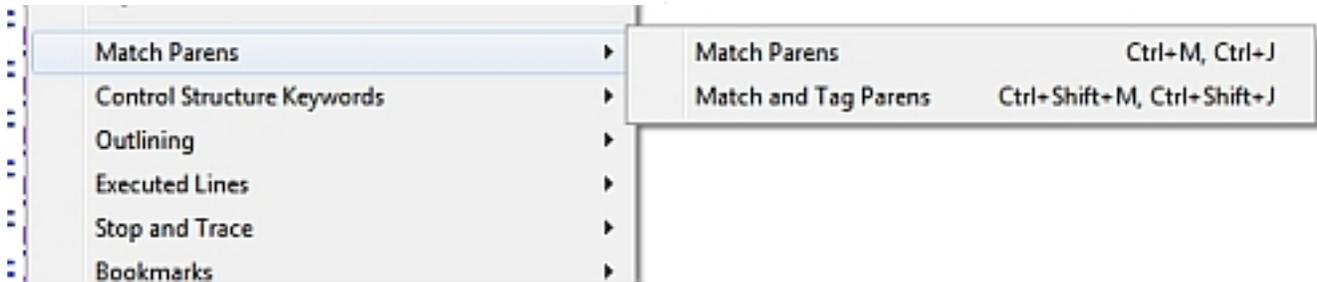


Figure 3: Updated keyboard shortcut keys for Match Pares and Match and Tag Pares

Update to the Gather Menu Option

The accelerator key on the 'G' in the Gather menu item in the Edit menu has moved to the 'g' in the new Outlining menu item.

APL+Win 16.0 Bug Fixes:

1. In version 15.1, the result value from `⊠CN` may be corrupted when the result is an empty nested array, which can lead to an APL+Win crash. Example,

```
result←⊠cn 0 0 ρ ''
↑result    a this will cause APL+Win to crash
```

2. In versions 7.2 and newer, the `⊠VGET` system function created orphans (bad referenced object pointers) in the workspace when the named argument on the right of `⊠VGET` was the same as the default return value or the pre-existing value in the workspace happened to be the same as the named argument.
3. In version 15.1, the `⊠DEF` system function failed for a locked function (⌘).
4. In versions 10.0-15.1, the tooltipstyle property values did not work for the CommandButton (on CommandBar) and Page (on Selector) classes.
5. Searching for a keyword in the online help file that was started from within the development system could cause the development system to hang.



6. In version 15.1 and earlier, APL+Win may hang when executing the `□COPY` system function from a User Command function.
7. In version 15.1 and earlier, the euro (€), commute (~) and mismatch (≠) characters did not appear from the keyboard shortcut keys in the Edit control with style property 2048.

APLNext C# Script Engine v2.0.34.0

New CSE architecture

A CSE instance runs in the application domain of a SignalR server that is separate from the APL+Win application domain. The local SignalR server runs on the target workstation using a port# specific to the CSE instance. The APL+Win `□cse` system function uses a lightweight, ActiveX client to access the SignalR server.

With the new CSE Architecture the memory used by .Net objects created in a CSE instance is released when the SignalR server containing that CSE instance is closed, even if the APL+Win session which initializes and closes the CSE instance is not closed. This was not possible in the previous versions of the CSE which ran in the same application domain as APL+Win.

CSE as a 64-bit Windows process

When the CSE runs as a 64-bit process, it is able to access memory over 2GB and create multiple .Net objects each over 2GB, up to the limit of the memory on the target workstation [192GB on Windows 7 Professional]. This was not possible in the previous versions of the CSE which ran in the same 32-bit application domain as APL+Win.

Required Windows Operating System

This version of the CSE requires 64-bit hardware and Windows operating system.

Updated .Net Framework version

This version of the CSE requires the Microsoft .Net Framework v4.6.1 (full).

Enhanced string substitution for CSE ExecStmt and GetValue methods

The CSE string substitution feature of the CSE 'ExecStmt' and 'GetValue' methods has been enhanced to support substitution of APL+Win data into C# text strings.



Enhanced CSE logging

The CSE log functions have been enhanced with logs for the SignalR server, ActiveX SignalR client and the C# script engine.

Enhanced CSE documentation & examples

Additional CSE examples have been developed and improvements in the CSE documentation have been implemented. They're available here: [CSE documentation and examples](#).

New CSE installer Option

The CSE installer has an optional 'APPDIR' parameter to indicate the desired installation folder. The default installation folder is '...\APLWIN16\APLNext.CSScriptEngine'.

Enhanced CSE 'Close' method

In this version of the CSE the `□cse` 'Close' method operates the same way as in prior versions of the CSE. This method closes the SignalR server, the C# script engine instance and the ActiveX SignalR client so that the memory used by the SignalR server and the C# Script engine instance are released even if the APL+Win instance is not closed.

Optionally the programmer may provide a Boolean argument to the `□cse` 'Close' method to retain the SignalR server containing the C# script engine instance.

If the optional Boolean argument to the `□cse` 'Close' method is 0, the action of this method is the same as if no Boolean argument was provided.

If the optional Boolean argument to the `□cse` 'Close' method is 1, the SignalR server instance will be retained and may be re-used later in the APL+Win session. This will reduce start-up time when creating a subsequent C# script engine instance of the same name and using the same port# for message communication between APL+Win and the C# script engine.

Enhanced CSE 'Init' and 'InitSTA' methods

An optional port# argument is available for these methods. When no SignalR port# is programmer-specified, port# 10999 will be used. The port# is used by the `□cse` system function to communicate with the associated SignalR server. A unique port# must be associated with each SignalR server instance, so that re-use of a SignalR server instance is possible in the same APL+Win session.

Enhanced [CSE] section in APL+Win .ini file

Additional programmer-defined values will be considered by the CSE, if they are present in the [CSE] section of the APL+Win .ini file:



- Server Window Visible
- Reconnection Timeout
- Connection Timeout
- Connection Interval
- Log File Path

Enhanced CSE object model

- New CSE object properties: logfilepath, tracelevel, visible
- New CSE instance methods: GetLogFileName, GetPortNumber
- New CSE system property: version
- Deprecated CSE instance property: version
- Deprecated CSE instance method: GetVersionNumber

APL2000 Staff
www.apl2000.com

Corporate Office
PO Box 361
Brielle NJ 08730
732-223-5575

Sales Office
One Research Ct., Ste. 325
Rockville MD 20850
301-208-7150