



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

## API Reference

External Media Adapter API

---

## Contents

- 1 External Media Adapter API
  - 1.1 initializeAsync(done)
  - 1.2 sendCbSessionToken(token)
  - 1.3 isAgentConnected()

# External Media Adapter API

The External Media Adapter API can be used to integrate Co-browse with your external media; for example, with external chat.

An external media can be connected to Co-browse via an adapter. An adapter is a JavaScript object that is assigned to the `_genesys.cobrowse.primaryMedia` option and implements the specified interface. An external media adapter may implement the following methods:

## initializeAsync(done)

Implement the `initializeAsync` method in your external media adapter when the external media initializes asynchronously and you cannot be sure the external media is ready as it is passed to the instrumentation. This method may start the (asynchronous) external media's initialization or it may subscribe to the initialization if the external media is started elsewhere.

If the `initializeAsync` method is implemented, the Co-browse JavaScript will call the method and pass it a `done` callback. You must call the `done` callback when your media finishes initialization.

### Important

Note the following about the `initializeAsync` method:

- This method is called by the Co-browse JavaScript Application every time it initializes such as after every page load.
- This method is called **before** the Co-browsing button is shown. The button will be shown only after you call the passed `done` callback in your code.

The following is an example of an external chat adapter named `MyChatAdapter`:

```
function MyChatAdapter() {
  // initialize chat
  this.initializeAsync = function(done) {
    $.get('/chat/configuration', function(config) {
      var chat = new MyChat(config);
      // tell cobrowse chat is ready
      done();
    });
  };
};

// or if you have a chat with event-based API that is initiated elsewhere
function MyChatAdapter() {
  this.initializeAsync = function(done) {
    myChat.on('initialized', done);
  };
};
```

```
};
```

### Tip

You can use the `initializeAsync` method to restore your external media after a page reload. For example, if you have a chat with a `restoreChat` function that needs to be called after page reload, you can call this `restoreChat` method in the `initializeAsync` method of the external media adapter passed to Co-browse.

Example:

```
// in the adapter:
myChatAdapter.initializeAsync = function(done) {
  myChat.restoreChat().then(done);
};

// ...
// and then in Co-browse instrumentation
var _genesys = {
  cobrowse: {
    primaryMedia: myChatAdapter
  }
};

// Now after every page reload Co-browse will
// automatically restore your chat.
```

## sendCbSessionToken(token)

Implement this method to configure the external media channel to pass the Co-browse session token to the agent. The Co-browse session token is a string consisting of 9 digits.

Example:

```
myChatAdapter = {
  sendCbSessionToken: function(sessionToken) {
    myChat.sendMessage('User has started Co-browse session: ' + sessionToken);
  }
};
```

### Tip

You may customize how the Co-browse token is passed to the agent. If you use a Genesys agent desktop, such as Workspace Desktop Edition with the Co-browse plugin or Workspace Web Edition, you may want the agent to join a Co-browse session as soon as he or she receives the token. To do so, wrap the Co-browse token in a `{start:<TOKEN>}` message.

Example:

```
// For example:  
myChatAdapter.sendCbSessionToken = function(token) {  
  myChat.sendMessage('{start:' + token + '}');  
};
```

### isAgentConnected()

#### Important

This method must return a boolean.

The integration module checks the return value of this method before calling the `sendCbSessionToken` method. If `isAgentConnected` returns false, the user will be asked to connect with an agent via phone before starting Co-browse. If the `isAgentConnected` is absent, the user will be asked to connect with an agent via phone or chat before starting Co-browse.