PACS Unified Connector Guide

Everbridge Suite

April 2026

Everbridge Suite
2026
Printed in the USA

Everbridge, Inc.
8300 Boone Blvd. Suite 800. Vienna, VA 22182
Toll-Free (USA/Canada) +1.888.366.4911

Visit us at www.everbridge.com

Overview

The **Everbridge PACS Connector** enables integration between a **Physical Access Control System (PACS)** and **Everbridge Open iPaaS** to support location awareness and event-driven Incident response.

This guide provides instructions for installing, configuring, and monitoring the unified PACS Connector, as well as configuring Everbridge to process access and security events. It is intended for system administrators responsible for:

- Deploying the PACS Connector
- Configuring integration with supported PACS platforms
- Managing Everbridge iPaaS settings and event workflows

About the PACS Connector

The Everbridge PACS Connector integrates a Physical Access Control System (PACS) with Everbridge Open iPaaS to enable real-time visibility and automated response to Access and Security Events.

The connector collects data from supported PACS platforms, including Badge Access Events, reader activity, and optional Security Events (such as alarms or door conditions), and sends that information to Everbridge for processing.

Within Everbridge, this data is used to:

- Update the **Last Known Location** of Contacts based on badge activity
- Provide location visibility in the Everbridge Map and Contact Records
- Trigger Incidents, Scenarios, and Notifications based on configured Event conditions

How the Connector Works

The PACS Connector runs as a single Windows service within an environment and performs the following functions:

- Connects to the PACS API to collect access and event data
- Normalizes and processes incoming events
- Sends events to Everbridge iPaaS over HTTPS
- Maintains a heartbeat connection for monitoring and health status
- Everbridge iPaaS then routes the data to the appropriate systems:
  - Contacts (Dynamic Locations) for location updates
  - Incident Communication for event-driven Workflows and Notifications

Supported PACS Systems

The connector supports the following PACS platforms:

- **Lenel OnGuard** (via OpenAccess API)
- **Software House C•CURE 9000** (via Victor Web Service REST API)

Deployment Overview

The PACS Connector is typically installed on:

- The PACS server, or
- A server with network access to the PACS API endpoints

NOTE: All communication with Everbridge occurs over HTTPS (port 443).

What Changed in the Unified Connector

The unified Everbridge PACS Connector introduces a simplified architecture and configuration model compared to earlier versions of the integration.

In previous versions, the integration required two separate components:

- **PACS Exporter** - Collected data from the PACS system
- **PACS Importer (Agent)** - Sent data from the Exporter to Everbridge

The unified connector replaces these components with a single Windows service that handles both PACS data collection and communication with Everbridge iPaaS.

> **IMPORTANT:** The legacy PACS Exporter and PACS Importer components are no longer required and should not be installed for new deployments.

Key Changes

Area	Previous Model	Unified Connector
Architecture	Two services (Exporter and Importer)	Single connector service
Installation	Separate installations and configuration for each component	Single installer and service
Configuration	Split across multiple components	Centralized in a single Configuration Editor
Communication	Internal communication between Exporter and Importer	No inter-service communication required
Platform Requirements	.NET Framework 4.x	.NET 8 runtime
Event Processing	Importer handled queuing and delivery	Connector handles collection, queuing, and delivery
Validation	Manual verification of connectivity	Built-in connection testing for PACS

What Remains the Same

Although the connector architecture has changed, the Everbridge-side configuration and behavior remain consistent.

The following concepts still apply:

- Everbridge iPaaS configuration, including agent setup and API key usage
- Identifier mapping between PACS users and Everbridge Contacts
- Reader-to-location mapping for updating Last Known Location
- Security Event configuration, including filters and conditions
- Monitoring and heartbeat validation

Practical Impact

The unified connector simplifies deployment and reduces the number of components to install, configure, and maintain, resulting in:

- Fewer services to manage
- Streamlined configuration workflow
- Reduced risk of misconfiguration between components
- Improved visibility into connectivity and health

Existing iPaaS configurations and mapping strategies can continue to be used with the unified connector.

Migration from Legacy Connector

The unified PACS Connector replaces the legacy **PACS Exporter** and **PACS Importer** components, as described in What Changed in the Unified Connector.

This section outlines how to migrate an existing deployment to the unified connector using the updated installation and configuration model.

> **IMPORTANT:** This section applies to environments upgrading from a legacy PACS Connector deployment. For new installations, proceed to Prerequisites.

Migration Overview

Migration to the unified connector involves:

- Replacing the legacy Exporter and Importer services
- Installing the unified PACS Connector
- Reconfiguring PACS and Everbridge connection settings
- Validating event processing and location updates

Key Changes in Migration

When migrating from a legacy deployment:

- The PACS Exporter and PACS Importer are no longer used.
- Inter-service communication settings (such as Exporter URL and credentials) are not required.
- Configuration is consolidated into a single Configuration Editor.
- Event processing and queueing are handled within the connector.

Migration Considerations

Before migrating:

- Record existing configuration values, including:
  - PACS API endpoints
  - Service Account credentials
  - API key and Agent Configuration
  - Reader-to-location mappings

    - Identifier configuration
- Ensure a rollback plan is available in case issues occur during migration.

Migration Process (High-Level)

1. Stop existing PACS Exporter and Importer services.
2. Install the unified PACS Connector.
3. Configure:
    - PACS connection settings
    - Everbridge iPaaS settings
    - Optional database and proxy settings
4. Start the connector service.
5. Validate:
    - PACS connectivity
    - Heartbeat activity
    - Reader synchronization
    - Access Event processing

Post-Migration Validation

After migration:

- Confirm that existing reader mappings are intact or reconfigured as needed.
- Verify that Contact location updates are functioning correctly.
- Test Security Event processing, if configured.

Legacy Component Removal

After successful validation:

- Remove or uninstall legacy PACS Exporter and PACS Importer components.
- Clean up any unused configuration files or services.

See Also

- For installation steps, see Installing the PACS Connector.
- For configuration details, see Configuring the Connector.
- For validation steps, see Post-Installation Validation.

Prerequisites

Before installing and configuring the Everbridge PACS Connector, ensure that the required system, network, PACS, and Everbridge components are in place.

This section outlines the requirements that must be satisfied to support a successful deployment of the connector and integration with Everbridge Open iPaaS.

Scope of Requirements

The prerequisites are organized into the following areas:

- **System and Network Requirements** - Supported operating systems, runtime dependencies, and connectivity requirements for communication with PACS systems and Everbridge services
- **PACS Requirements** - Supported PACS platforms, API availability, and service account requirements
- **Everbridge Requirements** - iPaaS enablement, agent configuration, identifier setup, and location configuration

General Considerations

Keep the following considerations in mind when working with the PACS connector:

- The PACS Connector is deployed as a Windows service within the customer environment.
- Access to PACS APIs must be available from the connector host.
- Outbound HTTPS connectivity to Everbridge iPaaS is required.
- Required credentials, API keys, and configuration values should be prepared in advance.

System and Network Requirements

The Everbridge PACS Connector requires a supported Windows environment, appropriate runtime dependencies, and network connectivity to both the PACS system and Everbridge Open iPaaS.

System Requirements

The connector must be installed on a supported Windows system.

- **Operating System**
  - Windows Server 2016 or later
  - Windows 10 or Windows 11 (supported for testing or non-production use)
- **Runtime**
  - .NET 8.0 Runtime must be installed on the connector host.
- **Permissions**
  - Administrative privileges are required to install and configure the connector service.

Network Requirements

The connector requires network access to both the PACS system and Everbridge services.

- **PACS Connectivity**
  - Network access from the connector host to the PACS API endpoints
  - PACS communication typically occurs over the internal network.
- **Everbridge Connectivity**
  - Outbound HTTPS access (port 443) to the Everbridge iPaaS endpoint
  - Default endpoint:
    - `https://ipaas-ingestion.everbridge.net/`

Proxy Configuration

Proxy configuration may be required in environments where outbound internet access is restricted.

- Proxy settings apply only to outbound communication to Everbridge iPaaS.
- PACS communication does not use proxy configuration.

- Proxy URL and credentials (if required) should be available during configuration.

Deployment Considerations

The connector is deployed as a Windows service and is typically installed on:

- The PACS server, or
- A server with reliable network access to the PACS API endpoints

The installation host must maintain consistent connectivity to both the PACS system and Everbridge iPaaS to ensure reliable event processing.

PACS Requirements

The Everbridge PACS Connector integrates with supported Physical Access Control Systems (PACS) through their exposed APIs. The PACS environment must be properly configured and accessible to ensure successful data collection and event processing.

Supported PACS Systems

The connector supports the following PACS platforms:

- **Lenel OnGuard**
  - Integration via OpenAccess API
- **Software House C•CURE 9000**
  - Integration via Victor Web Service REST API

API Availability

The PACS system must expose a functioning API endpoint that is accessible from the connector host.

Lenel OnGuard

- OpenAccess API must be enabled and running.
- Default API endpoint:
  - `https://<server>:8080/api/openaccess/`
- The API should be reachable from the connector host.

C•CURE 9000

- Victor Web Service must be installed and running
- Default API endpoint:
  - `http://<server>/victorwebservice/api/`
- Event (SignalR) endpoint:
  - `http://<server>/victorwebservice/signalr`
- The API can be validated by navigating to:
  - `http://<server>/victorwebservice/api/Generic/version`

Service Account Requirements

A service account is required to authenticate with the PACS API.

- The account must have permission to:
  - Access the PACS API
  - Retrieve access events and reader data
- Credentials for this account are configured during connector setup.

Network Accessibility

The connector host must be able to communicate with the PACS system over the network.

- The PACS API endpoints must be reachable from the connector host.
- PACS communication typically occurs over the internal network.
- Proxy configuration is not used for PACS communication.

Licensing Requirements

Ensure the appropriate PACS licenses are installed and active.

- **Lenel OnGuard**
  - License: EPC-311-xxx (based on system size)
- **C•CURE 9000**
  - License: CC9WS-EVERBDG
- **Partner Integration**
  - License: IPC-311-EVRBG01

Everbridge Requirements

The Everbridge PACS Connector requires configuration within Everbridge Open iPaaS to receive, process, and route access and security events.

The Everbridge environment must be prepared before installing and configuring the connector.

iPaaS Enablement

Everbridge Open iPaaS must be enabled for the Account and Organization.

- Contact the Everbridge representative to enable iPaaS if it is not already available.
- Dynamic Locations must also be enabled to support Last Known Location updates.

Agent Configuration and API Key

An iPaaS agent configuration is required to allow the connector to authenticate and send data to Everbridge.

- Create an agent configuration in Everbridge Open.
- An API key is automatically generated during this process.
- The API key is required during connector configuration.

The agent configuration defines how data is received and processed within Everbridge.

Identifier Configuration

A shared identifier must be defined to correlate PACS users with Everbridge contacts.

- The identifier can be:
  - **Everbridge External ID**, or
  - An **Additional Information** field configured in Everbridge
- Supported Additional Information field types:
  - Text (Textbox)
  - Number

This identifier must match the corresponding field in the PACS system.

Contact Data Preparation

Contact Records must include values for the selected identifier.

- Identifier values must be consistent between the PACS system and Everbridge.
- Differences in formatting (such as case, spacing, or leading zeros) may prevent proper matching.

Location Configuration

Locations must be configured in Everbridge to support reader mapping and location updates.

- Upload or create buildings in Everbridge.
- Configure the appropriate **Location Source** for the PACS integration (for example, OnGuard or C•CURE 9000). The Location Source name must match the value sent by the connector.

Integration User (Optional)

An integration user may be configured to represent iPaaS activity within Everbridge. The selected user must have:

- Account Administrator or Organization Administrator permissions
- API access enabled

If configured, this user is displayed as the actor for iPaaS actions.

Configuring Everbridge

Before installing the PACS Connector, the Everbridge environment must be configured to receive and process data from the PACS system.

This configuration establishes how access events are associated with Contacts, how locations are mapped, and how data is routed through Everbridge Open iPaaS.

Configuration Overview

The Everbridge configuration includes the following key components:

- **Identifier Configuration** - Defines how users in the PACS system are matched to Everbridge Contacts.
- **Agent Configuration** - Establishes the connection between the connector and Everbridge iPaaS using an API key.
- **Reader and Location Mapping** - Associates PACS readers with Everbridge locations to support Last Known Location updates.

Process Summary

The general workflow for configuring Everbridge is:

- Define the identifier used to correlate PACS users with Everbridge Contacts.
- Create an iPaaS agent configuration and obtain the API key.
- Configure reader-to-location mappings after the connector begins sending data.

See Also

- For identifier configuration details, see Everbridge Identifier and Client Identifier.
- For agent setup, see Creating an Agent Configuration.
- For mapping readers to locations, see Reader and Location Mapping.

Everbridge Identifier and Client Identifier

The PACS Connector uses a shared **identifier** to correlate users in the PACS system with Contacts in Everbridge.

This mapping ensures that access events are associated with the correct Contact and that Last Known Location updates are applied accurately.

Everbridge Identifier

The **Everbridge Identifier** defines which field in Everbridge is used to match incoming PACS data to contacts. The identifier can be configured as one of the following:

- **External ID** - Uses the Everbridge External ID field when it contains the appropriate values
- **Additional Information Field** - Uses a custom field in Everbridge to store the identifier.
    - Supported Additional Information field types include:
        - Text (Textbox)
        - Number
    - When using an Additional Information field:
        - The field must be created before selecting it as the identifier.
        - The field must contain values that correspond to the user identifier in the PACS system.

Client Identifier

The Client Identifier defines the field in the PACS system that represents the user identity.

The following options are available:

- **Default** - Uses the badge ID as the identifier.
- **Custom Field** - Specifies a different PACS field to use as the identifier.

The selected Client Identifier must correspond to the Everbridge Identifier.

Identifier Mapping Requirements

For successful user resolution, identifier values must match exactly between the PACS system and Everbridge. Any mismatch may prevent access events from being associated with the correct contact. Values must be consistent in:

- Format
- Case
- Spacing
- Leading zeros

Typical identifier values include employee ID or email address, depending on system configuration.

Contact Synchronization

When the Everbridge Identifier is changed, a full Contact synchronization may be required, which updates existing Contacts to use the selected identifier. Without synchronization, previously configured Contacts may not resolve correctly.

Configuration Location

Identifier settings are configured in Everbridge Open from **Settings** > **Everbridge Open** > **iPaaS** > **Settings**. After selecting or changing the Everbridge Identifier, follow the prompts to complete any required synchronization.

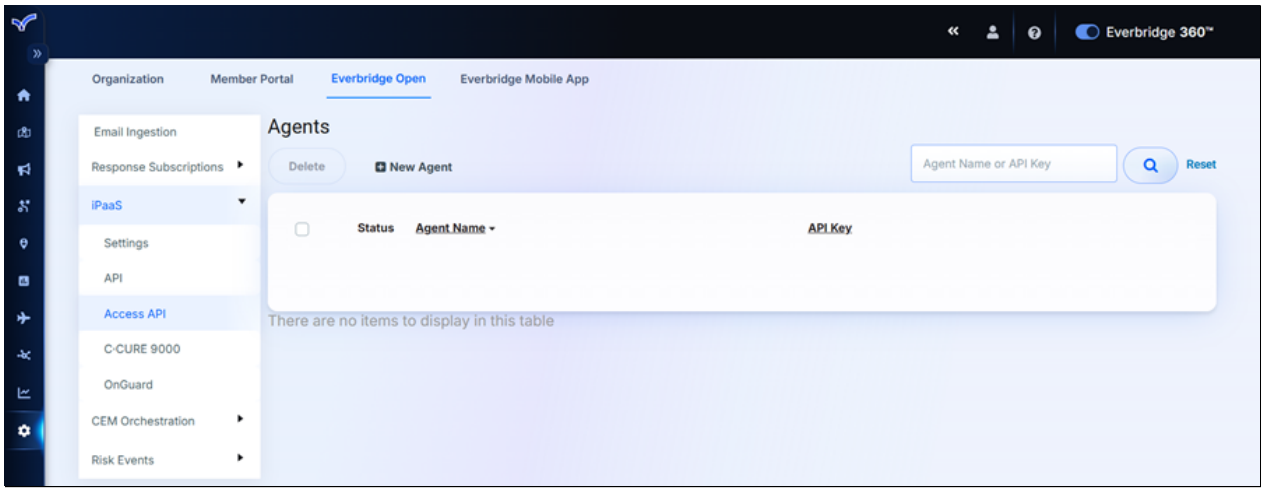
Creating an Agent Configuration

An **Agent Configuration** defines how the PACS Connector communicates with Everbridge Open iPaaS.

Each connector instance uses an Agent Configuration to authenticate with Everbridge and send access and Security Event data.

Accessing Agent Configuration

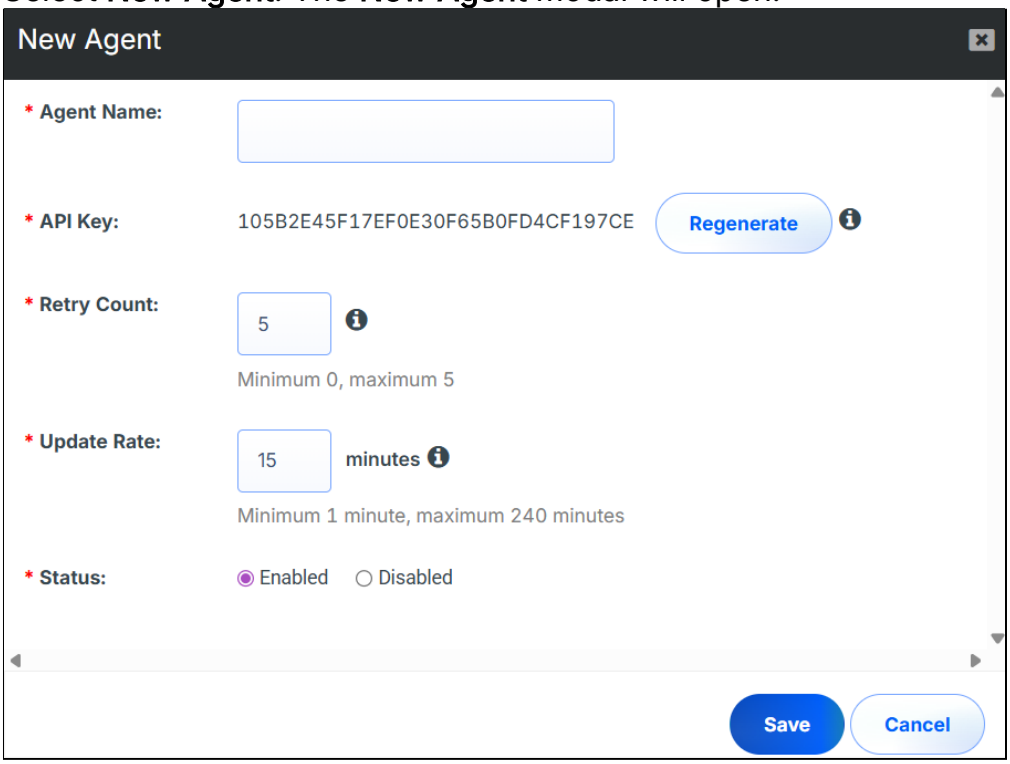
Agent Configurations are managed in **Everbridge Open**, which can be found at the Organization level by navigating to **Settings** > **Everbridge Open** > **iPaaS** > **Access API**.



Creating a New Agent

To create a new Agent Configuration:

1. Select **New Agent**. The **New Agent** modal will open.



2. Enter the required Agent Configuration values:
   - **Agent Name** - A descriptive name used to identify the connector instance.
   - **API Key** - Automatically generated value used by the connector to authenticate with Everbridge iPaaS. This value is required during connector configuration.
   - **Retry Count** - Number of retry attempts for failed messages.
     - Valid range is 0 to 5.
   - **Update Rate** - Frequency at which data is sent to Everbridge.
     - Valid range is 1 to 240 minutes.
   - **Status**
     - **Enabled** - Allows the connector to send data to Everbridge.
     - **Disabled** - Prevents data from being received.
3. Click **Save**.

An API key is generated automatically when the Agent Configuration is created.

Additional Configuration (Optional)

Additional settings may be configured as needed:

- **Integration User** - Specifies the User associated with iPaaS actions. Must have:
  - Account Administrator or Organization Administrator permissions

    - API access enabled
- **Incident Owner** - Specifies the User associated with Incidents created by this Agent Configuration. Must have:
    - Incident Operator or Incident Administrator permissions

If configured, these Users appear in Everbridge as the actors for related actions.

Managing Agent Configurations

Existing Agent Configurations can be modified or deleted from the same interface.

- Select an Agent to update configuration values.
- Regenerate the API key if required.
- Disable or delete Agents that are no longer in use.

Reader and Location Mapping

Reader and location mapping associates PACS badge readers with locations in Everbridge. This mapping enables the connector to update the Last Known Location of Contacts based on badge activity.

When an Access Event is received, the connector uses the mapped reader to determine the corresponding Everbridge location.

How Reader Mapping Works

The PACS Connector sends reader information to Everbridge at regular intervals. Reader records appear in Everbridge Open iPaaS after the connector is running. Each reader must be mapped to a corresponding Everbridge location (typically a building).

Once mapping is complete:

- Access Events at a reader update the contact's Last Known Location.
- Location data becomes available in the Everbridge Map and Contact Records.

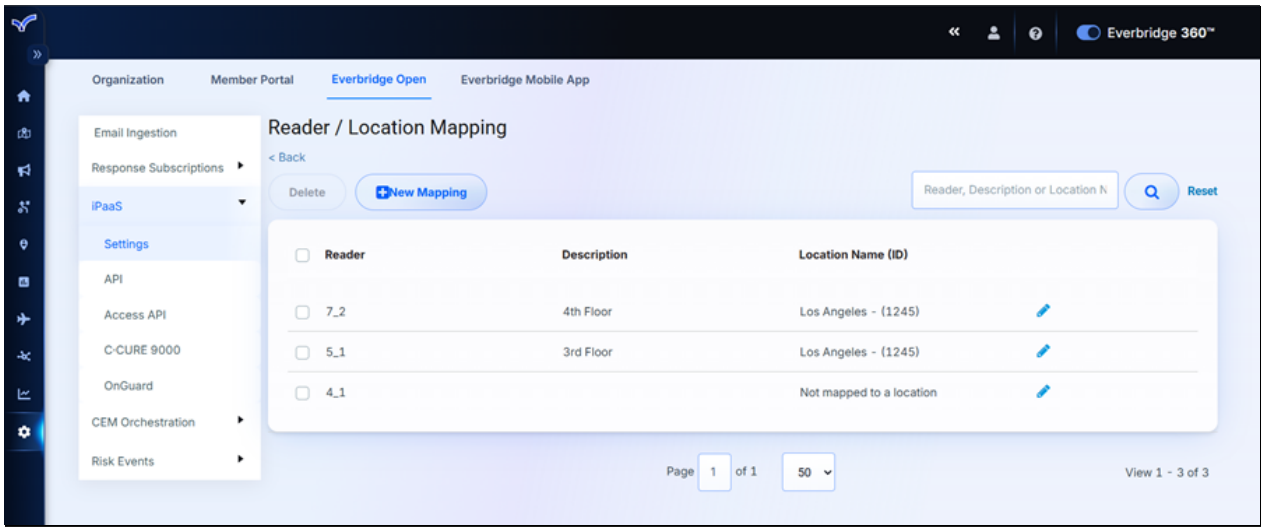
Prerequisites for Mapping

Before mapping readers, ensure the following:

- Buildings or locations are created in Everbridge.
- The appropriate Location Source is configured for the PACS integration.
- The connector is running and successfully sending reader data to Everbridge.

Mapping Readers to Locations

Reader mapping is configured in Everbridge Open from **Settings** > **Everbridge Open** > **iPaaS** > **Settings** > **Reader/Location Mapping**.



From this page:

- View available readers received from the PACS system.
- Assign each reader to a corresponding Everbridge location.
- Modify or remove existing mappings as needed.

Each reader should be mapped to the location that best represents where Access Events occur.

> **NOTE:** Reader information is synchronized periodically. Newly added readers may not appear immediately.

API-Based Mapping (Optional)

For environments with a large number of readers, mapping can be performed using iPaaS API endpoints.

Typical workflow:

1. Retrieve the list of readers from Everbridge.
2. Update each reader record with the corresponding Everbridge Location ID.
3. Submit the updated mappings to Everbridge.

When using the API:

- The **assetId** field must contain the Everbridge Location ID.
- Other fields should not be modified unless required.

Mapping Considerations

Each reader should be mapped to a valid location to ensure accurate updates. Unmapped readers do not update Last Known Location. Changes to mappings take effect as new Access Events are processed.

Validation

After completing reader mapping, to validate:

1. Trigger a Badge Event at a mapped reader.
2. Verify that the Contact's Last Known Location is updated correctly in Everbridge.

Installing the PACS Connector

The Everbridge PACS Connector is deployed as a single Windows service that connects the PACS system to Everbridge Open iPaaS.

This section describes how to install the connector and register the service on a supported system.

Installation Overview

The installation process includes the following steps:

1. Running the connector installer
2. Registering the connector as a Windows service
3. Launching the **Configuration Editor** to define connection settings

After installation, the connector must be configured and started before it begins processing events.

Deployment Considerations

The connector is typically installed on:

- The PACS server, or
- A server with network access to the PACS API endpoints

The installation host must meet all system and network requirements and have access to required credentials and configuration values.

Running the Installer

The PACS Connector is installed using a single installer that registers the connector as a Windows service and launches the **Configuration Editor**.

Prerequisites

Ensure the following prerequisites are met:

- The installation host meets system and network requirements.
- The **.NET 8.0 Runtime** is installed.
- Administrative privileges are available.
- Required PACS and Everbridge configuration values are available.

Installation Steps

To install the PACS Connector:

1. Launch the installer package
2. Accept the license agreement
3. Select the installation folder. The default location is:
    - `C:\Program Files (x86)\Everbridge\Everbridge PACS Connector\`
    - To install in a different location, select **Browse** and choose a folder.
4. Complete the installation.
5. After installation completes, the **Configuration Editor** opens automatically.

Post-Installation Behavior

After the installer completes:

- The PACS Connector is registered as a Windows service.
- The **Configuration Editor** is launched to capture connection settings.
- The connector is not active until configuration is completed and the service is started.

After installation, proceed to Configuring the Connector to define PACS, Everbridge iPaaS, and optional database and proxy settings.

Command-Line Installation (Optional)

The PACS Connector can be installed using a command-line method in environments where the installer package is not used or where automated deployment is required.

When to Use Command-Line Installation

Command-line installation is typically used in the following scenarios:

- Automated or scripted deployments
- Environments where graphical installers are restricted
- Advanced administrative workflows

Installation Steps

To install the connector using the command line:

1. Copy the connector application files to the target installation directory.
2. Open an elevated command prompt (Run as Administrator).
3. Navigate to the installation directory.
4. Register the connector as a Windows service by running the following command:
    - `Detrios.Everbridge.Svc.exe -install`
5. After the service is registered, launch the **Configuration Editor** to define connection settings.

Post-Installation

After completing the command-line installation:

- The connector is registered as a Windows service.
- Configuration must be completed before the service is started.
- The service can be managed through the **Windows Services** console.

Configuring the Connector

After installation, the PACS Connector must be configured to establish connectivity with the PACS system and Everbridge Open iPaaS.

Configuration is performed using the **Configuration Editor**, which is launched automatically after installation and can be reopened from the installation directory.

Configuration Overview

The **Configuration Editor** is used to define all required connection settings for the connector, including:

- **PACS Connection Settings** - Define how the connector connects to the PACS API.
- **Everbridge iPaaS Settings** - Configure the connection to Everbridge using the API key from the agent configuration.
- **Database Settings** - Specify how events are stored and queued before being sent to Everbridge.
- **Proxy Settings (Optional)** - Configure outbound connectivity when a proxy is required.

Configuration Workflow

The typical configuration process includes:

1. Enter **PACS connection details**, including API endpoints and credentials.
2. Enter **Everbridge iPaaS settings**, including the API key.
3. Configure database settings, if required.
4. (Optional) Configure proxy settings.
5. Test the PACS connection.
6. Save the configuration.

After the configuration is saved, the connector is ready to be started.

Configuration File

The connector stores configuration settings in a JSON file located in the installation directory.

- Sensitive values, such as passwords, are encrypted when saved.
- Changes to the configuration file are automatically detected by the connector.

Next Steps

- To configure PACS connectivity, see PACS Connection Settings.
- To configure Everbridge connectivity, see Everbridge iPaaS Settings.
- To define storage options, see Database Settings.
- To validate and finalize configuration, see Testing and Saving the Configuration.

After configuration is complete, proceed to Starting and Verifying the Service.

PACS Connection Settings

PACS Connection Settings define how the connector communicates with the **Physical Access Control System (PACS)** to retrieve access and event data.

These settings are configured in the **Configuration Editor** and establish the connection to the PACS API.

PACS Connection Fields

The following fields are required to configure the PACS connection:

- **PACS Type** - Specifies the PACS platform used by the integration. Supported values:
  - LenelOnGuard
  - SoftwareHouse (C•CURE 9000)
    - Selecting a PACS type automatically populates default API endpoint values.
- **PACS API Server** - The base URL of the PACS API. For example:
  - **OnGuard:**
    - `https://<server>:8080/api/openaccess/`
  - **C•Cure 9000**
    - `http://<server>/victorwebservice/api/`
- **PACS API Event Server** - PACS API Event Server.
  - Not used for Lenel OnGuard.
  - Required for C•CURE 9000. Example:
    - `http://<server>/victorwebservice/signalr`
- **PACS API Username** - The Service Account username used to authenticate with the PACS API.
- **PACS API Password** - The password associated with the PACS API Service Account.
  - This value is encrypted when saved in the configuration file.

Configuration Notes

- PACS connection settings apply only to communication with the PACS system.
- These settings do not configure connectivity to Everbridge.

- The configured credentials must have sufficient permissions to:
  - Access the PACS API
  - Retrieve Access Events and reader data

Validating the Connection

The **Configuration Editor** includes a Test function to validate PACS connectivity. The test verifies:

- API endpoint accessibility
- Authentication using the provided credentials

A successful test confirms that the connector can communicate with the PACS system. If the test fails:

- Verify the API URL and endpoint format.
- Confirm that the PACS service is running and accessible.
- Validate credentials and permissions.

Everbridge iPaaS Settings

Everbridge iPaaS Settings define how the connector communicates with Everbridge Open iPaaS to send access and security event data.

These settings are configured in the **Configuration Editor** and use the API key generated from the Agent Configuration.

Everbridge iPaaS Connection Fields

The following fields are required to configure connectivity to Everbridge:

- **iPaaS URL** - The base URL for the Everbridge iPaaS ingestion API. The appropriate URL for the environment should be confirmed during deployment. Default value:
  - `https://ipaas-ingestion.everbridge.net/`
- **API Key** - The authentication key associated with the Agent Configuration in Everbridge.
  - Generated when creating an Agent Configuration
  - Required for the connector to send data to Everbridge

Configuration Notes

- The API key must correspond to a valid and enabled Agent Configuration.
- Each connector instance should use a dedicated agent configuration to simplify monitoring and management.
- The connector uses these settings to:
  - Authenticate with Everbridge
  - Send Access Events and Security Events
  - Transmit reader and location data

Connectivity Requirements

- Outbound HTTPS access (port 443) to the iPaaS URL is required.
- If a proxy is required for outbound connectivity, it must be configured separately in the connector settings.

Validation and Behavior

- Everbridge connectivity is validated when the connector service starts
- A successful connection results in:

- Heartbeat messages being sent to Everbridge
- The agent appearing as active in the iPaaS interface

If connectivity issues occur:

- Verify the API key is correct and active.
- Confirm network access to the iPaaS endpoint.
- Validate proxy configuration, if applicable.

Database Settings

Database Settings define how the connector stores and queues events before sending them to Everbridge Open iPaaS.

The connector uses a local data store to ensure that events are not lost if connectivity to Everbridge is temporarily unavailable.

Database Options

The connector supports two storage options:

- **SQLite** (Default) - A lightweight, file-based database that requires no additional configuration
- **SQL Server** (Optional) - A full database deployment used for environments with higher volume or specific data management requirements

> **NOTE:** SQLite stores data locally in the installation directory and requires write access for the Service Account.

Configuration Fields

The following fields are used to configure database settings:

- **Using SQL Server** - Specifies whether SQL Server is used for event storage
  - **No** - Uses the default SQLite database
  - **Yes** - Enables SQL Server configuration
- **Database Connection String** - The connection string used to connect to the SQL Server database
  - Required only when SQL Server is enabled
  - Must reference an existing database instance
    - Example:

```
Server=<server>;Database=<database>;User
Id=<user>;Password=<password>;
```

Configuration Notes

- SQLite is recommended for most deployments and requires no additional setup

- When using SQL Server:
  - The target database must already exist.
  - Required tables are created automatically by the connector.
- The database is used to:
  - Temporarily store events before transmission
  - Maintain reliability during network interruptions

Behavior and Considerations

- Events are queued locally and sent to Everbridge at regular intervals.
- If connectivity to Everbridge is interrupted, Events remain in the queue until connectivity is restored.
- Database performance may impact event throughput in high-volume environments.

Testing and Saving the Configuration

After entering configuration values, the settings must be validated and saved before the connector can be started.

The **Configuration Editor** provides tools to test PACS connectivity and persist configuration settings.

Testing the PACS Connection

The Test function validates connectivity to the PACS system.

- Verifies that the PACS API endpoint is reachable
- Confirms authentication using the configured credentials

A successful test indicates that the connector can communicate with the PACS system. If the test fails:

- Verify the PACS API Server URL and endpoint format.
- Confirm that the PACS API service is running.
- Validate the username and password.
- Check network connectivity between the connector host and the PACS system.

Saving Configuration

After validating the PACS connection, save the configuration.

- Select **Save** in the **Configuration Editor**.
- Configuration settings are written to a JSON file in the installation directory.

When saving configuration:

- Sensitive values, such as passwords, are encrypted.
- Existing configuration values are overwritten.

Configuration Behavior

After the configuration is saved:

- The connector monitors the configuration file for changes.
- Updates to configuration settings are detected automatically.
- A service restart is not required for most configuration changes.

NOTE: The connector does not process events until configuration is saved and the service is started.

Proxy Configuration

Proxy settings define how the connector communicates with Everbridge Open iPaaS in environments where outbound internet access requires a proxy server.

Proxy configuration is optional and should be used only when required by network policy.

When Proxy Configuration Is Required

Proxy settings are needed when:

- Outbound HTTPS traffic must pass through a proxy server.
- Direct access to external endpoints is restricted.

If the connector host can access the Everbridge iPaaS endpoint directly, proxy configuration is not required.

Proxy Configuration Fields

The following fields are used to configure proxy settings:

- **Proxy URL** - The address of the proxy server. For example:
  - `http://proxy.company.com:8080`
- **Proxy Username** - The username used to authenticate with the proxy server
  - Optional if the proxy does not require authentication
- **Proxy Domain** - The domain associated with the proxy user account
  - Optional
  - Required only in environments using domain-based authentication
- **Proxy Password** - The password for the proxy user account
  - Optional
  - Encrypted when saved in the configuration file

Configuration Notes

- Proxy settings apply only to communication with Everbridge iPaaS.
- PACS communication does not use proxy configuration.
- Proxy credentials must allow outbound HTTPS access to the iPaaS endpoint.

Starting and Verifying the Service

After configuration is complete, the PACS Connector service must be started to begin processing events and communicating with Everbridge Open iPaaS.

This section describes how to configure the Windows service, start the connector, and verify that it is operating correctly.

Service Overview

The PACS Connector runs as a Windows service on the installation host. Once started, the service performs the following functions:

- Connects to the PACS system to retrieve access and event data
- Sends data to Everbridge Open iPaaS
- Synchronizes reader information
- Maintains a heartbeat for monitoring and health status

Process Summary

Starting and validating the connector includes the following steps:

1. Configure the Windows service logon account, if required.
2. Start the connector service.
3. Verify that the service is running and processing data.

Verification Overview

Successful operation can be confirmed through:

- Connector log output
- Heartbeat activity in Everbridge iPaaS
- Reader synchronization
- Access Event processing

Next Steps

- To configure the service account, see Configuring the Windows Service Log On.
- To start the service, see Starting the Service.
- To validate connector operation, see Post-Installation Validation.

Configuring the Windows Service Log On

The PACS Connector runs as a Windows service and may require a specific User account to access PACS resources and system directories.

By default, the service is installed using the **LocalSystem** account. In environments where additional permissions are required, the **Service Logon Account** should be updated.

When to Change the Service Account

Updating the Service Logon Account may be necessary when:

- The PACS system requires authentication using a domain account
- Access to network resources or APIs is restricted
- The LocalSystem account does not have sufficient permissions

Configuring the Service Log On Account

To update the Service Logon Account:

1. Open the **Windows Run** dialog.
2. Enter:
    - `services.msc`
3. Select **OK** to open the **Services** console.
4. Locate the **PACS Connector** service.
5. Right-click the service and select **Properties**.
6. Select the **Log On** tab.
7. Select **This account** and enter the required credentials
8. Select **OK**.

If prompted to grant the **Log On As A Service** right, accept the prompt.

Verification

After updating the Service Account:

- Confirm that the correct account is displayed in the Log On As column
- Ensure the account has:
    - Permission to access the PACS API
    - Read and write access to the installation and log directories

NOTE: A domain account is commonly required in environments that use Active Directory–based authentication.

Starting the Service

After configuration is complete, the PACS Connector service must be started to begin processing events and communicating with Everbridge Open iPaaS.

To start the PACS Connector service:

1. Open the Windows Run dialog
2. Enter:
   - `services.msc`
3. Select **OK** to open the **Services** console.
4. Locate the **PACS Connector** service.
5. Right-click the service and select **Start**.
6. Verify that the **Status** is displayed as **Running**.

Configuring Startup Behavior (Optional)

To configure the service to start automatically when the system starts:

1. Right-click the service and select **Properties**.
2. On the **General** tab, set the **Startup type** to **Automatic**.
3. Select **OK**.

Service Behavior

When the service is running:

- The connector establishes communication with the PACS system.
- Event data is collected and processed.
- Data is transmitted to Everbridge Open iPaaS.
- Heartbeat messages are sent at regular intervals.

Post-Installation Validation

After the PACS Connector service is started, validate that the connector is operating correctly and communicating with both the PACS system and Everbridge Open iPaaS.

Successful validation confirms that events are being processed and that the integration is functioning as expected.

Validation Checklist

Verify the following:

- The connector service is running.
- The connector is successfully connecting to the PACS system.
- Heartbeat messages are being sent to Everbridge.
- Reader data is synchronized.
- Access Events are processed and received by Everbridge.

Reviewing Connector Logs

Connector activity is recorded in the log file located on the installation host.

1. Navigate to the log directory:
   - `C:\ProgramData\Detrios\logs\`
2. Open the log file:
   - `Detrios.Everbridge.log`
3. Look for:
   - Successful connection messages to the PACS system
   - Successful communication with Everbridge iPaaS
   - Reader synchronization activity
   - Absence of repeated error messages

Verifying Heartbeat Activity

The connector sends periodic heartbeat messages to Everbridge iPaaS.

- Confirm that the agent appears as active in the Everbridge iPaaS interface.
- Verify that heartbeat updates are received at regular intervals.

A valid heartbeat indicates that the connector is running and able to communicate with Everbridge.

Validating Reader Synchronization

After the connector starts:

- Reader data is sent to Everbridge.
- Readers appear in the Reader/Location Mapping interface.

Confirm that:

- Readers are visible in Everbridge.
- Reader mappings can be configured.

> **NOTE:** In new installations, the PACS Connector may require additional time to process and synchronize initial data from the PACS system. During this period, reader data and Access Events may not appear immediately in Everbridge. The duration of initial synchronization depends on the size of the PACS environment, including the number of cardholders, readers, and events. In large deployments, this process may take several hours to complete.

Testing Access Events

To confirm end-to-end functionality:

1. Trigger a badge access event at a mapped reader
2. Verify that:
    - The event is processed by the connector.
    - The corresponding contact is updated in Everbridge.
    - The Last Known Location reflects the correct location.

Troubleshooting Validation Issues

If validation fails:

- Review connector logs for errors.
- Verify PACS API connectivity and credentials.
- Confirm the API key and Everbridge connectivity.
- Check network access and proxy configuration, if applicable.

NOTE: The connector is considered fully operational when all validation checks pass and access events are successfully reflected in Everbridge.

Identity Resolution and Reader Mapping

Accurate identity resolution and reader mapping are required for the PACS Connector to correctly associate Access Events with Contacts and locations in Everbridge.

These configurations determine:

- Which Contact is associated with an Access Event
- Which location is assigned based on reader activity

Overview

Identity Resolution defines how users in the PACS system are matched to Everbridge contacts, while **Reader Mapping** associates PACS readers with Everbridge locations.

Both configurations must be accurate for the connector to update Last Known Location and support event-driven workflows.

Key Considerations

- Identifier values must match exactly between the PACS system and Everbridge.
- Readers must be mapped to valid locations.
- Misconfiguration may result in:
  - Events not being associated with Contacts
  - Incorrect or missing location updates

Next Steps

- For details on identifier matching, see Identity Resolution.
- For mapping guidance, see Reader Mapping Best Practices.

Identity Resolution

Identity resolution determines how Access Events from the PACS system are matched to Contacts in Everbridge.

Accurate identity resolution ensures that events are associated with the correct Contact and that Last Known Location updates are applied correctly.

How Identity Resolution Works

The PACS Connector uses a shared identifier to correlate:

- A User Record in the PACS system
- A Contact Record in Everbridge

When an Access Event is received:

- The connector extracts the identifier value from the event.
- The value is matched to the configured Everbridge Identifier.
- If a match is found, the event is associated with the corresponding Contact.

Identifier Matching Requirements

For successful identity resolution, identifier values must match exactly between systems.

- Values must be consistent in:
  - Format
  - Case
  - Spacing
  - Leading zeros
- Any mismatch may prevent the connector from resolving the event to a Contact.

Typical identifier values include employee ID or email address.

Common Resolution Issues

Identity resolution may fail under the following conditions:

- Identifier values are missing from Contact Records.
- Values are formatted differently between systems.
- The selected identifier field does not match the PACS field.
- Contact data is outdated or not synchronized.

When identity resolution fails:

- Access Events are received but not associated with a Contact.
- Last Known Location is not updated.

> **IMPORTANT:** Identity resolution depends on exact value matching. Even minor formatting differences may prevent correct association.

Maintaining Data Consistency

To ensure reliable identity resolution:

- Keep Contact data synchronized with the PACS system.
- Use a consistent identifier format across systems.
- Validate identifier values during initial setup and ongoing updates.

See Also

- For configuration details, see Everbridge Identifier and Client Identifier.
- For location mapping, see Reader Mapping Best Practices.

Reader Mapping Best Practices

Reader mapping determines how Access Events are translated into locations in Everbridge. Accurate mapping ensures that Last Known Location updates reflect the correct physical location of Contacts.

Mapping Principles

Each PACS reader should be mapped to a location that best represents where access events occur.

- Map readers to the most relevant building or location.
- Ensure that each reader has a valid mapping.
- Avoid leaving frequently used readers unmapped.

Location Accuracy

Accurate location mapping is critical for effective event response.

- Map readers to locations that reflect real-world access points.
- Avoid overly broad mappings that reduce location precision.
- Ensure that building and location data in Everbridge is up to date.

Handling Multiple Readers

In environments with many readers:

- Use consistent naming conventions for readers and locations.
- Group readers logically by building or area.
- Verify mappings in bulk when onboarding new sites or systems.

Unmapped Readers

Readers that are not mapped to a location do not update Last Known Location. Access events from unmapped readers are still received, however, no location updates are applied.

Regularly review reader lists to identify and map new or unmapped readers.

Validating Mappings

After configuring mappings:

- Trigger Access Events at representative readers.
- Confirm that the correct location is applied to the Contact.

- Verify results in the Everbridge Map or Contact Record.

Maintaining Mappings

Reader mappings should be reviewed periodically.

- Update mappings when:
  - New readers are added
  - Buildings or locations change
  - PACS configurations are modified
- Remove or update outdated mappings to maintain accuracy.

See Also

- For mapping configuration steps, see Reader and Location Mapping.
- For contact matching behavior, see Identity Resolution.

Configuring Security Events

The PACS Connector can process Security Events from the PACS system and route them through Everbridge Open iPaaS to support automated Incident response.

Security Event configuration defines how incoming events are filtered, evaluated, and used to trigger Incidents, Scenarios, or Notifications.

Overview

Security event processing includes the following components:

Event Filtering - Determines which events are sent from the PACS system to Everbridge.
Incident Ownership and Attribution - Defines which users are associated with actions performed by iPaaS.
Conditions and Notifications - Evaluates incoming events and triggers predefined responses.

How Security Events Are Processed

When Security Event integration is enabled:

- Events are received from the PACS system.
- Events are filtered based on configured criteria.
- Events are evaluated against defined conditions.
- Matching conditions trigger Incidents, Scenarios, or Notifications.

If no conditions match, the event is received but no action is taken.

Key Considerations

- Security Event integration is optional and disabled by default.
- Event filters should be configured carefully to avoid excessive or irrelevant events.
- Conditions are evaluated in priority order, and the first match determines the outcome.

See Also

- To enable and configure event filtering, see Enabling Security Events and Event Filtering.
- To configure ownership settings, see Incident Owner and Integration User.
- To define event-driven responses, see Conditions and Notifications.

Enabling Security Events

Security Event Integration must be enabled before the PACS Connector can process and forward security-related events to Everbridge Open iPaaS.

By default, Security Event processing is disabled.

Enabling Event Processing

Security event processing is configured within the agent configuration in Everbridge Open.

To enable Security Events:

1. Navigate to the Agent Configuration in Everbridge Open.
2. Locate the **Event Filters** setting.
3. Select one of the following options:
    - **Include events**
    - **Exclude events**

Event Filtering

Event filtering determines which Security Events are sent from the PACS system to Everbridge Open iPaaS.

Filters are configured as part of the **Agent Configuration** in **Settings** > **Everbridge Open** > **iPaaS** > **Access API** and control which events are included or excluded before further processing.

Filter Configuration

Event filtering is configured using the **Event Filters** setting in the Agent Configuration.

- Select one of the following options:
  - Include events
  - Exclude events
- Specify one or more event values to include or exclude

Event Filter Behavior

- If no filter values are specified, no Security Events are sent.
- Filter values support simple wildcard matching:
  - An asterisk (*****) represents any sequence of characters.
- To include all events, specify:
  - `*`
- Multiple values can be specified as a comma-separated list. For example:
  - `duress*,LNL_Access*`
- Filter values are case-sensitive.

> **NOTE:** Using an asterisk (*****) includes all available Event Types and should be used cautiously to avoid excessive event volume.

Filtering Considerations

Keep the following considerations in mind when using filtering:

- Use specific filter values to limit event volume.
- Avoid broad patterns unless all events are required.
- Review event naming conventions in the PACS system to define effective filters.
- Filtering applies only to Security Events.
- Access Events used for location updates are not affected by these filters.

Behavior After Filtering

After filtering is applied:

- Only matching events are forwarded to Everbridge.
- Events are evaluated against configured conditions.
- If a condition matches, an Incident, Scenario, or Notification is triggered.
- If no conditions match, the event is received but no action is taken.

See Also

- To enable event processing, see Enabling Security Events.
  To define event-driven responses, see Conditions and Notifications.

Incident Owner and Integration User

The **Integration User** and **Incident Owner** settings define how actions performed by Everbridge Open iPaaS are attributed within Everbridge.

These settings control which user is associated with automated actions, such as processing events and creating Incidents.

Integration User

The Integration User represents the User Account associated with general iPaaS activity. It displays as the actor for actions performed by iPaaS, and also applies to updates, such as data processing and system activity.

Requirements

The Integration User must:

- Be an **Account Administrator** or **Organization Administrator**
- Have **API access** enabled

Incident Owner

The Incident Owner represents the User Account associated with Incidents created by Security Event conditions. It's configured at the Agent level, and displays as the user who created, updated, and closed Incidents and Scenarios.

Requirements

The Incident Owner must be an **Incident Operator** or **Incident Administrator**.

When to Configure These Settings

These settings are optional but recommended when:

- Tracking ownership of automated actions is required
- Distinguishing between system activity and incident ownership is important
- Auditing or reporting on event-driven activity

Configuration Notes

- Usernames must match exactly and are case-sensitive.
- If the configured user is invalid or removed, event processing may fail.

- If both values are configured:
  - The Incident Owner is used for Incident-related actions.
  - The Integration User is used for general iPaaS activity.

See Also

- For Agent Configuration details, see Creating an Agent Configuration.
- For event-driven behavior, see Conditions and Notifications.

Conditions and Notifications

Conditions define how Security Events are evaluated and determine when Incidents, Scenarios, or Notifications are triggered in Everbridge.

Each condition specifies criteria that incoming events must match in order to initiate a predefined response.

How Conditions Work

When a Security Event is received:

- The event is evaluated against configured conditions.
- Conditions are processed in priority order.
- The first matching condition determines the action taken.

If a condition matches, an Incident or Scenario is created, and a Notification is sent using the selected template. If no conditions match, the event is received but no action is performed.

Creating a Condition

Conditions are configured within the **Agent Configuration** in **Everbridge Open** from **Settings** > **Everbridge Open** > **iPaaS** > **Access API**.

To create a condition:

1. Open the **Agent Configuration**.
2. Navigate to the **Conditions** section.
3. Select **New Condition**.
4. Enter the required configuration values:
    - **Condition Name** - A descriptive name used to identify the condition
    - **Launch** - The Notification Template or Scenario triggered when the condition matches
        - Only active templates are available for selection.
    - **Immediately Close** - Specifies whether the Incident is closed automatically after the Notification is sent
    - **Event Criteria** (Key-Value Pairs) - Defines the conditions that incoming events must match
        - Criteria are specified as key-value pairs.
        - Valid characters include:
            - Letters (A–Z, a–z)

        - Numbers (0–9)
        - Underscore (_)
        - Hyphen (-)
    - **Update Incident** - Defines how updates to matching events are handled. Options include:
        - Send a Notification when changes occur
        - Ignore updates
    - **Close Incident** - Defines how Incident closure is handled. Options include:
        - Send a Notification on closure
        - Close without Notification
5. Save the condition.

Condition Ordering

Conditions are evaluated in the order in which they are listed. The first condition that matches an event determines the outcome, while remaining conditions are not evaluated for that event. Reordering conditions changes the priority of event evaluation.

Configuration Considerations

Keep the following considerations in mind when working with conditions:

- Define conditions using clear and specific criteria.
- Avoid overlapping conditions that may produce unintended results.
- Place higher-priority conditions earlier in the list.
- Ensure that the selected templates are active and properly configured.
- Validate conditions using representative event data where possible.

See Also

- For event filtering, see Event Filtering.
- For ownership settings, see Incident Owner and Integration User.

Monitoring the Connector

The PACS Connector provides visibility into system activity, health status, and event processing through Everbridge Open iPaaS and connector logs.

Monitoring ensures that the connector is operating correctly and that access and Security Events are being received and processed as expected.

Overview

Connector monitoring includes the following areas:

- **iPaaS Activity and Message Status** - Provides visibility into incoming events, processing status, and message outcomes
- **Heartbeat Monitoring** - Indicates whether the connector is actively communicating with Everbridge
- **Location Updates** - Confirms that Access Events are updating Contact locations correctly

Monitoring Objectives

Monitoring should confirm that:

- The connector is running and connected to Everbridge.
- Events are being received and processed successfully.
- Errors or failures are identified and addressed.
- Location updates reflect expected behavior.

Key Health Indicators

A healthy connector typically shows:

- Regular heartbeat activity
- Successful message processing in iPaaS
- Consistent reader synchronization
- Accurate Last Known Location updates

See Also

- To review activity and message status, see iPaaS Activity and Agent Health.
- To monitor connectivity, see Heartbeat Monitoring.
- To validate location updates, see Viewing Location Updates.

iPaaS Activity and Agent Health

Everbridge Open iPaaS provides visibility into connector activity, message processing, and overall agent health. This information can be used to monitor event flow, identify errors, and verify that the connector is operating correctly.

Viewing iPaaS Activity

Connector activity can be viewed in the **iPaaS** interface from **Settings** > **Everbridge Open** > **iPaaS** > **Activity**. The **Activity** view displays messages received from the connector and their processing status.

Message Details

The following information is available for each message:

- **Source** - Identifies the source system of the message
- **Agent Name** - The name of the Agent Configuration associated with the connector
- **Request ID** - A unique identifier for the message
- **Message Status** - Indicates the processing result. Possible values:
  - SUCCESS
  - FAIL
  - INPROGRESS
- **Received Date** - The date and time the message was received
- **Source ID** - Identifier associated with the originating event
- **Everbridge ID** - The ID of the associated Incident or Scenario, if one was created

> **NOTE:** Selecting a Request ID provides additional details for troubleshooting.

Agent Health Status

Agent health indicates whether the connector is actively communicating with Everbridge, which is based on periodic heartbeat checks. A healthy agent sends regular heartbeat messages, while missing or delayed heartbeats may indicate a connectivity or service issue.

Typical indicators include:

- **Healthy** (Normal) - Regular communication with Everbridge
- **Warning** - Intermittent communication issues
- **Error** - No communication detected

Monitoring Considerations

Consider the following when reviewing agent health:

- Review message status regularly to identify failed or delayed processing.
- Investigate repeated failures or error conditions.
- Confirm that expected events are being received and processed.
- Use activity data to:
  - Validate configuration changes
  - Troubleshoot event processing issues
  - Monitor ongoing system behavior

See Also

- For connectivity validation, see Heartbeat Monitoring.
- For troubleshooting issues, see Troubleshooting.

Heartbeat Monitoring

The PACS Connector sends periodic heartbeat messages to Everbridge Open iPaaS to indicate that the service is running and able to communicate with Everbridge.

Heartbeat monitoring provides a reliable way to verify connector health and detect connectivity issues.

Heartbeat Behavior

The connector sends heartbeat messages at regular intervals. A successful heartbeat indicates that:

- The service is running.
- The connector can communicate with Everbridge iPaaS.

Missing or delayed heartbeat messages may indicate:

- Network connectivity issues
- Service interruption
- Configuration problems

Verifying Heartbeat Activity

Heartbeat activity can be verified in Everbridge:

- Confirm that the agent appears as active in the iPaaS interface.
- Verify that heartbeat updates are received at expected intervals.

Consistent heartbeat activity indicates that the connector is operating normally.

Heartbeat API (Optional)

Heartbeat status can also be monitored using the iPaaS API via the following endpoint:

- `GET https://ipaas-ingestion.everbridge.net/ipaas/v1/agent/heartbeat`

Expected behavior

- **HTTP 200 (OK)** - Indicates a valid heartbeat response.
- **HTTP 404 (Not Found)** - Indicates that no heartbeat has been received.

- Any other response may indicate an error condition

The response includes:

- Current connection state
- Health status of the connector
- Any reported errors

Monitoring Considerations

Keep the following monitoring considerations in mind:

- Use heartbeat monitoring to detect service interruptions quickly.
- Integrate the heartbeat endpoint with external monitoring tools, if required.
- Investigate missing or irregular heartbeat activity promptly.

See Also

- For overall activity monitoring, see iPaaS Activity and Agent Health.
- For troubleshooting issues, see Troubleshooting.

Viewing Location Updates

Access Events processed by the PACS Connector update the Last Known Location of Contacts in Everbridge. These updates can be viewed in Everbridge to confirm that identity resolution and reader mapping are functioning correctly.

Where Location Updates Appear

Location updates are available in the following areas:

- **Contact Record**
  - Displays the Contact's Last Known Location
  - Includes the most recent location and timestamp
- **Dynamic Locations** (Upload View)
  - Displays batches of location updates received from the connector
  - Provides visibility into successful and failed updates

Viewing Location Details

To view location updates for a Contact:

1. Open the Contact Record in Everbridge.
2. Navigate to the **Dynamic – Last Known Locations** section.

This view shows the current location associated with the Contact and the time of the most recent update.

Validating Location Updates

To confirm that location updates are working as expected:

- Trigger an Access Event at a mapped reader
- Verify that:
  - The correct Contact is identified.
  - The correct location is assigned.
  - The update appears within the expected timeframe.

Troubleshooting Missing or Incorrect Updates

If location updates are not appearing or are incorrect:

- Verify that identity resolution is configured correctly.
- Confirm that the reader is mapped to the correct location.
- Ensure that Access Events are being received and processed.

- Review connector logs and iPaaS activity for errors.

Location Monitoring Considerations

- Regularly review location updates to ensure accuracy.
- Validate updates after configuration changes.
- Investigate discrepancies between expected and actual locations.

See Also

- For mapping configuration, see Reader and Location Mapping.
- For identity matching, see Identity Resolution.
- For activity monitoring, see iPaaS Activity and Agent Health.

Advanced Configuration

The PACS Connector provides additional advanced configuration options for fine-tuning event processing, performance, and system behavior.

These settings are intended for advanced use cases and are typically configured by experienced administrators or in coordination with Everbridge Support.

> **IMPORTANT:** Modify advanced configuration settings only when required. Incorrect values may impact connector performance or reliability.

Overview

Advanced configuration settings are defined in the connector configuration file and are not exposed in the Configuration Editor.

These settings allow for customization of:

- **Event Processing Behavior** - Control how access and security events are filtered and handled.
- **PACS-Specific Configuration** - Adjust behavior for supported PACS platforms.
- **Processing Intervals** - Define how frequently events, readers, and health checks are processed.
- **Database Behavior** - Configure storage options and performance characteristics.

When to Use Advanced Configuration

Advanced configuration may be required in the following scenarios:

- High-volume environments requiring performance tuning
- Custom event filtering or processing requirements
- Integration with external systems or monitoring tools
- Troubleshooting complex or environment-specific issues

Configuration Method

Advanced settings are stored in a JSON configuration file located in the installation directory. The connector monitors the configuration file for changes. Updates are applied automatically without requiring a service restart in most cases.

Important Considerations

- Changes to advanced settings can affect connector performance and behavior.
- Settings should be modified only when necessary and with a clear understanding of their impact.
- Default values are appropriate for most deployments.

See Also

- For event processing settings, see Event Processing Settings.
- For PACS-specific options, see PACS-Specific Settings.
- For interval configuration, see Worker Interval Settings.
- For database tuning, see Database Settings (SQL Server).

Event Processing Settings

Event processing settings control how the PACS Connector collects, filters, and forwards access and Security Events to Everbridge Open iPaaS.

These settings are defined in the connector configuration file and are used to customize event behavior beyond the default configuration.

> **IMPORTANT:** Increasing the number or type of events sent to Everbridge may significantly increase event volume and processing load.

Common Event Processing Settings

The following settings are commonly used to control event behavior:

- **SendAllAccessEvents** - Controls whether all Access Events are forwarded to Everbridge
  - **false** (default) - Only Access-granted Events are sent.
  - **true** - All Access Events are sent, including denied or failed attempts.
- **SendSecurityEvents** - Controls whether Security or Hardware Events are forwarded to Everbridge
  - **false** (default) - Security Events are not sent.
  - **true** - Security Events are sent and processed.
- **CardholderIdField** - Specifies the PACS field used to identify the cardholder
  - Must match the identifier configured in Everbridge
  - Value depends on the PACS system
  - Examples:
    - OnGuard:
      - `Badge.<fieldname>` for badge fields
      - `<fieldname>` for cardholder fields
    - C•CURE 9000:
      - Standard personnel or badge field names
- **CacheCardholders** - Controls whether cardholder data is cached locally
  - **true** (default) - Enables caching to improve performance
  - **false** - Disables caching and retrieves data directly from the PACS system

- **EventBatchSize** - Defines the maximum number of events sent to Everbridge in a single request
  - **Default**: 1000
  - Larger values may improve throughput in high-volume environments.
- **GetAllEvents** (C•CURE 9000 only) - Controls whether historical events are retrieved on startup
  - **false** (default) - Processes only new events
  - **true** - Retrieves historical events when the service starts

Configuration Notes

- Default values are appropriate for most deployments.
- Changes to these settings may affect event volume, performance, and processing behavior.
- Increasing event volume may impact:
  - Network usage
  - Processing time
  - iPaaS message throughput

When to Modify These Settings

Modify event processing settings only when necessary, such as:

- Enabling Security Event processing
- Capturing additional Access Event Types
- Tuning performance for high-volume environments
- Troubleshooting event processing issues

See Also

- For Security Event configuration, see Configuring Security Events.
- For identifier configuration, see Everbridge Identifier and Client Identifier.
- For performance tuning, see Worker Interval Settings.

PACS-Specific Settings

PACS-specific settings provide additional configuration options that apply to individual PACS platforms. These settings allow for fine-tuning of event collection and processing behavior.

These options are defined in the connector configuration file and should be modified only when required.

Lenel OnGuard Settings

The following settings apply to Lenel OnGuard integrations:

- **AccessEventFilter** - Defines a custom OpenAccess filter expression for Access Events
  - Used to control which Access Events are collected
  - Supports OpenAccess query syntax
    - Example:
      - `event_type eq 0` or `event_type eq 1`
- **HardwareEventFilter** - Defines a custom OpenAccess filter expression for Hardware or Security Events
  - Used to limit non-access events collected from the PACS system
    - Example:
      - `event_type ne 0` and `event_type ne 1`
- **AccessEventSubscriptionId** - Stores the Subscription ID for Access Events
  - Managed automatically by the connector
  - Allows the connector to resume Event Subscriptions after restart
- **HardwareEventSubscriptionId** - Stores the Subscription ID for Hardware Events
  - Managed automatically by the connector
  - Should not be modified manually

C•CURE 9000 Settings

The following settings apply to C•CURE 9000 integrations:

- **EventsIncluded** - Defines a list of Hardware or Security Event names to be forwarded to Everbridge
  - Supports wildcard patterns
  - Only matching events are processed

    - Example:
        - `["*Alarm*", "Door*", "*Tamper"]`

- **EventQueueDegreesOfParallelism** - Controls the number of parallel threads used for processing events
    - **Default:** 4
    - Higher values increase throughput but may use more system resources.
- **GetAllEvents** - Controls whether historical events are retrieved on startup
    - **false** (default) - Processes only new events
    - **true** - Retrieves historical events

Configuration Notes

- PACS-specific settings should be modified only when required for advanced scenarios.
- Incorrect values may impact event collection or processing.
- Some settings are managed automatically by the connector and should not be changed manually.

When to Use PACS-Specific Settings

These settings may be used in the following scenarios:

- Customizing event collection behavior
- Reducing event volume from the PACS system
- Tuning performance for high-volume environments
- Troubleshooting PACS-specific issues

See Also

- For general event behavior, see Event Processing Settings.
- For performance tuning, see Worker Interval Settings.
- For database configuration, see Database Settings (SQL Server).

Worker Interval Settings

Worker interval settings define how frequently the PACS Connector performs background tasks, such as collecting events, sending data to Everbridge, and updating configuration.

These settings are defined in the connector configuration file and can be adjusted to control processing frequency and system load.

Common Worker Intervals

The connector uses multiple background processes (workers), each with a configurable interval.

- **pacsEventIntervalWorkerConfig** - Controls how often events are collected from the PACS system
  - **Default:** 60 seconds
- **queueToIpaasIntervalWorker** - Controls how often queued events are sent to Everbridge iPaaS
  - **Default:** 60 seconds
- **readerIntervalWorkerConfig** - Controls how often reader data is synchronized
  - **Default:** 60 seconds
- **heartbeatIntervalWorkerConfig** - Controls how often heartbeat messages are sent
  - **Default:** 60 seconds
- **everbridgeConfigWorkerConfig** - Controls how often configuration updates are retrieved from Everbridge
  - **Default:** 60 seconds
- **configIntervalWorkerConfig** - Controls how often the local configuration file is checked for changes
  - **Default:** 60 seconds
- **pacsCacheIntervalWorkerConfig** - Controls how often cached cardholder data is refreshed
  - **Default:** 600 seconds

Configuration Example

Worker intervals are defined in the configuration file using an interval value in seconds.

Example:

```
"readerIntervalWorkerConfig": {
  "IntervalInSeconds": 120
}
```

Configuration Notes

- Default interval values are appropriate for most deployments.
- Changes to interval values affect how frequently tasks are executed.
- Lower interval values:
  - Increase processing frequency
  - May increase system load
- Higher interval values:
  - Reduce system load
  - May introduce delays in processing or updates

When to Modify Worker Intervals

Adjust worker intervals only when necessary, such as:

- Optimizing performance in high-volume environments
- Reducing system resource usage
- Adjusting responsiveness of event processing or synchronization

See Also

- For event processing behavior, see Event Processing Settings.
- For PACS-specific tuning, see PACS-Specific Settings.
- For database configuration, see Database Settings (SQL Server).

Database Settings (SQL Server)

The PACS Connector supports the use of a **SQL Server database** for event storage and queueing in environments that require enhanced performance, scalability, or centralized data management.

SQL Server configuration is optional and should be used only when the default SQLite database does not meet deployment requirements.

> **NOTE:** Ensure the SQL Server instance is accessible from the connector host and that network/firewall rules allow database connectivity.

When to Use SQL Server

SQL Server may be preferred in the following scenarios:

- High-volume event processing environments
- Centralized database management requirements
- Integration with existing database infrastructure
- Enhanced performance and scalability needs

Configuration Fields

The following settings are required when using SQL Server:

- **Using SQL Server** - Enables SQL Server as the event storage mechanism
  - **true** - Use SQL Server
  - **false** - Use the default SQLite database
- **Database Connection String** - Specifies the connection details for the SQL Server instance
  - Required when SQL Server is enabled
  - Must reference an existing database
    - Example:
      - `Server=<server>;Database=<database>;User Id=<user>;Password=<password>;`

Configuration Requirements

- The target SQL Server database must exist prior to configuration.
- The connector automatically creates required tables during initialization.
- The service account must have:
  - Permission to connect to the database
  - Read and write access to the database

Behavior and Considerations

- When SQL Server is used, events are stored in the database prior to transmission to Everbridge.
- This configuration supports reliable event delivery in environments requiring centralized or high-volume data processing.
- Database performance may impact:
  - Event throughput
  - Processing latency

Comparison with SQLite

Feature	SQLite (Default)	SQL Server
Setup	No configuration required	Requires database setup
Deployment	Local file-based	External database
Performance	Suitable for most deployments	Better for high-volume environments
Management	Minimal	Requires database administration

See Also

- For general database configuration, see Database Settings.
- For performance tuning, see Worker Interval Settings.
- For event behavior, see Event Processing Settings.

Troubleshooting

This section provides guidance for identifying and resolving common issues with the PACS Connector.

Troubleshooting typically involves reviewing connector logs, verifying connectivity, and confirming configuration settings.

> **NOTE:** If issues persist after completing these steps, contact Everbridge Support with relevant log files and configuration details.

General Troubleshooting Steps

When diagnosing issues:

- Review connector logs for errors or warnings.
- Confirm that all prerequisite configuration steps have been completed.
- Validate connectivity to both the PACS system and Everbridge iPaaS.
- Test configuration changes using known events.

Service Does Not Start

Possible Causes

- Missing or incorrect configuration
- Insufficient permissions for the service account
- Missing runtime dependencies

Resolution

- Verify that the configuration file is present and valid.
- Confirm that the .NET 8.0 Runtime is installed.
- Ensure the service account has:
  - Permission to access the installation directory
  - Permission to access required system resources
- Review the connector log file for error messages:
  - `C:\ProgramData\Detrios\logs\Detrios.Everbridge.log`

Cannot Connect to PACS

Possible Causes

- Incorrect PACS API URL
- PACS service not running
- Invalid credentials
- Network connectivity issues

Resolution

- Verify the PACS API endpoint is correct and reachable.
- Confirm that the PACS API service is running.
- Validate the username and password.
- Ensure network access between the connector host and the PACS system.

Events Not Appearing in Everbridge

Possible Causes

- Incorrect API key
- Connectivity issues with Everbridge iPaaS
- Event filtering configuration
- Connector service not running

Resolution

- Verify the API key matches an active agent configuration.
- Confirm outbound HTTPS connectivity to the iPaaS endpoint.
- Review event filtering settings to ensure events are included.
- Confirm that the connector service is running.
- Check logs for errors or failed message processing.

Location Updates Not Occurring

Possible Causes

- Identity resolution issues
- Reader not mapped to a location
- Events not processed correctly

Resolution

- Verify that identifier values match between PACS and Everbridge.
- Confirm that the reader is mapped to a valid location.
- Ensure Access Events are being received and processed.
- Review iPaaS activity for message status.

Heartbeat Not Detected

Possible Causes

- Connector service not running
- Network connectivity issues
- Invalid configuration

Resolution

- Confirm that the service is running.
- Verify connectivity to the Everbridge iPaaS endpoint.
- Check logs for communication errors.
- Validate configuration settings.

High Event Volume or Performance Issues

Possible Causes

- Broad event filtering (e.g., *)
- High-frequency processing intervals
- Large number of events from the PACS system

Resolution

- Refine event filtering to include only necessary events.
- Adjust worker interval settings to reduce processing frequency.
- Review event processing settings (such as batch size and caching).

Database Connectivity Issues (SQL Server)

Possible Causes

- Incorrect connection string
- Database not available

- Insufficient permissions

Resolution

- Verify the SQL Server connection string.
- Confirm that the database exists and is accessible.
- Ensure the service account has appropriate database permissions.

Uninstalling the PACS Connector

The PACS Connector can be removed using standard Windows uninstall methods or command-line tools.

Uninstalling Using Windows Settings

To uninstall the PACS Connector using Windows Settings:

1. Open **Windows Settings**.
2. Navigate to **Apps** > **Apps & features**.
3. Locate the **Everbridge PACS Connector**.
4. Select **Uninstall** and follow the prompts.

Command-Line Uninstallation

To uninstall using the command line:

1. Open an elevated command prompt.
2. Navigate to the installation directory.
3. Run:
    - `Detrios.Everbridge.Svc.exe -uninstall`

Optional Cleanup

After uninstalling:

- Delete the installation directory, if required.
- Remove log files from:
    - `C:\ProgramData\Detrios\logs\`

Reference

Default Paths and Endpoints

This section provides reference values for commonly used file paths, service names, and endpoints used by the PACS Connector.

These values may vary by environment but are included here for convenience.

Installation Path

Default installation directory:

- `C:\Program Files (x86)\Everbridge\Everbridge PACS Connector\`

Log File Location

Connector logs are stored in the following directory:

- `C:\ProgramData\Detrios\logs\`
    - **Primary log file:** `Detrios.Everbridge.log`

Service Name

The PACS Connector is installed as a Windows service.

- **Display Name** - Everbridge PACS Connector Service
- The service can be managed through the Windows Services console (services.msc)

Everbridge iPaaS Endpoint

Default Everbridge iPaaS ingestion endpoint:

- `https://ipaas-ingestion.everbridge.net/`

Heartbeat API Endpoint

Endpoint used to retrieve connector heartbeat status:

- `GET https://ipaas-ingestion.everbridge.net/ipaas/v1/agent/heartbeat`

PACS API Endpoints (Examples)

Typical PACS API endpoints include:

- Lenel OnGuard
  - `https://<server>:8080/api/openaccess/`
- C•CURE 9000
  - `http://<server>/victorwebservice/api/`
- C•CURE 9000 Event Endpoint
  - `http://<server>/victorwebservice/signalr`

Configuration File Location

Connector configuration is stored in a JSON file located in the installation directory.

- The file is created during configuration.
- The connector monitors the file for changes and applies updates automatically.