

Dr. Woohoo!

URL DRWOOHOO.COM

TEL 760.889.1247

drew@drwoohoo.com

ExtendScript Toolkit + SwitchBoard FAQs

General Questions

Q: I want to be able to control the CS apps from a script/swf, however I am a designer and/or I know how to write JavaScript/ActionScript 1/2. how shall I proceed?

A: I would suggest starting with the ExtendScript Toolkit (ESTK) application but it is the easiest and fastest way to begin extending the Creative Suite applications. If you own a copy of a CS3 or CS4 application, chances are it is already installed on your computer (refer to the ESTK FAQs below).

If you have written JavaScript, Processing, ActionScript 1 or 2, you will fill right at home in ESTK. Your greatest challenge is simply getting familiar with the API, or as Adobe likes to call it, the DOM of the Creative Suite application you are interested in extending.

When you are comfortable writing ExtendScripts, then you can move on to using one of the other more advanced tools like Photoshop Panel (creates a Flash panel using the CSXSLibrary.swc), PatchPanel and/or SwitchBoard.

Q: There are 4 different developer tools for extending the Creative Suite applications now: ExtendScript Toolkit, Photoshop Panel (creates a Flash panel using the CSXSLibrary.swc), SwitchBoard and PatchPanel (in prerelease). How do I know which one is the best for my needs?

ESTK is great for getting started with extending the Creative Suite applications via scripts. Learning how to control the ESTK compatible CS application is fast and relatively easy in ESTK because writing ExtendScripts fills like writing JavaScript and early ActionScript code and because it is not interdependent on other languages and development tools. It is a sketchpad that allows you to explore the possibilities. The knowledge you gain and the ExtendScripts you write when using ESTK can be transferred to the other 3 options with relative ease.

If you are interested in developing Flash Panels that run within Photoshop CS4, that leverage your knowledge and your library of ExtendScripts, Photoshop Panels (using the CSXSLibrary.swc) is a helpful tool – although now that PatchPanel is available on Adobe Labs, you might want to choose your next step carefully. Although it is possible to write all of your ExtendScript code within your ActionScript, if you have a lot of it, you will want to use ESTK to write longer ExtendScripts and Flex Builder 3 along with the CSXSLibrary SWC that is included with the Photoshop Panel Developer's Guide download to write your code. By using both tools, you can troubleshoot your ExtendScript code using the CSXSLibraryAIRLogger – although I highly recommend using PatchPanel if because you will be able to take advantage of the Flex Builder debugger which, in my opinion is superior.

When you want to have AIR applications extend the Creative Suite applications, the SwitchBoard SWC library for Flex Builder is the tool for you. The advantage is the fact that you inherit all of the capabilities of AIR, ActionScript 3 and à la carte features from the compatible CS3 and/or CS4 applications of your

choice. That includes Photoshop, Illustrator and InDesign. As you will see below in the SwitchBoard FAQs, ESTK still plays an important roll in conjunction with Flex Builder during the development process.

The final option is PatchPanel (PP), which is now available on Adobe Labs [here](#). Like the CSXSLibrary SWC included in the Photoshop Panel Developer's Guide, the PatchPanel SWC allows you to extend the compatible CS applications via a Flash panel. But there is a lot more. PatchPanel is also compatible with CS3 versions of Photoshop, Illustrator and InDesign. The main difference between the CS3 and CS4 versions of the Flash panel via PP is the fact that in the CS3 applications, you launch a Window object rather than a Panel and it's accessible typically through the File > Scripts menu rather than Windows > Extensions in Photoshop CS4. The major difference between PatchPanel and it's siblings is the fact that with the cs3.swc or cs4.swc in place, you can do all of your development within Flex Builder. That means code-hinting and completion, strict data-typing, re-factoring, *and* debugging etc., are all part of the deal. It's truly beautiful and deserves a standing-ovation.

ExtendScript Toolkit FAQs

Q: What is ExtendScript Toolkit (ESTK)?

A: ESTK is an application that allows you to write ExtendScripts (ES) – Adobe's enhanced version of JavaScript files – that are dedicated to extending the Creative Suite applications.

Q: How do I get ESTK?

A: If you have a CS3 or CS4 application, chances are you already have ESTK installed. The default location is:

CS3:

- <hard drive>/Applications/Utilities/ExtendScript Toolkit 2
- C:\Program Files\Adobe Utilities\ExtendScript Toolkit 2

CS4:

- <hard drive>/Applications/Utilities/ExtendScript Toolkit CS4
- C:\Program Files\Adobe Utilities\ExtendScript Toolkit CS4

Q: What CS applications can I control with ESTK?

A: Photoshop, Illustrator, After Effects, InDesign and Bridge. Even though ESTK has been around since CS1, the DOMs will evolve from version to version so the naming conventions of classes, methods and properties might change and/or disappear, but for the most part if you write an ES for a CS3 application, little to no changes will be required if you run it in a CS4 application.

DreamWeaver, Flash and FireWorks are controlled via JSFL scripts, which is, as far as I know, currently not supported by ESTK.

Q: Where is the documentation for ESTK located at?

A: There are several pdf documents that are worthy of your attention when you are first starting including the following: Adobe Intro to Scripting, ExtendScript Toolkit ReadMe and the JavaScript Tools Guide. In addition, the Photoshop/Illustrator JavaScript Ref and the Photoshop/

Illustrator Scripting Guides are the documents I refer to most often because they contain the DOM and simple examples of how to use the classes, methods and properties.

Because I typically need access to these documents when I'm working within ESTK, rather than searching your hard drive for them, you can copy them into the ExtendScript Toolkit 2/SDK/ or ExtendScript Toolkit CS4/SDK/ directory and the next time you restart ESTK you will have access to them via the Help menu.

CS3:

- <hard drive>/Applications/Adobe Photoshop CS3/Scripting Guide/Photoshop CS3 JavaScript Ref.pdf
- <hard drive>/Applications/Adobe Photoshop CS3/Scripting Guide/Photoshop CS3 Scripting Guide.pdf
- C:\Program Files\Adobe Photoshop CS3\Scripting Guide\Photoshop CS3 JavaScript Ref.pdf
- C:\Program Files\Adobe Photoshop CS3\Scripting Guide\ Photoshop CS3 Scripting Guide.pdf

CS4:

- <hard drive>/Applications/Adobe Photoshop CS4/Scripting/Documents/Photoshop CS4 JavaScript Ref.pdf
- <hard drive>/Applications/Adobe Photoshop CS4/Scripting/Documents/Photoshop CS4 Scripting Guide.pdf
- C:\Program Files\Adobe Photoshop CS4\Scripting\Documents\Photoshop CS4 JavaScript Ref.pdf
- C:\Program Files\Adobe Photoshop CS4\Scripting\Documents\Photoshop CS4 Scripting Guide.pdf

These documents can also be downloaded at:

[Adobe Introduction to Scripting](#)
[Photoshop Scripting](#)
[Illustrator Scripting](#)
[InDesign Development Center](#)
[InDesign CS3 Scripting Guide](#)

Q: What is the Object Model Viewer?

A: The Object Model Viewer, accessed from within ESTK from the Help menu, allows you to browse and search the DOMs of all supported target applications, including the ScriptUI and Core JavaScript classes.

Q: How do we extend the DOM if a feature/function is not included in it?

A: It's possible to add your own custom methods to the DOM during runtime. With that said, Photoshop is the only CS4 application that ESTK supports with a plug-in that can record what features and functions a user is interacting with when the plug-in is installed. By default, the ScriptingListener plug-in is not installed. For more information on installing the ScriptingListener plug-in and extending the DOM, refer to Jeff Tranberry's posting [here](#).

Q: Can I create windows with UI components?

A: Yes. ESTK can create three different types of window objects: Modal, Floating Palette & Main. Refer to the JavaScript Tools Guide CS4 (or CS3) and search for 'Creating a window' for more information. ESTK includes standard UI components for developers.

Q: Can I localize it?

A: ESTK does support localization. Refer to the JavaScript Tools Guide CS4 (or CS3) and search for 'Localization in ScriptUI objects' for more information.

Q: Can I use an ES in several different CS applications?

A: Yes. A single ES can be used between multiple applications, however, due to the differences in the DOMs between each CS application, you will need to determine what the current host application is and use the appropriate code set. This could be done by simply defining the #target and a variable with the name of your target application.

Q: Can I use an ES in a single CS application and communicate with another CS application?

A: Yes. Refer to the JavaScript Tools Guide CS4 (or CS3) and search for 'Interapplication Communication with Scripts' for more information.

Q: Can I use ES cross platform?

A: This is a trick question. ESTK supports C/C++ integration, VBScript, AppleScript and Adobe's own version of JavaScript which is called ExtendScript. From my perspective, ExtendScripts provide the easiest path for cross platform development – when we are only considering the use of ESTK as the development tool. When we look at Photoshop Panels (using the CSXSLibrary.swc), SwitchBoard and PatchPanel, even though the learning curve increases, the ease of development once you are used to the dev tools increases dramatically – along with the new functionality that comes with using AIR and ActionScript.

Q: Can I use ESTK and Flash/Flex to create a window with a SWF that can communicate with my ES methods?

A: Yes. It is technically possible, however, using this approach is not compatible with CS3 applications running on Mac's Leopard. It is highly recommended that you consider using one of the SWC libraries in conjunction with Flex Builder and ESTK in order to create what are essentially Flash Panels that run within the CS host application. Adobe's Photoshop Panels (using the CSXSLibrary.swc) and PatchPanel (currently available only in prerelease) are two similar SWC libraries are the two dev tools of choice. If you are interested in creating AIR applications that extend the CS3 & CS4 applications, SwitchBoard is the Flex SWC library for you. More on SwitchBoard below.

Q: What scripting language/version is ES like?

A: It feels like JavaScript, Processing or an early version of ActionScript, e.g., strict data-typing is not a requirement. If you are just beginning to write scripts, this could be a plus. If you are an advanced programmer spoiled by all of the new features that are offered in dev tools like Flex Builder, you will fill more at home using PatchPanel to create your Flash plug-ins for the CS apps because it supports strict data-typing, code-completion, etc.

Q: How do you install ES so they are accessible to the CS host application?

A: Save your script to:

CS3:

- <hard drive>/Applications/Adobe Photoshop CS3/Presets/Scripts/
- C:\Program Files\ Adobe Photoshop CS3\Presets\Scripts\

CS4:

- <hard drive>/Applications/Adobe Photoshop CS4/Presets/Scripts
- C:\Program Files\ Adobe Photoshop CS4\Presets\Scripts\

If this is the first time you are installing your script, restart Photoshop and your script will be available in File > Scripts. If you have already installed your rescript and started Photoshop after the fact, you do not need to restart Photoshop again.

Q: How does a user access the ES?

A: Your script will be available in Photoshop under File > Scripts.

Q: Is there a way to write ES in Flex with maybe JSEclipse?

A: Yes. However, it will be near impossible to debug within Flex in this manner. If you are interested in developing only in Flex Builder, the PatchPanel (currently only available in prerelease) SWC library is the tool for you.

Debugging

Q: How do I run a trace?

A: You can use any of the following to output your results in the JavaScript Console window within ESTK: `$.write()` or `$.writeln()`. `alert()` will launch a dialog prompt within the target CS application.

Q: How do I comment out code?

A: Using the same technique that you would in ActionScript:
`// or /* */`

Q: I've reviewed the DOM document and I am trying to retrieve runtime data on a specific object. How do I do that within ESTK?

A: Set breakpoints in your code. The Data Browser panel within ESTK contains a realtime view of all the DOM objects as well as your variables. You can view the values of your variables, expand arrays and objects to see the contents, etc. For more information, refer to the JavaScript Tools Guide CS3 (or CS4) and search for Tracking data or Data Browser.

SwitchBoard FAQs

Q: What is SwitchBoard (SB)?

A: Adobe® SwitchBoard is the code name for an Adobe Flex® library and set of services that make it possible for Adobe AIR applications to work with Adobe Creative Suite® (CS3 and above) applications. The SwitchBoard installer installs two services onto your machine:

- ▶ The SwitchBoard Service forwards messages between your AIR application and Creative Suite applications.
- ▶ The SwitchBoard Launcher launches target applications as needed.

AIR developers need only include a Flex library (SwitchBoard.swc) in their projects, to send and receive scripts to and from other Creative Suite applications.

The URL for SwitchBoard on the Adobe Labs Web site is:
<http://labs.adobe.com/wiki/index.php/SwitchBoard>

Q: What CS apps can I control with SB?

A: Photoshop, Illustrator and InDesign CS3 *and* CS4.

Q: What are the requirements for a developer?

A: SwitchBoard is free. The full developer and end-user requirements are located on Adobe Labs [here](#).

Q: I plan to distribute my AIR/SB application and would like to send out update notifications. is this possible?

A: Yes. SwitchBoard is a set of services that build on top of all the features and functions of an Adobe AIR application. Anything an AIR application can do, including update notifications, an AIR/SB application can do.

Q: I plan to distribute my AIR/SB as a commercial application but would like to include registration and authentication to prevent unauthorized use of it. is this possible?

A: Yes. Although this is a service that is not provided by Adobe, it is possible for a third-party developer to create the registration and authentication back-end required.

Debugging

The process for debugging your mxml/AS code will be the same. Debugging the ES code is a little more complicated, requiring you to use ESTK in conjunction with Flex Builder.

Q: How do I debug my embedded ES/JS code (which is highly recommended)?

A: I start out by creating an ES file that contains one or methods in it within ESTK as a sketch. I will call that method and pass it any hardcoded values within ESTK while targeting the application I intend to extend. Using this approach allows me to verify the code is working prior to integrating it into the Flex environment using one of the new extensible SWC libraries. If there are problems with the script, I can set breakpoints, step through the objects within the Data Browser panel and output to my JavaScript Console if necessary. Once I embed this script into my Flex Builder project, I am only able to do the latter at that point in time.

Q: Async communication: using handlers to get the results back from the host application.

A: SwitchBoard uses BridgeTalk on AIR to pass the code from the AIR application to the target CS application. BridgeTalk communicates asynchronously with the host application, so if you want to handle responses coming back from the target CS application, you can set up several different types of handlers including: `onResult` and `onError`. For more information on using these handlers, refer to the SwitchBoard Getting Started Guide pdf and search for 'Step 3: Specify how to handle a response'.

Q: How do I prevent an error in my code from stopping the application?

A: It's highly recommended that when you are about to send your message via BridgeTalk to the target application, that you wrap it up in a Try/Catch clause. My typical method for sending the message looks like the following:

```
public function sendThruBridgeTalk( myMsg : String, target : String ):void
```

```

{
    //create the Message
    var msg : Message = new Message;
    msg.target = target;
    msg.body = myMsg;
    // set the onResult handler
    msg.onResult = function( msg : Message ) : void
    {
        results.text = "Results: " + msg.body;
    }
    //set the onError handler
    msg.onError = function( msg : Message ) : void
    {
        // put the error message into the result box
        results.text = msg.body;
    }

    try
    {
        msg.send();
    } catch( e : Error )
    {
        results.text = "Error: " + e.toString();
    }
}

```

Q: I'm debugging my AIR application, which was able to communicate with the CS app a minute ago but now when I do this [click a button, etc.], nothing happens. suggestions?

A: There are times when I will be heavily testing my AIR application and then suddenly the two-way communication between the AIR application falls silent. There are two ways you can confirm that the communication channel between both applications is open.

- 1). Open the AIRSend AIR application that was included with the SwitchBoard SDK and run a trace from it to the target application that you are working with. The menu should be populated with a list of the different CS applications that are accessible. If the list is empty, there could be a problem with the SwitchBoardLauncher service.
- 2). On a Mac, open your Activity Monitor application (/Applications/Utilities) and verify that the SwitchBoardLauncher is running. On a PC, you can verify that it is running within Task Manager. If it is not running or it lists it as Not Responding, you will need to reboot your computer, which will restart the service.