Error Signals | Developer Documentation

Error Signals

Updated: Feb 6, 2026

Upcoming deprecation: Starting **April 15, 2026**, the `PendingIntent`-based handshake method for authentication templates will be deprecated. If you are currently using `PendingIntent` to initiate handshakes or verify app identity, the OTP Android SDK is the preferred way to migrate.

The OTP Android SDK features a simplified workflow for implementing one-tap and zero-tap authentication templates. You can learn how to use it below.

This document describes Android-only error signals that can help you debug one-tap autofill authentication templates and zero-tap authentication templates.

If your message fails the eligibility check, the one-tap autofill button will be replaced with a copy code button. In addition, there may be device or WhatsApp client settings that prevent message notifications. To help with debugging, our apps surface some error information via the `com.whatsapp.OTP_ERROR` intent. In these situations you will receive an error key and message instead of the user's one-time passwords or verification code.

Note that some of these error signals will only surface if you are running WhatsApp in the Android emulator.

Key	Description
`ambiguous_delivery_destination` *Emulator only*	**Ambiguous delivery destination** There are multiple active OTP requests for the packages specified by this template, and we could not determine which package to deliver the code to. This can happen when multiple applications specified in the template's `supported_apps` array have initiated the handshake (sent the `com.whatsapp.otp.OTP_REQUESTED` intent) within the past 10 minutes.
`incompatible_os_version`	**Incompatible Android version** This can happen when you initiate the handshake (send the `com.whatsapp.otp.OTP_REQUESTED` intent) but the device is running a version of Android older than v19.

Key	Description
`incorrect_signature_hash` *Emulator only*	**Incorrect signature hash** This can happen when you initiate the handshake (send the `com.whatsapp.otp.OTP_REQUESTED` intent) and our app receives an authentication template message that uses a one-tap autofill button, but the package name in the message does not produce the message's signature hash.
`missing_handshake_or_disorder`	**Missing handshake / Order of operations** This can happen when our app receives an authentication template message with a one-tap autofill button but the handshake was not initiated.
`otp_request_expired`	**OTP request expired** This can happen when an authentication template that uses a one-tap autofill button is delivered to the user but more than 10 minutes (or the number of minutes indicated in the template's `code_expiration_minutes` property, if present) have passed since you initiated the handshake. In this situation, we display the copy code button instead.
`whatsapp_message_notification_disabled` *Emulator only*	**Message notification disabled in WA settings** This can happen when you initiate the handshake (send the `com.whatsapp.otp.OTP_REQUESTED` intent) but the user has disabled notifications in the WhatsApp app or WhatsApp Business app (within our app settings).
`whatsapp_notification_disabled` *Emulator only*	**WA notification disabled in device level** This can happen when you initiate the handshake (send the `com.whatsapp.otp.OTP_REQUESTED` intent) but the user has disabled app notifications for our apps (device level settings).

Integration

The error signals are delivered via broadcasted intent so you must implement `BroadcastReceiver` to listen for error signals.

In manifest.xml

```
<receiver
 android:name=".app.otp.OtpErrorReceiver"
 android:enabled="true"
 android:exported="true" >
   <intent-filter>
       <action android:name="com.whatsapp.otp.OTP_ERROR"/>
   </intent-filter>
</receiver>
```

Receiver class - Using the SDK (Preferred)

Implement `onReceive` and use a `WhatsAppOtpIncomingIntentHandler` object to process the debug signals.

```
publicclassOtpErrorReceiverextendsBroadcastReceiver{@Overridepublicvoid onReceive(Context context,Intent intent){WhatsAppOtpIncomingIntentHandler whatsAppOtpIncomingIntentHandler =new WhatsAppOtpIncomingIntentHandler();
     whatsAppOtpIncomingIntentHandler.processOtpDebugSignals(
                          intent,// your function to handle the signal(debugSignal)-> handleSignal(debugSignal),// your function to handle any error(error, exception)-> handleError(error, exception));}}
```

Receiver class - Without the SDK

```
publicclassOtpErrorReceiverextendsBroadcastReceiver{publicstaticfinalString OTP_ERROR_KEY = "error";publicstaticfinalString OTP_ERROR_MESSAGE_KEY ="error_message";@Overridepublicvoid onReceive(Context context,Intent intent){String otpErrorKey = intent.getStringExtra(OTP_ERROR_KEY);String otpErrorMessage = intent.getStringExtra(OTP_ERROR_MESSAGE_KEY);// Handle errors}}
```

Revision #3
Created 2026-04-01 14:29:11 UTC by New Admin
Updated 2026-04-06 17:49:42 UTC by New Admin