

ADOBE® CONNECT™ 9

Collaboration Builder Toolkit SDK



Legal notices

For legal notices, see http://help.adobe.com/en_US/legalnotices/index.html.

Contents

Using the Adobe Connect Collaboration Builder SDK 2.0

Introduction	1
Development environment	1
Use case examples	1
Basic List Sync sample application	2
Building and testing applications	2
Using the SyncConnector API	4

Using the Adobe Connect Collaboration Builder SDK 2.0

Introduction

Use Adobe® Connect™ Collaboration Builder SDK to develop Flex® applications that add new functionality to Adobe Connect Meetings. The SDK contains the following:

- SyncConnector.swc—a Flex component that allows you to synchronize multiple SWF files loaded into Share pods in Adobe Connect meeting rooms.
- ConnectionEmulator.swc—a Flex component that connects to the local SWF file that contains the ServicesEmulator.swc component.
- ServicesEmulator.swc—a Flex component that emulates the synchronization functionality of Adobe Connect server in a local SWF file.
- Source code and examples.
- Help files.

Download the Adobe Connect Collaboration Builder SDK at www.adobe.com/devnet/adobeconnect.html.

Development environment

This SDK is for Flex developers who have a working knowledge of Flex, Flex components, and ActionScript® and want to develop collaborative Adobe Connect applications.

Collaboration Builder applications must use the Flex 3.5 SDK and be compiled for Flash Player 9.

Use case examples

Typical applications built with the Collaboration Builder SDK have one continuous shared state, with little synchronization traffic devoted to private states. The following are a few sample applications that can be built with the Collaboration Builder SDK:

- Synchronized content navigation (for example, Adobe® Presenter, Adobe® Captivate™, and brochureware)
- Dashboard applications (for example, sales dashboards with shared state and navigation)
- Collaborative applications with shared state (for example, mortgage or ROI calculators and shared text editors)
- Multiuser games
- E-learning content

In general, any applications with shared states are good candidates for the Collaboration Builder SDK.

Note: *The Collaboration Builder API (application programming interface) does not allow access to the Adobe Connect object model.*

Basic List Sync sample application

Run the sample application in Adobe Connect

- 1 Launch an Adobe Connect meeting room, for example, <https://my.adobeconnect.com/yourmeetingname>.
- 2 Do the following to upload the Basic List Sync application to the meeting's Share pod:
 - a In the Adobe Connect meeting, choose Pods > Share > Add New Share.
 - b In the Share pod, click the Share My Screen drop-down list and select Share Document.
 - c In the Select Document to Share dialog, click Browse My Computer, and browse to the following file:
CollaborationSDK\Samples\Output\BasicListSync.swf.
- 3 Repeat steps 1 to launch another instance of the same room.
- 4 Select items in the list and see them synchronized in both rooms.

Open the sample application in Flash Builder

The Basic List Sync application is a simple multiuser application. Analyze it on your computer and run it in an Adobe Connect Meeting room.

- 1 Open Flash® Builder®.
- 2 Choose File > New > Flex Project.
- 3 In the New Flex Project dialog add a name in the Project Name field. Click Finish.
- 4 Right click on the src folder under the new project created and select the Import option.
- 5 In the Import dialog under General Section select File System. Click Next.
- 6 From the Directory field browse to Sample Apps\Basic List Sync\src folder.
- 7 Select BasicListSync.mxml. Click Finish.
- 8 Right click on the project created and select Properties.
- 9 Select Flex build Path option and click on the Library Path tab.
- 10 Click on Add SWC and browse to libs/SyncConnector.swc. Click OK.
- 11 Right click on BasicSyncList.mxml and choose Set Default Application.
- 12 Right click src and choose delete to remove all the other files except, BasicListSync.mxml.

Building and testing applications

Create an application with the Collaboration Builder SDK

To create an application with the Collaboration Builder SDK, link your project to SyncConnector.swc under the libs folder.

The SyncConnector component is the core of the SDK; it contains the API that lets the Collaboration Builder application communicate with Adobe Connect.

The emulator components, ConnectionEmulator and ServicesEmulator, allow you to test applications locally, without using Adobe Connect.

- 1 Open Flash Builder.
- 2 Choose File > New > Flex Project and do the following:
 - a Enter a Project name and click Next.
 - b Configure the Output folder and click Next.
 - c In the Library path panel, click Add SWC Folder. Browse to the Collaboration Builder SDK/Libs folder and click OK. This folder contains the files ConnectionEmulator.swc, ServiceEmulator.swc, and SyncConnector.swc.
 - d Click Finish.
- 3 Use the SyncConnector API to build an application. Add a SyncConnector component to the application:

```
<ns1:SyncConnector x="285" y="70" id = "syncConnector" syncMessageReceived =  
"syncMessageReceived(event)" caughtUp="caughtUp(event)"/>
```

Create methods syncMessageReceived and caughtUp. For details, see the basiclist sample. For the complete list of events, see the SyncConnector API documents, located in the CollaborationSDK/help folder.

- 4 Create the methods syncMessageReceived and caughtUp. See the basiclist sample. The syncConnector instance is the gateway to all the events room of Adobe Connect. For a complete list of events please refer to the SyncConnector API documents.

Test an application in Flash Builder without Adobe Connect

- 1 Do one of the following:
 - Create a Flex file with a ServicesEmulator component in it. Compile the Flex file and leave the SWF file running.
 - Add the sample code:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"  
xmlns:ns1="com.adobe.sync.components.*">  
    <ns1:ServicesEmulator/>  
</mx:Application>
```

This file ServiceEmulatorApp.swf is present in the output folder. You can also use this existing file.

- 2 To test the application without Adobe Connect, add a ConnectionEmulator component to the application:

```
<ns1:ConnectionEmulator x="285" y="150" id = "connectionEmulator"/>
```

Before you run the application in Adobe Connect, comment out the ConnectionEmulator component.

- 3 Open the Collaboration Flex project which needs to be tested. Add the following line in the main application file:

```
<ns1:ConnectionEmulator id = "connectionEmulator" bSyncConnector="{syncConnector}"/>
```

- 4 Compile the Collaboration Builder application.

Note: Add serviceEmulatorApp.swf and your application SWF as trusted files in the Flash Player settings. For more information, see [this settings page](#). Remove these files from the trusted setting after testing.

- 5 Run another instance of the Collaboration Builder application SWF file (usually in stand-alone Flash Player).

This SWF file syncs with the SWF file compiled in Flash Builder.

- 6 When you have tested your application's logic locally, remove the ConnectionEmulator component from the application's MXML file and recompile the SWF file.
- 7 Upload the SWF file to the Share pod of an Adobe Connect Meeting room.

- 8 Open another instance of the same meeting room to observe the synchronization of the Share pods.

Test an application in Adobe Connect

- 1 If the application contains the ConnectionEmulator component, remove it from the application or comment it out of the application.
- 2 Do the following to upload the Collaboration Builder application to the meeting's Share pod:
 - a In the Adobe Connect Meeting, choose Pods > Share > Add New Share.
 - b In the Share pod, click the Share My Screen drop-down list and select Share Document.
 - c In the Select Document to Share dialog, click Browse My Computer, select the SWF file, and click Open.
- 3 Open another instance of the same meeting to observe the synchronization of the Share pods.
If there is a problem with your application, return to Flash Builder to fix the error and compile a new SWF file.
Repeat this procedure.

To retest an application:

If a previous version of the Collaboration Builder application is running in an Adobe Connect Meeting, do the following:

- 1 In the Share pod that contains the application, select the menu and choose Hide.
- 2 In the meeting room, select Pods > Manage Pods. and delete the old version of the Collaboration Builder application.
- 3 Select the Share pod that contained the Collaboration Builder application, and click Delete. Click Done.
- 4 Close the meeting room and reenter it.

Using the SyncConnector API

The SyncConnector component is the core of the Collaboration Builder SDK. Methods on SyncConnector send *sync messages* to Adobe Connect. SyncConnector dispatches events to notify the application that messages were received or that some other state (for example role, sync, or unsync) has changed.

To build an application with the SDK, add the SyncConnector component as a library in a Flash Builder file and as a component in the application. Use the SyncConnector API to communicate with other instances of the Collaboration Builder application.

Communicating between application instances

The Collaboration Builder SDK uses *sync messages* to communicate between application instances. A sync message is an object sent to each client that contains information about one aspect or property of the application; a sync message updates a client.

To send sync messages, call `SyncConnector.dispatchSyncMessage()`. To receive sync messages, use the `SyncConnector.syncMessageReceived` event. This method and event combination is the basis of communication in the SDK.

Sync messages can be any type or structure of data—they are typically, but not necessarily, objects.

Each message has a message name that you select; it's a good idea to use the name of the property the message pertains to. For example, when you synchronize the `selectedIndex` property of a List component, you might choose the message name "selectedIndex".

Adobe Connect saves all sent messages. It uses the saved messages for recording and to catch up users who join late.

Note: Because of security concerns, by default, meeting participants cannot call the `dispatchSyncMessage()` method to dispatch messages through Adobe Connect. A host or presenter can call the `SyncConnector.allowParticipantPublish()` method to allow participants to publish a specified message.

Understanding the catch-up phase

Applications that use the SyncConnector component go through a catch-up phase as they are loaded into a Share pod in a meeting room. This phase allows users who are new to the room to receive the accumulated state of the application. Adobe Connect stored sync messages as they were originally sent, and resends them privately to the new user. During the catch-up phase, the value of the SyncConnector component's `isCaughtUp` property is `false`. After all saved messages are received, SyncConnector dispatches a `caughtUp` event.

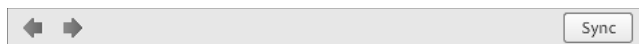
Note: You may want your application to handle sync messages differently depending on whether the application is caught up to the full model.

The following events are dispatched in the following order before the catch-up phase begins:

Event	Description
<code>whiteBoardToggle</code>	Changes the <code>isWhiteBoardOn</code> property from undefined to false.
<code>pointerToggle</code>	Changes the <code>isPointerOn</code> property from undefined to false.
<code>roleChanged</code>	Changes the <code>role</code> property from undefined to the current user role
<code>syncModeChanged</code>	Changes the <code>isSynced</code> property from undefined to true.
<code>sizeChanged</code>	Changes the <code>podHeight</code> and <code>podWidth</code> properties from undefined to the height and width of the Share pod hosting the application.

Understanding the meeting sync state

The SDK allows applications to operate in either a synced or unsynced state.



Toggle the Sync button in the Share pod to change the sync state.

The SyncConnector component has an `isSynced` property, and a corresponding `syncModeChanged` event. These APIs correspond to the Sync button in the Share pod. Sync messages are sent and received regardless of the sync state. An application may choose to ignore sending or receiving outside events when the value of `isSynced` is `false`. Likewise, when not synced, the application may want to display a different user interface that allows participants to navigate where previously only presenters could (for example, as in Adobe Presenter presentations).

Types of sync messages

The SyncConnector component supports two types of sync messages: *stateful* messages and *delta* messages. The type of message you send affects the way Adobe Connect saves your messages. The type of message also affects the way Adobe Connect retrieves messages later for new users to catch up with the application state.

Stateful messages represent the entire state of a specific aspect or property. Stateful messages are the more common type of message. With stateful messages, Adobe Connect saves and sends the last message of a given message name only during the catch-up phase.

Delta messages must be taken as a queue (series) to build the entire state of an application. With delta messages, Adobe Connect saves all the messages of a given message name as a queue and sends the entire queue during the catch-up phase. Delta messages are practical for messages that would otherwise be too large to send.

Consider the following example: you want to synchronize two aspects of a shared text editor—the user name of a typing user, and the contents of the text area.

For the user name, use stateful messages named `whoIsTyping`, because only the last message is important—it contains the entire state of that property. If a new user joins during typing, the new user receives only the last message.

For the contents of the text area, use delta messages named `textChanges`. As people type (imagine an increasingly long document), you can send the new text as it's being added, not the entire contents of the text area over and over again. If a new user joins during typing, Adobe Connect sends the queue of `textChanges` messages, so the text area can be filled. This case is uncommon —most messages are stateful—but essential for certain situations.

To send a message as a delta, use the SyncConnector component's `dispatchSyncMessage()` method with the optional third parameter, `isDelta`, set to `true`. To clear a queue of delta message, send one stateful sync message with that message name. You can then add a queue of new delta messages.