**Cordova and Ionic Plugin Installation - Mobile SDK 7.1138.0 or Later**

Overview

This article explains how to install, start, and test the Glassbox Cordova and Ionic Plugin in the Mobile SDK version **7.1138.0** and later. The plugin is for Cordova apps that do not contain native screens.

If you are using the Mobile SDK version **7.1136.0** or earlier, do not follow the instructions in this article. See [Cordova and Ionic Plugin Installation - Mobile SDK 7.1136.0 or Earlier](https://support.glassbox.com/s/article/17167469902994#UUID-f41a8dc6-5fac-4cc9-634a-83a16a4a86c7).

If your Ionic app uses Capacitor, install the Glassbox Capacitor Plugin instead of the Cordova and Ionic Plugin. See [Capacitor Plugin Mobile SDK Installation](https://support.glassbox.com/s/article/11686217490321#UUID-bf3981d8-cc7e-69bd-e4ff-291a3ce8c20d).

When you install the Cordova and Ionic Plugin, both the Glassbox Mobile SDK and its Cordova and Ionic extensions are automatically installed.

[Learn more about the Mobile SDK](https://support.glassbox.com/s/article/5790432440977#UUID-27b5d280-d78e-aa4b-3d93-47f2a8840ead)

**Note:** Section 1. Install the plugin applies to remote installation only. If your environment does not allow for remote installation, follow the instructions for manual installation from the link in that section. The rest of this article applies to both installation types.

Need To Know

Use the Glassbox Cordova and Ionic Reporting plugin with your Cordova app as a simple alternative to adding the Glassbox JavaScript Detector to each web page in your app.

The plugin uses the JavaScript Detector. Since the JavaScript Detector captures sessions from web pages and a Cordova app is effectively a website running in a native container on a mobile device, the plugin serves as a JavaScript injector to the web pages using the JavaScript Detector. The JavaScript Detector captures the session and the plugin eliminates the need to add the JavaScript Detector to each web page.

Version Requirements

Note the following:

* The Cordova and Ionic Plugin is available from the Mobile SDK version **6.6.62** and later.
* If you are using Glassbox **7**, you must use the Mobile SDK version **6.6.62 SP 1** or later. See [Am I Using Glassbox 7?](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4534709571280033767471858657)

Prerequisites

* The Glassbox app ID, app name, and the URL to which data is reported (Glassbox provides these)
* If you are using Glassbox **7**, a **CUID** (an additional ID) is required. See [Am I Using Glassbox 7?](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4534709571280033767471858657)

Glassbox provides the CUID. If you do not have your **CUID**, contact your Glassbox Professional Services Engineer or Customer Success Manager.

* If you are a Glassbox cloud customer, the URL to the Glassbox JavaScript Detector that Glassbox manages on the remote server is required.
* If you are a Glassbox on-premises or self-hosted customer:
  + The JavaScript Detector must be deployed before the plugin is installed.
  + The URL to that deployed JavaScript Detector is required.
* If your app is an Ionic app, a Glassbox user account with the admin role is needed to configure the WebUI for adoptedStyleSheets to be rendered correctly in Session Replay. See [2. (Ionic Only) Configure for adoptedStyleSheets](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4555347797224033442967831039).
* A Glassbox user account (to test the installation)

## How To

### 1. Install the Plugin

1. If you are using yarn version **2.0** or higher, set the yarn configuration scope for the project by executing the following command:

yarn config set --json npmScopes '{"glassbox":{"npmRegistryServer":"[https://glassbox.jfrog.io/artifactory/api/npm/glassbox-npm/"}](https://glassbox.jfrog.io/artifactory/api/npm/glassbox-npm/%22%7D)}' -H

1. Set the npm configuration for the Cordova and Ionic Plugin by executing the following command:

npm config set @glassbox:registry <https://glassbox.jfrog.io/artifactory/api/npm/glassbox-npm/>

If you are using Windows PowerShell, execute the following command:

npm config set `@glassbox:registry <https://glassbox.jfrog.io/artifactory/api/npm/glassbox-npm/>

1. Install the Cordova and Ionic Plugin by executing the following command:

npm install @glassbox/glassbox-client-recording-plugin

1. In the directory where you installed the plugin, navigate to the @glassbox/client-recording-plugin/www/remote directory directory.
2. Open the gb-injector.js file for editing.
3. In the window.\_gb\_config section, set detectorSrc as shown below.
4. var glassbox = cordova.require('glassbox-client-recording-plugin.glassbox');
5. window.\_gb\_config = {
6. detectorSrc :'[detectorsScriptSrcUrl]',
7. sessionId: -1,
8. visitorId: -1,
9. readyCallback: undefined,
10. };
11. if(window.\_cls\_config === undefined) {
12. window.\_cls\_config = {};

}

Where:

* + [detectorsScriptSrcUrl]:

This is the URL to the Glassbox JavaScript Detector or a JavaScript loader file depending on what type of Glassbox customer you are as explained below.

* + - **Glassbox cloud customers:**

[detectorsScriptSrcUrl] is the URL to the JavaScript Detector that Glassbox manages. Glassbox provides it. See [Prerequisites](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4619471786108833043433579389).

* + - **Glassbox on-premises or self-hosted customers:**

Depending on how you have chosen to deploy your JavaScript Detector, [detectorsScriptSrcUrl> is either the URL to the JavaScript Detector on your host server or the URL to the JavaScript loader file on your host server.

**Note:** A JavaScript loader file loads the JavaScript Detector file.

### 2. (Ionic Only) Configure for adoptedStyleSheets

If your app is an Ionic app, you need to configure the JavaScript Detector and Session Replay in the Glassbox Console for adoptedStyleSheets functionality, as explained below. This is required to record the session and render it in Session Replay correctly.

How you configure for adoptedStyleSheets depends on whether you are a Glassbox cloud customer, or a Glassbox on-premises or self-hosted customer.

**Select the relevant tab and follow the instructions:**

1. The JavaScript Detector must be configured for adoptedStyleSheets to be recorded correctly. Contact your Glassbox Professional Services Engineer or Customer Experience Manager to configure the JavaScript Detector for adoptedStyleSheets.
2. The WebUI must be configured for adoptedStyleSheets to be rendered correctly in Session Replay.

Configure the WebUI by doing the following:

* 1. Open **Administration Tools (webuitools)**.
  2. Select the **Webui Configuration** tab.
  3. Add the line shown below to the webReplayadvanced section.

"supportAdoptedStyles": true

For example:

"webReplay": {

"advanced": {

"allowInteractiveMode": true,

"clickmapCrossSessionVisualNames": true,

"domBasedContentLoading": true,

"supportAdoptedStyles": true,

.

.

.

},

.

.

.

* 1. Click **SAVE**.

### 3. Initialize Within the App

In the deviceready event, add the following line:

var glassbox = cordova.require('glassbox-client-recording-plugin.glassbox');

For example:

document.addEventListener('deviceready', onDeviceReady, false);

function onDeviceReady() {

// Cordova is now initialized.

var glassbox = cordova.require(‘glassbox-client-recording-plugin.glassbox’);

.

.

.

}

### 4. Configure the Plugin for a JavaScript Detector on a Remote Server

How you configure the Cordova Plugin for the JavaScript Detector on a remote server depends on the type of Glassbox customer you are.

**Select the relevant tab and follow the instructions:**

If you are a Glassbox cloud customer, Glassbox will configure the JavaScript Detector on the Glassbox cloud to work with the plugin.

If you want non-default configuration parameters set, contact your Glassbox Professional Services Engineer or Customer Experience Manager.

### 5. Add the Plugin

Add the Cordova and Ionic Plugin to the project by executing the following command:

cordova plugin add @glassbox/glassbox-client-recording-plugin

### 6. Start the Mobile SDK

This section explains how to start the Mobile SDK using the Cordova and Ionic Plugin with its default behaviors in native and hybrid apps.

**Important**: If after starting your app it runs successfully but no recorded session is created in Glassbox, see [Why Don't I See My Mobile Session in Glassbox?](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm73372279787076)

**Note**: Advanced start configurations are presented in [Step 5: Apply Advanced Start Configurations](https://support.glassbox.com/s/article/20282420446866).

**Select the tab that applies to your app and follow the instructions:**

To start the Mobile SDK in a Cordova native app, call the Mobile SDK glassbox.startSession API function as explained below.

**glassbox.startSession API Function**

startSession: function(onSuccess, onError, url, appId, cuid)

**glassbox.startSession API Function Parameters**

| **Parameter** | **Description** | **Mandatory or Optional** |
| --- | --- | --- |
| appid | The application ID. Glassbox provides this. | Mandatory |
| url | The URL to which data is reported. Glassbox provides this. | Mandatory |
| cuid | The ID used to start the Mobile SDK for Glassbox **7**, see [Prerequisites](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4619471786108833043433579389). | Mandatory for new applications defined in Glassbox **7**  Not required for applications that were defined before upgrading to Glassbox 7  Not relevant for versions older than Glassbox 7 |

For example:

glassbox.startSession(

function() {

// Your code goes here

},

function(e) {

// Your code goes here

},

<YOUR\_GLASSBOX\_REPORTING\_URL>,

<YOUR\_GLASSBOX\_APPLICATION\_ID>,

<YOUR\_GLASSBOX\_CUID>);

Where:

* <YOUR\_GLASSBOX\_REPORTING\_URL>: The reporting URL provided by Glassbox
* <YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox
* <YOUR\_GLASSBOX\_CUID>: The CUID provided by Glassbox and is only required for Glassbox Application version **7**.

**Important**: If you are not using Glassbox **7**, omit the <YOUR\_GLASSBOX\_CUID> parameter.

In hybrid apps, the Mobile SDK captures the native content and the JavaScript Detector captures the web content. The sessions recorded by each of the Mobile SDK and JavaScript Detector will be consolidated (sessionized) into one seamless session.

By default, the Mobile SDK does not inject the JavaScript Detector to capture the web content. Injecting the JavaScript Detector requires that an additional Mobile SDK parameter is specified. The instructions for that hybrid parameter are shown in the article.

**Warning**: For Cordova apps that contain native screens, start the Mobile SDK from the app. Do not use the following Cordova and Ionic Plugin API functions:

* startSession
* stopSession
* restartSession

The other available Cordova and Ionic Plugin API functions can be used.

**Note**: The JavaScript Detector supports many configuration options including the masking of hybrid app web content. For information about the JavaScript Detector configuration. See [JavaScript Detector Configuration](https://support.glassbox.com/s/article/JavaScript-Detector-Configuration).

#### **For Older Cordova Versions**

If you are not using the current Cordova version (for example, you are using Cordova **5.1.1**), one or more of the instructions in the following steps may be required:

1. If WKWebview support is not enabled, enable it by setting WK\_WEB\_VIEW\_ONLY to 1 in both CordovaLib and the App Project Build Settings.
2. If cordova-plugin-wkwebview-engine is not installed, install it by executing the following commands:
3. npm install cordova-plugin-wkwebview-engine

cordova plugin add --link node\_modules/cordova-plugin-wkwebview-engine

1. For Android only, if you are using a Cordova version earlier than 10.0.0:
   1. Add the whitelist plugin to the app project to enable the JavaScript Detector to report events. Execute the following command:

cordova plugin add <https://github.com/apache/cordova-plugin-whitelist.git>

* 1. Add the following to the config.xml file:

**Important:** The URL must end with /\*.

<allow-navigation href="[<YOUR\_GLASSBOX\_REPORTING\_URL>]/\*" />

Where:

* + - <YOUR\_GLASSBOX\_REPORTING\_URL>: The reporting URL provided by Glassbox

 Start the Mobile SDK in Hybrid Mode

Instructions about how to start the Mobile SDK in Android apps and iOS apps are shown in the sections below.

Android

void .hybridMode()

**To start the Mobile SDK in a Cordova hybrid app on Android:**

1. If you do not have one already, add a class that extends the  Application  class. Override its onCreate function and add the following code to start capturing:
2. try {
3. Glassbox.start(aSettingsBuilder()
4. .withApplicationCtx(this)
5. .hybridMode()
6. .build())
7. } catch (GlassboxRecordingException e) {
8. e.printStackTrace();

}

1. In the Android application project's AndroidManifest.xml file, add the following inside the application tag:

android:name="[ClassName]"

Where:

* + [ClassName]: The name of the class you created or used in the previous step

1. Add the following service tag to the AndroidManifest.xml file:
2. <service android:name="com.clarisite.mobile.GlassboxJobX"
3. android:exported="false"
4. android:permission="android.permission.BIND\_JOB\_SERVICE">
5. <meta-data android:name="url" android:value="<YOUR\_GLASSBOX\_REPORTING\_URL>" />
6. <meta-data android:name="appid" android:value="<YOUR\_GLASSBOX\_APPLICATION\_ID>" />

</service>

Where:

* + <YOUR\_GLASSBOX\_REPORTING\_URL>: The reporting URL provided by Glassbox
  + <YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox

1. In the class that extends CordovaActivity (which is usually named MainActivity). update the onCreate method as follows:

**Important:** The init method must be called first, then trackView, and finally loadUrl. This order is mandatory.

public class MainActivity extends CordovaActivity

{

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

init();

try {

Glassbox.trackView(appView.getView());

} catch (GlassboxRecordingException e) {

e.printStackTrace();

}

// Set by in config.xml

loadUrl(launchUrl);

}

}

 iOS

settings.hybridMode = true

To start the Mobile SDK in a Cordova hybrid app on iOS:

* Call the Glassbox iOS SDK start API function in the AppDelegate class, in application:didFinishLaunchingWithOptions:, with hybridMode set to true.

For example:

Swift

let settings = GLAStartupSettings()

settings.appId = "<YOUR\_GLASSBOX\_APPLICATION\_ID>"

settings.reportUrl = "<YOUR\_GLASSBOX\_REPORTING\_URL>"

settings.hybridMode = true

try Storyboard.start(settings)

Where:

* <YOUR\_GLASSBOX\_REPORTING\_URL>: The reporting URL provided by Glassbox
* <YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox

Objective-C

GLAStartupSettings \*setting = [[GLAStartupSettings alloc] init];

setting.appId = @"<YOUR\_GLASSBOX\_APPLICATION\_ID>";

setting.reportUrl = @"<YOUR\_GLASSBOX\_REPORTING\_URL>";

setting.hybridMode = YES;

NSError \*error;

[Storyboard start:url appId:appid error:&error];

Where:

* <YOUR\_GLASSBOX\_REPORTING\_URL>: The reporting URL provided by Glassbox
* <YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox

For Sessionizing Hybrid Apps on Glassbox **6**

If you are using Glassbox 6, add the cookie and query string sessionizing, see [Am I Using Glassbox 7?](https://support.glassbox.com/s/article/5832981248401#UUID-6ee69a9f-fe61-06ab-1717-a79d9b313431_section-idm4534709571280033767471858657)

**To sessionize hybrid apps on Glassbox 6:**

Add cookie and query string sessionizing:

<cfgScope scope="[<YOUR\_GLASSBOX\_APPLICATION\_ID>]">

<cfgUseIP on="false"/>

<cfgSessionID key="\_CLS\_S" overrideIP="true" scope="global" priority="10" overrideUA="true">

<lookin>

<location location="param\_and\_cookie" parameter="\_CLS\_S"

caseSensitive="false" />

</lookin>

</cfgSessionID>

</cfgScope>

Where:

* <YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox

Set Up the Analyzer

If you are a Glassbox cloud customer, the Analyzer is set up for you by Glassbox. You do not need to make any changes.

If you are a Glassbox on-premises or self-hosted customer, set up the Analyzer by adding the lines below to the sessionizing file clingine/conf/http/sessionizing.xml.

<cfgScope scope="[<YOUR\_GLASSBOX\_APPLICATION\_ID>]">

<cfgUseIP on="false"/>

<cfgSessionID key="\_CLS\_S" overrideIP="true" scope="global" priority="10" overrideUA="true">

<lookin>

<location location="param\_and\_cookie" parameter="\_CLS\_S"

caseSensitive="false" />

</lookin>

</cfgSessionID>

</cfgScope>

Where:

<YOUR\_GLASSBOX\_APPLICATION\_ID>: The application ID provided by Glassbox.

The ID is shown in the Console on the Applications screen and the app's settings page.

### A screenshot of a computer Description automatically generated 7. Test the Mobile SDK

The suggested procedure for testing the Mobile SDK is provided below.

To complete the procedure, you will need a Glassbox user account. If you do not have one, collaborate with someone in your organization who has an account and is familiar with the Session Replay feature.

**Important**: The Mobile SDK supports callbacks that can help you solve issues that might occur while testing your installation. These callbacks can log useful behind-the-scenes information on session status. You can share that information with Glassbox when needed. See [How to Report Session Status for Debugging](https://support.glassbox.com/s/article/6091371275409#UUID-89eb1c10-56bd-87d8-25b1-3b5173c32e99).

**To test the Mobile SDK:**

1. Open the app that is running the Mobile SDK and perform several activities that demonstrate some or all of the following on one or more screens:
   * Open more than one screen.
   * Tap, scroll, and select from the menus.
   * Enter information in input fields.
   * Perform actions that result in AJAX requests and responses.
2. In the Glassbox Console, click **Recorded Sessions** in the menu bar on the left.
3. In the Time Frame Selection window, select **Active Sessions** (if you completed the session within the last few minutes) or **Past Hour** (if you completed the session more than a few minutes ago).
4. Locate your test session in the **Recorded Sessions** list. Alternatively, search in either **Quick Filters** or **Show Me**.
5. Open the session in **Session Replay** by clicking Play  on the right. The recorded session is shown. The activities you performed are listed in the Activity Tree on the left and the Session Replay is shown on the right.
6. Verify that your recorded session contains the information captured from the device by doing the following:
   * Play the session by clicking Play  on the replay control bar.
   * Examine the activities in the Activity Tree by expanding an item. For example, view the clicks and scrolls.
   * Examine the activities in more detail by clicking **EXPERT VIEW**. For example, view the document and page loads, AJAX Requests, and mouse movements.