

Grandstream Networks, Inc.

GDMS Networking

Alert Notifications to Third-Party Platforms Guide



Overview

This guide provides step-by-step instructions for integrating GDMS Networking with external messaging platforms: **Telegram®**, **WhatsApp®** **Cloud API**, and **Grandstream Wave** to receive real-time system alerts outside the web interface.

By connecting GDMS to messaging apps and clients your team already uses, you can:

- Get instant notifications for critical events (e.g., device offline, network disruptions).
- Centralize alerts in group chats, mobile apps, or the Wave client for faster response.
- Extend alert visibility beyond email or dashboard-only notifications.

The guide walks you through:

- **Prerequisites:** What accounts, tokens, and IDs you need in advance.
- **Telegram Setup:** How to create and activate a bot, and link it to GDMS.
- **WhatsApp Setup:** How to use Meta's Cloud API with a business number and approved message template.
- **Wave Setup:** How to use a Wave interface link and UCM extension to receive alerts in the Wave client.
- **Alert Configuration:** How to select which alerts should be sent to each platform.

Whether you're managing one site or multiple networks, this integration ensures the right people are notified at the right time wherever they are.

Prerequisites

Before configuring alerts in GDMS Networking, make sure you have the following:

- **GDMS Networking account:** Admin access to your instance (for example, gdms.cloud or an on-prem GWN Manager deployment).
- **Telegram account:** Required to create a bot via BotFather.
 - More information: <https://telegram.org>
- **WhatsApp-capable phone number:**
 - A number you can use as a sender through the WhatsApp Cloud API.
 - It must support WhatsApp and follow E.164 format (e.g., **+14155550123**).
- **Meta developer setup for WhatsApp**, including:
 - A Meta Developer account and a Meta App with the WhatsApp product added
 - WhatsApp Cloud API access
 - Full instructions: <https://developers.facebook.com/docs/whatsapp/cloud-api/get-started>
 - **Business verification (if required):** Some features of the WhatsApp® Business Platform, such as sending alerts via Cloud API, may require your Meta business account to complete verification. Refer to the Meta Help Center for up-to-date requirements.
- **Grandstream UCM / Wave environment**, including:
 - A UCM system with Wave support enabled
 - A Wave user/extension that can sign in to the Wave desktop client
 - Permission to access the **Management Portal** → **Integration** page in Wave in order to create an **Interface Link URL**, **Token**, and **Wave extension number** for alerts.

Note:

This guide assumes you already have all required accounts, credentials, and tokens. For platform-specific provisioning steps (e.g., Meta WhatsApp Cloud API configuration or UCM/Wave deployment), refer to the official vendor documentation, as policies and user interfaces may change over time.

Telegram Setup

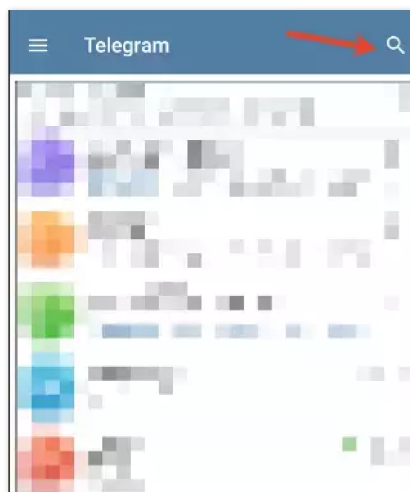
To receive alert notifications in Telegram, you'll need to create a Telegram bot using **BotFather**, activate it by sending a message, and then connect it to your GDMS Networking account using the bot's API token. Once configured, GDMS will be able to send alert messages directly into your Telegram chat.

Note: For more information on Telegram bots, you can refer to the official guide from Telegram here:

<https://core.telegram.org/bots/tutorial>

Search for BotFather in Telegram

Open the Telegram app and tap the search icon (top right corner). Type **BotFather** into the search bar. You'll see the verified BotFather account appear in the results.



Search on Telegram



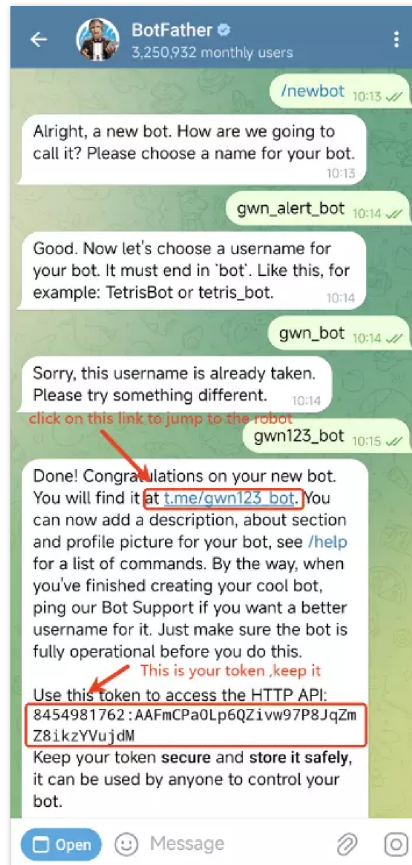
BotFather

Create a New Bot

Once inside the BotFather chat, tap **Start** (if needed), then type `/newbot` and follow the prompts:

1. **Choose a bot name:** This can be anything, e.g. `GWN Alert Bot`
2. **Choose a username:** Must be unique and must end in `bot`, e.g. `gwn_alert_bot`

If the username is taken, BotFather will prompt you to try another. Once successful, BotFather will respond with a **link to your bot** (e.g. `t.me/gwn123_bot`) and a **bot token**.



Create a New Bot

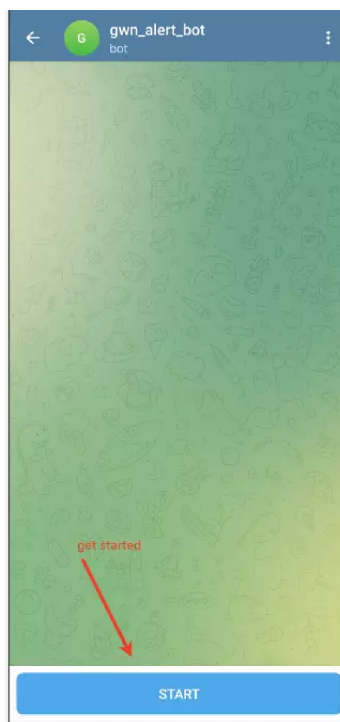
Note:

Important: Copy and securely store the bot token. You'll need this token to integrate with GDMS. Example token:

8454981762:AAFmCPa0Lp6QZivw97P8JqZmZ8ikzYVujdM This token is what GDMS uses to authenticate and send messages via your bot.

Start the Bot

Click the link provided by BotFather (e.g. `t.me/gwn123_bot`) to open your bot's chat window. Press **Start** to initiate the bot.



Start the Bot



Initiate the conversation

Note:

If you skip this step, messages from GDMS will not be delivered. Telegram bots cannot initiate conversations users must start them first.

Link the Bot to GDMS

1. Log in to your GDMS Networking or GWN Manager instance
2. Navigate to: **Alerts** → **Alert Notification**.
3. From the **3rd-Party Platforms** dropdown, select **Telegram**

4. Paste the **bot token** you copied earlier into the **Token** field

5. Under the alert categories (System Alert, Performance Alert, etc.), toggle **3rd-Party Platform** for the alerts you want to forward to Telegram.

Alerts > Alert Notification

Each account can independently set alerts they want to receive and the email address to receive them.

Email Address: +

3rd-Party Platforms: +

* Token: 1-512 characters

System Alert Performance Alert Security Alert Network Alert

Alert Detail	Notification Type
Manager	
AP offline for more than 30 mins	Web ☐ Email ☐ App ☐ 3rd-Party Platform ☒
AP online (After offline for 30 mins)	Web ☐ Email ☐ App ☐ 3rd-Party Platform ☒
Router offline for more than 30 mins	Web ☐ Email ☐ App ☐ 3rd-Party Platform ☒
Router online (After offline for 30 mins)	Web ☐ Email ☐ App ☐ 3rd-Party Platform ☒

Link the Bot to GDMS

Note:

Example: Toggle on alerts like “AP offline for more than 30 mins” or “Router offline for more than 30 mins” under System Alert to push those events to Telegram.

WhatsApp Setup

GDMS Networking supports WhatsApp Cloud API integration for delivering alert notifications to a specified mobile number using a pre-approved message template. This setup relies on Meta’s developer platform, where you’ll create a template, obtain a long-lived access token, and retrieve your phone number ID. Once these credentials are entered in GDMS, the system can deliver real-time alerts directly to your WhatsApp chat, using dynamic message content defined by your alert preferences.

Notes:

- To ensure successful delivery, the message template must follow strict format requirements, and the WhatsApp Business account must be properly configured on Meta’s platform.
- The following steps reflect the current WhatsApp Cloud API setup process as of this writing. However, Meta may update the platform UI, token policies, or configuration flow at any time.

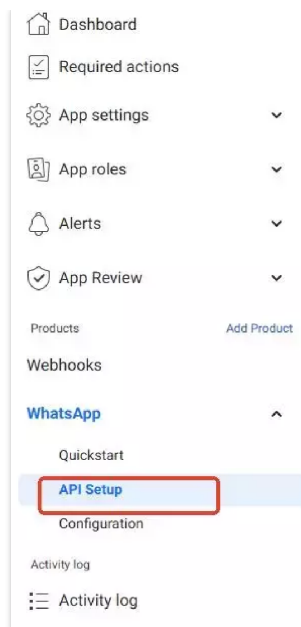
We strongly recommend reviewing the official documentation if anything has changed or does not align with your interface:

<https://developers.facebook.com/docs/whatsapp/cloud-api>

Access the WhatsApp API Setup in Meta Developer Console

Log in to your [Meta for Developers](#) account and open the app you created for WhatsApp integration. In the left-hand navigation menu, locate the **WhatsApp** product section.

Click on **API Setup** under the “Quickstart” section. This will take you to the WhatsApp Cloud API configuration area, where you’ll generate the credentials and test messages used for alert delivery.



WhatsApp API Setup

Note:

If the WhatsApp product isn't listed yet, make sure you've added it to your app by selecting **Add Product** and choosing WhatsApp from the available options.

Generate Access Token and Collect Required Credentials

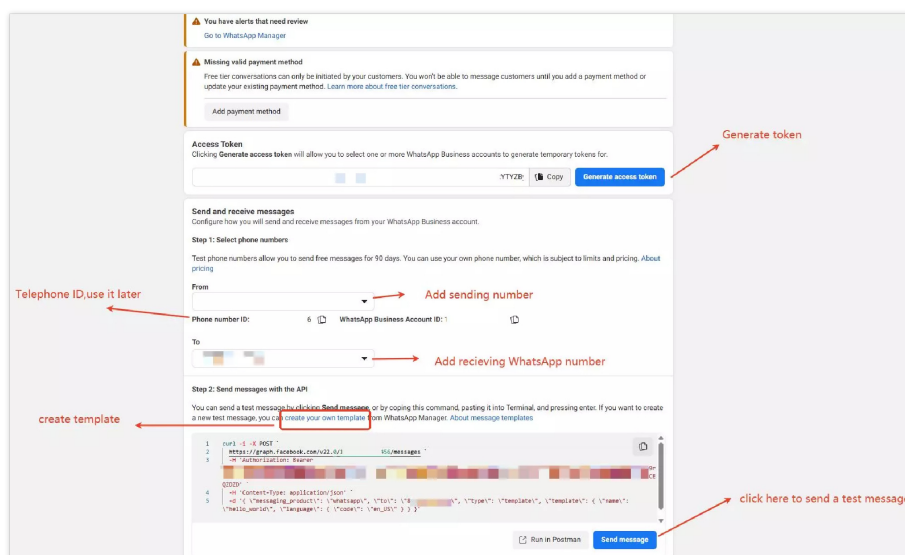
On the **API Setup** page, scroll to the **Access Token** section and click **Generate access token**. This token is used to authenticate API requests sent from GDMS to the WhatsApp Cloud API.

Once generated, copy and store the token securely, as it will be required when configuring WhatsApp alerts in GDMS.

Next, scroll to the **Send and receive messages** section:

- In the **From** dropdown, select your **WhatsApp Business sender number** (provisioned through Meta).
- In the **To** dropdown, select a **recipient number**. This should be an active WhatsApp number used for message testing.

Just below, locate the **Phone number ID** (also called *Telephone ID*). This is a unique identifier for the sender number and will be needed in the GDMS alert configuration later.



Generate Access Token and Collect Required Credentials

You will also find a sample `curl` command under **Step 2** that can be used to send a test message through the WhatsApp Cloud API. This example demonstrates how to structure an API request using a message template and your access credentials.

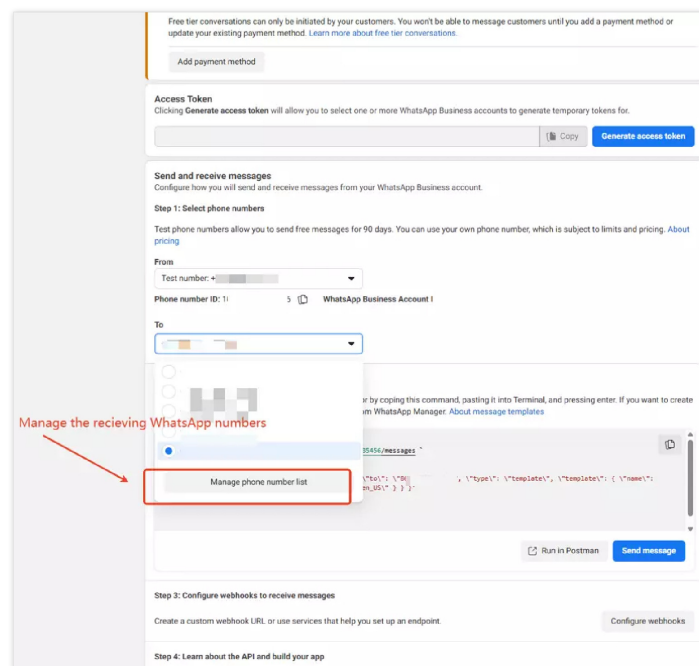
Note:

You may skip this test at this stage and proceed with template creation first. A valid template is required before test messages can be sent successfully.

Add or Manage Recipient Test Numbers

To send test messages during the WhatsApp Cloud API setup, Meta allows you to define up to **five recipient phone numbers**. These numbers are temporary and intended for development and testing only.

In the **Send and receive messages** section, under the **To** dropdown, click on **Manage phone number list**.



Add or Manage Recipient Test Numbers

This will open the interface where you can add, remove, or update test recipient numbers. These numbers must be WhatsApp-enabled and will receive test messages using your approved templates. You'll configure these in the next step.

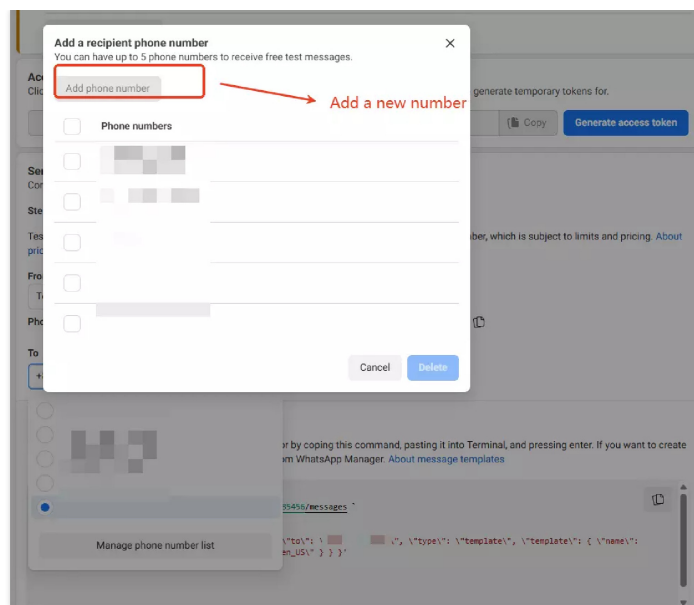
Note: You must use a valid personal or testing phone number here. These numbers are not visible to end users and are only used during the development and validation phase of your integration.

Add a New Test Recipient Number

After clicking **Manage phone number list**, a pop-up window will appear allowing you to manage test recipients.

To add a new number:

1. Click the input field labeled **Add phone number**.
2. Enter a valid mobile number in **E.164 format** (e.g., `+14155550123`).
3. Press **Enter** or confirm to add the number to the list.



Add a New Test Recipient Number

You can define up to **five test recipients**. These numbers will be eligible to receive development messages via the WhatsApp Cloud API during the testing phase. This step is essential before sending any template-based messages from Meta's test interface.

Note:

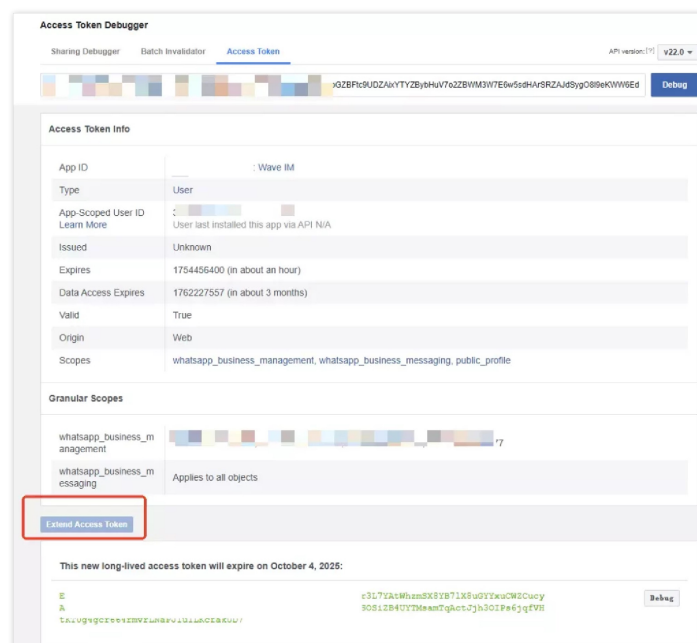
Ensure the phone number has WhatsApp enabled and that it can receive messages directly from your Meta setup.

Extend Access Token (Recommended)

By default, the access token generated for your WhatsApp Cloud API session is **short-lived** and expires within an hour. To ensure consistent operation especially when used with services like GDMS you should generate a **long-lived access token**.

To do this:

1. Navigate to the [Access Token Debugger](#) in the Meta Developer portal.
2. Paste your short-lived token into the tool and click **Debug**.
3. After the information is displayed, click **Extend Access Token** (at the bottom of the page).



Once extended, a **new long-lived token** will be generated. Copy and store this token securely it will be used in the GDMS integration to authenticate WhatsApp messages.

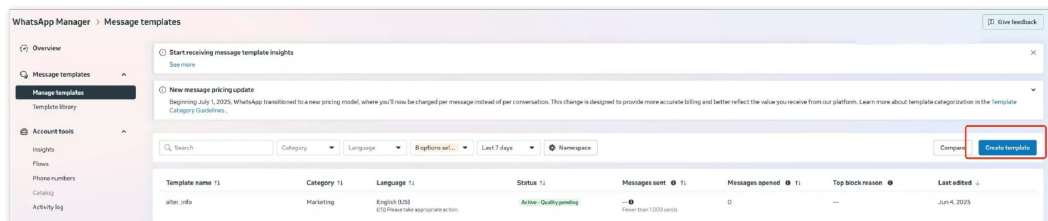
Note: Long-lived tokens can last up to 60 days or more, depending on your app configuration. Always handle this token securely, and avoid exposing it publicly.

Open the Message Templates Section

To send messages through the WhatsApp Cloud API, you must use **pre-approved message templates**. This is a Meta requirement for initiating messages to users via the Cloud API. Templates define the content and format of outbound messages (such as alerts or notifications), and they must be reviewed and approved by Meta before use.

Follow these steps to begin:

1. Navigate to **WhatsApp Manager** in your Meta Developer account.
2. From the left sidebar, go to **Message Templates > Manage Templates**.
3. On the top right, click the **Create template** button.



Open the Message Templates Section

You will now proceed to define a new message template, which will include:

- The **category** of your message,
- The **language**,
- And the content (such as header, body, and optional footer or buttons).

In the next steps, we'll walk through configuring the template content itself.

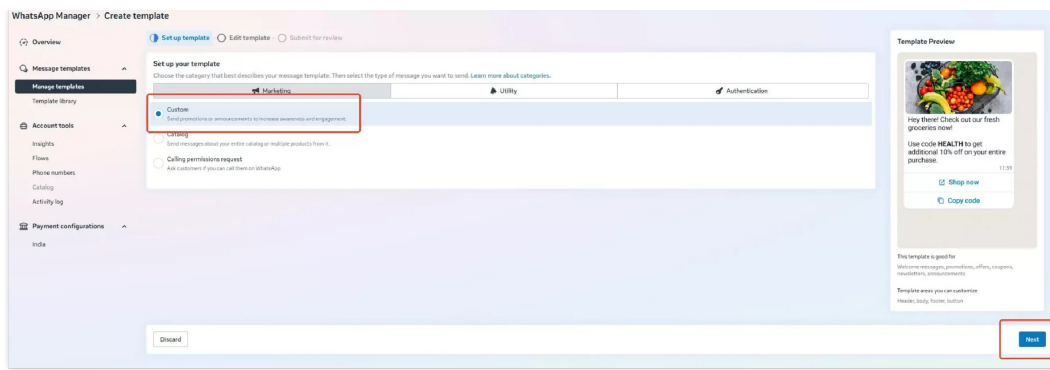
Choose Template Category and Type

In this step, you'll define the **purpose and structure** of your WhatsApp message template.

1. Under **Set up template**, choose a **category** that best reflects the nature of your message. For alert notifications, you can safely use the **"Marketing"** category, then select the **Custom** template type.
 - **Custom** templates are flexible and suitable for general announcements or system alerts.
 - Avoid using Catalog or Call permissions request types, as they are not relevant for GDMS alerting.
2. Once selected, click **Next** to proceed to the content configuration stage.

Important:

In the next step, you must name the template exactly `gdms_networking_alert`. This name is required by the GDMS platform to correctly trigger alert delivery. If the name does not match, the message will not be delivered.



Choose Template Category and Type

Note: Choosing the wrong category or type may lead to rejections during Meta's template approval. Use "Marketing + Custom" for the best chance of approval for alert messages.

Configure Template Content

Now define the structure and content of the WhatsApp message template. This is the message users will receive when an alert is triggered by GDMS Networking.

1. Template Name:

- The template name **must be exactly** `gdms_networking_alert`.
- This specific name is required to match the alert system logic in GDMS Networking. If the name is different, the alert messages will not be delivered.

2. Language:

- Choose the appropriate language for the message (e.g., English US).

3. Content Sections:

- **Header (Optional):** You can include a short title such as *Alerting* (text type).
- **Body (Required):** This is the main content of your alert. You must include a variable placeholder like `{{1}}` so GDMS can insert the dynamic alert message.
 - Example Body: `{{1}}` Please take appropriate action.
 - Example Variable Sample: The configuration of device (C0:74:AD:E0:D7:8F) has been modified by a local user

4. Buttons (Optional):

- You may add action buttons (up to 10) such as "Acknowledge" or "View Dashboard," though they are not required for GDMS alerts.

5. Once done, click **Submit for review** to send the template for Meta approval.

Configure Template Content

Note: Your template will not be usable until it's approved and marked "Active." Avoid using real user data or formatting that violates Meta's template policies.

Monitor Template Approval Status

After submitting your template, it needs to go through Meta's review process. You can track its status from the **Manage templates** section within the WhatsApp Manager.

1. Navigate to the **Message templates** section > **Manage templates**.
2. Locate your submitted template by its name **you must use** `gdms_networking_alert`.

Important:

This specific template name is required by GDMS Networking for message delivery. Other names will not work.

3. Check the **Status** column:
 - A status of **Active – Quality pending** means the template is already approved and usable, but still undergoing additional monitoring.
 - You must wait for the template to reach at least "Active" status before sending live messages through GDMS.

Search	Category	Language	Options	Last 7 days	Namespace	Compare	Create template
Template name	Category	Language	Status	Messages sent	Messages opened	Top block reason	Last edited
alter info	Marketing	English(US)	Active - Quality pending	0	0	---	Jun 4, 2025

Monitor Template Approval Status

Note: You can start using a template as soon as it reaches the "Active" phase, even if quality assessment is still in progress.

Configure WhatsApp Alerts in GDMS Networking / GWN Manager

Once your WhatsApp access token, phone number, and approved message template are ready, you can enable alert delivery via WhatsApp directly from the GDMS or GWN Manager interface.

To do this:

1. Go to **Alerts > Alert Notification** in your GDMS or GWN Manager account.
2. Under the **Third-Party Platforms** section:
 - Select **WhatsApp** from the dropdown menu.

- In the **Telephone ID** field, paste the phone number ID you obtained earlier from the WhatsApp API Setup screen.
- In the **Token** field, paste your long-lived **Access Token**.
- In the **Telephone Number** field, enter the actual WhatsApp number you linked earlier — make sure to enter it in international format **without spaces** (e.g., **8618376391234**).

3. Scroll down to the **Notification Type** section and select which alerts you want to send through WhatsApp:

- Enable the **Third-Party Platform** toggle for any alert type you wish to push to WhatsApp.

For example, you might enable:

- ☒ **AP offline for more than 30 mins**
- ☒ **Router offline for more than 30 mins**
- ☒ **Switch offline for more than 30 mins**
- ...and others, depending on what's important for your environment.

4. Click **Save** once you've finished configuring.

Third-Party Platforms: WhatsApp

* Telephone ID: 108459148685477 (Phone number ID)

* Token: EAAG1Z8EqD1ABPCr3AU5uN6DW7OJ1jY65hYvywhoyXAX9Kv (Access token)

* Telephone Number: 8618376391234 (WhatsApp number (No spaces))

Alert Detail

Alert Detail	Web	Email	App	Third-Party Platform
Manager				
AP offline for more than 30 mins	☐	☐	☐	☒
AP online (After offline for 30 mins)	☐	☐	☐	☒
Router offline for more than 30 mins	☐	☐	☐	☒
Router online (After offline for 30 mins)	☐	☐	☐	☐
Switch online (After offline for 30 mins)	☐	☐	☐	☐
Switch offline for more than 30 mins	☐	☐	☐	☐
Device configuration sync failed	☐	☐	☐	☐
The current device has a time deviation of 30 minutes	☐	☐	☐	☐
Router				
Router upgraded successfully	☐	☐	☐	☐
Router upgrade failed	☐	☐	☐	☐
Router temperature is too high	☐	☐	☐	☐
Switch				
Switch temperature is too high	☐	☐	☐	☐

Buttons: Cancel, Save

Configure WhatsApp Alerts in GDMS Networking GWN Manager

Tip: You can use different alert types for email, app, or WhatsApp independently. Each notification method works in parallel.

Wave Setup

Wave integration allows GDMS Networking and GWN Manager to forward alert notifications directly to Grandstream Wave, so that events such as device status changes or network issues appear as messages inside the Wave client for a specific UCM extension. Using an interface link created in the Wave Management Portal, GDMS Networking / GWN Manager sends alert details to Wave, which then delivers them through the Notification Assistant. Once the interface link, token, and target extension are configured on both sides, supported alerts are pushed in real time to the selected Wave user alongside their regular calls and chats.

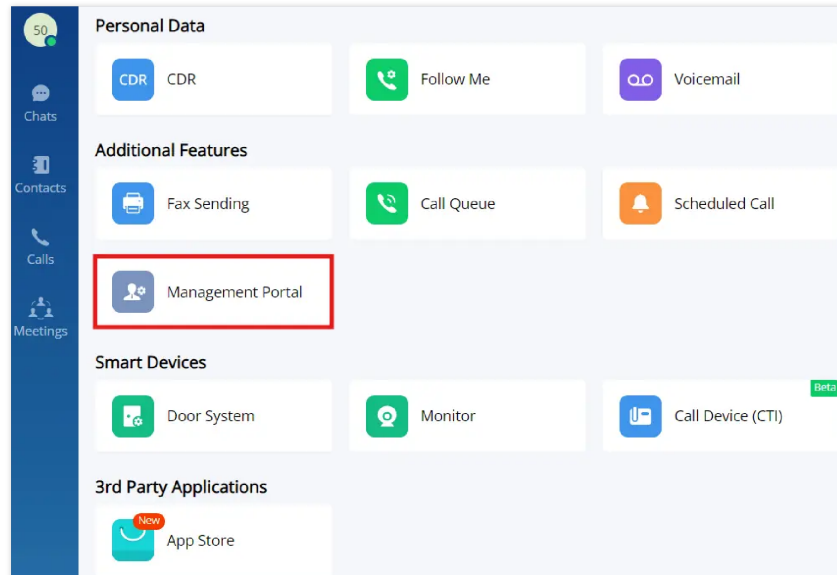
Open the Wave Management Portal

1. Sign in to the Grandstream Wave desktop client using the UCM extension that will receive GDMS Networking alerts.
2. Under Application, locate the **Management Portal** tile under **Additional Features** and click it.

- This opens the management interface for the logged-in Wave user and exposes integration options that can receive notifications from external systems.
- The extension must have the appropriate Wave / user-portal privileges assigned on the UCM side.

For detailed information about Wave roles and permissions, refer to the Grandstream **Wave Administrator Guide**:

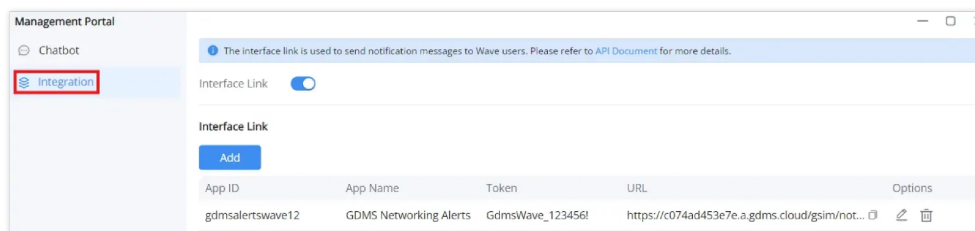
<https://documentation.grandstream.com/knowledge-base/wave-administrator-guide/>



Open the Wave Management Portal

Enable the Integration Interface

1. In the **Management Portal**, use the left navigation menu and select **Integration**.
2. At the top of the page, turn on the **Interface Link** switch.
 - This enables the HTTP interface that allows external systems to send notification messages to Wave users.
3. Confirm that the **Interface Link** section is visible and that existing integration entries (if any) are listed.

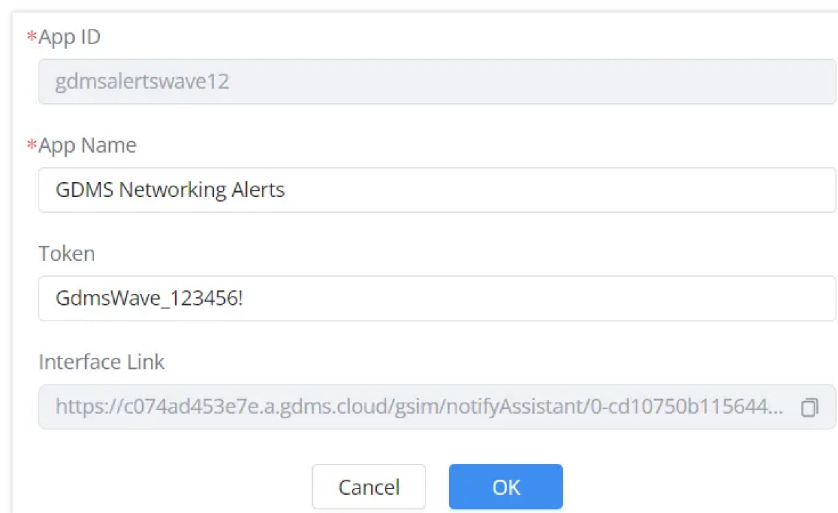


Enable the Integration Interface

Create an Interface Link for GDMS Networking Alerts

1. On the **Integration** page, in the **Interface Link** section, click **Add** to create a new integration entry.
2. In the **Add Interface Link** dialog, configure the following fields:
 - **App ID**
 - Enter a unique identifier for this integration, such as **gdmsalertswave 12**.
 - This value is used internally to identify the application.
 - **App Name**
 - Enter a descriptive name that will appear in Wave, for example **GDMS Networking Alerts**.

- This label will be shown in the Notification Assistant when alerts are received.
 - **Token**
 - Enter a secure string that will be used by GDMS Networking / GWN Manager to authenticate requests to Wave.
 - Example: **GdmsWave_123456!**
3. After filling in the fields, confirm the dialog (for example, by clicking **OK** or **Generate Interface Link**, depending on the client version).
 4. The system automatically generates an **Interface Link** URL for this app. This URL is displayed in the **URL / Interface Link** field of the integration entry.
 5. Copy both of the following values and store them temporarily:
 - The full **Interface Link URL**
 - The **Token** value



The screenshot shows a dialog box with the following fields and values:

- *App ID**: gdmsalertswave12
- *App Name**: GDMS Networking Alerts
- Token**: GdmsWave_123456!
- Interface Link**: <https://c074ad453e7e.a.gdms.cloud/gsim/notifyAssistant/0-cd10750b115644...>

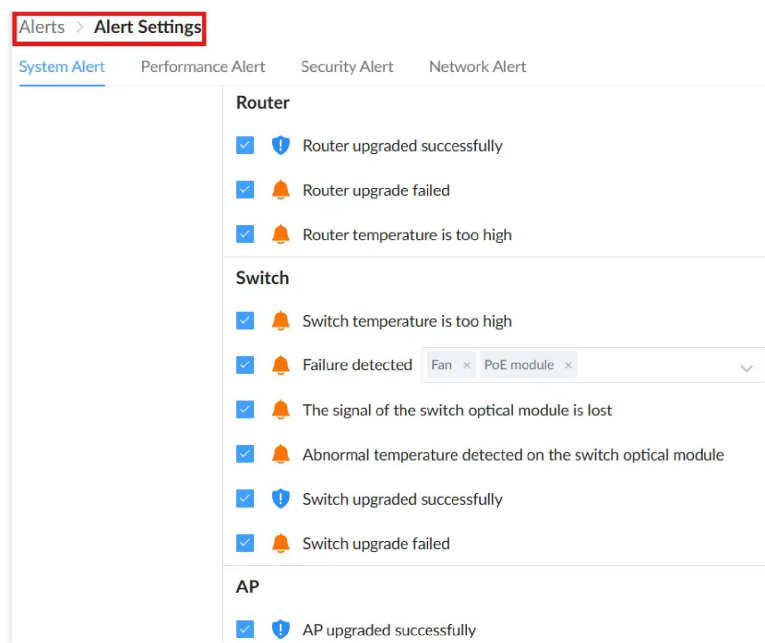
At the bottom of the dialog are two buttons: **Cancel** and **OK**.

Create an Interface Link for GDMS Networking Alerts

These values will be used when configuring Wave as a third-party platform in GDMS Networking / GWN Manager.

Select Which Alerts Will Be Sent to Wave

1. Log in to **GDMS Networking / GWN Manager** with an account that has permission to manage alerts for the target organization or network.
2. In the left navigation menu, go to **Alerts** → **Alert Settings**.
3. For each device category (for example **Router**, **Switch**, **AP**), review the list of available alert types.
4. Enable the checkbox next to each alert you want GDMS Networking to generate and potentially send to Wave.
 - Examples include:
 - *Router temperature is too high*
 - *Switch temperature is too high*
 - *Failure detected*
 - *AP upgraded successfully*
5. Save the alert settings when finished.



Select Which Alerts Will Be Sent to Wave

Enabling an alert here ensures that GDMS Networking can generate the event; the delivery method (including Wave) is configured in the next step.

Configure Wave as a Third-Party Platform in Alert Notification

1. In **GDMS Networking / GWN Manager**, remain under **Alerts** and open **Alert Notification**.
2. In the **3rd-Party Platforms** section, click **Add New Item** if needed, then set **3rd-Party Platforms** to **Wave**.
3. Fill in the Wave integration fields as follows:
 - **Request Address**
 - Paste the **Interface Link URL** that was generated in the Wave Management Portal for the **GDMS Networking Alerts** app.
 - Use the URL exactly as displayed in Wave; do not modify the path or parameters.
 - **Token**
 - Enter the same **Token** value that was configured for the app in the Wave Integration page.
 - This allows Wave to validate that alert requests originate from an authorized source.
 - **Extension**
 - Enter the **UCM extension number** of the Wave user who should receive these alerts (for example, 5050).
 - The specified extension must be registered on the UCM and logged in to Wave.
4. Under the **Notification Type** section, for each alert you want to deliver to Wave, enable the **3rd-Party Platform** toggle.
 - You can independently enable/disable **Web**, **App**, and **3rd-Party Platform** delivery for each alert.
 - To send alerts only to Wave, it is sufficient to enable **3rd-Party Platform** for the relevant entries.
5. Click **Save** to apply the changes.

Alerts > **Alert Notification**

3rd-Party Platforms ⓘ

* Request Address 1-256 characters

* Token 1-512 characters

* Extension ⊖

[Add New Item](#) ⊕

[System Alert](#) [Performance Alert](#) [Security Alert](#) [Network Alert](#)

Alert Detail	Notification Type ⓘ
Manager	
AP offline for more than 30 mins	Web ☐ App ☒ 3rd-Party Platform ☒
AP online (After offline for 30 mins)	Web ☐ App ☒ 3rd-Party Platform ☒
Router offline for more than 30 mins	Web ☐ App ☒ 3rd-Party Platform ☒
Router online (After offline for 30 mins)	Web ☐ App ☒ 3rd-Party Platform ☒

Configure Wave as a Third Party Platform in Alert Notification

Verify Alerts in GDMS Networking / GWN Manager

1. Generate a test condition that will trigger one of the enabled alerts (for example, disconnecting a network port or simulating a device status change in a lab environment).
2. In **Alerts**, open the **Alerts** list view.
3. Confirm that the expected alert entries appear, including the device identifier, alert detail, type, level, and time.
 - This confirms that GDMS Networking / GWN Manager is detecting the event and creating an alert record that can be forwarded to third-party platforms such as Wave.

Alerts ⚙️ Alert Settings 📄 Alert Notification

[Delete](#) [Delete All](#) [Mark All as Read](#)

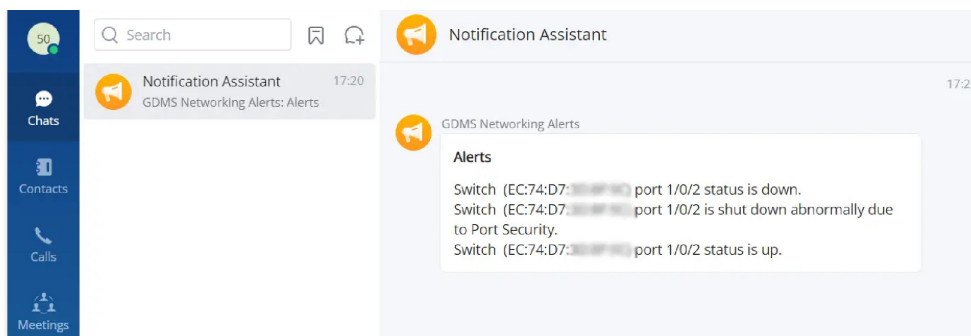
☐ Start Date End Date All Types All All Levels MAC/Name

☐	Alert Detail	Alert Type	Level	Alert Time
☐	Switch [EC:74:D7:...] port 1/0/2 status is down	Network	Notice	2025-11-13 05:09PM
☐	Switch [EC:74:D7:...] port 1/0/2 is shut down abnormally due to Port Security	Network	Warning	2025-11-13 05:09PM
☐	Switch [EC:74:D7:...] port 1/0/2 status is up	Network	Notice	2025-11-13 05:09PM

Verify Alerts in GDMS Networking GWN Manager

Confirm Alert Delivery in Grandstream Wave

1. Return to the **Wave** desktop client that is logged in with the extension configured in the **Extension** field (for example, 5050).
2. In the left navigation menu, go to **Chats**.
3. Open the **Notification Assistant** conversation.
4. Confirm that a new message from **GDMS Networking Alerts** (or the App Name you configured) appears, containing the alert details sent from GDMS Networking / GWN Manager.
 - The message should include information such as device identifier, port information, and alert description (for example, *port status is down* or *port is shut down abnormally due to port security*).
5. Use this notification flow to validate that subsequent network events will be delivered to Wave users in real time.



Confirm Alert Delivery in Grandstream Wave

Trademarks Notice

WhatsApp® is a registered trademark of Meta Platforms, Inc. Telegram® is a registered trademark of Telegram Messenger LLP. All product names, logos, and brands are property of their respective owners. Use of these names does not imply any affiliation with or endorsement by them.
