Legal Information

Book Name: FlexNet Edge Server 2021.01 Installation and Operations Guide
Part Number: FNES-2021.01-UG00
Product Release Date: January 2021

Copyright Notice

Copyright © 2021 Flexera Software

This publication contains proprietary and confidential information and creative works owned by Flexera Software and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software is strictly prohibited. Except where expressly provided by Flexera Software in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, must display this notice of copyright and ownership in full.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see https://www.revenera.com/legal/intellectual-property.html. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.
# Contents

1 FlexNet Edge Server 2021.01 Installation and Operations Guide ........................................ 5  
   Product Support Resources ................................................................. 5  
   Contact Us ......................................................................................... 6  

2 About FlexNet Edge ............................................................... 7  
   Introduction to FlexNet Edge .............................................................. 7  
   FlexNet Edge Architecture ................................................................. 7  
   Flow of Requests When Using FlexNet Edge ........................................ 9  
   API Flow to Request an Update from FlexNet Edge ............................... 10  
   Device Status Information .................................................................. 12  
   Obtaining Download Files from FlexNet Edge ...................................... 14  
   Notification Service Base URL ......................................................... 14  

3 Installing FlexNet Edge .................................................... 15  
   Supported Platforms ....................................................................... 15  
   Obtaining the FlexNet Edge ZIP File ................................................. 15  
   Installing FlexNet Edge ................................................................. 16  
      Installing FlexNet Edge on Host Linux Platform ......................... 16  
      Installing FlexNet Edge in a Docker Container ............................ 17  
   Configuring FlexNet Edge to Connect to the Updates and Insights Notification Service ......................................................... 19  
   Authorizing FlexNet Edge ............................................................... 20  
   Custom Branding of FlexNet Edge User Interface ............................. 21  

4 Using FlexNet Edge ...................................................... 23  
   FlexNet Edge User Interface ............................................................ 23  
      Devices Page / Home Page .......................................................... 24  
      File Cache Page ........................................................................... 26
The FlexNet Edge Server is an optional installation, located on the customer’s network, that can communicate with the Updates and Insights Notification Service. Its purpose is to forward requests from devices that have no direct connectivity to the Internet.

- End-user devices send their requests for updates to FlexNet Edge, which queues them until FlexNet Edge can connect to Updates and Insights Notification Service.
- FlexNet Edge will then carry out the requests for updates, and relay the results back to the devices once it has retrieved the required information.

Information on installing and using FlexNet Edge is presented in the following sections:

Table 1-1 • FlexNet Edge Server 2021.01 Installation and Operations Guide

<table>
<thead>
<tr>
<th>Topic</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>About FlexNet Edge</td>
<td>Introduces you to the purpose of FlexNet Edge and explains its workflow.</td>
</tr>
<tr>
<td>Installing FlexNet Edge</td>
<td>Explains how to obtain the FlexNet Edge ZIP file, and how to install and configure FlexNet Edge on a Linux platform or in a Docker container.</td>
</tr>
<tr>
<td>Using FlexNet Edge</td>
<td>Briefly introduces you to the FlexNet Edge user interface, and explains how to access the FlexNet Edge API.</td>
</tr>
</tbody>
</table>

Product Support Resources

The following resources are available to assist you with using this product:

- Revenera Product Documentation
- Revenera Community
- Revenera Learning Center
- Revenera Support
Revenera Product Documentation

You can find documentation for all Revenera products on the Revenera Product Documentation site:

https://docs.revenera.com

Revenera Community

On the Revenera Community site, you can quickly find answers to your questions by searching content from other customers, product experts, and thought leaders. You can also post questions on discussion forums for experts to answer. For each of Revenera’s product solutions, you can access forums, blog posts, and knowledge base articles.

https://community.revenera.com

Revenera Learning Center

Revenera offers a variety of training courses—both instructor-led and online—to help you understand how to quickly get the most out of your Revenera products. The Revenera Learning Center offers free, self-guided, online training classes. You can also choose to participate in structured classroom training delivered as public classes. You can find a complete list of both online content and public instructor-led training in the Learning Center.

https://learning.revenera.com

Revenera Support

For customers who have purchased a maintenance contract for their product(s), you can submit a support case or check the status of an existing case by making selections on the Get Support menu of the Revenera Community.

https://community.revenera.com

Contact Us

Revenera is headquartered in Itasca, Illinois, and has offices worldwide. To contact us or to learn more about our products, visit our website at:

http://www.revenera.com

You can also follow us on social media:

- Twitter
- Facebook
- LinkedIn
- YouTube
- Instagram
About FlexNet Edge

This section introduces you to FlexNet Edge and explains how it works.

- Introduction to FlexNet Edge
- FlexNet Edge Architecture
- Flow of Requests When Using FlexNet Edge
- API Flow to Request an Update from FlexNet Edge
- Device Status Information
- Obtaining Download Files from FlexNet Edge
- Notification Service Base URL

Introduction to FlexNet Edge

The FlexNet Edge Server is an optional installation, located on the customer’s network, that can communicate with the Updates and Insights Notification Service. Its purpose is to forward requests from devices that have no direct connectivity to the Internet.

- End-user devices send their requests for updates to FlexNet Edge, which queues them until FlexNet Edge can connect to Updates and Insights Notification Service.
- FlexNet Edge will then carry out the requests for updates, and relay the results back to the devices once it has retrieved the required information.

FlexNet Edge only supports communication via REST to the Notification Service. Devices will use the same REST APIs for updates whether they are communicating to the Notification Service directly or via FlexNet Edge.

Note • For information about the base URLs for the Notification Service and FlexNet Edge, see Anonymous Devices.

Devices communicating with FlexNet Edge must be registered with FlexNet Operations. Registration can be done manually through the FlexNet Operations Producer Portal, through a device licensing capability request (if licensing is being used), or using the /registrations REST API to the Notification Server.

FlexNet Edge Architecture

FlexNet Edge is an on-premises Linux-based server deployed by the end customer for the purpose of supplying updates to devices within a private network where devices will not directly connect to FlexNet Operations (referred to as the back-office).
FlexNet Edge (referred to as Edge), is part of the Updates and Insights module for FlexNet Operations.

The following diagram provides an overview of how FlexNet Edge and FlexNet Operations interact.

![Updates and Insights Architecture Diagram]

Figure 2-1: Updates and Insights Architecture

Updates and Insights functionality is accessed through a REST API provided by the Notification Service of FlexNet Operations. FlexNet Edge mimics this REST API, but all requests it receives are queued until it is able to forward them to the Notification Service.

Once FlexNet Edge receives a response from the Notification Service, it stores the response, ready to send to the device the next time the device polls for completion of its request.

If Edge cannot establish a connection to the Notification Service, it will attempt repeatedly at certain intervals, until all pending requests have been fulfilled. The frequency with which Edge connects to the Notification Service could be daily, weekly or even one or more months.

Once Edge has completely dealt with a request from a device, it will be able to respond to the device with a list of updates.

- This list might be empty, which means there are no updates for the device to apply.

- If the list is not empty, then the device has to make a decision about what to do next - it must select an update to apply, request the update files from Edge, and use those files to achieve the selected software upgrade.
Flow of Requests When Using FlexNet Edge

FlexNet Edge, located on the customer’s network, is used in implementations where end-user devices have no direct connectivity to the Internet. Devices send their requests to FlexNet Edge, which communicates with the Notification Service.

Devices communicating with FlexNet Edge must be registered with FlexNet Operations. Registration can be done manually through the Producer Portal or through a device licensing capability request (if licensing is being used). Alternatively, devices can be registered through FlexNet Edge Server using the /registrations REST API.

The workflow uses the /statusLog and /updates APIs, as described in Status Log and Updates, respectively.

**Important** • FlexNet Edge must be authorized to communicate with the Notifications Server. For more information, see Authorizing FlexNet Edge.

**Figure 2-2**: FlexNet Edge Update Workflow Diagram

The following table describes the FlexNet Edge Update Workflow diagram.

**Table 2-1** • FlexNet Edge Update Workflow

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Device requests update</td>
<td>A device requests update information, using /uai/2.0/updates.</td>
</tr>
<tr>
<td>2</td>
<td>FlexNet Edge returns polling ID</td>
<td>FlexNet Edge returns a polling ID so that the device can periodically poll for the requested information.</td>
</tr>
<tr>
<td>3</td>
<td>Device polls for update information</td>
<td>The device begins to poll for the update information using /uai/2.0/updates/{pollingid}.</td>
</tr>
</tbody>
</table>
At all stages, the device can make status log updates to FlexNet Edge, which will synchronize the status log updates to FlexNet Operations via the Notification Service.

### API Flow to Request an Update from FlexNet Edge

The steps to acquire an update are the same whether the device is communicating to the Notification Service directly or through FlexNet Edge, and are documented at the following swagger link:

#### Table 2-1 • FlexNet Edge Update Workflow

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4 | FlexNet Edge requests and polls for update information | As soon as FlexNet Edge can connect to the Notification Service, it begins to request and then poll for update information. 

*Note* • FlexNet Edge activity can be paused by the end user, if there is a period during which devices should not update. During this pause, FlexNet Edge will not connect to the Notification Service.  

| 5 | FlexNet Edge requests manifests for each update | FlexNet Edge requests all update available information from the Notification Service, and once this has been returned, FlexNet Edge requests the manifests for each available update.  

| 6 | FlexNet Edge obtains download files needed for each update | FlexNet Edge next obtains all download files needed for each update from the appropriate file delivery service.  

*Note* • FlexNet Edge caches all updates, manifests and download files. Any other devices requesting the same update information will be responded to immediately from that cache. By default, the cache is cleared once every 30 days, but this is user configurable.  

| 7 | FlexNet Edge returns update information to the device | Finally, FlexNet Edge returns the update information to the device. The device must then request:  

1. The manifest for its chosen update using the APIs described in Request Manifest File.  

2. The necessary files using a FlexNet Edge API described in Obtaining Download Files from FlexNet Edge.  

At all stages, the device can make status log updates to FlexNet Edge, which will synchronize the status log updates to FlexNet Operations via the Notification Service.
General flow for successfully receiving an update is outlined using the following APIs:

Table 2-2 • Requesting an Update from FlexNet Edge

<table>
<thead>
<tr>
<th>Description</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device requests an update</td>
<td>POST /uai/&lt;version&gt;/updates</td>
</tr>
<tr>
<td>Device checks for response</td>
<td>GET /uai/&lt;version&gt;/updates/&lt;polling id&gt;</td>
</tr>
<tr>
<td>Device requests manifest with details of the files</td>
<td>POST /uai/&lt;version&gt;/updates/manifests</td>
</tr>
</tbody>
</table>

During a request for an update from FlexNet Edge, the following occurs:

1. Device requests an update—Device sends initial request to Edge via the /uai/<version>/updates endpoint. Once Edge receives this request, it will immediately respond back to the client with a polling ID it can use to periodically check the status of its request.

2. Edge forwards request to the Notification Service—Edge will forward the request to the Notification Service (unless the update is already cached) via the same REST endpoint /uai/<version>/updates, awaiting a polling ID from the Notification Service.

3. Edge calls to retrieve information about the update (manifest UUID)—Once Edge has the polling ID, it will make a call back to the /uai/<version>/updates/<polling id> endpoint to retrieve information about the update that was authored in the back-office by the supplier. One of the key items in the update is the manifest UUID, which it will use to retrieve the download package contents.

4. Edge calls the manifests endpoint to get the contents of the update—Edge will make a call back to the /uai/<version>/updates/manifests endpoint, to get the contents of the updates which include the URLs to the files to be downloaded by the client, SHA2 and MD5 checksums for integrity checking, and an option whether to execute the file or not.

5. Edge downloads all files in manifest response—Edge will then download all the files in the manifest response and store them in its local cache once it has validated the specific hashes.
   - If the checksums are not valid, Edge will attempt the download process again. This could happen if there is an error in transmission due to network drop, for example.
   - The files to be downloaded can be hosted in FlexNet Operations ESD module or could be externally hosted by the supplier.

6. Edge responds back to the device—Once all the files have been downloaded and validated, Edge is ready to respond back to the device upon the next callback request. It will return a 200 HTTP code, and the results of the original update request (no update or update with information including manifest UUID to extract the download package information).

The following is a diagram of the update request flow.
Figure 2-3: Update Request Flow

Note the following regarding the update request flow:

- Devices should use the `/uai/2.0/registrations` API in the Edge Server to register themselves with FlexNet Operations before carrying out the above requests.

- The cache cannot be disabled. This means that if the entitlement check is enabled, the first device calling in will go through the entitlement check, and if successful then the update will be delivered to Edge, and cached. However, the next device’s request for updates will be fulfilled directly from the cache, with no further requests going through the entitlement checks in the back office.

For account entitlement checks, it is assumed that all devices interacting with Edge belong to the same account and so are identically entitled. However, for device based entitlements, this workflow mandates the device would always have to call into the back office as Edge has no notion of Accounts or Entitlements. Currently disabling the cache to force this call to the back-office every time is not possible.

Device Status Information

During the update process, a device should send status through the `/uai/<version>/statusLog` endpoint so that information can be recorded in FlexNet Operations to provide insights through the portal UI and through device extracts. This is how the supplier can ensure information about an end customer’s disconnected devices is logged in FlexNet Operations.

Because the status update has been sent via FlexNet Edge, the last status of the device can all be viewed in FlexNet Edge user interface so that an end user can determine the current state of the request flow and eventually the product version installed on the device.

The application can determine how frequently a status update should be sent. It is recommended to send the status update as the device progresses through the update.
The current available status are as follows:

### Table 2-3 • Device Statuses

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Delivered</td>
<td>The response has been seen by the device. The device can choose either to respond to the update by downloading it, or ignore the update.</td>
</tr>
<tr>
<td>Download Started</td>
<td>Once the download process has been started, the device can notify Edge of the status.</td>
</tr>
<tr>
<td>Download Failed</td>
<td>The device can send this status for a number of reasons, such as network outage, invalid checksum, or failed download.</td>
</tr>
<tr>
<td>Download Completed</td>
<td>This is typically sent after all the contents of the update have been downloaded.</td>
</tr>
<tr>
<td>Installation Started</td>
<td>Installation process is independent of the Updates and Insights architecture, but the device can still record this as an event.</td>
</tr>
<tr>
<td>Installation Completed</td>
<td>Once the update has been successfully applied, this status gives visibility into the software running on the device.</td>
</tr>
</tbody>
</table>

The following is a diagram listing the device statuses:

![Diagram](image)

**Figure 2-4:** Device Sending Update Status
Obtaining Download Files from FlexNet Edge

A device communicating with FlexNet Edge must request all download files from FlexNet Edge, because it has no direct access to the Internet. To do this, the device must:

1. Extract the URL for each download file from the manifest.
2. Calculate the MD5 sum of the file URL.
3. Request the file from FlexNet Edge using the REST API /uai/2.0/updates/manifests/manifest_files/{id}.

Notification Service Base URL

The Notification Service is the service within FlexNet Operations that handles update requests from authorized devices or, if used, the FlexNet Edge Server. The Notification Service communicates with the other FlexNet Operations back offices services to retrieve and deliver update information to the requesting device or FlexNet Edge.

Notification Service Base URL

The REST API paths for the Notification Service that are described in this section must be prefixed with the following base URL:

https://siteID-ns-uat.flexnetoperations.com/

where siteID is your organization’s specific site ID supplied by Flexera.

Note • The URL above points to a User Acceptance Test (UAT) environment indicated by the -uat part following the siteID. For production environments, the -uat is omitted.
Installing FlexNet Edge

This section explains how to install and configure FlexNet Edge.

- **Supported Platforms**
- **Obtaining the FlexNet Edge ZIP File**
- **Installing FlexNet Edge**
- **Configuring FlexNet Edge to Connect to the Updates and Insights Notification Service**
- **Authorizing FlexNet Edge**
- **Custom Branding of FlexNet Edge User Interface**

**Supported Platforms**

Currently, FlexNet Edge can only be installed and run on Linux platforms, either directly on the host platform or as a Docker container on the host.

**Obtaining the FlexNet Edge ZIP File**

FlexNet Edge is released as a ZIP file which can be obtained from the Flexera Product and License Center.

<table>
<thead>
<tr>
<th>Task</th>
<th>To obtain the FlexNet Edge ZIP file:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Login to the Flexera Product and License Center using your Flexera Community credentials.</td>
</tr>
<tr>
<td>3.</td>
<td>Click on FlexNet Operations Cloud Documentation. The Product Download page opens, listing files available for download.</td>
</tr>
</tbody>
</table>
Installing FlexNet Edge

You can install FlexNet Edge either directly on the host Linux platform or in a Docker container on the host platform.

- Installing FlexNet Edge on Host Linux Platform
- Installing FlexNet Edge in a Docker Container

Installing FlexNet Edge on Host Linux Platform

To install FlexNet Edge directly a host Linux platform, perform the following steps.

<table>
<thead>
<tr>
<th>Task</th>
<th>To install FlexNet Edge directly on a host Linux platform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>From the downloaded ZIP file obtained in Obtaining the FlexNet Edge ZIP File, extract the edge-flexera-&lt;version&gt;-&lt;date&gt;.run and install.sh files.</td>
</tr>
<tr>
<td>2.</td>
<td>Copy both files to the /opt directory.</td>
</tr>
<tr>
<td>3.</td>
<td>Make both files executable.</td>
</tr>
<tr>
<td>4.</td>
<td>Change your directory to /opt and run the install script using the following command: run ./install.sh edge.run</td>
</tr>
</tbody>
</table>

Starting and Stopping FlexNet Edge on a Linux Platform

To start or stop FlexNet Edge on a Linux platform, perform the following steps.

<table>
<thead>
<tr>
<th>Task</th>
<th>To start or stop FlexNet Edge on a Linux platform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>To start FlexNet Edge, execute the following command: /etc/init.d/edge start</td>
</tr>
<tr>
<td>2.</td>
<td>To stop FlexNet Edge, execute the following command: /etc/init.d/edge stop</td>
</tr>
</tbody>
</table>

Location of Log Files

After running FlexNet Edge once, log files are available in the following directory for debugging: /opt/edge/logs
Installing FlexNet Edge in a Docker Container

Information about installing FlexNet Edge in a Docker container is presented in the following sections:

- Installing in a Docker Container
- Starting or Stopping FlexNet Edge Server in a Docker Container
- Cleaning Up a Docker Container
- Getting a Command Shell in a Docker Container

Installing in a Docker Container

To install FlexNet Edge in a Docker container, perform the following steps.

<table>
<thead>
<tr>
<th>Task</th>
<th>To install FlexNet Edge in a Docker container:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Obtain the FlexNet Edge ZIP file, as described in Obtaining the FlexNet Edge ZIP File.</td>
</tr>
<tr>
<td>2.</td>
<td>Unzip the ZIP file. The following contents will be listed:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf</td>
<td>07/05/2020 13:08</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>theme</td>
<td>07/05/2020 13:08</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>Dockerfile</td>
<td>03/05/2020 13:08</td>
<td>File</td>
<td>1 KB</td>
</tr>
<tr>
<td>edge-run</td>
<td>24/03/2020 13:08</td>
<td>RUN File</td>
<td>12,200 KB</td>
</tr>
<tr>
<td>install.sh</td>
<td>12/12/2019 16:34</td>
<td>SH File</td>
<td>5 KB</td>
</tr>
</tbody>
</table>
3. Create the following Docker file:

```bash
FROM ubuntu:18.04
COPY . opt
WORKDIR /opt/edge
RUN 
  cd /opt && 
  chmod u+x install.sh && 
  chmod u+x edge.run
EXPOSE 8080 8888 9000
RUN 
  cd /opt && 
  echo "\nPrint working directory: " && 
  pwd && 
  echo "\nList of contents of /opt directory: " && 
  ls -l /opt && 
  echo "\nInstalling Edge server..." && 
  ./install.sh && 
  echo "\nList of contents of /opt/edge directory: " && 
  ls -l /opt/edge
# A long running command which has Docker keep the image from quitting after installation is complete
CMD tail -f /dev/null
```

4. Build and run the Docker container using the following commands:

```bash
docker build -t edge_image .
docker run --name edge_container -p 8080:8080 -p 8888:8888 -p 9000:9000 -d  edge_image
```

### Starting or Stopping FlexNet Edge Server in a Docker Container

Run the following from the command line to start or stop the FlexNet Edge Server in a Docker container:

```bash
docker exec edge_container sh -c "/etc/init.d/edge start"
docker exec edge_container -c "/etc/init.d/edge stop"
docker exec -it edge_container /bin/bash
```

### Cleaning Up a Docker Container

Run the following from the command line to stop FlexNet Edge and clean up the Docker container:

```bash
docker stop edge_container
docker rm edge_container
docker rmi edge_image
docker rmi $(docker images --filter "dangling=true" -q --no-trunc)
```

### Getting a Command Shell in a Docker Container

Run the following from the command line to get to a command shell while the Docker container is running:

```bash
docker exec -it flexera_demo /bin/bash
```
Configuring FlexNet Edge to Connect to the Updates and Insights Notification Service

To configure FlexNet Edge to connect to Updates and Insights Notification Service, perform the following steps.

**Task** To configure FlexNet Edge to connect to the Updates and Insights Notification Service:

1. Locate the flexera.json file in the `/opt/edge/config` directory:

   ```json
   {
     "fno_protocol": "https",
     "fno_host": "se02-ns-ci.flexnetoperations.com",
     "fno_port": "443",
     "fno_base_url": "/uai/2.0",
     "authorization_header_template": "Bearer %s",
     "signed_update_request_url": "/signed-updates",
     "update_request_url": "/updates",
     "signed_manifest_request_url": "/signed-updates/manifests",
     "manifest_request_url": "/updates/manifests",
     "status_log_url": "/updates/statusLog",
     "manifest_filestore_path": "flexera_tmp",
     "manifest_filestore_ttl_seconds": 2592000,
     "manifest_filestore_purge_job_queue_config": {
       "max_queue_size": 10,
       "burst_buffer_size": 0,
       "num_workers": 1,
       "rate_limit_seconds": 3600
     },
     "update_request_job_queue_config": {
       "max_queue_size": 10,
       "burst_buffer_size": 0,
       "num_workers": 1,
       "rate_limit_seconds": 5
     },
     "update_item_job_queue_config": {
       "max_queue_size": 10,
       "burst_buffer_size": 0,
       "num_workers": 1,
       "rate_limit_seconds": 5
     },
     "status_log_job_queue_config": {
       "max_queue_size": 10,
       "burst_buffer_size": 0,
       "num_workers": 1,
       "rate_limit_seconds": 5
     }
   }
   ```

2. Set `fno_host` to the host providing Update and Insights information.

3. Set `fno_port` to the correct port on the specified `fno_host`. 
4. Set `fno_protocol` to `http` or `https`, depending on the desired protocol. The REST interface for the Notification Service supports both.

5. Restart FlexNet Edge for these changes to take effect.

**Authorizing FlexNet Edge**

FlexNet Edge must be authorized before it will be able to communicate with the Updates and Insights Notifications Service. To enable JWT (JSON Web Tokens) authorization, perform the following steps.

**Task**  
**To authorize FlexNet Edge:**

1. Request a public key for checking signed update responses.
   - While the device might not request signed updates, FlexNet Edge will always use the `/signed-updates` API, and requesting the public key will create the private key in FlexNet Operations.
   - The public key must be made available to any devices requesting signed updates, but FlexNet Edge does not need the key to be loaded in its UI or install directories.

2. Create a JSON Web Token and install it in FlexNet Edge.

   **Important** • All requests to the Notification Service, from devices or from FlexNet Edge, must be authorized using a JWT and so FlexNet Edge must be provided with this token.

   There are three ways to provide FlexNet Edge with the token:
   - Upload the JWT file on the **Settings** page of the FlexNet Edge user interface.
   - Place the JWT file in the following directory on the FlexNet Edge server’s operating system:
     `/opt/edge/conf/authorization-token`
   - Upload the JWT file using its `/authorization-token` REST API.

   **Important** • If you are manually setting the authorization token in the `authorization-token` file, make sure that a new line character `\n` is not accidentally introduced into the file. If a new line character exists in this file, it will cause the requests against the Edge Server to fail due to “invalid authorization header”.

3. Stop and then restart FlexNet Edge in order for the change to take effect.
Custom Branding of FlexNet Edge User Interface

To include any special branding of the FlexNet Edge UI, perform the following steps.

**Important** • Custom branding should be done before FlexNet Edge is installed so that the installer scripts will pull this information during installation.

<table>
<thead>
<tr>
<th>Task</th>
<th>To perform custom branding of the FlexNet Edge user interface:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Unzip the ZIP file downloaded in Obtaining the FlexNet Edge ZIP File and open the <code>/opt/theme</code> directory and locate the following files:</td>
</tr>
<tr>
<td></td>
<td>logo.png</td>
</tr>
<tr>
<td></td>
<td>brand.css</td>
</tr>
<tr>
<td>2.</td>
<td>To change the logo that appears at the top left of the FlexNet user interface, replace the logo.png file with a customized logo file of the same name and size.</td>
</tr>
<tr>
<td>3.</td>
<td>To adjust the look and feel of the FlexNet Edge user interface, modify the brand.css file.</td>
</tr>
</tbody>
</table>

**Note** • Any custom CSS that you add to brand.css will override the settings in the main app.css file used by the site.

4. Install FlexNet Edge, as described in Installing FlexNet Edge. After installation, these custom branded files will be found in the following directory:

`/opt/edge/www_flexera/theme`
Using FlexNet Edge

This section briefly introduces you to the FlexNet Edge user interface, and explains how to access the FlexNet Edge API documentation.

- FlexNet Edge User Interface
- FlexNet Edge REST API

FlexNet Edge User Interface

The FlexNet Edge User Interface provides insights to end users on what devices have been requesting updates, and the status of the latest request. Using the FlexNet Edge user interface, users can view information about devices in the network and the versions those devices have updated to. In addition, using this interface, the end user can create groups of devices and schedule the update frequency for the group (or turn off updates for the group).

Note • FlexNet Edge does not display the complete history of the request (like FlexNet Operations does), but gives the current state of the device to end customers. Any status log event sent to FlexNet Edge will be forwarded to FlexNet Operations, so that the entire history can be maintained there, to provide the supplier with a complete view of the device update history to use for reporting.

The following sections describe the pages in the FlexNet Edge user interface:

- Devices Page / Home Page
- File Cache Page
- Settings Page
- Device Groups Page
- Update Requests Page
Devices Page / Home Page

The Devices page (also the Home page) shows the latest status of each unique device request.

![FlexNet Edge Devices Page / Home Page](image)

**Figure 4-1**: FlexNet Edge Devices Page / Home Page

The navigation menu on the left provide access to the **Device Groups, Update Requests, File Cache, and Settings** pages.

- Information Displayed in the Devices List
- Storage of Device Information
- List Controls

### Information Displayed in the Devices List

For each request in the **Devices** list, the following information is displayed:

**Table 4-1 • Device Information**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device ID</strong></td>
<td>A unique identifier for a device, as registered in FlexNet Operations. The supplier’s application must determine the unique properties of the device and use them to create an identifier in communication with FlexNet Edge and FlexNet Operations.</td>
</tr>
<tr>
<td><strong>Version ID</strong></td>
<td>The download package version being delivered to the device as part of the update process. Version ID is a term that is meaningful to the end user; FlexNet Edge is not exposing FlexNet Operations terminology to the end user.</td>
</tr>
<tr>
<td><strong>Update Name</strong></td>
<td>A name recognizable to an end user, given by the supplier to an update in FlexNet Operations when the update is created.</td>
</tr>
<tr>
<td><strong>Platform</strong></td>
<td>The platform that the device runs on. This field can list standard operating system platforms, or platforms defined by the supplier.</td>
</tr>
</tbody>
</table>
The following parameters make a request unique in FlexNet Edge:

- DeviceID
- PackageID
- Platform
- Language

If the same Device ID requests a different package ID or platform, these will show up as separate items on the list on the Devices page.

**Note** • Currently, FlexNet Edge does not show Language in the Devices page list view.

### Storage of Device Information

The device information is stored by FlexNet Edge in a LedisDB and remains indefinitely. The only purge of the cache that is allowed are the files that are downloaded as part of the update. Currently, there is no way to clear the device status in that database.

### List Controls

The list on the Devices page (as well as other FlexNet Edge pages) includes controls that provide robust search functionality, as well as sorting and pagination features.

![Device List Controls](image)

**Figure 4-2:** Device List Controls
The following controls are available:

- **Pagination**—This list has an end-user-controlled pagination capability, allowing you to select the number of records displayed per page, and giving you arrows to jump back and forth in the page listing.

- **Searching**—This list has search capability across all the device data known to Edge. All columns can be selected in the search drop down for searching, with **Device ID** being the default. The search box can be used for a simple text search or an advanced search, using regular expressions to search large sets of data.

*Note* • In order for FlexNet Edge to know about a device, a status event has to have been sent first to FlexNet Edge.

- **Sorting**—Each column has a sorting control, with the default sort being set to the **Received** column.

### File Cache Page

On the **File Cache** page, you can clear the cache of updates, so that device requests will go to the back-office. The **File Cache** page lists the current files in the cache.

By default, the cache will be automatically purged every 30 days. This default value is specified in the **Cache Purge Period** field on the **Settings Page**. However, if you want to clear the cache on-demand, just click the **Purge Cache** button.

If the supplier is frequently publishing updates, the FlexNet Edge cache should be periodically cleared to see if any new updates exist, which it can then cache for further device requests.

*Figure 4-3:* File Cache Page
Settings Page

On the Settings page, you can perform the following tasks:

- Setting the Cache Purge Period
- Entering an Authorization Token

**Figure 4-4: Settings Page**

**Setting the Cache Purge Period**

On the Settings page, you can change the Cache Purge Period setting from the default value, 30 days, to whatever you desire as minimum number of days to purge the cache. Enter a new value in the field and then click Save.

**Entering an Authorization Token**

FlexNet Edge must be authorized before it will be able to communicate with the Updates and Insights Notification Service. To authorize FlexNet Edge, enter a JWT (JSON Web Token) into the Upload New Authorization Token field. For more information see Authorizing FlexNet Edge.
Device Groups Page

On the **Device Groups** page, you can group devices so that you can schedule updates (or disable them) for an entire group. You can specify that a group of devices only be updated on certain days or certain times of the month, or completely disable a group so that those devices receive no updates until manually resumed.

![Device Groups Page](image)

**Figure 4-5:** Device Groups Page

The **Device Groups** page lists all existing device groups. To create a new device group, click **Create Group**. To delete a device group, click **Delete**.

- Creating a New Device Group
- Editing an Existing Device Group
Creating a New Device Group

To create a new device group, perform the following steps.

**Task**

1. On the **Devices** page, click **Create Group**. The **Group Details** view opens.

2. In the **Group Name** field, enter a name to identify the group.
3. Under **Update Settings**, select one of the following options:

- **Do not allow updates**—Do not permit any of the devices in this group to be updated.
- **Allow updates at any time**—Allow the devices in this group to be updated at any time.
- **Only allow updates as scheduled**—Allow the devices in this group to be updated only at the frequency and days specified. You can choose **Every week** or **Every other week** as the **Frequency**, and you can select the **Days** of the week when updates are permitted to occur:

![Update Settings Image]

4. To add devices to this group, click **Edit Devices**. The device list is populated with all devices that do not already belong to a device group.

![Device Groups Image]
Note • When you click **Edit Devices**, the name of that button switches to **Save Devices**.

5. Select the devices that you want to add to the group. You can use the List Controls to perform searches to locate devices.

6. After you have selected all of the devices, click **Save Devices**. The selected devices are now listed.

7. Click **Save** to save your new device group. It will now be listed on the Devices page.

**Editing an Existing Device Group**

To edit an existing device group, perform the following steps.

**Task**

**To edit an existing device group:**

1. Open the **Devices** page.

2. Click on the name of the device group that you want to edit. The Group Details view opens.

3. Make edits to the **Group Name**, **Update Settings**, and/or **Devices in Group**, as described in Creating a New Device Group.

4. Click **Save**.
Update Requests Page

The **Update Requests** page lists all update requests on the FlexNet Edge server. You can perform a search to locate a particular update. You can also click **Remove** to remove an update from the list.

![Update Requests Page](image)

**Figure 4-6:** Update Requests Page

**FlexNet Edge REST API**

The FlexNet Edge REST APIs enable you to integrate FlexNet Edge capabilities into your application or to build your own user interface for Edge.

The FlexNet Edge Server 2020 R2 API Reference Guide describes this REST API. You can access this API reference guide on the [docs.flexera.com](http://docs.flexera.com) site:

- FlexNet Edge Server API Reference Guide (HTML)
- FlexNet Edge Server API Reference Guide (PDF)

**Note** • A set of REST services for the Updates and Insights Notification Service can be found at the following URL of your FlexNet Operations instance:

https://<siteID>.flexnetoperations.com/flexnet/swagger-ui.html#/  

These APIs are used by client applications to request and receive updates from FlexNet Edge. For more information see Getting Started with Updates and Insights.

**FlexNet Edge Base URL**

The REST endpoint base URL for the FlexNet Edge Server includes the server’s host name and port number in the format `EdgeServerHostName:port`.

**Note** • The Edge Server exposes ports 8080, 8888, and 9000. For detailed information on the ports supported by FlexNet Edge Server, see [Supported Ports](#) in the FlexNet Edge Server API Reference Guide.

The host name can be either a network address (such as `myserver.example.com`) or an IP address, followed by a port number such as:

- `https://myserver.example.com:8080`
- `http://111.2.3.44:8080/`
**Note** • Currently, the FlexNet Edge Server only supports HTTP from the client.
Chapter 4 Using FlexNet Edge

FlexNet Edge REST API