TECH NOTE

HOW-TO GUIDE

Importing tag data in CSV format into Canary

This Technical Note contains all the information required to import tag data that is saved in CSV format into a Dataset through the Canary Admin application.

21/05/2021 – Ver. 1.0.0

KNOW YOUR LIMITLESS

TOC

1. Introduction

Canary supports the import of tag data that is stored in CSV format. This technical note describes the principal points to follow for a successful import.

The Canary System consists out of three components, Data Collection, Data Storage and Data Analytics. The component that will be utilized in this document is the Data Collection component, where it will be shown how to use the CSV Collector. Canary Collectors can log data from multiple data sources, including OPC UA/DA, MQTT, and various other SCADA solutions.

To import tag data in CSV format into Canary, it is expected that you have the Canary Admin application installed and running on your computer.

There are 3 main steps that you need to follow to utilize the CSV import service:

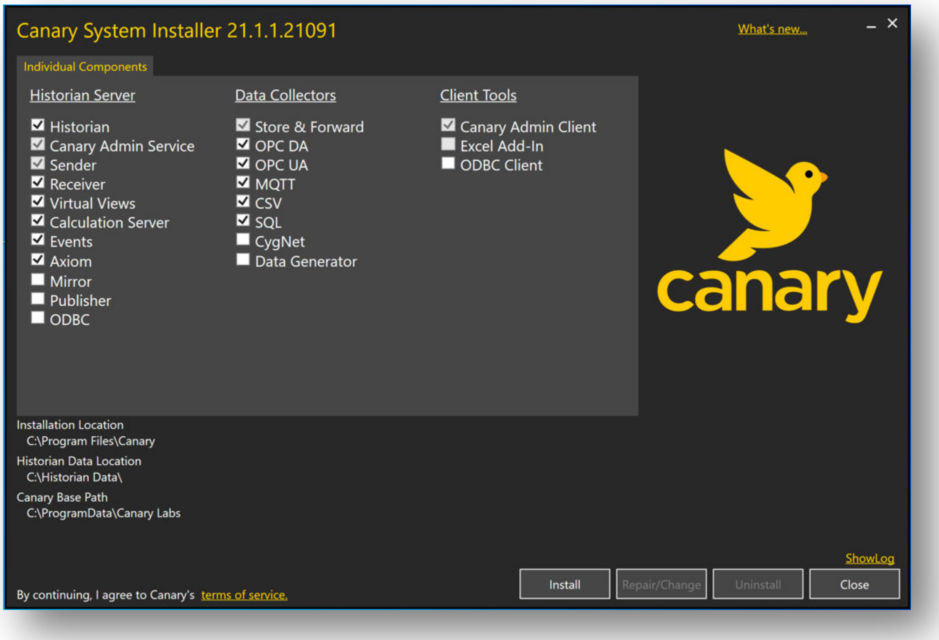
1. Install CSV Data Collector component
2. Create config files
3. Specify CSV files to import

These steps are described in greater detail in the following sections

2. Install CSV Data Collector component

The CSV import service is available in the Canary Collectors options. The Canary CSV Collector component need to be installed to be able to access this service:

1. Launch the Canary Admin installer. If a complete install has been done, you can skip this step. If not, you need to customize the current install to allow this feature



Canary System Installer 21.1.1.21091

Individual Components

Historian Server
- ☑ Historian
- ☑ Canary Admin Service
- ☑ Sender
- ☑ Receiver
- ☑ Virtual Views
- ☑ Calculation Server
- ☑ Events
- ☑ Axiom
- ☐ Mirror
- ☐ Publisher
- ☐ ODBC

Data Collectors
- ☑ Store & Forward
- ☑ OPC DA
- ☑ OPC UA
- ☑ MQTT
- ☑ CSV
- ☑ SQL
- ☐ CygNet
- ☐ Data Generator

Client Tools
- ☑ Canary Admin Client
- ☐ Excel Add-In
- ☐ ODBC Client

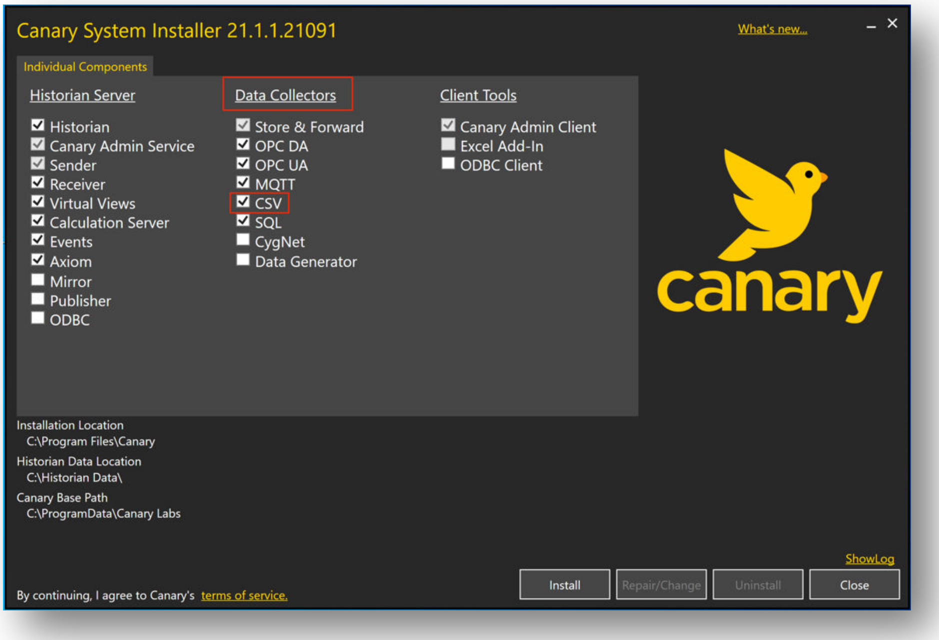
Installation Location
C:\Program Files\Canary

Historian Data Location
C:\Historian Data\

Canary Base Path
C:\ProgramData\Canary Labs

By continuing, I agree to Canary's terms of service.

2. Enable the CSV option under the "Data Collectors" tab. Also make sure you at least have the Historian, Canary Admin Service and Canary Admin Client options ticked as well



Canary System Installer 21.1.1.21091

Individual Components

Historian Server
- ☑ Historian
- ☑ Canary Admin Service
- ☑ Sender
- ☑ Receiver
- ☑ Virtual Views
- ☑ Calculation Server
- ☑ Events
- ☑ Axiom
- ☐ Mirror
- ☐ Publisher
- ☐ ODBC

Data Collectors
- ☑ Store & Forward
- ☑ OPC DA
- ☑ OPC UA
- ☑ MQTT
- ☑ CSV
- ☑ SQL
- ☐ CygNet
- ☐ Data Generator

Client Tools
- ☑ Canary Admin Client
- ☐ Excel Add-In
- ☐ ODBC Client

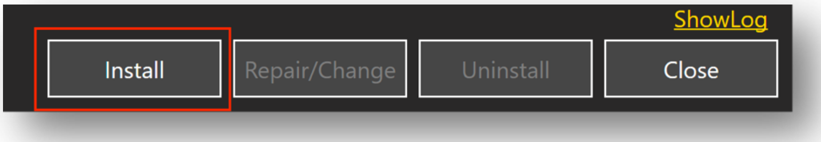
Installation Location
C:\Program Files\Canary

Historian Data Location
C:\Historian Data\

Canary Base Path
C:\ProgramData\Canary Labs

By continuing, I agree to Canary's terms of service.

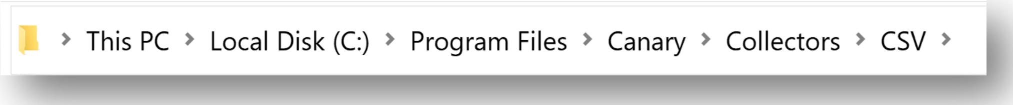
3. Click on the "Install" button. Once it has completed the installation, the next step is to create new config files



3. Create config files

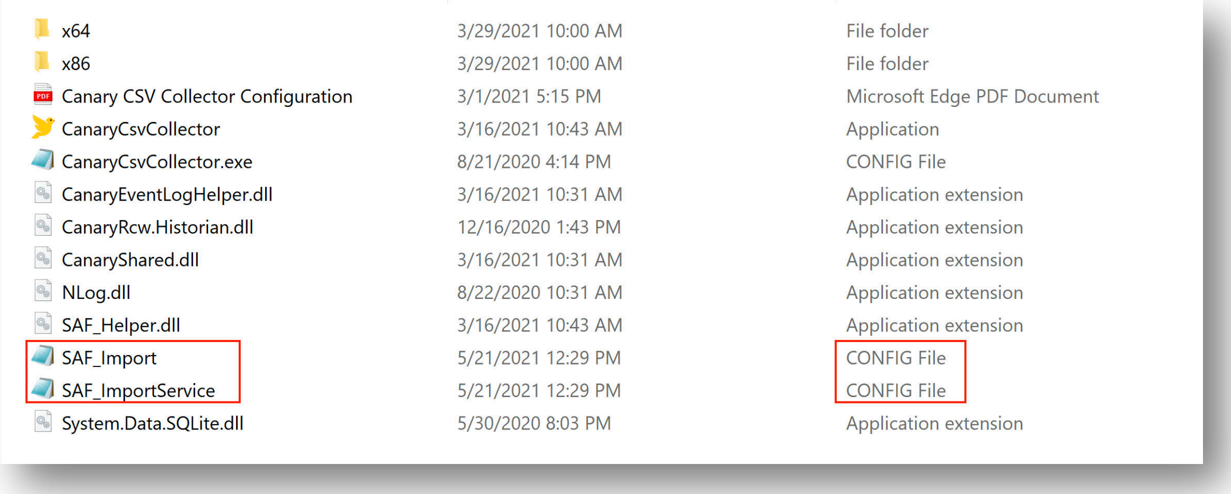
Two config files need to be created to tell the data collector how to import the data from the CSV files into a dataset located in a historian.

1. Go to the Program Files folder in your C:\ drive and navigate to Canary > Collectors > CSV



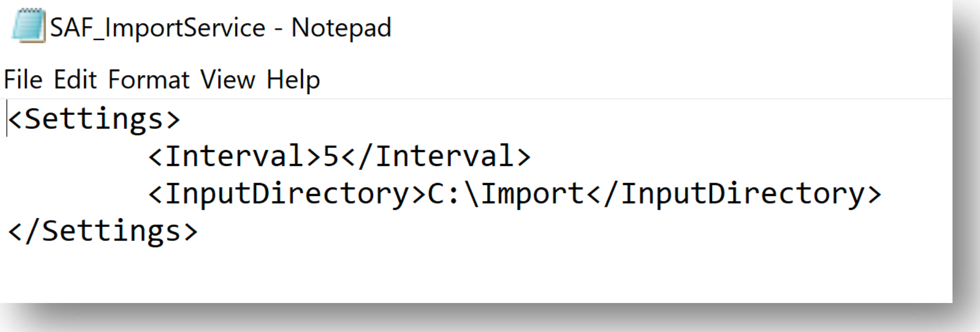
2. In this folder, you need to create the config files. The first file is called *SAF_ImportService.config*

In this config file you need to specify the time in seconds the service will check a directory for a new CSV file and in which directory to search the import files. The service can be configured to monitor multiple directories if needed. This file needs to reside in the same folder as the SAF_ImportService.exe



x64	3/29/2021 10:00 AM	File folder
x86	3/29/2021 10:00 AM	File folder
Canary CSV Collector Configuration	3/1/2021 5:15 PM	Microsoft Edge PDF Document
CanaryCsvCollector	3/16/2021 10:43 AM	Application
CanaryCsvCollector.exe	8/21/2020 4:14 PM	CONFIG File
CanaryEventLogHelper.dll	3/16/2021 10:31 AM	Application extension
CanaryRcw.Historian.dll	12/16/2020 1:43 PM	Application extension
CanaryShared.dll	3/16/2021 10:31 AM	Application extension
NLog.dll	8/22/2020 10:31 AM	Application extension
SAF_Helper.dll	3/16/2021 10:43 AM	Application extension
SAF_Import	5/21/2021 12:29 PM	CONFIG File
SAF_ImportService	5/21/2021 12:29 PM	CONFIG File
System.Data.SQLite.dll	5/30/2020 8:03 PM	Application extension

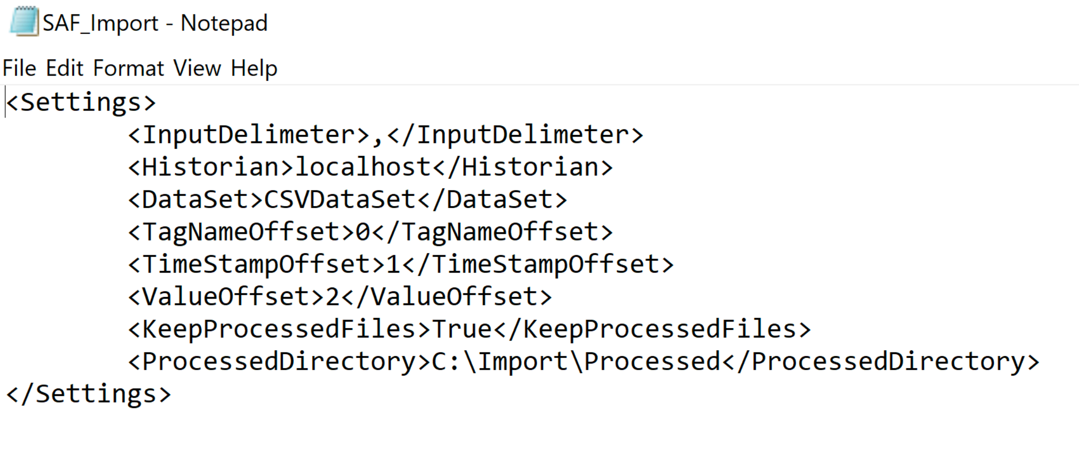
Below is an example of how the config file should look. The <Interval> specifies the time in seconds the service will check for new files and the <InputDirectory> specifies the directory in which the service should monitor for new files. Notice how there are 2 input directories specified



```
<Settings>
        <Interval>5</Interval>
        <InputDirectory>C:\Import</InputDirectory>
</Settings>
```

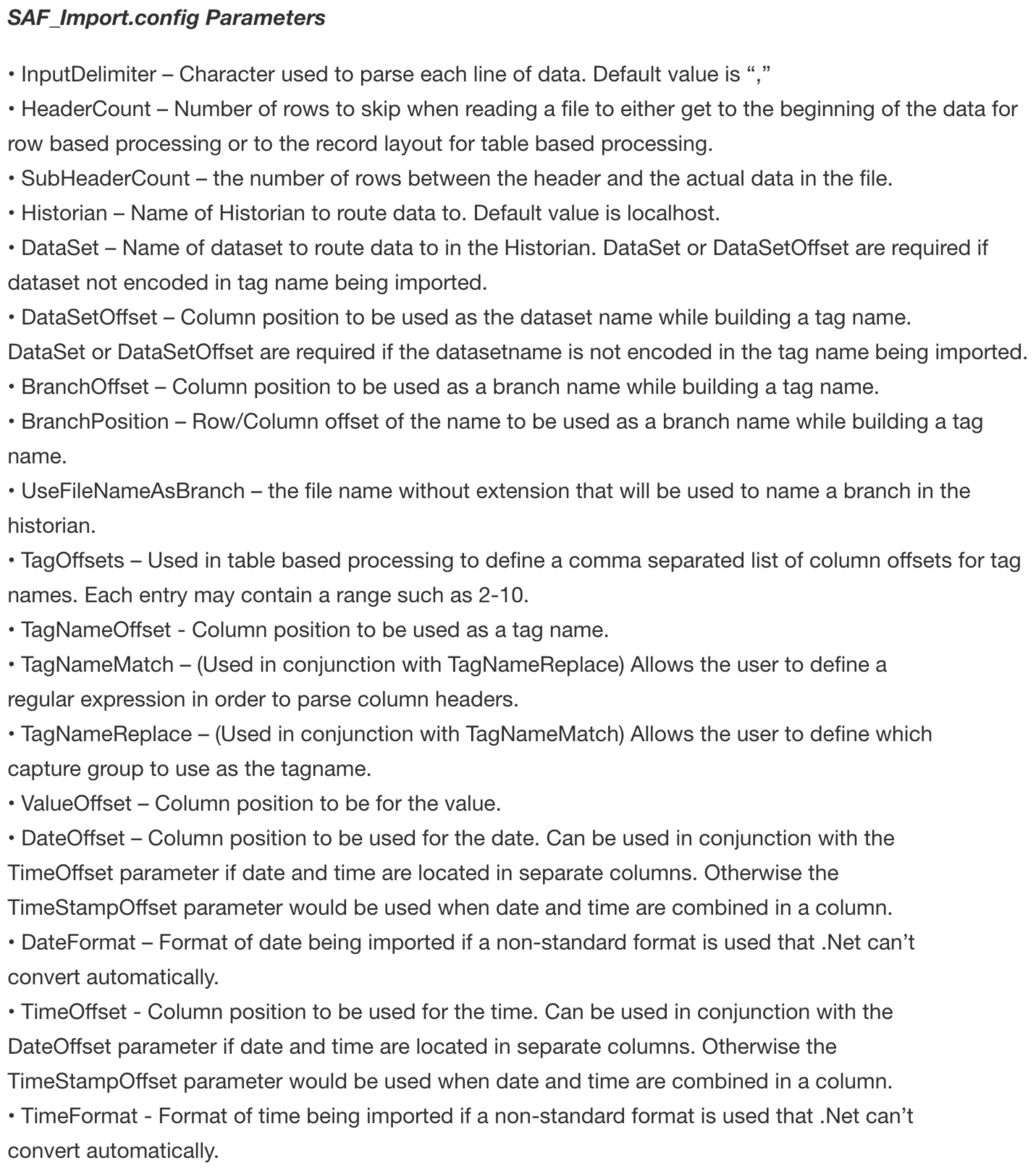
3. The second file is called *SAF_Import.config*

In this config file multiple parameters need to be specified that will be used to match the format of the CSV file to be imported into the historian. There are over 30 parameters to choose from. Here is an example of how the file should look



```
<Settings>
        <InputDelimeter>,</InputDelimeter>
        <Historian>localhost</Historian>
        <DataSet>CSVDataSet</DataSet>
        <TagNameOffset>0</TagNameOffset>
        <TimeStampOffset>1</TimeStampOffset>
        <ValueOffset>2</ValueOffset>
        <KeepProcessedFiles>True</KeepProcessedFiles>
        <ProcessedDirectory>C:\Import\Processed</ProcessedDirectory>
</Settings>
```

Below is a list of all the parameter that can be specified. This list can also be found in C:\Program Files\Canary\Collectors\CSV\Canary CSV Collector Configuration



SAF_Import.config Parameters

- InputDelimiter – Character used to parse each line of data. Default value is “,”
- HeaderCount – Number of rows to skip when reading a file to either get to the beginning of the data for row based processing or to the record layout for table based processing.
- SubHeaderCount – the number of rows between the header and the actual data in the file.
- Historian – Name of Historian to route data to. Default value is localhost.
- DataSet – Name of dataset to route data to in the Historian. DataSet or DataSetOffset are required if dataset not encoded in tag name being imported.
- DataSetOffset – Column position to be used as the dataset name while building a tag name. DataSet or DataSetOffset are required if the datasetname is not encoded in the tag name being imported.
- BranchOffset – Column position to be used as a branch name while building a tag name.
- BranchPosition – Row/Column offset of the name to be used as a branch name while building a tag name.
- UseFileNameAsBranch – the file name without extension that will be used to name a branch in the historian.
- TagOffsets – Used in table based processing to define a comma separated list of column offsets for tag names. Each entry may contain a range such as 2-10.
- TagNameOffset - Column position to be used as a tag name.
- TagNameMatch – (Used in conjunction with TagNameReplace) Allows the user to define a regular expression in order to parse column headers.
- TagNameReplace – (Used in conjunction with TagNameMatch) Allows the user to define which capture group to use as the tagname.
- ValueOffset – Column position to be for the value.
- DateOffset – Column position to be used for the date. Can be used in conjunction with the TimeOffset parameter if date and time are located in separate columns. Otherwise the TimeStampOffset parameter would be used when date and time are combined in a column.
- DateFormat – Format of date being imported if a non-standard format is used that .Net can’t convert automatically.
- TimeOffset - Column position to be used for the time. Can be used in conjunction with the DateOffset parameter if date and time are located in separate columns. Otherwise the TimeStampOffset parameter would be used when date and time are combined in a column.
- TimeFormat - Format of time being imported if a non-standard format is used that .Net can’t convert automatically.

• TimeStampOffset - Column position to be used for the date and time. Can be used if date and time are located in the same column. Otherwise the DateOffset and TimeOffset parameters would be used when date and time are in separate columns.
• DateTimeFormat - Format of datetime being imported if a non-standard format is used that .Net can't convert automatically.
• FileDate – Allows the user to define the date of the hdb file within the historian.
• StartDatePosition – For file formats that are interval-based this gives the user an initial datetime to use to calculate the timestamp of each value.
• IntervalOffset – For file formats that are interval-based this gives the user the delta between each row
• IntervalOffsetUnits – The units that are used for the interval between each row
• QualityOffset - Column position to be used for the quality.
• DataTypeOffset - Column position to be used if the data type is included with the value.
• KeepProcessedFiles – Determines if files are copied to another directory or deleted when processing is complete.
• ProcessedDirectory – Directory to copy files to when processing is complete if the KeepProcessedFiles is set to true.
• AutoCreateDataSet – Determines if SAF would create a dataset in the Historian if the dataset doesn't currently exist. Default value is true.
• AllowInsertedData – Determines if SAF will insert out of order data or if data must be received in chronological order. Default value is true.
• AutoWriteNoData – Determines if SAF will insert a no data quality for each tag when a session disconnects. Default value is false.
• NoDataValues – A comma separated list that will cause SAF to insert a no data quality for a particular TVQ if the value supplied in the file matches one of the values in the list.
• UseSystemCultureForValueParsing – This is a Boolean value that tells us to use the machine's culture instead of the invariant culture.

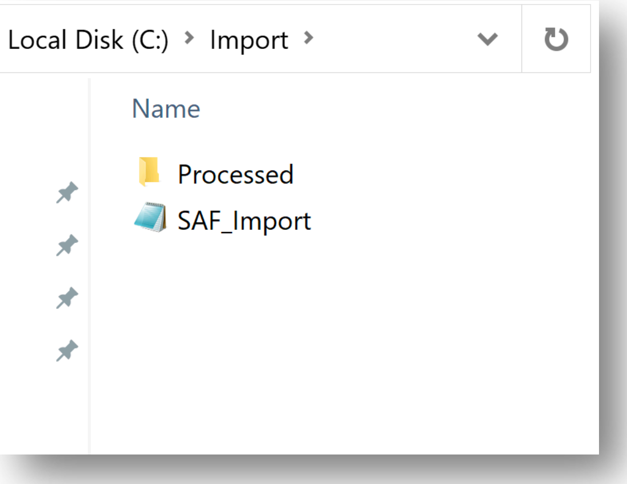
• UTCtimeStamp – This is a Boolean value that indicates whether or not the timestamps are in UTC. The default value is false and assumes the timestamps are local time.

4. Specify CSV files to import

The SAF_Import.config file should be copied and placed in the same directory in which the CSV files will be placed. Ensure that this folder is public, otherwise the collector won't be able to process the files.

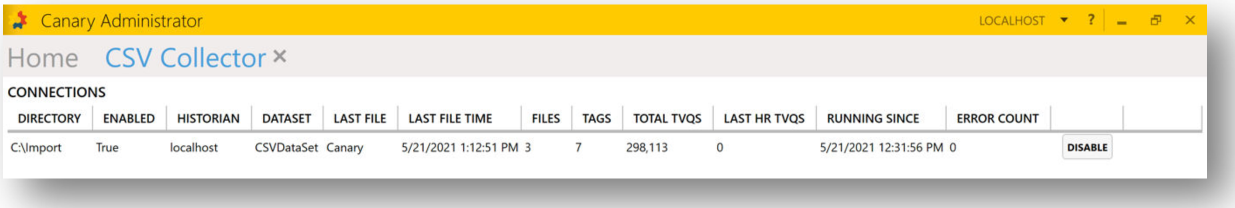
In the SAF_ImportService.config file, make sure the folder you chose as <InputDirectory> is where you paste the SAF_Import.config file along with the CSV files.

In this example, the folder is named "Import" and a <ProcessedDirectory> named "Processed" is created inside of the Input Directory folder. This is where the processed CSV files will be moved once the Collector has completed processing each of the files



The Import folder is a public folder placed directly in the C: drive. Once you start adding the CSV files in this folder, according to the <Interval> period you set in the SAF_ImportService.config file, the files will be processed and moved to the Processed folder, if you chose to configure it. This is a good practice to implement. If this parameter is excluded, the processed CSV file will stay in the public folder where you placed it.

Switching over to the Canary Admin application, if you open the "CSV Collector" panel, the connections to the directory you set up will be visible



CONNECTIONS

DIRECTORY	ENABLED	HISTORIAN	DATASET	LAST FILE	LAST FILE TIME	FILES	TAGS	TOTAL TVQS	LAST HR TVQS	RUNNING SINCE	ERROR COUNT	
C:\Import	True	localhost	CSVDataSet	Canary	5/21/2021 1:12:51 PM	3	7	298,113	0	5/21/2021 12:31:56 PM	0	DISABLE

When you open the "Historian" panel and navigate to the <DataSet> you specified, you will see the CSV data that is being imported according to your configurations

5. Summary

To utilize the CSV import service in Canary, here is a summary of what you need to do:

1. Install CSV Data Collector component
2. Create config files
3. Specify the CSV files to import

Element8 has data migration tools available for the conversion of exported data into CSV format. Feel free to contact us if you need help with your data migration.

For additional resourcing, please direct yourself to the Canary user manual:
https://help.canarylabs.com/hc/en-us/articles/360051080774-The-Canary-System-Quick-Start-Guide-V20-1-0-