ADOBE CONNECT SDK 11.0

Adobe Connect offers the unique ability to have the functionality of a meeting room extended by adding a 'Custom Pod', much the same way a smartphone is enhanced by installing an 'app'. These custom pods can be used to allow multi-user interactions within the meeting, they can provide a synchronized display of information for all users and can even send and receive data from external websites. Simple examples include a countdown-timer that lets a host show everyone when the meeting will start, or an interactive game that displays a leaderboard with all the participants' scores, or a map application that combines user location data with map data from an external mapping website. The possibilities are endless! You can sample other custom pods that are available from our Apps page – www.adobeconnect.com/apps



Developers can create these applications using standard HTML and JavaScript programming. The Adobe Connect SDK provides a set of standard programming interfaces (Methods, Events, Properties, and Constants) that can be added to an application to enable meeting specific activities. The developer might want choose which variables are synchronized for all users in the meeting (such as the countdown timer value), or to limit the permissions to view certain UI elements or enter data in a field (so only hosts can set the time in the countdown), or even to change the behavior of a custom pod when it is being viewed in a recording.

A custom pod is delivered as a zip package containing source files, a manifest, and libraries. Meeting hosts simply load this zip package into an empty Share pod to install the application into any meeting room, or the package can be uploaded to the Content library to be accessed across meeting rooms.

Adobe Connect SDK Introduction

The Adobe Connect SDK 11.0 includes the ability to create and upload custom pods that can be viewed by attendees in both HTML-based and Adobe Flash-based clients. In order to achieve this, two versions of each custom pod must be developed, one HTML-based for participants joining with the HTML client or with the Windows application (based on the Chromium Embedded Framework), and one Flash-based for anyone joining the meeting in the Adobe Connect application on the Mac, or with Flash in a browser. For this release, both versions must be combined into a single 'hybrid' zip container and uploaded into the meeting, however the Flash-based interface will soon be deprecated and only the HTML-based application will be required. If you do not require Flash support – a simple Flash (swf) 'stub' file can be substituted to satisfy the package requirements. We will soon eliminate the need to include any Flash files or libraries.



THIS DOCUMENT DESCRIBES:

- Development platform
- Sample Custom Pods
- Package Structure
- Coding Guidelines
- Message Handling
- Code Samples

DEVELOPMENT PLATFORM

Developers can create different levels of custom pods.

- 1. Applications that do not require synchronization such as a clock that displays the user's local time. This kind of application does not actually leverage the SDK, but it is still required for consistency.
- 2. Applications that are plain HTML and JavaScript.
- 3. Applications that utilize a framework such as React.js.

Though you could use any framework for your JavaScript development, the Adobe Connect meeting interface uses React.js and there may be performance advantages to using this framework for your custom pod development.

Adobe Connect SDK Introduction

SAMPLE CUSTOM PODS

We have included three basic custom pod examples to help understand some of the features and package structure.

SimplestCustomPod (plain HTML and JavaScript)

SIMPLESTCUSTOMPOD.ZIP	•••
Send Sync Message	
Sync Message Name:	
Sync Message Value:	
Send Message	
Message From Flash	

BasicListSync (uses React.js)

BASICLISTSYNC.ZIP			Stop Sharing	1	€.	[]] •••
Sending Sync Event						
Sync Message Name	Event Acknowledgemer	t From Flash	Allow Publish For this	s Event		
Sync Message Data			Event Name			
Send Sync Message		h	Send to Flash			
Checking Info Method	S					
GetBreakoutDetails GetMyUserDetails		Output of Info Method				
GetUserList		false				
GetMeetingInfo GetPodInfo)			1.		
GetConfig						
Listening to Local Ever	nts					
EventLogger		(Make API Call)				
caughtUp Called	11					

PACKAGE STRUCTURE

At a minimum, a custom pod must include a version of the custom pod as a Flash binary file (swf), the same pod as an HTML application (htm), a manifest file to list all the files as assets **breezemanifest.xml**, and a **lib** folder containing the JavaScript library that provides synchronization of events between them **connect_customPodSDK.js**. You may include additional support files provided they are referenced correctly within the manifest file.

▼ 🛄 CustomPod
breeze-manifest.xml
CustomPod_html5.htm
🛃 CustomPod.swf
🔻 🔜 lib
connect_customPodSDK.js

These documents must be compressed into a single archive format (zip) file before being loaded into an Adobe Connect Share pod or the Adobe Connect Content library.



Note – it is possible to develop a 'non-synchronized' custom pod that does not require the *lib* folder containing the *connect_customPodSDK.js* file. This is ideal for custom pods that read only local data, such as a clock, or only provide local interactions, such as a calculator.

HTML documents must be indicated by the extension *.htm*, while Flash binary files must include the *.swf* extension .

DO NOT USE THE '.html' EXTENSION FOR HTML FILES.

Primary HTML file naming convention: *fileName_html5.htm*.

Please ensure you include the *html5.htm* with your filename.

Example: If you have created a custom pod called PodExample, then you would have the following files:

PodExample.swf

PodExample_html5.htm

Note: The custom pod zipped folder should have all files and the *lib* folder at the root directory and not inside any other folder.

If a file is listed in the *breeze-manifest.xml* but that file is not present in the zip package, then Adobe Connect will give a conversion error when uploading.

CODING GUIDELINES

SECURITY SANDBOX

You can load external HTML pages into a Custom Pod iFrame, however you must only call pages from the same origin within the same frame. X-Frame-Options for the Adobe Connect client is set to 'sameorigin'. The security sandbox will block calls to other origins and prevent pop-ups.

INTEGERS

Due to limitations within our transmuxing service (ACTS), we have restricted integers to 4-bytes.

Current ranges for integers are:

int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295

As a workaround the integer message can be sent as a *String* and then can be casted back to integer.

MESSAGE HANDLING

dispatchSyncMessage

When this is called from the custom pod SDK it passes the data from JavaScript in HTML to ActionScript Code in Flash and vice versa when called from Flash.

dispatchSyncMessage(msgName, msgVal, isDelta, echoBack)

e.g. dispatchSyncMessage("name", "My Pod", false, false)

CODE SAMPLES

INTIALIZE LIBRARY

//Initialize library

// while using React it is recommended that you initialize the library inside componentDidMount method

var cpu = ConnectCustomSDK.SyncConnector | | {};

// to inform the main client that library is getting initialised

cpu.init(onConfigured, "com.adobe.connect.basiclistsync","10.5","connectsdkhook");

cpu.registerCallback("userLeft", updateUserList); cpu.registerCallback("userJoined", updateUserList); cpu.registerCallback("userDetailsChanged", updateUserList); cpu.registerCallback("userStatusChanged", updateUserList); cpu.registerCallback("roleChanged", updateUserList); cpu.registerCallback("syncMessageReceived", syncMessageReceived); cpu.registerCallback("caughtUp",caughtUp); cpu.registerCallback("update", breakOutSession);

onConfigured = function() { /*code to load the content of html*/

```
}
```

BREEZE MANIFEST

```
<?xml version="1.0" encoding="utf-8" ?>
<breeze-manifest version="1.0" xmlns="http://breeze.macromedia.com/ns/breeze-manifest" generator="ZB">
```

```
<document type="custom-pod" id="com.adobe.connect.reference" version="1.0.001"
minimumConnectMobileVersion="2.4" minimumSDKversion="9.4.002" minimumConnectServerVersion="9.4.2"/>
```

<assets>

```
<asset type="document-view">
	<entry href="CustomPod.swf" href-swf="CustomPod.swf" href-html5="CustomPod_html5.htm" />
	<file href="CustomPod.swf"/>
	<file href="CustomPod_html5.htm"/>
	<file href="lib/"/>
	<file href="lib/"/>
	<file href="lib/connect_customPodSDK.js"/>
	</asset>
</asset>
```

</breeze-manifest>