

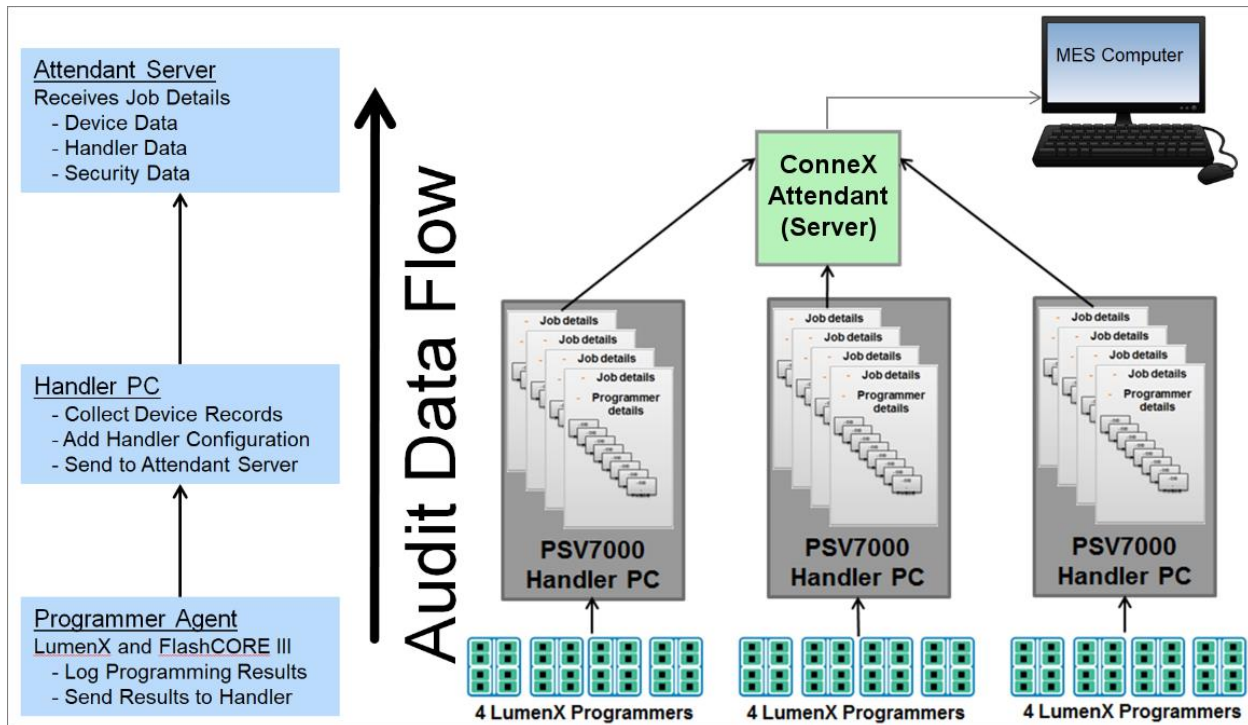


Getting Started with ConneXTM

Use this document as a companion guide as you tour the ConneX product and get an overview of its functions and features. This document provides an end-to-end walkthrough of ConneX and assumes you have experience running jobs in the underlying LumenX/FlashCORE III programming technologies.

Chapter 1: Installation.....	3
Verify the Installation.....	7
Chapter 2: Configuration	9
Configure ConneX Handler	9
Configure ConneX Attendant	14
Chapter 3: Verification.....	16
Validate Data Logging.....	18
Validate Data Exports	21
Chapter 4: Customization	28
Create a New Template	30
Customize the New Template	32
View Customized Data Output	43
Chapter 5: License Activation/Upgrading.....	50
Appendix A: Audit Record Details/Event Reference	57
LumenX Programming Details.....	57
FlashCORE III Programming Details.....	59

ConneX is a web-based application that provides event logging and manufacturing traceability for your programming parts and systems. You install ConneX Handler on each "agent" computer (such as a Handler PC running LumenX jobs) so that programmer, device/part, and job details are collected as audit records. Then you install ConneX Attendant on a server that receives these audit records (from each agent/Handler) for centralized monitoring, management, and analysis.



To establish an audit record of programming, device/part, and job details for traceability, ConneX performs data collection at the programmer, Handler PC, and Attendant Server levels. To protect the security and integrity of the audit trail, ConneX disallows any writing, updating, or removal of existing records in its encrypted database. You cannot extract data from the Programmer Agent nor ConneX Handler; only through ConneX Attendant can you access the logged data as XML files that ConneX Attendant saves with each programming job.

While you can view limited data in real-time, the core functionality of ConneX is all about secure and consistent **data collection**. ConneX does NOT provide data visualization features but allows you to customize the XML data output (fields and format) to align with the data import process in your particular environment and systems, where you run the reporting and analytics.

Chapter 1: Installation

In this opening Chapter, we'll cover some basic requirements before running the ConneX Setup Wizard. Then we'll verify the installation by opening ConneX in a browser window.

Before installing ConneX, make sure the following requirements are met.

Hardware Requirements

- LumenX Desktop Programmer/Programming Unit, or
- LumenX Programmer/Programming Unit in a PSV5000/PSV7000 System
- One or more test devices/microchips for programming

Follow the “*Getting Started Guide for LumenX*” to connect the LumenX programmer to the computer. Pay careful attention to the Electrostatic Discharge (ESD) safety precautions and Shock Hazard warnings to prevent personal injury and equipment/device damage. After establishing connectivity between the LumenX programmer and the computer, we're ready for software installation.

Software Requirements

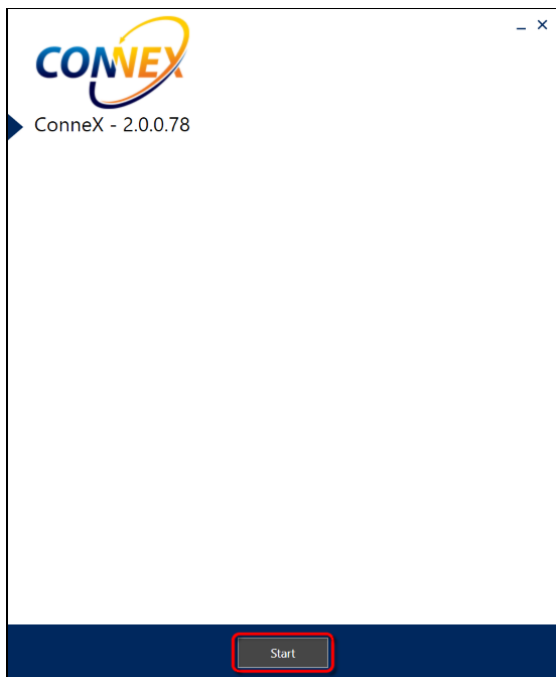
- Windows 7 (32/64-bit) or Windows 10 (32/64-bit)
- LumenX™ Data Management Suite (DMS) version 1.4.1 or above
<http://www.dataio.com/Technology/LumenX>
- Change the “FIS-PS=false” setting to **True** in your C:\AH700\WinAH400.ini file

The ConneX Setup Wizard detects and installs any software prerequisites needed. For the simplicity of this tutorial, we'll install ConneX Handler and ConneX Attendant on the same computer.

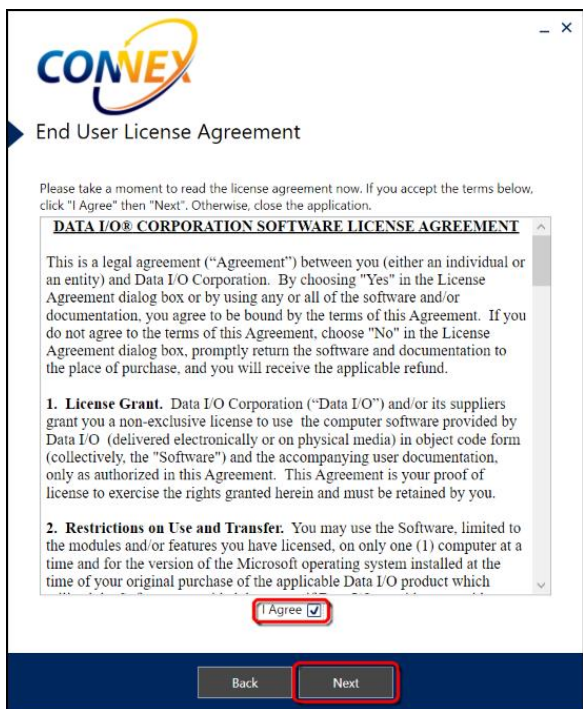
Run the Setup Wizard

Let's complete the following steps to install the ConneX software.

1. Double-click the ConneX Installer (for example, the **setup.exe** for version 2.0.0.78 shown here).
2. On the ConneX Setup Wizard start screen, let's click **Start** at the bottom.

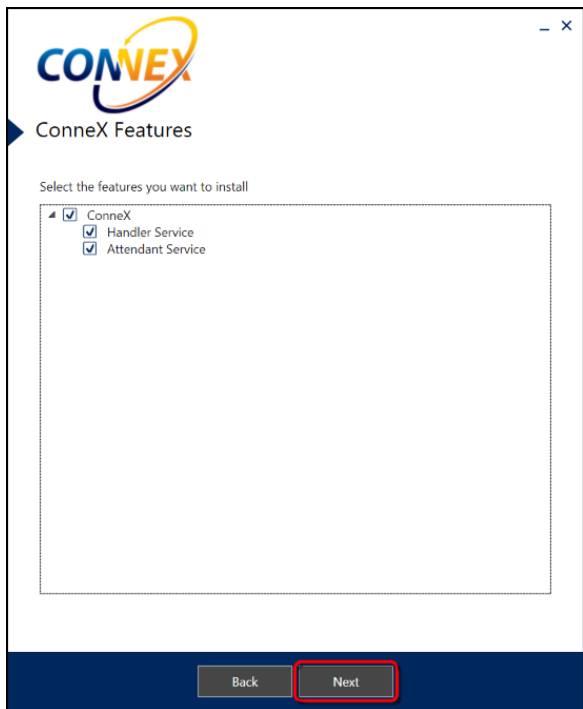


3. On the **End User License Agreement** page, to agree with the terms and conditions, check the box for **I Agree**, and then click **Next**.



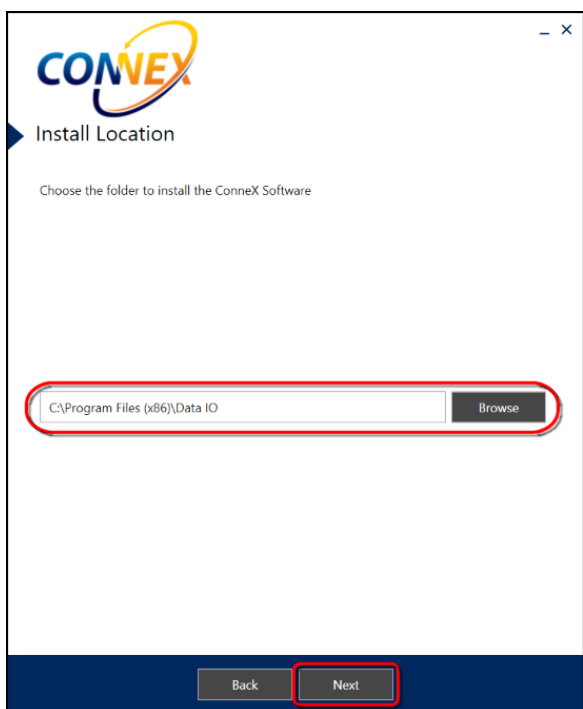
The screenshot shows a window titled "End User License Agreement" with the ConnEX logo at the top. Below the logo, there is a paragraph of text explaining the license agreement. A scrollable area contains the "DATA I/O CORPORATION SOFTWARE LICENSE AGREEMENT" text, which includes sections on "License Grant" and "Restrictions on Use and Transfer". At the bottom of the scrollable area, there is a checkbox labeled "I Agree" which is checked. Below the scrollable area, there are two buttons: "Back" and "Next". The "Next" button is highlighted with a red rectangle.

4. On the **ConnEX Features** page, confirm all features are selected, and then click **Next**.

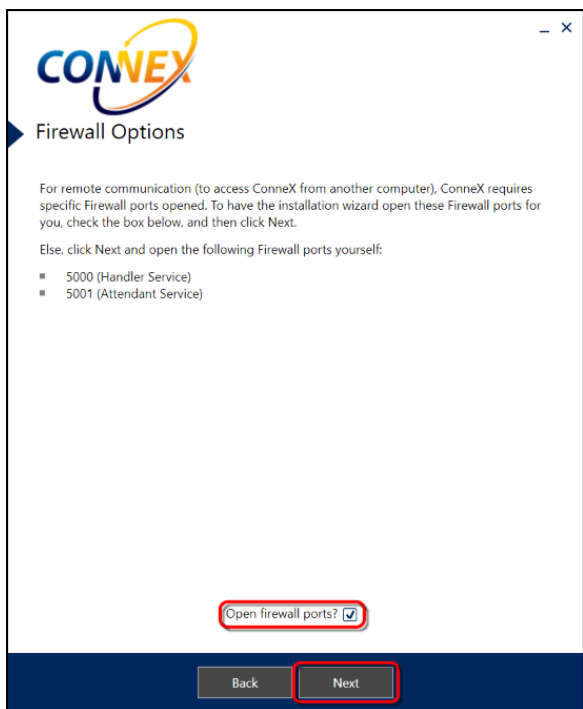


The screenshot shows a window titled "ConnEX Features" with the ConnEX logo at the top. Below the logo, there is a section titled "Select the features you want to install". Inside this section, there is a list of features with checkboxes: "ConnEX", "Handler Service", and "Attendant Service". All three checkboxes are checked. Below the list, there are two buttons: "Back" and "Next". The "Next" button is highlighted with a red rectangle.

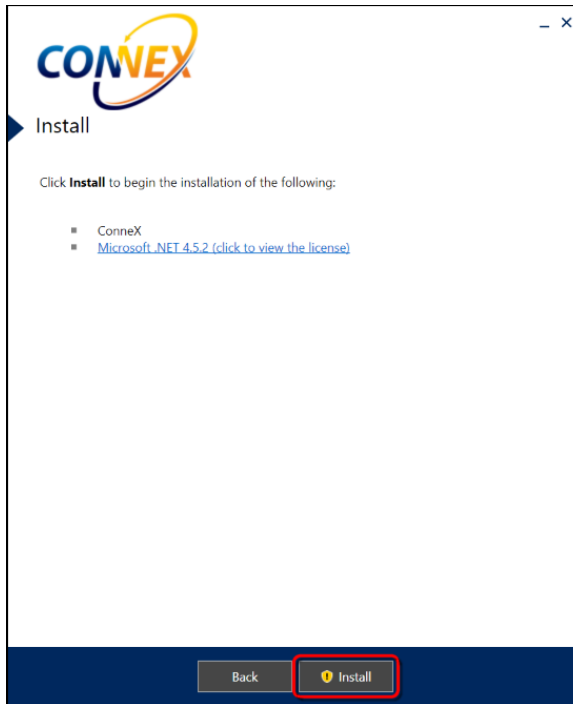
5. On the **Install Location** page, confirm the installation location (or click **Browse** to select another folder), and then click **Next**.



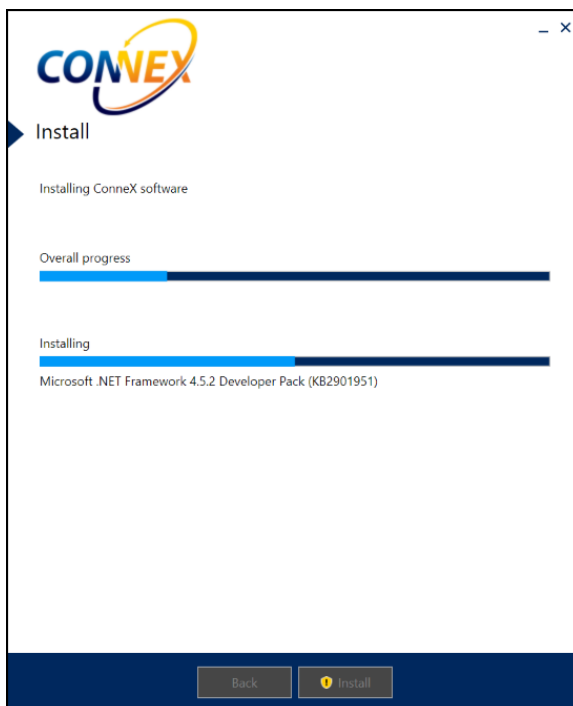
6. On the **Firewall Options** page, check the box for **Open firewall ports**, and then click **Next**.



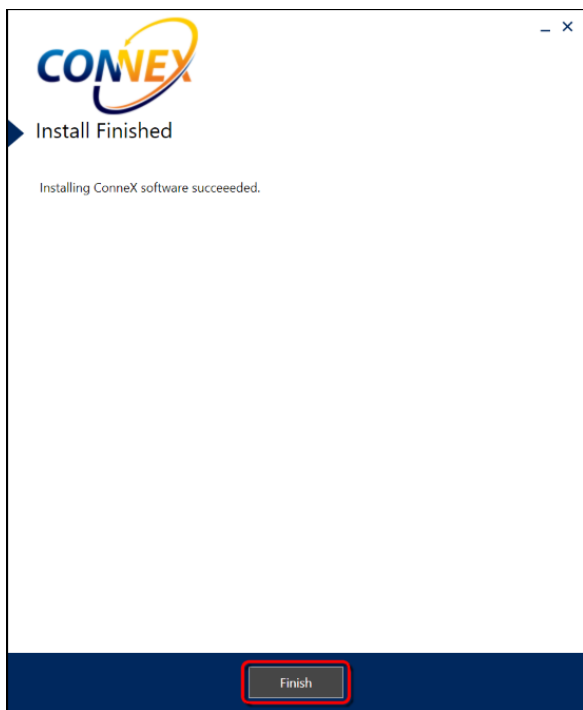
7. On the **Install** page, click **Install**.



8. In the progress screen, let's verify that ConneX is installing.



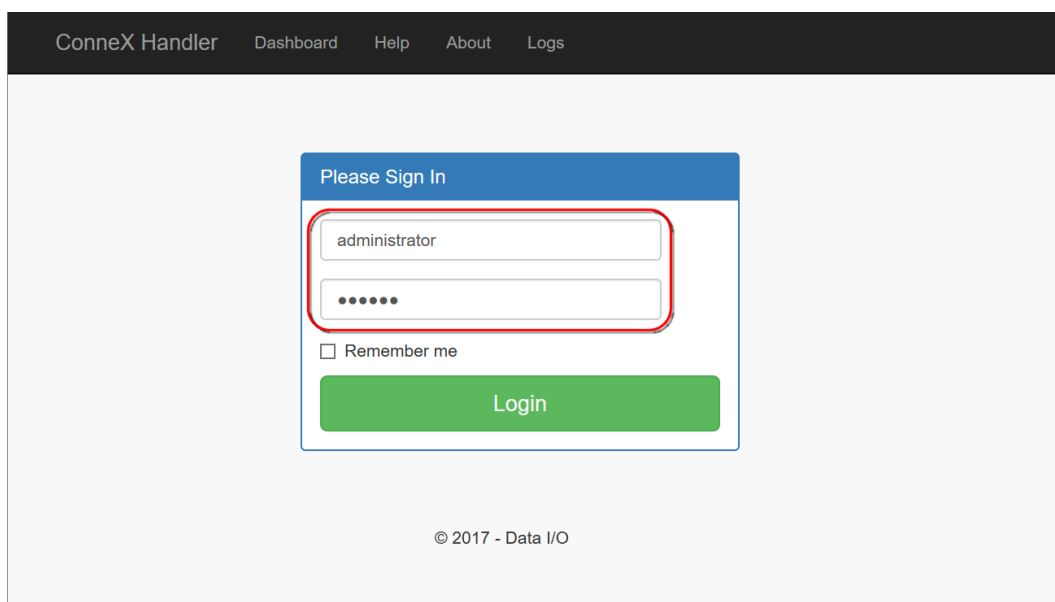
9. On the **Install Finished** page, click **Finish**.



Verify the Installation

To verify that ConneX is installed, let's open a browser to <http://localhost:5000>.

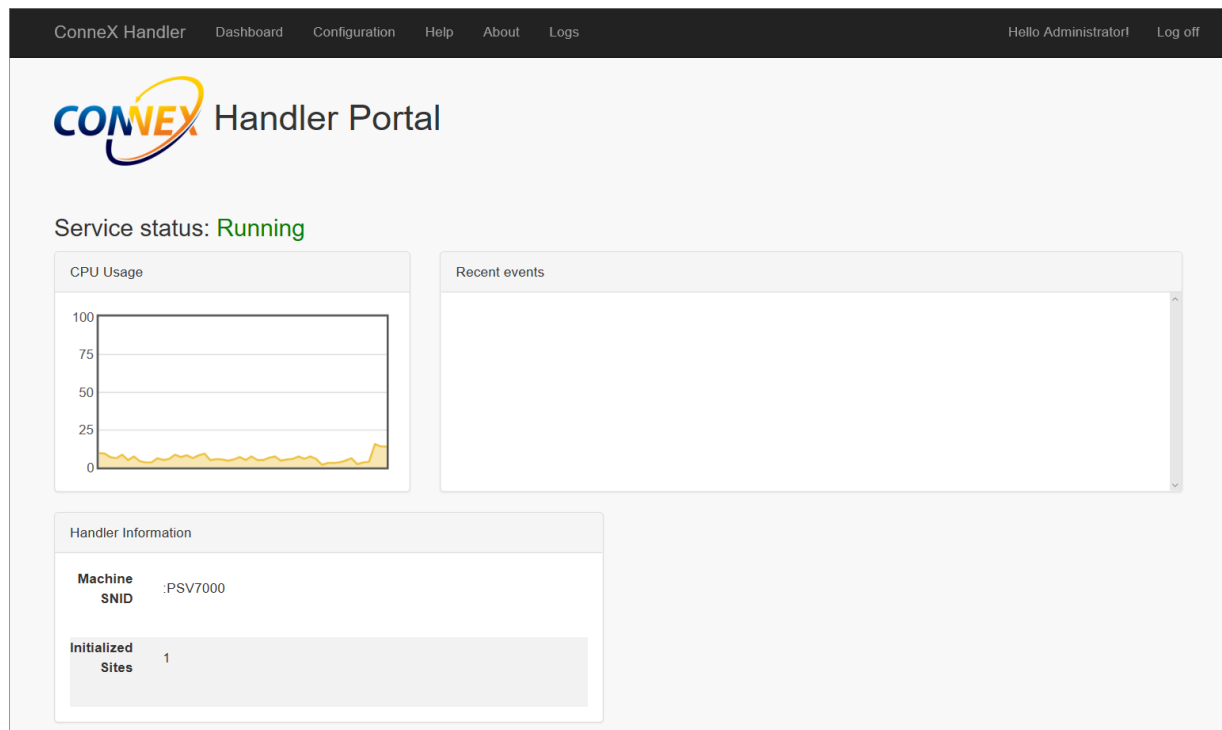
On the Login page, use the default Administrator account to login (username = *administrator* and password = *password*).



With ConneX installed, we now have logging and traceability ready for our programming system(s). Feel free to click around and explore the ConneX interface. No programming jobs have run since installation, so the database of audit records is currently empty. But we'll change that later with some test records.

As you can see, most of ConneX's functionality and capability runs under-the-hood: the data collection and logging services operate behind the scenes. In its user interface, ConneX enables granular record viewing with individual event and socket-level details (such as programming times). But by design, ConneX does not display any cumulative data visualizations or aggregated-level reporting.

The simplicity of ConneX is its ability to substantiate part-manufacturing authenticity and traceability by collecting detailed programming statistics and logging them securely without much manual intervention. So the majority of the Chapters in this document actually focus on how to format the exported data that ConneX makes available for external reporting and analytics.



First, let's proceed with the Configuration Chapter to ensure connectivity between ConneX and the programmer(s). Then we'll ensure connectivity between ConneX Handler and ConneX Attendant.

Note: Now is a good time to double-check that your software is configured correctly. Specifically, ensure that "FIS-PS=false" was changed to **True** in your **C:\AH700\WinAH400.ini** file (as stated in the Software Requirements section earlier).

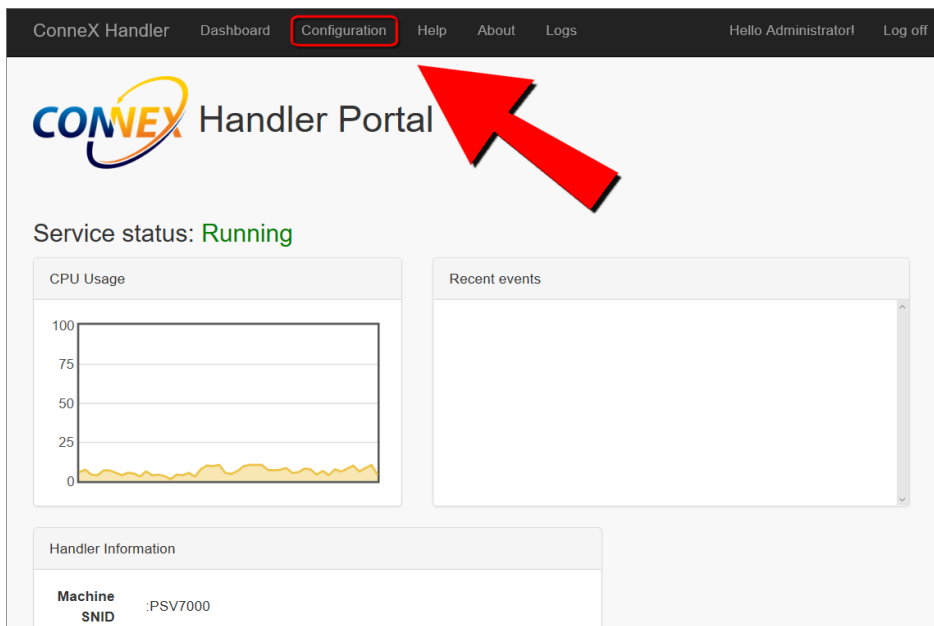
Chapter 2: Configuration

Since ConneX is largely about data collection, the configuration process focuses on establishing connectivity so that data passes properly between programmer, Handler, and Attendant. As mentioned in Chapter 1, you should follow the “*Getting Started Guide for LumenX*” (or FlashCORE III) to connect and establish connectivity between the programmer(s) and the computer. In this Chapter, we’ll ensure connectivity between programmer(s), Handler, and Attendant.

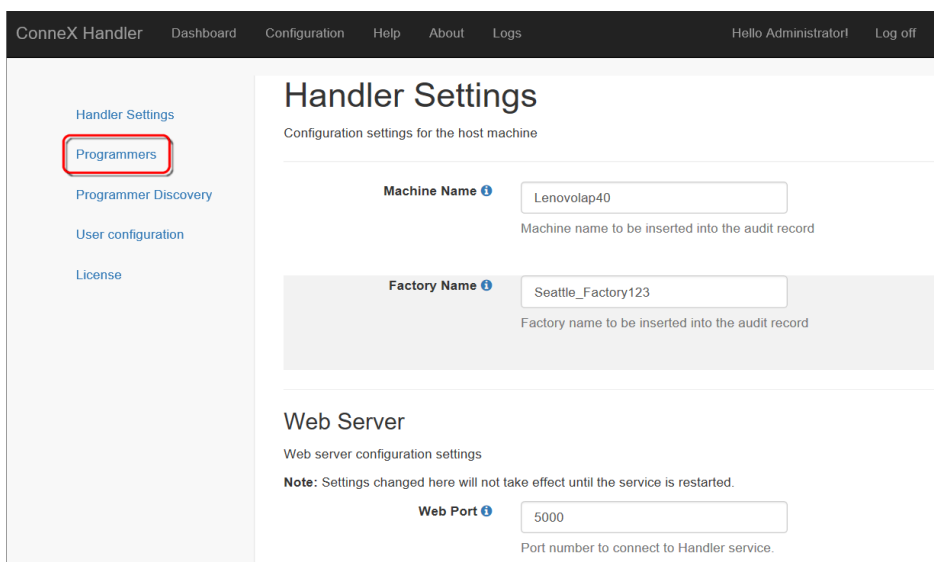
Configure ConneX Handler

Let’s make sure that ConneX Handler sees the programmer(s).

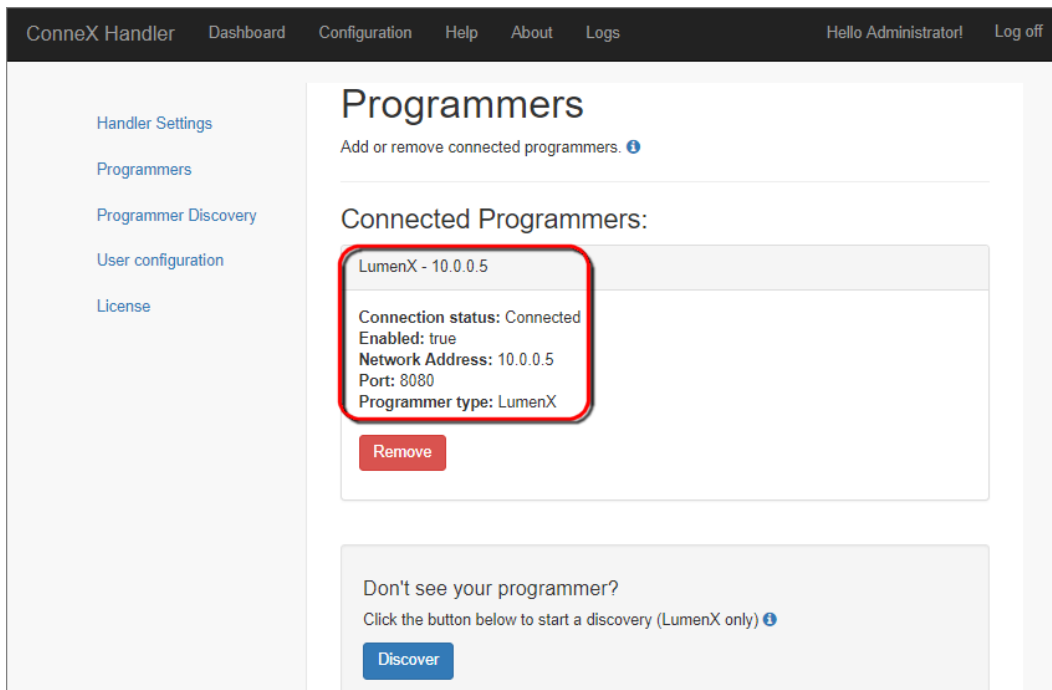
1. In the browser displaying **ConneX Handler Portal**, click **Configuration** in the main navigation.



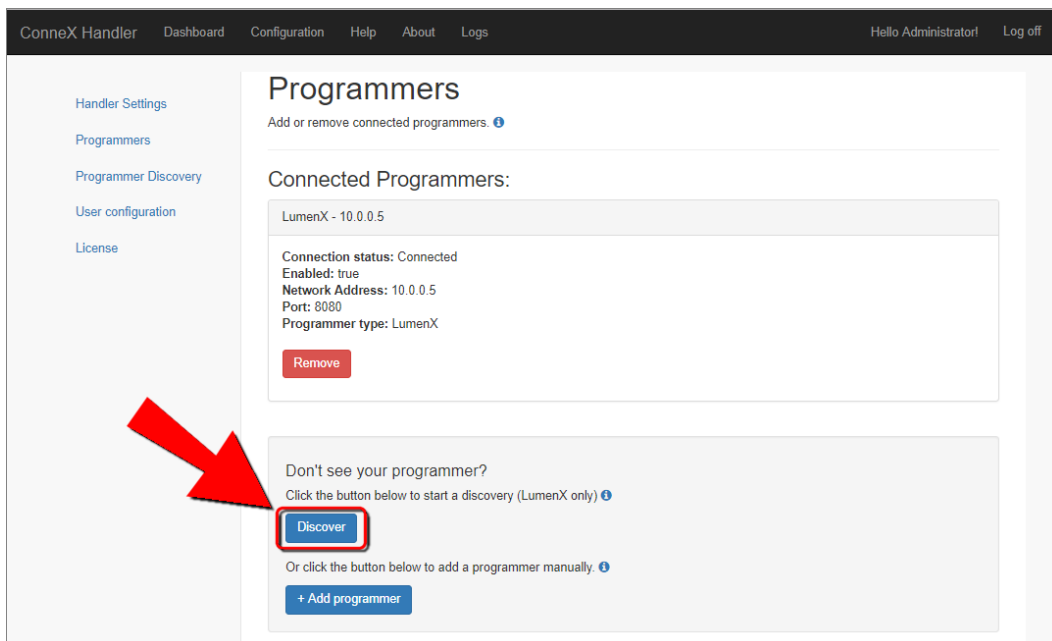
2. On the **Handler Settings** page, click **Programmers** in the left navigation.



3. On the **Programmers** page, verify that the programmer/programming unit is detected and displayed under the **Connected Programmers** list.



4. If the programmer is not displayed in the list (or to add another programmer), we can click **Discover** at the page bottom to have ConneX automatically connect to new programmers.

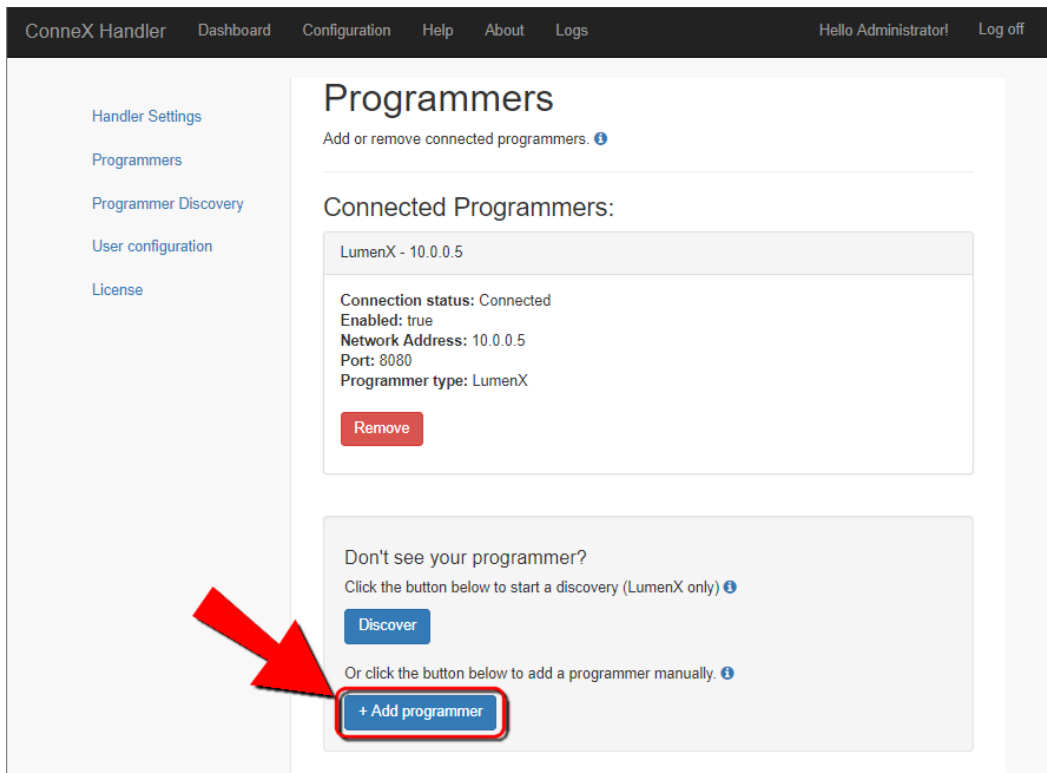


5. To configure the settings for automatic discovery, click **Programmer Discovery** (in the left pane), and then complete the appropriate fields.

Section	Field	Description
General settings	Enabled	Check this box to enable automatic programmer discovery.
	Poll Interval	Specify a time interval (in milliseconds) for periodic programmer discovery checks. Keep the interval above 10,000 so that discoveries do not overlap. The default interval is 60,000 milliseconds.
LumenX settings	Binding IP	The IP address the ConneX uses to search for LumenX programmers. For example, if your programmer has an IP address of 10.0.0.10, then bind it to your computer's IP address that is in the same subnet.
	New Programmer Prefix	Specify a friendly name for the discovered programmer (ConneX appends this friendly name as a prefix to the IP address of the new programmer). The default prefix is "LumenX -".
FlashCORE settings	Binding IP	The IP address the ConneX uses to search for FlashCORE programmers. For example, if your programmer has an IP address of 10.0.0.10, then bind it to your computer's IP address that is in the same subnet.
	Binding Port	The port number that ConneX uses to search for FlashCORE programmers. The default port is 7527.
	New Programmer Prefix	Specify a friendly name for the discovered programmer (ConneX appends this friendly name as a prefix to the IP address of the new programmer). The default prefix is "FlashCORE -".

For example on FlashCORE:

6. To manually add a programmer by IP address, let's click **Add programmer**.



NOTE: Quit, Exit, and Close the AH700 software before adding a programmer.

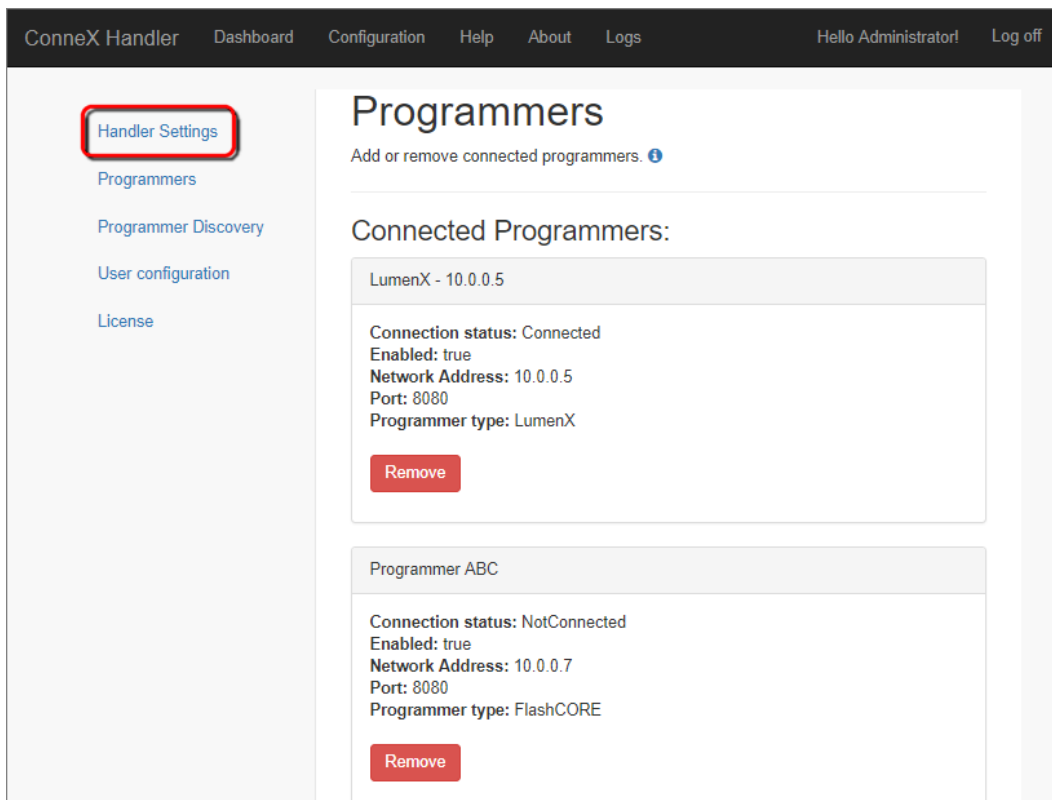
7. In the **Add programmer** dialog box, complete all of the fields, and then click **Add programmer**.

The screenshot shows the 'Add programmer' dialog box. It contains the following fields and controls:

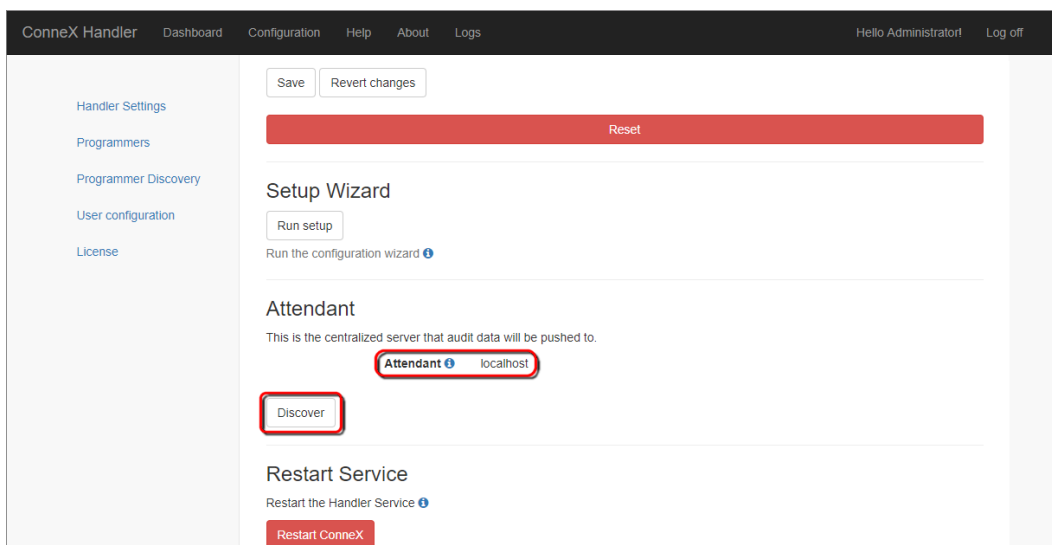
- Programmer Name:** Text input field with 'Programmer ABC' entered.
- Network address:** Text input field with '10.0.0.7' entered.
- Port:** Text input field with '8080' entered.
- Programmer type:** Dropdown menu with 'FlashCORE' selected.
- Enabled:** Checkmark is checked.
- Buttons:** 'Add programmer' (highlighted with a red box) and 'Close'.

With the programmer(s) connected to ConneX Handler, let's verify the IP address that Handler uses to connect with Attendant.

8. On the **Programmers** page, click **Handler Settings** in the left navigation.



9. On the **Handler Settings** page, scroll down to the **Attendant** section, and verify the appropriate IP address for ConneX Attendant (else click **Discover** to have ConneX automatically detect and connect to the Attendant server). For this walkthrough demo, use **localhost** for Attendant.



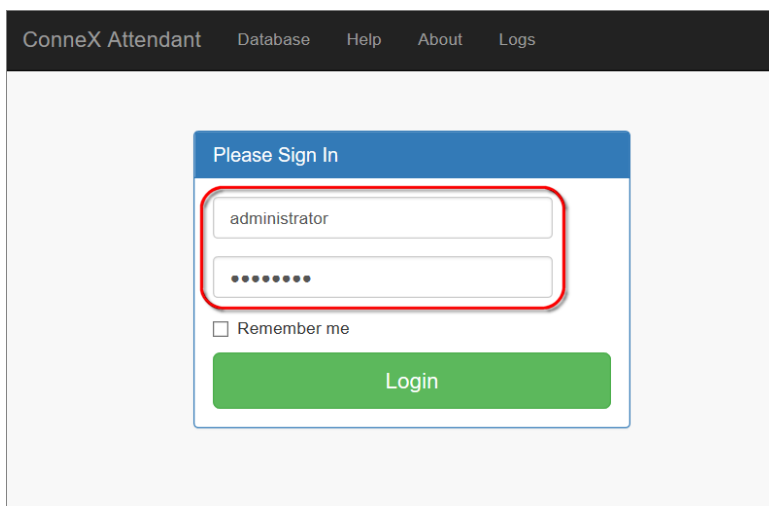
Configure ConneX Attendant

In the previous section on configuring ConneX Handler, we verified connectivity between programmer and Handler. We also verified the IP address that Handler uses to connect with Attendant.

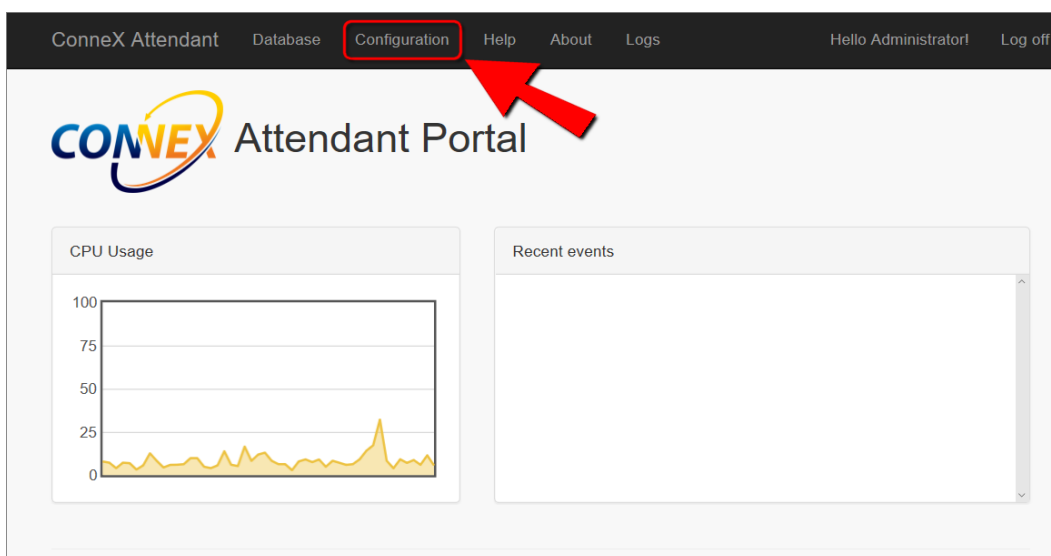
Now for ConneX Attendant, we'll first check its configuration before we start sending some test records in the next Chapter to validate data logging.

Check Attendant Configuration

1. To access ConneX Attendant, let's open a browser to <http://localhost:5001>.
2. On the Login page, use the default Administrator account to login (username = *administrator* and password = *password*).



3. On the **ConneX Attendant Portal** page, let's click **Configuration** in the main navigation.



4. On the **Attendant Settings** page, scroll down and explore the various configuration options.

ConneX Attendant Database Configuration Help About Logs Hello Administrator! Log off

Attendant Settings
Handlers
User configuration

Attendant Settings

Configuration settings for the host machine

Templates

Generate audit records automatically ☒

Output Location C:\ProgramData\DataIO\ConneXAttendant

Output Templates

LumenX template:
CSV Template
Template to use for LumenX records

FlashCORE template:
Default FlashCore
Template to use for FlashCORE records

Filename format

LumenX file name
{AuditRecord.TimeStamp}_{AuditRecord.
Edit

FlashCORE file name
{AuditRecord.AuditRecordFlashCore.Tim

After exploring the Attendant configuration, we're finally ready to start sending events and logging data as audit records in the ConneX database.

Chapter 3: Verification

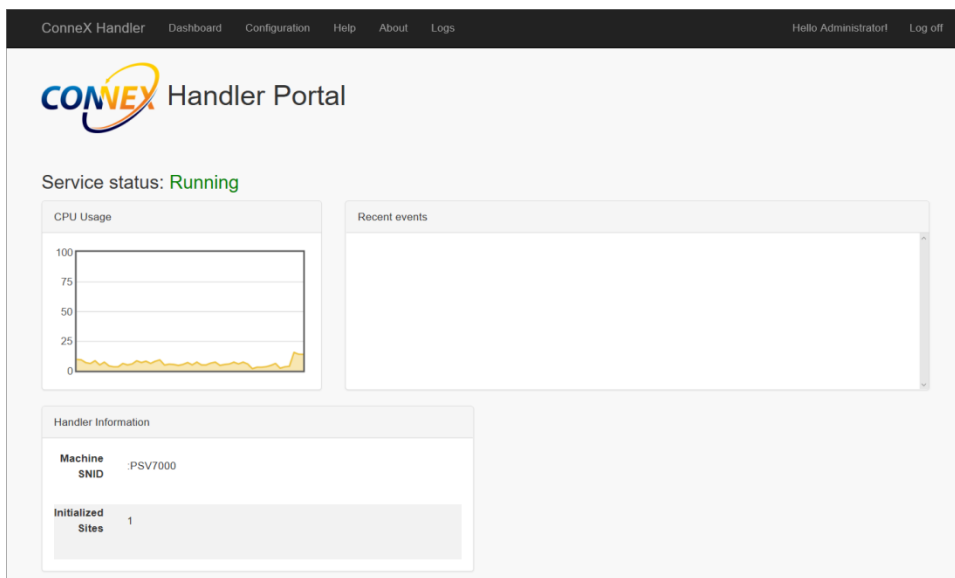
With connectivity now established between programmer(s), Handler, and Attendant, let's open the Handler Portal and Attendant Portal in two separate browsers (and keep them open) so we can see real-time data come through the ConneX system as we send test events.

Then we'll switch over to the Handler and Attendant dashboards to see the historical data that ConneX logged. In the last section of this Verification chapter, we'll look at the audit records that ConneX exports and saves as files on the computer hard drive.

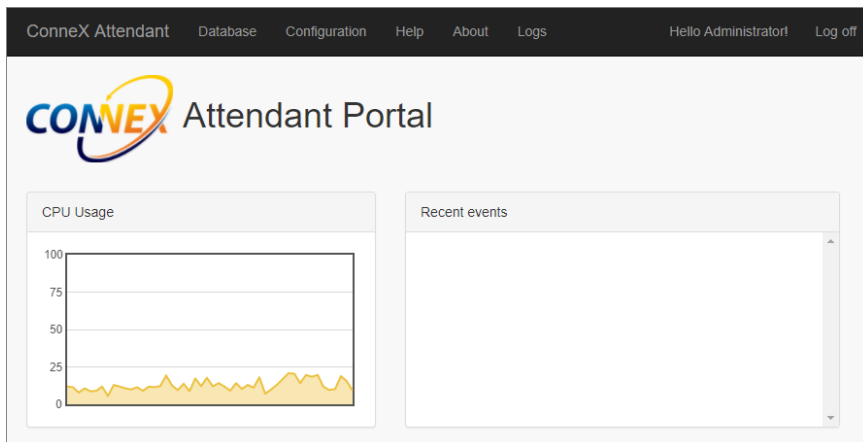
Open ConneX Portal Pages

Before we send job data and submit test records to validate data logging and exporting, let's open two separate browsers to view real-time data as it comes in.

1. Open a browser window to the Handler Portal page at <http://localhost:5000>.



2. Open another browser window to the Attendant Portal page at <http://localhost:5001>.



Each Portal page provides important real-time visibility into LumenX job details for production personnel on the factory floor (such as Engineers, Technicians, and Operators). ConneX enables authorized users to select a specific event and drill-down to view all of the data in its audit record. For example, selecting an event in ConneX Attendant will reveal a socket-by-socket report of job statistics.

ConneX Attendant
Database
Configuration
Help
About
Logs
Hello Administrator!
Log off

Audit Record

Record ID: 2

Programmer: 001-035-088-144-082-061-119-008-238

Job Name: IFX V15 (AlgorithmId: 33773737325102592)

Socket: 1

Program: 69.6 (milliseconds)	Program: 57.417 (milliseconds)	Program: 63.96 (milliseconds)	Verify: 38.578 (milliseconds)	Verify: 46.007 (milliseconds)	Verify: 412.162 (milliseconds)	Pre Operations: 1440.971 (milliseconds)	Post Operations: 0.921 (milliseconds)
-------------------------------------	---------------------------------------	--------------------------------------	--------------------------------------	--------------------------------------	---------------------------------------	--	--

Socket: 5

Program: 59.554 (milliseconds)	Program: 67.112 (milliseconds)	Program: 58.282 (milliseconds)	Verify: 44.624 (milliseconds)	Verify: 39.813 (milliseconds)	Pre Operations: 720.937 (milliseconds)	Post Operations: 0 (milliseconds)
---------------------------------------	---------------------------------------	---------------------------------------	--------------------------------------	--------------------------------------	---	--

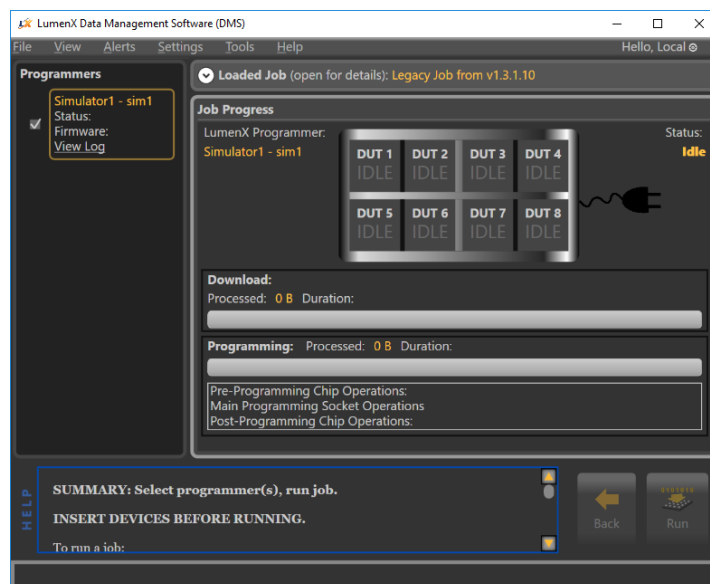
▼ Raw Data

```
{
  "AuditData": {
    "@SchemaVersion": "1.4.6366.28259",
    "@AuditRecordId": "13a9a8e2-17a1-4823-a5e8-36e5c98e041d",
    "Records": {
      "AuditRecord": [
        {
          "TimeStamp": "7/26/2017"
        }
      ]
    }
  }
}
```

Copy to clipboard Convert to XML

Submit Test Record(s)

To generate a test event with some data for an audit record, run one or more programming jobs in LumenX or TaskLink (for FlashCORE III). See LumenX Help or TaskLink Help to create and run the job.

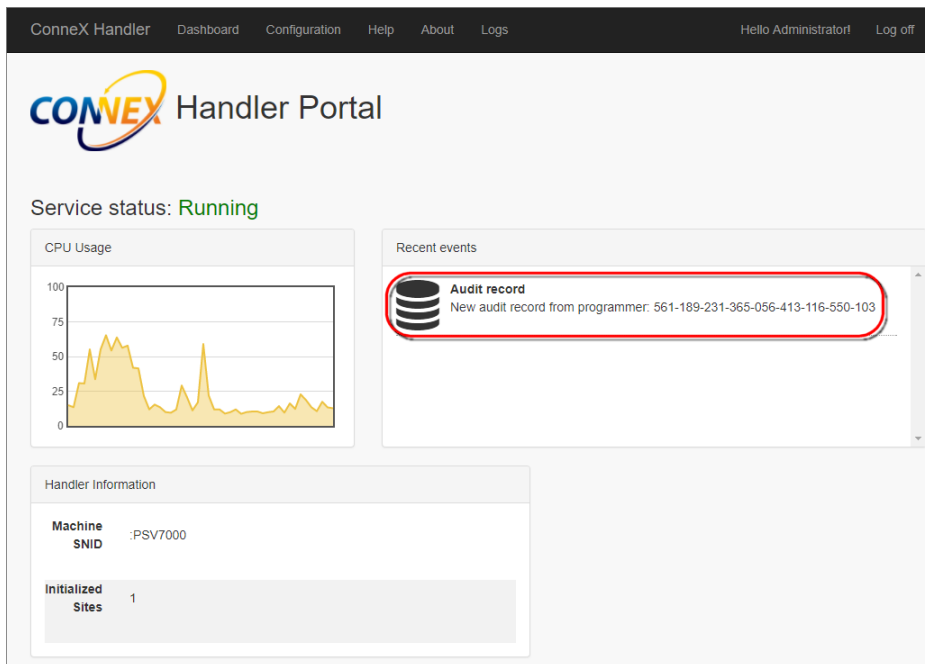


Validate Data Logging

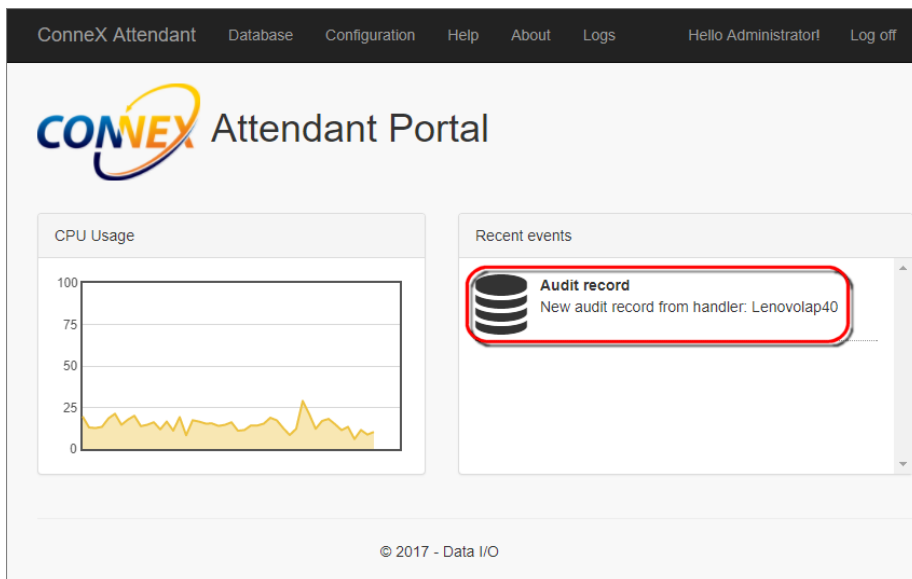
As the job runs and completes, we can view the Portal pages to see the LumenX event appear as real-time data. And we can also view the dashboards to see the event logged as historical data.

View Real-Time Data

1. In the browser window for the **Handler Portal** page, view the event under **Recent events**. Note that events appear only for LumenX jobs, not FlashCore (you can view FlashCore job events in real-time on the **Dashboard** page).



2. Similarly, view the Attendant Portal page to see the event appear in real-time.

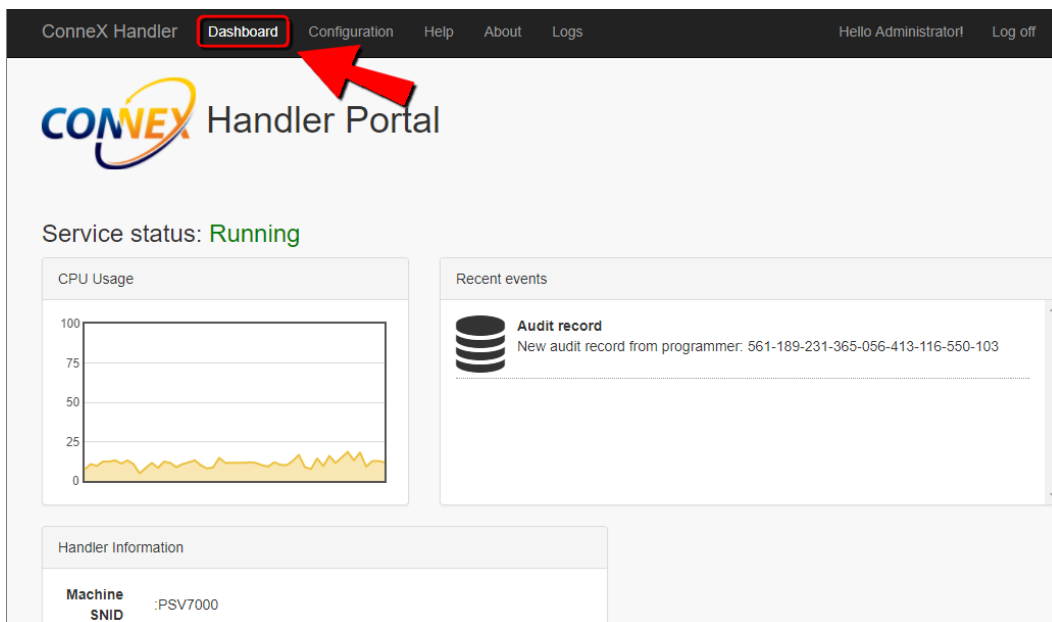


As programming jobs run and complete, ConneX Handler logs each event/audit record in an internal database (residing on the Handler computer) by individual programmer. Then as ConneX Handler pushes audit records to ConneX Attendant, it packages additional Handler information to the record so that Attendant can distinguish records by Handler (recall that multiple programmers can connect to a Handler, and multiple Handlers may feed a centralized Attendant Server as illustrated on page 2).

The **Recent events** list on each Portal page will clear if we refresh the page (or leave and return to the Portal page) because it is displaying events in real-time. But we always have access to the static log of events on the Dashboard page.

View Historical Data

1. In the browser window for ConneX Handler, click **Dashboard** in the main navigation.



2. On the **Audit Records** page, we see the historical data logged in the Handler database.

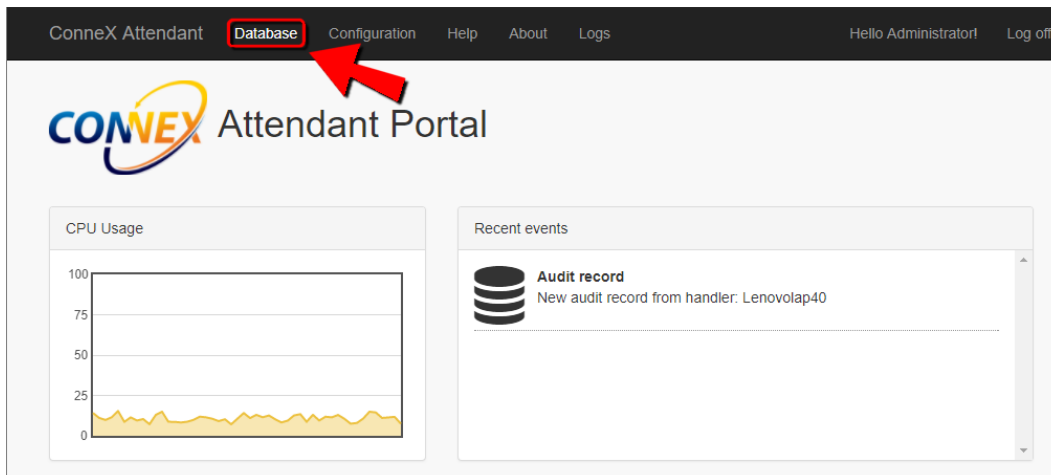
ConneX Handler Dashboard Configuration Help About Logs Hello Administrator! Log off

Audit Records

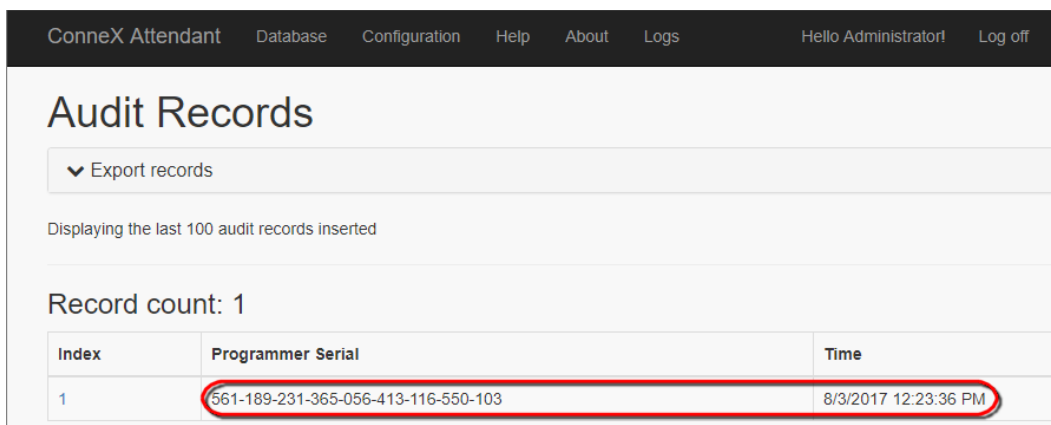
Record count: 1

Index	Programmer Serial	Time
1	561-189-231-365-056-413-116-550-103	8/3/2017 12:23:36 PM

3. In the browser window for ConneX Attendant, click **Database** in the main navigation.



4. On the **Audit Records** page, we see the historical data logged in the Attendant database.



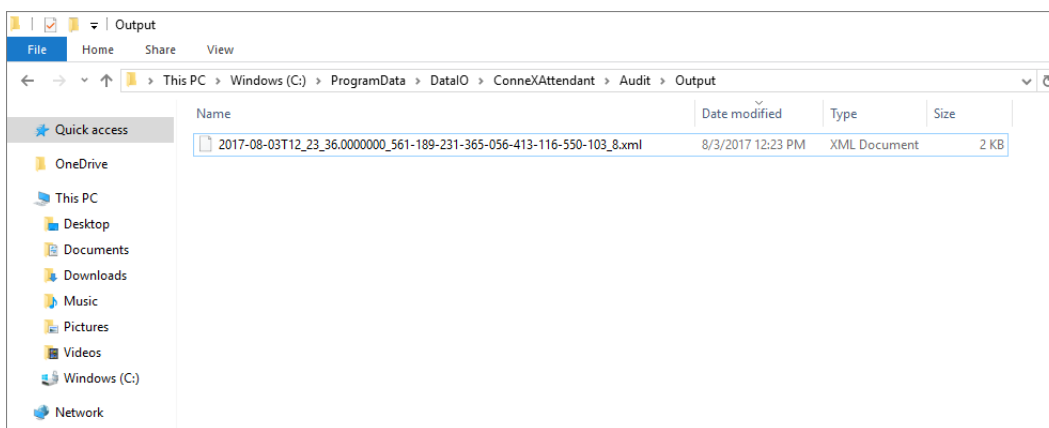
Now that we've viewed real-time and historical data residing in the ConneX database, let's look at the exported data that ConneX makes available for external reporting and analytics.

Validate Data Exports

In addition to viewing programming statistics within the ConneX Portal pages and dashboards, we can access the audit records that ConneX makes available externally as files on the hard drive. By default, ConneX copies each audit record from the database to an XML file located at *C:\ProgramData\DataIO\ConneXAttendant\Audit\Output*.

Through an audit record template in ConneX, we can change the output format from XML to JSON, TXT, or CSV. We can also modify the log entries and data fields within the record. Then the exported data can be imported into an external database or analytics tool for data visualization and business intelligence reporting. Let's look at the default XML output and see how it's structured.

1. Open Windows/File Explorer and navigate to *C:\ProgramData\DataIO\ConneXAttendant\Audit\Output*.



2. Using Notepad, Wordpad, or any text editor, open the XML file that ConneX created with our test programming job run.
3. Verify that the audit record appears similar to the output on the following page.

Sample XML Audit Record/Data Output

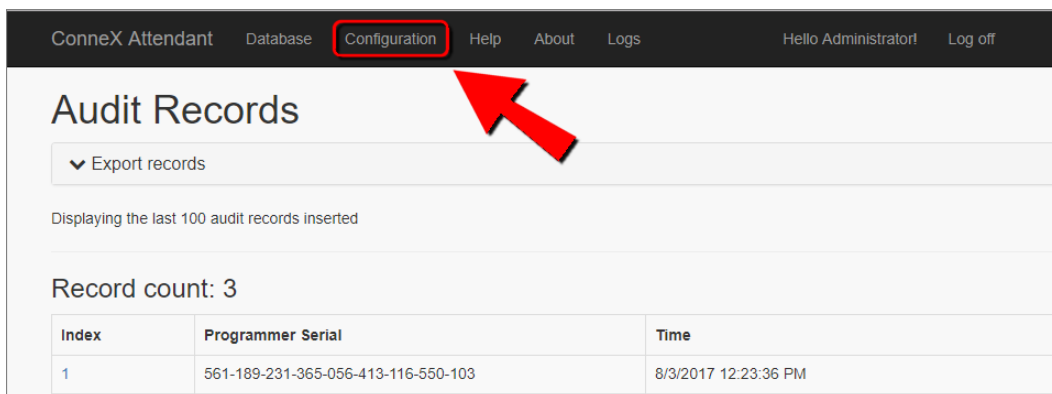
```

<?xml version="1.0" encoding="utf-8" ?>
<devicerecord>
  <timestamp>2017-08-03T12:23:36.0000000</timestamp>
  <system>
    <name>Lenovolap40</name>
    <softwareversion>2.0.0.78</softwareversion>
    <ip>
      <address>10.0.0.5</address>
    </ip>
    <machineSNID>:PSV7000</machineSNID>
    <machineName>Lenovolap40</machineName>
    <factoryName>Texas_FactoryXYZ</factoryName>
    <programmer>
      <progClass>LumenX</progClass>
      <progSerialNum>561-189-231-365-056-413-116-550-103</progSerialNum>
      <progSWVersion>1.3.6198.30204</progSWVersion>
      <adapter>
        <number>110008</number>
        <serialNumber>561-189-231-365-056-413-116-550-103</serialNumber>
        <socketCount>146079</socketCount>
        <adapterIndex>8</adapterIndex>
      </adapter>
    </programmer>
  </system>
  <job>
    <jobName>Job ABC</jobName>
    <jobDescription></jobDescription>
    <algorithmID>34761085891379200</algorithmID>
  </job>
  <partDetails>
    <uid></uid>
    <progResults>
      <resultCode>0</resultCode>
      <resultCodeName>Success</resultCodeName>
      <passPosition></passPosition>
      <eraseTime>0</eraseTime>
      <blankCheckTime>0</blankCheckTime>
      <programTime>44.643</programTime>
      <verifyTime>26.413</verifyTime>
      <deviceError></deviceError>
    </progResults>
  </partDetails>
  <serialData></serialData>
</devicerecord>

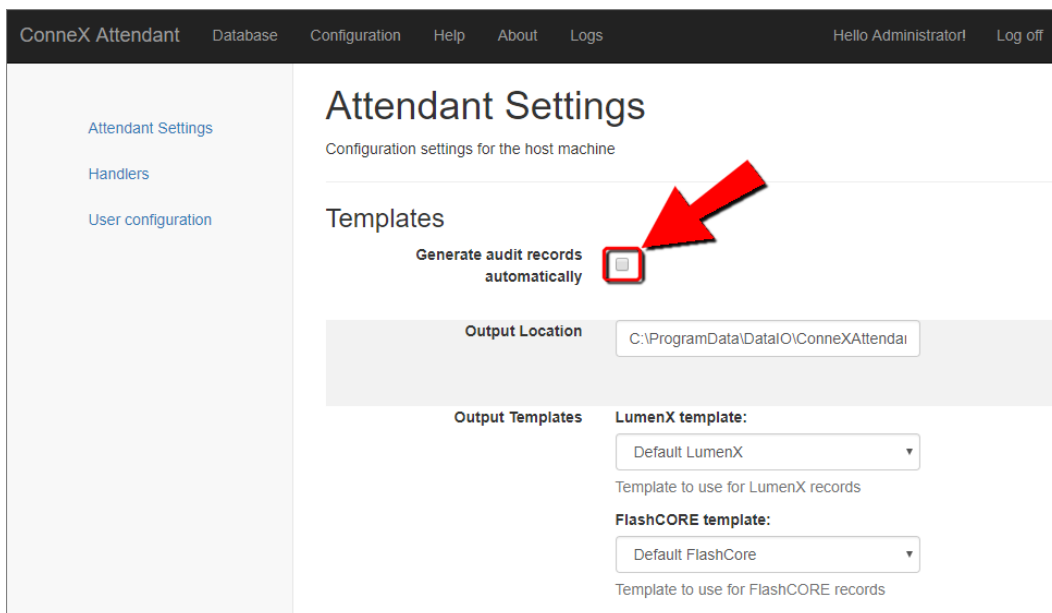
```



4. To disable the automatic export feature or change the output location, let's click **Configuration** in the main navigation of ConneX Attendant.

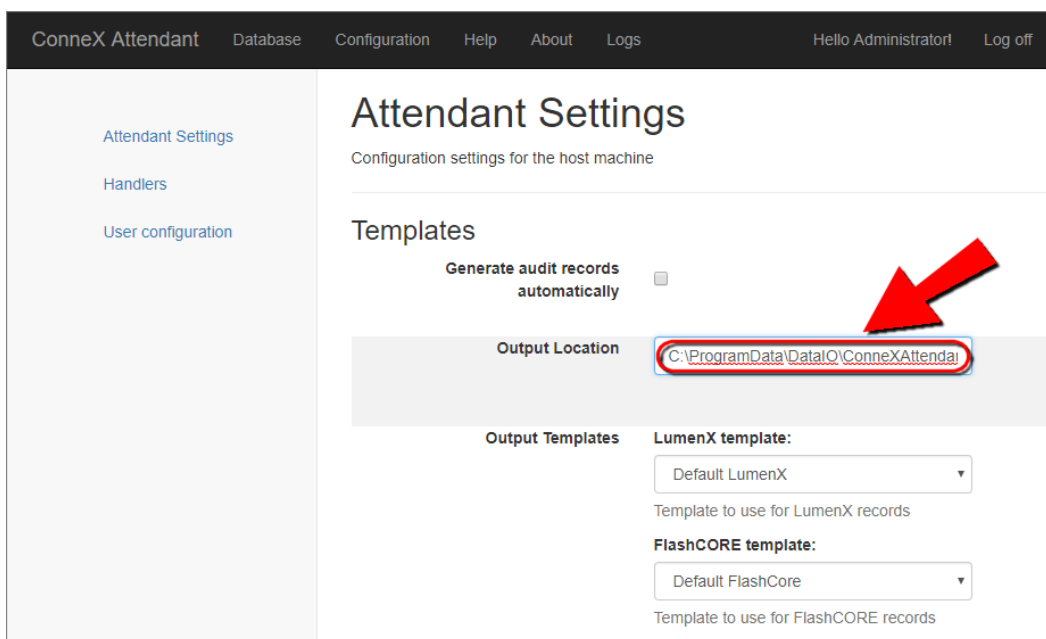


5. On the **Attendant Settings** page, uncheck the box for **Generate audit records automatically** to disable automatic data exports.



NOTE: Unchecking this box simply disables the automatic generation and exporting of audit records from the ConneX database (ConneX will not copy audit records from its database and save them as files on the hard drive). But ConneX still logs the events in its own internal database for auditing and traceability (it just doesn't make them available externally until or unless you check this box).

6. To change the output folder where ConneX exports the XML files to, type a different path in the **Output Location** box.



Keep in mind that if you remotely connect to the ConneX Attendant Server, the output location resides on the Attendant computer that is running the ConneX Attendant service (not your local terminal).

Now that we've seen the data output that ConneX can export, let's look at the default template that ConneX uses to define the fields and format of the audit record. Later at the end, we'll create a new template that has different fields, uses a different file-naming convention, and saves as a different file format that makes it easier to import audit data from ConneX into an external reporting tool.

View Default XML Export Template

To see how the data structure is defined for the sample XML audit record on page 21, let's look at the default output template that essentially defines an XML schema for the logged data.

1. On the **Attendant Settings** page, at the bottom of the **Templates** section, click **Modify Templates**.

The screenshot shows the 'Attendant Settings' page for 'ConneX Attendant'. The left sidebar has links for 'Attendant Settings', 'Handlers', and 'User configuration'. The main content area is titled 'Attendant Settings' and 'Configuration settings for the host machine'. Under the 'Templates' section, there is a checkbox for 'Generate audit records automatically' which is checked. Below this is the 'Output Location' set to 'C:\ProgramData\DataIO\ConneXAttendant'. The 'Output Templates' section includes 'LumenX template:' with a dropdown set to 'Default LumenX' and 'FlashCORE template:' with a dropdown set to 'Default FlashCore'. The 'Filename format' section shows 'LumenX file name' as '{AuditRecord.Timestamp}_{AuditRecord.' and 'FlashCORE file name' as '{AuditRecord.AuditRecordFlashCore.Tim'. At the bottom of the 'Templates' section, a 'Modify Templates' button is highlighted with a red rectangle and a red arrow points to it from the left.

2. On the **Templates** page, notice how the default templates cannot be removed.

The screenshot shows the 'Templates' page in the ConneX Attendant configuration. It has a 'Create new' button and a search bar. Below the search bar is a table with columns: 'Template Name', 'Format', 'Last Modified', and an action column. The table lists two templates: 'Default FlashCore' and 'Default LumenX', both in 'xml' format, with a 'Last Modified' date of '1/1/0001 12:00:00 AM'. In the action column, there are two 'Remove' buttons, each with a red 'X' icon. Both 'Remove' buttons are highlighted with red rectangles.

Template Name	Format	Last Modified	
Default FlashCore	xml	1/1/0001 12:00:00 AM	✖ Remove
Default LumenX	xml	1/1/0001 12:00:00 AM	✖ Remove

3. On the **Templates** page, click **Default LumenX** to view its XML template/format.

ConneX Attendant Database Configuration Help About Logs Hello Administrator! Log off

Templates

Create new

Show 10 entries Search:

Template Name	Format	Last Modified	
Default FlashCore	xml	1/1/0001 12:00:00 AM	✕ Remove
Default LumenX	xml	1/1/0001 12:00:00 AM	✕ Remove

Showing 1 to 2 of 2 entries

Previous 1 Next

4. On the Default LumenX template page, notice again how default templates cannot be altered (but we can create a new template using the same schema and make modifications).

ConneX Attendant Database Configuration Help About Logs Hello Administrator! Log off

[Go back to templates](#)

Default LumenX

Template name

Template is marked as read only. Saving and editing have been disabled.

File Formatting

Save

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <devicerecord>
3   <timestamp>{AuditRecord.Timestamp}</timestamp>
4   <system>
5     <name>{AuditRecord.HandlerInfo.Name}</name>
6     <softwareversion>{AuditRecord.HandlerInfo.Version}</softwareversion>
7     <ip>
8       <address>{AuditRecord.HandlerInfo.IpAddresses}</address>
9     </ip>
10    <machineSNID>{AuditRecord.HandlerInfo.MachineSNID}</machineSNID>
11    <machineName>{AuditRecord.HandlerInfo.MachineName}</machineName>
12    <factoryName>{AuditRecord.HandlerInfo.FactoryName}</factoryName>
13    <programmer>
14      <progClass>{AuditRecord.Programmer.Class}</progClass>
15      <progSerialNum>{AuditRecord.Programmer.SerialNumber}</progSerialNum>
```

- To view the XML schema (or field definitions), scroll down to the template code box.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <devicerecord>
3   <timestamp>{AuditRecord.Timestamp}</timestamp>
4   <system>
5     <name>{AuditRecord.HandlerInfo.Name}</name>
6     <softwareversion>{AuditRecord.HandlerInfo.Version}</softwareversion>
7     <ip>
8       <address>{AuditRecord.HandlerInfo.IpAddresses}</address>
9     </ip>
10    <machineSNID>{AuditRecord.HandlerInfo.MachineSNID}</machineSNID>
11    <machineName>{AuditRecord.HandlerInfo.MachineName}</machineName>
12    <factoryName>{AuditRecord.HandlerInfo.FactoryName}</factoryName>
13    <programmer>
14      <progClass>{AuditRecord.Programmer.Class}</progClass>
15      <progSerialNum>{AuditRecord.Programmer.SerialNumber}</progSerialNum>
16      <progSWVersion>{AuditRecord.Programmer.FirmwareVersion}</progSWVersion>
17      <adapter>
18        <number>{AuditRecord.Programmer.Adapter.AdapterId}</number>
19        <serialNumber>{AuditRecord.Programmer.Adapter.AdapterSerialNumber}</serialNumber>
20        <socketCount>{AuditRecord.Programmer.Adapter.LifetimeActuationCount}</socketCount>
21        <adapterIndex>{AuditRecord.Programmer.Adapter.SocketIndex}</adapterIndex>
22      </adapter>
23    </programmer>
24  </system>
25  <job>
26    <jobName>{AuditRecord.Job.JobName}</jobName>
27    <jobDescription>{AuditRecord.Job.JobDescription}</jobDescription>
28    <algorithmID>{AuditRecord.Job.AlgorithmId}</algorithmID>
29  </job>
30  <partDetails>
31    <uid>{AuditRecord.PartDetail.Result.AlgoDeviceDetails.device.UID}</uid>
32    <progResults>
33      <resultCode>{AuditRecord.PartDetail.Result.Code}</resultCode>

```

Now we can see how these entries in the template define the entries in the XML data output earlier on page 21. But let's suppose that the default template does not align well with the data structure in our external reporting tool and we need to make some changes:

- Remove some fields from the template because the external reporting tool does not have equivalent fields mapped, and/or management does not currently care about these statistics
- Add some other fields because although the default template does not include them, ConneX is capable of logging additional statistics and details that are not included in the default template (complete list in Appendix)
- Change the sequence of some fields to match the data sequence in the external reporting tool
- Reformat the template code from XML syntax to comma-delimited syntax to simplify import
- Change the output file format from XML to CSV to simplify import

To implement these changes, we'll copy the template code from a default template (which we cannot alter) into a new template and make our customizations in the new template. Then we'll view the results of our customizations by generating an audit record and comparing its file output with our modified template.

Chapter 4: Customization

Now we'll move on from exploring the defaults in ConneX to making some customizations. In this Chapter we'll define our own entries and structure for the audit record(s) that ConneX generates. We'll also change the data output format to accommodate our external data import process and systems.

Specifically, we'll cover the following steps in this Chapter:

- Create a new template
- Customize the new template
 - Add and remove fields
 - Change the sequence of fields
 - Reformat the template code from XML to CSV
 - Modify the output format from XML to CSV
 - Select the customized template in ConneX
- View the customized data output

Before creating a new template, let's copy the code from the Default LumenX template so we can paste and use the code in our own template.

1. To open the Default LumenX template, login to ConneX Attendant (at <http://localhost:5001>) and click **Configuration** in the main navigation at the top.
2. On the **Attendant Settings** page, at the bottom of the **Templates** section, click **Modify Templates**.

ConneX Attendant Database Configuration Help About Logs Hello Administrator! Log off

Attendant Settings

Configuration settings for the host machine

Attendant Settings
Handlers
User configuration

Templates

Generate audit records automatically ☒

Output Location C:\ProgramData\DataIO\ConneX\Attendant\

Output Templates

LumenX template:
Default LumenX
Template to use for LumenX records

FlashCORE template:
Default FlashCore
Template to use for FlashCORE records

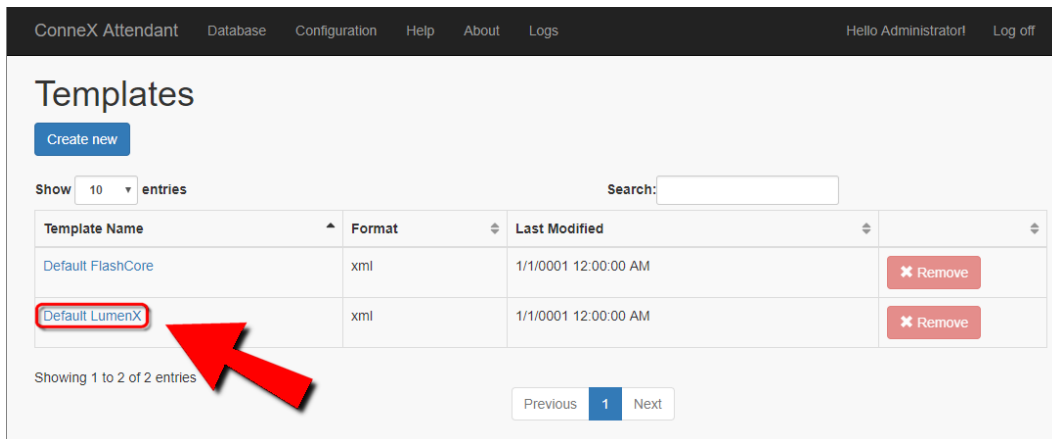
Filename format

LumenX file name
{AuditRecord.TimeStamp}_{AuditRecord.
Edit

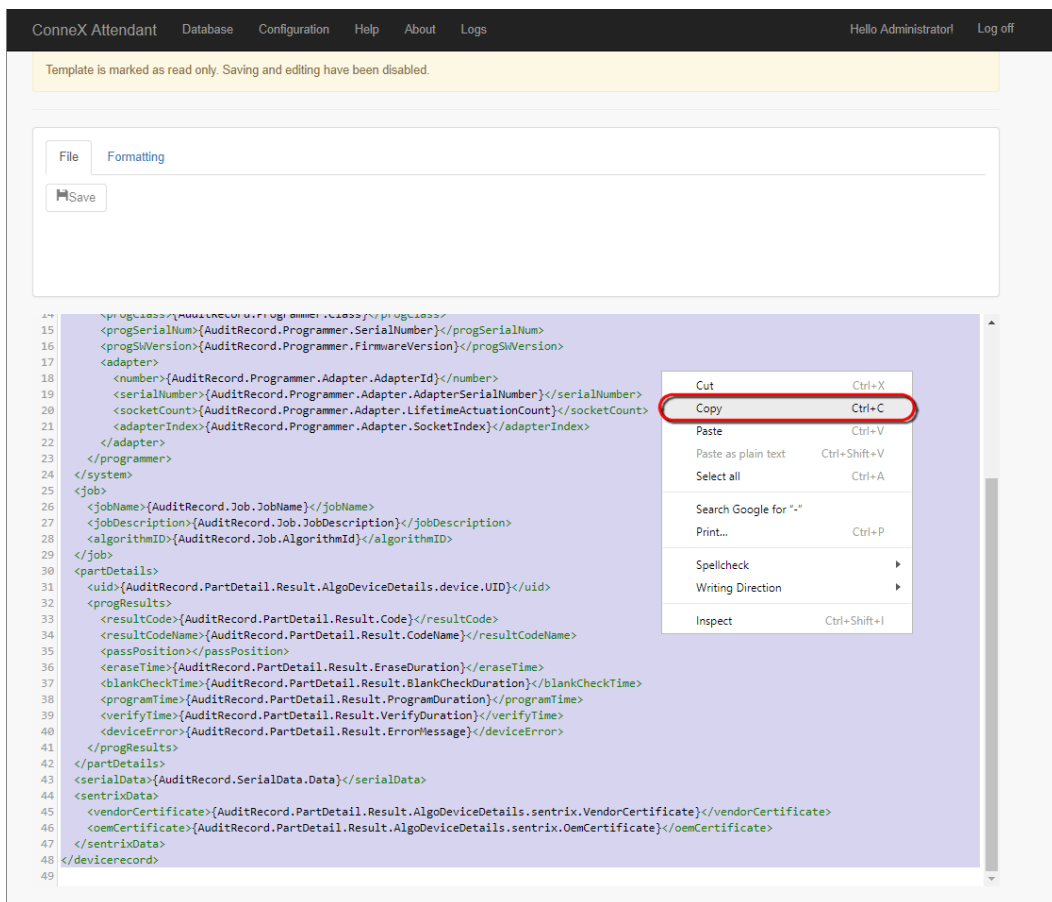
FlashCORE file name
{AuditRecord.AuditRecordFlashCore.Tim
Edit

Modify Templates

- On the **Templates** page, let's click **Default LumenX** to open its template.



- On the Default LumenX template page, scroll down to the template code box at the bottom, select all of the template code, and then copy it.



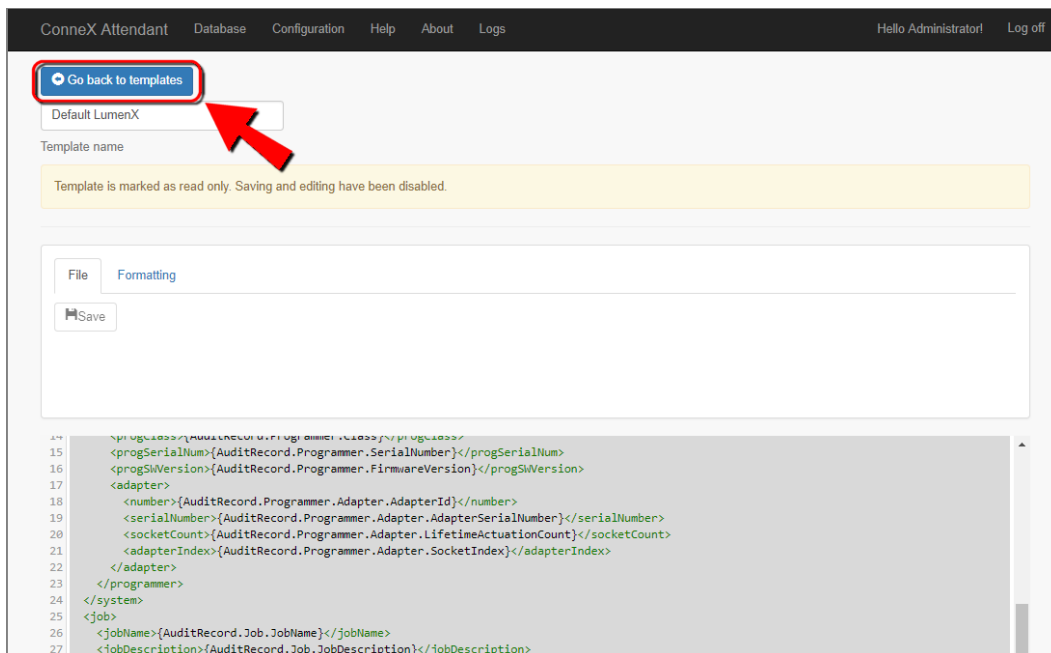
Now we'll create a new template and use (paste) the copied Default LumenX template code as a starting point for our own customized template.



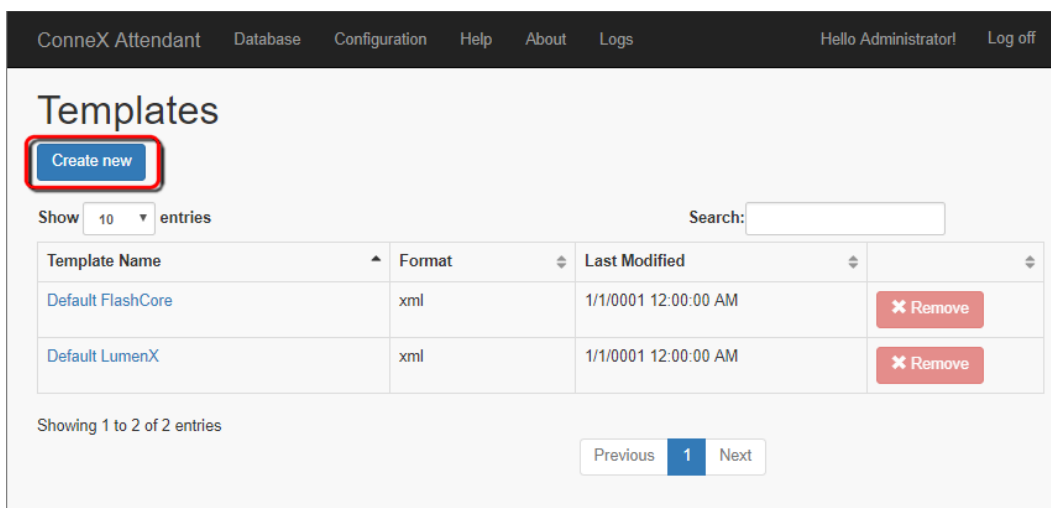
Create a New Template

After copying the code from the Default LumenX template in the previous section, let's paste it into a new template of our own. We're essentially creating a clone of the Default LumenX template so we can customize it.

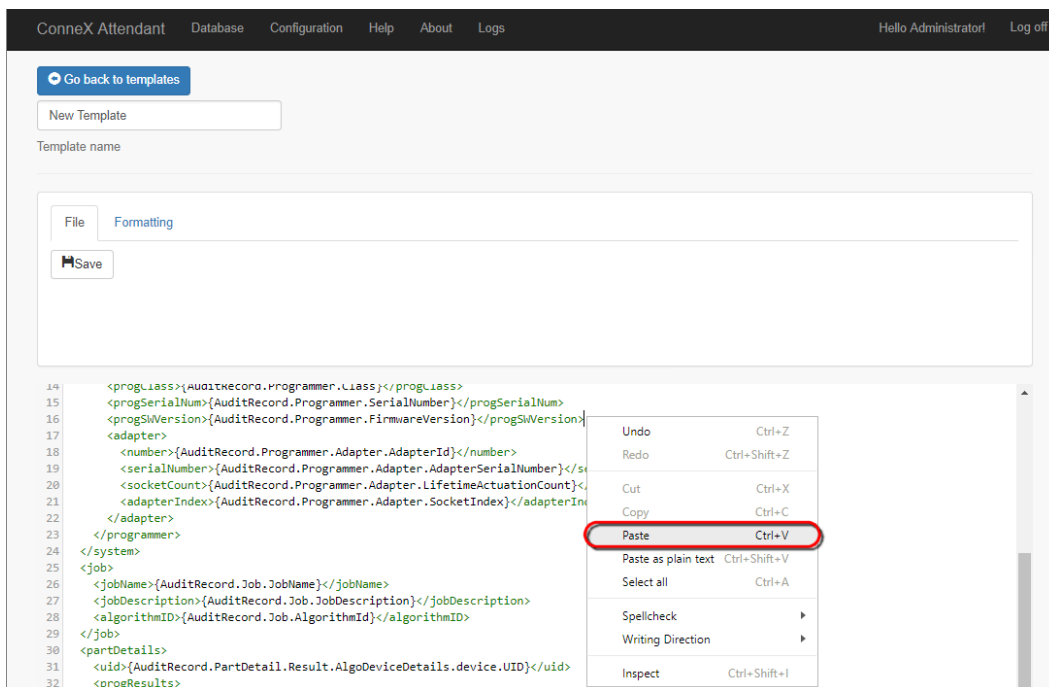
1. Let's scroll up to the top of the page and click **Go back to templates**.



2. On the **Templates** page, click **Create new**.

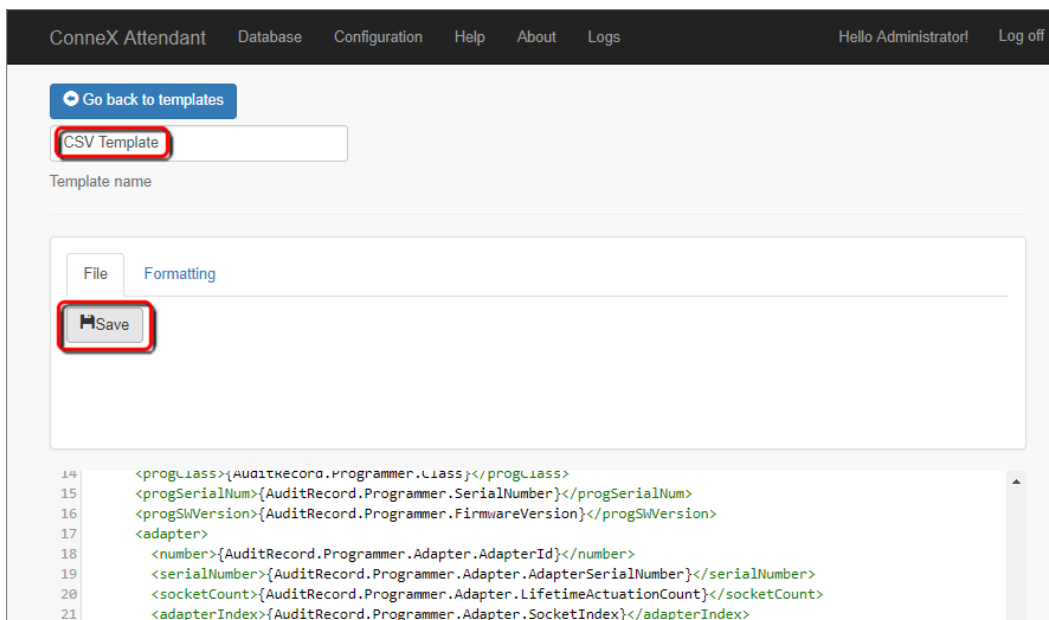


- On the **New Template** page, right-click in the template code box at the bottom, and then click **Paste**.



The code from the Default LumenX template now appears in our new template.

- At the top of the page, in the **Template name** box, type **CSV Template**, and then click **Save**.



Now we'll modify the template code and its output format so that it imports easily into an external reporting tool.



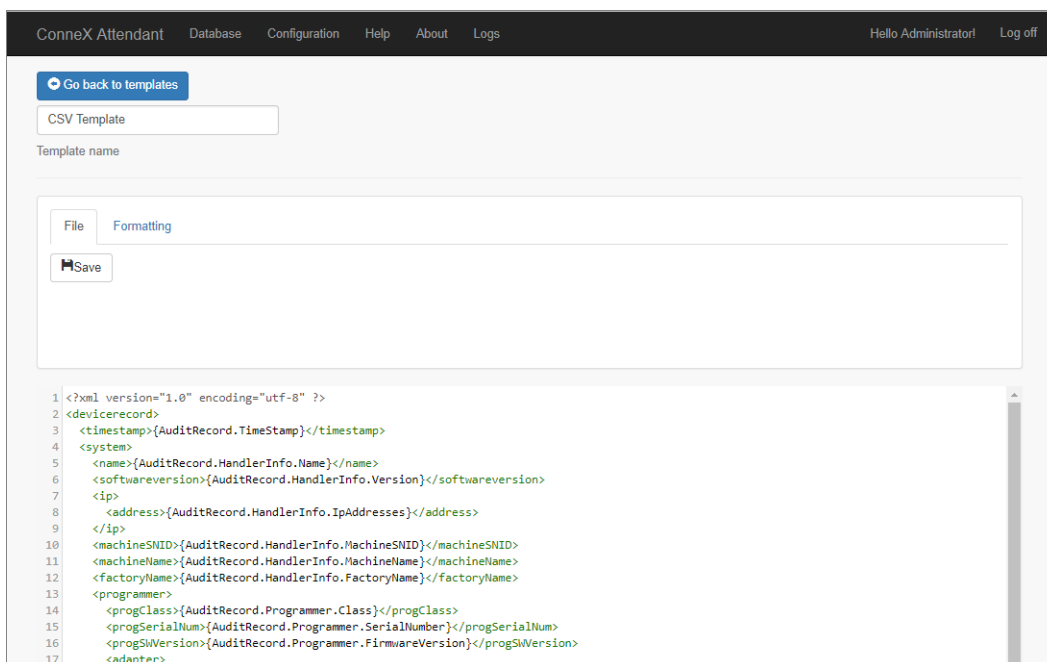
Customize the New Template

After pasting the Default LumenX template code into our own new template, we now have our own version/clone of the Default LumenX template. Let's modify some entries in the template so that later, ConneX can export audit records containing fields aligned with the external reporting tool (use the template as a mechanism to match the data fields between the two systems).

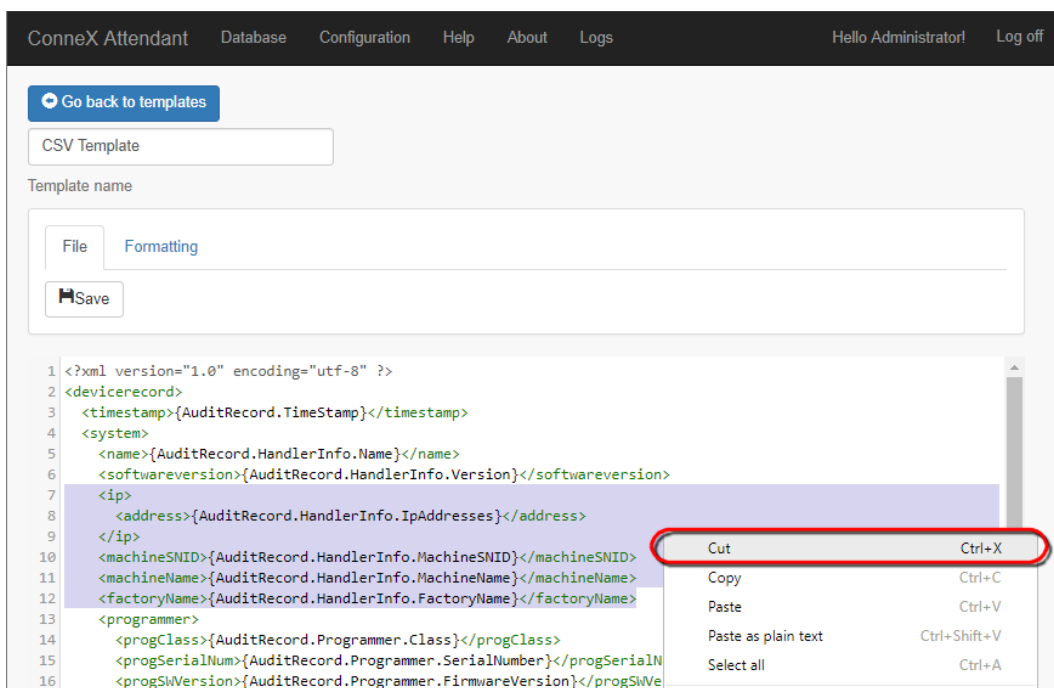
Adding and Removing Fields

Because this sample demo involves running everything on a single computer, let's remove some machine and factory-related entries from our template.

1. If our **CSV Template** is not already open, then login to ConneX Attendant, click **Configuration**, click **Modify Templates**, and then click **CSV Template**.

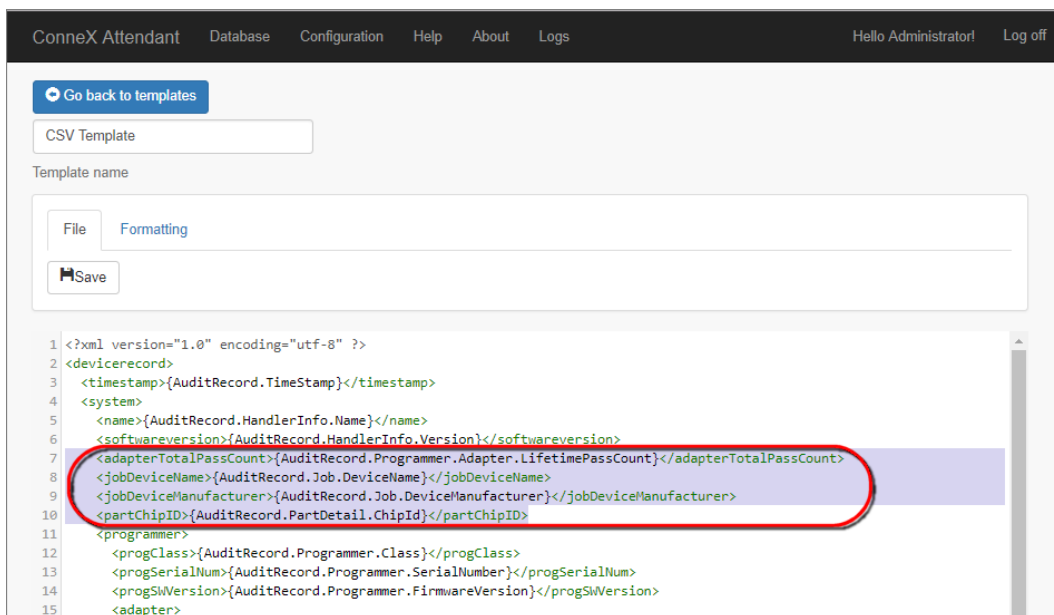


2. In the template code box at the bottom, let's remove the entire lines for the following entries:
 - `<address>{AuditRecord.HandlerInfo.IpAddresses}</address>`
 - `<machineSNID>{AuditRecord.HandlerInfo.MachineSNID}</machineSNID>`
 - `<machineName>{AuditRecord.HandlerInfo.MachineName}</machineName>`
 - `<factoryName>{AuditRecord.HandlerInfo.FactoryName}</factoryName>`

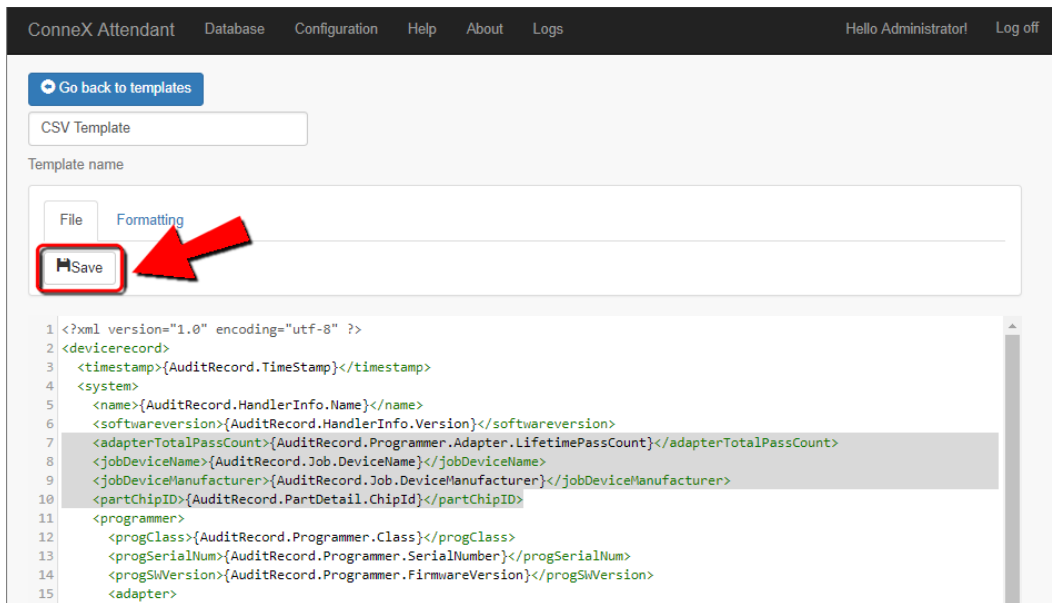


3. Now anywhere in the template code box, let's add (type or paste) the following lines:

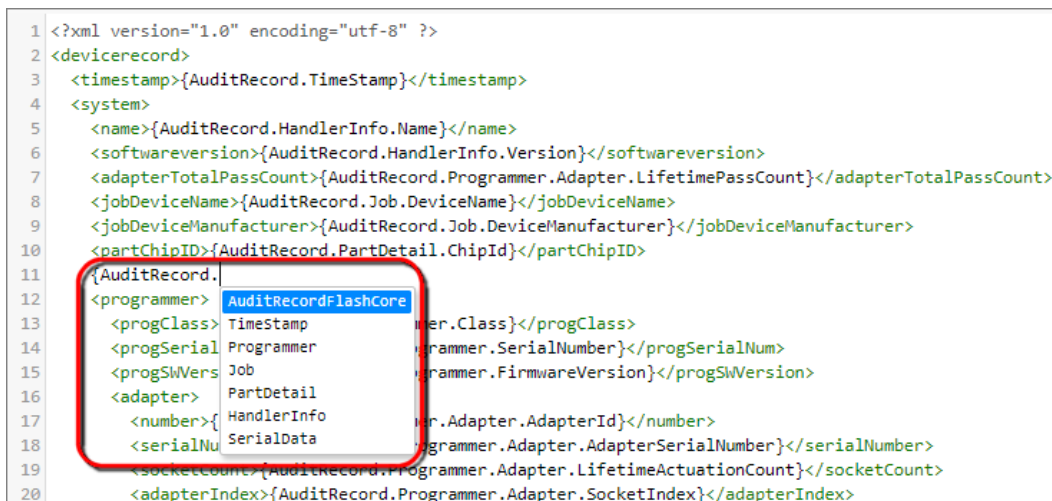
- `<adapterTotalPassCount>{AuditRecord.Programmer.Adapter.LifetimePassCount}</adapterTotalPassCount>`
- `<jobDeviceName>{AuditRecord.Job.DeviceName}</jobDeviceName>`
- `<jobDeviceManufacturer>{AuditRecord.Job.DeviceManufacturer}</jobDeviceManufacturer>`
- `<partChipID>{AuditRecord.PartDetail.ChipId}</partChipID>`



4. After defining the fields that we'll want in our audit records, let's **Save** the template.



Test This! Anywhere in the template code box, start typing the string “**{AuditRecord.**” and notice how after you type the period character, IntelliSense appears. You can leverage the IntelliSense code-completion feature to discover supported elements and auto-complete the fields in your ConneX template.



Changing the Sequence of Fields

For field sequencing of the data entries in our template (and eventually our audit records), ConneX offers full control over the template code so we can arrange the fields however we like (typically to suit the data import process and policies of our organization). For the brevity of this sample demo, let's simply copy two socket operation fields and paste them near the top of the template.

1. In the template code box for our **CSV Template**, let's scroll down and **Cut** the two entire lines:

- `<programTime>{AuditRecord.PartDetail.Result.ProgramDuration}</programTime>`
- `<verifyTime>{AuditRecord.PartDetail.Result.VerifyDuration}</verifyTime>`

The screenshot shows a code editor with XML template code. Lines 37 and 38 are highlighted in blue, and a red box is drawn around them. A context menu is open over these lines, with the 'Cut' option selected and circled in red. The code visible includes fields like <progClass>, <progSerialNum>, <progSWVersion>, <adapter>, <number>, <serialNumber>, <socketCount>, <adapterIndex>, </adapter>, </programmer>, </system>, <job>, <jobName>, <jobDescription>, <algorithmID>, </job>, <partDetails>, <uid>, <progResults>, <resultCode>, <resultCodeName>, <passPosition>, <eraseTime>, <blankCheckTime>, <programTime>, <verifyTime>, <deviceError>, and </progResults>.

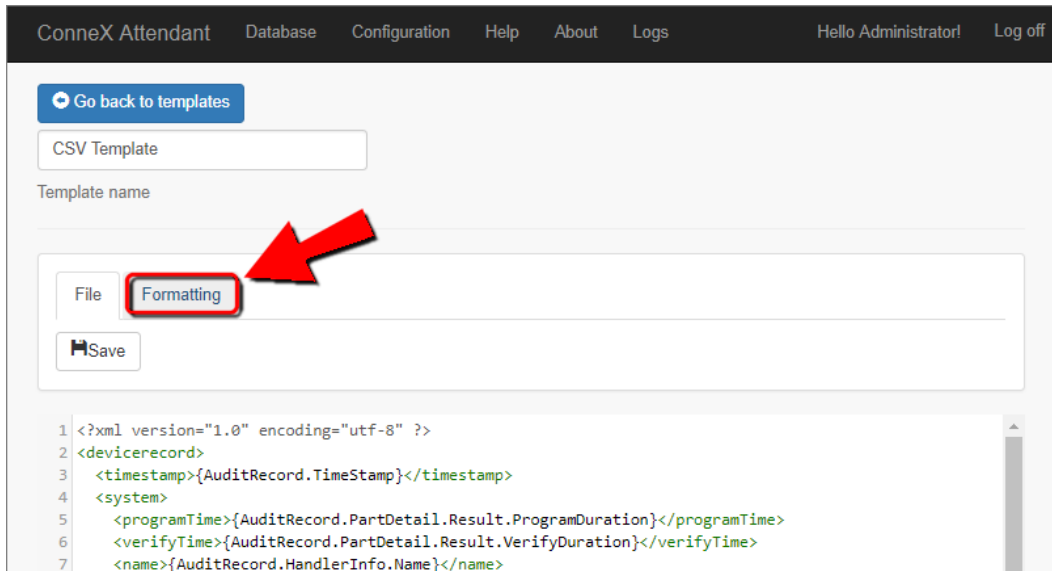
2. Now let's scroll up and **Paste** these two entries near the top of the template and click **Save**.

The screenshot shows the ConneX Attendant web interface. At the top, there's a navigation bar with 'ConneX Attendant', 'Database', 'Configuration', 'Help', 'About', 'Logs', 'Hello Administrator!', and 'Log off'. Below this, there's a 'Go back to templates' button and a search box containing 'CSV Template'. Under 'Template name', there are 'File' and 'Formatting' tabs. The 'File' tab is active, and a red box is drawn around the 'Save' button. Below the tabs, the XML code for the template is displayed. Lines 5 and 6 are highlighted in blue and circled in red, showing the two XML tags that were pasted: `<programTime>{AuditRecord.PartDetail.Result.ProgramDuration}</programTime>` and `<verifyTime>{AuditRecord.PartDetail.Result.VerifyDuration}</verifyTime>`. The rest of the code includes <?xml>, <devicerecord>, <timestamp>, <system>, <name>, <softwareversion>, <adapterTotalPassCount>, <jobDeviceName>, <jobDeviceManufacturer>, and <partChipID>.

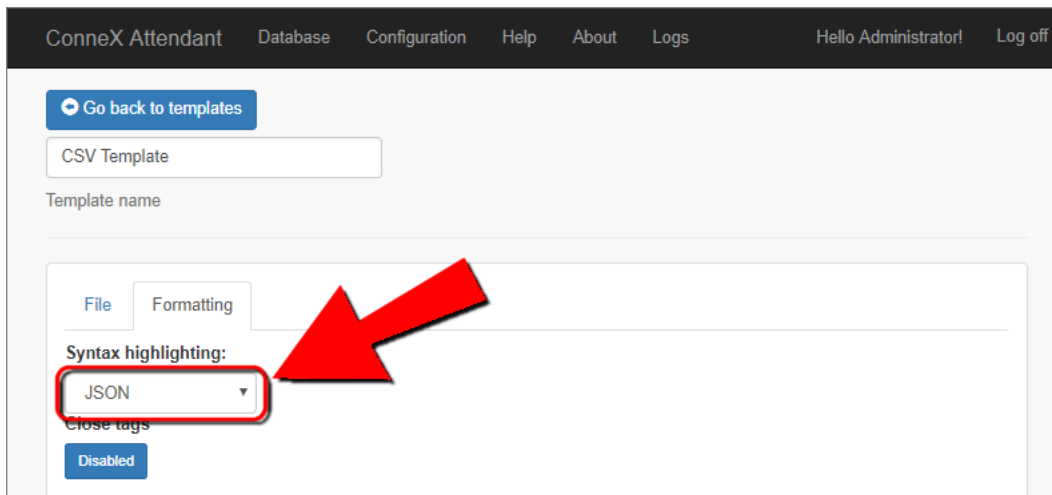
Reformatting the Template Code

After defining the field entries and their sequence in our template, let's complete our template code customizations by converting the code syntax from XML to CSV/comma-delimited. To make this process easier, we'll leverage IntelliSense in ConneX to color-code the extraneous XML elements we can delete from our CSV template. Then we'll convert the remaining entries from XML to comma-delimited.

1. Just above the template code box for our **CSV Template**, click the **Formatting** tab.

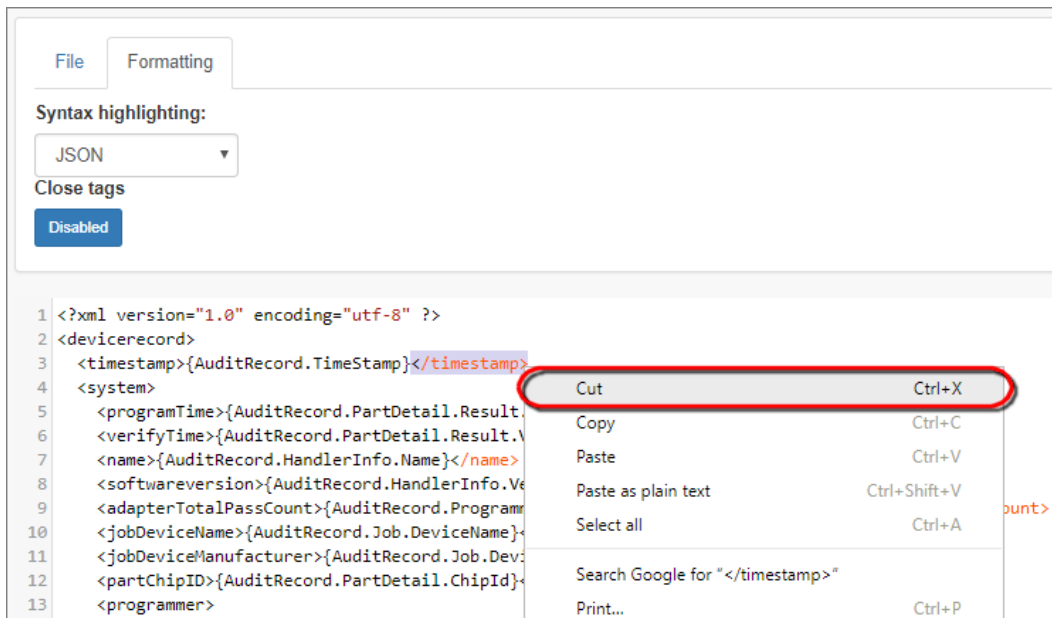


2. On the **Formatting** tab, from the **Syntax highlighting** drop-down list, select **JSON**.



The Syntax highlighting selection only affects code editing in the template code box below and not the actual file output format (which we'll change later). We selected JSON specifically because the JSON syntax does not require any closing tags or elements. So the built-in parser highlights in orange color all the extraneous XML closing tags. Selecting JSON makes it easier for us to remove the template code not needed for our eventual CSV output.

3. In the template code box, delete all of the XML closing tags in orange color (including the 'left bracket/less-than' character).



4. Verify that your template code now looks similar to the following (no closing tags!):

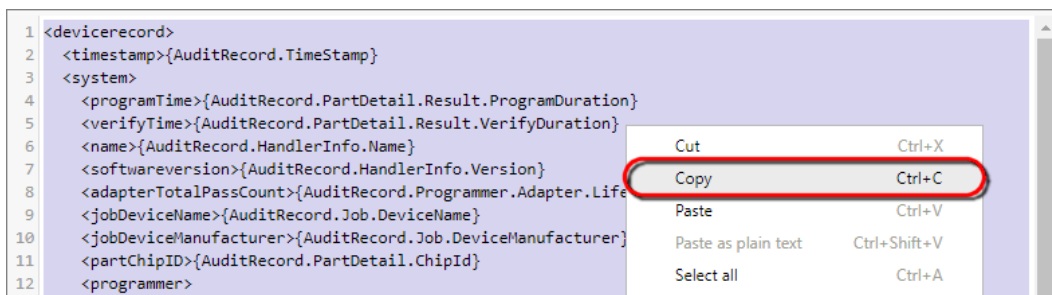
```

1 <devicerecord>
2   <timestamp>{AuditRecord.TimeStamp}
3   <system>
4     <programTime>{AuditRecord.PartDetail.Result.ProgramDuration}
5     <verifyTime>{AuditRecord.PartDetail.Result.VerifyDuration}
6     <name>{AuditRecord.HandlerInfo.Name}
7     <softwareversion>{AuditRecord.HandlerInfo.Version}
8     <adapterTotalPassCount>{AuditRecord.Programmer.Adapter.LifetimePassCount}
9     <jobDeviceName>{AuditRecord.Job.DeviceName}
10    <jobDeviceManufacturer>{AuditRecord.Job.DeviceManufacturer}
11    <partChipID>{AuditRecord.PartDetail.ChipId}

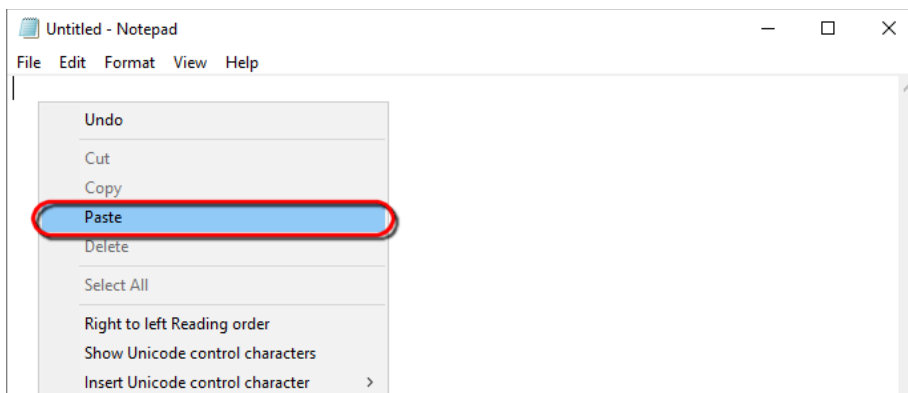
```

To convert the remaining code from XML syntax to comma-delimited, we'll copy all of the code from the template box and paste it into a new Notepad document where we'll use Find-and-Replace to edit it.

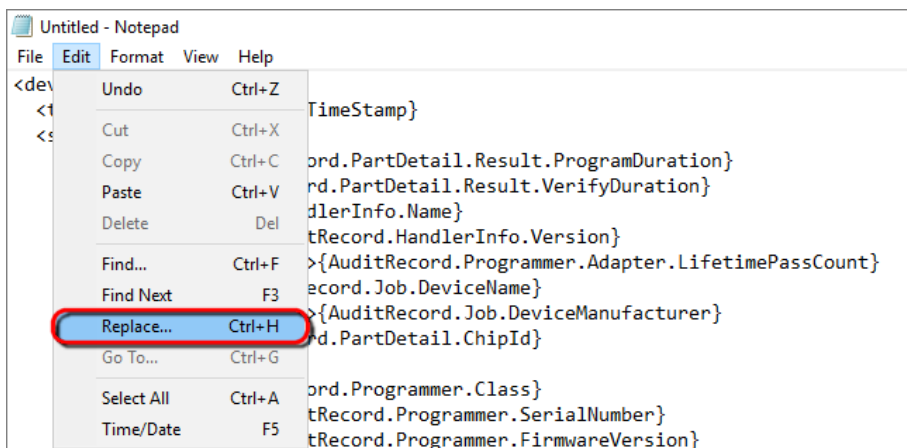
5. In the template code box, Select All of the code and **Copy** it (to the computer clipboard).



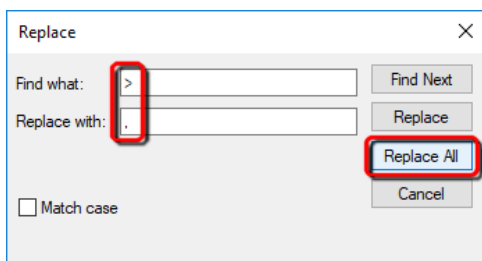
6. Start Notepad (ex. Start > Windows Accessories > Notepad) and **Paste** in the copied template code.



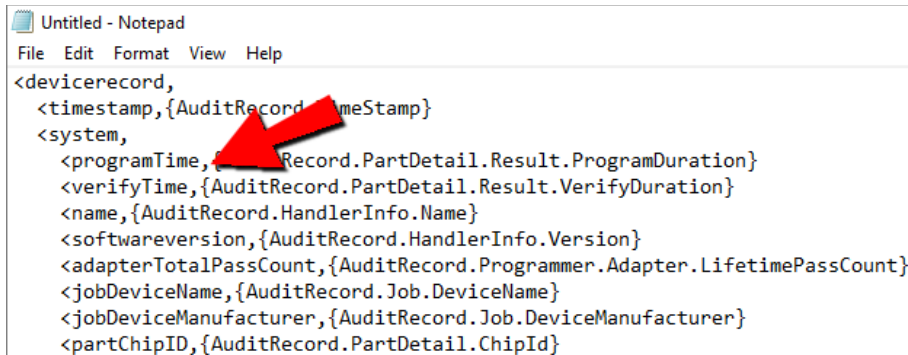
7. To convert the 'right bracket/greater-than' (>) character with a comma for our CSV, click **Edit**, and then click **Replace**.



8. In the **Replace** dialog box, in the **Find what** box, type the 'right bracket/greater-than' (>) character.
9. In the **Replace with** box, type the comma character, and then click **Replace All**.



Notepad replaces all of the 'right bracket/greater-than' (>) characters with a comma.

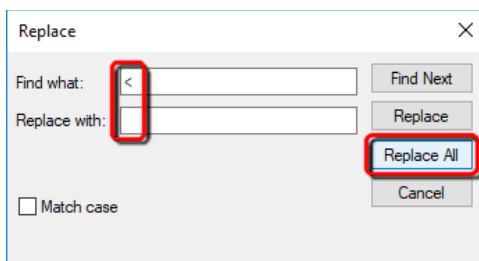


```

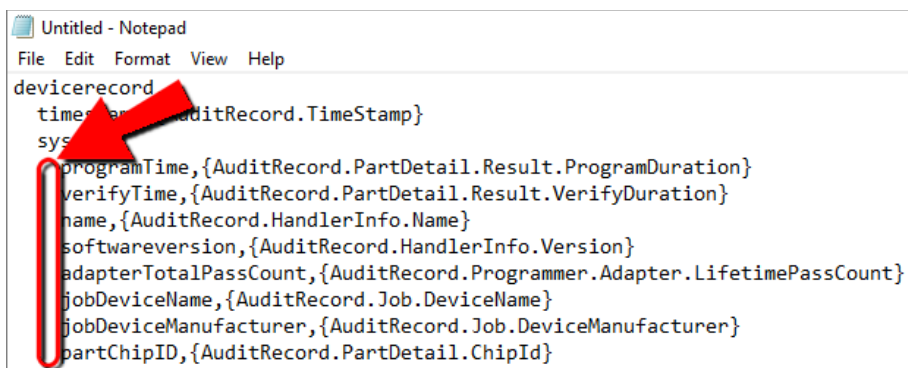
Untitled - Notepad
File Edit Format View Help
<devicerecord,
  <timestamp,{AuditRecord.Timestamp}
  <system,
    <programTime,{AuditRecord.PartDetail.Result.ProgramDuration}
    <verifyTime,{AuditRecord.PartDetail.Result.VerifyDuration}
    <name,{AuditRecord.HandlerInfo.Name}
    <softwareversion,{AuditRecord.HandlerInfo.Version}
    <adapterTotalPassCount,{AuditRecord.Programmer.Adapter.LifetimePassCount}
    <jobDeviceName,{AuditRecord.Job.DeviceName}
    <jobDeviceManufacturer,{AuditRecord.Job.DeviceManufacturer}
    <partChipID,{AuditRecord.PartDetail.ChipId}
  
```

We'll repeat these steps to remove the 'left bracket/less-than' (<) characters from our template code.

10. In the **Replace** dialog box, in the **Find what** box, type the 'left bracket/less-than' (<) character.
11. In the **Replace with** box, delete the comma character (leaving the box blank), and click **Replace All**.



Notepad replaces all of the 'left bracket/less-than' (<) characters with nothing (effectively removing the left brackets from our template).



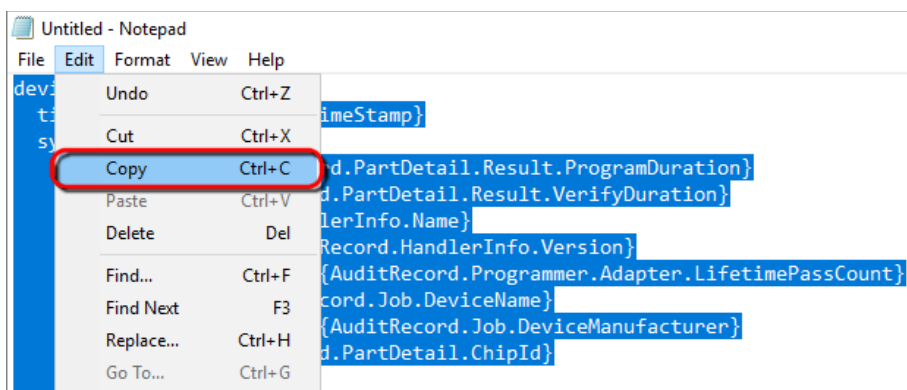
```

Untitled - Notepad
File Edit Format View Help
devicerecord
timestamp,{AuditRecord.Timestamp}
system,
  programTime,{AuditRecord.PartDetail.Result.ProgramDuration}
  verifyTime,{AuditRecord.PartDetail.Result.VerifyDuration}
  name,{AuditRecord.HandlerInfo.Name}
  softwareversion,{AuditRecord.HandlerInfo.Version}
  adapterTotalPassCount,{AuditRecord.Programmer.Adapter.LifetimePassCount}
  jobDeviceName,{AuditRecord.Job.DeviceName}
  jobDeviceManufacturer,{AuditRecord.Job.DeviceManufacturer}
  partChipID,{AuditRecord.PartDetail.ChipId}

```

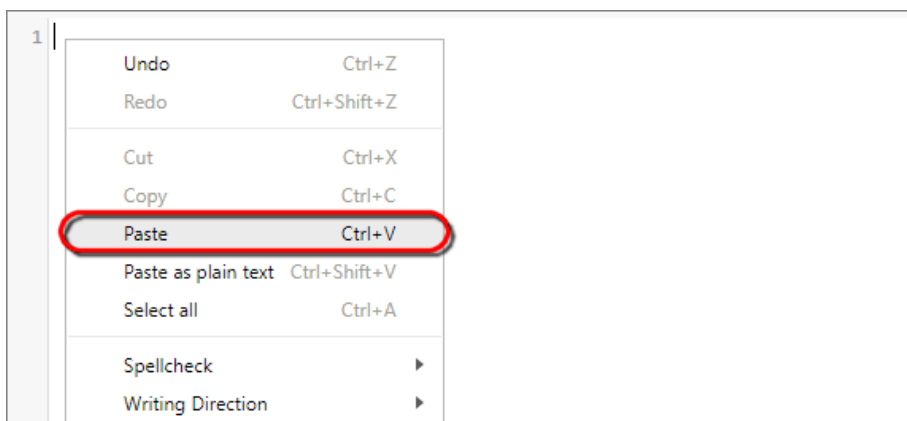
Now that we completed the process of converting our template code from XML to comma delimited, let's save the code back into our CSV Template.

12. In Notepad, click **Edit** (at the very top), and then click **Select All**.
13. Click **Edit** again, and then click **Copy**.

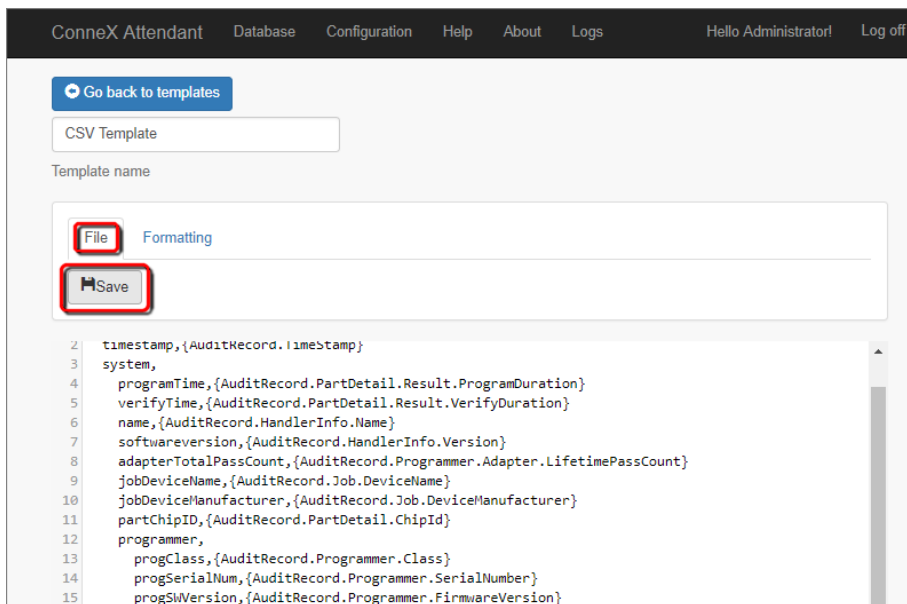


14. Return to ConneX Attendant and delete all of the existing code in the template code box.

15. Now right-click anywhere in the template code box, and then select **Paste**.



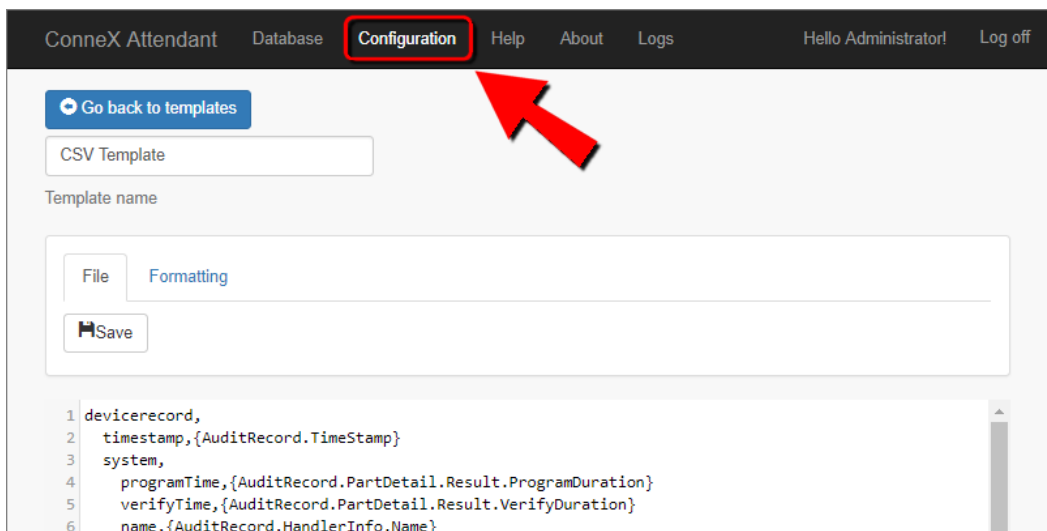
16. To save the customized code in our CSV Template, let's click the **File** tab, and then click **Save**.



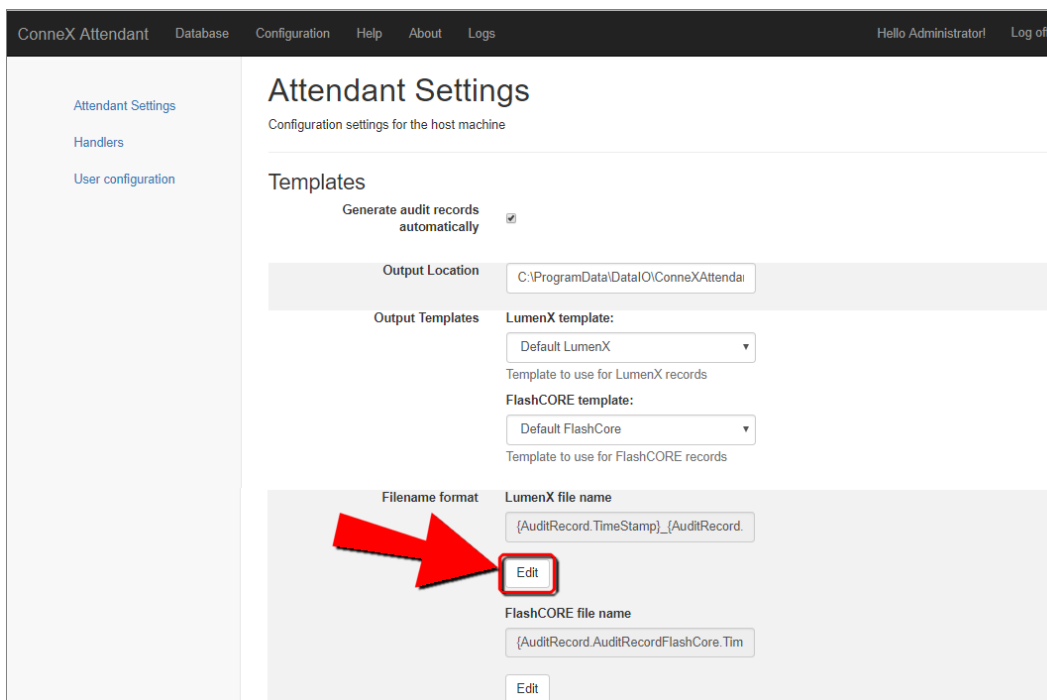
Modifying the Output Format

Now that we're done with customizing the template code, only two items remain in this Customization Chapter: changing the output file format from .XML to .CSV and changing the Output Template selection from Default LumenX to our CSV Template.

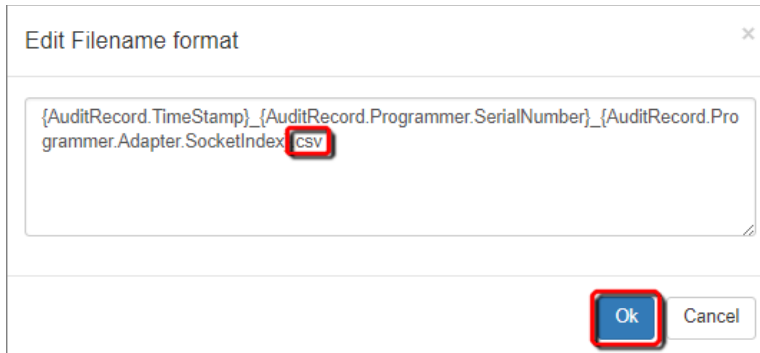
1. To change the output file format from .XML to .CSV, let's click **Configuration** in the main navigation.



2. On the **Attendant Settings** page, in the **Templates** section, under **Filename format**, click **Edit** for **LumenX file name**.



3. In the **Edit Filename format** dialog box, change the file extension from **xml** to **csv**, and then click **OK**.

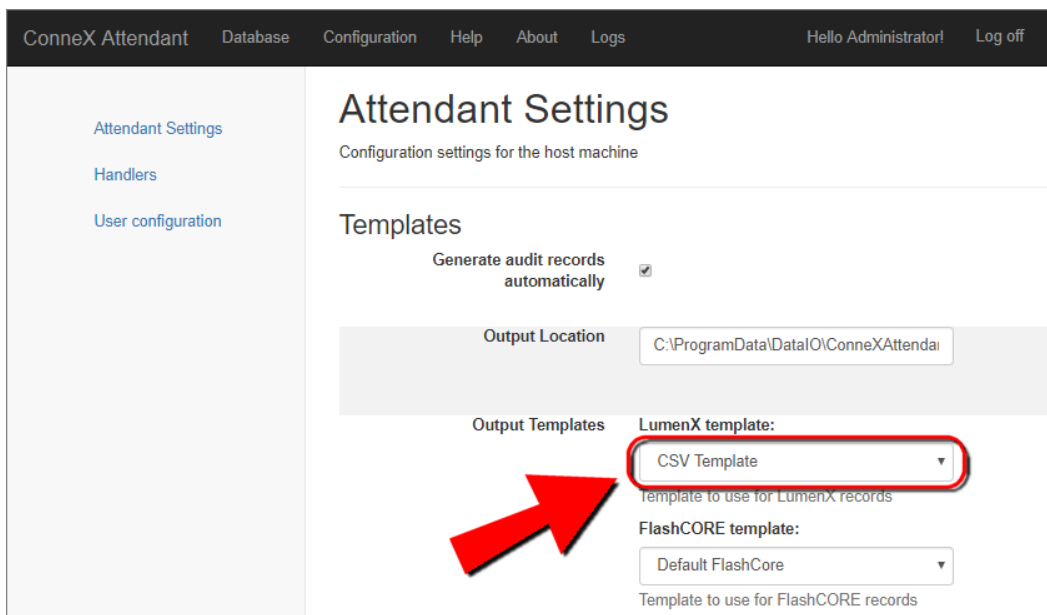


Note that in this dialog box, we can also change the filename convention to suit our particular environment and systems by updating the strings/tags before the filename extension. For example, we can re-sequence and change the variables comprising the filename here. Then as programming jobs complete and ConneX exports the data as audit records to the computer hard drive, the exported files will follow the filename convention that we set.

Selecting the Customized Template in ConneX

After modifying the template code and its output format, our template customizations are complete. Now we simply need to instruct ConneX Attendant to start using our new template for the audit records.

1. On the **Attendant Settings** page, under **Output Templates**, from the **LumenX template** drop-down list, select our **CSV Template**.



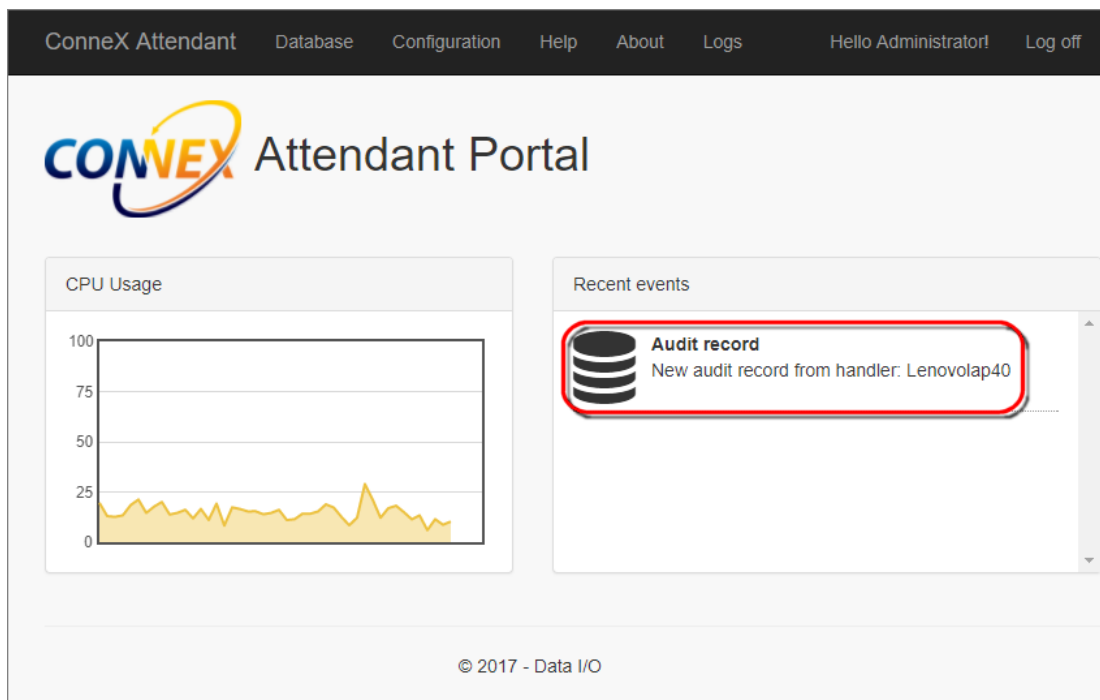
2. To save our change, let's scroll to the bottom and click **Save**.

The screenshot shows the 'Configuration' tab in the ConneX Attendant interface. The left sidebar has links for 'Attendant Settings', 'Handlers', and 'User configuration'. The main content area is titled 'Password complexity' and includes a note: 'Note: Settings changed here will not take effect until the service is restarted.' Below this, there are checkboxes for 'Require Digit', 'Require Lowercase', 'Require Non Letter or Digit', and 'Require Uppercase', all of which are currently unchecked. A 'Required Length' field is set to '6'. At the bottom, there are 'Save' and 'Revert changes' buttons. A large red arrow points to the 'Save' button, which is also highlighted with a red rectangle.

Now run a test programming job so ConneX can log the event details. Then we can check the data fields in the exported audit record(s) to see the effects of our customizations to the template.

View Customized Data Output

As before in Chapter 3, we can view the event being logged dynamically as real-time data by opening the ConneX Handler Portal page (or the Attendant Portal page).

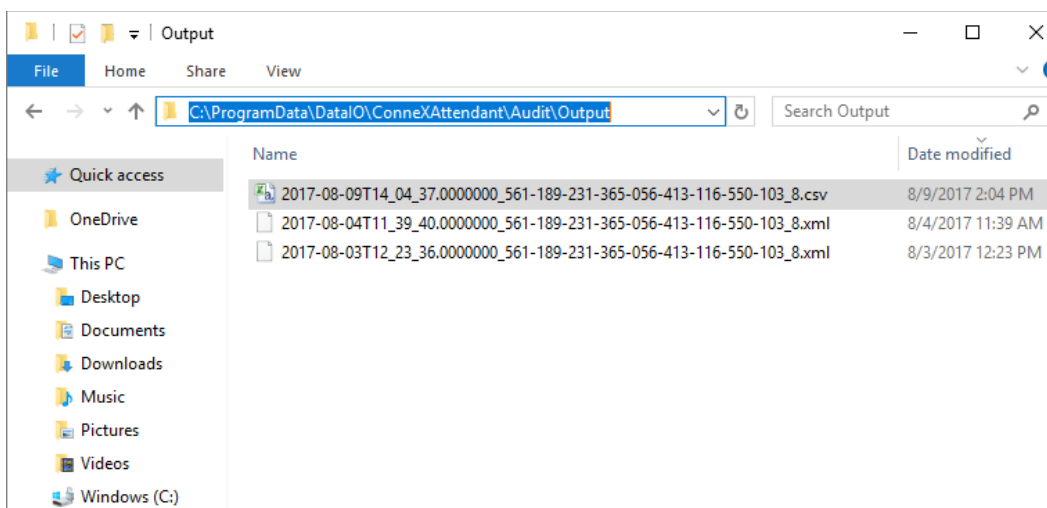


We can also view the event as static or historical data by switching over to the Handler Dashboard page (or Attendant Database page).

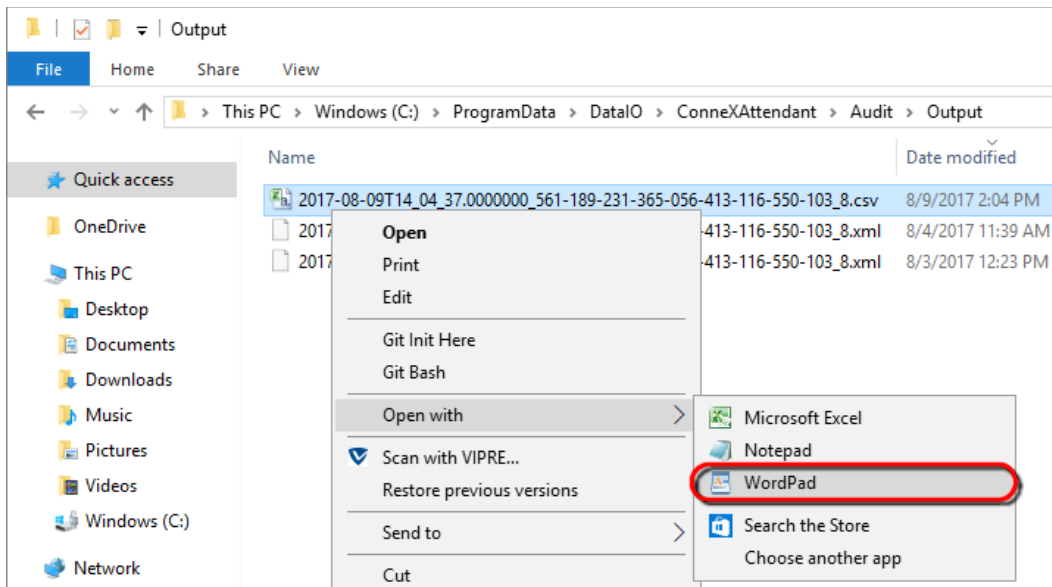
ConneX Attendant Database Configuration Help About Logs Hello Administrator! Log off		
<h2>Audit Records</h2>		
▼ Export records		
Displaying the last 100 audit records inserted		
Record count: 1		
Index	Programmer Serial	Time
1	561-189-231-365-056-413-116-550-103	8/3/2017 12:23:36 PM

But now let's see the audit record that ConneX exports to the hard drive. Because of the customizations we made to the output template, we should find a .CSV file containing comma-separated values of our data entries (rather than the default XML-formatted entries and file).

1. Start Windows/File Explorer and navigate to the **Output** folder (by default, *C:\ProgramData\DataIO\ConneXAttendant\Audit\Output*).



2. Right-click the exported CSV file, point to **Open with**, and then click **WordPad**.

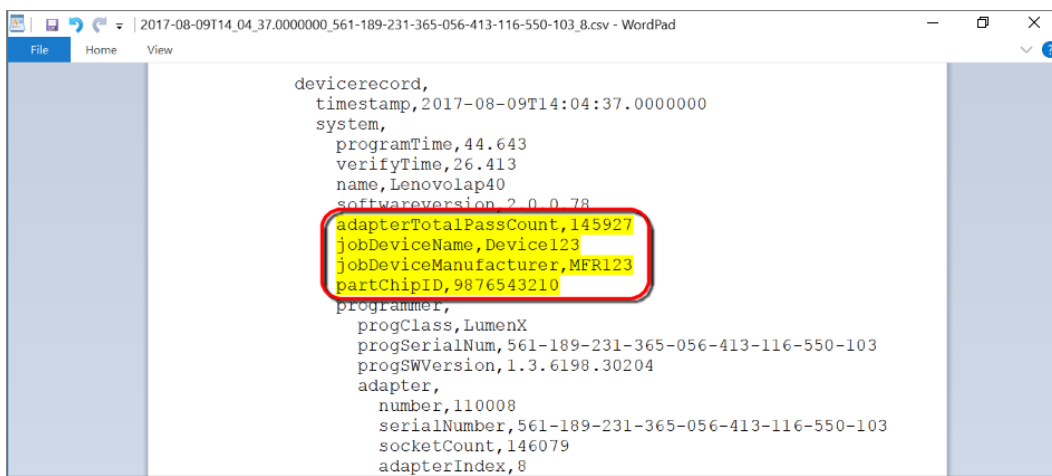


3. In WordPad, verify that the audit record does not include the following fields (because we removed them from the template):

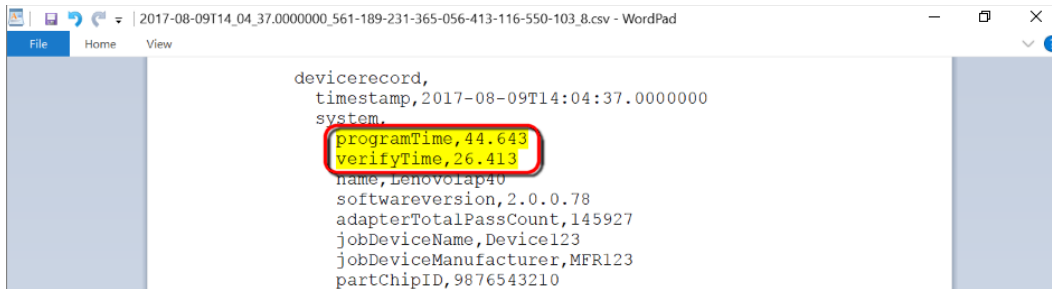
- `<address>{AuditRecord.HandlerInfo.IpAddresses}</address>`
- `<machineSNID>{AuditRecord.HandlerInfo.MachineSNID}</machineSNID>`
- `<machineName>{AuditRecord.HandlerInfo.MachineName}</machineName>`
- `<factoryName>{AuditRecord.HandlerInfo.FactoryName}</factoryName>`

4. In WordPad, verify that the audit record includes the following fields because we added them:

- `<adapterTotalPassCount>{AuditRecord.Programmer.Adapter.LifetimePassCount}</adapterTotalPassCount>`
- `<jobDeviceName>{AuditRecord.Job.DeviceName}</jobDeviceName>`
- `<jobDeviceManufacturer>{AuditRecord.Job.DeviceManufacturer}</jobDeviceManufacturer>`
- `<partChipID>{AuditRecord.PartDetail.ChipId}</partChipID>`



5. In WordPad, verify our re-sequencing of fields (**programTime** and **verifyTime** are now near the top).



6. Lastly, verify that the audit record is formatted with comma-separated values instead of XML tags.

And that concludes this Chapter on customization. We learned how to customize a ConneX template to define the data fields and format for the audit records that ConneX outputs. Then after running a test programming job, we observed the effects of our template customizations by opening an exported audit record and seeing how its data fields follow the structure and sequence of our customized template.

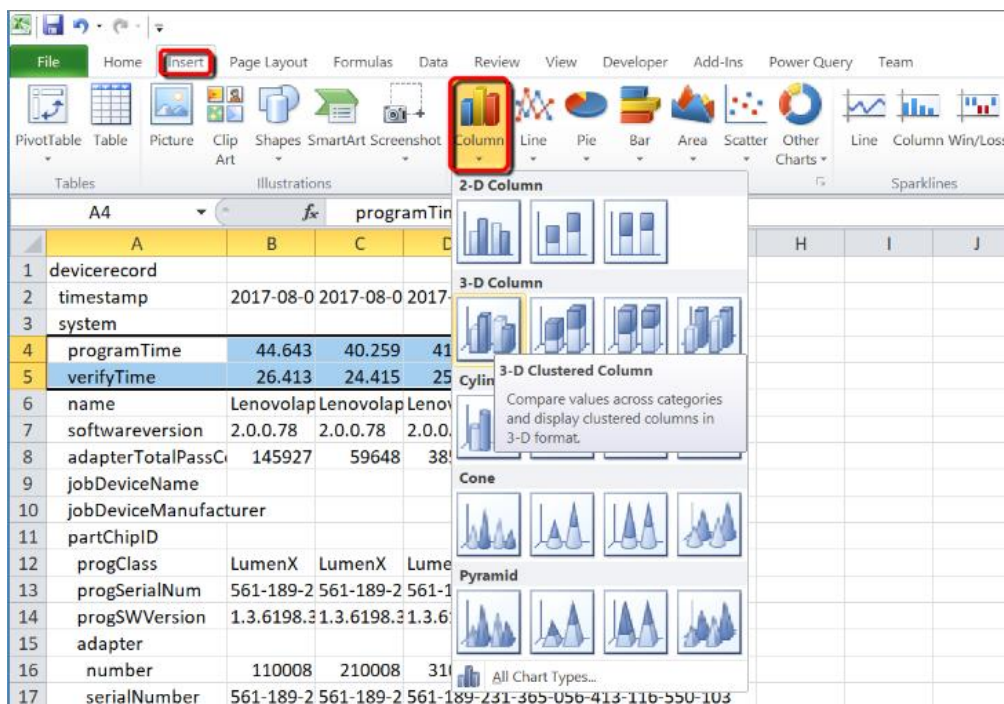
Template customization is a simple way to align the data sets between ConneX and an external reporting tool. There are other ways of course (such as taking the default ConneX output and running data transformations to convert the output to match your system), but you'll have to work with your team to make decisions about which strategy is most desirable for your particular environment.

Viewing ConneX Data with Microsoft Office Excel

While auditing and traceability are critical for regulatory compliance, we can extend the value of the raw data that ConneX provides by visualizing it with a basic program like Microsoft Office Excel or Google Sheets. Let's walk through a quick example of viewing ConneX data in an external program.

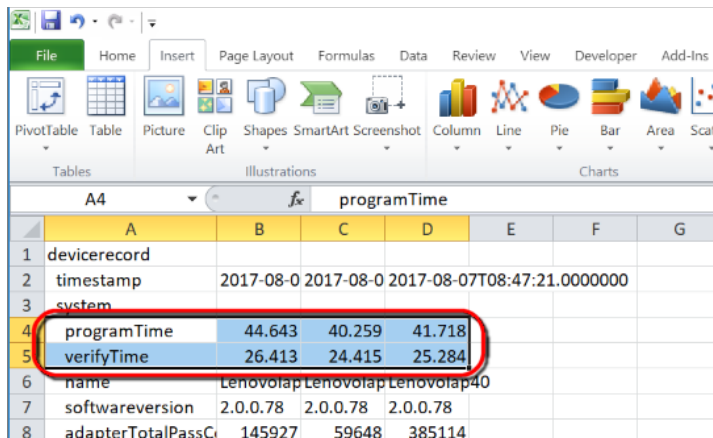
Instead of opening the CSV audit record using WordPad or some external reporting tool that may require data transformations, we can open it with Microsoft Excel and use the built-in design features in Excel to view the data. This example shows the type of insight we can gain by using ConneX data and ordinary off-the-shelf software without configuring any expensive data reporting and analytics tools.

Opening our exported CSV file with Excel is simple but not very beneficial because we'd like to spot patterns and trends across multiple programmers and programming job runs. So we can add/import the data from other job runs into new columns within our CSV file. Then in Excel we can drag to select the columns and rows of data we want to visualize, and Excel draws a line graph or pie chart for us (after clicking the **Insert** ribbon in Excel and selecting a chart from the **Charts** ribbon panel).



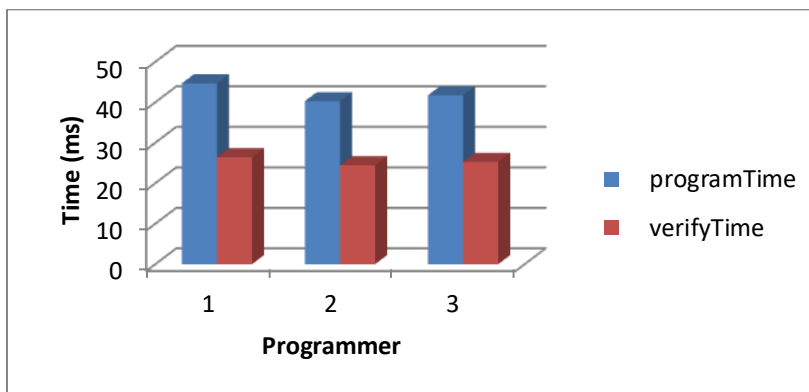
Let's suppose we want to see if any of our programmers are experiencing unusually high latency while programming. Sure we can look at the logged numerical time values pertaining to these two rows in the CSV (rows 4 and 5 in the screenshot above), but if we have multiple Handlers with multiple programmers and hundreds or thousands of programming jobs, then this type of analysis becomes increasingly inefficient and/or impossible. So let's leverage the built-in design features in Microsoft Excel to help us visualize the data. (Google Sheets offers the same basic data design/visualization features.)

In Excel, let's select the columns and rows of data for **programTime** and **verifyTime**.

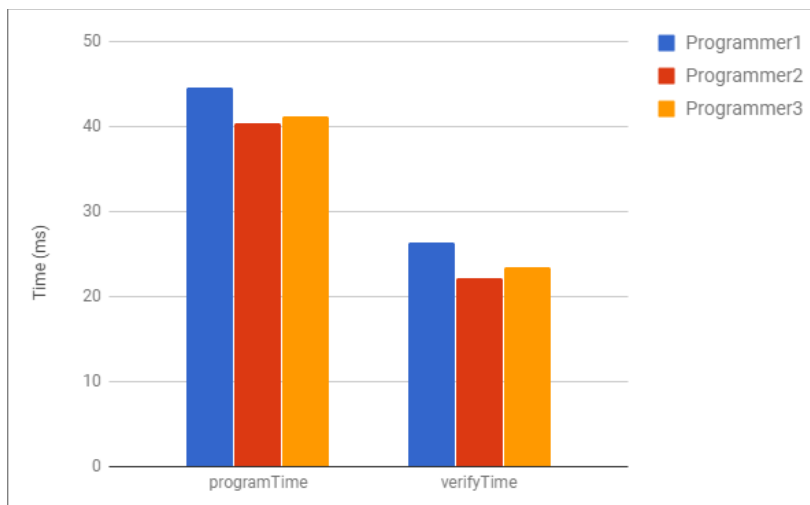


	A	B	C	D	E	F	G
1	devicerecord						
2	timestamp	2017-08-0	2017-08-0	2017-08-07T08:47:21.0000000			
3	system						
4	programTime	44.643	40.259	41.718			
5	verifyTime	26.413	24.415	25.284			
6	name	LenovoIap	LenovoIap	LenovoIap40			
7	softwareversion	2.0.0.78	2.0.0.78	2.0.0.78			
8	adapterTotalPassC	145927	59648	385114			

Now with the data selected, we can click the **Insert** ribbon in Excel and choose a chart from the **Charts** ribbon panel. For example, selecting a *3D Column Chart* produces the following data visualization.



Completing the same few steps in Google Sheets results in the following data visualization:



Experiment with ConneX data in Excel or Sheets to easily compare programming metrics such as:

- Machine activity by date/time
- Activity/Productivity by programmer (Waiting, Setup, Produce/Programming, Error..)
- Job activities as percentages
- Throughput of each job

If we're fortunate enough to have a business intelligence group or a team of data scientists, then we can use ConneX data to build all types of data visualizations and real-time dashboards to quickly analyze the programming data and draw knowledge from it.



Chapter 5: License Activation/Upgrading

In this Chapter we'll activate a ConneX license. The process is the same for upgrading to a different license. Only the computer running ConneX Handler requires license activation; ConneX Attendant is essentially free because it is functionally useless without data from ConneX Handler.

Online Activation

To activate or upgrade a ConneX license:

1. Login to ConneX Handler and click **Configuration** in the main navigation.
2. On the **Handler Settings** page, click **License** in the left navigation.

ConneX Handler Dashboard Configuration Help About Logs Hello Administrator! Log off

Handler Settings
Programmers
Programmer Discovery
User configuration
License

Handler Settings

Configuration settings for the host machine

Machine Name ?
Machine name to be inserted into the audit record

Factory Name ?
Factory name to be inserted into the audit record

Web Server
Web server configuration settings
Note: Settings changed here will not take effect until the service is restarted.

Web Port ?
Port number to connect to Handler service.

3. On the **License Settings** page, scroll down to the **License Activation** section, complete the **License ID** and **License Password** fields, and then click **Activate Online**. (The unique ID and password correspond with a specific licensing level, as defined in your contractual agreement with Data I/O.)

ConneX Handler Dashboard Configuration Help About Logs Hello Administrator! Log off

Handler Settings
Programmers
Programmer Discovery
User configuration
License

License effective end date ? 9/29/2017 Active ✓

License Length ? 60 days

License Activation

License ID ?
License ID provided at the time of purchase

License Password ?
License password provided at the time of purchase

Activate Online
Activate license using your computer's internet connection

Manual Activation

Manual Activation

If the Handler PC is offline and not connected to the Internet, then we'll have to perform manual activation instead of online activation. For manual activation, we'll need the following:

- An Internet-connected computer (for retrieving the license activation code)
- A portable storage device such as a USB Flash Drive or other memory card (for transporting the activation request from the offline Handler PC to the Internet-connected computer)

To manually activate a ConneX license:

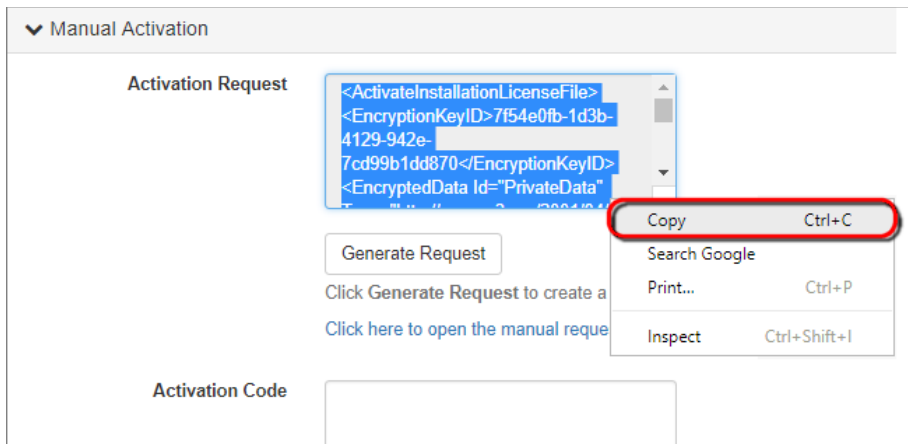
1. On the **License Settings** page, in the **License Activation** section, click **Manual Activation**.

The screenshot shows the 'ConneX Handler' interface. The top navigation bar includes 'Dashboard', 'Configuration', 'Help', 'About', and 'Logs'. The user is logged in as 'Hello Administrator!'. The left sidebar lists 'Handler Settings', 'Programmers', 'Programmer Discovery', 'User configuration', and 'License'. The main content area shows 'License effective end date' as 9/29/2017 (Active), 'License Length' as 60 days, and 'License Activation' details including 'License ID' (123456789) and 'License Password' (masked). Below these details is an 'Activate Online' button. At the bottom, the 'Manual Activation' option is highlighted with a red box and a red arrow.

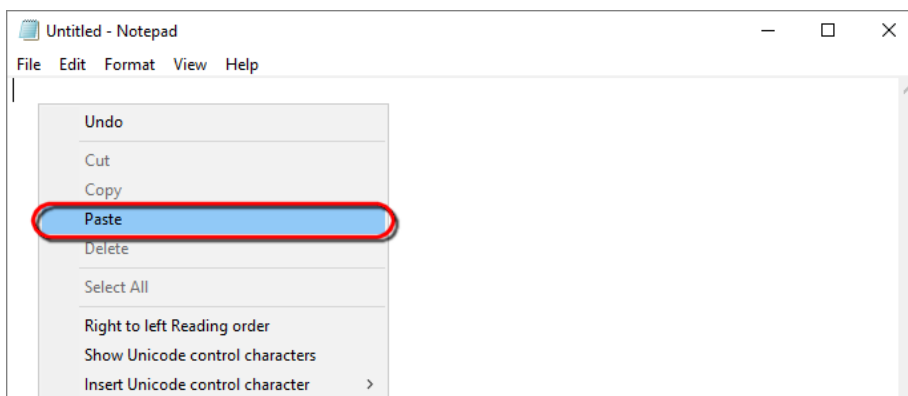
2. In the **Manual Activation** panel that expands, click **Generate Request**.

The screenshot shows the 'Manual Activation' panel. It contains an 'Activation Request' section with a large text area for the request. Below this is a 'Generate Request' button, which is highlighted with a red box. Below the button, there is a link: 'Click Generate Request to create a license request' and another link: 'Click here to open the manual request page'. At the bottom, there is an 'Activation Code' section with a text input field.

3. In the **Activation Request** box, select all of the text and **Copy** it (to the computer clipboard).



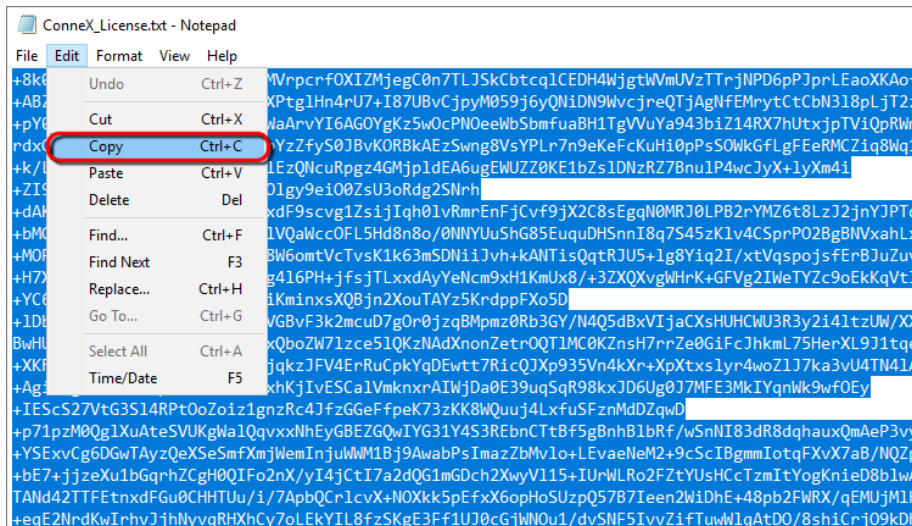
4. Open Notepad (ex. Start > Windows Accessories > Notepad) and **Paste** in the copied text.



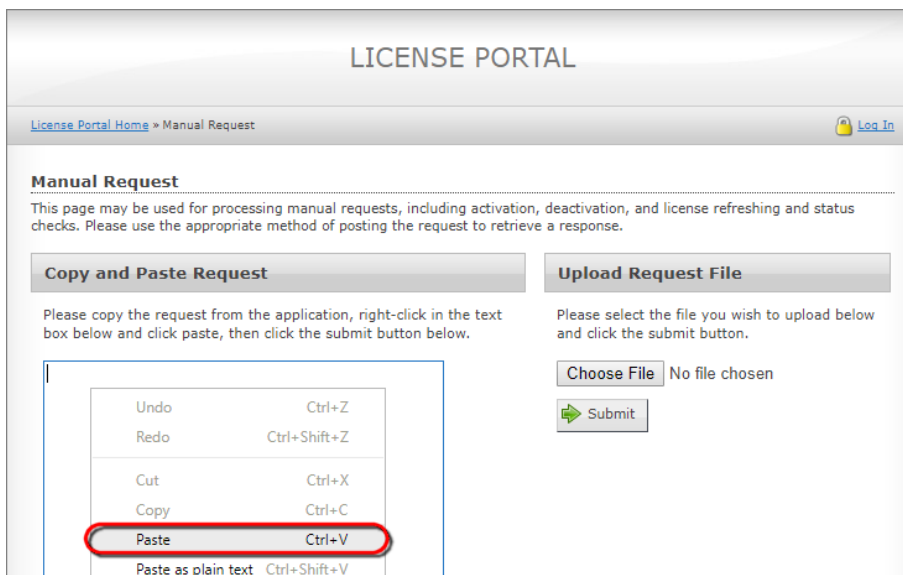
5. Insert a portable storage device into the Handler PC.
6. In Notepad, **Save** the text file to the external storage device.



7. Remove the storage device from the Handler PC and insert it into an Internet-connected computer.
8. On the Internet-connected computer, start Windows/File Explorer and navigate to the external storage device.
9. Open the text file and **Copy** all of its contents (to the computer clipboard).



10. On the Internet-connected computer, open a browser to <https://activate.dataio.com/customers/ManualRequest.aspx>.
11. On the **License Portal** page, **Paste** the activation request code into the box.



12. To send the activation request code, click **Submit** at the bottom of the page.

LICENSE PORTAL

License Portal Home » Manual Request [Log In](#)

Manual Request

This page may be used for processing manual requests, including activation, deactivation, and license refreshing and status checks. Please use the appropriate method of posting the request to retrieve a response.

Copy and Paste Request

Please copy the request from the application, right-click in the text box below and click paste, then click the submit button below.

```
4bZynCvRZWbJ2EEAAGKGI0QsDe8HV1ciGkvsIq8x
EQ8TRZ7Ja06+wh7fwk51AiXityTXqvSLvwFidr4+t
Rv7Mka/YTIGSbklIBH7zatC+7ohEPG1LZMqMD3Ey0
YFR/+GZPaHCxtjz9Gps8csGbsdMrU92He2bP4fe/R
ap6hQAf0dw5f7Th2fq+KZhPg==</CipherValue>
</CipherData></EncryptedData><Signature
xmlns="http://www.w3.org/2000/09/xmldsig#
">
<SignatureValue>kHr7c5THKgj5tXx0LGqydnh7F
JVPvEo5FPpoPiJIS2K0VwEMTosf0ri30NumMfawdL
EYTVguPT77dAvXJxRgK15sDohTmOeodBbs0WzNt65
D5yUe3/hBKuE2iCdH2ZjB14NU+cCD/LYxTWZcrNi0
TbRZ6109+jyKuwHgf95Mb4=</SignatureValue>
</Signature>
</ActivateInstallationLicenseFile>
```

Submit

Upload Request File

Please select the file you wish to upload below and click the submit button.

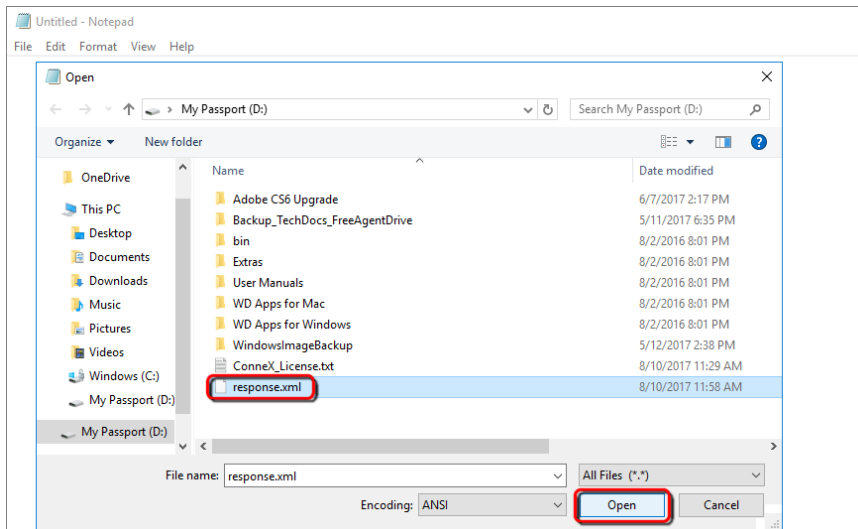
Choose File No file chosen

Submit

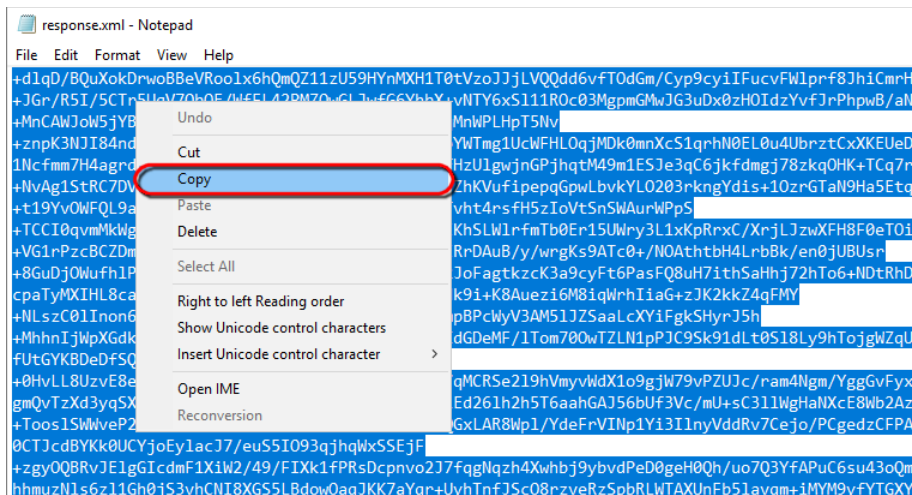
13. On the **Response** page, click **Download** and save the activation response code file.
14. Start Windows/File Explorer and copy the response code file to the external storage device.
15. Remove the external storage device from the Internet-connected computer.

To activate the ConneX license using the response code file:

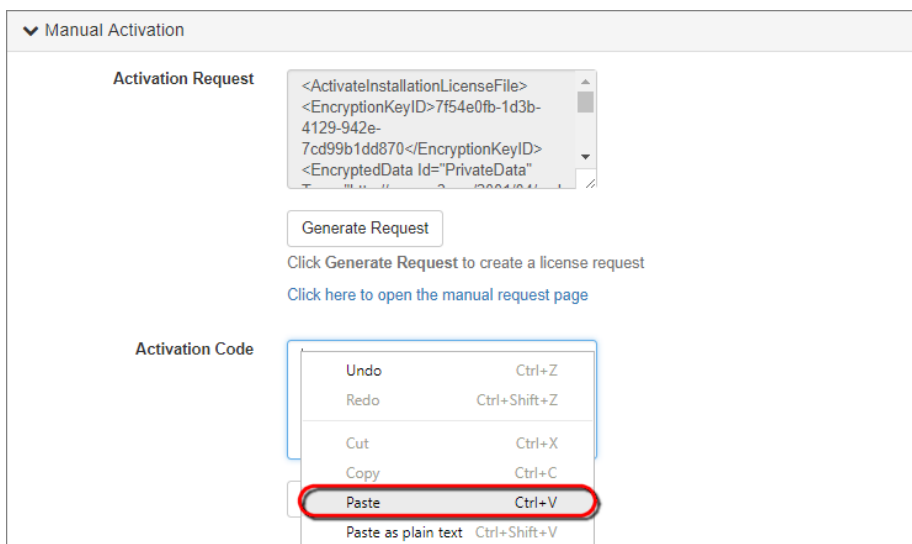
1. On the Handler PC, insert the external storage device containing our activation response code.
2. Start Notepad and **Open** the response code file (**response.xml**).



3. Select all of the response code text and **Copy** it (to the computer clipboard).



4. Return to the ConneX browser window.
5. In the **License Activation** section, **Paste** the response code into the **Activation Code** box.



6. Now finally, let's click **Apply license**.

Manual Activation

Activation Request

```
<ActivateInstallationLicenseFile>  
<EncryptionKeyID>7f54e0fb-1d3b-  
4129-942e-  
7cd99b1dd870</EncryptionKeyID>  
<EncryptedData Id="PrivateData"  
7f54e0fb-1d3b-4129-942e-7cd99b1dd870
```

Generate Request

Click Generate Request to create a license request

[Click here to open the manual request page](#)

Activation Code

```
njUCFS67uErcSD731aLx0qx36xnu8  
NN6WVZ+gxFEVys=  
</SignatureValue>  
</Signature>  
</ActivateInstallationLicenseFile>
```

Apply license

7. Verify that the activation code worked.

Manual Activation

Activation Request

Generate Request

Click Generate Request to create a license request

[Click here to open the manual request page](#)

Activation Code

```
Qv6Lu/Vxdm0d561ym9h9gWBkN5S5  
IDaNEt1ClyjN69i9J6LdiPAFvU3hF4e  
FiLOBEY4ojw=</SignatureValue>  
</Signature>  
</ActivateInstallationLicenseFile>
```

Apply license

License applied successfully

Appendix A: Audit Record Details/Event Reference

This section lists all of the events and details that ConneX may log in an audit record (not all fields appear in every audit record).

LumenX Programming Details

Event/Statistic	Description
TimeStamp	Date and Time (in UTC) of the programming event.
Programmer.Class	Type of programming unit (LumenX or FlashCORE III).
Programmer.FirmwareVersion	Firmware version of the programming unit.
Programmer.SerialNumber	Serial number of the programming unit.
Programmer.SystemVersion	System version of the programming unit.
Programmer.ProgrammerIP	IP address of the programming unit.
Programmer.Adapter.AdapterId	Unique ID of the socket adapter on the programming unit.
Programmer.Adapter.AdapterSerialNumber	Serial number of the socket adapter on the programming unit.
Programmer.Adapter.CleanCount	Number of times the "clean adapter module" reminder message was displayed to Operators.
Programmer.Adapter.LifetimeActuationCount	Total number of times the socket adapter is mechanically actuated over the life of the adapter.
Programmer.Adapter.LifetimeContinuityCount	Total number of devices that ran continuity check over the life of the adapter.
Programmer.Adapter.LifetimeContinuityFailCount	Total number of devices that failed continuity check over the life of the adapter.
Programmer.Adapter.LifetimeFailCount	Total number of devices that failed to complete all operations of a job over the life of the adapter.
Programmer.Adapter.PassCount	Total number of devices that passed all operations of a job over the life of the adapter.
Programmer.Adapter.SocketIndex	Index number of a particular socket adapter.
Programmer.Adapter.AdapterState	The state of the socket adapter.
Job.AlgorithmID	Unique ID that specifies the particular algorithm used in the job.
Job.JobID	Unique ID that specifies a particular job.



Job.JobName	Name of the job.
Job.JobDescription	Description of the job.
Job.DeviceName	Name of the device.
Job.DeviceManufacturer	Name of the device manufacturer.
PartDetail.ChipID	Unique chip ID on the device.
PartDetail.RawChipID	Raw chip ID on the device.
PartDetail.Result.SocketIndex	Index number of the socket adapter in which the device was placed and programmed.
PartDetail.Result.Code	Unique status code that represents Pass, Fail, or Other (see below).
PartDetail.Result.CodeName	Name of the result code/status.
PartDetail.Result.ProgramDuration	Amount of time elapsed to complete programming.
PartDetail.Result.VerifyDuration	Amount of time elapsed to verify programming.
PartDetail.Result.Times.Time	Total time minus the time for Blank Check, Erase, Program, and Verify operations.
PartDetail.Result.AlgoDeviceDetails.CID	Unique Chip ID correlating the algorithm with the device.
PartDetail.Result.BlankCheckDuration	Amount of time elapsed to perform the Blank Check operation.
PartDetail.Result.EraseDuration	Amount of time elapsed to perform the Erase operation.
PartDetail.Result.ErrorMessage	Specific message describing the error.
HandlerInfo.name	Name of the ConneX Handler computer.
HandlerInfo.IpAddresses	IP address of the ConneX Handler computer.
HandlerInfo.Version	Version number of the ConneX Handler computer.
SerialData.Data	Serialization pattern to be programmed into devices.

FlashCORE III Programming Details

Event/Statistic	Description
TimeStamp	Date and Time (in UTC) of the FlashCORE programming event.
SystemID	Unique ID of the FlashCORE system.
SystemManagementID	Unique management ID of the FlashCORE system.
FlashCoreJob	Name of the FlashCORE job.
FlashCoreIP	IP address of the host computer where the FlashCORE programming unit is installed.
FlashCoreSystemIP	IP address of the FlashCORE system.
FlashCoreProgrammerIP	IP address of the FlashCORE programming unit.
FlashCoreSystemFirmware	Firmware version of the FlashCORE programming unit.
FlashCoreSystemName	Name of the FlashCORE programming unit.
FlashCoreSystemAddressAndPort	IP address and port number of the FlashCORE system.
FlashCoreImageChecksum	Checksum result/value of the image to be programmed into the device.
FlashCoreJobDescription	Description of the job.
FlashCoreDeviceManufacturer	Name of the FlashCORE device manufacturer.
FlashCoreDeviceName	Name of the FlashCORE device.
FlashCoreDataFilename	Name of the image file containing the software to program into the hardware device.
FlashCoreTotalPassedDevices	Total number of devices that passed all operations of a job over the life of the FlashCORE adapter.
FlashCorePassQuantity	Target number of devices to program and pass verification (as set by Operator at time of run).
FlashCoreBackplaneBoardRevision	Version of the backplane board in the FlashCORE programming unit.
FlashCoreControllerBoardRevision	Version of the controller board in the FlashCORE programming unit.
FlashCoreWaveformBoardRevision	Version of the waveform board in the FlashCORE programming unit.
FlashCorePssVersion	Version of the host computer where the



	FlashCORE programming unit is installed.
FlashCorePssHostName	Computername of the host computer where the FlashCORE programming unit is installed.
FlashCorePssHostID	Unique ID of the host computer where the FlashCORE programming unit is installed.
FlashCoreAdapterNumber	Unique ID of the FlashCORE adapter on the programming unit.
FlashCoreAdapterSerialNumber	Serial number of the FlashCORE adapter on the programming unit.
FlashCoreAdapterSocketIndex	Index number of a particular socket adapter.
AdapterAdditionalAttributes	Other properties of the socket adapter.
ResultCode	Unique status code that represents Pass, Fail, or Other (see below).
ResultCodeName	Name of the result code/status.

Result Codes

The following table defines each possible result code.

Result Code	Description
-1	Socket ignored-- the system was not instructed to use the socket.
0	Pass (Programming succeeded).
1	Failed continuity check (typically retried by Handler).
2	Failed current check.
3	Failed electronic identification check.
4	Failed device erase operation.
5	Failed blank check operation.
6	Failed programming.
7	Failed verification.
8	Failed reading device.
9	General error.