CEP 6 Feature List

CEP 6.0 Feature List for CC 2015 June release

- CEP 6.0 Compiler Upgrade
- Asynchronous interface for executing scripts
- Refactor IME Implementation on Mac OS X
- New PlugPlug IMS API IMS_fetchUserProfileData
- New JSON API to customize context menu
- New JavaScript APIs to tell if a CEP panel is hidden or shown
- Customize context menu - Be able to add icon to context menu items
- New JavaScript APIs for getting and setting panel window size
- Support notifications when fly-out menu is opened and closed
- Provide a fallback mechanism for localization
- Support undo and redo shortcut keys
- Support ESC key

CEP 6.1 Feature List for CC 2015 November Release

- CEP 6.1 integrated new version of CEF 3 and io.js.
    - Io.js is a fork of Node.js. It is supposed to be fully compatible with Node.js in CEP 6.0. Before asking why io.js, please read http://anandmanisankar.com/posts/nodejs-iojs-why-the-fork.
    - Io.js and Node.js had reunited again recently. So in next release we will probably use Node.js again.
- Support validating extension signature in CEP HTML Engine, in order to reduce host application launch time.
- Support remembering last location and size of CEP dialogs.
- Support getting and setting the title of extension windows.
- Node.js is disabled by Default
    - To enable Node.js for whole extension, specify '--enable-nodejs' in extension manifest.
    - To enable Node.js for an iframe, add property 'enable-nodejs'.
- Support handling invalid server certificate better.
    - When you try to open a URL, if server certificate is invalid, CEP will show an error page. You can also register a callback and use your own customized error page.
- Improvement of Key Event Handling
- Support localization of default items in context menu.
- Support loading panel extension but not showing it.
- Support specifying CEP dialog size as a percentage of screen size. Here is an example.

```
<UI>
    <Type>Modeless</Type>
    ...
    <Geometry>
        <ScreenPercentage>
            <Height>50%</Height>
            <Width>50%</Width>
        </ScreenPercentage>
        ...
    </Geometry>
</UI>
```

- Get and Set the title of extension windows (CEP 6.1 introduces two APIs to set and get the title of extension windows. Those functions work with modal and modeless extensions in all Adobe products, and panel extensions in Owl apps.)

```
CSInterface.prototype.setWindowTitle = function(title)
{
    window.__adobe_cep__.invokeSync("setWindowTitle", title);
};

CSInterface.prototype.getWindowTitle = function()
{
    return window.__adobe_cep__.invokeSync("getWindowTitle", "");
};
```

- Better JSON Support in CEP Event JavaScript APIs
    - Defects in CEP 6.0 and Former Releases
        - CEP 6.0 treats each attribute of the event.data object as a string. For example, if you pass the string below through a CEP event.

          ```
          {"myBoolKey": false,"myIntKey": 7,"myFloatKey":
          5.4,"myStringKey": "testテスト测试","myArrayKey":
          [5.4,true,false,{"yellow": true,"green": false},7]}
          ```

        - When you receive the event, event.data is an object with several attributes such as “myBoolKey” and “myIntKey”. But the value of those attribute are all strings. For example, the value of “myBoolKey” is “false” rather than false; the value of “myIntKey” is “7” rather than 7. If the value of an attribute is an array or JSON string, they are all treated as a regular string. Besides, CEP does not support non-ASCII characters in CEP events, especially on Windows platform. So, the result is:

          ```
          { myBoolKey: "false", myIntKey: "7", myFloatKey:
          "5.400000", myStringKey: "testXXXXX", myArrayKey:
          "[5.4,true,false,{"yellow": true,"green": false},7]"}
          ```

    - Improvement in CEP 6.1
        - CEP 6.1 keeps the type of all attributes in event.data as what they are. If the same string is passed through event.data by CEP 6.1, the result will be:

```
{
    "myBoolKey": false,
    "myIntKey": 7,
    "myFloatKey": 5.4,
    "myStringKey": "testテスト測试",
    "myArrayKey": [
        5.4,
        true,
        false,
        {
            "yellow": true,
            "green": false
        },
        7
    ]
}
```

- myArrayKey is an array object rather than a JSON string, and it has an anonymous object which includes “yellow” and “green” attribute.
- Therefore, Since CEP parses JSON and returns the parsed JavaScript object to you, you do not need to call JSON.Parse() to parse the result from event.data anymore.
- Besides, you can pass non-ASCII characters and binary data by CEP events.

CEP 6.1 Feature List for CC 2016 June Release

- Support the new "mixed context" mode for HTML extensions using node.js.
    - The "mixed context" mode is disabled by default. Unlike the default "separate context" mode where a "required" node module is in a separate JavaScript context, a "required" node module and the JavaScript code that "requires" it are in the same context in the new "mixed context" mode, eliminating all the inconveniences in the old "separate context" mode.
    - To enabled it, add command line parameter "--mixed-context" to your extension manifest.

      ```
      <Parameter>--mixed-context</Parameter>
      ```

- Refactor IPC Implementation on Mac.
- Bug fixes in CEF 3 library.
- Support fake top-level frame so that websites like twitter, google and facebook can be embedded.
- Support HTML contenteditable attribute.
- Support HTML tel input type.