transition

Moving to a smart future

Low Level Design: Power Systems Analysist (PSA) Software

October 2022

Scottish & Southern Electricity Networks

Table of Contents

1 INTRODUCTION 4
1.1 Document Purpose 4
1.2 Key Terms (in) 4
1.3 Management Overview 5
2 OVERALL DESIGN 7
2.1 PSA and S&D 8
2.2 File Detection System 10
2.3 File Naming Convention and Folder Structure 11
3 Core Functions 13
3.1 Determine Constraints and Calculate Flexibility Requirements 13
3.2 Calculate Sensitivity Factors 17
3.3 Constraint Resolution 20
3.4 Data File Creation and Analysis 23
4 SOFTWARE ENVIRONMENT 25
5 RINA Sensitivity Factor Algorithm 26
6 PSA S&D Configuration File 27
7 Run Time Parameters File 29

1 INTRODUCTION

1.1 Document Purpose

The purpose of this document is three fold:

1. Provide a detailed description of the functionality of the software actually developed and used during the trials
2. Provide observations on any future developments of a solution for BAU
3. Provide comments on the issues that arose during the development and operation of the software during the trials

1.2 Key Terms (in)

Term	Abbrev	Description
Power systems analysis	PSA	The term used for the software developed in-house for the analysis of the network to determine if constraints exist. This works in conjunction with S&D
Select and dispatch tool	S&D	The term used for the software developed by TNEI that manages the select and dispatch process of contracts to resolve constraints. This works in conjunction with PSA
Business as usual	BAU	When a system is used in an operational capacity to manage the live network
Requirements traceability catalogue	RTC	Previously approved documents detailing the requirements from the users and subject matter experts
Distributed Energy Resource	DER	An asset able to provide power in a flexible manner (flex asset)
Flexibility Requirement	FR	The resulting power (kW) injection required to resolve the Constraint (flex reqt)
Sensitivity Factor	SF	A calculated value that measures a DERs effectiveness to resolve a constraint
Constraint Resolution	CR	The iterative process between S&D and PSA to resolve a constraint
PowerFactory	PF	The software system from Digsilent used to run the loadflow calculations
Registro Italiano Navale	RINA	Specialist in energy transition and electrical research
Graphical User Interface	GUI	The user interface of the PSA and S&D software

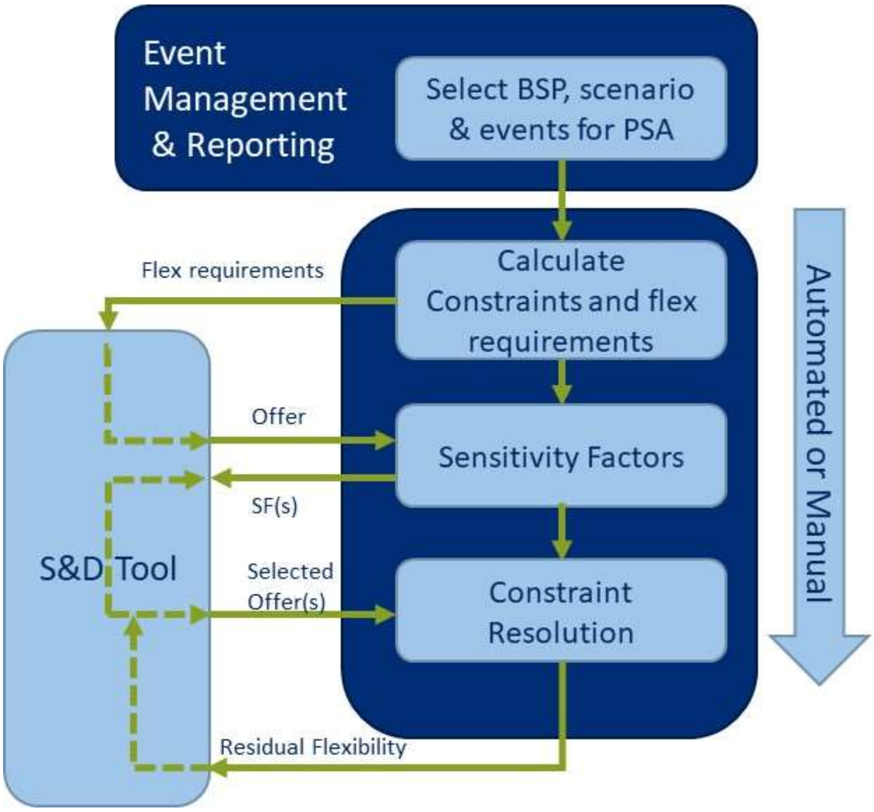
1.3 Management Overview

This section is a high-level management and technical overview of the PSA software, it's requirements, design, development, testing and use during the trials.

It is assumed the reader is familiar with the high-level objectives of the TRANSITION Technical Trials that took place during March, April, and May 2023.

The requirements for PSA are defined in REF-001 Requirements Traceability Catalogue and REF-002 Context Diagram.

The diagram below outlines the core components of the PSA software:



The core components of PSA are to:

1. Identify constraints on the network (where a network element has exceeded a defined % loading threshold)
2. Calculate the power required to resolve the constraint, at the location of the constrained network element, and communicate these to the S&D Tool
3. Receive offers of flexible power, to resolve the constraint, from participants via the S&D Tool
4. Calculate the network sensitivity factors (SFs) associated with these offers and communicate these to S&D Tool
5. Receive candidate responses from S&D Tool and determine if they resolve the constraint(s) and communicate the results to S&D Tool, including any residual power requirements. This is an iterative process until all constraints are resolved, with or without the use of "dummy assets"

These steps can be conducted in an automated process or step by step manual process to gain more insights and understanding. The usual running mode during the trials was automatic.

PSA was developed using Python scripts to automate PowerFactory. The scripts managed the graphical user interface (GUI), data inputs/outputs and the processing of PowerFactory load-flow calculations.

PSA also included API interfaces to SIA (Demand and generation 10 day ahead forecasts) and NeRDA (Near Real-time Data Access to network switch positions)

The PSA software simulated constraints on the network through a complex set of inputs and processes.

It was based upon the following part of the SSEN network:

- Simplified network model:
    - Single BSP (Cowley Local)
    - Single Primary (Rose Hill) although in reality all primaries were modelled

It enabled the following services to be modelled:

- Services:
    - SPM
    - SEPM
    - Secure
    - Dynamic

It enabled very complex scenarios to be modelled through the use of maintenance and contingency inputs:

- Scenarios:
    - Base model (Normal running arrangement of network elements and switches)
    - Maintenance (Planned outages)
    - Contingency (N-1(s), unplanned outages)
    - Maintenance and Contingency (Planned and unplanned outages)

REF-006, PSA Final Report PSA Final Report v1.0.docx should be read to fully understand the considerations and recommendations for progressing development of a PSA and S&D type solution forward into a BAU application.

2 OVERALL DESIGN

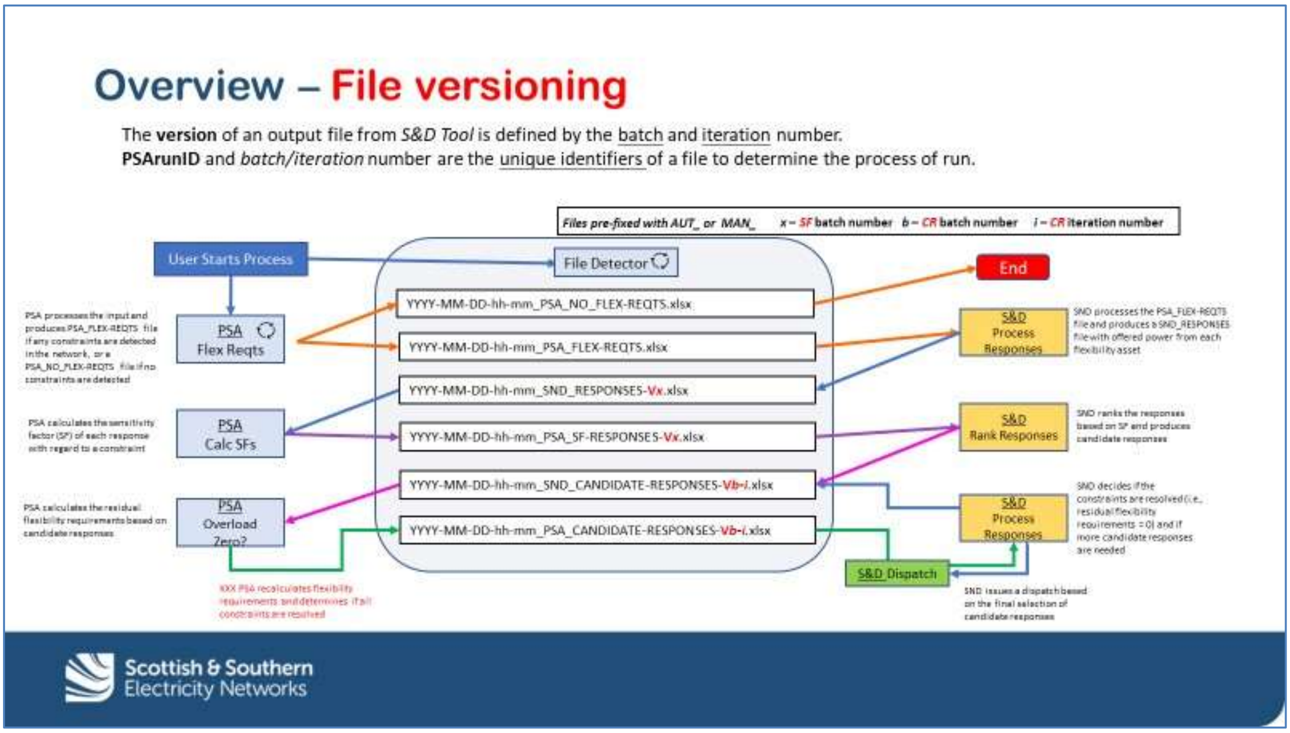
The overall solution consists of three separate software modules (PSA, S&D, File Detector). These modules communicate with each other through a data file exchange mechanism.

The high-level functionality of each module is as follows:

PSA	• Detect constraints on the network • Calculate flexibility requirements • Calculate sensitivity factors (SFs) • Check if constraint(s) are resolved
S&D	• Receive responses (offers) from Participants • Select best valid responses based on SFs and costs • Iterate through responses with PSA until constraints are resolved (possibly using dummy assets)

File Detector	• Detect files created by S&D
	    o Calculate SFs
	    o Check constraint resolution

The following diagram shows the file processing interactions between PSA and S&D:



2.1 PSA and S&D

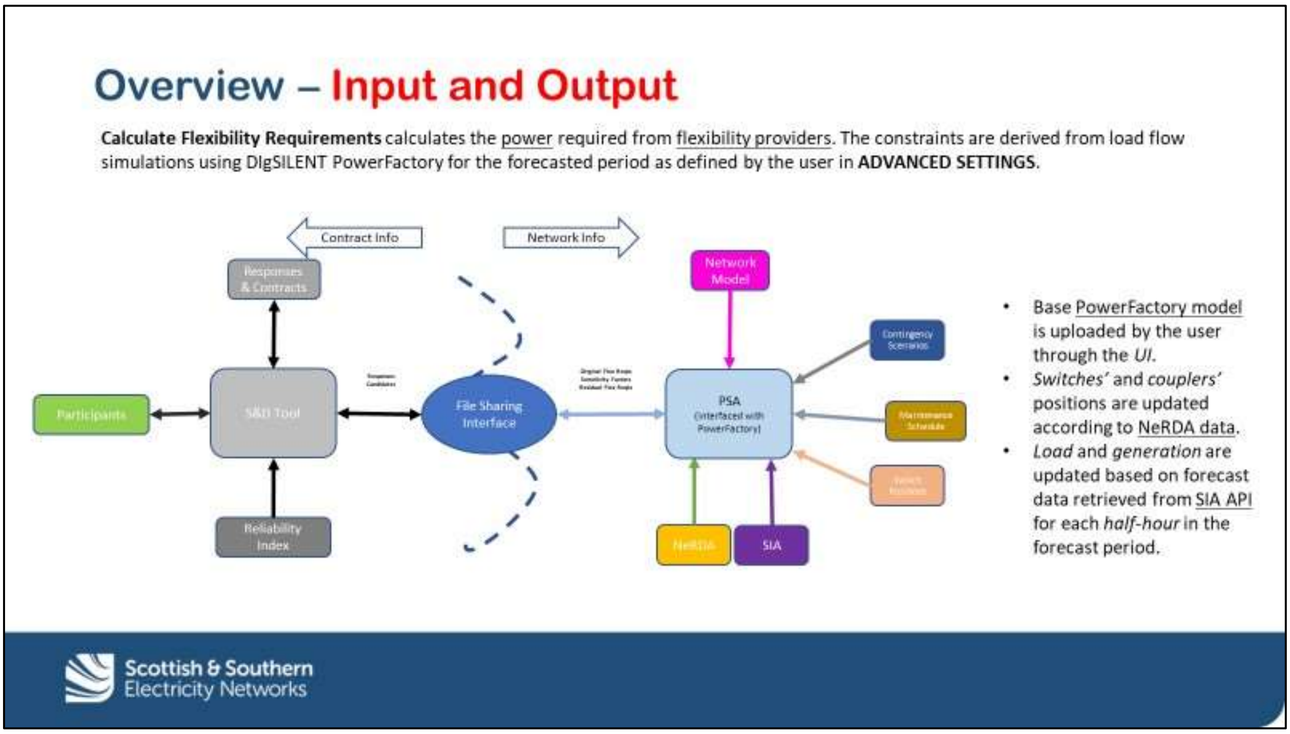
The following sub-sections focus on the high-level design of PSA and S&D and the differences between them. PSA, S&D, and File Detector were all written using the Python programming language. Details of the low-level design and implementation are included on the module specific sections. PSA and File Detector were developed by SSEN and S&D was developed by TNEI.

One key design aim was to develop a standalone Power Systems Analysis tool based upon the automation of DigSilent's PowerFactory. Other secondary design features such as data visibility and flexibility of data input were also taken into consideration during the design and development process.

2.1.1 Data View

One of the key design aims was to clearly separate the functions of PSA and S&D and to make the data more visible to users. This was in reaction to dealing with the previous monolithic software solutions that gave users very little visibility of data, especially the intermediate steps to the final outcomes.

PSA deals with network related processes and data, S&D deals with Participant contract and dispatch related processes and data as is shown in the diagram below:



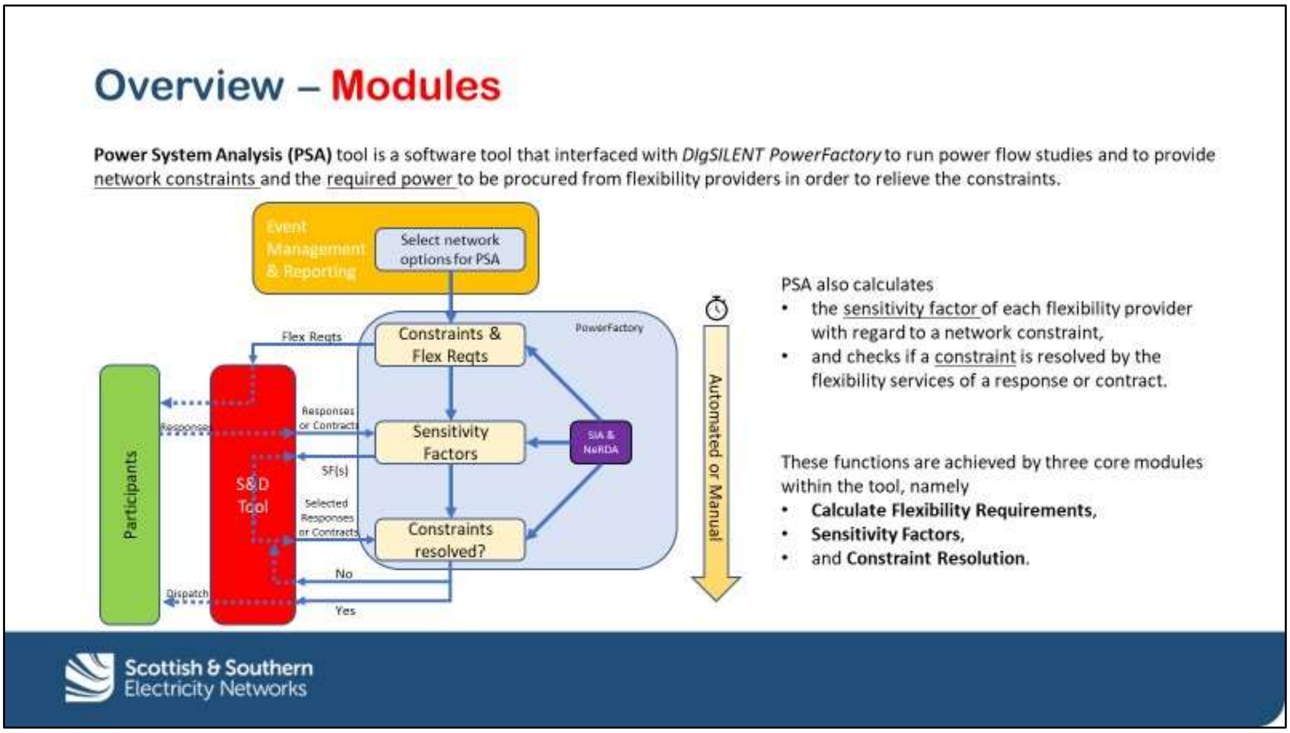
2.1.2 Functional View

This clear separation of processes and data between PSA and S&D was followed in the design of PSA with the three core functions of PSA being separated with simple clearly defined data file interfaces.

These three core functions are:

1. Detect network constraints and calculate flexibility requirements
2. Calculate sensitivity factors
3. Determine if constraint(s) are resolved

These core functions and the surrounding processes are shown in the diagram below:



2.1.3 Scenario Processing

On of the requirements was to be able to model different operating scenarios that map onto the services (SPM, SEPM, Secure, and Dynamic). These scenarios are known as BASE, MAINT, CONT, and MAINT_CONT and defined as follows:

Scenario	Description
BASE	The basic network topology, using current or previous NeRDA data
MAINT	The basic network topology modified with maintenance data that simulates specific assets being in/out of service at specific predetermined times which are used for determining future constraints over the 10 day simulation period
CONT	The basic network topology modified with contingency data that simulates specific assets being in/out of service at specific

	predetermined times which are NOT used for determining future constraints over the 10 day simulation period
MAIN_CONT	The combination of the MAINT and CONT scenarios

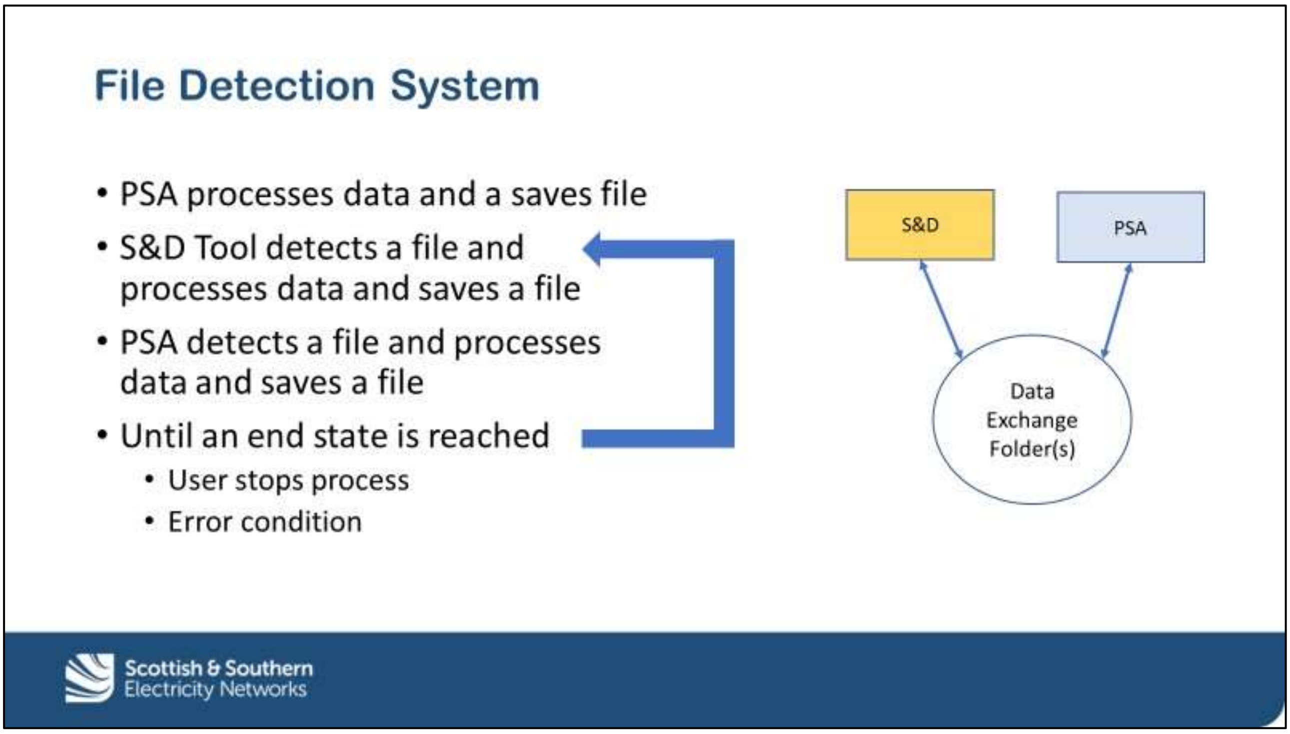
These scenarios are further augmented by the addition of EVENT and SWITCH data which are as follows:

EVENT	A file containing times at which a specific asset will generate or demand a defined amount of power on the network
SWITCH	A file containing times at which a specific switch will change its state to OPEN or CLOSED

Using the above scenarios and other input files all the required services were able to be simulated.

2.2 File Detection System

The File Detection System detects files generated by S&D and starts the relevant PSA sub-process. S&D has its own File Detection “watchdog” system, which is documented in the TNEI Low Level Design.



S&D generates the following files:

Responses	• Contains responses (offers) from Participants that have been validated and require SFs to be calculated by PSA • Starts the PSA_SF sub-process

Candidate responses	• Contains combinations of responses for PSA to determine if they resolve the constraint(s) • Starts the PSA_CR sub-process

2.3 File Naming Convention and Folder Structure

Every file generated by PSA or S&D is an Excel (.xlsx) file and has the following naming convention:

XXX_YYYY-MM-DD-HH-mm_SYS_FILETYPE.xlsx

XXX_YYYY-MM-DD-HH-mm is known as the PSArunID and is the unique identifier for all files associated with this run and also the name of the folder for storing the data files.

XXX_ is either AUT_ or MAN_ indicating PSA running in automatic or manual mode.

YYYY-MM-DD-HH-mm is the date/time stamp that the PSA run started.

SYS is the system that produced the file PSA or SND

FILETYPE describes the contents of the file, as follows:

FILETYPE	Contents
PSA_FLEX-REQTS	Flex power reqts for the detected constraints
PSA_NO-FLEX-REQTS	Indicates that no constraints were detected and hence no flex reqts
SND_RESPONSES-Vx*	Responses from Participants that require SFs calculating
PSA_SF-RESPONSES-Vx*	Calculated SFs for the responses
SND_CANDIDATE-RESPONSES-Vb-i*	Proposed responses to attempt to resolve the constraints
PSA_CANDIDATE-RESPONSES-Vb-i*	Indication of whether each constraint has been resolved or additional power is required

*The version numbers -Vx and -Vb-i are explained in the relevant sections on Sensitivity Factors and Constraint Resolution.

2.3.1 Folder Structure

When a PSA run is started, by selecting the automated or manual detection of constraints, a folder is created with the name of the PSArunID. The location of the PSArunID folder is dictated by the run time variable

PSA_SND_TOP_WORKING_LEVEL contained in the PSA_SND_Config.txt initialisation and setup file. See the section on Running PSA for more details.

Multiple sub-folders are created from the PSArunID folder that contain specific data and interim results for that run. Folders are as follows:

Folder name	**Description of contents**
0 – INPUT_DATA	Copy of all input files and run time parameters to enable complete reproduction of the run
1 – SIA_DATA	SIA forecast data for each asset
2 – NeRDA_DATA	NeRDA switch/circuit breaker position data
3 – BASE	All results and interim data files for the BASE scenario
4 – MAINT	All results and interim data files for the MAINT scenario
5 – CONT	All results and interim data files for the CONT scenario
6 – MAINT_CONT	All results and interim data files for the MAINT_CONT scenario
7 – ANALYSIS	Output files from user selected analysis functions
8 – EVENT_DATA	Interim data files for generation events during the run
9 – SWITCH_DATA	Interim data files for switch changes during the run
TEMP_SF_FILES	Temporary sensitivity factor data files
TEMP_CR_FILES	Temporary constraint resolution data files

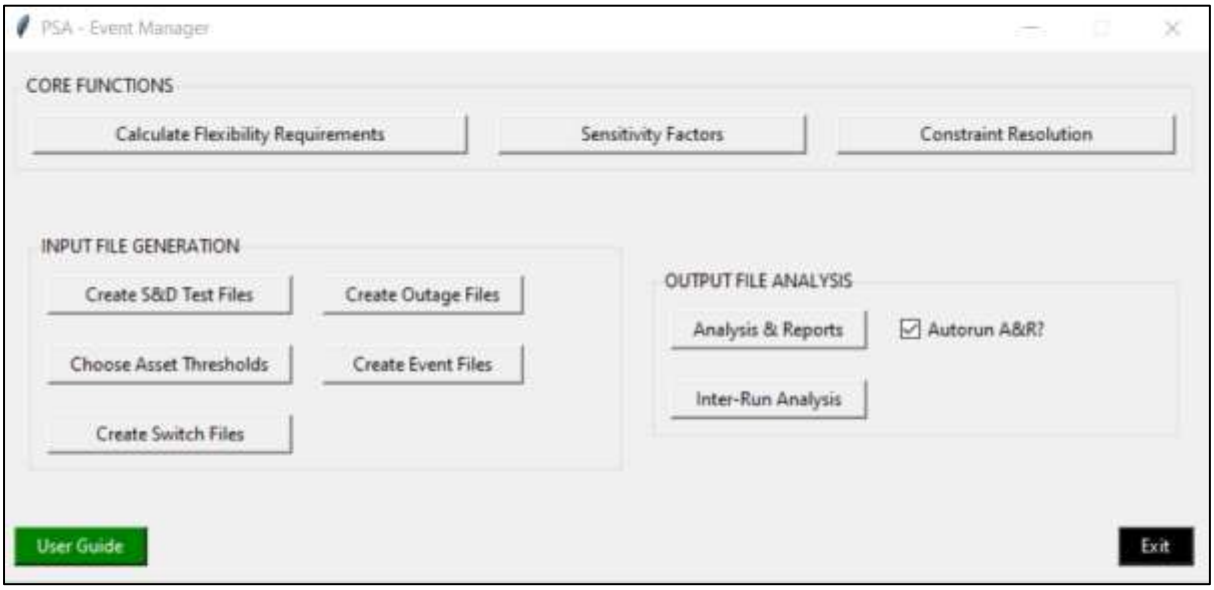
The location of the folders used by the PSA and S&D file detection systems are dictated by the run time variables contained in the PSA_SND_Config.txt initialisation and setup file. See the section on Running PSA for more details.

PSA_SND_FILE_DETECTOR_FOLDER - Transfer of files from PSA to S&D

SND_PSA_FILE_DETECTOR_FOLDER - Transfer of files from S&D to PSA

3 Core Functions

The PSA solution consists of three core functions assisted by a set of data file creation and results analysis functions as shown in the GUI below:



The following sections detail the functionality and algorithm for each of the functions.

3.1 Determine Constraints and Calculate Flexibility Requirements

The calculation of the flexibility requirements is the first step in the entire PSA and S&D process.

In the GUI screen, shown below, the user enters all the data required for this and all subsequent core processes (sensitivity factors and constraint resolution).

The values of data inputs and all specified data file names are recorded in the RUN_TIME_PARAMETERS.txt file and copies of the files stored in the sub-folder “0 - INPUT_DATA”.

The MODELS are the PowerFactory network model and the Asset Data File. This Asset Data File is key to the processing of the loading thresholds, SIA and NeRDA data. See Section XX on Asset Data File. All Primaries that exist in the network model will be processed.

The SCENARIOS data files have been described earlier.

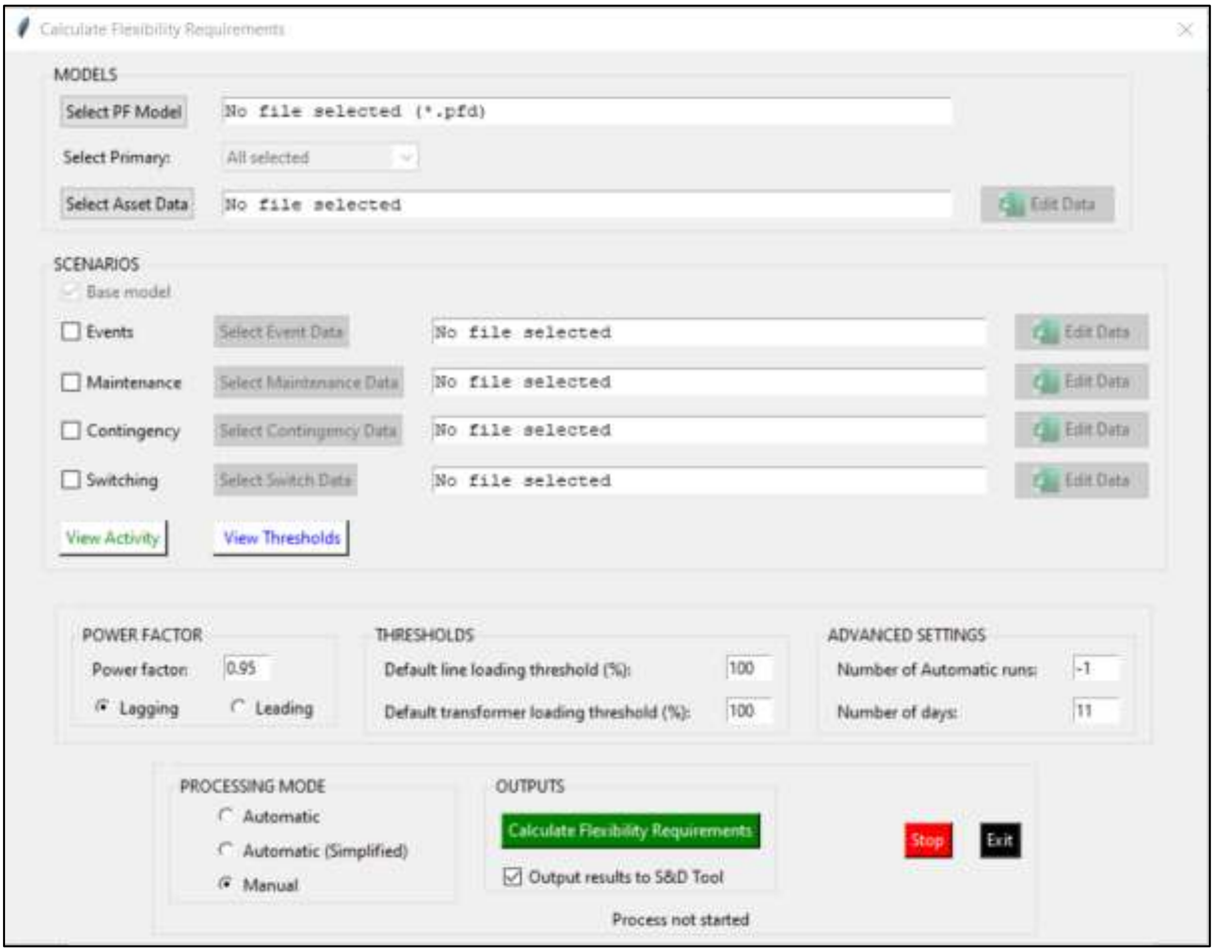
The POWER FACTOR, THRESHOLDS, and ADVANCED SETTINGS should all be left in their default states.

The PROCESSING MODE options are as follows:

Mode	Description
Automatic*	A repeating set of all scenarios using all the scenario data that has been specified
Automatic (Simplified)*	A repeating single scenario run using all the scenario data that has been specified
Manual	A single scenario run using all the scenario data that has been specified

*The process will repeat every T seconds as defined in the PSA_SND_Config.txt file where T is controlled by the variable PSA_SND_CALC_FLEX_REQTS_INTERVAL, it will also repeat the number of times as specified by the GUI entry “Number of Automatic runs”, this defaults to -1 which means run until stopped. During the trials T was set to 5,400 (90 minutes).

The OUTPUTS section allows the user to disable the output to S&D for testing purposes. It also contains the control buttons to start and stop the process.



3.1.1 Processing Steps

The initial steps of calculating flexibility requirements are the very similar for subsequent processes (sensitivity factors and constraint resolution) and are fundamental to the entire PSA and S&D processes.

The process is split into two stages, initialisation and calculations (including outputs):

The initialisation steps, in order, are as follows:

Initialisation Steps	Description
Load config data	Read PSA_SND_Config.txt
Load PSArunID data	Read PSArunID RUN TIME PARAMETERS.txt
Start PF and load project	Get a PF licence, load PF model and activate the project
Load validation data	Master asset register contains a list of all valid assets
Load MAINT file	Read maintenance scenario data
Load CONT file	Read contingency scenario data
Load SIA data	Get SIA data via API and save to file
Load NeRDA data*	Get NeRDA data via API and save to file
Load EVENTS file	Read EVENTS for each scenario
Load SWITCH file	Read SWITCH changes for each scenario
Validate all inputs	Check all asset names against master register
Main calculation loop	See below…

“The NeRDA switch positions stay constant throughout the scenarios unless modified by the SWITCH data (see below)

The calculation steps are as follows:

Loop each scenario	BASE, MAINT, CONT, MAINT_CONT as required
Loop each day	Number of days defined in GUI (usually 11)
Calc downscaling factors	Downscaling factors vary as a function of daily SIA data
Loop each half hour	48 HH in a day (default setting)
Update events	Update additional generation/demand events
Update switches	Update switch positions from default NeRDA positions
Update asset in/out service	Apply maintenance and contingency in/out of service to assets
Apply downscaling factor	Needs to be done for every calculation
Update PF model	Optimised for minimum PF database updates
Run load flow	Uses default parameters from PSA_SND_Config file
Extract results	Get data from PF and save to file

Outside of all the loops detailed above, the final stage of the process is to convert the constrained assets into a flexibility requirement. This is simply converting the %over-loading of the asset into an equivalent power (kW) required at the point of the constraint.

3.1.2 Outputs

There are a number of different outputs, including intermediary files, these are detailed in the Low Level Design. The key outputs are the individual PSA_FLEX_REQTS files stored under each Scenario sub-folder, and the final output files:

XXX-YYYY-MM-DD-hh-mm_PSA_CONSTRAINTS.xlsx

XXX-YYYY-MM-DD-hh-mm_PSA_FLEX-REQTS.xlsx

These files contain the network constraints and corresponding flexibility requirements for the entire PSArunID. The PSA_FLEX-REQTS file is copied to the S&D shared folder.

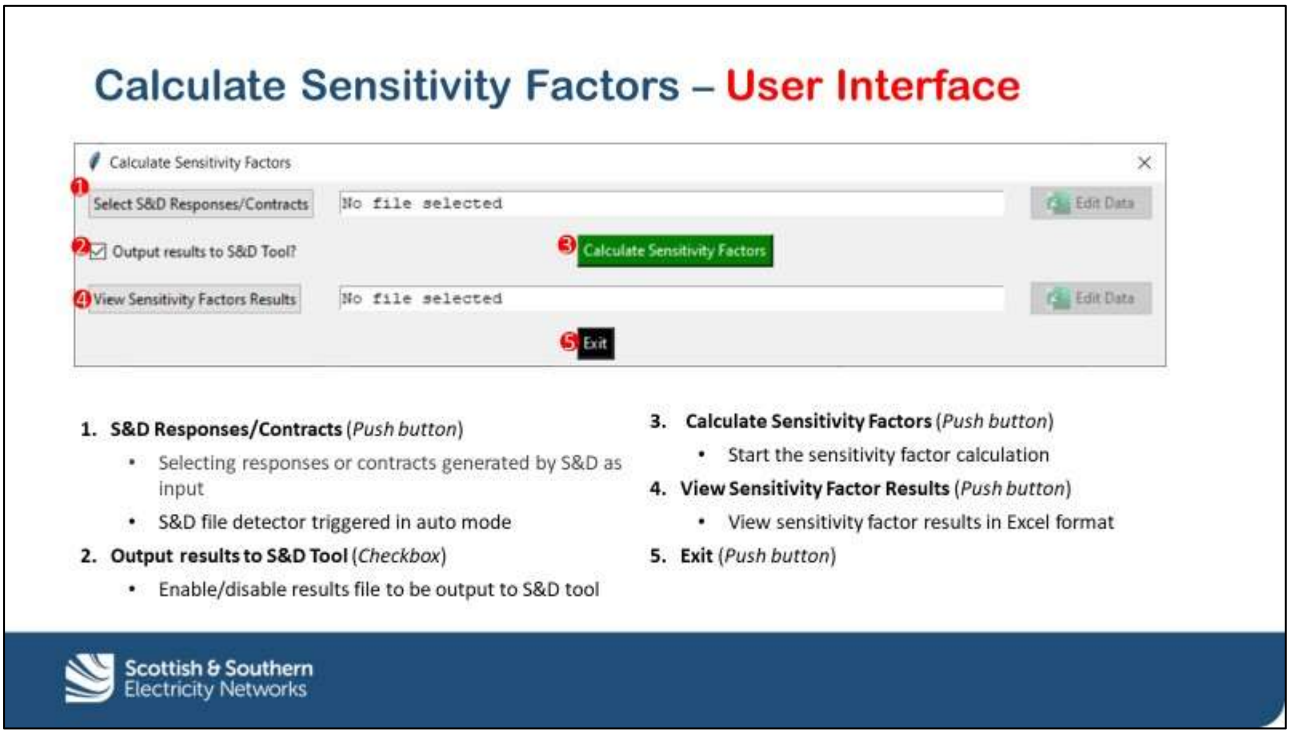
If no constraints are detected then PSA_NO-CONSTRAINTS and PSA_NO-FLEX-REQTS files are created. The PSA_NO-FLEX-REQTS file is copied to the S&D shared folder to alert S&D to not expect any results.

3.2 Calculate Sensitivity Factors

The process of calculating the sensitivity factors can be performed manually through the GUI or automatically through the PSA and S&D interface. The two different methods use the same code, one via the GUI and the other as a sub-process to enable multi-processing of results.

During the trials the normal operation was using the automated processing mode with PSA and S&D automatically passing data files between themselves and processing the results as per the earlier diagram.

The GUI is shown below:



3.2.1 Processing Steps

The initial steps of calculating sensitivity factors are very similar to the other core processes. The process is split into two stages, initialisation and calculations (including outputs):

The initialisation steps, in order, are as follows:

Load config data	Read PSA_SND_Config.txt
Load PSArunID data	Read PSArunID RUN TIME PARAMETERS.txt
Start PF and load project	Get a PF licence, load PF model and activate the project
Load config data	Read PSA_SND_Config.txt
Load PSArunID data	Read PSArunID RUN TIME PARAMETERS.txt

Start PF and load project	Get a PF licence, load PF model and activate the project
Load validation data	Master asset register contains a list of all valid assets
Load S&D input file	Load SND_RESPONSES_Vb input file
Aggregate responses	Aggregate SND_REPSONSES to improve processing
Calc historical flag	Included in SND_RESPONSES, if TRUE then read SIA/NeRDA data from file, otherwise from API
Load SIA data*	See above
Load NeRDA data**	See above
Load EVENTS file	Read EVENTS for each scenario
Load SWITCH file	Read SWITCH changes for each scenario
Validate all inputs	Check all asset names against master asset register
Calc downscaling factors	Can be calculated in advance
Main calculation loop	See below…

*If calc historical is FALSE load new SIA data to reflect any changes in SIA forecast
**If calc historical is FALSE load new NeRDA data to reflect current switch positions

The calculation steps are as follows:

Loop through each scenario	The responses are sorted by scenario
Loop through times of responses	The responses are sorted by start time
Update asset in/out service	Apply maintenance and contingency in/out of service to assets
Apply downscaling factor	Needs to be done for every calculation
Update events	Update additional generation/demand events
Update switches	Update switch positions from default NeRDA positions
Loop through each response	Calculate sensitivity factor for each entry
Calculate before values*	Update PF and run load flow and store interim results
Calculate after values*	Update PF and run load flow and store interim results
Determine SF value	Calc SF from before and after values
Save and output results	Save results to output file PSA_SF-RESPONSES _Vb

*The precise algorithm used is defined in Section 7, note that the algorithm doesn't model many to many SFs, this was considered too much detail at this stage of the trials.

3.2.2 Outputs

There are a number of different intermediate outputs which are detailed in the Low Level Design. The key output from this process is the sensitivity factor for each entry in the SND_RESPONSES-Vb input file,

The intermediate outputs are stored in the PSArunID sub-folder TEMP_SF_FILES, the main PSA output is stored in the top-level PSArunID folder:

XXX-YYYY-MM-DD-hh-mm_PSA_SF-RESPONSES-Vb.xlsx

The Vb suffix in the input and output filename enables S&D to keep track of the different batches that it sends to PSA for processing. The different “batches” represent the services (SPM, SEPM, Secure, and Dynamic) that the responses refer to. The “b” in Vb starts at 0 and is incremented for each subsequent batch. PSA will always use the same value as the input file.

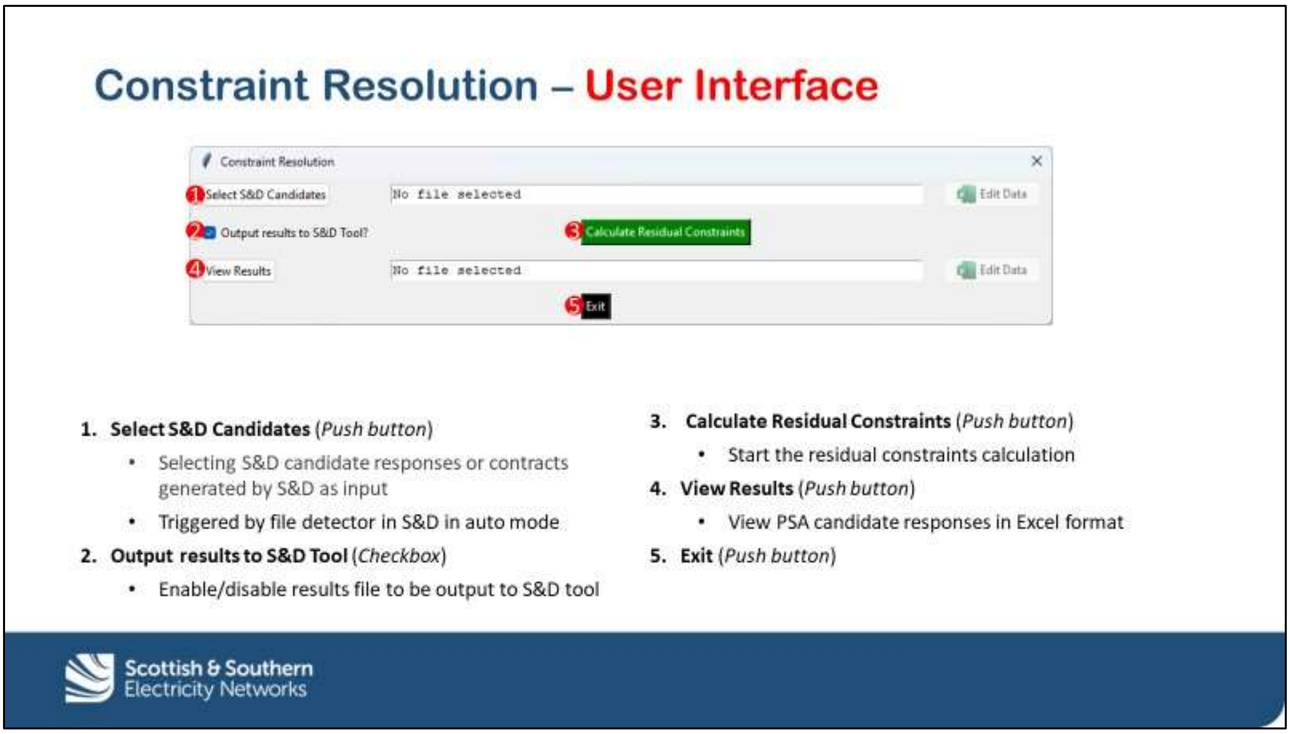
During the trials it was decided to introduce a minimum threshold for the sensitivity factor. This is defined by the parameter PSA_SF_THRESHOLD in PSA_SND_Config.txt and was set to 0.001. If any sensitivity factor was less than this threshold value it was set to 0 (zero). This resulted in significant improvements in the S&D “solver” algorithm and reduced the number of iterations dramatically, see next section on Constraint Resolution.

3.3 Constraint Resolution

The process of determining if the constraints have been resolved can be performed manually through the GUI or automatically through the PSA and S&D interface. The two different methods use the same code, one via the GUI and the other as a sub-process to enable multi-processing of results.

During the trials the normal operation was using the automated processing mode with PSA and S&D automatically passing data files between themselves and processing the results as per the earlier diagram.

The GUI is shown below:



3.3.1 Processing Steps

The initial steps of determining if constraints have been resolved are very similar to the other core processes. The process is split into two stages, initialisation and calculations (including outputs):

The initialisation steps, in order, are as follows:

Load config data	Read PSA_SND_Config.txt
Load PSArunID data	Read PSArunID RUN TIME PARAMETERS.txt
Start PF and load project	Get a PF licence, load PF model and activate the project
Load config data	Read PSA_SND_Config.txt
Load PSArunID data	Read PSArunID RUN TIME PARAMETERS.txt
Start PF and load project	Get a PF licence, load PF model and activate the project

Load validation data	Master asset register contains a list of all valid assets
Load S&D input file	Read SND_CANDIDATE-RESPONSES_Vn-i input file
Aggregate responses	Aggregate responses to improve processing
Calc historical flag	Included in input file, if TRUE then read SIA/NeRDA data from file, otherwise from API
Load SIA data*	See above
Load NeRDA data**	See above
Load EVENTS file	Read EVENTS for each scenario
Load SWITCH file	Read SWITCH changes for each scenario
Validate all inputs	Check all asset names against master register
Calc downscaling factors	Can be calculated in advance
Main calculation loop	See below…

*If calc historical is FALSE load new SIA data to reflect any changes in SIA forecast
**If calc historical is FALSE load new NeRDA data to reflect current switch positions

The calculation steps, in order, are as follows:

Step	**Description**
Loop through each scenario	The responses are sorted by scenario
Loop through times of constraints	The responses are sorted by start time
Update asset in/out service	Apply maintenance and contingency in/out of service to assets
Apply downscaling factor	Needs to be done for every calculation
Update events	Update additional generation/demand events
Update switches	Update switch positions from default NeRDA positions
Loop through each constraint	Determine if each constraint is resolved
Apply offered power	Apply each offered power in the input file as appropriate
Load flow calculations	Run load flow and save results
Determine any additional flex required	Calculate what power (kW) is required to resolve the constraint
Save and output results	Save results to output file PSA_CANDIDATE-RESPONSES_Vn-i

3.3.2 Outputs

There are a number of different intermediate outputs which are detailed in the Low Level Design. The key output from this process is the value of additional flex required field in the output file PSA_CANDIDATE-RESPONSES-Vn-i.

The intermediate outputs are stored in the PSArunID sub-folder TEMP_CR_FILES, the main PSA output is stored in the top-level PSArunID folder:

XXX-YYYY-MM-DD-hh-mm_PSA_CANDIDATE-RESPONSES-Vn-i.xlsx

The Vn-i suffix in the input and output filename enables S&D to keep track of the different asset combinations and iterations that it sends to PSA for processing.

The “n” in Vn-i identifies the combination of assets and starts at 0 and is incremented for each subsequent combination.

The “i” in Vn-i identifies the iteration across this combination of assets, it starts at 0 and is incremented for each subsequent calculation. The number of iterations was controlled by S&D. During the trials a number of improvements were made to the processing, including the PSA_SF_THRESHOLD (see earlier) and the “law of diminishing returns” in the S&D “solver” algorithm, which stopped the iterations if there was only a very small convergence between iterations. In addition to these changes the maximum number of iterations allowed was set to 10.

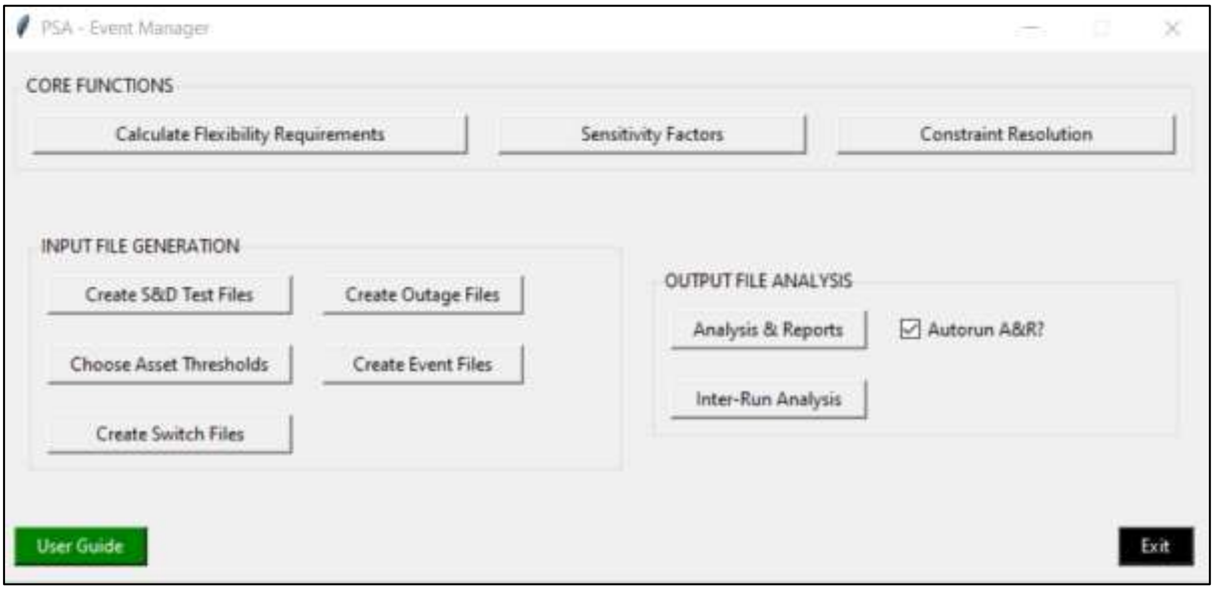
PSA will always use the same values of Vn-i as the input file.

3.4 Data File Creation and Analysis

Some of the key user requirements were around the ability to easily edit and create input files and perform analysis of the final results. The following sections cover these two topics.

3.4.1 Input File Generation

The successful creation of input files became very apparent during the trials and the input file generation options were created and made available through the main GUI.



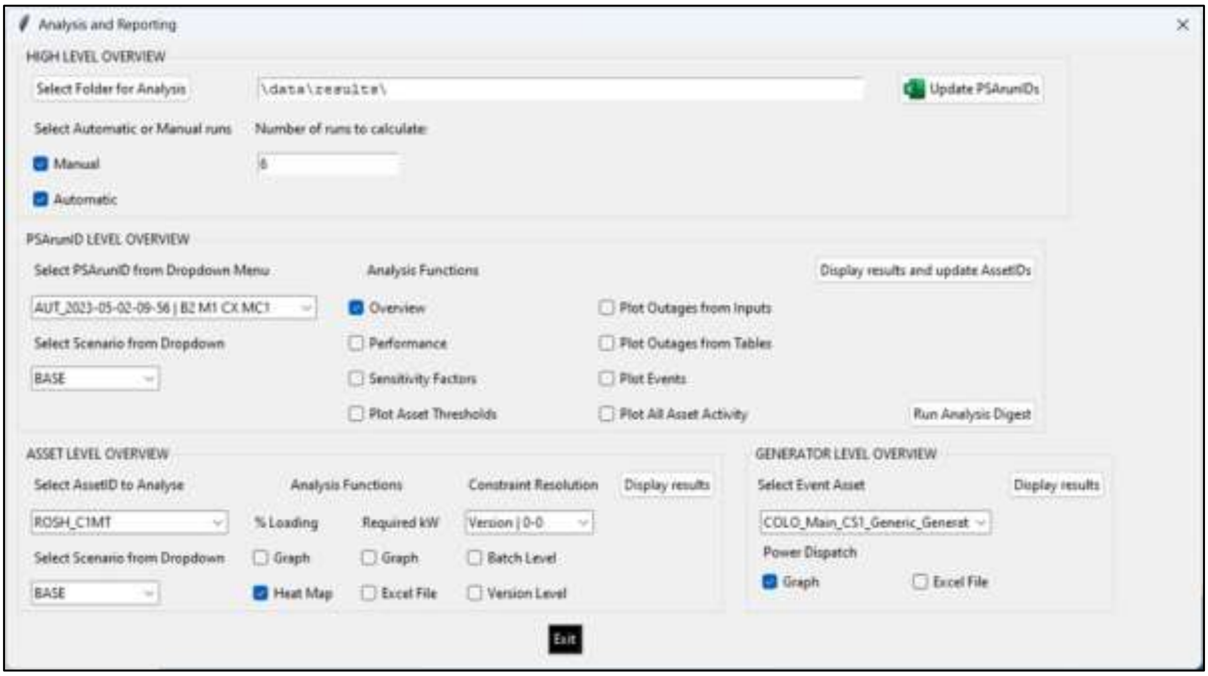
Each of these Input File Generation functions is explained briefly:

Function	Description
Create S&D Test Files	Enables the user to create dummy S&D output files based upon real PSA output files.
Create Outage Files	Create maintenance and contingency files which are validated and able to be viewed graphically in a scenario timeline
Choose Asset Thresholds	This enables users to select %loading thresholds for assets that will reliably create constraints.
Create Event Files	Create EVENTS files which are validated and able to be viewed graphically in a scenario timeline. These determine when assets can generate or demand more power and are used to simulate complex scenarios.
Create Switch Files	Create SWITCH files which are validated and able to be viewed graphically in a scenario timeline. These determine when switches are change state between ON and OFF and are used to simulate complex scenarios. These override NeRDA positions.

3.4.2 Output File Analysis

There area large number of output files created. These analysis functions allow users to view data in a graphical form for use in reports.

The GUI is shown below:



There are three levels of analysis which are briefly described further:

Level	Description
High	Displays a selection of PSArunIDs (AUT or MAN) and the number of constraints detected in each of the scenarios BASE(B), MAINT(M), CONT(C), and MAINT_CONT(MC)
PSArunID	Enables a wide range of analysis functions to be selected for the chosen PSArunID and selected Scenario
Asset	For the selected PSArunID (above) enables detailed graphs and statistics to be displayed, especially the “heat maps”
Generator	For the selected PSArunID (above), if any EVENTS assets exist then displays of power generation/demand are displayed

4 SOFTWARE ENVIRONMENT

The software was written using Python (version 3.9.13). Visual Studio Code, for editing, and a source code configuration control solution (GitBash) for configuration control purposes.

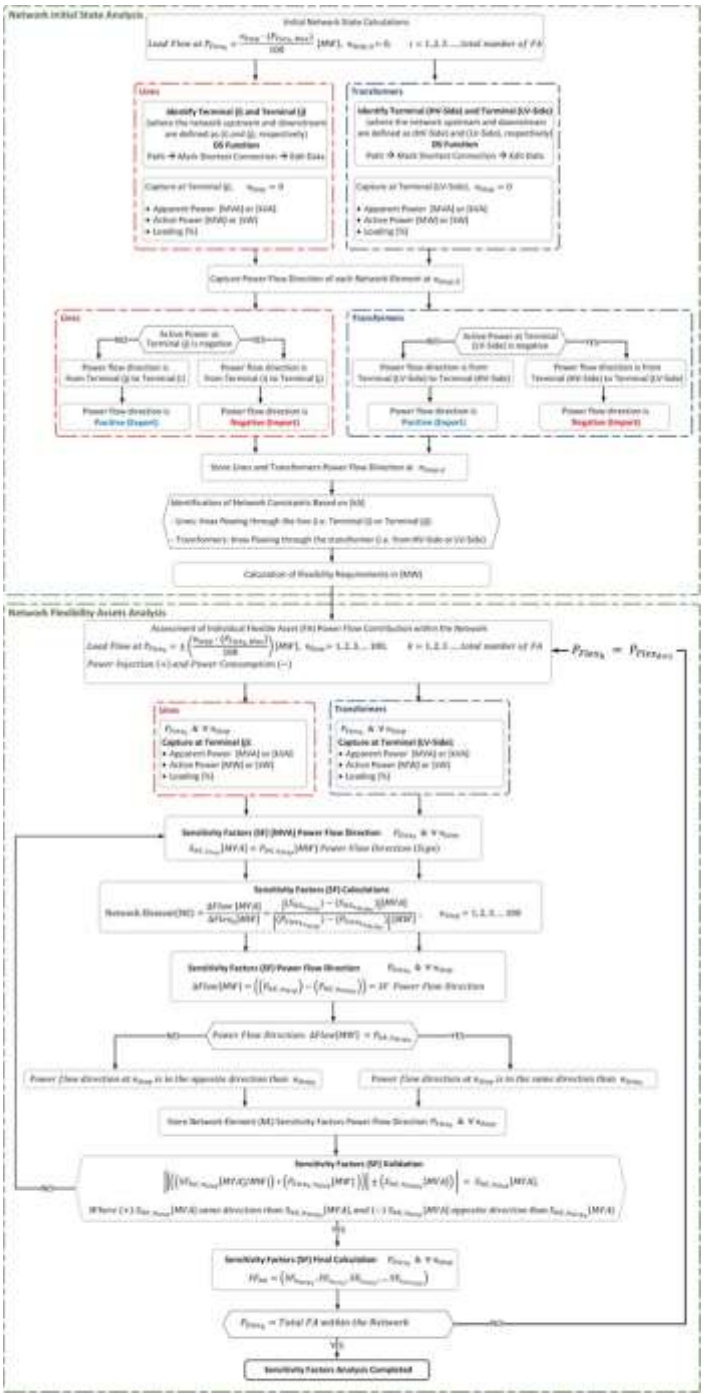
The source code was held in an SSEN Azure devops environment. https://vsosse.visualstudio.com/Networks-Transition-Technical-Trials-PSA

These Python scripts provide the user interface, access the external SIA and NeRDA data, read and process all local user created input files, and provide a processing wrapper around Power Factory.

They also provide the interface to S&D through the File Detection System that was also written in Python.

5 RINA Sensitivity Factor Algorithm

This diagram defines the precise algorithm for the Sensitivity Factor calculations and analysis work conducted by RINA



6 PSA S&D Configuration File

The file PSA_SND_Config.TXT contains setup and configuration data to enable PSA to run. It is read multiple times throughout the processing. Its contents are explained below:

Item	Description
SND_PSA_FILE_DETECTOR_FOLDER	Folder for S&D to PSA file transfer
PSA_SND_FILE_DETECTOR_FOLDER	Folder for PSA to S&D file transfer
PSA_SND_ROOT_WORKING_FOLDER	Top level PSA folder for all data
PSA_SND_UTC	Boolean for clock changes
PSA_SND_EMAIL	Email address
PSA_SND_SIA_USER	SIA login username
PSA_SND_SIA_TOKEN	SIA API token
PSA_SND_SIA_TIMEOUT	Timeout (secs) for SIA URL access
PSA_SND_SIA_BASE_URL	SIA URL for data
PSA_SND_SIA_LOGIN_URL	SIA URL for login
PSA_SND_SIA_URL_FEEDER	SIA API URL for feeder demand data
PSA_SND_SIA_URL_GROUP	SIA API URL for group demand data
PSA_SND_SIA_URL_GEN	SIA SPI URL for generation data
PSA_SND_SIA_PARAMS	SIA output Excel tab names
PSA_SND_NERDA_USER	NeRDA username
PSA_SND_NERDA_TOKEN	NeRDA API token
PSA_SND_NERDA_URL	NeRDA data access URL
PSA_SND_NERDA_TIMEOUT	Timeout (secs) for NeRDA URL access
PSA_SND_CALC_FLEX_REQTS_INTERVAL	Repeat period for PSA AUT runs
PSA_SND_HH_MODE	Boolean to create Half Hour data
PSA_SF_THRESHOLD	Sensitivity Factor threshold
PSA_SND_PF_USER	PowerFactory username
PSA_SND_PF_PATH	PowerFactory library path
PSA_PF_LDF_MODE	PowerFactory load flow parameter
PSA_PF_ALGRTHM	PowerFactory load flow parameter
PSA_PF_TRAFO_TAP	PowerFactory load flow parameter
PSA_PF_SHUNT_TAP	PowerFactory load flow parameter
PSA_PF_ZIP_LOAD	PowerFactory load flow parameter

PSA_PF_Q_LIMITS	PowerFactory load flow parameter
PSA_PF_PHASE_SHIFT	PowerFactory load flow parameter
PSA_PF_TRAFO_TAP_LIMIT	PowerFactory load flow parameter
PSA_PF_MAX_ITER	PowerFactory load flow parameter

7 Run Time Parameters File

The run time parameters file contains the data used to run that specific PSArunID, it is located in the top level PSArunID folder and has the following format and contents:

Filename: AUT_YYYY-MM-DD-hh-mm-RUN TIME PARAMETERS.TXT

Item	**Description**
BSP_MODEL	Name of PowerFactory .pfd network model
PRIMARY	Which single (or all) primary selected to model
ASSET_DATA	Name of file containing the generic asset data
SWITCH	Boolean – Are SWITCH events used
SWITCH_FILE	Name of SWITCH file if used
EVENTS	Boolean – Are EVENTS used
EVENT_FILE	Name of EVENTS file if used
MAINTENANCE	Boolean – MAINT scenario
MAINT_FILE	Name of MAINT file if used
CONTINGENCY	Boolean – CONT scenario
CONT_FILE	Name of CONT file if used
TIME_STEP	Not used
KW_STEP	Not used
POWER_FACTOR	Power factor reduction from feeders to secondary
LAGGING	Leading or lagging application of power factor
LINE_THRESHOLD	Default line %overload threshold
TX_THRESHOLD	Default transformer %overload threshold
AUTOMATIC	Defines running mode (AUT or MAN)
DAYS	Number of days to process
HALF_HOURS	Number of half hours to process (used for testing)

*** End of Document ***