**Using AutoConfig PFL Generator**

Software Procedure

CAL-AN-011

|  | | |
| --- | --- | --- |
| REVISION NO. | DATE | | ISSUED FOR |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| File path: C:\Users\benjamin.ng\Documents\GitHub\PflGenerator\CAL-AN-011 Using PFL Generator.docx | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TITLE | | NAME | | POSITION TITLE | | SIGNATURE | | DATE | |
| Author | | Ashley Cogan | |  | |  | |  | |
| Reviewer | | Benjamin Ng | |  | |  | |  | |
| Approver | |  | |  | |  | |  | |
| Authoriser | |  | |  | |  | |  | |

Contents

[1 Introduction 1](#_Toc514148215)

[2 Prerequisites 1](#_Toc514148216)

[3 Known Limitations 1](#_Toc514148217)

[4 Glossary 1](#_Toc514148218)

[5 Relinking Scan Points Procedure 2](#_Toc514148219)

[5.1 Overview 2](#_Toc514148220)

[5.2 Breaking Scan Links 2](#_Toc514148221)

[5.3 Creating Scan Points 2](#_Toc514148222)

[5.4 Creating Scan Links 5](#_Toc514148223)

[5.5 Modifying Component Attributes 7](#_Toc514148224)

[5.6 Generating Script Files 8](#_Toc514148225)

[5.7 Transferring to Application Node 9](#_Toc514148226)

[5.7.1 Linux Users 9](#_Toc514148227)

[5.7.2 Windows Users 9](#_Toc514148228)

[5.8 Deploying Script 10](#_Toc514148229)

[5.9 Rolling Back Changes 10](#_Toc514148230)

List of Tables

Table 1: Glossary 1

List of Figures

Figure 1- Break Scan Link CSV File Information 2

Figure 2 - SCADAPack Configuration Grab 3

Figure 3 - Creating Analogue Points 3

Figure 4 - Creating Binary Input Points 4

Figure 5 - Creating Counter Input Points 4

Figure 6 – Analogue Break Points Information Examples 6

Figure 7 – Link Points Configuration File Example 7

Figure 8 - Attribute Modification 8

Figure 9 - Files For Script 8

Figure 7 - Example Command 9

# Introduction

The purpose of this document is to provide instruction on the using the AutoConfig tool to generate PFL configurations and to instruct the user on how to deploy these scripts on the system.

# Prerequisites

* Python 2.7
* Easygui (python package)
* Access to the Horizon Power SCADA network (XCAL account)
* SSH Connection Client
* Working understanding of the PowerOn Fusion system

# Known Limitations

* Can only import Binary Input, Counter and Analogue Input point types

# Glossary

Table : Glossary

|  |  |
| --- | --- |
| Term | Description |
| Scan Point | An entry in the Real-Time Database (PowerOn Fusion) that tells and RTU to poll a point. |
| Scan Link | A database entry that links a Scan Point to a Component |
| Component | A component is an object within PowerOn Fusion (generally a graphical representation of a symbol) that makes up the PowerOn Fusion system |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Relinking Scan Points Procedure

## Overview

To update the Scan Links for each component scan point within PoF, there are three tasks to be ran with the PFL Generator. Each of these tasks are describe below:

* Breaking Scan Links - Breaks existing scan links to components in the system
* Create Scan Points - Creates new scan points to target the new RTU
* Create Scan Links - Creates new scan links to components in the system
* Generating PFL files – PFL files and BASH scripts are created to be imported into PoF to modify and update Scan Links.

Each of these tasks requires specific configured input files (i.e. comma separated values) to generate the final set of configuration files (pfl files to be imported by PoF).

|  |  |
| --- | --- |
| **Caution:**  **caution_icon** | *The input parameters of the data files must match the formatting in the document precisely as there is the possibility of causing irreversible damage to the system. Be vigilant!* |

## Breaking Scan Links

Breaking links is done by first identifying the RTU and card in PowerOn Fusion and creating a break link file. To create a break link file, a CSV file is created with the data table as outlined below (replacing the information on the table with your own info):

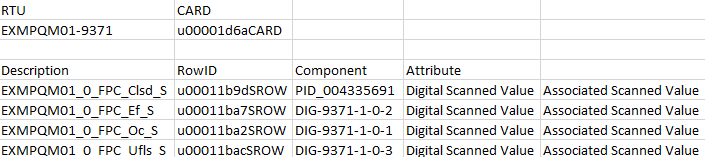


Figure - Break Scan Link CSV File Information

Once complete save this file as a .csv using the following file name format:

*break\_<card ID>\_<point type AI/BI/C>.csv*

One file is created per point type, AI is for Analogue Input, BI for Binary Input and CI for Binary Counters. The best place to get this information is from the existing PoF production system by using the Scada Configuration tool.

## Creating Scan Points

Creating scan points is completed by running the PFL generator tool with the specify RTU file. The flag –scadaPackParse is used with the tool to specify a SCADAPack RTU configuration is used for parsing the file.

The script can automatically generate point import tables from a SCADAPack configuration file using the following command:

PflGenerator.py –-scadaPackParse <RTU Configuration File>

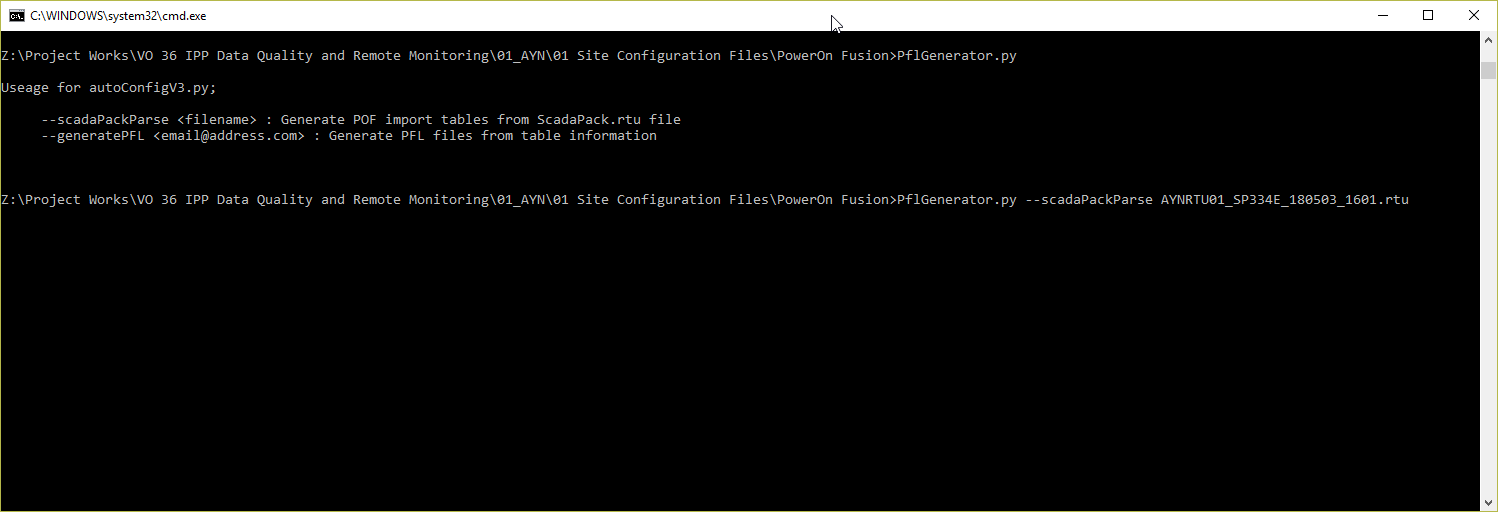


Figure - SCADAPack Configuration Grab

After the script is ran, four (4) files are created for each type of IO for the RTU:

* createAI.csv for analogues inputs
* createBI.csv for binary inputs
* createBO.csv for binary outputs
* createCI.csv for binary counters

Each of these CSV files contains the structure for the Scan Points for the RTU to be imported into PoF. Extra information is required to be entered to complete the file for processing with the PFL generator tool.

For the analogue import file, complete the data table with the new RTU name and Card ID that is created under PoF. Once complete, save this file as a .csv using the following format:

*create\_<card ID>\_AI.csv*

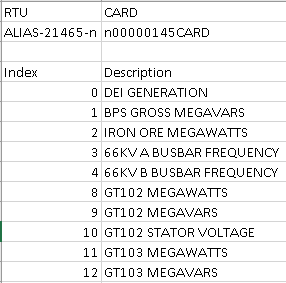


Figure - Creating Analogue Points

For the binary import file, complete the data table with the new RTU name and Card ID that is created under PoF. Once complete, save this file as a .csv using the following format:

*create\_<card ID>\_BI.csv*

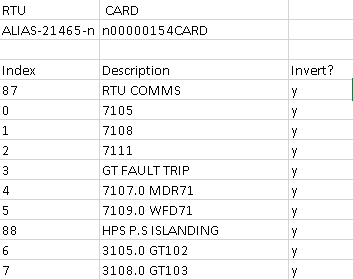
**

Figure - Creating Binary Input Points

Note: Placing ‘y’ in the invert column will generate the configuration required to invert this signal in the PowerOn Fusion system.

For the counter import file, complete the data table with the new RTU name and Card ID that is created under PoF. Once complete, save this file as a .csv using the following format:

*create\_<card ID>\_CI.csv*

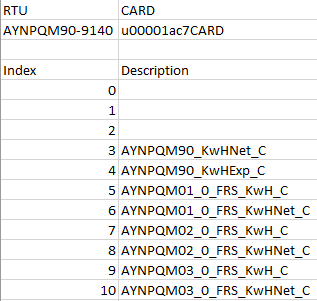
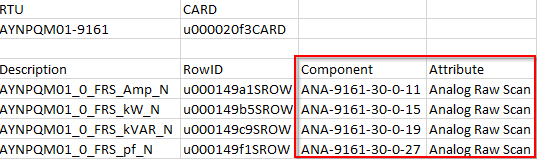


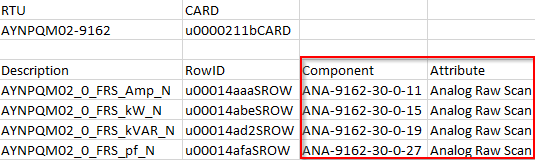
Figure - Creating Counter Input Points

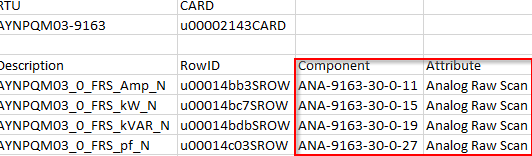
## Creating Scan Links

To create Scan Links configuration files, a combination of the Scan Points files and the information from existing PoF information are entered together into a new CSV file. Each point that is defined in the Create Scan Point files *create\_<AI/BI/CI>.csv* must have a ‘Component’, ‘Attribute’, and optionally an ‘Associated Value’ to be assigned for it be linked in PoF. Once the information has been entered for each point, it can be saved as a .csv file using the following format:

*link\_<card ID>\_<point type AI/BI/CI>.csv*







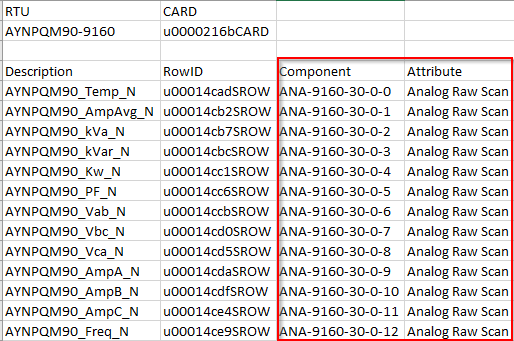


Figure 6 – Analogue Break Points Information Examples

The information from each break point configuration file (Figure 6) are combined into the new scan link configuration file (Figure 7) as per each point that is created under the create scan points files.

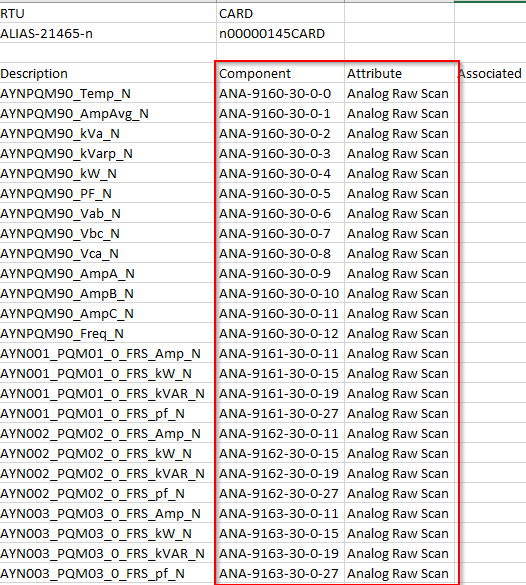


Figure 7 – Link Points Configuration File Example

## Modifying Component Attributes

Modifying component attributes is done by specifying the component, attribute and the old and new values in PowerOn Fusion.

To create an attribute modification file, complete the data table as outlined below (replacing the information on the table with your own info). Once complete save this file as a .csv using the following format;

*attributeMod\_<description>.csv*

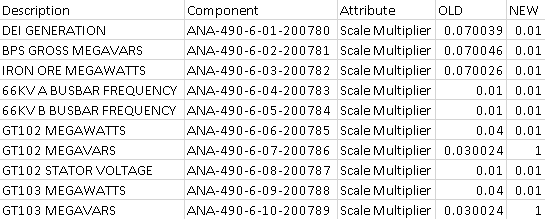


Figure - Attribute Modification

The OLD value is the current value of the component attribute and the NEW value is the value you wish to change the attribute to. It is important to complete both parameters as the script that generated the rollback files will take both values.

## Generating Script Files

After you have prepared the data you should have a folder layout that looks something like this;

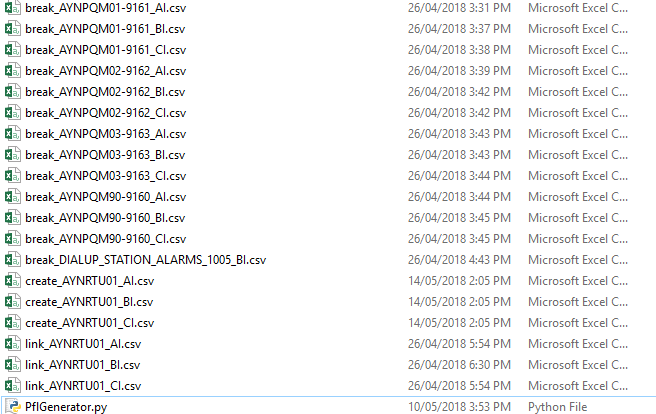


Figure - Files For Script

1. Hold shift and right click on the folder and click ‘*Open command window here’*
2. Run the script using:  
     
   python PflGenerator.py
3. The output prints available commands for this script.
4. To generate configuration files, use the command:  
     
   python PflGenerator.py –-generatePFL your.email@bhpbilliton.com

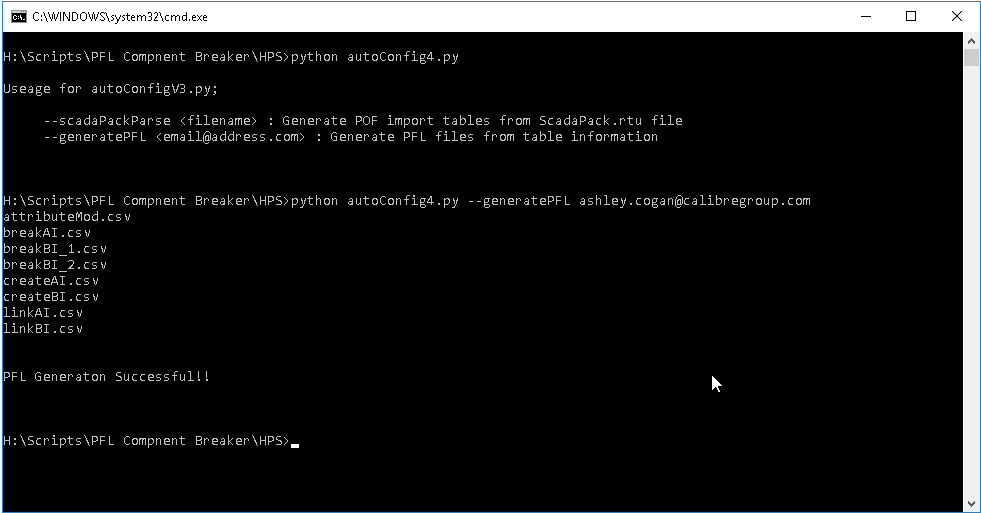


Figure - Example Command

1. Rename the folder pflFiles to something relevant to your project and zip.

## Transferring to Application Node

To run the script, it must first be copied over to the application node and then executed as the ‘enmac’ user.

|  |  |
| --- | --- |
| **Caution:**  **caution_icon** | *It is strongly recommended to run this script on the test environment before rolling out on production. Test both the main deployment and the rollback script.* |

### Linux Users

1. Copy *pflFiles.zip* to your /tmp directory on TSTAPP01 using the following command:

scp pflFIles.zip <yourxcal>@tstapp01.scada.horizonpower.com.au:/tmp

1. Connect to TSTAPP01 using ssh by entering the following command:

ssh tstapp01 -l <your xcal>

1. Change user to enmac using the following command:

sudo su – enmac

1. Copy file to each site’s files directory using the following command:

cp /tmp/pflFIles zip ~/<sitename>files

### Windows Users

1. Copy *pflFiles.zip* to your /tmp directory on TSTAPP01 using the program WinSCP and your XCAL account:

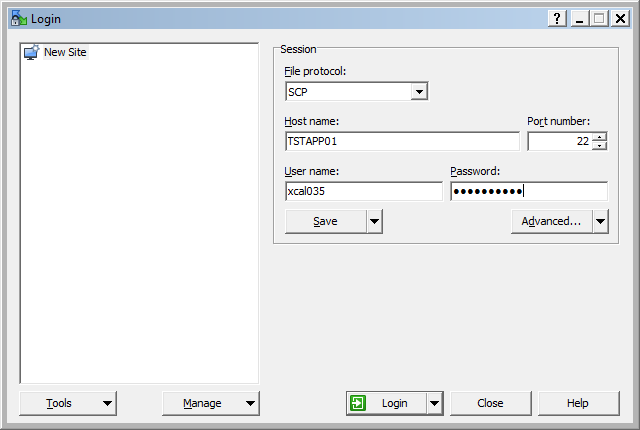


Figure 10 – WINSCP Login Screen

1. Connect to TSTAPP01 using Putty and SSH connection and login under your XCAL account:

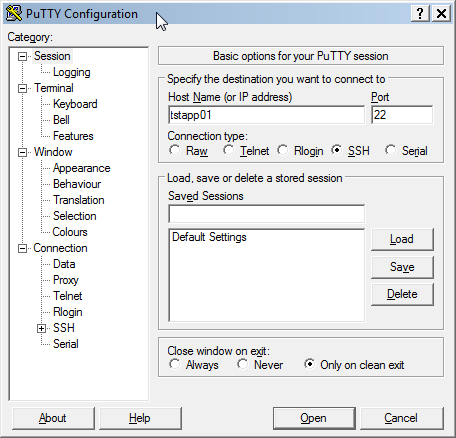


Figure 10 – PUTTY Login Screen

1. Under the Putty terminal instance, change user to ‘enmac’ using the following command:

sudo su – enmac

1. Copy file to plant\_file directory using the following command:

cp /tmp/pflFIles zip ~/<sitename>files

## Deploying Script

Three scripts are created under the PFL Generator, main.sh, createPoints.sh, and linker.sh.

The main.sh script is used to create new points, break existing links and relinking to the new points.

The createPoints.sh is used to only create new points for the new RTU, when linking is not yet ready.

The linker.sh is used to break existing links and relinking to the new points that are created, once the new RTU is commissioned.

Based on the requirements of the user, each of these scripts can be ran individually to achieve different results.

To execute the scripts, follow the steps below:

1. Under the site’s file directory, unzip files using the following command:

unzip pflFiles.zip

1. Navigate to the main directory:

cd ~/<sitename>files/main

1. Run the shell script required:

bash <main/createPoints/linker>.sh

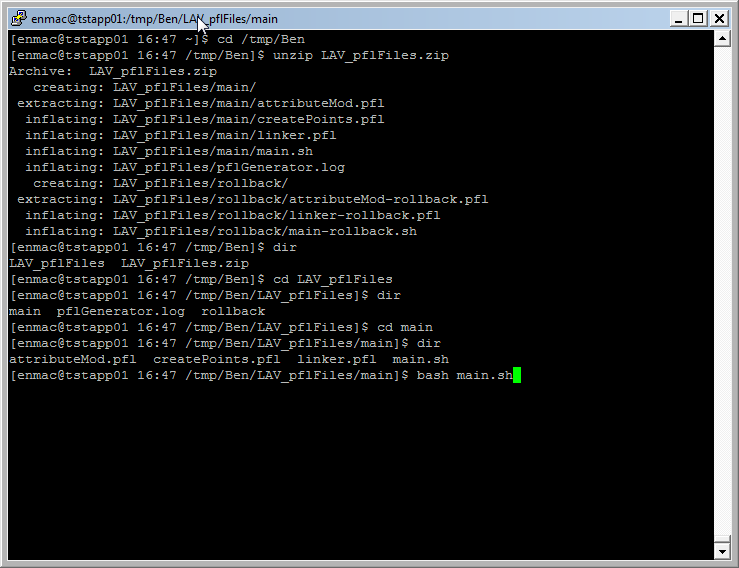


Figure 10 – Deploying main.sh under Putty

## Rolling Back Changes

When roll back is required after points have been broken and linked to the new RTU, the rollback script can be deployed to revert any changes made to PoF.

To rollback, simply follow the steps below:

1. Navigate to the rollback directory:

cd ~/<sitename>files/rollback

1. Run the shell script required:

bash main-rollback.sh