What's New with HTMLRenderer

Brian Becker, Dyalog APL Tools Architect
HTMLRenderer

- Dyalog APL object
  - 'hr' ⎕WC 'HTMLRenderer' 'Hello!'
  - (hr←⎕NEW ,⊂'HTMLRenderer').HTML←'Hello!'
What's New with HTMLRenderer?

```
(hr←⎕NEW ,⊂'HTMLRenderer').HTML←'Hello!'
```

Hello!
What's New with HTMLRenderer?

- Dyalog APL object
  - 'hr' ⎕WC 'HTMLRenderer' 'Hello!'
  - (hr←⎕NEW ,⊂'HTMLRenderer').HTML←'Hello!'
- Interface to the Chromium Embedded Framework (CEF)
  - Chromium Web Browser – the foundation for Google Chrome
- Renders HTML5/CSS/JavaScript

- Adobe Acrobat
- Amazon Music
- Evernote
- Google Web Designer
- MATLAB
- Sling TV
- Steam
History

- Born in 2017 (Dyalog v16.0)
  - What is this?
  - Sometimes makes a mess (SYSERROR)
History

- Born in 2017 (Dyalog v16.0)
  - What is this?
  - Sometimes makes a mess (SYSERROR)
- Grows up a bit in 2018 (Dyalog v17.0)
  - Falls over less often
History

- Born in 2017 (Dyalog v16.0)
  - What is this?
  - Sometimes makes a mess (SYSERROR)
- Grows up a bit in 2018 (Dyalog v17.0)
  - Falls over less often
  - Interesting to play with
History

- Born in 2017 (Dyalog v16.0)
  - What is this?
  - Sometimes makes a mess (SYSErrOR)
- Grows up a bit in 2018 (Dyalog v17.0)
  - Falls over less often
  - Interesting to play with
- Standing up (Dyalog v17.1)
  - Started communicating more (WebSockets)
  - Going places (Windows, macOS and Linux)
Cross platform

- Windows / macOS / Linux
- Write once, deliver on any platform
- TamStat
# New and changed properties

Properties have been implemented on platforms that support them

<table>
<thead>
<tr>
<th>Property</th>
<th>Windows</th>
<th>Mac OS</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>HTML</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Posn</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Size</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>URL</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Coord</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Border</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Visible</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Event</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Sizeable</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Moveable</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SysMenu</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MaxButton</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MinButton</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IconObj</td>
<td>+</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>Data</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Attach</td>
<td>+</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>KeepOnClose</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>AsChild</td>
<td>+</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>InterceptedURLs</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>CEFVersion</td>
<td>+</td>
<td>+</td>
<td>!</td>
</tr>
<tr>
<td>Caption</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MethodList</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ChildList</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>EventList</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>PropList</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Properties have been implemented on platforms that support them.
New and changed properties

- Properties have been implemented on platforms that support them
- IconObj added
- CEFVersion added
- Caption added
  
  `<title>My Caption</title>
  Caption←'My Caption'

- InterceptedURLs didn't change… (sort of)
**InterceptedURLs**

- Introduced in v17.0
- Determines whether a request for a resource is passed to the net, or back to APL

```
<table>
<thead>
<tr>
<th><em>dyalog_root</em></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
</tr>
</tbody>
</table>
```

A `dyalog_root` is the "default" URL that would be prepended to requests for "local" resources

```html
<script src="https://somecdn.com/scripts/utils.js"/>
```
(via the net)

```html
<script src="/scripts/utils.js"/>
```
(via APL callback)

- Probably made obsolete in v17.1
New and interesting methods

- ShowDevTools
- Renderer.ShowDevTools 1
New and interesting methods

```html
<!DOCTYPE html>
<html>
<head>
  <style>
    #wrapper {
      position: relative;
      transform: rotateX(-90deg) rotateY(180deg);
    }
  </style>
  <script src="cubie.js"></script>
</head>
<body id="wrapper">
  <div id="cube" style="transform: rotateX(-90deg) rotateY(180deg);">
    <div class="face one">X</div>
    <div class="face two">Y</div>
    <div class="face three">Z</div>
    <div class="face four">X</div>
    <div class="face five">Y</div>
    <div class="face six">Z</div>
  </div>
  <p>This face intentionally left blank.</p>
</body>
</html>
```
New and interesting methods

- ShowDevTools
  - Renderer.ShowDevTools 1

- PrintToPDF
  - Prints rendering to PDF file
  - Renderer.PrintToPDF '/tmp/myfile.pdf'
New Events

- **SelectCertificate**
  
  Used when requesting a resource that requires a certificate. A dialog box opens to allow you to select the certificate to use.

- **DoPopup**
  
  When the client requests that a resource be loaded in a new window, a DoPopup event is triggered.
New Events

∇ {r}←DoPopupDemo2 args;prop
[1]    :If 0∊⍴args
[2]        prop←('Size'(10 20))('Event'('onClose' (⊃⎕SI)))('Event'('onDoPopup' (⊃⎕SI)))
[3]        Renderers←,⎕NEW'HTMLRenderer'prop  A create a new renderer
[6]        :Select 2⊃args  A select based on event
[7]            :Case 'DoPopup'  A DoPopup event?
[8]                Renderers←,⎕NEW'HTMLRenderer'('URL'(3⊃args))  A create a new renderer
[9]            :Case 'Close'    A Close window event?
[10]               ⎕EX'Renderers'
[12]    :EndIf
∇
WebSockets

- What's a WebSocket?
  - A bi-directional, asynchronous, communications channel between client and server
  - In standard HTTP, all requests are initiated from the client. The server can only respond to a request.
  - The client can request to "upgrade" the HTTP connection to a WebSocket, which the server can accept or reject.

- Once a WebSocket is established, either side can send a message. No response is required.
- Either side can close the WebSocket.
- You can have multiple WebSockets to any combination of local or remote servers.
WebSockets

- **Uses**
  - Pushing information
    - "financial ticker", chat services, notifications, education
  - Databinding
    - Keeping data in your workspace in sync with the data displayed in the GUI
      - Changes made in the GUI are updated in the workspace
      - Changes in the workspace are displayed on the client
Utilities and Frameworks

- HttpUtils
  - Utilities to convert HTTPRequest event arguments and results
Utilities and Frameworks

10>args
first=Dr.+Donald&last=L%27Duck

req←⎕NEW HttpUtils.Request args

req.FormData

<table>
<thead>
<tr>
<th>first</th>
<th>Dr. Donald</th>
</tr>
</thead>
<tbody>
<tr>
<td>last</td>
<td>L'Duck</td>
</tr>
</tbody>
</table>
Utilities and Frameworks

- **HttpUtils**
  - Utilities to convert HTTPRequest event arguments and results

- **EasyGUI**
  - A collection of interactions that can be used stand-alone or in combination
  - Josh will be presenting EasyGUI in the next presentation

- **DUI – Dyalog User Interface**  
  - Library for Web Content Creation
  - Server
    - MiServer for running over the net
    - HRSERVER (HTMLRenderer server) for running as a desktop app

---

#dyalog19

What's New with HTMLRenderer?
**DUI**

- **Page class**
  - Self-contained MiPage using internal HTMLRenderer

- **MiSite**
  - `DUI.Run '/path/to/misite'` A run locally using HTMLRenderer
  - `DUI.Run '/path/to/misite' port` A run on net using MiServer
In the pipeline

- A couple minor tweaks to HTMLRenderer
  - Extend default root (http[s]://dyalog_root) to allow other protocols (ws[s]:// for WebSockets)
  - Support references to file:// to read files directly rather than pass request back to the workspace

- DUI
  - DoPopup support
  - Databinding
  - Documentation, Tutorials
What's New with HMTLRenderer?