

## HCL Discover Flutter Plugin Integration Guide



## Table of Contents

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Discover Flutter Plugin with Native iOS and Android SDKs.....              | 3  |
| Environments for Flutter Plugin Dev tools .....                            | 3  |
| Discover Flutter Plugin Structure.....                                     | 3  |
| Accessing the Discover Flutter Plugin.....                                 | 3  |
| Integrating and Enabling the Discover Flutter plugin in a Flutter App..... | 4  |
| Step 1 : Add the Discover plugin in the "pubspec.yaml" of your app .....   | 4  |
| Step 2 : Flutter Plugin Usage in Flutter App .....                         | 4  |
| Step 3 : Enable Discover Framework in the main() method.....               | 4  |
| Step 4: Configuring POST message URL .....                                 | 5  |
| Configuring iOS and Android SDKs .....                                     | 5  |
| Android .....                                                              | 5  |
| iOS.....                                                                   | 5  |
| Common Use-Cases for Native SDK Integration via Plugins.....               | 5  |
| Methods and its payload .....                                              | 5  |
| Type 1 - Client State Messages .....                                       | 5  |
| Type 2 - ScreenView messages .....                                         | 6  |
| Type 3 - Connections messages .....                                        | 6  |
| Type 4 - Control messages.....                                             | 7  |
| Type 5 - Custom event messages .....                                       | 7  |
| Type 6 - Exception messages .....                                          | 8  |
| Type 10 - Layout messages .....                                            | 8  |
| Type 11 - Gesture messages.....                                            | 9  |
| Type 13 - Geolocation messages.....                                        | 9  |
| Type 15 - Form completion.....                                             | 10 |
| Troubleshooting Steps.....                                                 | 10 |
| FAQ.....                                                                   | 11 |

## Discover Flutter Plugin with Native iOS and Android SDKs

Discover Flutter plugin serves as a bridge between Dart code and native platform code (iOS and Android). It allows developers to access platform-specific features using native SDKs while writing most of their applications in Dart. This document details how Discover Flutter plugin is integrated into Flutter applications using native iOS and Android SDKs.

### Environments for Flutter Plugin Dev tools

- Android Studio Ladybug | 2024.2.1 and above version
- Xcode Version 16.4 and above version
- Supports Dart language 3.7 and above

### Discover Flutter Plugin Structure

Discover Flutter plugin has the following structure:

```
discover_flutter_plugin/
|
|__ lib/
|   |__ discover_flutter_plugin.dart      # Dart-facing API
|
|__ android/
|   |__ src/main/Kotlin/...              # Kotlin code for Android
|
|__ ios/
|   |__ classes/...                      # Swift code for iOS
|
|__ test/                                # Optional: unit tests for plugin logic
|
|__ example/                             # Example app demonstrating usage
|
|__ pubspec.yaml                         # Package metadata
|
|__ README.md                           # Documentation
|
|__ LICENSE                             # License file
|
|__ CHANGELOG.md                        # Version history
```

### Accessing the Discover Flutter Plugin

The plugin will be made available as a download from HCL Software Licensed site for customers.

# Integrating and Enabling the Discover Flutter plugin in a Flutter App

## Step 1 : Add the Discover plugin in the “pubspec.yaml” of your app

```
pubspec.yaml dependencies:  
  discover_flutter_plugin:  
    path: <discover_plugin_path>/discover_flutter_plugin
```

## Step 2 : Flutter Plugin Usage in Flutter App

Add the Import Statement below in the Widget file where the Plugin code must be accessed

```
import package:discover_flutter_plugin/discover_flutter_plugin.dart
```

## Step 3 : Enable Discover Framework in the main() method

```
void main() async  
{  
  ....  
  DiscoverFlutterPlugin().enableDiscoverFramework();  
  ....  
}
```

Enables the Discover Framework when the App is Launched with bool value type with true or false as return type; generates a Session id when app launches and this session id will be post back to server to track the event types with active sessions.

Sample payload generated in the console if successful:

```
{  
  "messageVersion": "12.0.0.1",  
  "serialNumber"  
: 1,  
  "sessions": [  
    {  
      "EOCoreSessionID":  
"77E753BE17F6485D8D078C7ADC55D51B", "id":  
"F160550A87B64C05999644B1242094D4",  
      "startTime": 1753093309418,  
    }  
  ]  
}
```

Developers do not access “android/” or “ios/” folders directly; the plugin system integrates native code during the build automatically.

## Step 4: Configuring POST message URL

The POST message URL should be configured in both the platforms under respective file:

- **Android:** `DiscoverBasicConfig.properties`
- **iOS:** `DiscoverBasicConfig.plist`

## Configuring iOS and Android SDKs

Config files Location path in flutter project

### Android

`flutter_app/android/app/src/main/assets`

[https://help.hcl-software.com/HCLDiscover/12.1.13/en/SDK/MobileSDKConfigurations/configurations\\_for\\_android.html](https://help.hcl-software.com/HCLDiscover/12.1.13/en/SDK/MobileSDKConfigurations/configurations_for_android.html)

### iOS

`/flutter_app/io`

[https://help.hcl-software.com/HCLDiscover/12.1.13/en/SDK/MobileSDKConfigurations/configurations\\_for\\_ios.html](https://help.hcl-software.com/HCLDiscover/12.1.13/en/SDK/MobileSDKConfigurations/configurations_for_ios.html)

## Common Use-Cases for Native SDK Integration via Plugins

- User interaction tracking (text changes, print screen detection)
- Logging and analytics (e.g., DCF, Firebase)
- Location tracking (CoreLocation/Geolocation)
- Network monitoring (e.g., logConnection)
- Exception handling (native error reporting)

## Methods and its payload

### Type 1 - Client State Messages

The Client state messages provide information about the client. They can be delivered on a schedule or when changes occur to the environment state on the client.

**Method Name:** `DiscoverFlutterPlugin().updateOrientation ()`;

**Sample Payload:**

```
{
  "fromWeb":
  false, "type":
  1,
  "offset": 71376,
  "screenviewOffset": 60,
  "count": 7,
  "mobileState":
  {
    "freeStorage": 1347649536,
```

```

        "freeMemory": 1861832704,
        "battery": 100,
        "ip": "10.0.2.16",
        "carrier": "T-Mobile",
        "orientation": 0,
        "pageOrientation": 0,
        "connectionType": "WIFI",
        "networkReachability":
        "ReachableViaWIFI", "androidState": {
            "keyboardState": "HIDDEN_FALSE"
        }
    }
}

```

## Type 2 - ScreenView messages

ScreenView messages show steps of Mobile screen changes in a mobile application, or steps in a business process.

**Method Name:** Auto Call from Native SDKs.

**Sample Payload:**

```

{
    "fromWeb":
    false, "type":
    2,
    "offset": 71316,
    "screenviewOffset": 0,
    "count": 6,
    "screenview":
    {
        "type": "LOAD",
        "name": "MyScreenViewName"
    }
}

```

## Type 3 - Connections messages

Connection messages provide information about how the client application manages requests or responses.

**Method Name:** `DiscoverFlutterPlugin().logConnectionToNative ()`;

**Sample Payload:**

```

{
    "fromWeb":
    false, "type":
    3,
    "offset": 166720,
    "screenviewOffset": 0,
    "count": -1,
}

```

```

        "connection":
        {
            "url": "https:\\\\www.apple.com\\/",
            "statusCode": 0,
            "responseDataSize": 0,
            "initTime": 166720,
            "responseTime": 0,
            "loadTime": 166720
        }
    }
}

```

## Type 4 - Control messages

Control messages are used to log user actions and behaviour. They consist of a control identifier and the value returned by that control.

**Note:** Type 4 event works in iOS platform only.

**Method Name:** `DiscoverFlutterPlugin().logTextChangeEvent ();`

**Sample Payload:**

```

"tlType": "label",
"idType": -4,
"position": {
    "y": 0,
    "x": 0,
    "width": 0,
    "height": 0
},
"style": {
    "textBGAlphaColor":
    0, "hidden": false,
    "textAlphaColor": 1,
    "bgColor": 0,
    "alphaBGColor": 0,
    "textColor": 0,
    "textBGColor": 0
},
"currState": {

```

## Type 5 - Custom event messages

The Custom event messages are used to custom log any event from any place in the application. They log client data that are application specific.

**Method Name:** `DiscoverFlutterPlugin().logCustomDiscoverEvent ();`

**Sample Payload:**

```


```

```
{
  "fromWeb":
  false, "type":
  5,
  "offset": 4747,
  "screenviewOffset": 0,
  "count": 2,
  "customEvent":
  {
    "name": "CUSTOM MSG - Person Details"
  }
},
```

## Type 6 - Exception messages

The exception messages record the name and description of the exception that occurs.

**Method Name:** `DiscoverFlutterPlugin().logExceptionToNative ()`;

**Sample Payload:**

```
{
  {
    "fromWeb": false,
    "type": 6,
    "offset": 4485,
    "screenviewOffset": 0,
    "count": -1,
    "exception": {
      "name": "class java.lang.Exception",
      "description": "Dart Exception\nRangeError (length): Invalid value:
Not in inclusive range 0..2: 5\n#0      List.[]
(dart:core- patch\growable_array.dart)\n#1
_PersonalScreenState.triggerErrorExceptionLog
(package:discover_flutter_plugin_usage_example\creens\personal_screen.da
rt:7 8:17)\n#2 _PersonalScreenState.logCustomDiscoverEvents
(package:discover_flutter_plugin_usage_example\creens\personal_screen.da
rt:5 5:5)\n#3 _PersonalScreenState.build
(package:discover_flutter_plugin_usage_example\creens\personal_screen.da
rt:1 48:5)\n#4 StatefulElement.build
(package:flutter\src\widgets\framework.dart:5743:27)\n#5
ComponentElement.performRebuild
```

## Type 10 - Layout messages

Layout messages show the current display layout of a native page with base64 image.

**Method Name:** `DiscoverFlutterPlugin().logNativePrintScreenEvent ()`;

**Sample Payload:**

A Geolocation message logs a user's location information. The message consists of a control identifier and a Geolocation value.

If the user permits to use location data, Geolocation returns the latitude, longitude, and accuracy values. If the user does not permit to use location data, Geolocation returns an error code and error string.

**Method Name:** `DiscoverFlutterPlugin().logLocationToNative();`

**Sample Payload:**

```
{
  "fromWeb":
  false, "type":
  13,
  "offset": 1212,
  "screenviewOffset": 0,
  "count": -1,
  "geolocation":
  {
    "lat": 37.4220936,
    "long": -122.083922,
    "accuracy": 600
  }
}
```

## Type 15 - Form completion

Returns a bool value on the successful submission of a particular form in a screen.

**Method Name:** `DiscoverFlutterPlugin().logFormCompletionToNative ();`

**Sample Payload:**

```
{
  "fromWeb":
  false, "type":
  15,
  "offset": 166209,
  "screenviewOffset": 0,
  "count": -1,
  "formCompleti
n": {
    "submitted":
    false,    "valid":
    false
  }
}
```

## Troubleshooting Steps

**Note:** These are common Flutter troubleshooting steps and not specific to Discover Plugin.

1. flutter clean-- used to delete build artifacts and reset your Flutter project to a clean state. It helps resolve many common issues during development.
2. flutter pub get -- is used to fetch and install all the Dart and Flutter package dependencies listed in your project's pubspec.yaml file.

3. Pod Update – Updates the pod file used in iOS projects to update Cocoa Pods dependencies.

## FAQ

### Question

When and what type of events are auto triggered for Android and iOS?

### Answer

The following types are auto triggered on both Android and iOS platforms:

- Type 1
- Type 2
- Type 10 with Base64
- Type 13

The following types must be manually triggered on both Android and iOS platforms:

- Type 3
- Type 4 (supported only on iOS for the current plugin)
- Type 5
- Type 6
- Type 11
- Type 15

### Question

What data is auto captured in these events for Android and iOS?

### Answer

Refer the provided Sample Payload for each Type. It contains the example for data collected for that Type. This collected data is sent back to HCL Discover.

### Question

What data is auto captured in these events for Android and iOS?

### Answer

Refer the provided Sample Payload for each Type. It contains the example for data collected for that Type. This collected data is sent back to HCL Discover.