APL+Win Version 17.0

May 10, 2017

The APL+Win v17.0 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 v17.0 Installer*.

Enhancements and Changes

- **New Support for DPI-awareness in APL+Win**
  Added support for DPI-awareness in the session manager window and the □WI objects (excluding any ActiveX controls and objects).

- **New dpi property**
  Added the read-only system object property dpi that specifies the DPI-Aware state of APL. The four-element nested vector has the following values:
  [1] returns 1 if APL+Win is running in DPI aware mode; 0 if DPI un-aware mode.
  [2] returns the system-wide DPI scaling percentage of the system (100 by default, 200 for 200% scaling, etc.).
  [3] returns the dots per inch of the system-wide DPI scaling (96 at 100% scaling, 192 at 200% scaling, etc.).
  [4] returns the pixel height of the default font.

- **Improved placement of Find and Replace Dialogs**
  Corrected placement of the Find and Replace dialogs when using multiple displays. This means the Find and Replace dialogs appears where they were last positioned instead of the primary display.

- **The scale property scalemode element 5 now returns virtual pixels instead of pixels**
  There is a new configuration setting [Config]VirtualPixels that controls whether scalemode=5 units are virtualized for DPI scaling or unscaled pixels; 1 (default) means virtual pixels are scaled to the current DPI level and 0 means not scaled.

- **New scalemode element added to scale property**
  The scale property has been extended to support scalemode = 6 to return true pixels.

- **New defaultFont property (system (#) object)**
  This enhancement improves the default font selection process for □wi objects when running in DPI aware mode. This dramatically improves the font size versus window layout behaviour when running at high DPI scaling levels.

- **New style property value 65536 added to Button object**
  The new style property value 65536 has been added to the Button and UButton objects to allow their caption to wrap when the width of the caption is wider than the width of these objects.

- **New opaque property available for some APLGUI controls**
  The opaque property defines the behaviour of the opaque feature in some APLGUI controls. The opaque property

can have a value of 0 or 1. The default value is 1. A value of 0 causes the background of the object to be transparent such that the parent window behind it is visible through the background area of the control. For example, a Check control that's a child of a Picture control will display the image assigned to the Picture control as the background of the Check control.

- **New usereplstr property for the ⎕NFE system object**
  This property controls when the APL+Win carriage return, ⎕TCNL, needs replacing with Windows carriage return in text transferred from APL+Win to a Windows file and vice versa. By default, the value of the usereplstr property is set to 1 to perform the character replacement.

- **⎕NFE's encoding property now supports hyphenated encoding names**
  This means UTF-8 will be treated the same as UTF8.

- **New [Config]LogCatch in APLW.INI configuration file**
  Added [Config]LogCatch to -1 to APLW.INI configuration file to improve ⎕WCALL and ⎕WI exception handling.

- **New LC.Charts v2.83 released**
  A new update to the LC.Charts application by Eric Lescasse that can display your APL data in a variety of charts.

Bug Fixes

- Addressed bug in versions 12.0 to 16.2 that resulted in wrong values being displayed in some cases due to rounding errors in the system. For example, notice how the first value for N is formatted in the result with trailing ...9990 rather than ...9998 in the example below:

```
  N←9.999999999999998
  ⎕PP←16
  (N) (1 2 3)
9.999999999999990  1 2 3
  2 2ρ(N) (1 2 3) 9.99 (3 2 1)
9.999999999999990  1 2 3
9.990000000000000  3 2 1
```

  Here is the corrected behaviour:

```
  N←9.999999999999998
  ⎕PP←16
  (N) (1 2 3)
9.999999999999998  1 2 3
  2 2ρ(N) (1 2 3) 9.99 (3 2 1)
9.999999999999998  1 2 3
9.990000000000000  3 2 1
```

  Here's another formatting issue:

```
'K17G<ZZZZZZZZZZZZZZZZZ9.ZZZZZZZZZZZZZZZZZ>' ⎕FMT 100987
      100986.9999999999
```

which is corrected by this fix:

```
'K17G<ZZZZZZZZZZZZZZZZZ9.ZZZZZZZZZZZZZZZZZ>' ⎕FMT 100987
      100987
```

- Fixed limit property for RichEdit and Edit controls
  Addressed problem that prevented pasting or entering more than 32767 and 30000 characters (their default limits) in the RichEdit and Edit controls, respectively, when the controls were first opened and the limit properties hadn't been set to 0 or a value exceeding the default limits.

- Addressed bug that caused APL, started without enabling DPI awareness, to shrink in size when starting an external C# application that did DPI awareness enabled.

- Fixed numeric editor row and column header font
  1. the font was too small at high DPI scaling levels
  2. the background of the font and the color of the header cell did not match.
  3. the column header width was sized too narrow to read all column headers for a very large number of columns containing small
  numbers such as the example below:

  ```
  v←1 1000000⍴0
  ```

- The Write and Read methods in the ⎕NFE system object did not support multi-lined strings.

- Addressed the problem with the display of the CommandBar control in a Frame control when the Form control was created hidden or closed. In this case, the CommandBar appeared with a black background color when APL+Win is run with a manifest file.

- Addressed the problem when Checkbox and Option controls are nested inside a style=7 (transparent) Frame control that is also nested inside a CommandBar control. In these cases, the background color of the Checkbox and Options controls were not painted correctly (gray rectangle behind them) when APL+Win is run with a manifest file.

- Addressed problems when multiple style=0 Frame controls are nested inside each other. In this case the caption did not display in the same font for all the Frame controls after the top-level Frame control when APL+Win is run with a manifest file.

- Addressed problem when a CommandBar control is a child of a style=0 Frame. In this case, various painting problems, like the CommandBar control appearing with a black background color, occurred when APL+Win is run with a manifest file.

- Addressed problem when executing ⎕CN and ⎕CM with a space as the left argument and the letter 'q' in the right argument. In these cases, APL+Win could crash.

- Addressed problem in the function editor where a control structure block in a single line did not expand while collapsed when a search word was found in the collapsed statement.

- Bug Fix: Addressed problem in dyadic thorn (⍕) with an empty left argument that sometimes reported a DOMAIN ERROR and sometimes reported a random result instead of correctly reporting a LENGTH ERROR.

  ```
  E.g.

        (0ρ0)⍕10 20
  ************************ 20
        (0ρ0)⍕10 20
  DOMAIN ERROR
        (0ρ0)⍕10 20
             ^
  ```

APLNext C# Script Engine v3.0.24.0 Updates

- **New CSE installer**
  - 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, therefore elevated privileges are required to properly install the CSE.
  - 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.
  - This version of the CSE requires APL+Win v17.0.01.
  - 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.

- **Enhanced CSE documentation & examples**

  Improvements in the CSE documentation have been implemented. Additional CSE examples have been developed. Some examples have been modified from prior versions to properly operate in this version of the

CSE. which may be modified in a future version of the CSE. The complete list of modified examples are listed in the *APL+Win C# Script Engine Update_v3.0.24.0.pdf*.

- **New CSE instance property: texttransfer**
  This property controls the way that APL+Win sends text array information to the CSE.

- **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.

- **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.

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.

APL2000
www.apl2000.com