transition

Moving to a smart future

# Power System Analysis (PSA) Software Final Report

July 2023

Scottish & Southern Electricity Networks

# Table of Contents

# 1 INTRODUCTION

This document is the final report on the Power Systems Analysis (PSA) software that was implemented and used for the TRANSITION Technical trials.

If required, more detailed information can be found in the Reference Documents.

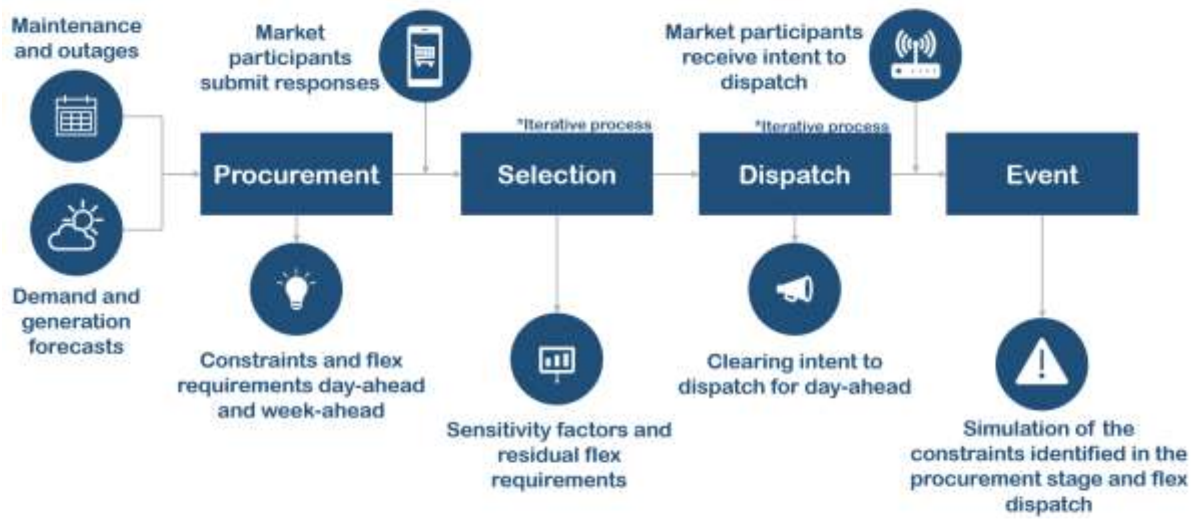
The purpose of this document is five-fold:

1. Executive summary – Provide a high-level management overview including key findings and recommendations
2. Provide an overview of the aims of the software, the key functional processes, and the design and operational constraints on the system
3. Provide a high-level overview of the key functionality of the software and the interfaces to other systems used during the trials
4. Provide a high-level overview of the results and analysis of the outputs
5. Provide details of the observations and learnings during the design, development, testing, and trials and the recommendations for any future BAU PSA system

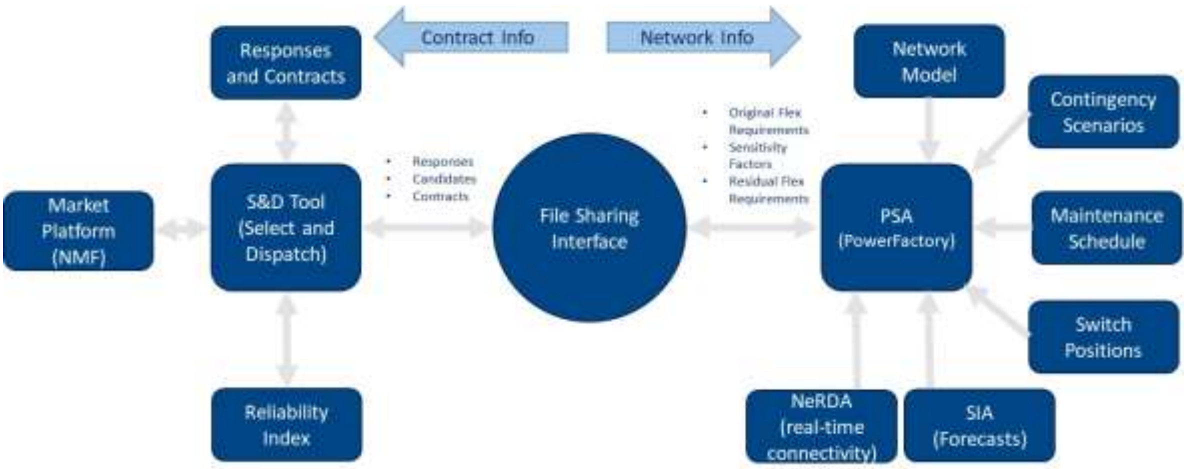
The PSA system is one element in the end-to-end process of identifying the need for, and procuring, flexible power. The other elements are the Select and Dispatch Tool (S&D) and Participants (through a swivel chair interface with the Market Platform (NMF).

The trial used the Cowley Local BSP (Bulk Supply Point) network, and specifically the Rose Hill Primary, with PSA creating simulated network constraints to enable Participants to respond with offers of flexible power to resolve those constraints, through a selection process managed by S&D. This included PSA calculating sensitivity factors (SFs) for S&D to use in the evaluation of the responses to determine those that have a beneficial/positive impact with the least cost. This also includes validating that the contracted flexibility would resolve the constraints completely in an iterative process between S&D and PSA.

The end-to-end process is shown below, this encompasses the high-level interactions between all the elements in the process.




There is a clear distinction to the types of data that PSA and S&D process. PSA processes Network information, including interfaces to external data sources for forecast demand and generation data and network switch positions. S&D processes contractual information and the interface to Participants. PSA and S&D communicate and share data through a file-based interface, as shown below:




One key design aim was to develop an in-house PSA capability that was completely independent from other systems and that used PowerFactory as the main load flow calculation engine, this was achieved extremely effectively. The Select and Dispatch Tool (S&D) was developed by TNEI through a competitive procurement process.

The other sections in this document go into more detail, although still at a high-level. More detailed information can be found in the reference documents.

# 2 EXECUTIVE SUMMARY

The development of the PSA software was a pilot project for the scripting and automation of PowerFactory. The PSA software was designed, developed, tested and operated in-house as part of the TRANSITION Technical Trials project. The PSA software was developed in Azure DevOps, written in Python and interfaced to PowerFactory software for the core load flow calculations.

## 2.1 PSA and Process Steps (within the end to end process)

The PSA software core processes are listed below, but other elements include input file creation, scenario specification, parameter setting, and detailed analysis of results.

Due to the nature of the network area covered by the trials, PSA was required to simulate the constraints but still using live forecast data and near real-time switch configuration data.

The core PSA processes are:

- **Enable simulation of constraints:** Enable users to enter maintenance, contingency, switching, and generation events to simulate constraints.
- **Identify network constraints**: Identify power constraints on the network by detecting assets that exceed their % loading threshold.
- **Calculate flexibility requirements**: calculate the flexibility power requirements of the constrained assets at the point of the constraint.
- **Calculate sensitivity factors**: Calculate SFs for participant assets responding with offers to resolve the constraint(s). This is triggered by inputs from S&D.
- **Constraint resolution**: Determine if proposed solutions, from S&D, resolve the constraint(s) and can be dispatched via outputs to S&D, or if additional flexibility is required. This residual required power is key to enabling the solution to converge.

During the trials PSA performed 419 individual runs, simulating 49,040 constraints across SPM, SEPM, Secure, and Dynamic scenarios. The performance of the current software identified some improvements would be required, however indicating that a PSA solution would be capable of meeting any half hourly processing requirement, especially for Dynamic requests.

## 2.2 Sensitivity Factors

The concept of using SFs is new to the market and is a key differentiator in the process of selecting a participant asset in resolving the constraint(s). The SFs methodology and algorithm was developed in conjunction with RINA, a market leader in network analysis and electrical systems research. The final RINA/SSEN report is available as REF-007.

The SF is a measure of how effective the offered power is in resolving a particular constraint. It varies between constrained network assets, individual assets, and network configurations.

SFs are a measure of the amount of power that is lost in the journey from the provider to the constraint, and also whether the power injection has a negative or positive effect on the network. Or more precisely “A change in power flow in a network element with respect to a change in power import or export from a flexible asset”

These proved to be fundamental in the selection of the solutions to resolve the constraints created during the trials.

## 2.3 Constraint Resolution

The final process is to determine if the candidate responses from S&D resolve the constraint. This is an iterative process as S&D calculates the optimal solution based on PSA calculating if the constraint is resolved and any additional residual flexible power requirements.

## 2.4 External Interfaces and Exception Handling

PSA has a number of interfaces with external systems: S&D for handling proposed solutions to constraints; SIA Partner’s API for the 10 day ahead demand and generation forecasts; and Open Grid’s NeRDA API for near real-time network switch positions.

In general, the interfaces worked well, although a more robust exception handling process needs to be implemented in any BAU solution. The file-based interface between PSA and S&D worked very well, and the API’s worked well, however better exception handling and fault recovery needs to be implemented in a BAU solution. These are detailed in a later section – Learnings and Recommendations.

## 2.5 Observations and BAU considerations

The scope of the trials was limited to an area of the network that was well defined in terms of accurate network model (CIM PowerFactory model).

- More areas of the SSEN network need to be modelled to the same level of accuracy, to enable PSA studies for flexibility procurement.

The trial used the Cowley Local BSP network and created simulated network constraints. This part of the network is relatively small and doesn’t exhibit many constraints. This limited tests on scalability and performance for BAU considerations.

- The PSA system should be used on another part of the network that is more prone to exhibit real constraints rather than simulating them.

During the trials period, observations were made regarding the performance of the system and improvements that could be made in the speed of processing and exception handling. These are detailed in a later section – Learnings and

Recommendations. If the system was purely detecting constraints on a part of the network then it would be capable of meeting the design constraints for all of the possible scenarios (SPM, SEPM, Secure, Dynamic).

- The external interfaces from SIA Partners and Open Grid need to be more robust and fail-over process put in place, both within the external services provided and the PSA software

## 2.6 Conclusion and Recommendations

These conclusions and recommendations apply to the PSA software and any future BAU PSA solution. However, the ability for PowerFactory to be automated, through the use of scripts, and to perform time consuming, repetitive, and complex tasks has been achieved and should be built upon in-house. This should include investigating the current, or planned, manual use of PowerFactory in other areas of the business (e.g. connections) and the feasibility of automating those processes.

### 2.6.1 Conclusions

The key conclusions are as follows:

1. PowerFactory can be successfully automated through Python scripts and interfaced with other SSEN systems and data sources
2. SFs play a key role in the cost-effective selection of flexibility and the resolution of constraints through the deployment of flexible assets
3. The use of SFs and their impact will need to be communicated to Participants to enable a better understanding of how they impact the selection process
4. Accurate network models are essential, as they underpin the identification of constraints, and the final selection and dispatch of flexibility
5. Resilience and fail-over processes for external interfaces are essential

### 2.6.2 Recommendations

The key recommendations are as follows:

1. Investigate the feasibility of deploying PSA to part(s) of the network that exhibit real constraints to further understand how a BAU solution could scale and meet the processing requirements. This should include an interface to existing maintenance planning systems. The key benefits are:
    a. Continues the momentum behind PSA, and builds a BAU proof of concept
    b. Drives the creation of more accurate network data and CIM models
    c. Includes actual scheduled maintenance plans
    d. Creates a better understanding of scalability, hardware sizing, and PowerFactory licence requirements for future BAU deployment

2. Investigate the feasibility of automating non-flexibility related PowerFactory processes. The key benefits are:
    a. Maximise the investment made/planned in the use of PowerFactory
    b. Reduce manual effort and timescales of repetitive PowerFactory tasks through automation
3. Continue/accelerate the process of updating asset management and GIS systems to reflect the real world so accurate network models can be utilised in the future. The key benefits are:
    a. Accurate source data means you can apply computer models effectively with a high degree of confidence in the results
    b. Accurate source data means you reduce costs of maintenance and repairs because you know what assets you have where and what state they are in. Which means lower Customer Minutes Lost (CML) and Customer Interruptions (CI)

# 3 CONTEXT

## 3.1 Key Solution Aims (Objectives)

From a PSA perspective the objectives of the Technical Trials were as follows:

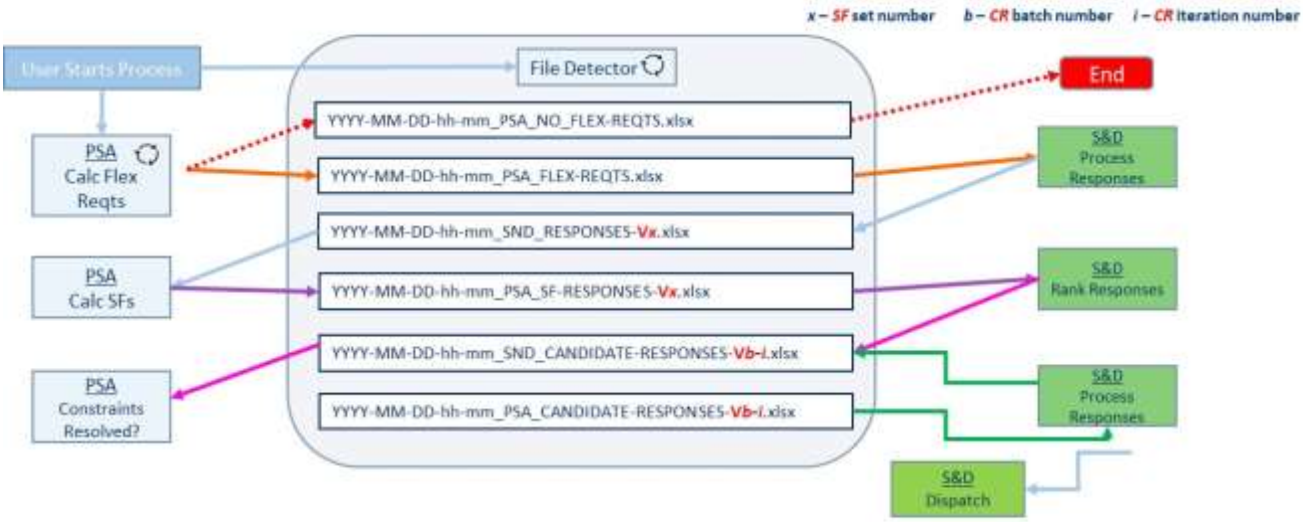
- Demonstrate that PowerFactory processes can be automated through scripts
- Deploy a standalone PSA solution
- Testing the end to end process for flexibility by:
    - Incorporating short term operational forecast & topological datasets
    - Calculating constraints on the network and resulting flexibility requirements
    - Validating responses and sensitivity factors
    - Determining if proposed dispatch(s) reduces overload to zero
- Using these services:
    - Sustain Peak Management SPM (including Export)
    - Secure Constraint Management SCM
    - Dynamic
- Across these time horizons:
    - Week and day ahead

## 3.2 Functional Processes

The high-level functions of PSA are to identify constraints on the network; calculate flexibility requirements; communicate these to S&D Tool; receive offers from S&D Tool; calculate network sensitivity factors associated with these responses and communicate these to S&D Tool; receive candidate responses from S&D Tool and determine if they resolve the constraint(s) and communicate this to S&D Tool.

These steps can be conducted in an automated process or step by step manual process to gain more insights and understanding.

The diagram below shows the flow of data between the two systems:



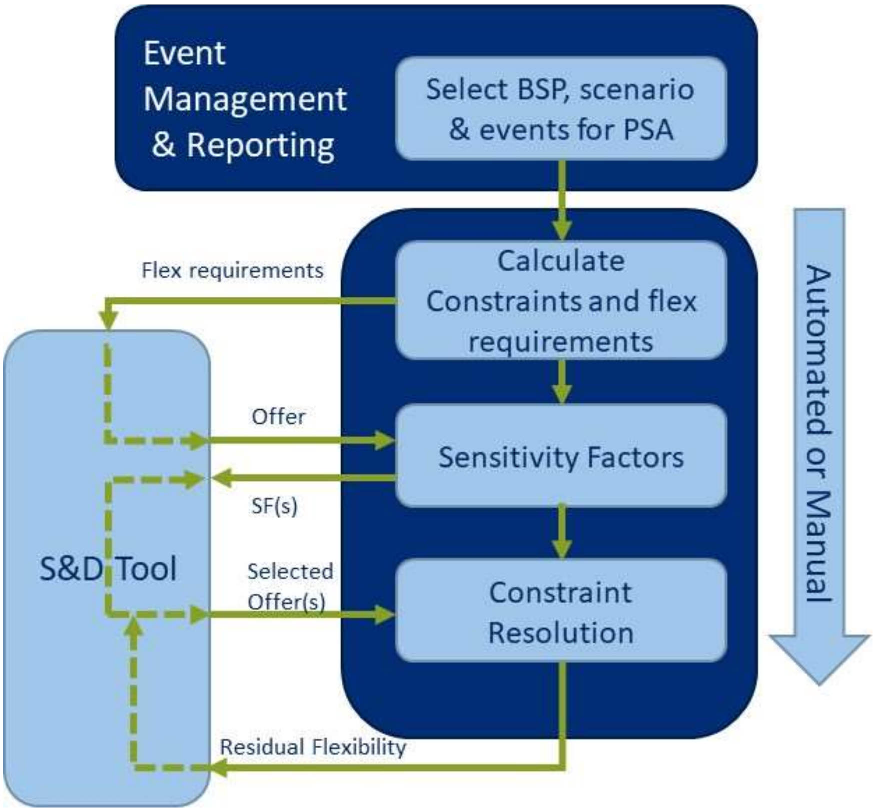

## 3.3 Constraints

The overall trials were constrained by the area of the network that was selected. The trial used the Cowley Local BSP network and created simulated network constraints. This part of the network is relatively small and doesn't exhibit many constraints. This limited tests on scalability and performance for BAU considerations.

# 4 SOLUTION OVERVIEW

## 4.1 Key Elements

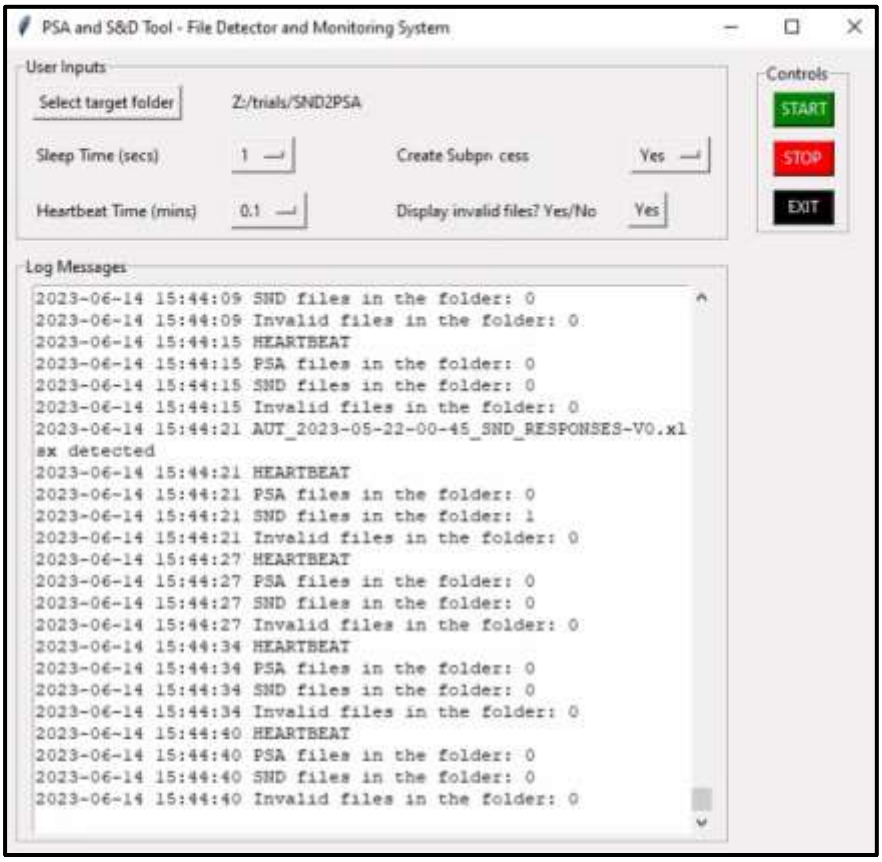
The PSA software comprises of two main parts, the File Detector System and the Event Manager control module. The key interactions and process steps between PSA and S&D are shown below:




The File Detector System sits between PSA and S&D and provides the file-based interface between the systems. This allows the end-to-end process to move through each individual stage. The diagrams below shows how the process works and the user interface:

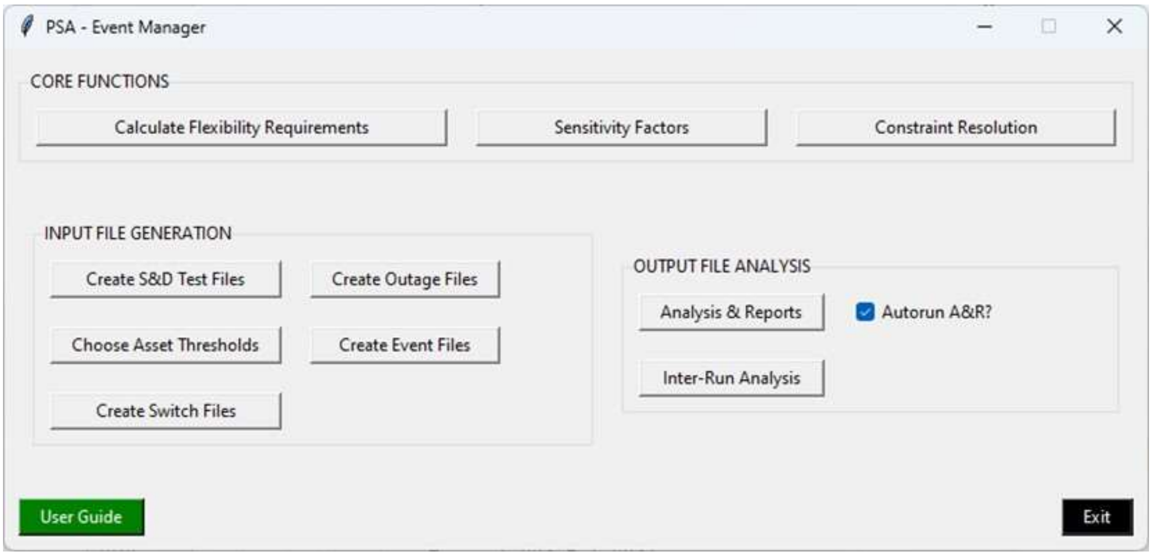
- PSA processes data and a saves file
- S&D Tool detects a file and processes data and saves a file
- PSA detects a file and processes data and saves a file
- Until an end state is reached
  - User stops process
  - Error condition


User interface for File Detector Systems:




The File Detector System is used for detecting and responding to requests from S&D to calculate SFs and perform constraint resolution processes.

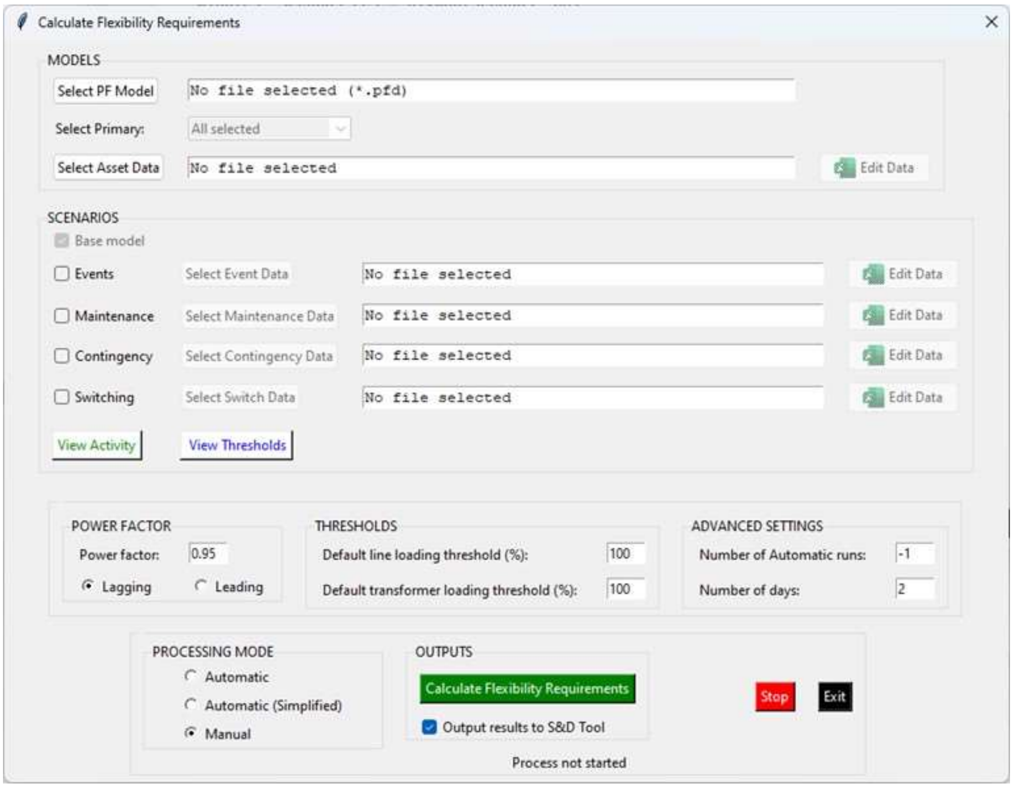
The Event Manager module enables all of the data input files to be created/edited and the core functions to be accessed:




The PSA core functions (shown above) are:

- **Calculate Flexibility Requirements**: Identify power constraints on the network and calculate the flexibility power requirements at the point of the constraint. This can run in manual or automated mode. To allow constant detection of constraints Automated mode is selected.
- **Sensitivity Factors**: Calculate SFs for participant assets responding with offers to resolve the constraint(s). This is triggered by inputs from S&D.
- **Constraint resolution**: Determine if proposed solutions, from S&D, resolve the constraint(s) and can be dispatched via outputs to S&D

They key function of calculating the flexibility requirements enables the user to specify all the input data files and the run-time parameters for scenarios to be modelled:




The other core functions of Sensitivity Factors and Constraint Resolution can be run in manual or automated mode, but during the trials were triggered automatically by requests from S&D.

# 5 RESULTS OF TRIALS

## 5.1 Parallel Running and Multi-Processing

One key element of the learnings was how quickly could a PSA solution calculate the results and could it be deployed to meet the flexibility use cases with the shortest demand time.

The original design of the PSA software enabled PowerFactory processes to be run in parallel through multi-processing sub-processes. This is especially relevant for enabling the core function of calculating flexibility requirements to operate independently of the sensitivity factor and constraint resolution processes.

During the trials the core function of determining flexibility requirements processed a run of results every 90 minutes using a timing mechanism. The actual run time of each PSA run was less than 90 minutes depending on the four scenarios selected (BASE, MAINT, CONT, MAINT_CONT).

In any BAU solution these four scenarios can be run in parallel and this processing time reduced to approximately 12 minutes (including other identified performance improvements). This would therefore enable the PSA and S&D processes to be completed within a 30 minute period if required.

The file detection system, responsible for detecting output from S&D and creating sub-processes to calculate sensitivity factors and check constraint resolution, ran continuously in parallel with the core PSA flexibility requirements process. However, the design of S&D didn't allow for parallel processing of multiple sensitivity factors or constraint resolutions.

The limiting factor to the number of multi-processes running in parallel is the number of PowerFactory licences. Seven licences were available, two full licences (that included GUI interface) and five much cheaper "engine mode" licences (purely used for running scripts).

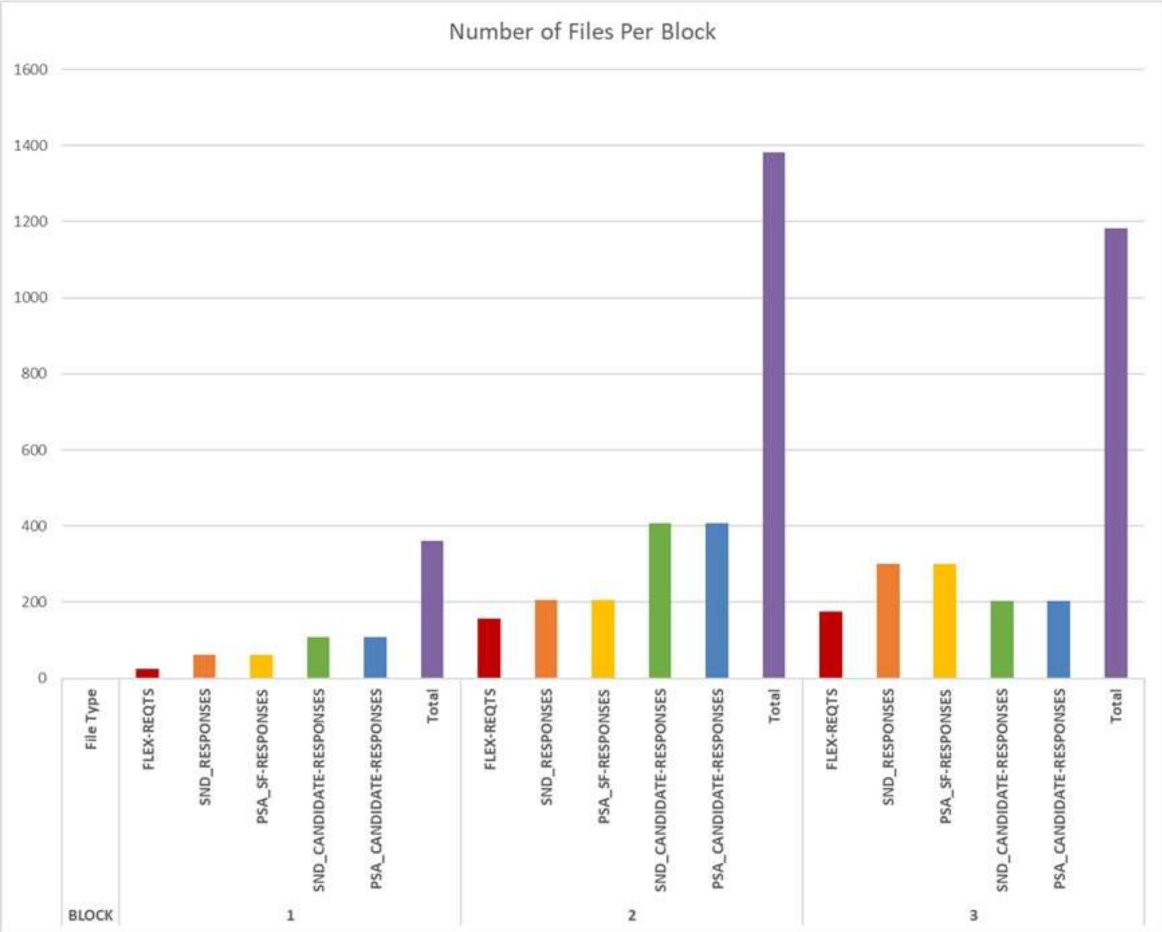
The number of PowerFactory licences required for any BAU solution would depend on the final design of the PSA software and the size of the network area being covered. However, the "engine mode" licences are more than adequate for running a BAU solution.

## 5.2 Key Processing Figures

The trials were conducted over a three month period and divided into 3 blocks. Each block included specific tests for SPM, SEPM, Secure and Dynamic requests. The overall volume of files processed and their processing times are highlighted below.

### 5.2.1 Processing Volumes

During the trials it was decided to automatically generate flexibility requirements every 90 minutes. The volume of files generated by PSA and the subsequent interactions with S&D across the three blocks of trials processing are shown below:




PSA also created responses to inputs from S&D from the sensitivity factor and constraint resolution processes. The outputs of these files included different versions and iterations within each version, but covered all scenarios:

| | |
|---|---|
| Sensitivity factor files (all versions) | 439 |
| Average number of SF versions | 1.5* |
| Constraint resolution files (all batches and iterations) | 850 |
| Average number of CR batches | 2* |
| Average number of CR iterations | 1.2* |

*These numbers varied greatly, however through improvements in processing logic and iteration convergence tolerances these numbers more accurately represent the final processing algorithms.

### 5.2.2 Processing Times

As stated above, PSA processed 419 runs, each run and scenario in each run will vary slightly in processing time. The key part of the processing time is the calculation of the loading on the assets over a time sequence of every 30 minutes for 11 days, that's a total maximum of 528 calculations. Any improvement in this half hourly loop will significantly improve the overall performance times. Running the scenarios in parallel, rather than sequentially, would also be a BAU essential requirement.

The broad range of processing times are as follows:

| **Processing element** | **Time (approx.)** |
|---|---|
| Initialisation | 45 secs |
| Downscaling factors/feeder calculations | 0.6 secs x 528 = 5 mins (approx.) |
| Load flow calculations | 0.5 secs x 528 = 4.5 mins (approx.) |
| Extract results from PowerFactory | 0.5 secs x 528 = 4.5 mins (approx.) |
| Total elapsed time (per scenario) | 14-16 mins |

The Sensitivity Factors and Constraint Resolution processes follow a very similar workflow and absolute processing times vary greatly depending on the number of entries in the input file received from S&D. The following table shows average processing times:

| | Initialisation | Each calculation | Typical duration |
|---|---|---|---|
| Sensitivity Factor | 45 secs | 1.3 secs | 3-4 mins |
| Constraint Resolution | 45 secs | 7.8 secs | 1-2 mins |

These figures can be extrapolated out using the number of entries in the S&D input file. The size of these files varied significantly throughout the trial depending on the individual scenario being modelled, from 5-10 entries to a few ‘000s.

## 5.3 Sensitivity Factors (SFs)

A key learning from the trials was the use of SFs as part of the selection process within S&D to offer candidate responses to PSA as possible solutions to resolve the constraints.

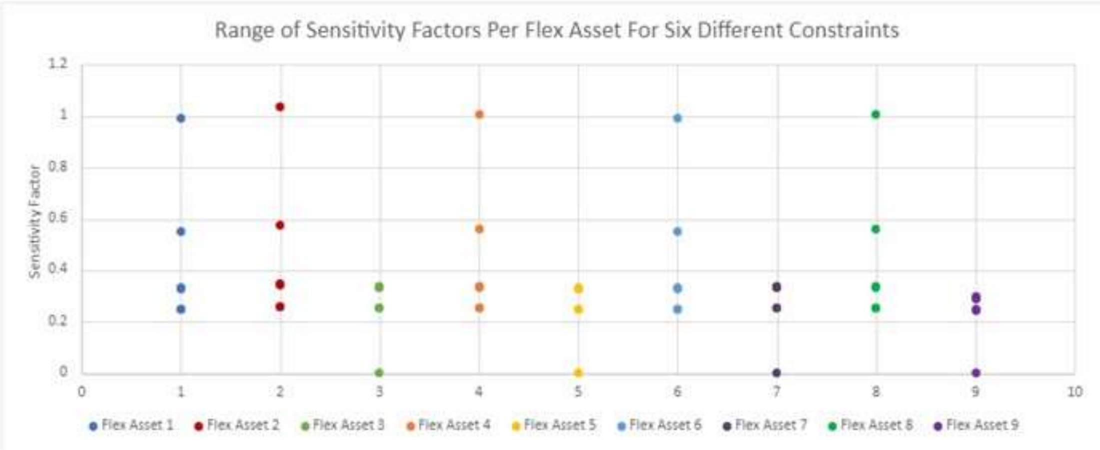
The methodology and the algorithm used to calculate the SFs was developed in conjunction with RINA, a market leader in the field of electrical networks and associated research.

One of the technical objectives of PSA was to determine the amount of power required at the point of the constraint that will resolve that constraint. This requires the calculation and use of SFs to adjust the amount of power offered by the Participants and their asset. This more accurately models the actual impact, on the constraint, of the power being injected at the location of the Participant’s asset on the network.

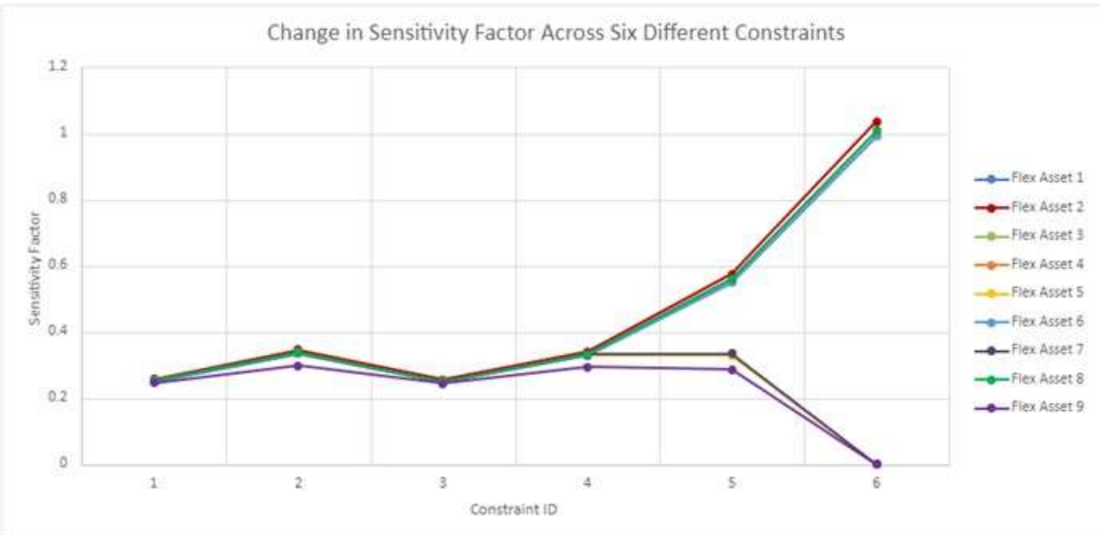
This means that the SF for a Participant's asset varies from request to request depending on the current network conditions, this can be shown in the graphs below:

The data has been made anonymous so that specific participants and assets cannot be identified.

The first graph shows nine flexible assets and their associated SFs for six different constraints. Some SF values are duplicated and so do not show on the graph. This demonstrates that SFs for a specific asset can vary greatly depending upon network conditions.




This second graph shows the same nine flexible assets and six constraints but demonstrates how SFs vary differently for assets against specific constraints.



# 6 LEARNINGS AND RECOMMENDATIONS

This section details the issues and observations encountered during the trials and the subsequent learnings and recommendations. Some of the learnings, especially performance improvements, were implemented during the trials period.

## 6.1 Data

1. Use static data (fixed pre-processed data) as much as possible, minimise reading data from PF during processing as much as is practical
2. Validate all data before going live (SIA, NeRDA, PF, EO etc)
3. Ensure data consistency in asset names between different data sources (SIA, NeRDA, PF, EO etc)
4. Consistent use of time zone in all areas, GUI, API, calculations, outputs (UTC v GMT)
5. Use of timeseries 2d arrays (day and HH) for the storage of all time-based data. Includes switching, events, maintenance, SIA forecast. Easier to visualise and manipulate if all in a consistent format.
6. Interface to planned maintenance schedules (BAU operations)
7. Model contingency more appropriately (based on actual live data)
8. The non-linearity of SFs may be largely irrelevant unless the asset is at a very low or high loading % or the injected power is significant. This requires further investigation. Ref SF report from JPO/RINA
9. Importance of SF direction (helpful or harmful) (sign of SF +ve or -ve)

## 6.2 Performance

1. Investigate the need to calculate all 11 days (0-10 days) flex reqts if forecast data accuracy falls off after 3-4 days – BASE only
2. Optimise (even more) the inner most half hour processing loop (see PowerFactory below) 48 x HH x 11 days = 528 calculations per scenario at 0.5 secs per calc = 4 mins 24 secs
3. Design the code so that PSA calc flex reqts can run in parallel (scenarios each use a separate licence)
4. Reduce need to load the network model every time for SFs and CRs. Evidence that S&D requested SFs quickly in sequences as well as CRs, large overhead 45+ secs to load PF model each time
5. Consider SF pre-calculated tables and SF threshold (even greater than 0.001) for setting SF = 0
    a. Especially for normal running mode
    b. Known constrained asset to DER pairings in normal running mode

c. Investigate a simplified SF calculation based on interpolation of existing data points

6. Downscaling factors – Investigate a faster way of calculating and accessing within HH loop –Vectorize lambda functions
7. Storage of PF objects in data frames implemented to significantly improve performance – Only look for PF objects once and store in DF rather than PF
8. Enable any S&D like tool to operate in parallel, sequential slowed the process down substantially
9. Understand and correct when system sleeps and doesn't always wake up, increase priority of process or write own sleep function. This was seen between the automatic repeat cycles and when accessing the PF results (very intermittent, every 6 weeks, and not repeatable) – Needs further analysis
10. Investigate the RINA SF calculation for small changes in inputs producing large changes on SF values at different levels of % loading

## *6.3 PowerFactory*

1. Location of database. Unless you really need to have a networked copy and changes reflected across the network, make local copies, network access is very slow
2. Use PF engine mode licence to run scripts and no GUI, reduced licence costs
3. PF database read/writes – enable/disable local cache to improve performance significantly
4. PF multi-processing – manage workspaces in the system, watch out for not deleting workspaces after use
5. Licence use per user, limited to one process per user unless multi-processing mode and careful management of workspaces
6. PF has parallel capability but not for load flow calcs, only for contingency analysis and other complex calcs
7. Investigate the use of PF time series functionality for load flow calcs to step through the HH steps and update the Plini and Qlini values for all loads rather than individually

## *6.4 Exception Handling*

1. PSA and S&D Interface could be improved with better error handling and progress updates
2. SIA API calls, error handling, keep last read data as fail-over mechanism
    a. Validation of SIA data – Check magnitude MW or KW etc, UTC
    b. Better updates from supplier on software changes
3. NeRDA API calls, error handling, keep last read data as fail-over mechanism

   a. Better updates from supplier on software changes

4. PowerFactory load flow calculations, do what on error? We didn't experience any errors unless the SIA input data was wrong.
   a. Cowley Local BSP was unusually stable, needs testing on other networks
   b. Keep results and use PF to determine what went wrong
   c. Design in fault handling/tolerance if load flow calc returns an error, generally this will only occur if the input data (SIA) is wrong

## 6.5 External Hardware/Software Support and Testing

1. HCL hardware provision was adequate, no more than that, no real service
2. HCL software support was poor.
3. HCL/SSEN relationship needs to refocus on a customer service oriented approach rather than transactional process
4. Infosys testing was an unexpected time consuming overhead that added little value and only trained the Infosys staff

# 7 APPENDIX - PSA “AS-BUILT” SOFTWARE

This section contains a link to the final detailed documentation of the “As-Built” PSA software and any design/development decisions and the reasons behind them.

REF-006 PSA As-Built Software v1.0.docx

*** End of Document ***