
TLDR; How to build a panel development environment

1. Install everything

You'll need some things.

2. Configure the environment

Make Premiere Pro load unsigned panels, and generate useful information.

3. Create JavaScript using browser-based developer tools

Load your panel into your browser, for standard JavaScript debugging.

4. Create ExtendScript using ExtendScript Toolkit (ESTK)

Rapidly develop usable ExtendScript, in a live scripting environment.

5. Package and Deploy your panel

How to provide your panel (and any desired collateral files) to users.

Always feel free to ask questions!

1. Install Everything

Creative Cloud

<http://creative.adobe.com>

Use the Creative Cloud app to install Adobe Premiere Pro CC, Adobe ExtendScript Toolkit CC, and Adobe Extension Manager CC.

Adobe Extension Signing Toolkit

Note Panel creation no longer requires Extension Builder, but you still need the `zxpsigncmd` command line tool:

Win: <http://download.macromedia.com/pub/labs/extensionbuilder3/ccextensionswin64.exe>

Mac: <http://download.macromedia.com/pub/labs/extensionbuilder3/ccextensionsmac.zip>

PProPanel sample project:

<https://github.com/Adobe-CEP/Samples/tree/master/PProPanel>

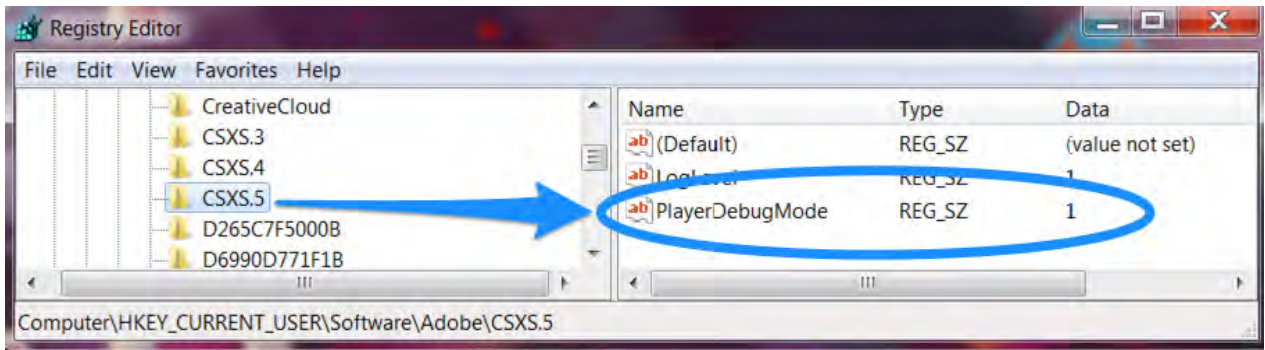
2. Configure the environment

Make Premiere Pro load unsigned panels

To change Premiere Pro's preferences, type the following into a Mac OS X Terminal, then relaunch Finder:

```
defaults write /Users/<username>/Library/Preferences/com.adobe.CSXS.  
5.plist PlayerDebugMode 1
```

On Windows, make the following entry into the registry:



Put panels here, to have Premiere Pro load them:

```
C:\Program Files (x86)\Common Files\Adobe\CEP\extensions
```

```
/Library/Application Support/Adobe/CEP/extensions
```

Optional: Enable Premiere Pro's console and additional log files

Show Premiere Pro's Console window using <ctrl-F12> (Windows) or <cmd-F12> (Mac). In that window, type the following:

```
debug.set CreateLogFilesThatDoNotExist = true [then press enter]
```

Find those logs here:

```
Win: "%AppData%\Roaming\Adobe\Premiere Pro\8.0\logs\Premiere Pro Log.txt"
```

```
Mac: "~/Library/Preferences/Adobe/Premiere Pro/8.0/logs/Premiere Pro  
Log.txt"
```

3. Create JavaScript using browser-based developer tools

To enable debugging of panels using Chrome's integrated debugger, put a file named `.debug` into your extension's folder (peer of the `/CSXS` folder). The contents of the file should resemble the following:

```
<?xml version="1.0" encoding="UTF-8"?>  
  <ExtensionList>  
    <Extension Id="com.example.PProPanel">  
      <HostList>  
        <Host Name="PPRO" Port="7777"/>  
      </HostList>  
    </Extension>  
  </ExtensionList>
```

Now, when the panel is active, you can debug the panel in your web browser by browsing to `localhost:7777`, and selecting your panel.

Optional: Turn on CEP logging

Find CEP logs here. Note that mac Library path is the system's library, not the user's.

Win: `~\AppData\Local\Temp\csxs5-PPRO.log`

Mac: `/Library/Logs/CSXS/csxs5-PPRO.log`

Set logging level in Windows Registry, or MacOS X .plist:

Win: `HKEY_CURRENT_USER/Software/Adobe/CSXS.5`

Mac: `~/Library/Preferences/com.adobe.CSXS.5.plist`

Optional: Configure CEP

It's possible to configure CEP runtime environment with environment flags:

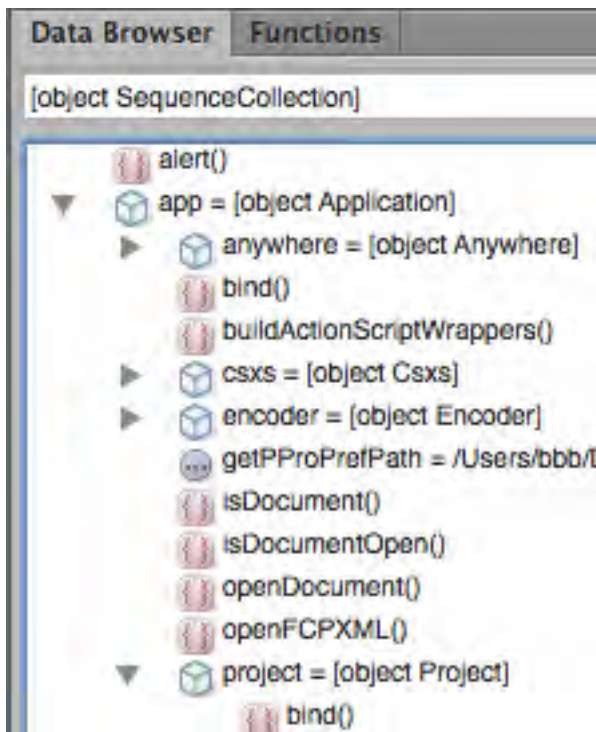
<http://peter.sh/experiments/chromium-command-line-switches/>

To use these flags from within your panel, insert them in your manifest:

```
<Extension Id="xx.yy.zz">
  <DispatchInfo>
    <Resources>
      <MainPath>./html/index.html</MainPath>
      <CEFCCommandLine>
        <Parameter>--enable-mediastream</Parameter>
      </CEFCCommandLine>
    </Resources>
    ...
  </DispatchInfo>
```

4. Create ExtendScript using ExtendScript Toolkit (ESTK)

To start a Premiere Pro scripting session, select 'Adobe Premiere Pro CC 2014' from the target application list, then click the chain link to launch and connect. Once in the session, use the Data Browser to view the DOM.



5. Package and deploy your panel

Obtain or create a signing certificate

To create: `ZXPSignCmd -selfSignedCert <countryCode> <stateOrProvince> <organization> <commonName> <password> <outputPath.p12>`

Example:

```
./ZXPSignCmd -selfSignedCert US California Adobe "Bruce Bullis"
password certificate.p12
```

Sign your panel

To sign:

```
ZXPSignCmd -sign <inputDirectory> <outputZxp> <p12> <p12Password>
```

Example: This assumes your panel directory, panelDir, is in the same directory as ZXPSignCmd.

```
./ZXPSignCmd -sign panelDir/ PanelName.zxp certificate.p12
password
```

Distribute the panel

You now have a cross-platform .zxp file which Extension Manager can install on any system with PPro installed. Send us a copy!