APLNext C# Script Engine 3.0.24.0 for APL+Win v17.0

Update Documentation

The APLNext C# Script Engine (CSE) is accessible from APL+Win via the ⎕cse system function. The CSE enables an APL+Win application system programmer to use the C# programming language to access the Microsoft .Net Framework.

- **New CSE installer: 64- and 32-bit Windows Operating System**
    - The CSE installer has been updated: "APLNext CSE Components Setup v3.0.24.0.exe". Before running the CSE installer check the file's properties to be sure it is not 'blocked'. The CSE installer registers ActiveX components of the CSE on the target workstation and puts .Net components in the Global Assembly Cache, therefore elevated privileges are required to properly install the CSE.
    - The CSE installer will now install the CSE to a 32-bit or 64-bit version of the Windows operating system.
    - Features of the CSE which rely on accessing large memory spaces are not available when the CSE is installed in a 32-bit Windows operating system environment. Refer to CSE example #192.

- **New CSE instance property: 'texttransfer'**

  This property controls the way that APL+Win sends text array information to the CSE. The default value of the property is 0 which means that the .Net 'String' type is the appropriate .Net data type of the .Net container which will receive text sent by APL+Win to the CSE. This default reflects the operation of all prior versions of the CSE. A 'texttransfer' property value of 1 indicates that the .Net 'Byte' type is the appropriate .Net data type of the .Net container which will receive text sent by APL+Win.

  Note that all CSE examples reflect a 'texttransfer' property value of 0, unless otherwise stated in the CSE documentation.

- **Enhanced Error Message**

  The C# exception error message returned by the GetLastError method has been enhanced to include the exception message, exception stack trace, inner exception message, and inner exception stack trace.

- **CSE Dual Event Channel Architecture**

  This version of the CSE implements a dual event channel so that custom event and routed events can be independently handled. This is necessary because APL+Win is a single-threaded application and the current version of the Microsoft SignalR server technology, used to contain the .Net portion of the CSE, now operates fully asynchronously.

- **Enhanced CSE documentation & examples**

  Improvements in the CSE documentation have been implemented. Some examples have been modified from prior versions to properly operate in this version of the CSE.
    - Example #035: Difference between ‘AddEventHandler’ and ‘AddEventHandlerEx’
    - Example #061: AddEventHandler method in a CSE script
    - Example #069: CSE script containing several instances of ‘APL:’ execution control transfer
    - Example #103: SetValue/GetValue: Text scalar behavior depends on target .Net container
    - Example #123: Illustrates the available CSE instance properties
    - Example #150: Setting/Getting System.Windows.Forms control properties via events
    - Example #151: System.Windows.Forms .Net assembly with code behind
    - Example #152: System.Windows.Forms .Net assembly with custom public properties
    - Example #153: Reserved: WPF .Net assembly with custom public properties
    - Example #154: Reserved: WPF .Net assembly with FieldModifier property set to public
    - Example #161: ADO.Net
    - Example #163: Creating a custom .Net event
    - Example #173: Set/Get Char[] array values
    - Example #184: CSE AddEventHandlerEx method
    - Example #185: CSE AddCustomEventHandler method
    - Example #189: Bad variable type example
    - Example #197: CSE ‘Close’ 1 method
    - Example #208: SetValue for public static string property
    - Example #209: CSE ‘transpose’ property and APL+Win text data
    - Example #210: CSE ‘transpose’ property and APL+Win text data of rank three
    - Example #211: CSE ‘texttransfer’ property
    - Example #212: CSE ‘texttransfer’ property
    - Example #213: CSE ‘texttransfer’ property
    - Example #214: Use .Net WebClient to download file via http or https

  Pre-compiled .Net assemblies loaded by the CSE must be compiled for the applicable Microsoft Windows operating system version [32- or 64- bit] or they may be compiled for ‘Any CPU’. Some of the CSE examples depend on .Net assemblies which were not created by APLNext, e.g. SQLite, so for these .Net assemblies the applicable authors or vendors would be the source of the appropriately-compiled version of these assemblies. The following .Net assemblies, created by APLNext are used in the CSE examples have been re-compiled for ‘Any CPU’, if necessary:
    - APLNext.CryptString
    - APLNext.CseExtensionMethods
    - APLNext.NetInfo1
    - APLNext.VBEx1
    - Example151
    - Example152
    - Example153

  - Example154
  - MyNamespace.MyAsm1

- **Windows registry checks minimized**
  - The installation path and version number will be obtained only once from the Windows registry the first time any CSE method or property is used. These values will be retained as values of static class members during the APL+Win session.

- **Microsoft .Net Framework version**
  This version of the CSE requires the Microsoft .Net Framework v4.6.1 (full). This component is available at no cost from Microsoft.

- **Required APL+Win version**
  This version of the CSE requires APL+Win v17.0.01.

- **Required Windows Operating System**
  - Windows 7 SP1, 10 or Windows Server 2008, 2012, 2016 or subsequent versions
  - For full availability of CSE features, especially those related to large .Net objects over 2GB, 64-bit hardware and 64-bit Windows operating system installed on the target workstation are required.

- **CSE version compatibility with APL+Win versions**
  The enhancements described here apply to version 3.0.24.0 of the CSE which is associated with APL+Win v17.0.01. CSE versions are linked to specific versions of APL+Win and the Microsoft .Net Framework. This version of the CSE is not compatible with APL+Win prior to version v17.0.01. CSE versions prior to v2.0.34.0 are not compatible with APL+Win if the Microsoft .Net Framework 4.6.1 has been installed on the target workstation.

  The CSE development team strives to maintain an invariant programming syntax among CSE versions, however new features of the CSE may require new programming syntax. Detailed documentation of these CSE enhancements including their implementation technology is included with the CSE documentation and examples.