POWERFACTORY

PowerFactory 2022

User Manual

PF2022

POWER SYSTEM SOLUTIONS
MADE IN GERMANY

DIgSILENT PowerFactory

Version 2022

User Manual

Online Edition

DIgSILENT GmbH

Gomaringen, Germany

August 2022

Publisher:
DIgSILENT GmbH
Heinrich-Hertz-Straße 9
72810 Gomaringen / Germany
Tel.: +49 (0) 7072-9168-0
Fax: +49 (0) 7072-9168-88

Please visit our homepage at:
https://www.digsilent.de

August 2022
r8992

Contents

Part I

General Information

Chapter 1

About this Guide

This User Manual is intended to be a reference for users of the DIgSILENT *PowerFactory* software. This chapter provides general information about the contents and the used conventions of this documentation.

1.1 Contents of the User Manual

The first section of the User Manual provides General Information, including an overview of *PowerFactory* software, a description of the basic program settings, and a description of the *PowerFactory* data model.

The next sections describe *PowerFactory* administration, handling, and power system analysis functions. In the Power System Analysis Functions section, each chapter deals with a different calculation, presenting the most relevant theoretical aspects, the *PowerFactory* approach, and the corresponding interface.

The online version of this manual includes additional sections dedicated to the mathematical description of models and their parameters, referred to as Technical References. To facilitate their portability, visualisation, and printing, the papers are attached to the online help as PDF documents. They are opened by clicking on the indicated links within the manual.

It is recommended that new users commence by reading Chapter 4 (*PowerFactory* Overview), and completing the *PowerFactory* **Tutorials**.

1.2 Used Conventions

Conventions to describe user actions are as follows:

Buttons and Keys Dialog buttons and keyboard keys are referred to with bold and underline text formatting. For example, press the **<u>OK</u>** button in the *PowerFactory* dialog, or press **<u>CTRL+B</u>** on the keyboard.

Menus and Icons Menus and icons are usually referenced using *Italics*. For example, press the *User Settings* icon, or select *Tools* → *User Settings...*

Other Items "Speech marks" are used to indicate data to be entered by the user, and also to refer to an item defined by the author. For example, consider a parameter "x".

Chapter 2

Contact

For further information about the company *DIgSILENT*, our products and services please visit our web site, or contact us at:

DIgSILENT GmbH

Heinrich-Hertz-Str. 9

72810 Gomaringen / Germany

www.digsilent.de

2.1 Direct Technical Support

DIgSILENT experts offer direct assistance to *PowerFactory* users with valid maintenance agreements via telephone or online via support queries raised on the customer portal.

To register for the on-line portal, select *Help* → *Online User Registration...* or go to directly to the registration page (link below). Log-in details will be provided by email shortly thereafter.

To log-in to the portal, enter the email (or Login) and Password provided. When raising a new support query, please include the *PowerFactory* version and build number in your submission, which can be found by selecting *Help* → *About* PowerFactory ... from the main menu. Note that including relevant *.pfd file(s) may assist with our investigation into your query. The customer portal is shown in Figure 2.1.1.

Phone: +49-(0)7072-9168-50 (German)
+49-(0)7072-9168-51 (English)

User Registration: https://www.digsilent.de/en/user-registration.html

User Login: https://www.digsilent.de/en/user-login.html

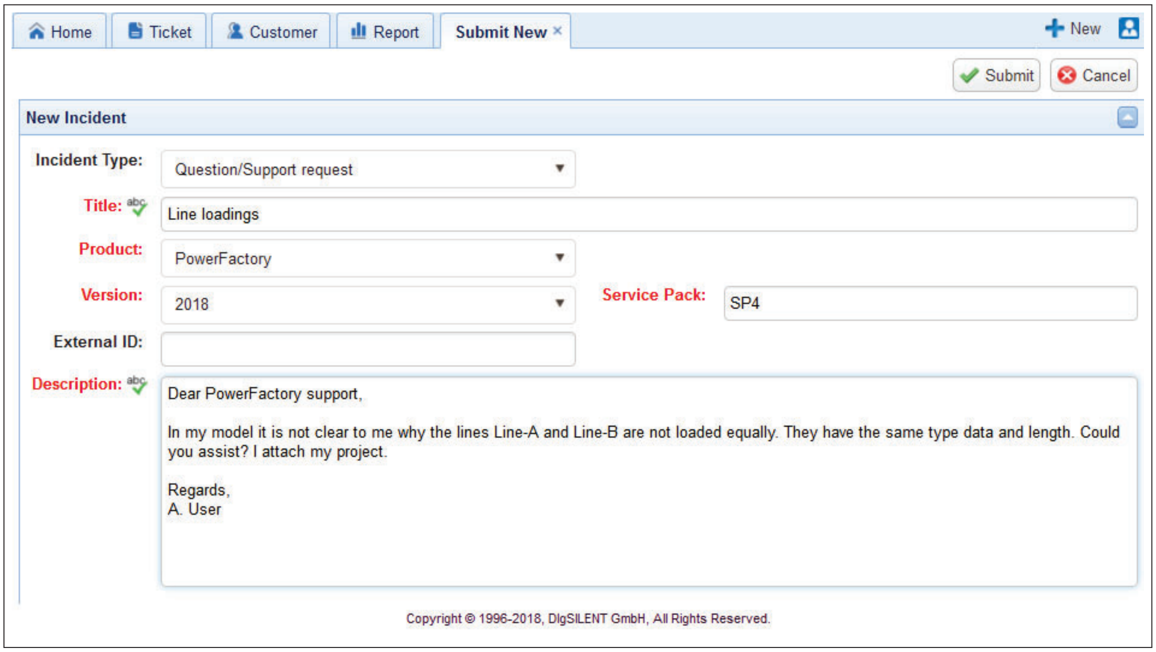


Figure 2.1.1: DIgSILENT customer portal

2.1.1 *PowerFactory* Support Package

Sometimes as part of a ticket investigation customers are asked to provide a "*PowerFactory* Support Package". This can be generated from within *PowerFactory* from the top menu by doing *Help* → *Create support package...*. The support package can be saved on the user's computer then attached to the ticket. The contents of the "*PowerFactory* Support Package" are listed in the Advanced Installation and Configuration Manual.

2.1.2 Licence Support Package

If a customer's problem is thought to be licence related, the customer may be asked to provide a "Licence Support Package". This can be generated from the Licence Manager, where a *Create Licence Support Package* button is provided. The support package can then be saved on the user's computer then attached to the ticket.

2.2 Knowledge Base

A "Knowledge Base" database of information, based on an FAQ format, is available for any users (whether registered or not) to look for answers to their questions. The knowledge base contains interesting questions and answers regarding specific applications of *PowerFactory*.

Knowledge Base: https://www.digsilent.de/en/faq-powerfactory.html

2.3 General Information

For general information about *DIgSILENT* or your *PowerFactory* licence, please contact us via:

Phone: +49-(0)7072-9168-0

Fax: +49-(0)7072-9168-88

E-mail: mail@digsilent.de

Chapter 3

Documentation and Help System

DIgSILENT *PowerFactory* is provided with a complete help package to support users at all levels of expertise. Documents with the basic information of the program and its functionality are combined with references to advanced simulation features, mathematical descriptions of the models and of course application examples.

PowerFactory offers the following help resources, which can be accessed either directly from *PowerFactory* or from the *DIgSILENT* download area (`https://www.digsilent.de/en/downloads.html`):

- **Getting Started:** a document describing the first steps to follow after receiving the installation DVD or downloading the software from the *DIgSILENT* download area. The Getting Started document covers the basic installation options.
- **Advanced Installation and Configuration Manual:** in this document, advanced installation options e.g. multi-user database, installation on an application server, and the Offline mode installation, are covered. The Offline mode guide is available in Section 5.5: Offline Mode User Guide.
- **Tutorials:** information for new users and hands-on tutorials. Access via Help menu of *PowerFactory*.
- **User Manual:** this document. Access via Help menu of *PowerFactory*. Current and previous manuals (PDF files) can also be found on the in the *DIgSILENT* download area.
- **Technical References:** description of the models implemented in *PowerFactory* for the different power systems components. The technical reference documents are available on the menu *Help* → *Technical References*.
- **Additional Packages:** additional information and/or examples about specific *PowerFactory* functions are available on the menu *Help* → *Additional Packages*. The additional packages are:
  - Programming Interface (API)
  - DGS Data Exchange Format
  - OPC Interface
  - DPL Functions Extensions
  - Dynamic Models C Interface
- **Context Sensitive Help:** pressing the key **F1** while working with *PowerFactory* will lead directly to the related topic inside the User Manual.
- ***PowerFactory* Examples:** the window *PowerFactory* Examples provides a list of application examples of *PowerFactory* calculation functions. Every example comes with an explanatory document that can be opened by clicking on the *Show Documentation* button (📖). Additionally, videos demonstrating the software handling and its functionalities are available.

The *PowerFactory* Examples window will “pop up” automatically every time the software is open, this could be deactivated by unchecking the *Show at Startup* checkbox. *PowerFactory* Examples are also accessible on the main menu, by selecting *File* → *Examples. . .* .

- **Release Notes:** for all new versions and updates of the program *Release Notes* are provided, which document the implemented changes. They are available on the menu *Help* → *Release Notes*
- **Knowledge base:** see Chapter 2: Contact
- **Technical Support:** see Chapter 2: Contact
- **Website:** www.digsilent.de

Chapter 4

PowerFactory Overview

The calculation program DIgSILENT *PowerFactory*, is a computer-aided engineering tool for the analysis of transmission, distribution, and industrial electrical power systems. It has been designed as an advanced integrated and interactive software package dedicated to electrical power system and control analysis in order to achieve the main objectives of planning and operation optimisation.

"*DIgSILENT*" is an acronym for "**DIg**ital **SI**mu**L**ation of **E**lectrical **N**e**T**works". *DIgSILENT* Version 7 was the world's first power system analysis software with an integrated graphical single-line interface. That interactive single-line diagram included drawing functions, editing capabilities and all relevant static and dynamic calculation features.

PowerFactory was designed and developed by qualified engineers and programmers with many years of experience in both electrical power system analysis and computer programming. The accuracy and validity of results obtained with *PowerFactory* has been confirmed in a large number of implementations, by organisations involved in the planning and operation of power systems throughout the world.

To address users' power system analysis requirements, *PowerFactory* was designed as an integrated engineering tool to provide a comprehensive suite of power system analysis functions within a single executable program. Key features include:

1. *PowerFactory* core functions: definition, modification and organisation of cases; core numerical routines; output and documentation functions.
2. Integrated interactive single line graphic and data case handling.
3. Power system element and base case database.
4. Integrated calculation functions (e.g. line and machine parameter calculation based on geometrical or nameplate information).
5. Power system network configuration with interactive or on-line SCADA access.
6. Generic interface for computer-based mapping systems.

Use of a single database, with the required data for all equipment within a power system (e.g. line data, generator data, protection data, harmonic data, controller data), means that *PowerFactory* can easily execute all power simulation functions within a single program environment - functions such as load flow analysis, short-circuit calculation, harmonic analysis, protection coordination, stability analysis, and modal analysis.

Although *PowerFactory* includes highly-sophisticated power system analysis functions, the intuitive user interface makes it possible for new users to very quickly perform common tasks such as load flow and short-circuit calculations.

The functionality purchased by a user is configured in a matrix-like format, where the licensed calculation functions, together with the maximum number of buses, are listed as coordinates. The user can then, as required, configure the interface and functions according to their requirements.

Depending on user requirements, a specific *PowerFactory* licence may or may not include all of the functions described in this manual. As requirements dictate, additional functionality can be added to a licence. These functions can be used within the same program interface with the same network data. Only additional data, as may be required by an added calculation function, need be added.

4.1 General Concept

The general *PowerFactory* program design concept is summarised as follows:

Functional Integration

DIgSILENT *PowerFactory* software is implemented as a single executable program, and is fully compatible with Windows 7, Windows 8 and Windows 10. The programming method employed allows for fast selection of different calculation functions. There is no need to reload modules and update or transfer data and results between different program applications. As an example, the Load Flow, Short-Circuit, and Harmonic Load Flow analysis tools can be executed sequentially without resetting the program, enabling additional software modules and engines, or reading and converting external data files.

Vertical Integration

DIgSILENT *PowerFactory* software uses a vertically integrated model concept that allows models to be shared for all analysis functions. Studies relating to “Generation”, “Transmission”, “Distribution”, and “Industrial” analysis can all be completed within *PowerFactory*. Separate software engines are not required to analyse separate aspects of the power system, or to complete different types of analysis, as DIgSILENT *PowerFactory* can accommodate everything within one integrated program and one integrated database.

Database Integration

One Database Concept: DIgSILENT *PowerFactory* provides optimal organisation of data and definitions required to perform various calculations, memorisation of settings or software operation options. The *PowerFactory* database environment fully integrates all data required for defining study cases, Operation Scenarios, single line graphics, textual and graphical Results, calculation options, and user-defined models, etc. Everything required to model and simulate the power system is integrated into a single database which can be configured for single and/or multiple users.

Project Management: all data that defines a power system model is stored in “Project” folders within the database. Inside a “Project” folder, “Study Cases” are used to define different studies of the system considering the complete network, parts of the network, or Variations on its current state. This “project and study case” approach is used to define and manage power system studies with object-oriented software. *PowerFactory* uses a structure that is easy to use, avoids data redundancy, and simplifies the task of data management and validation for users and organisations. The application of study cases and project Variations in *PowerFactory* facilitates efficient and reliable reproduction of study results.

Multi-User Operation: multiple users each holding their own projects or working with data shared from other users are supported by a “Multi-user” database operation. In this case the definition of access rights, user accounting and groups for data sharing are managed by the *PowerFactory* Administrator user.

Offline Mode: in some instances, a network connection to a server database may not be available. To address this, *PowerFactory* provides functionality to work in Offline Mode. The required project data is cached to the user’s local machine, which can then later be synchronised to the server database. Offline Mode functionality includes the ability to lock and unlock projects and to edit projects as read-only.

User Interface Customisation

By default, “Base Package” and “Standard” user profiles are available in *PowerFactory*. Profiles can be selected from the main menu under *Tools* → *Profiles*. The “Base Package” profile limits the icons displayed on the main toolbar to those typically used by new users, such as load flow and short-circuit commands. The database Administrator can create and customise user profiles, in particular:

- Customise the element dialog pages that are displayed.
- Customise element dialog parameters. Parameters can be Hidden (not shown) or Disabled (shown but not editable).
- Fully configure Main Toolbar and Drawing Toolbar menus, including definition of custom DPL Commands and Templates with user-defined icons.
- Customise Main Menu, Data Manager, and context-sensitive menu commands.

Chapter 6: User Accounts, User Groups, and Profiles (Section 6.7 Creating Profiles) details the customisation procedure.

Database, Objects, and Classes

PowerFactory uses a hierarchical, object-oriented database. All the data, which represents power system Elements, single line graphics, study cases, system Operation Scenarios, calculation commands, program Settings etc., are stored as objects inside a hierarchical set of folders. The folders are arranged in order to facilitate the definition of the studies and optimise the use of the tools provided by the program.

The objects are grouped according to the kind of element that they represent. These groups are known as “Classes” within the *PowerFactory* environment. For example, an object that represents a synchronous generator in a power system is of a Class called *ElmSym*, and an object storing the settings for a load flow calculation is of a Class called *ComLdf*. Object Classes are analogous to computer file extensions: each Class has a specific set of parameters that defines the objects it represents. As explained in Section 4.7 (User Interface), the edit dialogs are the interfaces between the user and an object; the parameters defining the object are accessed through this dialog. This means that there is an edit dialog for each class of object.

The main classes that the *PowerFactory* user is likely to come across fall into these categories:

- ***.Elm*** Network elements
- ***.Typ*** Type objects, assigned to network element in order to configure parameters often supplied by manufacturers.
- ***.Int*** Structural objects, for example *.IntFolder
- ***.Set*** These objects generally contain settings
- ***.Com*** Command objects, for example *.ComLdf for the load flow command

---

Note: Everything in *PowerFactory* is an object; an object is defined by its Class and the objects are stored according to a hierarchical arrangement in the database tree.

4.2 *PowerFactory* Simulation Functions

PowerFactory incorporates a comprehensive list of simulation functions, described in detail in part *Power System Analysis Functions* of the manual, including the following:

- Load Flow Analysis, allowing meshed and mixed 1-,2-, and 3-phase AC and/or DC networks (Chapter 25)
- Short-Circuit Analysis, for meshed and mixed 1-,2-, and 3-phase AC networks (Chapter 26)

- Sensitivities / Distribution Factors, for voltage, branch flow and transformer sensitivities (Chapter 48)
- Low Voltage Network Analysis (Section 25.4.2: Advanced Load Options)
- Contingency Analysis (Chapter 27)
- Quasi-Dynamic simulation, which allows to perform several load flow calculations in a period of time (Chapter 28)
- Network Reduction (Chapter 49)
- Protection Analysis (Chapter 33)
- Arc-Flash Hazard Analysis (Chapter 34)
- Cable Analysis, including cable sizing and cable ampacity calculation (Chapter 35)
- Power Quality and Harmonics Analysis (Chapter 36)
- Connection Request Assessment (Section 36.8)
- Transmission Network Tools (Chapter 40), including:
  – PV curves calculation
  – QV curves calculation
  – Power Transfer Distribution Factors
  – Transfer Capacity Analysis
- Distribution Network Tools (Chapter 41), including:
  – Hosting Capacity Analysis
  – Backbone Calculation
  – Voltage Sag
  – Tie Open Point Optimisation
  – Phase Balance Optimisation
  – Voltage Profile Optimisation
  – Optimal Equipment Placement
  – Optimal Capacitor Placement
- Outage Planning: tool for management of planned outages (Chapter 42)
- Probabilistic Analysis (Chapter 44)
- Reliability Analysis Functions:
  – Reliability Assessment (Chapter 45)
  – Optimal Power Restoration (Chapter 46)
  – Optimal Remote Control Switch (RCS) Placement (Section 46.3)
  – Optimal Manual Restoration (Section 46.4)
  – Generation Adequacy Analysis (Chapter 47)
- Optimal Power Flow (Chapter 38)
- Unit Commitment (Chapter 39)
- Economic Analysis Tools (Chapter 43), including:
  – Techno-Economical Calculation (Section 43.2)
  – Power Park Energy Analysis (Section 43.4)
- State Estimation (Chapter 50)
- RMS Simulation: time-domain simulation for electromechanical transients (Chapter 29)

- EMT Simulation: time-domain simulation of electromagnetic transients (Chapter 29)
- Motor Starting Functions (Chapter 51)
- Modal / Eigenvalue Analysis (Chapter 32)
- Model Parameter Identification (Chapter 31)

4.3 General Design of *PowerFactory*

PowerFactory is primarily intended to be used and operated in a graphical environment. That is, data is entered by drawing the network elements, and then editing and assigning data to these objects. Data is accessed from the graphics page by double-clicking on an object. An input dialog is displayed and the user may then edit the data for that object.

Figure 4.3.1 shows the *PowerFactory* Graphical User Interface (GUI) when a project is active. The GUI is discussed in further detail in Section 4.7

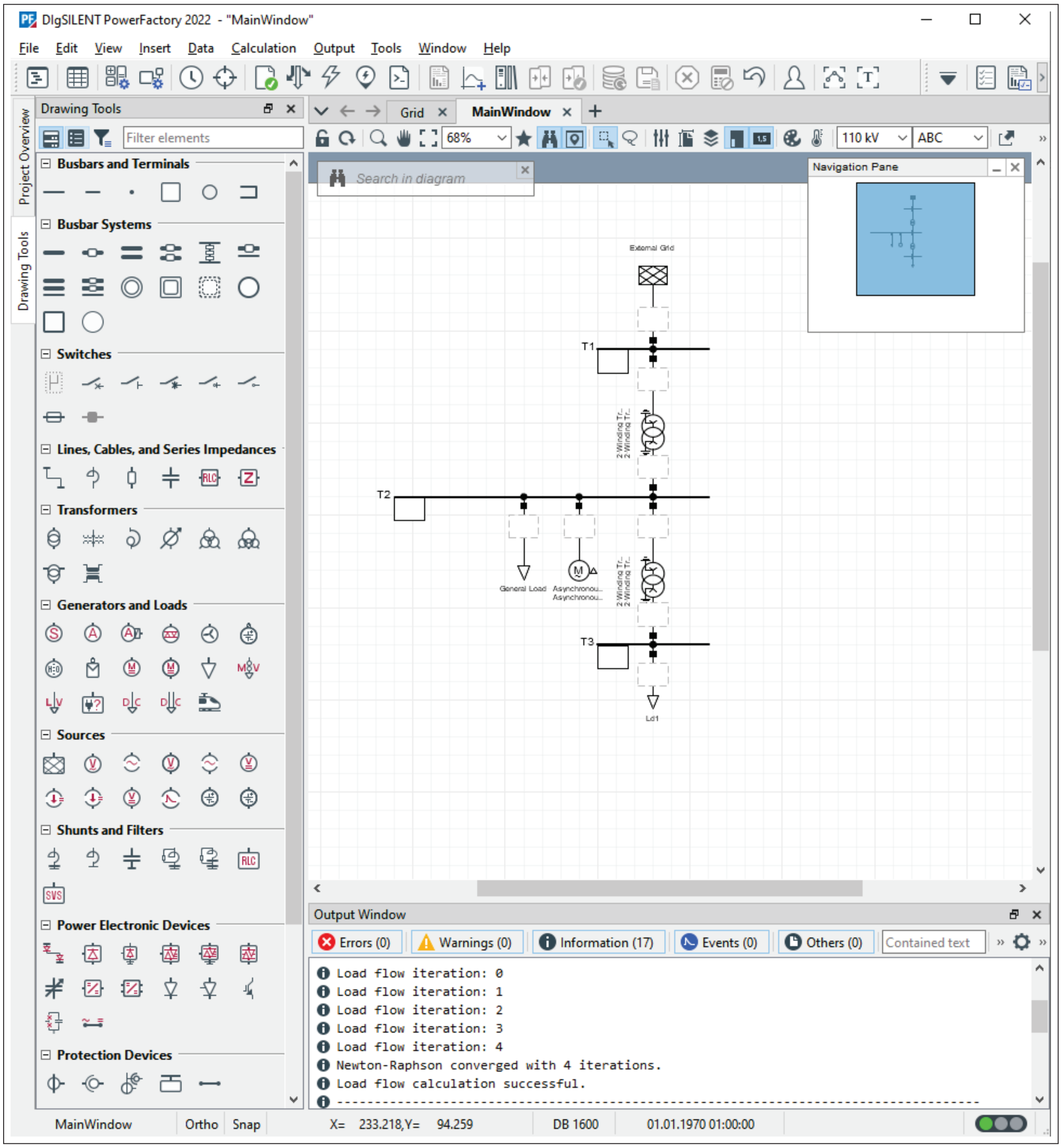


Figure 4.3.1: *PowerFactory* Main Window

All data entered for objects is hierarchically structured in folders for ease of navigation. To view the data and its organisation, a "Data Manager" is used. Figure 4.3.2 shows the Data Manager window. The Data Manager is similar in appearance and functionality to a Windows Explorer window.

Within the Data Manager, information is grouped based on two main criteria:

1. Data that pertains directly to the system under study, that is, electrical data.
2. Study management data, for example, which graphics should be displayed, what options have been chosen for a Load Flow Calculation command, which Areas of the network should be considered for calculation, etc.

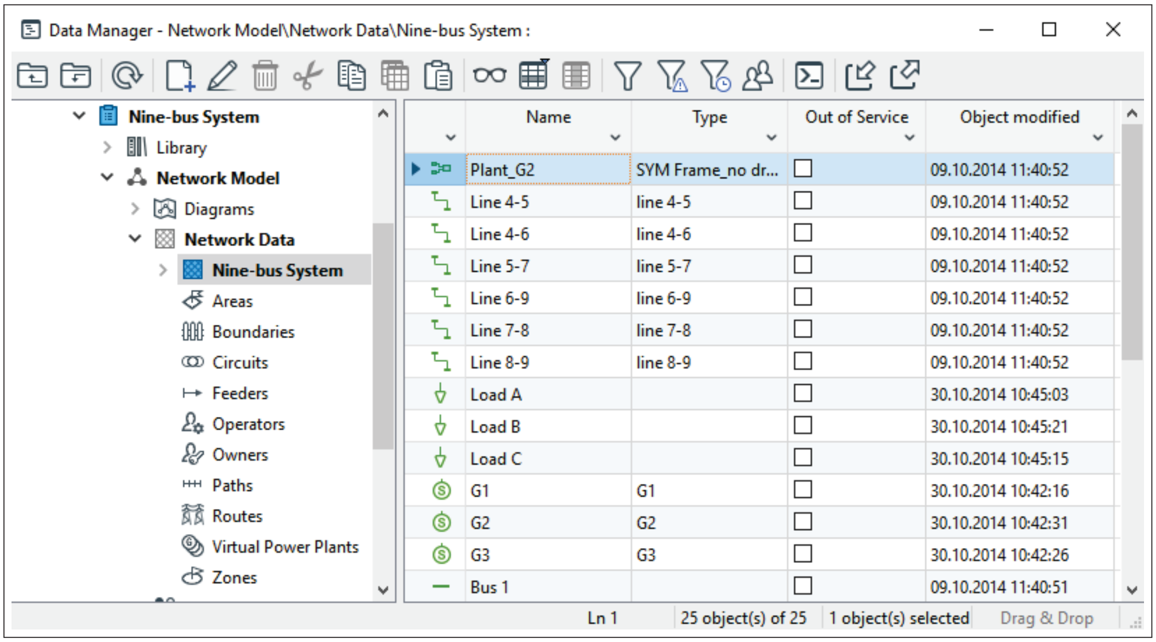


Figure 4.3.2: *PowerFactory* Data Manager

Note that most user-actions can be performed in both the single line graphic and the Data Manager. For example, a new terminal can be added directly to the single line graphic, or alternatively created in the Data Manager. In the latter case, the terminal could be shown in the single line graphic by using the *Diagram Layout Tool*, by "dragging and dropping" from the Data Manager, or by creating a new Graphical Net Object in the Data Manager (advanced).

4.4 Type and Element Data

Since power systems are constructed using standardised materials and components, it is convenient to divide electrical data into two sets, namely "Type" data and "Element" data sets.

- Characteristic electrical parameters, such as the reactance per km of a line, or the rated voltage of a transformer are referred to as Type data. Type objects are generally stored in a global library (either the *DIgSILENT* Library or a Custom Library) or Project Library, and are shown in red. For instance, a Line Type object, *TypLne* ( ).
- Data relating to a particular instance of equipment, such as the length of a line, the derating factor of a cable, the name of a load, the connecting node of a generator, or the tap position of a transformer are referred to as Element data. Element objects are generally stored in the Network Data folder, and are shown in green. For instance, a Line Element object, *ElmLne* ( ).

Consider the following example:

- A cable has a Type reactance of "X" Ohms/km, say 0.1 Ohms/km.
- A cable section of length "L" is used for a particular installation, say 600 m, or 0.6 km.
- This section (Element) therefore has an reactance of X * L Ohms, or 0.06 Ohms.

Note that Element parameters can be modified using Operation Scenarios (which store sets of network operational data), and Parameter Characteristics (which can be used to modify parameters based on the study case Time, or other user-defined trigger).

4.5 Data Arrangement

The *PowerFactory* database supports multiple users (as mentioned in 4.1) and each user can manage multiple projects. “**User Account**” folders with access privileges only for their owners (and other users with shared rights) must then be used. User accounts are of course in a higher level than projects.

Figure 4.5.1 shows a snapshot from a database as seen by the user in a Data Manager window, where there is a user account for “User”, and one project titled “Project”. The main folders used to arrange data in *PowerFactory* are summarised below:

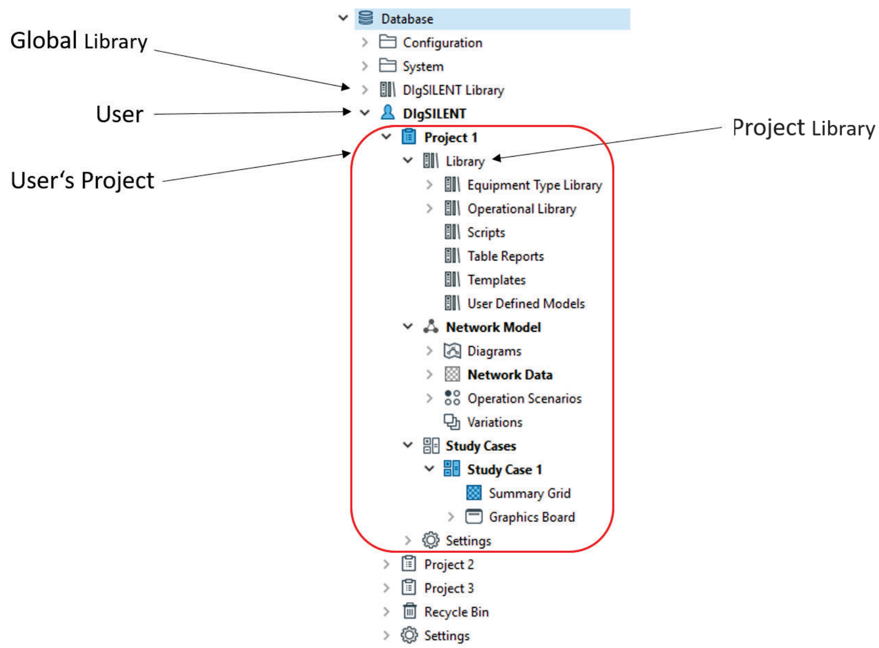


Figure 4.5.1: Structure of a *PowerFactory* project in the Data Manager

4.5.1 *DIgSILENT* Global Library

A global library (called “*DIgSILENT* Library”) is supplied with *PowerFactory* and should not be edited by the customer. When a database is migrated to a new version, the *DIgSILENT* Library will be refreshed with all the new and updated information and if a user were to have made changes and additions, all that information would be lost. If the user wants to include user-specific models etc, a custom global library can be created (see section 4.5.2)

The *DIgSILENT* Library contains a wide range of pre-defined models, including:

- Type data for standard components such as conductors, motors, generators, and transformers.
- Standard control system frames, models, and macros (i.e. transfer functions and logic blocks, etc).
- Standard protection devices models.
- Pre-defined model templates, including:
  - Battery Systems
  - Distributed Energy Resources
  - FACTS
  - Grid-forming Converters

  - HVDC
  - Loads
  - Photovoltaic
  - Plant Controllers
  - Steam/Gas/Diesel Power Plants
  - Variable Speed Drives
  - Wind Turbines (including DFIG, Fully Rated, IEC and WECC)

- Standard DPL scripts.

Documentation about the various elements and types in the *DIgSILENT* Library can be found directly in the description page of the elements, in the Appendix B: Standard Models in *PowerFactory* and in Technical References Document (Templates).

4.5.1.1 Versioning in the Global Library

All the main objects (types, models etc.) in the *DIgSILENT* Library are managed using versions, for example a type object might be at v002.1, where the main version number (002 in this example) is incremented when a material change to the object is made, and the minor version number (.1 in this example) is incremented when a small change that will not affect calculation results is made.

A version page shows the version number and a *Change Log* to show the modification history.

The distinction between the two types of change is important because when the *DIgSILENT* global library is updated due to the installation of a new version of *PowerFactory* by the user, all minor version changes will be implemented automatically, whereas main version updates will not be done automatically. Instead, users can decide whether to use the updated models, or continue using the older versions, which will be automatically retained in subfolders.

4.5.2 Custom Global Library

When several users work for the same company, there is often a need to share a custom library between these users. Instead of copying the library from user to user, a better approach is to create an additional library in the global area which will be visible for all the users.

The custom library should be defined by the Administrator as follows:

- Right-click on the “Database” folder
- Select *New* → *Library* from the context menu
- Chose a library name and click **OK**

The new library is now created at the same level of the hierarchy as the *DIgSILENT* global library.

The user groups with access to the library, and their level of access, are defined in the *Sharing* page of the edit dialog of the library.

Several libraries can be created, but only one can be edited by a user at a time. In order to edit the library, it has to be active; the library is activated using *Right click* → *Activate*.

Once an additional global library is added, it should be set in the User Settings as *Used Library* (see chapter User Settings, Section 7.1).

The versioning of the Custom Global Library works in exactly the same way as for the *DIgSILENT* Library, described in Section 4.5.1.1

4.5.3 Project Library

The Project Library contains the equipment types, network operational information, scripts, templates, and user-defined models (generally) only used within a particular project. A particular project may have references to the project library and / or global library. The Project Library folder and sub-folders are discussed in detail in Chapter 14 (Project Library).

4.5.4 Diagrams

Single line graphics are defined in *PowerFactory* by means of graphic folders of class *IntGrfNet* ( ). Each diagram corresponds to a IntGrfNet folder. They are stored in the *Diagrams* folder ( ) of the Network Model. Single line diagrams are composed of graphical objects, which represent components of the networks under study. Graphical components reference network components and symbol objects (*IntSym*).

The relation between graphical objects and network components allows the definition and modification of the studied networks directly from the single line graphics. Network components can be represented by more than one graphical object (many *IntGrf* objects can refer to the same network component). **Therefore, one component can appear in several diagrams.**

These diagrams are managed by the active study case, and specifically by an object called the *Graphics Board*. If a reference to a network diagram is stored in a study case's Graphics Board, when the study case is activated, the diagram is automatically opened. Diagrams can be easily added and deleted from the Graphics Boards.

Each diagram is related to a specific Grid (*ElmNet*). When a grid is added to an active study case, the user is asked to select (among the diagrams pointing to that grid) the diagrams to display. References to the selected diagrams are then automatically created in the corresponding Graphics Board.

Chapter 9 (Network Graphics), explains how to define and work with single line graphics.

4.5.5 Network Data

The Network Data folder holds network data (Element data) in "Grid" folders and object Grouping information.

Grids

In *PowerFactory*, electrical network information is stored in "Grid" folders (*ElmNet*,). A power system may have as many grids as defined by the user. These grids may or may not be interconnected. As long as they are active, they are considered by the calculations. Data may be sorted according to logical, organisational and/or geographical areas (discussed further in Section 4.6: Project Structure).

Note: A Grid (and in general any object comprising the data model) is active when it is referred to by the current study case. Only objects referred in the current (active) study case are considered for calculation. In the Data Manager, the icon of an active Grid is shown in blue, to distinguish it from inactive Grids.

For details of how to define grids refer to Chapter 8.Basic Project Definition, Section 8.2 (Creating New Grids).

Grouping Objects

In addition to Grid folders, the Network Data folder contains a set of objects that allow further grouping of network components. By default, when a new project is created, new empty folders to store these grouping objects is created inside the Network Model folder.

For details of how to define grouping objects, refer to Chapter 15: Grouping Objects.

4.5.6 Variations

During the planning and assessment of a power system, it is often necessary to analyse different variations and expansion alternatives of the base network. In *PowerFactory* these variations are modelled by means of “Variations”. These are objects that store and implement required changes to a network, and can be easily activated and deactivated. The use of Variations allows the user to conduct studies under different network configurations in an organised and simple way.

Variation objects (*IntScheme*,) are stored inside the Variations folder () which resides in the Network Model folder. Variations are composed of “Expansion Stages” (*IntStage*), which store the changes made to the original network(s). The application of these changes depends on the current study time and the activation time of the Expansion Stages.

The study time is a parameter of the active study case, and is used to situate the current study within a time frame. The activation time is a parameter given to the Expansion Stages, to determine whether or not, according to the study time, the changes contained within the Expansion Stages are applied to the network. If the activation time precedes the study time, the changes are applied to the original network. The changes of a subsequent expansion stage add to the changes of its predecessors.

In order that changes to the network configuration are applied and can be viewed, a Variation must be activated. These changes are contained in the expansion stage(s) of this active Variation. Once the Variation is deactivated, the network returns to its original state. Changes contained in an Expansion Stage can be classified as:

- Modifications to network components.
- Components added to the network.
- Components deleted from the network.

Note: If there is no active Operation Scenario, modifications to operational data will be stored in the active Variation.

4.5.7 Operation Scenarios

Operation Scenarios may be used to store operational settings, a subset of Element data. Operational data includes data that relates to the operational point of a device but not to the device itself e.g. the tap position of a transformer or the active power dispatch of a generator. Operation Scenarios are stored in the Operation Scenarios folder.

4.5.8 Study Cases

The Study Cases folder holds study management information. Study cases are used to store information such as command settings, active Variations and Operations Scenarios, graphics to be displayed, and study results. See Chapter 13 (Study Cases) for details.

4.5.9 Settings

Project settings such as user-defined diagram styles for example, which differ from global settings, are stored inside the Settings folder. See section 8.1.2 (Project Settings)

4.6 Project Structure

Most of the data referred to in the previous section 4.5 is project data. The default high-level arrangement of this data in the project is as shown here:

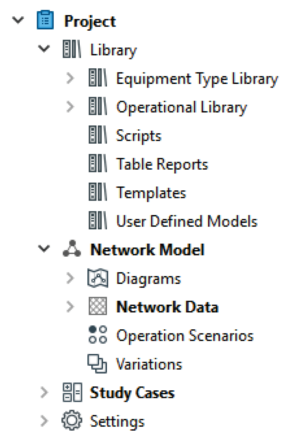


Figure 4.6.1: Basic Project Hierarchy

The structure of project data depends on the complexity of the network, use of the model, and user preferences. The user has the flexibility to define network components directly within the Grid, or to organise and group components in a way that simplifies management of project data.

Consider the example network data arrangement shown in Figure 4.6.2 In this case, two busbar systems (*ElmSubstat* in *PowerFactory*) have been defined, one at 132 kV, and one at 66 kV. The two busbar systems are grouped within a Site, which includes the 132 kV / 66 kV transformers (not shown in Figure 4.6.2). A Branch composed of two line sections and a node connects “132 kV Busbar” to “HV terminal”. Grouping of components in this way simplifies the arrangement of data within the Data Manager, facilitates the drawing overview diagrams, and facilitates storing of Substation switching configurations.

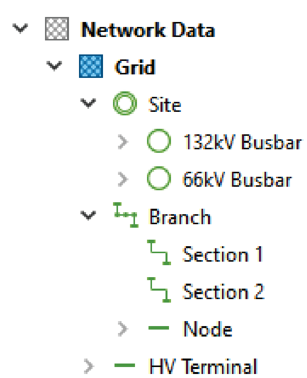


Figure 4.6.2: Example Project Structure

The following subsections provide further information regarding the *PowerFactory* representation of key network topological components.

4.6.1 Nodes

In *PowerFactory*, nodes connecting lines, generators, loads, etc. to the network are generally called "Terminals" (*ElmTerm*). Depending on their usage within the power system, Terminals can be used to represent Busbars, Junctions, or Internal Nodes (their usage is defined by a drop down menu found in the Basic Data page of the terminal dialog). According to the selected usage, different calculation functions are enabled; for example the short-circuit calculation can be performed only for busbars, or for busbars and internal nodes, and so on.

4.6.2 Edge elements

The term "Edge Element" refers to an element connected to a node or to more than one node. Includes single-port elements such as loads, and multi-port elements such as transformers.

4.6.3 Branches

Elements with multiple connections are referred to "branches" (as distinct from a *Branch Element* (*ElmBranch*), which is a grouping of elements, discussed in Section 4.6.9). Branches include two-connection elements such as transmission lines and transformers, and three-connection elements such as three-winding transformers, AC/DC converters with two DC terminals, etc.

For information about how to define transmission lines (and cables) and sections refer to Chapter 11: Building Networks. Technical information about transmission line and cable models is provided in Technical References Document (Line (*ElmLne*)).

4.6.4 Cubicles

When any edge element is directly connected to a Terminal, *PowerFactory* uses a "Cubicle" (*StaCubic*) to define the connection. Cubicles can be visualised as the panels on a switchgear board, or bays in a high voltage yard, to which the branch elements are connected. A Cubicle is generally created automatically when an element is connected to a node (note that Cubicles are not shown on the single line graphic).

4.6.5 Switches

To model complex busbar-substation configurations, switches (*ElmCoup*) can be used. Their usage can be set to Circuit-Breaker, Disconnector, Switch Disconnector, or Load Switch. The connection of an *ElmCoup* to a Terminal is carried out by means of an automatically generated Cubicle without any additional switch (*StaSwitch*) object.

4.6.6 Substations

Detailed busbar configurations are represented in *PowerFactory* as Substations (*ElmSubstat*). Separate single line diagrams of individual substations can be created. Substation objects allow the use of running arrangements to store/set station circuit breaker statuses (see Chapter 14: Project Library, Section 14.3: Operational Library).

For information about how to define substations refer to Chapter 11: Building Networks.

4.6.7 Secondary Substations

Secondary Substations (*ElmTrfstat*) are smaller, simpler substations, typically used for single-transformer connections.

4.6.8 Sites

Network components including Substations and Branches can be grouped together within a “Site” (*ElmSite*). This may include Elements such as substations / busbars at different voltage levels. For information about how to define sites refer to Chapter 11: Building Networks.

4.6.9 Branch Elements

Similar to Substations, Terminal Elements and Line Elements can be stored within an object called a Branch Element (*ElmBranch*). Branches are “composite” two-port elements that may be connected to a Terminal at each end. They may contain multiple Terminals, Line sections (possible including various line types), and Loads etc, but be represented as a single Branch on the single line graphic. As for Substations, separate diagrams for the detailed branch can be created with the graphical editor.

For information about how to define branches refer to Chapter 11: Building Networks, sections 11.2 and 11.5.

4.7 User Interface

An overview of the *PowerFactory* user interface is provided in this section, including general discussion of the functionality available to enter and manipulate data and graphics.

4.7.1 Overview

4.7.1.1 General handling

The *PowerFactory* user interface consists of a number of tool windows and tool bars, which can be freely moved around or docked, as the user chooses, together with the graphics board.

Tool windows and tool bars

The tool windows are:

- **Project Overview**
- **Drawing Tools**
- **Output Window**

By default, these are “docked” into position, but can be resized or moved around as required. A window is moved by clicking on the top portion of the window and dragging to the required position. This can simply be an “undocked” location, but if the tool window approaches a position where it can be docked, this will be indicated by a blue background as shown here:

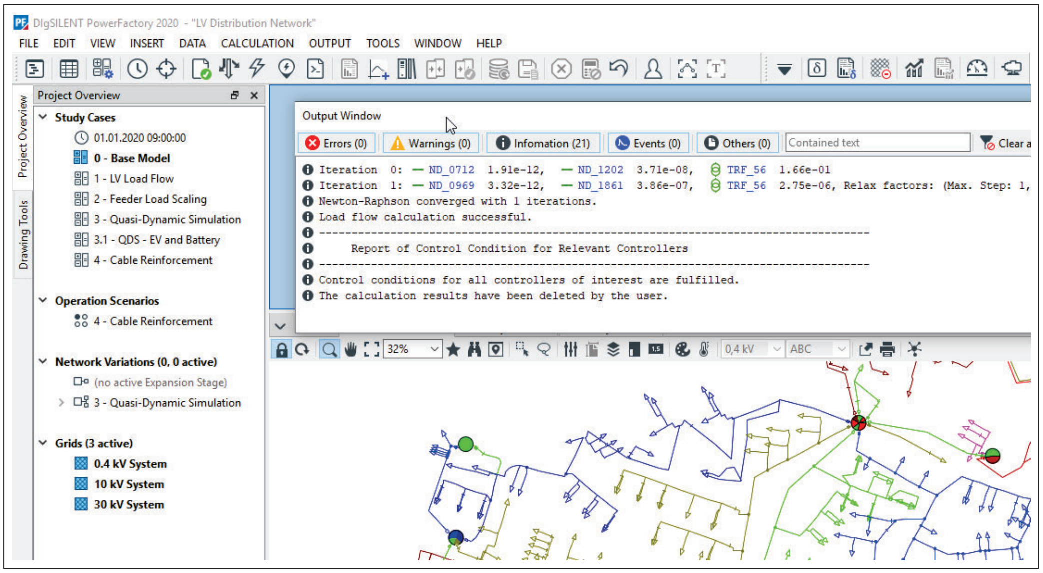


Figure 4.7.1: Docking a tool window

Toolbars can also be moved around. The icons are divided into sets separated by a barrier , and a set of icons can be moved by clicking on the barrier and dragging it to the required location.

A useful option to be aware of is *Reset windows layout*, which is found in the main Window menu. This will restore the various tool windows to their default positions. Note that this does not affect the graphic board layout, split etc., as this is a study-case specific configuration.

Graphics board

The graphics board, sometimes called the graphical editor, uses a tabbed view concept, enabling a number of graphics to be open at any one time. The currently-selected graphic is often referred to as the active graphic.

If the user wishes to see more than one graphic at the same time, split-screen working can be used. The screen can be split either vertically or horizontally at any one time, and graphical pages moved between the splits (or new splits created) by right-clicking on the graphic tab and choosing the relevant option. The options available are listed in the Network Graphics chapter, in section 9.2.5.

4.7.1.2 Key Features of the user interface

The main *PowerFactory* window is shown in Figure 4.7.2.

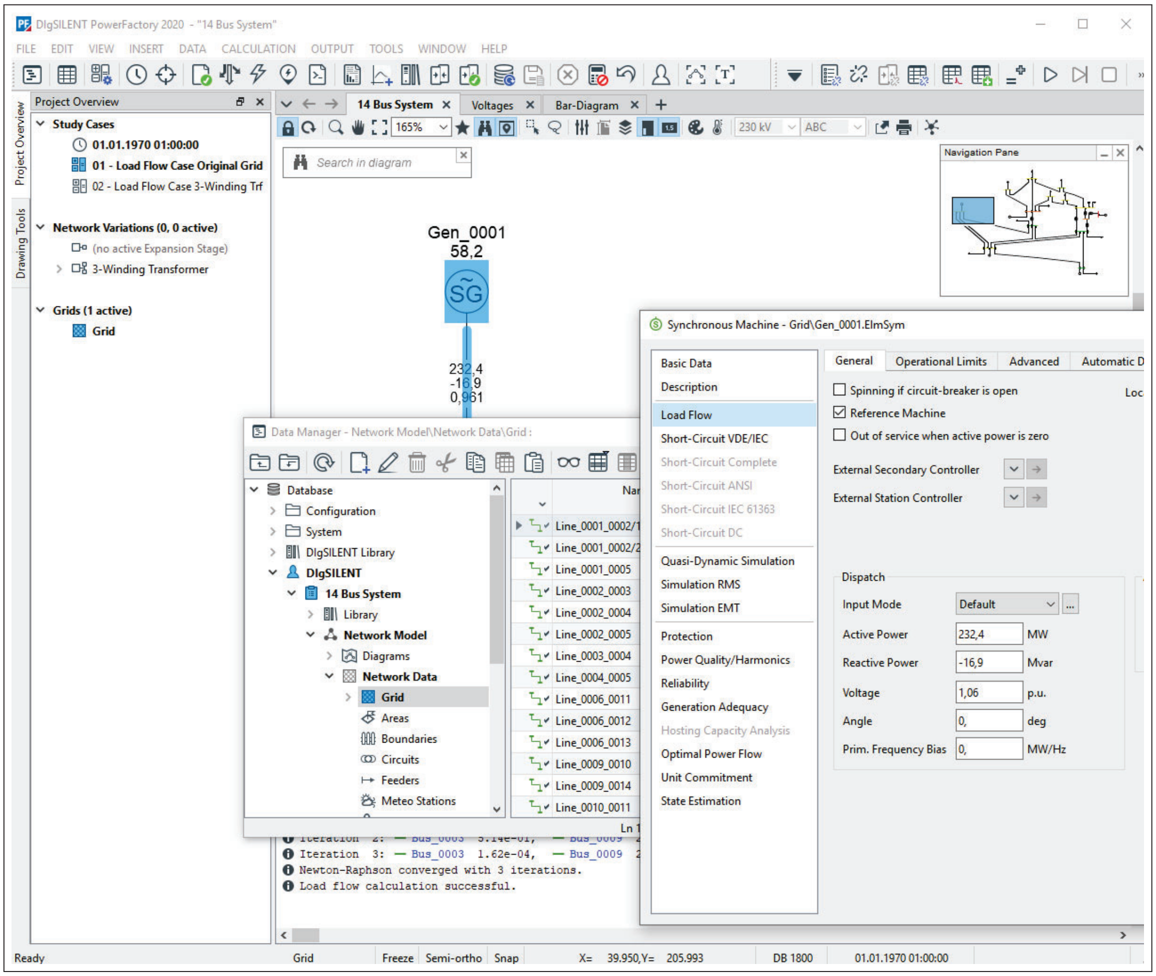


Figure 4.7.2: *PowerFactory* user interface

Key features of the main window are as follows:

1. The main window includes a description of the *PowerFactory* version, and standard icons to Minimise, Maximise/Restore, Resize, and Close the window.

2. The main menu bar includes drop-down menu selections. The main menu is discussed further in Section 4.7.2 (Menu Bar).

3. The Main Toolbar includes commands and other icons. The Main Toolbar is discussed in further detail in Section 4.7.3 (Main Toolbar).

4. The Graphical Editor displays single line diagrams, block diagrams and/or simulation plots of the active project. Studied networks and simulation models can be directly modified from the graphical editor by placing and connecting elements.

5. When an object is right clicked (in the graphical editor or in the Data Manager) a *context* sensitive menu with several possible actions appears.

6. When an object is double clicked its edit dialog will be displayed. The edit dialog is the interface between an object and the user. The parameters defining the object are accessed through this edit dialog. Normally an edit dialog is composed of several “pages”. Each page groups parameters that are relevant to a certain function. In Figure 4.7.2 the Load Flow page of a generator is shown, where only generator parameters relevant to load flow calculations are shown.

7. The Data Manager is the direct interface with the database. It is similar in appearance and functionality to a Windows Explorer window. The left pane displays a symbolic tree representation of the complete database. The right pane is a data browser that shows the content of the currently selected folder. The Data Manager can be accessed by pressing the *Data Manager* icon () on the left of the main toolbar. It is always 'floating', and more than one can be active at a time. Depending on how the user navigates to the Database Manager, it may only show the database tree for selecting a database folder, or it may show the full database tree. The primary functionality of the Data Manager is to provide access to power system components/objects. The Data Manager can be used to edit a group of selected objects within the Data Manager in tabular format. Alternatively, objects may be individually edited by double clicking on an object (or *right-click* → *Edit*).

8. The output window is shown at the bottom of the *PowerFactory* window. The output window is discussed in further detail in Section 4.7.4 (The Output Window).

9. The Project Overview window is displayed by default on the left side of the main application window between the main toolbar and the output window. It displays an overview of the project allowing the user to assess the state of the project at a glance and facilitating easy interaction with the project data.

10. The Drawing Tools window is by default also on the left-hand side of the user interface. In Figure4.7.2 it is behind the Project Overview, but it can be selected via the tab, and will automatically come to the front if "freeze mode" is removed (see section 9.2.7 for more information.)

11. *PowerFactory* has a very flexible window moving and docking concept that aims to optimise work when using more than one monitor, allowing the user to:
    - create as many groups of tabs rightward or downward in the Main Window as required and freely move the tabs between these groups
    - create a new floating window by moving a tab from the Main Window or from an already existing floating window
    - move tabs from a floating window to the Main Window (docking) or to another already existing floating window

    Moving and docking options apply to both tabs displaying single-line diagrams and those containing plots. Even tabular reports can be moved or docked freely as described above. The different moving and docking options can be accessed by right-clicking on the tab. Another method, perhaps more intuitive, is to simply drag the tab to the centre of the screen to create a new floating window or to an existing window to add the tab to it.

12. *PowerFactory* dynamically adapts to any screen resolution and DPI scaling setting. The application perfectly respects DPI scaling when moving windows between two screens of different resolution settings, e.g. when working with a laptop and an additional monitor, or two monitors of different resolution settings.

4.7.2 Menu Bar

The menu bar contains the main *PowerFactory* menus. Each menu entry has a drop down list of menu options and each menu option performs a specific action. To open a drop down list, either click on the menu entry with the left mouse button, or press the **Alt** key together with the underlined letter in the menu. Menu options that are shown in grey are not available, and only become available as the user activates projects or calculation modes, as required.

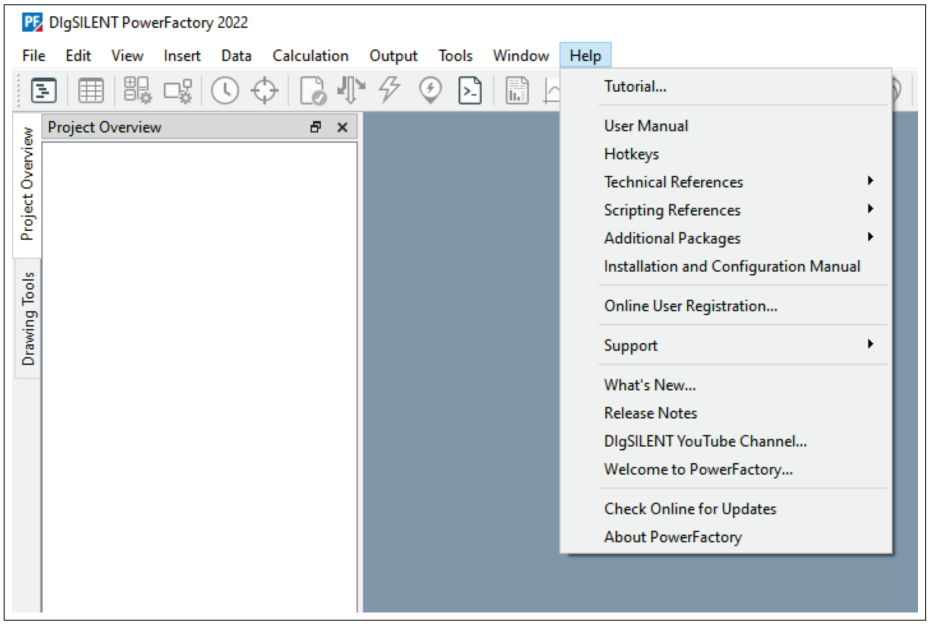


Figure 4.7.3: The help menu on the Menu bar

For example as show in Figure 4.7.3:

- To access *PowerFactory* tutorial: Press **Alt-H** to open the help menu. Use the keyboard to select Tutorial. . . .
- To access the User Manual: Left click the *Help* menu. Left-click the option *User Manual* to open the electronic User Manual.

4.7.3 Main Toolbar

The main *PowerFactory* toolbar provides the user with quick access to the main commands available in the program (see Figure 4.7.2). Buttons that appear in grey are only active when appropriate. All command icons are equipped with balloon help text which are displayed when the cursor is held still over the icon for a moment, and no key is pressed.

Note that the visibility of buttons to an individual user may depend on the User Profile. See section 6.7 for more details about this.

To use a command icon, click on it with the left mouse button. Those icons that perform a task will automatically return to a non-depressed state when that task is finished. Some command icons will remain depressed, such as the button to *Maximise Output Window*. When pressed again, the button will return to the original (non-depressed) state.

This section provides a brief explanation of the purpose of the icons found on the upper part of the toolbar. Icons from the lower part of the toolbar are discussed in Chapter 9 (Network Graphics (Single Line Diagrams)). Detailed explanations for each of the functions that the icons command are provided in the other sections of the manual.

 **Open Data Manager**

Opens a new instance of the Data Manager. When the option "Use multiple Data Manager" is enabled in the user settings menu (*User Settings* → *Data/Network Model Manager*) the user will be able to open as many instances of the Data Manager as required. If "Use multiple Data Manager" is disabled, the first instance of the Data Manager will be re-opened. For more information on the Data Manager refer to Chapter 10.

Open Network Model Manager

Opens the Network Model Manager, which is a browser for all calculation relevant objects. It provides a list of all elements (coloured in green) and types (coloured in red) that are in an active Grid: e.g. transformer types, line elements, composite models, etc. For more information, see Chapter 12: Network Model Manager.

Study Case Manager

Deactivates the currently-active study case and opens the Study Case Manager window. See section 13.2.1 for more details.

Date/Time of Calculation Case

Displays the date and time for the case calculation. This option is used when parameter characteristics of specific elements (e.g. active and reactive power of loads) are set to change according to the study time, or a Variation status is set to change with the study time.

Edit Trigger

Displays a list of all Triggers that are in the active study case. These Triggers can be edited in order to change the values for which one or more characteristics are defined. These values will be modified with reference to the new Trigger value. All Triggers for all relevant characteristics are automatically listed. If required, new Triggers will be created in the study case. For more information, see Chapter 18: Parameter Characteristics, Load States, and Tariffs. Section 18.2.

Network Data Assessment

Activates the Network Data Assessment command dialog to generate selected reports on network data or to perform model data verification. For more information see Section 26.6: Capacitive Earth-Fault Current or Section 25.6: Troubleshooting Load Flow Calculation Problems.

Calculate Load Flow

Activates the Load Flow Calculation command dialog. For more information about the specific settings, refer to Chapter 25: Load Flow Analysis.

Calculate Short-Circuit

Activates the short-circuit calculation command dialog. For more information, refer to Chapter 26: Short-Circuit Analysis.

Edit Short-Circuits

Edits Short-Circuit events. Events are used when a calculation requires more than one action or considers more than one object for the calculation. Multiple fault analysis is an example of this. If, for instance, the user multi-selects two busbars (using the cursor) and then clicks the right mouse button *Calculate* → *Multiple Faults* a Short-circuit event list will be created with these two busbars in it.

Execute Scripts

Displays a list of scripts that are available. See Section 4.8.1 for a general description of DPL scripts, and Chapter 23: Scripting for detailed information.

Output Calculation Analysis

Presents calculation results in various formats. The output is printed to the output window and can be viewed, or copied for use in external reports. Several different reports, depending on the calculation, can be created. For more information about the output of results refer to Chapter 19:Reporting and Visualising Results, Section 19.4.2.

Insert Plot

Opens the Insert Plot dialog, where different types of plot can be selected. For more information refer to Chapter 19:Reporting and Visualising Results, Section 19.7.

Documentation of Device Data

Presents a listing of device data (a device is the model of any physical object that has been entered into the project for study). This output may be used in reports, and for checking data that has been entered. Depending on the element chosen for the report, the user has two options; generate a short listing, or a detailed report. For more information refer to Chapter 19: Reporting and Visualising Results, Section 19.4.1.

Comparing of Results On/Off

Turns on/off comparing of calculation results. Used to compare results where certain settings or designs options of a power system have been changed from one calculation to the next. For more information refer to Chapter 19: Reporting and Visualising Results, Section 19.5.

Edit Comparing of Results

Enables the user to select the cases/ calculation results that are to be compared to one another, or to set the colouring mode for the difference reporting. For more information refer to Chapter 19:Reporting and Visualising Results, Section 19.5.

Update Database

Utilises the current calculations results (i.e. the calculation 'output' data) to change input parameters (i.e. data the user has entered). An example is the transformer tap positions, where these have been calculated by the Load Flow command option "Automatic Tap Adjust of Tap Changers." For more information refer to Chapter 25: Load Flow Analysis, Section 25.5.7.

Save Operation Scenario

Saves the current operational data to an Operation Scenario (e.g. load values, switch statuses, etc.). See Chapter 16: Operation Scenarios.

Break

Stops a transient simulation or DPL script that is running.

Reset Calculation

Resets any calculation performed previously. This icon is only enabled after a calculation has been carried out.

Note: If *Retention of results after network change* is set to *Show last results* (User Settings, *Miscellaneous* page), results will appear in grey on the single line diagram and on the Flexible Data tab until the calculation is reset, or a new calculation performed.

Undo

This is used to undo the last action which caused information to be written to the database.

User Settings

User options for many global features of *PowerFactory* may be set from the dialog accessed by this icon. For more information refer to Chapter 7: User Settings.

Maximise Graphic Window

Maximises the graphic window. Pressing this icon again will return the graphic window to its original state.

Maximise Output Window

Maximises the output window. Pressing this icon again will return the output window to its original state.

Change Toolbox

In order to minimise the number of icons displayed on the taskbar, some icons are grouped based on the type of analysis, and are only displayed when the relevant category is selected from the *Change Toolbox* icon. In Figure 4.7.4, the user has selected *Simulation RMS/EMT*, and therefore only icons relevant for RMS and EMT studies are displayed to the right of the *Change Toolbox* icon. If, for example, *Reliability Analysis* were selected then icons to the right of the *Change Toolbox* icon would change to those suitable for a reliability analysis.

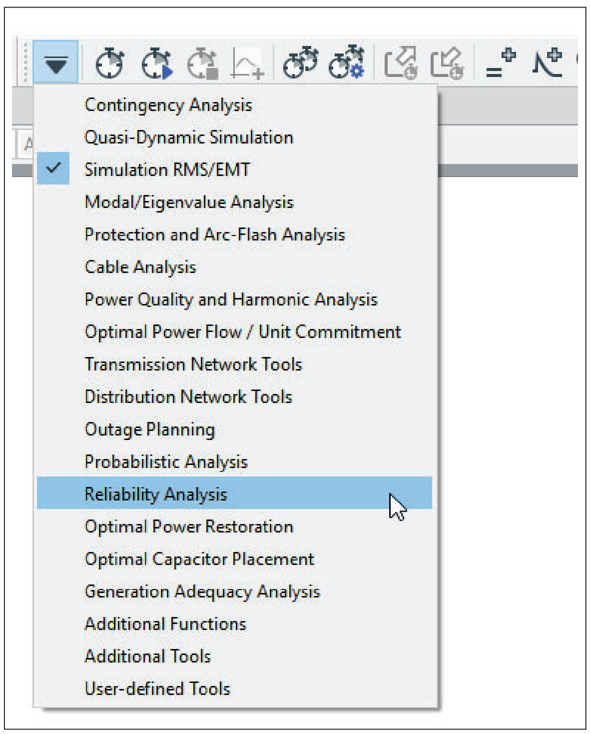


Figure 4.7.4: Change Toolbox selection

4.7.4 The Output Window

In addition to results presented in the single line graphics and / or Data Manager, the output window displays other textual output, such as error messages, warnings, command messages, device documentation, results of calculations, and generated reports, etc. This section describes output window use and functionality.

4.7.4.1 Sizing and Positioning the Output Window

The default location of the output window is "docked" (fixed) at the bottom of the main window, but like the other tool windows it can be moved around or resized.

Two additional options for the output window are:

- Pressing the *Maximise Graphic Window* icon ( ) on the main toolbar to enlarge the graphics board by hiding the output window.
- Pressing the *Maximise Output Window* icon ( ) icons on the main toolbar to enlarge the output window.

Moreover, when the output window is "undocked", it is possible for the user to maximise, minimise or close it using the standard icons.

4.7.4.2 Output Window Options

The contents of the output window may be stored, edited, printed, etc., using the icons shown on the right-hand pane of the output window. Some commands are also available from the context sensitive menu by right-clicking the mouse in the output window pane.

Opens the User Settings on the *Output Window* page.

Saves the selected text, or the complete contents of the output window if no selection was made, to a text, html or csv file.

Copies the selected text to the Windows Clipboard. Text may then be pasted in other programs.

The contents of the output window are displayed and immediately saved in a file.

Redirects the output window to be printed directly.

Searches the text in the output window for the occurrences of a given text.

Clears the output window by deleting all messages. Note that when the user scrolls back and clicks on previous messages in the output window, the output window will no longer automatically scroll with new output messages. The *Clear All* icon will "reset" scrolling of the output window. Ctrl\End can also be used to "reset" scrolling.

4.7.4.3 Using the Output Window

The output window facilitates preparation of data for calculations, and identification of network data errors. Objects which appear blue in the output window generally have a hyperlink so that they can be clicked with the left mouse button to open an edit dialog for the object. Alternatively, the object can be right-clicked and then *Edit, Edit and Browse Object*, or *Mark in Graphic* selected. This simplifies the task of locating objects in the single line graphic.

Additionally, options to jump between message types are available when selecting the option *Go to* → *Next/previous message*

4.7.4.4 Output Window Filters

In the output window, shown in figure 4.7.5, the messages are not only coloured, but icons are also used to indicate the category (error, warning, info, events,...); these categories can be filtered using the predefined filtering tabs. There is also a text filter, to find specific text strings in the output messages.

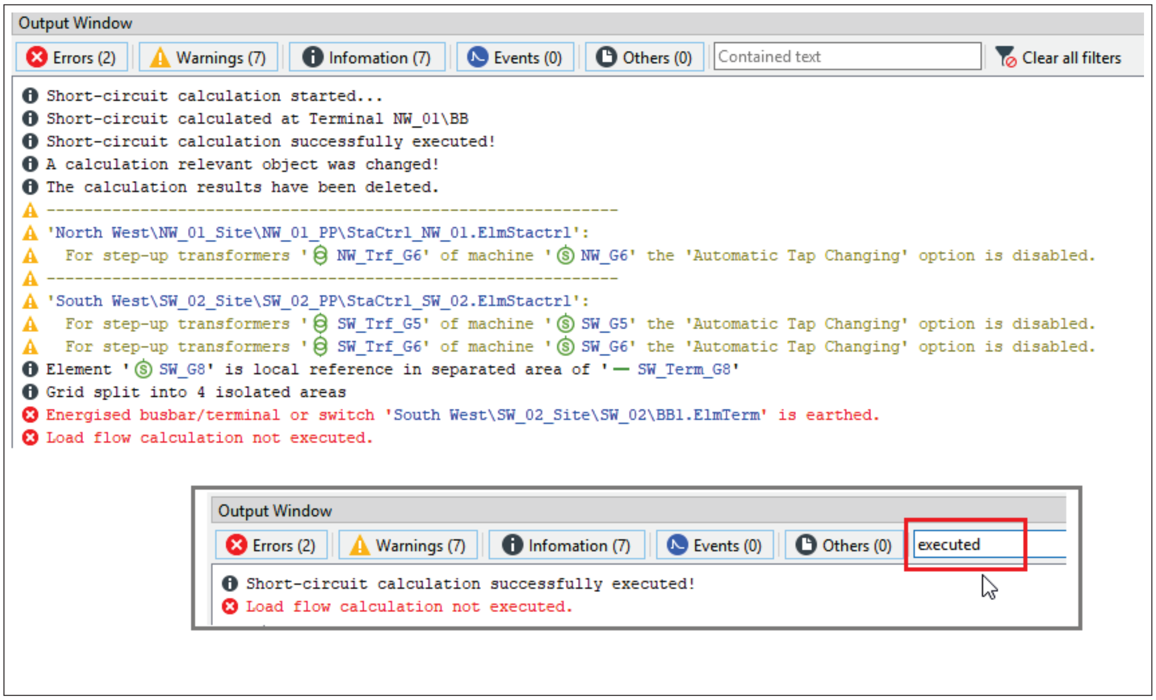


Figure 4.7.5: Output window

The messages in the output window are classified as following:

Error message: red.

Warning message: dark yellow.

Information messages: black.

Events messages: blue.

Output text: black.

The button *Clear all filters* can be used to remove all the selected filters.

4.7.4.5 Output Window Graphical Results

Reports of calculation results may contain bar graphical information. The “voltage profiles” report after a load flow calculation, for instance, produces bar graphs of the per-unit voltages of busbars. These bars will be coloured blue, green or red if the option *Show Verification Report* in the Load Flow Calculation command has been enabled. They will be cross-hatched if the bars are too large to display.

Part of a bar graph output is shown in Figure 4.7.6 The following formatting is visible:

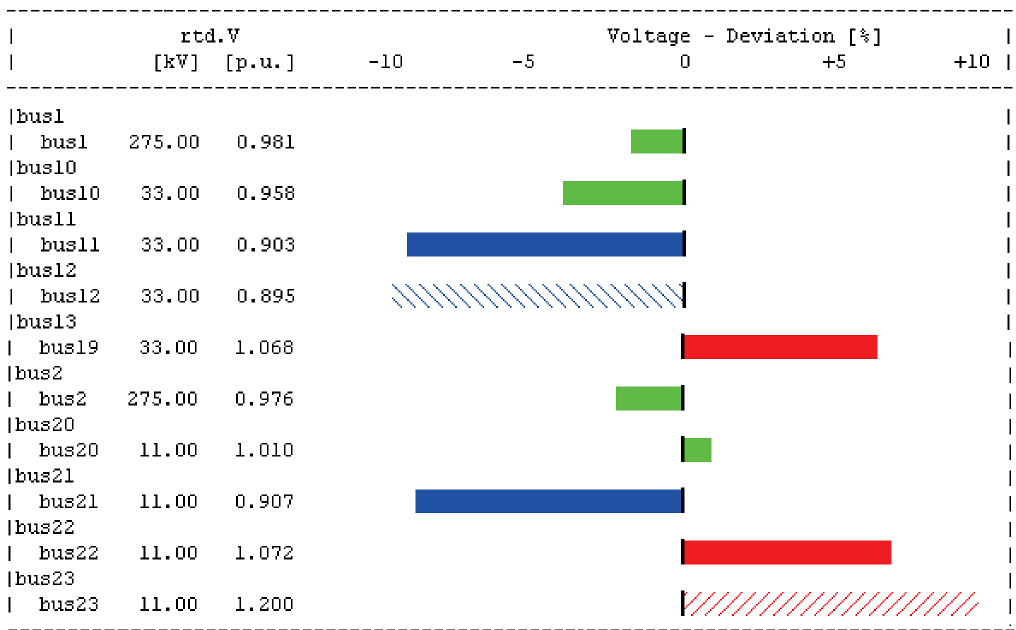


Figure 4.7.6: Output window bar diagram

- Green Solid Bar: Used when the value is in the tolerated range.
- Blue Solid Bar: Used when the value is below a limit.
- Red Solid Bar: Used when the value is above a limit.
- Cross-hatched Bar: Used when the value is outside the range.

4.7.4.6 Copying from the Output Window

The contents of the output window, or parts of its contents, may be copied to the built-in editor of *PowerFactory*, or to other programs. The lines that are to be copied are determined by the output window settings; by default what is shown in the output window is copied. The filters can be used to show only the messages of interest.

4.7.5 Use of Colouring in *PowerFactory*

Colouring is used widely in *PowerFactory* for the presentation of data and visualisation of results. Generally speaking, the user can either configure the colour scheme, or simply let *PowerFactory* apply default settings. In this section, the general approach and options are described.

It should be noted that in some contexts, in particular the colouring of curves in plots, colour palettes are offered. In other contexts, the user simply configures the chosen colour directly.

4.7.5.1 Colour palettes

A number of colour palettes are available. The use of palettes is particularly useful when configuring colours for plots, because by selecting a palette, the user automatically assigns a harmonious set of colours to the curves on the plot, according to the desired appearance.

The palettes available include *PowerFactory Standard*, which is the default palette, and *PowerFactory Classic*, which consists of the colours that were formerly used as a default colours in *PowerFactory*.

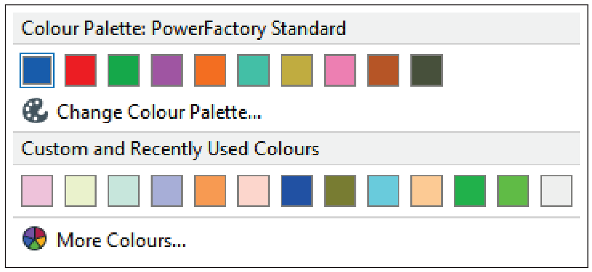


Figure 4.7.7: Choosing colours from the *PowerFactory Standard* palette

In addition, users can create, store and use their own colour palettes within their projects, using individual colour definition.

4.7.5.2 Individual colour definition

The configuration of individual colours is done via the *Select Colour* dialog, shown in Fig 4.7.8 (this dialog is also reached when clicking on *More colours . . .* in the dialog shown in figure 4.7.7). Colours can be defined using the standard HSV, RGBA or HTML formats, or simply by clicking on the colour panel. Another option is to use the *Pick Screen Colour* button to reproduce any colour visible on the user's screen.

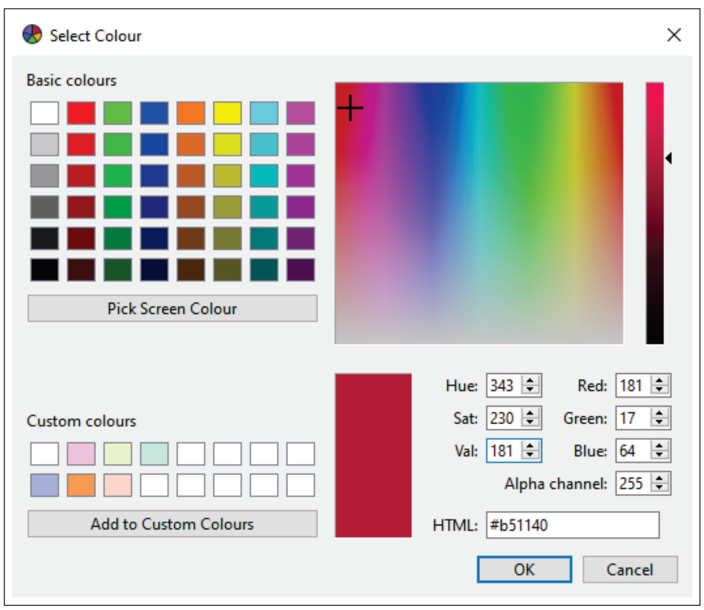


Figure 4.7.8: The *Select Colour* dialog

This dialog also allows the user to add a colour which they have defined to the list of custom colours, which will then be available for future use.

4.7.5.3 Colouring of plots

For many of the functions in *PowerFactory*, the results can be presented in plots. The various plot options are described in Section 19.7, but the general principle is that the user can select a palette and then the colours will be automatically assigned from that palette. Nevertheless, the user still has the option to configure the colouring of individual curves.

4.7.5.4 Colouring of elements in diagrams

There are wide-ranging options available for the colouring of single line diagrams, substation diagrams and geographic diagrams, which are all described in Section 9.3.7.1. These colouring options are based around network elements, and the user has the freedom to configure the colours used, via the *Colour Settings* buttons found in the *Diagram Colouring Scheme* dialog, which open the colour settings dialog.

On the General page of the colour settings dialog, the user will find an option to select a colour palette for diagrams. Selecting this palette will not automatically change any colour settings, but it will mean that the colours from the palette are offered as a default range when colours are subsequently defined for elements.

4.7.5.5 Colouring in the Network Model Manager and Data Manager

Some grouping objects such as Zones or Boundaries have colours associated with them. These colours are individually configurable, with the same option as above to assign a default palette for colour selection.

Another use of colour in the Network Model Manager and Data Manager is the colouring of fields in the flexible data page. These colours can be changed via the user settings (see Section 7.8), using the *Select Colour* dialog.

4.8 Scripting in *PowerFactory*

For automating tasks in *PowerFactory*, two scripting options are available: use of the inbuilt *DIgSILENT* **P**rogramming **L**anguage **DPL**, or scripting using Python.

4.8.1 DIgSILENT Programming Language (DPL) Scripts

DPL offers an interface to the user for the automation of tasks in *PowerFactory*. By means of a simple programming language and in-built editor, the user can define automation commands (scripts) to perform iterative or repetitive calculations on target networks, and post-process the results.

To find the name of an object parameter to be used in a DPL script, simply hover the mouse pointer over the relevant field in an object dialog. For example, for a general load, on the *Load Flow* page, hover the mouse pointer over the *Active Power* field to show the parameter name *plini*.

User-defined DPL scripts can be used in all areas of power system analysis, for example:

- Network optimisation
- Cable-sizing
- Protection coordination
- Stability analysis
- Parametric sweep analysis
- Contingency analysis

DPL scripts may include the following:

- Program flow commands such as 'if-else' and 'do-while'
- *PowerFactory* commands (i.e. load-flow or short-circuit commands: *ComLdf, ComShc*)

- Input and output routines
- Mathematical expressions
- *PowerFactory* object procedure calls
- Subroutine calls

DPL command objects provide an interface for the configuration, preparation, and use of DPL scripts. These objects may take input parameters, variables and/or objects, pass these to functions or subroutines, and then output results.

By default, DPL commands are stored inside the Scripts folder of the project.

Consider the following simple example shown in Figure 4.8.1 to illustrate the DPL interface, and the versatility of DPL scripts to take a user-selection from the single line graphic. The example DPL script takes a load selection from the single line graphic, and implements a while loop to output the Load name(s) to the output window. Note that there is also a check to see if any loads have been selected by the user.

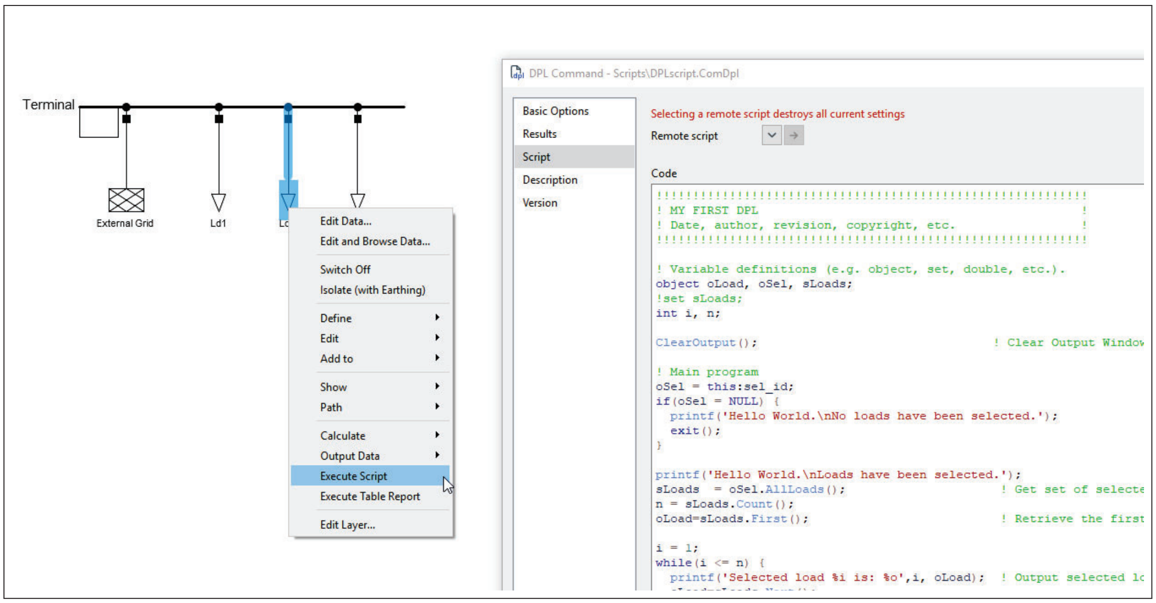


Figure 4.8.1: Example DPL Script

For further information about DPL commands and how to write and execute DPL scripts refer to Chapter 23 (Scripting), and the DPL Reference.

4.8.2 Python Scripts

In addition to DPL it is also possible to use the Python language to write scripts to be executed in *PowerFactory*.

This can be done in one of two ways:

- The script can be written directly in a Python object (*ComPython*) in *PowerFactory*, or
- For more complex scripts, a Python script can written in an external editor and linked to the Python command (*ComPython*) inside *PowerFactory*.

For further information about the Python command and how to write and execute Python scripts refer to Chapter 23 (Scripting), and the Python Reference.

Part II

Administration

Chapter 5

Program Administration

This chapter provides information on how to configure *PowerFactory*, and how to log on. More detailed descriptions of the installation, database settings and additional information can be found in the Advanced Installation and Configuration Manual.

5.1 Program Installation and Configuration

In general there are 3 primary questions to consider before installing *PowerFactory* software, which will determine the installation settings:

- Licence: Where should the licence key(s) reside?
- Installation: Where should *PowerFactory* be installed?
- Database: Where should the database reside?

Once *PowerFactory* has been installed, it can be started by clicking either on the Desktop or by selecting *PowerFactory* in the Windows Start menu. *PowerFactory* will then start and create a user account upon the initial user log-in. If the user is working in a single-user-database environment, *PowerFactory* will take the username from Windows by default.

The user will then be automatically logged on. If more user accounts are subsequently created, the user can switch to other users, or the Administrator account can be used to change the login policy so that a dialog is presented when the user starts to log in. See section 6.4.2 for more information.

In a multi-user-database installation (see Chapter 6: User Accounts, User Groups, and Profiles) new accounts and passwords are created by the administrator. The 'Administrator' account is created when installing *PowerFactory* and is used to create and manage users' accounts in a multi-user environment (see Chapter 6: User Accounts, User Groups, and Profiles). To log on as Administrator, the shortcut from the Windows Start Menu can be used. When already running a *PowerFactory* session, the user can select *Tools* → *Switch User* in the main menu to log-on as Administrator.

For further information about the role of the database administrator refer to Section 6.2: The Database Administrator. Many of the activities carried out by the Administrator are easily accessed using the Administration menu; this is found on the main toolbar but is only visible if the user is logged on as Administrator. See Section 6.3 for further details.

Changes to the application settings can be made using the *PowerFactory* Configuration dialog. Once *PowerFactory* is started, the Configuration dialog can be accessed via *Tools* → *Configuration* in *PowerFactory*'s main menu. Administrator rights are necessary to perform changes to these settings.

5.2 *PowerFactory* Configuration

The configuration of the application is stored in an “ini” file located with the executable.These settings can be changed within *PowerFactory* via the Configuration (*SetConfig*) dialog, which is available via *Tools* → *Configuration*. Depending on where the file is stored, Windows administrator rights might be required to change these settings.

5.2.1 General Page

On this page the user can select the application language for the session.

5.2.2 Database Page

This page allows the user to specify what kind of database will be used. The options are:

- A single-user database which resides locally on each computer
- A multi-user database (used in conjunction with the appropriate licence) which resides on a remote server. Here all users have access to the same data simultaneously. In this case, user accounts are created and administrated exclusively by the Administrator.

DIgSILENT *PowerFactory* provides drivers for the following multi-user database systems:

- Oracle
- Microsoft SQL Server
- PostgreSQL

For further information regarding the database configuration refer to the Advanced Installation and Configuration Manual.

5.2.3 Workspace Page

The *Workspace* page allows the user to set the workspace directory and the workspace backup directory. The workspace is used to store the local database, results files and log files. For further information regarding options for configuring and using the workspace, refer to Chapter 5.4.

5.2.4 External Applications Page

The *External Applications* page is used to configure the external programs.

Python

- **Interpreter**: The Python Interpreter to be used can either be selected by version or by directory.
- **Interpreter**: A number of Python versions are supported
- **Used Editor**: There are three options to set the Python editor:
  - ***internal:*** uses the internal editor provided by *PowerFactory*. This editor is the same used when writing DPL scripts. More information about this editor can be found in section 23.1.3.
  - ***system default:*** uses the system’s default editor for Python files (*.py); if no editor is defined as default for Python files, then the default editor for text files (*.txt) is used. This is the default option.

  - ***custom:*** here the user can customise which editor should be used to open Python files.

Visual Studio

Here the *Version* and *Shell Extension* of Visual Studio can be set. Visual Studio is used for the compilation of DSL models.

PDF Viewer

Here the User can select which program should be used to open “.pdf” files. The are three options to set the PDF viewer:

- ***system viewer:*** uses the system’s default editor for pdf files (*.pdf). This is the default option.
- ***Sumatra PDF:*** uses “Sumatra PDF” which is included in the *PowerFactory* installation.
- ***custom:*** here the user can customise which viewer should be used to open pdf files (*.pdf).

5.2.5 Network Page

The *Network* page is used to specify an HTTP proxy in the case where the user’s computer connects to the internet via a proxy server.

Proxy configuration

Three options are available for specifying the proxy configuration, including options to change the proxy settings externally to *PowerFactory* if required.

- Use the system proxy settings.
- Configure the proxy manually, supplying the host name and port number.
- Provide a path to a proxy auto-configuration file (PAC).

Proxy authentication

If it is necessary to provide authentication details to the proxy, this option is also checked, and the relevant protocol selected from the drop-down list. Unless the username and password are to be taken from the Windows user authentication details, they are entered here.

There is also a “Check internet Connection” button to check whether the configuration has been set up successfully.

5.2.6 Geographic Maps Page

On the *Geographic Maps* page, the default settings for background maps can be changed. The following parameters can be set:

- **Map Tile Cache**
  - **Directory**: Map cache directory where downloaded map tiles are stored (default: workspace directory). A custom directory can be specified if the cache should be shared across different *PowerFactory* installations.
  - **RAM cache limit**: This setting can be used to limit the amount of application memory that should be used.
- **Network Settings**
  - **Preferred tile size [pixels]**: Pixel dimensions of map tiles.
  - **Max server connections**: Maximum number of map tiles that are downloaded simultaneously.

  - **Download time-out**: Timespan after which a non-finished tile download is cancelled. This value may need to be increased for slow/unstable internet connections.

- **Google Maps for Business account**

  If Google Maps is to be used as the map provider, the "Google Maps for Business account" data must be set on this page as well. To acquire a licence, please contact Google sales: (http://www.google.com/enterprise/mapsearth).

Similarly, the licence keys for other map providers can be entered.

5.2.7 Advanced Page

General tab

- Paths in the **Additional Directories in PATH** field are used to extend the Windows path search. Typically this is required for the Oracle client.
- **Directories for external digex* libraries (DLL)**: The digex* libraries contain the compiled dynamic models.

Advanced tab

Settings on the *Advanced* tab should only be changed under the guidance of the DIgSILENT *PowerFactory* support (see Chapter 2 Contact).

5.3 Licence

5.3.1 Select Licence

In order to run *PowerFactory*, the user is required to define licence settings in theDIgSILENT *PowerFactory Licence Manager*, its dialog can be accessed via *Tools* → *Licence*→ *Select Licence...*

Note: The DIgSILENT *PowerFactory Licence Manager* can be started externally using the corresponding shortcut in the main installation folder of *PowerFactory* or in the *Windows* start menu.

The *Licence Access* defines the type of licence, which can be a local licence (either a licence file or a USB dongle) or a network licence.

Automatic search

This option searches automatically local and network licences via a broadcast and chooses the first one found without further input.

Local Softkey / USB dongle

If local softkey / USB dongle is chosen, the *Local Licence Settings* require the selection of a *Licence Container*. The locally found containers are available in the drop-down-list.

Network licence

If network licence is chosen, the server name has to be selected from the drop-down-list or entered manually in the *Network Licence Settings*. Pressing will refresh the list of available licence servers in the network. For the specified server the *Licence container* can be chosen from a drop-down-list or entered manually.

Selected Licence:

The field on the right side of the dialog shows various details relating to the selected licence. This includes the order ID (useful for any contact with the sales department), the customer ID (useful for contact with technical support), the maximum number of concurrent users for a multi user environment and a list of the licensed additional modules. Note that the expiry date of the maintenance period for the licence is also shown.

If problems with the licence occur, the button *Create Licence Support Package* creates a zipped file with the needed information for the support to identify the cause of the problems.

5.3.2 Activate / Update / Deactivate / Move Licence

These options are relevant for local licences, where the user has to manage the licence. In a network licence environment, this is done by the network administrator.

For the activation, the update and the deactivation process the licence related *Activation Key* has to be entered into the upcoming dialog.

A *PowerFactory* software licence softkey can be moved between computers a limited number of times per year. The licence move is a two-stage process:

1. An activated licence needs to be transferred back to the *DIgSILENT* server via the Deactivate Licence feature of the Licence Manager.
2. The deactivated licence can be activated again on any computer.

More information regarding licence types and their management is available in the Advanced Installation and Configuration Manual.

5.3.3 Licence and Maintenance Status

Via *Help* → *About PowerFactory*, users can get a summary of their *PowerFactory* installation, including version number, available calculation functions, and licence and maintenance information.

Left-clicking on the traffic light icon located in the lower right corner of the main window status bar gives quick access to this Help/About dialog, in which detailed information about licence and maintenance status can be found, as shown in Figure 5.3.1.

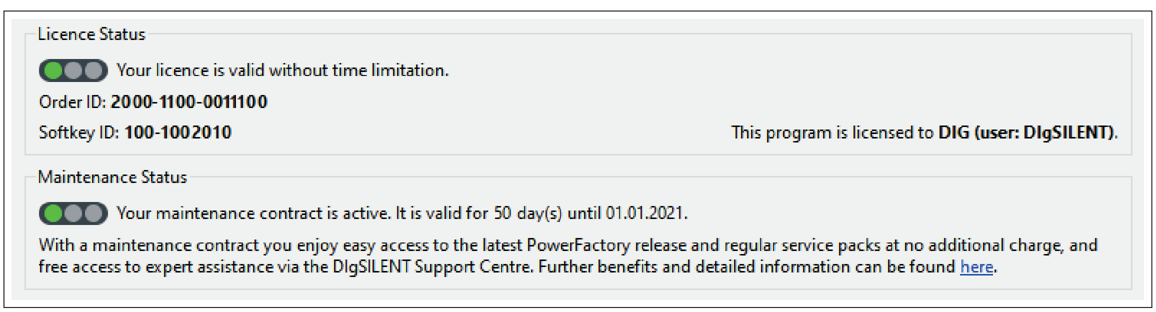


Figure 5.3.1: Licence and Maintenance Status

Two states are monitored and categorised according to a traffic light colour system.

State of time-limited licences

- Green: licence can be used for at least 6 days

- Yellow: licence expires in less than 6 days
- If the licence has expired, *PowerFactory* will not start

State of maintenance contract

- Green: maintenance contract is valid for at least 46 days
- Yellow: maintenance contract expires in less than 46 days
- Red: maintenance contract has expired

Note: The traffic light icon shows the more severe of the two states with the time-limited licence state having priority if both states have equal severity.

5.4 Workspace Options

By selecting *Tools* → *Workspace* from the main menu, the options described below are available.

5.4.1 Show Workspace Directory

The workspace directory can be seen by clicking *Tools* → *Workspace*→ *Show workspace directory*.

5.4.2 Import and Export Workspace

The ability to export and import the workspace can be a convenient way of transferring settings and local databases from one installation to another. The location of the directory can be configured via the *PowerFactory* Configuration menu.

To import the workspace, select *Tools* → *Workspace*→ *Import Workspace...*. This is a convenient way to import the entire workspace after a new installation.

To export the workspace, select *Tools* → *Workspace*→ *Export Workspace...*. The package will be saved as a .zip file.

5.4.3 Show Default Export Directory

The selection *Tools* → *Workspace*→ *Show Default Export Directory* from the main menu shows the user the directory that is used for the export. In particular, this directory is used for automated backups, e.g. before migration. The location of the directory can be configured via the *PowerFactory* Configuration menu.

5.4.3.1 Migration Types

Complete: the database structure and all projects will be altered and migrated immediately upon pressing the **OK** button.

Minimal: the database structure will be altered immediately, but the project migration will occur upon activation.

Minimal migration is recommended for the migration of large databases.

5.5 Offline Mode User Guide

This section describes working in offline mode. Installation of the offline mode is described in the Advanced Installation and Configuration Manual.

The Offline Mode concept was introduced with users of multi-user databases in mind. Users who have a Team Edition licence make use of a multi-user database because of the benefits it brings in terms of sharing data. Sometimes, however, users wish to work detached from the main database. The following terms are used in this section:

- **Online**: Connected to, and working in, the multi-user database
- **Offline**: Disconnected from the multi-user database and working in a local database cache.

5.5.1 Functionality in Offline Mode

5.5.1.1 Start Offline Session

Preconditions:

- A *PowerFactory* user account must already exist in the online database. The *PowerFactory* "Administrator" user is able to create user accounts.
- A user can only start an offline session if he/she is not currently logged on.

Note: the Administrator user is only allowed to work in online mode (not in offline mode).

To create an offline session, follow these steps:

- Start *PowerFactory*. In the Log-on dialog enter the user name and password.
- On the *Database* page, enter the *Offline Proxy Server* settings (see Figure 5.5.1)

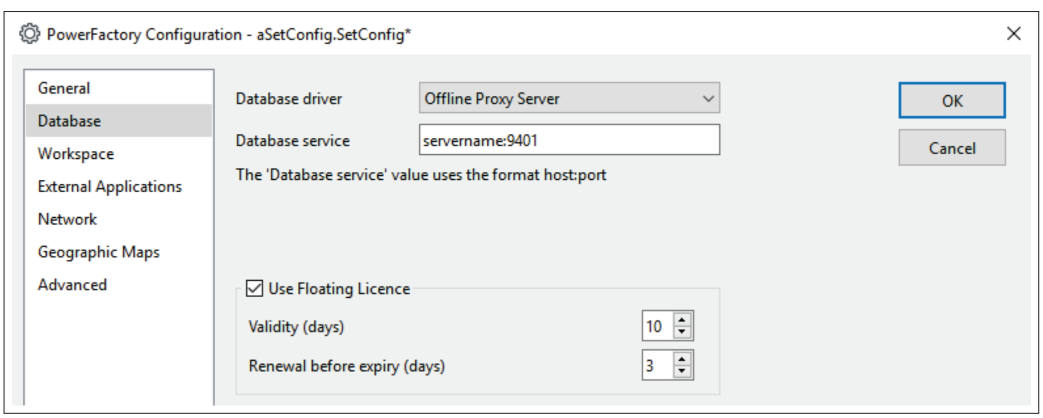


Figure 5.5.1: Log-on dialog, *Database* page

Note: Using a floating licence with the offline mode allows working with *PowerFactory* without connection to the licence server. Please note, that the usage of floating licences has to be included in the network licence and activated in the user settings.

- Press **OK**

- If the usage of a floating licence is configured, *PowerFactory* will generate the floating licence and adapt the licence settings. *PowerFactory* has to be started again afterwards.
- An information dialog appears, saying “Offline database isn’t initialised yet. The initialisation process may take several minutes”.
- Press **OK**
- Following initialisation, the usual *PowerFactory* application window is shown.

5.5.1.2 Release Offline Session

- From the main menu, select *File* → *Offline*→ *Terminate Offline session*
- A warning message is shown to confirm the synchronisation
- Press **Yes**
- All unsynchronised local changes will then be transferred to the server and the local offline database is removed.
- If a floating licence has been used in offline mode, this licence will be returned to the licence server.

5.5.1.3 Synchronise All

Synchronises global data (new users, projects added, projects removed, projects moved) and all subscribed projects.

- Open the main menu *File* → *Offline*→ *Synchronise all*

5.5.1.4 Subscribe Project for Reading Only

- Open the Data Manager and navigate to the project.
- Right-click on the project stub. A context menu is shown.
- Select *Subscribe project in offline mode for reading only*.

The project will then be retrieved from the *Offline Proxy Server* and stored in the local *Offline DB cache*.

5.5.1.5 Subscribe Project for Reading and Writing

Write access to the project is required.

- Open the Data Manager and navigate to the project.
- Right-click on the project stub. A context menu is shown.
- Select *Subscribe project in Offline mode for reading and writing*.

5.5.1.6 Unsubscribe Project

- Open the Data Manager and navigate to the project.
- Right-click on the project. A context menu is shown.
- Select *Unsubscribe project in Offline mode*.

5.5.1.7 Add a New Project

A new project is created in offline mode. It is available only in this offline session. Later this project should be published to other users and synchronised to the online database.

- Create a new project or import a PFD project file.
- Open the Data Manager and navigate to the project.
- Right-click on the project stub. A context menu is shown.
- Select *Subscribe project in Offline mode for reading and writing*.

5.5.1.8 Synchronise Project

Synchronises a subscribed project. If the project is subscribed for reading only, the local project will be updated from the online database. If the project is subscribed for reading and writing, the changes from the local offline database will be transferred to the online database.

- Open the Data Manager and navigate to the project
- Right-click on the project stub. A context menu is shown.
- Select *Synchronise*

5.5.2 Functionality in Online Mode

5.5.2.1 Show Current Online/Offline Sessions

The session status for each user is shown in the Data Manager.

Users who are working online appear like this , and those working offline like this .

5.5.3 Terminate Offline Session

There may occasionally be cases which require that an offline session be terminated by the Administrator; e.g. if the computer on which the offline session was initialised has been damaged and can no longer be used, and the user wants to start a new offline session on a different computer.

The Administrator is able terminate a session as follows:

- Right-click on the user; the context menu is shown.
- Select *Terminate session*
- A warning message is shown to confirm the synchronisation.
- Press **Yes**

5.6 Housekeeping

5.6.1 Introduction

Housekeeping automates the administration of certain aspects of the database; in particular purging projects, emptying user recycle bins and the deletion of old projects. Housekeeping is triggered by the execution of a Windows Scheduled Task; this can be set up to run at night, thus improving performance during the day by moving regular data processing to off-peak times. An additional benefit to housekeeping is that users will need to spend less time purging projects and emptying recycle bins, something that can slow down the process of exiting *PowerFactory*.

Housekeeping is only available for multi-user databases (e.g. Oracle, SQL Server). For details on scheduling housekeeping, see the *PowerFactory* Advanced Installation and Configuration Manual.

5.6.2 Configuring Permanently Logged-On Users

Normally, housekeeping will not process data belonging to logged-on users; however, some user accounts (e.g. those for a control room) may be connected to *PowerFactory* permanently. These users can be configured to allow housekeeping to process their data while they are logged on.

This is done from the User Settings dialog, Miscellaneous page, by selecting *Allow housekeeping task to operate when user is connected.*

Regardless of this setting, housekeeping will not operate on a user's active project.

5.6.3 Configuring Housekeeping Tasks

The Housekeeping command (*SetHousekeeping*) is used to control which housekeeping tasks are enabled.

It is recommended that the user move this object from Database \System\Configuration\Housekeeping to Database\Configuration\Housekeeping, in order to preserve the user's configuration throughout database upgrades.

The following sections discuss the different housekeeping tasks available in the Housekeeping dialog.

5.6.4 Project Archiving

Project archiving provides the following options:

- **Disable**: Archiving is not used.
- **Immediate archiving by the user**: by selecting "Archive" from the context menu, the project will be immediately archived and placed in the vault directory.
- **Deferred archiving by Housekeeping job**: by selecting "Archive" from the context menu, the project will be immediately archived, but not placed in the vault directory. This will happen automatically depending on the Housekeeping settings.

Important: The vault directory can be defined under "Tools\Configuration\Database \Vault Directory"

A project cannot be archived unless it is deactivated. By right-clicking on the project a context menu will appear. By selecting "Archive", the project will be moved to the Archived Projects folder of the user (*IntUser*). If specified in the Housekeeping archiving options, the project will be immediately placed in the vault directory.

Conversely, archived projects may also be restored. To restore an archived project, the user must select "Restore" from the context menu which appears after right-clicking on a deactivated project.

5.6.5 Configuring Deletion of Old Projects

If the option *Remove projects based on last activation date* has been selected in the Housekeeping dialog, when the Housekeeping is executed, for each user, each project will be handled according to the selected *Action*.

The *Action* options are:

- **Delete project**: deletes the project
- **Archive project**: archives the project

The project properties determine whether a project can be automatically deleted or archived.

The settings are found on the Storage page of the project dialog, and by default the option “Housekeeping project deletion” is not selected. If it is selected, a retention period can be specified, by default 60 days.

These defaults can be changed for new projects by using a template project (under Configuration/Default in the Data Manager tree).

The settings for multiple projects can be changed in a data manager on the Storage tab, as shown below in Figure 5.6.1. A value of ’1’ is equivalent to the Housekeeping option *Delete project* being selected.

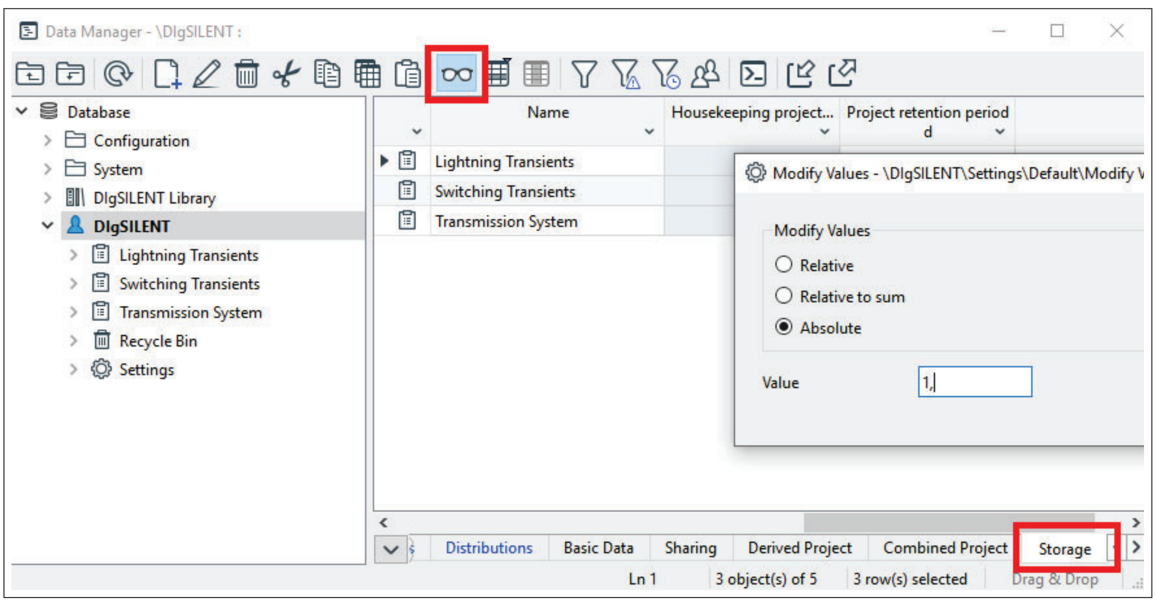


Figure 5.6.1: Setting parameters for multiple projects

A project will be deleted/archived by the housekeeping task if it meets the following criteria:

1. The project is configured for automatic deletion/archiving on the *Storage* page of the project properties.
2. The last activation of the project is older than the retention setting on the project.
3. It is not a base project with existing derived projects.
4. It is not a special project (e.g. User Settings, or anything under the System or Configuration trees).
5. The project is not locked (e.g. active).
6. The owner of the project is not connected, unless that user is configured to allow concurrent housekeeping (see Section 5.6.2).

5.6.6 Configuring Purging of Projects

A *PowerFactory* project contains records of changes to data, which makes it possible to roll back the project to an earlier state using versions (see section 21.2). However, as the user works with the project and makes changes to it, the number of records increases and it is useful to remove older, unwanted records in a process known as “purging”.

If *Purge projects* has been ticked in the Housekeeping dialog, when the Housekeeping is executed, each project will be considered for purging. A project that is already locked (e.g. an active project) will not be purged.

The criteria used by Housekeeping to purge a project are:

- If the project has been activated since its last purge.
- If it is now more than a day past the object retention period since last activation, and the project has not been purged since then.
- If the project is considered to have invalid metadata (e.g. is a pre-14.0 legacy project, or a PFD import without undo information).

Once housekeeping has been configured to purge projects, the automatic purging of projects on activation may be disabled by the user, thus preventing the confirmation dialog popping up. To do this, the option *Automatic Purging* should be to *Off* on the *Storage* page in the Project Properties dialog. This parameter can also be set to *Off* for multiple projects (see Section 5.6.5 for details).

5.6.7 Configuring Emptying of Recycle Bins

If *Delete recycle bin objects* is set in the Housekeeping dialog, when Housekeeping is executed, each user’s recycle bin will be examined.

Entries older than the number of days specified in the Housekeeping dialog will be deleted.

There is also an option *Prevent manual clearing of recycle bin*. If the Administrator sets this, individual users will not be able to clear their own recycle bins, or delete individual objects therein.

5.6.8 Monitoring Housekeeping

In order to ensure that housekeeping is working correctly, it should be regularly verified by an administrator. This is done by inspecting the HOUSEKEEPING_LOG table via SQL or the data browsing tools of the multi-user database. For each run, housekeeping will insert a new row to this table showing the start and end date/time and the completion status (success or failure). Other statistics such as the number of deleted projects are kept. Note that absence of a row in this table for a given scheduled day indicates that the task failed before it could connect to the database. In addition to the HOUSEKEEPING_LOG table, a detailed log of each housekeeping run is stored in the log file of the housekeeping user.

5.6.9 Summary of Housekeeping Deployment

The basic steps to implement housekeeping are:

1. Set up a Windows Scheduled Task, as described in the *PowerFactory* Advanced Installation and Configuration Manual.
2. Configure those users expected to be active during housekeeping, as described in Section 5.6.2.
3. Configure the Housekeeping dialog as described in Section 5.6.3.

4. If using the project deletion/archiving task, configure automatic deletion/archiving properties for new projects, as described in Section 5.6.5.
5. If using the project deletion/archiving task, configure automatic deletion/archiving properties for existing projects, as described in Section 5.6.5.
6. Regularly monitor the HOUSEKEEPING_LOG table to verify the status of housekeeping runs, as described in Section 5.6.8.

Chapter 6

User Accounts, User Groups, and Profiles

This chapter provides details of how to create and manage user accounts, user groups, and profiles. The information in this chapter is particularly relevant for a multi-user database (i.e. *Team Edition*), and will not generally be of so much interest to a user with a single-use installation.

Key objectives of the user account managing system are to:

- Protect the 'system' parts of the database from changes by normal (non-administrator) users.
- Configure and manage access to the database, via options for the authentication mode to be used and options for password management.
- Manage settings relating to data security and privacy.
- Facilitate both the sharing of user data and the restriction of data visibility between one user group and another.

The user account managing system provides each user with their own "private" database space. The user is nevertheless able to use shared data, either from the common system database or from other users, and may enable other users to use data from their private database.

The user account managing system manages this whilst using only one single database in the background, which allows for simple backup and management of the overall database.

The default name for a *PowerFactory* user (unless using *Team Edition*) is the Windows user name, which is automatically created when *PowerFactory* is started for the first time.

6.1 *PowerFactory* Database Overview

A brief introduction to the top level structure of the *PowerFactory* database is convenient before presenting the user accounts and their functionality.

The data in *PowerFactory* is stored inside a set of hierarchical directories. The top level structure is constituted by the following folders:

- **Configuration:** contains company specific customising for user groups, user default settings, project templates and class templates for objects. The configuration folder can only be edited by the administrator and is read only for normal users.
- **System:** contains all objects that are used internally by *PowerFactory*. The system folder contains

default settings provided by *DIgSILENT* and these should not be changed. They are automatically updated upon migration to a new *PowerFactory* version.

- **DIgSILENT Library:** contains all standard types and models provided with *PowerFactory*. The main library folder is read only for normal users.
- **User accounts:** contain user project folders and associated objects and settings.

The structure described above is illustrated in Figure 6.1.1

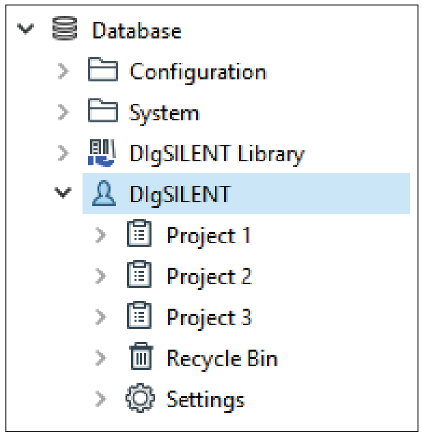


Figure 6.1.1: Basic database structure

6.2 The Database Administrator

A database administrator account is created with the *PowerFactory* installation. The main functions of the administrator are:

- Creation and management of user accounts.
- System database maintenance under the guidance of the *DIgSILENT* customer support.

Under a multiuser database environment, the administrator is the only user with permissions to:

- Add and delete users.
- Define users groups.
- Set individual user rights.
- Restrict or allow calculation functions.
- Set/reset user passwords.
- Create and edit Profiles (see Section 6.6 for details).
- Configure various settings such as housekeeping processes, parallel processing configuration, password security options etc.

The administrator is also the only user that can modify the main library and the system folders. Although the administrator has access to all the projects of all the users, it does not have the right to perform any calculations.

To log on as administrator, there are two options:

- Select the Shortcut in the Windows Start Menu *PowerFactory* 20nn (Administrator).

- Log into *PowerFactory* as a normal User and select via the Main menu *Tools* → *Switch User*. Select **Administrator** and enter the corresponding password.

For further information about the administrator role, refer to the Advanced Installation and Configuration Manual.

6.3 Administration Menu

To assist the administrator, an Administration menu is provided to give easy access to the more important settings; this is found on the main toolbar but is only visible if the user is logged on as Administrator. These are the options available from this menu:

6.3.1 User Management

The options available here are:

- Show Users...
- Show Groups...
- User Manager...

These are used to manage User accounts and User Groups as described in Sections 6.5 and 6.6.

6.3.2 Security and Privacy

The options available here are:

- Audit Log...
- Login Policy...
- Idle Session Timeout...
- External Data Access...
- Privacy...

These are described in Section 6.4 below.

6.3.3 Calculation Settings

The options available here are:

- Parallelisation...This allows the administrator to configure the Parallel Computing Manager.
- Linear Programming...This brings up a dialog which enables the administrator to configure which internal and external linear programming solvers should be made available to users who are using the Unit Commitment and Dispatch Optimisation module. See Section 39.3.7.2.

6.3.4 Housekeeping

This gives access to the dialog for setting up Housekeeping tasks. Details can be found in Section 5.6.

6.4 Security and Privacy

This section gives an overview of the security and privacy features which can be managed by the administrator. Note that more detail is provided in the Advanced Installation and Configuration Manual.

6.4.1 Audit Log

The Audit Log is a log of key activities on the database, and is useful for the administrator of a multi-user database.

6.4.1.1 Enabling the Audit Log

The log can be enabled by the administrator via the Administration Menu (*Administration* → *Security and Privacy*→ *Audit Log...*), where a retention period is also set. By default the log is not enabled, and it should be noted that if the log is enabled then later disabled, all records will be lost. The information in the Audit log is securely held in the database itself.

6.4.1.2 Using the Audit Log

An Audit Log command *ComAuditlog* can be created by the Administrator user and used to access the information in the log. The command options are:

- **Report**, to generate a high-level report to the Output Window
- **Export**, to export a detailed report
- **Check Integrity** As an additional assurance, it is possible to carry out a data integrity check on the Audit Log data to detect any data manipulation.

A more detailed description of the Audit Log and what it contains can be found in the Advanced Installation and Configuration Manual.

6.4.2 Login Policy Options

6.4.2.1 Authentication Mode

Here the administrator determines what authentication (username and password) mode will be used. The options are:

- **PowerFactory authentication** provides built-in user management, where the users must enter their *PowerFactory* usernames and passwords
- **Active Directory authentication** uses the external Windows Active Directory for user authentication
- **No authentication**

If the *Active Directory* authentication is selected, then the user can click on the "..." to the right of each group to select the Active Directory group, then give it an appropriate name within *PowerFactory*. All groups must be within the same domain. It should be noted that only one user in the second group (Administrators) may be logged in at any one time.

6.4.2.2 Password Policy

If the authentication mode *Powerfactory authentication* is selected, further options appear, allowing the administrator to impose rules to enforce regular password changes and/or put in place rules on password quality, such as length and character diversity.

6.4.3 Idle Session Timeout

In this option, the administrator can set a time-limit after which any idle *PowerFactory* session will be terminated. Such a session will be closed down in an ordered way, but it should be noted that unsaved scenario changes will be lost. A session will only be considered as idle if there has been no activity for the prescribed time, where "activity" means user activity such as mouse-clicks or keyboard actions. However, if a command is being executed (for example a long-running simulation, or a script), *PowerFactory* will wait until the command has been completed before terminating the session.

As well as being a good security measure, setting an Idle Session Timeout is useful in a multi-user environment because the licences are released back to the licence server.

6.4.4 External Data Access

The External Data Access dialog allows the administrator to specify permitted addresses in order to manage access to data outside *PowerFactory*. This control is related to *IntUrl* and *IntDocument* objects which are being used to access external data.

6.4.5 Privacy

This feature is available to allow database administrators to manage the visibility of user names. There are two options, which by default are not enabled:

- **Enable recording of modifying user in object**, which if checked will mean that the "Object modified by" information will include the *PowerFactory* user name.
- **Display system account in user object**, which if checked will mean that the Windows username will appear in the *IntUser* object when the user has an active session.

6.5 Creating and Managing User Accounts

In the case of an installation with a local database, the default name for a *PowerFactory* user is the Windows user name, which is automatically created when *PowerFactory* is started for the first time. (see Chapter 5: Program Administration). In this case the program will automatically create and activate the new account, without administrator intervention. In order to create other *PowerFactory* users if required, the 'User Manager' object can be used as described below:

In multi-user database installations, the administrator creates new user accounts by means of a tool called the 'User Manager', which is found in the Configuration folder.

To create a new user:

- Log on as Administrator. You can do so by starting the *PowerFactory* Administrator shortcut in the Windows Start menu or by switching the user via *Tools* → *Switch User* in the main tool bar.
- The User Manager can be accessed from the Administration menu on the main toolbar: *Administration* → *User Management*→ *User Manager...*

- Press the **Add User...** button.

The User edit dialog will be displayed. The settings are the following:

- *General page*
  - User Name: user Name that will be used for login to *PowerFactory* at startup
  - Full Name: full Name of the appropriate user. In case, that the parameter User Name is set to be an abbreviation.
  - Change Password: the administrator can change the user password here, without knowing the previous password. If this button is clicked by the user itself, the current password has to be entered as well.
  - Force password change: can be selected by the administrator.
  - User sharing: by adding different users into the list of permitted users, access for these users can be granted to login to the appropriate user account. If User A is in the list of permitted user, User A can access the user account without entering the user password.
- *Account page:*
  - Publishing user: by setting this flag, the user can be defined to be a publishing user. This means, that the user is visible to other users within the database and marked with a different symbol within the data manager. This option can be used to provide an user within the multiuser database, who publishes projects.
  - User account enabled: this setting can be used to enable/disable the user Account
  - User account is time-limited: this option will set the account to be time limited and therefore can be used for temporary users within the database.
  - Force Authentication server usage: setting this option also requires the definition of an authentication server within the *PowerFactory* configuration as explained in the manual.
- *Password Policy page:* On this page, the default Password policy (see section 6.4.2) can be customised by the administrator for the user.
- *Licence page:* if a licensed version with a restricted number of functions is used (i.e. you may have 4 licences with basic functionality, but only 2 stability licences), the *Licence* tab may be used to define the functions that a user can access. The *Multi-User Database* option should be checked for all users that will access the multi user database.
  As an alternative to allocating access to certain licence functions to individual users, it is possible to allocate access via User Groups instead. See section 6.6 below.
- *Parallel Computing:* here it can be defined whether the user is allowed to use parallel processing possibilities within *PowerFactory*. The "User defined" setting allows the individual user to customise the globally-defined allowed processes number to a lower number if required, for example to free up resources for other applications.
- *Optimisation:* the *Unit Commitment* module (see Chapter 39) offers the possibility to use in-built or external linear problem solvers, the latter requiring an additional licence module. Here, the administrator enables access to the preferred solver(s).

Existing users can be viewed via the Administration menu on the main toolbar: *Administration* → *User Management*→ *Show Users...*. The administrator can edit any user account to change the user name, set new calculation rights or change the password. To edit an existing user account:

- Right-click on the desired user and select *Edit* from the context sensitive menu. The User edit dialog will be displayed.

Any user can edit her/his own account by means of the User edit dialog. In this case only the full name and the password can be changed.

Note: The administrator is the only one who may delete a user account. Although users can delete all projects inside their account folder, they cannot delete the account folder itself or the standard folders that belong to it (i.e. the *Recycle Bin* or the *Settings* folder).

6.6 Creating User Groups

User groups are a useful way for managing various access rights and permissions within a multi-user database environment. For example, any project or folder in a user account may be shared, either with everybody or with specific user groups. User groups can also be used in conjunction with Profiles (see section 6.7) and for controlling access to licence modules.

User groups are created by the administrator via the User Manager. To create a new user group:

- Log on as Administrator.
- The User Manager can be accessed from the Administration menu on the main toolbar: *Administration → User Management→ User Manager...*
- Press the **Add Group...** button.
- Enter the name of the new group, optionally a description and press **Ok**.
- The new group is automatically created in the User Groups directory of the Configuration folder.

Existing groups can be viewed via the Administration menu on the main toolbar: *Administration → User Management→ Show Groups...*. The administrator can change the name of an existing group by means of the corresponding edit dialog (right clicking on it and selecting *Edit* from the context sensitive menu). Via the context sensitive menu, groups can also be deleted.

The administrator can add users to a group by:

- Copying the user in the Data Manager (right click on the user and select *Copy* from the context sensitive menu).
- Selecting a user group in the left pane of the Data Manager.
- Pasting a shortcut of the copied user inside the group (right-click the user group and select *Paste Shortcut* from the context sensitive menu).

Users are taken out of a group by deleting their shortcut from the corresponding group.

The administrator can also set the Groups *Available Profiles* on the *Profile* tab of the Group dialog.

In addition, the *Licence* page of the User Group can be used to configure which licence modules members of the group will have access to. For any individual user, the licence modules available to that user will be all those selected in that individual user's account set-up, plus any additional licence modules made available to the group(s) to which the user belongs.

For information about sharing projects, refer to Section 21.6 (Sharing Projects).

6.7 User Interface Customisation (Profiles)

Profiles can be used to configure aspects of the Graphical User Interface, such as toolbars, menus, dialog pages, and dialog parameters. By default, *PowerFactory* includes "Base Package" and "Standard" profiles, selectable from the main menu under *Tools → Profiles*. Selecting the "Base Package" profile limits icons shown on the Main Toolbar to those that are used with the Base Package of the software. The "Standard" profile includes all available *PowerFactory* icons.

Profiles are created in the *Configuration → Profiles* folder by selecting the *New Object* icon and then *Others → Settings→ Profile*. An administrator can create and customise profiles, and control User/User Group selection of profiles from the *Profile* tab of each group.

Figure 6.7.1 shows the Profile dialog for a new profile, *CustomProfile*, and Figure 6.7.2 illustrates aspects of the GUI that may be customised using this profile. This section describes the customisation procedure.

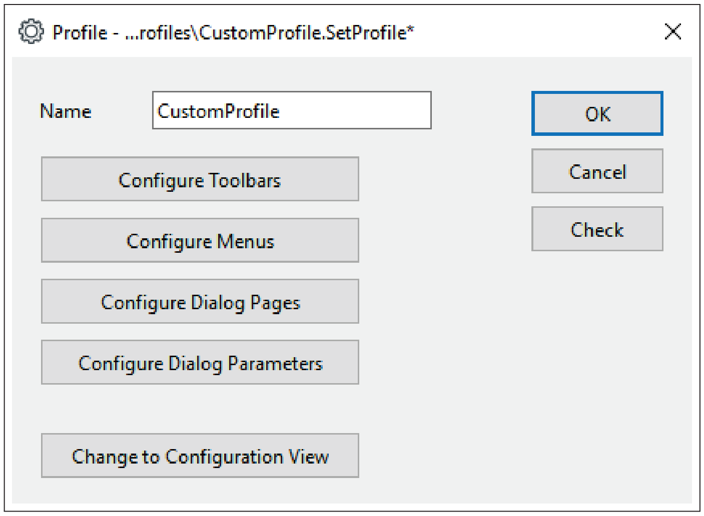


Figure 6.7.1: Profile dialog

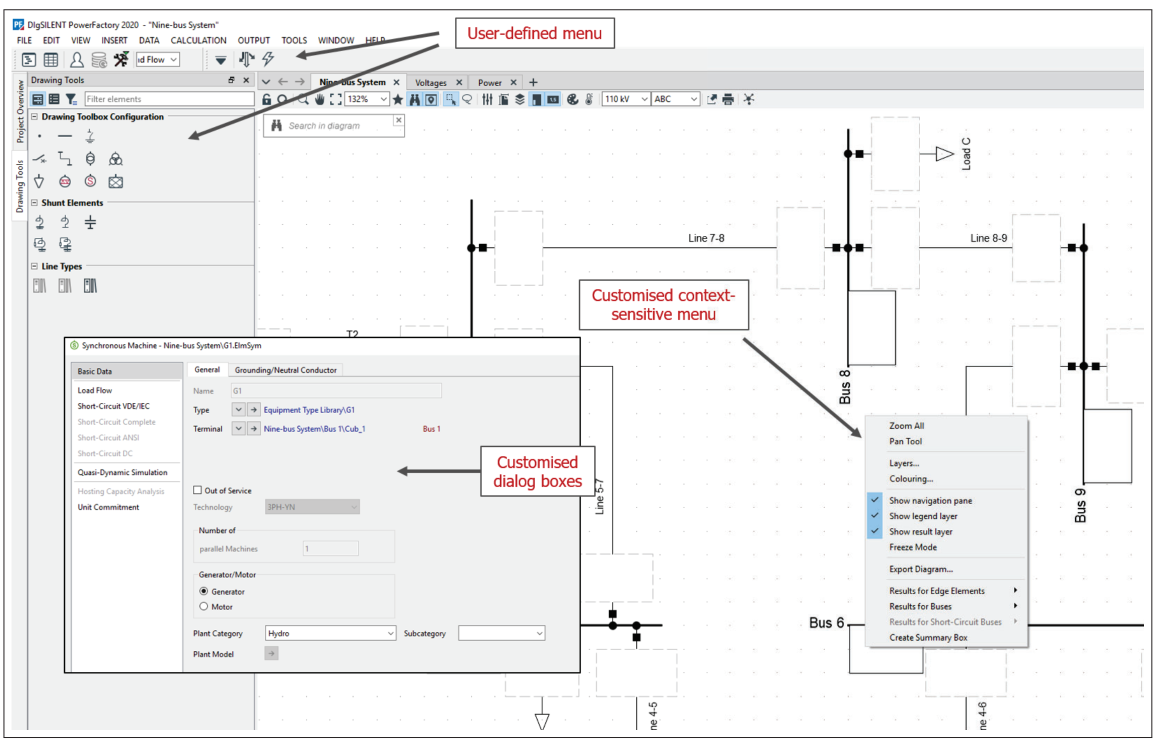


Figure 6.7.2: GUI Customisation using Profiles

6.7.1 Tool Configuration

Definition of Icons

Icons can be defined in the *Configuration* → *Icons* folder by selecting the *New Object* icon and then

Others → *Other Elements*→ *Icon (IntIcon)*. From the Icon dialog, icon images can be imported and exported. Icons should be 24 pixels by 24 pixels in Bitmap format (recommended to be 24-bit format).

Command Configuration

The User-defined Tools toolbar can be used to make commonly-used tools such as scripts and Add On Modules available to users. Changes and additions to the User-defined Tools toolbar can only be made by the administrator; from the top menu, *Tools* → *Tool Configuration...* is selected and the fields described below can be edited.

- **Command:** in this field, the relevant command or script is selected from the location where it has been stored.
  - **Scripts:** scripts may be stored within the Tool Configuration itself or in the *Configuration*, *Scripts* folder
  - **Com* objects:** generally, commands *Com** are stored within the Tool Configuration itself.
  - **Add On Modules:** add on module commands can be stored in the *Configuration*, *Add On* folder.
- **Edit:** if selected, the DPL command dialog will appear when a Command is executed. If de-selected, the DPL command dialog will not appear when a Command is executed.
- **Icon:** previously created icons can be selected, which will be shown on the menu where the command is placed. If no icon is selected, a default icon will appear (a Hammer, DPL symbol, or default Com* icon, depending on the Class type).

Template Configuration

- **Template:** the name of the template. The name may be for a unique template, or include wildcards (such as *.ElmLne) for selection of a group of templates. Templates should be in 'System/Library/Busbar Systems' folder, or in the 'Templates' folder of the active project.
- **Drawing mode:** the drawing mode can be set where there are multiple diagrammatic representations for a template (such as for a substation). Three options are available:
  - *Blank* will place the default (detailed) graphic of the template.
  - *Simplified* will place the simplified graphic of the template.
  - *Composite* will place a composite representation of the template.
- **Symbol name:** sets the representation of templates with a composite drawing mode (e.g. GeneralCompCirc or GeneralCompRect).
- **Icon:** previously created icons can be selected, which will be shown on the menu where the template is placed. If no icon is selected, a default icon will appear (a Template symbol or custom icon).
- **Description:** this description will be displayed when a user hovers the mouse pointer over the icon. If left blank, the template name will be displayed.

6.7.2 Configuration of Toolbars

The Main Toolbar and Drawing Toolbars can be customised using the Toolbar Configuration. The field *Toolboxes* may either refer to a *Toolbox Configuration* (*SetTboxconfig*) or a *Toolbox Group Configuration* (*SetTboxgrconfig*), which may in-turn refer to one or more *Toolbox Configurations*.

Figure 6.7.3 shows an example where there is a main toolbox, and a toolbox group. The toolbox group adds a *Change Toolbox* icon to the menu, which allows selection of *Basic Commands* and *Custom Commands* groups of commands.

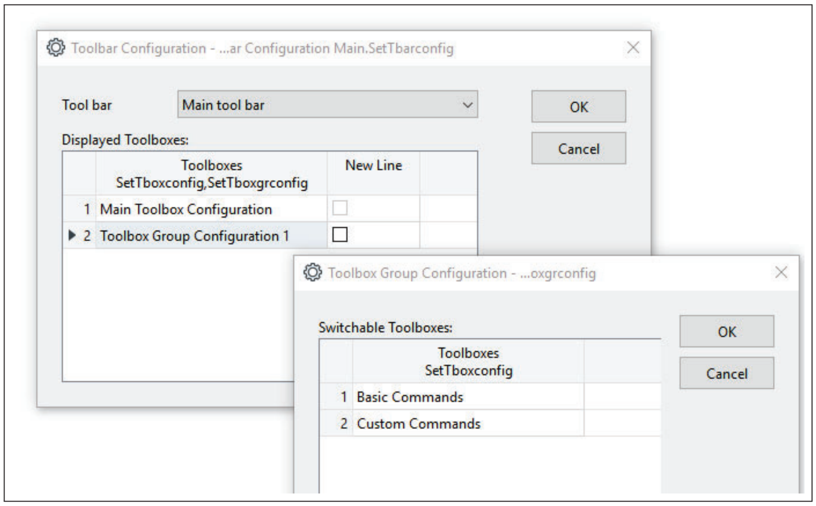


Figure 6.7.3: Toolbar Configuration

Each toolbox can be customised to display the desired icons, such as illustrated in Figure 6.7.4

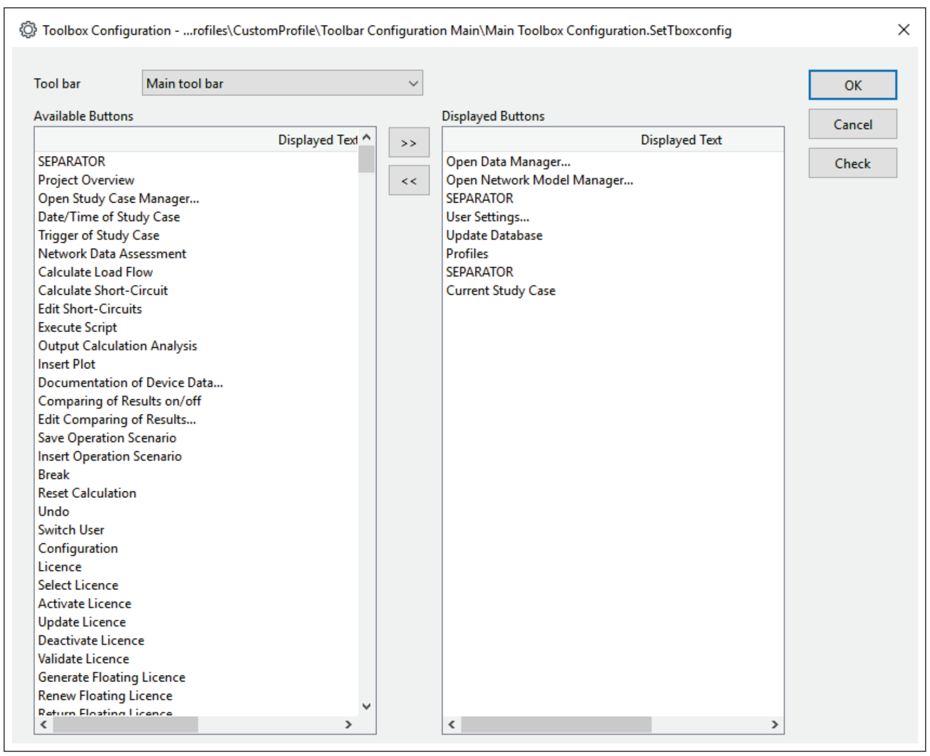


Figure 6.7.4: Toolbox Configuration

Prior to customising the displayed buttons and menu items etc, the user should first define any required custom Commands and Templates. A *Tool Configuration object* can be created in the *Configuration → Profiles* folder, or within a user-defined Profile, by selecting the *New Object* icon and then *Others → Settings→ Tool Configuration*. If created in the *Profiles* folder, the commands will be available from the "Standard" profile. Conversely, if the Tool Configuration object is created within a profile (*SetProfile*) the commands and templates will only be available for use in this profile. If there is a Tool Configuration within a user-defined profile, as well as in the *Profiles* folder, the Tool Configuration in the user-defined profile will take precedence. Optionally, customised icons can be associated with the Commands and Templates.

6.7.3 Configuration of Menus

The *Main Menu, Data Manager, Graphic, Plots*, and *Output Window* menus can be customised from the *Menu Configuration* dialog. The *Change to Configuration View* button of the Profile dialog is used to display description identifiers for configurable items, such as illustrated in the context-sensitive menu shown in Figure 6.7.5. The Menu Configuration includes a list of entries to be removed from the specified menu. Note that a Profile may include multiple menu configurations (e.g. one for each type of menu to be customised).

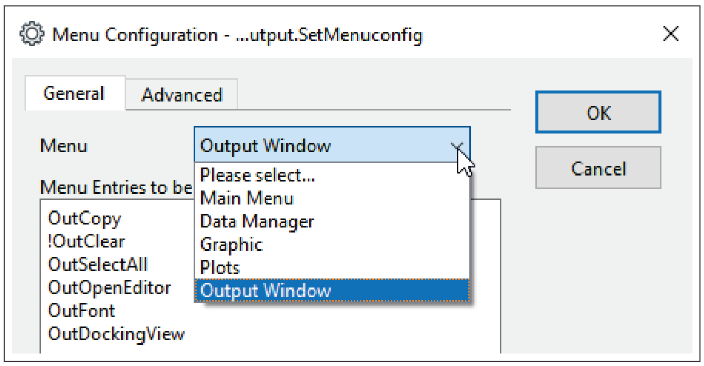


Figure 6.7.5: Menu Configuration

6.7.4 Configuration of Dialog Pages

The *Dialog Page Configuration* may be used to specify the Available and Unavailable dialog pages shown when editing elements, such as illustrated in Figure 6.7.6. Note that Users can further customise the displayed dialog pages from the *Functions* tab of their *User Settings*.

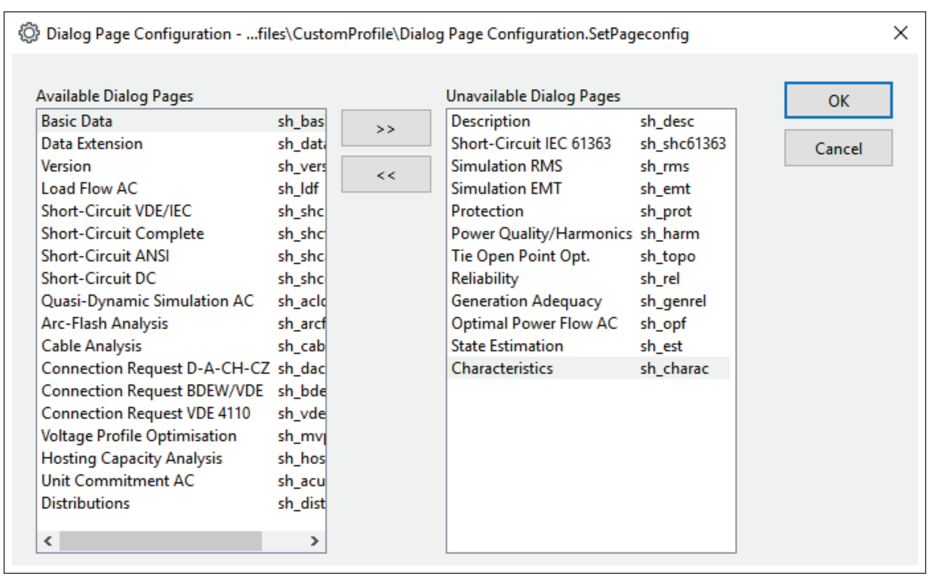


Figure 6.7.6: Dialog Page Configuration

6.7.5 Configuration of Dialog Parameters

The *Dialog Configuration* may be used to customise element dialog pages, such as illustrated for a Synchronous Machine element in Figure 6.7.7. “Hidden Parameters” are removed from the element dialog page, whereas “Disabled Parameters” are shown but cannot be modified by the user. A Profile may include multiple dialog configurations (e.g. one for each class to be customised).

Note that if a there is a Dialog Configuration for say, *Elm** (or similarly for *ElmLne,ElmLod*), as well as a dialog Configuration for *ElmLne* (for example), the configuration settings will be merged.

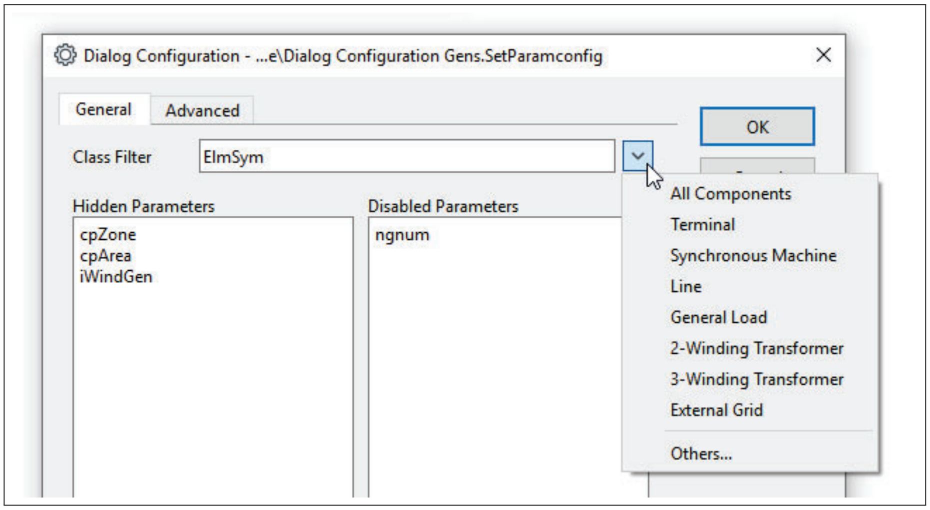


Figure 6.7.7: Dialog Configuration

Note: Configuration of Dialog parameters is an advanced feature of *PowerFactory*, and the user should be cautious not to hide or disable dependent parameters. Seek assistance from *DIgSILENT* support if required.

6.7.6 References

Profiles can also contain references to configurations. This allows several profiles to use the same configurations. These referenced configurations can either be stored in another profile or in a subfolder of the “Profiles” folder (e.g. a user-defined profile can use configurations from a pre-defined profile).

Chapter 7

User Settings

The User Settings dialog offers general settings which can be configured by the user individually. The dialog may be opened either by clicking the *User Settings* button () on the main tool bar, or by selecting the *Tools* → *User Settings...* menu item from the main menu.

7.1 Data/Network Model Manager Settings

The Data/Network Model Manager settings include:

Browser

- **Save Data Automatically.** The Data Manager and the Network Model Manager will not ask for confirmation every time a value is changed in the data browser when this option is selected.
- **Confirm Delete Action.** Causes a confirmation dialog to appear whenever something is about to be deleted.

Data Manager

- **Sort Automatically.** Specifies that objects are automatically sorted (by name) in the data browser.
- **Remember last selected object.** The last selected object will be remember when a new Data Manager window is opened.
- **Use multiple Data Manager.** When enabled, more than one Data Manager dialog can be opened at a time. When disabled only one Data Manager may be opened at a time and pressing the *New Data Manager* button will pop up the minimised Data Manager.

Operation Scenario

- If **Save active Operation Scenario automatically** is enabled, the period for automatic saving must be defined.
- **Automatically migrate to current configuration during activation.** The default is for this option to be selected. This means that after migration to a new version of *PowerFactory*, the operation scenarios will be migrated when a project is activated, and any new scenario data attributes will be assigned values based on the current status of the object. If the option is not selected, the scenario will not be migrated (and values will not be assigned) until that scenario is saved, and this may be the user preference. Note however that the scenarios have to be migrated to the current *PowerFactory* version if the Operation Scenario Manager (see Section 16.4) is used.

Export/Import Data (DZ/DZS)

Configures the export and import as follows:

- **Binary Data.** Saves binary data, such as results in the result folders, to the 'DZ' export files according to selection.
- **Export References to Deleted Objects.** Will also export references to objects which reside in the recycle bin. Normally, connections to these objects are deleted on export.
- **Export 'Modified by'.** Enables the export of information about who last changed an object (attribute 'modified by'). This information could conflict with data privacy rules and is therefore configurable.

Custom Library

In the *Used Library* field, the user can specify the additional global library to be displayed when assigning a type.

More information about creating a custom global library is available in section 4.5.2: Custom Global Library.

7.2 Window Layout

Here the user has some options for customising the default appearance and layout of the windows.

Tabbed Document Interface

The graphical pages are displayed as tabbed documents, with the tabs by default at the top of each page.

- **Show tab icons**: icons can be shown on the tabs, to help the user distinguish between different pages. With the option turned off, the tabs take up less space.
- **Show confirmation dialog when closing diagrams**: Users may prefer not to have the confirmation dialog. Closed (but not deleted) diagrams can of course be re-opened.
- **Tab position**: If the user prefers, the tabs can be shown at the bottom of the graphics.

Drawing Toolbox

Here the user can select whether group headers and/or element labels in the drawing toolbox should be shown or not. These options can also be changed within the toolbox itself.

7.3 Graphic Windows Settings

7.3.1 General tab

Open graphics board on study case activation

Causes the graphics windows to re-appear automatically when a study case is activated. When not checked, the graphics window must be opened manually via *Window* → *Graphic Board*.

Grid representation

The style and colour options allow the user to configure the appearance of the background grid, which is visible when freeze mode is turned off. The colour can be set back to default using a button at the bottom of the page.

Mark in Graphic

- The **colour** and **opacity** used when the objects are marked in the graphics can be defined.
- **Highlight small elements using additional markers**. Sometimes small objects such as terminals are hard to spot even when highlighted. If this option is selected, the position of such objects will be indicated using a marker.
- **Zoom in on marked elements**. If this is selected, the graphic where the object is to be shown will be zoomed in so that the object can be more easily seen. The level of zoom for both schematic and geographic diagrams can be configured by the user.

Background colours

- **Window**: The background colour for graphic windows can be configured.
- **Graphic page**: The background colour for network graphic diagrams can be configured.

The colours can be set back to default using a button at the bottom of the page.

Limit number of open site and substation diagrams

The user may set a limit to restrict the number of open graphic pages.

7.3.2 Advanced tab

Cursor

Defines the cursor shape:

- **Arrow.** A normal, arrow shaped cursor.
- **Tracking Cross.** A small cross.

Acceleration of Zooming

The higher the Acceleration factor, the more zoom there will be for a given mouse operation.

Update Graphic while Simulation is running

The graphic will be updated during the simulation.

General Options

- **Show Text only if height will be least**: Text smaller than the selected size will not be shown
- **Diagram window margin**: The graphical pages are by default shown against a dark background, with a small margin. The size of that margin can be adjusted here
- **Open diagrams in freeze mode after project activation**: the default is that this is selected
- **Allow resizing of branch objects**: If the option is enabled, the user can left click an edge element within the single line graphic and then resize it.
- **Show "Edit Graphic Object" in context sensitive menu**: If the option is enabled, when the user right-clicks on an element within the single line graphic, the option "Edit Graphic Object" will be offered.
- **Snap textboxes**: By default, this option is not enabled, allowing the user to position text-boxes precisely. However, selecting the option makes it easier to align text boxes with each other.

7.4 Output Window Settings

Enable Message filter

When un-checking this box, the filter buttons are removed from the output window.

Displayed Messages

This is where the filters used in the output window are defined. This, however, can be directly done in the output window.

Message format

- **No date and time:** the messages in the output window will be printed without a time stamp.
- **Date and time to the second:** the date and time of the system up to the second will be shown in every line of the output window.
- **Date and time to the millisecond:** the date and time of the system up to the millisecond will be shown in every line of the output window.
- **Full object names:** when an object is printed, the complete name (including the location path) is printed.

Font

The font used in the output window is set by clicking the button **Font...**

Page Setup

If the user wishes to print out the contents of the output window, the button **Page Setup...** offers a range of settings which can be used to configure the output.

Show confirmation dialog before clearing messages

This option is normally checked, to avoid users accidentally losing messages that they need, but deselecting it allows users to clear the output more quickly.

7.5 Profile Settings

PowerFactory provides standard profiles which define the configurations of the toolbars seen by the users. It is also possible for the Administrator to set up additional profiles, in order to provide customisation for different users (see Section 6.7) for details.

Here, the user can select the required profile and see the configuration details.

7.6 Functions Settings

The functions settings page provides check boxes for the function modules that are accessible from the Data Manager or from the object edit dialogs. The user may choose to see only certain modules in order to “unclutter” dialogs.

This may also be used to protect data by allowing only certain calculation functionality to be seen by certain users. This is particularly useful in a multi-user environment or when inexperienced users utilise *PowerFactory*.

7.7 Editor Settings

The editor used for DPL scripts, DSL equations and, if selected, Python scripts, can be configured on this page.

Options

- **Enable Auto Indent.** Automatically indents the next line.
- **Enable Backspace at Start of Line.** Will not stop the backspace at the left-most position, but will continue at the end of the previous line.
- **View blanks and Tabs.** Shows these spaces.
- **Show Selection Margin.** Provides a column on the left side where bookmarks and other markings are shown.
- **Show line Numbers.** Shows line numbers.
- **Enable Autocomplete.** A list of possible functions will be shown when writing a word inside the editor.
- **Tab Size.** Defines the width of a single tab.

Tabs

Toggles between the use of standard tabs, or to insert spaces when the tab-key is used.

Colours... Pressing this button opens the *Editor colour settings* dialog from which it is possible to specify a different colour scheme for each programming language used in the software as well as a separate colour scheme for use in the descriptive text fields of PowerFactory database objects. Different colours can be assigned for each different predefined class of data. By default a preview mode is shown where the impact of the chosen colouring scheme on an editable sample of text or code can be seen. An overall summary of the selected colour schemes can be shown by pressing the *Overview* button and from this view, colouring schemes can also be adjusted as well as copied and pasted from one column (programming language) to another. The default colouring scheme can be restored for each programming language independently by selecting the *reset* button, when the relevant tab is selected.

Font... Aa This option can be used to specify the appearance of the text used within the PowerFactory editor. The font, font style, size, effects and writing system can all be defined.

7.8 Colours Settings

To make it easier for users to identify the different sources of data easily, background colouring of the data fields is used, both in the network model manager and in the element dialogs.

As can be seen in Figure 7.8.1, the user can select different colours. See Section 4.7.5 for information about configuring colours. In addition, because data might belong to more than one category (e.g. operation scenario data which also has associated characteristics), the user can set priorities according to which information is considered more important. A *Reset to default* button allows the user to reset all these settings to their default values.

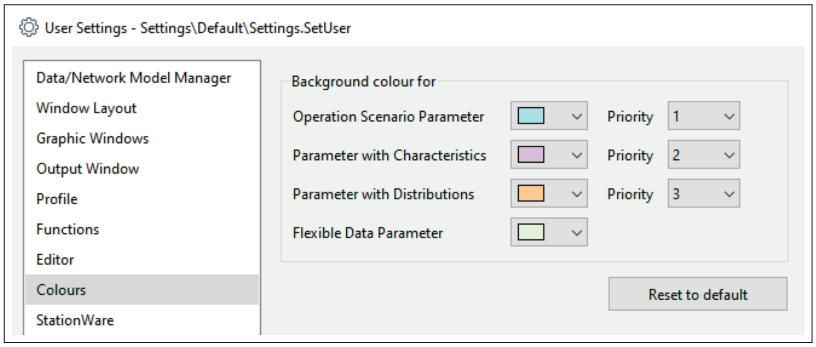


Figure 7.8.1: Data colouring options

7.9 StationWare Settings

When working with *DIgSILENTś StationWare* connection options are stored in the user settings.The connection options are as follows:

Service Endpoint

Denotes the *StationWare* server name. This name resembles a web page URL and must have the form:

- `https://the.server.name/psmsws/PSMSService.asmx` or
- `https://192.168.1.53/psmsws/PSMSService.asmx`

`https` denotes the protocol, `the.server.name` is the computer name (or DNS) of the server computer and `psmsws/PSMSService.asmx` is the name of the *StationWare* application.

Note: The default *StationWare* configuration requires SSL for the *StationWare* applications (web GUI and web services). Please use `http` instead of `https`, if SSL is not enabled for your *StationWare* applications.

Username/Password

Username and Password have to be valid user account in *StationWare*. A *StationWare* user account has nothing to do with the StationFactory user account. The very same *StationWare* account can be used by two different *PowerFactory* users.The privileges of the *StationWare* account actually restrict the functionality. For device import the user requires read-access rights. For exporting additionally write-access rights are required.

7.10 Offline Settings

These settings are only relevant if the installation has been configured to enable Offline mode to be used (see Section 5.5). Users will normally leave these settings at their default values.

- **Id contingent size.** It is necessary, when starting an Offline session, to reserve object ids in the main database so as to avoid any conflicts. This parameter specifies the number of ids to be reserved, and therefore the number of objects that could be created.
- **Id contingent warning threshold.** Once the user reaches this percentage of the above number of ids, a warning is issued. This can act as a prompt for timely resynchronisation of the user's changes.

7.11 Parallel Computing

The settings for parallel computing are centrally defined by the Administrator, as described in Section 22.4.

However, an individual user may wish to modify the settings and it is possible to do so on this page. A typical reason for this would be that the user wishes to make use of parallel computation but does not want to use the maximum allowed number of cores, because of a need to work on other applications outside *PowerFactory*at the same time. Here the user can opt to use fewer (but not more) cores than the maximum set by the Administrator.

On the Advanced tab, there are two more options:

- **Max. waiting time for process response**: The default setting here is 50s. A shorter time risks stopping processes which are still executing tasks, and a longer time could slow down the overall execution time. This setting should be changed only with care.
- **Transfer complete project to all processes**: The default is for this *not* to be selected. This is because normally *PowerFactory* transfers all data required. Only in exceptional circumstances should it be set, and the user should be aware that there is likely to be an adverse effect on performance.

7.12 Miscellaneous Settings

Localisation

- **Decimal Symbol.** Selects the symbol selected to be used for the decimal point.
- **Use operating system Format for Date and Time.** The operating system date and time settings are used when this is checked.

Retention of results after network change

When the option *Show last results* is selected, modifications to network data or switch status etc. will retained the results, these will be shown on the single line diagram and on flexible data pages in grey until the user reset the results (e.g. by selecting Reset Calculation, or conducting a new calculation).

Check for application updates

PowerFactory will remind the User if there are new updates available for the software. In this field is defined how often *PowerFactory* shall check for available updates. By default there will be a reminder every 14 days. The possible options are:

- **Manually:** the User will check for updates manually, no reminder will be shown.
- **On each application start:** a reminder will be shown every time *PowerFactory* is started.
- **According to interval:** a reminder will be shown according to the time defined in this field.

System Stage Profile

This setting relates to the old system stage system used before the current Variations were introduced. The options define the extent to which the user can create or modify the "old" stages.

Edit Filter before Execute

If this is selected, when the user uses a filter, a dialog box appears and the user may first change something or immediately press Apply; if this option is not checked then filters are just applied straightaway.

Allow housekeeping task to operate when user is connected

This option is only active if housekeeping is enabled.

Show 'Remove Contingencies' confirmation dialog in Contingency Analysis

When existing contingencies will be removed because they will be overwritten by new ones (e.g. when using the Contingency Definition tool), the default behaviour is to ask the user to confirm, in case of error. If the user prefers not to be asked, this option should be deselected.

Show 'Reset Calculation' confirmation dialog

This option can be deselected if the user does not want to be asked for confirmation when using the "Reset Calculation" button.

Show 'Example' dialog at startup

This option can be deselected if the user does not want to see the Example dialog at each log-on. It can also be deselected directly from the dialog itself.

Show 'Exit' dialog

When the user closes *PowerFactory* a confirmation dialog normally appears. In addition to confirming that the user wishes to exit, it offers options relating to project purging and recycle bin emptying. There is also a "Don't ask again" option which can be checked. This user setting is another way of disabling the confirmation dialog, or of course re-enabling it.

Show backup reminder dialog

If this option is set, the user will receive a reminder about making a database backup when logging in.

Confirm Delete Action

If this option is set, a confirmation dialog will appear whenever something is about to be deleted.

Part III

Handling

Chapter 8

Basic Project Definition

The basic database structure in *PowerFactory* and the data model used to define and study a power system is explained in Chapter 4 (*PowerFactory* Overview). It is recommended that users become familiar with this chapter before commencing project definition and analysis in *PowerFactory*. This chapter describes how to define and configure projects, and how to create grids.

8.1 Defining and Configuring a Project

There are three methods to create a new project. Two of them employ the Data Manager window and the third employs the main menu. Whichever method is used the end result will be the same: a new project in the database.

Method 1: Using the main menu:

- On the main menu choose *File* → *New*→ *Project*.
- Enter the name of the project. Make sure that the *Target Folder* is set to the folder in which the project should be created. By default it is set to the active user account folder.
- Press **Execute**.

Method 2: Using the element selection dialog from the Data Manager:

- In the Data Manager click on the *New Object* button ()
- In the field at the bottom of the Element Selection window (*IntPrj*) (after selecting option *Others* in the *Elements* field). Note that names in *PowerFactory* are case-sensitive.
- Press **Ok**. The project folder dialog will then open. Press **Ok**.

Method 3: Directly via the Data Manager:

- Locate the active user in the left-hand pane of the Data Manager.
- Place the cursor on the active user's icon or a folder within the active user account and right-click.
- From the context-sensitive menu choose *New* → *Project*. Press **Ok**. The project folder dialog will then open. Press **Ok**.

Note: The *ComNew* command is used to create objects of several classes. To create a new project it must be ensured that the *Project* option is selected.

In order to define and analyse a power system, a project must contain at least one grid and one study case. After the new project is created (by any of the methods described), a new study case is automatically created and activated. A dialog used to specify the name and nominal frequency of a new, automatically-created grid pops up. When the button **OK** is pressed in the grid dialog:

- The new grid folder is created in the newly-created project folder.
- An empty single line diagram associated with the grid is opened.

The newly-created project has the default folder structure shown in Figure 8.1.1. Although a grid folder and a study case are enough to define a system and perform calculations, the new project may be expanded by creating library folders, extra grids, Variations, Operation Scenarios, Operational Data objects, extra study cases, graphic windows, etc.

Projects can be deleted by right-clicking on the project name in the Data Manager and selecting *Delete* from the context-sensitive menu. Only inactive projects can be deleted.

Note: The default structure of the project folder is arranged to take advantage of the data model structure and the user is therefore advised to adhere to it. Experienced users may prefer to create, within certain limits, their own project structure for specific advanced studies.

If the user wishes to change the default structure of the project, it can be modified from the Administrator account. The default structure is defined in a project held the folder: System, Configuration, Default.

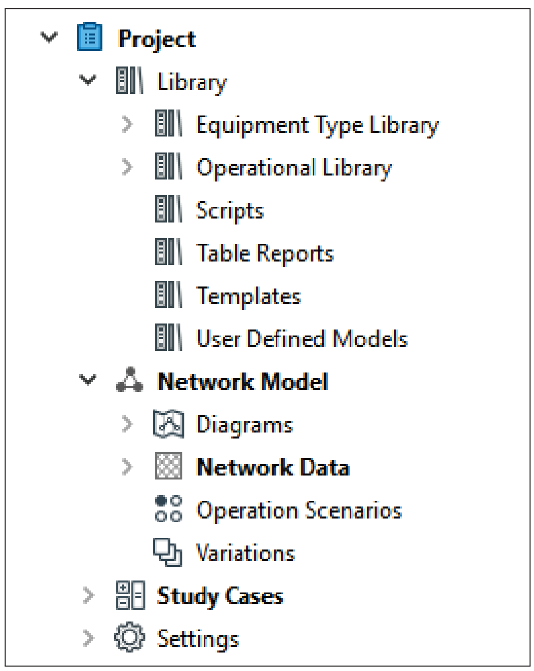


Figure 8.1.1: Default project structure

8.1.1 Project Dialog

The project (*IntPrj*) dialog can be accessed by selecting *Edit* → *Project Data*→ *Project...* on the main menu or by right-clicking the project folder in the Data Manager and selecting *Edit* from the context-sensitive menu.

The *Basic Data* page contains basic project settings and allows the creation of new study cases and grids:

- Pressing the → button in the *Project Settings* field opens the Project Settings dialog (*SetProj*). See section 8.1.2 for more information regarding the settings of the project.
- Pressing the **New Grid** button will create a new grid and will open the grid edit dialog. A second dialog will ask for the study case to which the new grid folder should be added. For additional information about creating a new grid refer to Section 8.2 (Creating New Grids)
- The **New Study Case** button will create a new study case and will open its dialog. The new study case will not be activated automatically. For further information about creating study cases refer to Chapter 13: Study Cases, Section 13.2 (Creating and Using Study Cases).
- When a project is created, its settings (i.e. result box definitions, report definitions, flexible page selectors, etc.) are defined by the default settings from the system library. If these settings are changed, the changes are stored in the folder "Settings" of the project. The settings from another project or the original (default) ones can be taken by using the buttons **Take from existing project** or **Set to default** in the *Changed Settings* field of the dialog. The settings can only be changed when a project is inactive.
- The button **Calculate** in the *Licence Relevant Nodes* field, calculates the number of nodes relevant to the *PowerFactory* licence, this number is the number of equipotential nodes on the network.
- The name of the active study case is shown in the lower part of the dialog window under *Active Study Case*. Its dialog can be opened by pressing the → button.
- Pressing the **Contents** button on the dialog will open a new data browser displaying all the folders included in the current project directory.

The *Sharing* page of the dialog allows the definition of the project sharing rules. These rules are particularly useful when working in a multi-user database environment. Further information is given in Chapter 21 (Data Management).

The *Derived Project* page provides information if the project is derived from a master project.

The *Combined Project* panel enables the user to combine additional projects with the current project if it is active. See section 21.7.1.3 for more details.

The *Storage* page provides information about the records stored in the project. A *PowerFactory* project contains records of changes to data, which makes it possible to roll back the project to an earlier state using versions (see section 21.2). However, as the user works with the project and makes changes to it, the number of records increases and it is useful to remove older, unwanted records in a process known as "purging". By default all changes within the last 7 days will be retained.

The *Migration* page provides information about the migration status of the project. The significance of the information displayed on the *Migration* page is described as follows:

- **Migration status:** this staus flag provides an indication to the user as to whether the project has already migrated to the relevant *PowerFactory* major version.
- **Migration Priority:** the migration priority of the project can be set on this page. This priority is used when using the *Minimal* migration option as described in section 5.4.3.1.
- **Project ID, Object ID and Timestamp ID:** these IDs represent the respective database IDs and would be relevant for the user while raising a Support request for Migration related queries.

- **Migrated to Build:** this represents the last build ID of the database currently being used to which the project is migrated. This information can be provided while raising Migration related Support queries.
- **Migrate Variations and Operation Scenarios folder:** applying this option will update the project's folder structure to match the latest default structure. The basic project structure was modified with *PowerFactory* version 2020 so that the *Variations* and *Operation Scenarios* folders are now situated directly under the *Network Model* folder. Existing projects are not migrated to the new structure by default when databases are migrated to the new structure, as this could cause problems with user-defined scripts. Instead, a dedicated option to "Migrate Variations and Operational Scenario folder" is provided on the *Migration* page in order for the user to manually adapt the project structure.
- **Down Migration of Plot Pages:** since *PowerFactory* 2021, a new plot framework has been introduced. However, if there is a requirement to export a project consisting of plots based on the new framework and to subsequently import it in a *PowerFactory* version which only supports the old plot framework, then the user can choose to migrate down the plot pages to the old framework using **Migrate down** before exporting the project. This operation will create additional plot pages in the project based on the old framework. Once the project has been exported then there is a possibility for the user to purge these additionally created plot pages using the **Purge down migrated pages** option.

The *Description* page is used to add user comments and the approval status.

8.1.2 Project Settings

The project settings dialog (*SetPrj*) can be accessed by selecting *Edit* → *Project Data*→ *Project Settings...* on the main menu or by pressing the → button in the *Project Settings* field of the project's dialog.

8.1.2.1 Validity Period

PowerFactory projects may span a period of months or even years, taking into account network expansions, planned outages and other system events. The period of validity of a project specifies the time span the network model is valid for.

The validity period is defined by the *Start Time* and *End Time* of the project. The study case has study time, which must fall inside the validity period of the project.

Start Time: Start of validity period.

End Time: End of validity period.

Status: This flag enables the user to label a project as Draft or Issued. It does not have any effect on functionality.

8.1.2.2 Formats and Units

On this page, units for the data entry and display of variables can be selected.

Input Variables

- **Units:** this parameter is used to change the system of measurement used when displaying element parameters. By default, results are entered and displayed in metric (SI) units. British Imperial units can be used instead, by selecting one of the options English-Transmission or English-Industry. The Transmission option uses larger units where appropriate (i.e. miles rather than feet for line length).
- **Lines/Cables Length unit, m:** for metric length units, this parameter allows the user to select the preferred prefix (such as k to use kilometers rather than meters).
- **Loads/Asyn. Machines P,Q, S unit VA, W, var:** this parameter allows the user to select the preferred prefix (such as M to use megawatts rather than watts).
- **Static Generators/Synchr. Machines P,Q, S unit VA, W, var:** this parameter allows the user to select the preferred prefix (such as M to use megawatts rather than watts).
- **Currency Unit:** for displaying values relevant for cost-related calculations, this parameter allows selection from a range of currency units (abbreviated in accordance with ISO 4217).

Output Variables

Use the drop-down menus to adjust the preferred prefix for the output variables. It is possible to define different settings for *Load Flow and Simulation* and for *Short-Circuit* calculation types.

The number of decimals can be defined independently for each of the following output variable:

- Voltage
- Current
- Power
- Percent (%)

- Degree
- Per unit (p.u.)
- Other units

8.1.2.3 Calculation Options

On the *Calculation Options* page, additional parameters used during the calculation are defined. The following options are available:

General

- **Base Apparent Power:** this is the value used during the calculation; the base power of each element is defined by its rated value.
- **Min. voltage for HV voltage level:** this describes the voltage threshold for network elements to be considered as high voltage elements. This information is used in various commands when defining acceptable load flow errors.
- **Min. voltage for MV voltage level:** this describes the voltage threshold for network elements to be considered as medium voltage elements. The upper range is defined by the threshold given in Min. voltage for HV voltage level. This information is used in various commands when defining acceptable load flow errors.
- **Min. Resistance, Min. Conductance:** the minimum resistance and conductance that will be assigned to the elements if none is defined.
- **Threshold Impedance for Z-model:** this parameter is used to control the modelling of currents (Y- and Z-models are used internally and this parameter controls the small-impedance threshold for swopping from one to the other). This control is designed to enhance the robustness of the algorithm and the user is recommended to leave the parameter at its default setting.

Topology

- **Settings for slack assignment:** this option only influences the automatic slack assignment (e.g. if no machine, or more than one machine, is marked as "Reference Machine")
  - **Auto slack assignment**:
    - **Method 1**: all synchronous machines can be selected as slack (reference machine);
    - **Method 2**: a synchronous machine is not automatically selected as slack if, for that machine, the option on its *Load Flow* page: *Spinning if circuit-breaker is open* is disabled.
    - **Off**: auto slack assignment is switched off; the grid will be considered as de-energised if no reference machine is defined.
  - **Priority for Reference Machine**: The criteria used for automatic selection of a reference machine are described under the Load Flow options, in section 25.3.2. However, the way in which the machines are prioritised for selection can be influenced using this setting. The options are:
    - **Rated Power**: Snom is used as the criterion, as described in page 472
    - **Active Power Capability**: Pmax-Pmin is used instead of Snom
    - **Active Power Reserve**: Pmax-Pgini is used instead of Snom
  - **Auto slack assignment for areas without connection to fictitious border grid**: For large network models covering multiple systems connected by so-called "fictitious border grids", this option can be used in conjunction with a *Fictitious border grid* flag on the *ElmNet* object, to have more control over which areas will be considered in the calculation, so that remote parts of the network that are not of interest for a particular study can be excluded. The default is for this option to be active.
- **Automatic Out of Service Detection:** when calculations are executed, if the *Automatic Out of Service Detection* parameter is selected, the calculation will treat de-energised elements as

though they have been made Out of Service (using the Out of Service flag). This means that they will not be considered in the calculations and no results will be displayed for them. It should be noted that this parameter does not affect elements which are isolated only by open circuit breakers (as opposed to other switch types). These are retained in the calculation because they could be energised via switch close actions at some point.

- **Determination of supplying transformers:** for a distribution network, this flag alters the way in which calculations determine which elements are supplied by which substations. If only voltage controlling transformers are considered, step-up transformers will not be considered as supplying transformers. This affects some calculations such as reliability analysis and the colouring mode *Topology*, *Supplied by Substation*.

Lines

- **Calculation of symmetrical components for untransposed lines:** the selection of one of these methods defines how the sequence components of lines in *PowerFactory* will be calculated:
  - **Method 1**: apply the 012-transformation (irrespective of line transposition). This is the standard method used;
  - **Method 2**: first calculate a symmetrical transposition for untransposed lines, and then apply the 012-transformation.
- **Earth wire reduction of towers:** when overhead lines are modelled using towers, the earth-wires are reduced before matrix transposition is carried out. In older versions the matrix was transposed first, so this setting allows users to use the older method if this is required for compatibility reasons.
- **Consider line compensation current for line loading calculation:** If selected, the total line current (sum of the line current plus the line compensation current) is considered when calculating the line loading. Otherwise, only the line current (without the line compensation current) is used to calculate the line loading.

8.1.2.4 Graphic

On the *Graphic* page, several parameters relevant for the graphical interface can be adjusted. The following options are available:

General

- **When connecting component to busbar**
  - **Circuit-Breaker:** when a component is connected to a busbar in a single line graphic, a switch (class *StaSwitch*) is automatically created. By default this switch will be a circuit breaker.
  - **Switch type depending on nominal voltage:** if this option is selected instead, there is the option to have different switch types created for two voltage levels (e.g. low and high voltage systems), with the threshold specified by the user.
- **Variations**
  - **Show inactive elements from other variations:** by default, inactive elements (for example elements created in a expansion stage which is not yet active) are displayed on network diagrams if they are not in freeze mode. They are shown in the colour selected on the first option of the colouring scheme, Energising Status, even if that option is deselected. If the *Show inactive elements from other variations* project setting is deselected, inactive elements will not be visible.
- **Insertion of new substations**
  - **Insert substations with bays:** when new substations are created, it is possible to include Bay elements (*ElmBay*). These group together the network elements that normally constitute a standard bay connection of a circuit to a busbar within the substation. This grouping is useful for visualisation but is also used by the Load Flow Calculation option *Calculate max. current at busbars*: see section 25.3.3.

Text Boxes

Customise fonts used for Single Line and Block diagrams. Text boxes used for *Labels* and *Results*

- **Default fonts for Single Line Diagrams**: to customise fonts for *Labels*, *Results* and *Title and Legends*.
- **Default fonts for Block Diagrams**: to customise fonts for *Blocks/slots*, *Signals* and *Title and Legends*.
- **Show jump-to labels at graphically half-connected lines**: to show additional labels for connections going from one diagram into another diagram.

Plots

- **Use new plot framework:** to enable the use of the new plot framework (e.g. class types GrpPage, PltLinebarplot, Plt) introduced with PF2021. By default, this option is active. The older plot framework can be used by deactivating this option.
- **Default colour palette:** to select a different colour palette than the default provided by *PowerFactory*.
- **Default plot style:** to select a different plot style than the default provided by *PowerFactory*.
- **Names in plot legends:** for element names shown in plot legends, the user has the option to include additional information to show where the element is located, e.g. site or substation (short name).

Geographic

- **Coordinate system:** the setting determines in which coordinate space the geographic coordinates of net elements in the project are stored/interpreted. The user can select from a range of coordinate systems (identified by their EPSG codes).
- **WGS-84 bounds:** these fields show the bounds of the selected system in WGS-84 coordinates.

8.1.2.5 Miscellaneous

Switching Actions

- **Isolate with earthing:** if this option is selected, when a planned outage is applied, the equipment will not only be isolated but earths will be applied, for example on the terminals at either end of a line. Similarly, when using the “right-click, isolate” option, earths will be applied.
- **Isolate opens circuit-breakers only:** when equipment is isolated using a planned outage or “right-click, isolate”, this option determines whether it is switched out using just circuit breakers or whether isolators adjacent to the breakers are also opened.

Planned Outages

- **Consider automatically upon study case activation:** if this option is selected, relevant *IntPlannedout* outages are automatically applied when a study case is activated.

Creation of Planned Outages

Outages can be represented using *IntPlannedout* objects (see chapter 42), and the default is that any new planned outage created will be of this class. If the user wishes instead to create the older *IntOutage* objects, this project setting should be changed to “Create *IntOutage* (obsolete)”.

8.1.2.6 External Data

External Data Directory: this is a folder on the user's hard disk, in which data files related to the *PowerFactory* project, such as images or Python scripts, are stored. The filepath has to be absolute, but can contain the placeholders shown on the dialog box:

- "$(InstallationDir)" for the installation directory of *PowerFactory*;
- "$(WorkspaceDir)" for the workspace directory.

Once defined, the external Data Directory can be used to link files to *PowerFactory* objects by the following abbreviation:

- "$(ExtDataDir)"

Check existence of referenced files: if this button is clicked, *PowerFactory* checks to see that all external files referenced in the project exist and reports those not found.

Display reminder after file selection: for certain file reference parameters, it is expected that the file will be in the External Data area and, if this check box is enabled, a dialog box will come up if a different location is specified. Note that it is permitted to select files outside the External Data directory; the dialog is just a reminder that the External Data directory is recommended.

8.1.2.7 Data Verification

Nominal Voltage Check

- **Max. allowed difference over Lines/Switches/Fuses:** these are the maximum permitted percentage differences between the nominal voltages at the two ends of a line or the two sides of a switch/fuse (as a percentage of the higher voltage). With differences above these thresholds, a load flow will fail with an error message; for smaller differences (but > 0,5%) a warning is given.
- **Max. allowed deviation from Terminal Voltage, for transformers and for other elements:** there is a check when running a load flow that the rated voltage of a transformer or other element is not too low compared with the nominal voltage of the terminal to which it is connected. With differences above these thresholds, a load flow will fail with an error message; for smaller differences (but > 10%) a warning is given.

Line couplings

- **Allowable difference in lenghts of lines:** this is the maximum permitted percentage difference in the length of those lines which are part of the same line coupling.

8.1.2.8 Substation/Site Types

Includes a list for substation and site types, configurable by the user, which can be used in geographic diagrams to distinguish graphically between different types of substation/sites by assigning different symbols to each type in the list. In a substation itself, the Substation Type is selected from the same list, in the Description page.

8.1.3 Activating and Deactivating Projects

To activate a project use the option *File* → *Activate Project* from the main menu. This shows a tree with all the projects in the current user's account. Select the project that should be activated. Alternatively, a project may be activated by right-clicking on it in the Data Manager and using the context-sensitive menu.

The last 5 active projects are listed under *File* in the main menu. The currently active project is the first entry in this list. To deactivate the currently active project, select it in the list. Alternatively, you may choose the option *File* → *Deactivate Project* from the main menu. To activate another project, select it in the list of the 5 last active projects.

Upon project activation, the user may see a message about project purging. The purge function is described above in section 8.1.1, *Storage* page.

Note: Only one project can be activated at a time.

8.1.4 Exporting and Importing Projects

Projects (or any folder in the database) can be exported using the *.pfd (*PowerFactory* Data) file format, or by exception for older applications by using the *.dz format. It is recommended to use the PFD format (*.pfd) whenever possible when exporting projects: the consumption of memory resources is significantly lower than with the old file format (*.dz) and functions such as historic data, time stamps and former versions are not supported by the old *.dz format.

A new project export method, the Snapshot Export, has been made available from Version 2017. This method, described in section 8.1.4.2, creates a file in a *.dzs format.

8.1.4.1 Exporting a Project

To export a project, select *File* → *Export...* → *Data...* from the main menu or click on the icon of the Data Manager. Alternatively projects can be exported by selecting the option *Export...* on the project context-sensitive menu (only available for inactive projects). The dialog is shown in Figure 8.1.2.

- **Objects to export:** this table shows all top-level objects that will be exported within the *.pfd file.
- **Export current state:** this option is visible if the project (or, object in the *Objects to export* table) has Versions defined. If enabled (default), the current state of the project will be exported. Otherwise, only the state of the selected Version/s will be exported.
- **Versions to export:** this table shows all Versions of the *Objects to export*, if any are available. By disabling the checkbox for specific Versions, the user can define which Version should or should not be exported. For master projects, the corresponding Version for the derived project must be selected. See Section 21.3.1 for further details.
- **Export data in retention period:** if enabled, data changes from within the retention period will be exported. See Section 8.1.1 for further details.
- **Export 'Modified by':** if enabled, the information who last changed an object is exported (attribute 'modified by'). This information could conflict with data privacy rules and is therefore configurable.
- **Export external data files (e.g. results files):** if enabled, calculation results (i.e. results files, plot data, etc.) will be exported. Otherwise, the calculation must be repeated after importing.
- **Export derived project as regular project:** this option is only available for derived projects, see Section 21.3.1. If enabled, a derived project will be exported as an 'adequate' project. In this case no master project is required. It should be noted that this project can no longer be reused as a derived project.
- **Export to former *PowerFactory* version:** if the project is intended to be imported into a former *PowerFactory* version, this flag must be activated and the version specified.
- **PFD file:** the path where the *.pfd file will be saved.

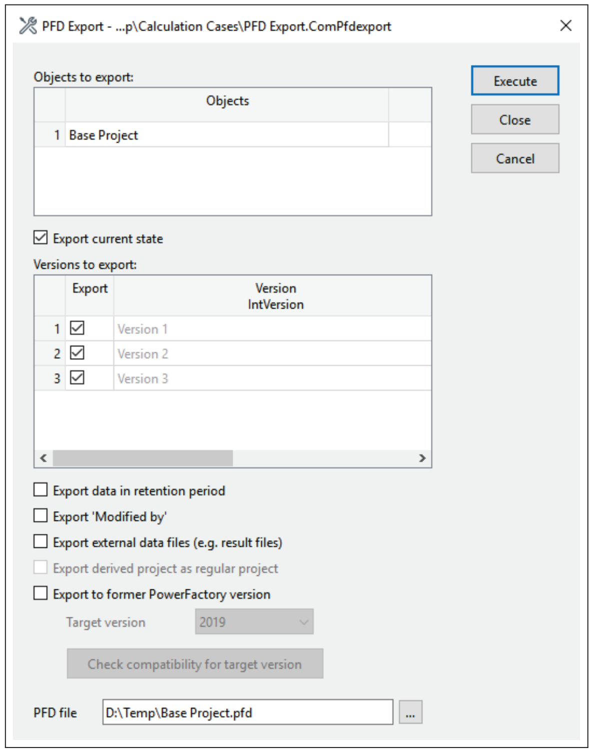


Figure 8.1.2: Export dialog

8.1.4.2 Snapshot Export

The Snapshot Export function enables the currently active status of a project to be exported, such that only the relevant objects are included. A project exported in this way is potentially a much smaller file, which nevertheless when reimported into *PowerFactory* can be used to reproduce analysis carried out in the original project study case.

Unlike the existing Project Export, where the project must first be deactivated, the Snapshot Export is performed on an active project. This way, *PowerFactory* can determine exactly which objects are active and which data are applicable as a result of an active scenario or active variations.

To carry out a snapshot export from a project, the required study case and scenario (if used) should be activated. Then *File* → *Export*→ *Project Snapshot (*.dzs)...* from the main menu is selected.

When the Snapshot Export is executed, the resulting file outside *PowerFactory* has the file extension .dzs. It can be imported just like a .pfd or .dz file and when activated can be used to perform the usual calculations such as load flow or simulations. Furthermore, it is possible for merge processes to be carried out between it and the source project, for example if there is a need to include additional data from the source project.

The Snapshot Export captures only the data required to reproduce the results of the active study case. Therefore the following objects, for example, will not appear in the resultant project:

- **Variations:** changes in active variation stages are consolidated. The exported project will contain therefore no variations.
- **Inactive study cases, scenarios and grids:** inactive study cases, scenarios and grids are not exported. The exported project will have one study case, and no scenarios; if a scenario had been active in the source project, the data will be represented in the network data.
- **Unused library objects:** only objects which are in use are exported, so unused information such

as type data which are not referenced will not be exported.

- **Characteristics:** the parameters which are modified by Characteristics will be set at the values determined by the Characteristics, but the Characteristics themselves will not be exported.
- **Operational Library:** operational data such as Thermal Ratings, which may contain variations, will be reduced to just the currently active values.

8.1.4.3 Importing a Project

There are several options to import projects:

- By selecting *File* → *Import...* → *Data...* from the main menu.
- By clicking on the icon in the Data Manager.
- By selecting *Import...* on the project context-sensitive menu (only available for inactive projects).
- *Drag and Drop* the **.pfd* file onto the *PowerFactory* GUI.

The user can select the file type to be imported from the drop-down menu in the Open dialog which pops up. Only when using the *Drag and Drop* option for the import, the type of file is automatically selected by *PowerFactory*.

In the PFD import dialog, the user can select the import path location. By default, it is the current location in the Data Manager. Moreover, the option *Activate project after import* automatically activates the project after importing it.

The import and export of information in other data formats is described in Chapter 24.

8.1.5 External References

In order to avoid problems when exporting/importing projects, it is recommended to check for external references before exporting the project. This can be done via the project context-sensitive menu, selecting the *External References* sub-menu and then the option *Check for External References*. The user can then select the External Locations, such as the Global Library or the "Configuration" folder. After pressing **OK**, a list of external project references is displayed in the output window.

If external references are found, these can be packed before the project is exported. This can be done through the context-sensitive menu of the project, by accessing the *External References* sub-menu and selecting the option *Pack External References*. Similar to the option described above, the user can select the External Locations. By pressing **OK**, a new folder in the project structure called "External" is created, which contains all formerly external objects.

The *Clear Unresolvable References* option allows the user to remove external references to objects that cannot be found, for example because they have been deleted or archived. It is important to mention that this option does not solve problems associated with incomplete data (e.g. no type found).

Note: The options described above are only selectable after the project has been deactivated.

To review the use of type objects, standard models, etc. from the *DIgSILENT* Library (and other external libraries) in a particular project, the option *Show External Types* from the *External References* sub-menu can be used. This generates a report in the Output Window, containing those types that are not stored inside the project library. Objects from external libraries can be opened up from the list in the Output Window. In addition, for each item listed, the number of objects using it is given. This acts as a hyperlink which can be used to bring up a list of objects in the project referencing this item.

As explained in Section 4.5.1.1, all the main objects in the global libraries are managed using versions. Via the option *Show Updates for External Types* from the *External References* sub-menu, the user can check whether later versions are available, and see what has been changed in the later versions.

8.1.6 Including Additional Documents

It may be useful sometimes to provide additional information such as a detailed documentation about a network element, to the user of a project. Such information can be provided by creating an *IntDocument* object within the project, for example within the project library. This is done by using the *New Object* button ( ), and selecting “Other Elements (Int*)” and then entering “IntDocument” in the Element field.

In the *IntDocument* object, there is a Filename field, which is populated by selecting the file location. This is sufficient to provide access to a document which is outside the *PowerFactory* database, but there is also an *Import* option, which enables the user to import the document itself into the *PowerFactory* database.

Such *IntDocument* objects (whether simply links or containing the actual document) can also be created elsewhere in the database, for example in the Configuration area. Network elements have a field called “Additional Data” on the description page, which is a convenient place to hold a reference to an *IntDocument* object.

8.2 Creating New Grids

When defining a new project a grid is automatically created. In case additional grids are required, various methods may be employed to add a grid folder to the current network model:

1. Open the edit dialog of the project and press the **New Grid** button.
2. Select *Insert* → *Grid...* on the main menu.
3. Right-click the Network Data folder (in the active project) in a Data Manager window and select *New* → *Grid* from the context-sensitive menu.

The dialog to create a new grid will then pop up. There the grid name, the nominal frequency and a grid owner (optional) may be specified. A second dialog will appear after the **Ok** button has been pressed. In this dialog the study case that the grid will be linked to must be selected. Three options are given:

1. **add this Grid/System Stage to active Study Case:** only available when a study case is active.
2. **activate a new Study Case and add this Grid/System Stage:** creates and activates a new study case for the new grid.
3. **activate an existing Study Case and add this Grid/System Stage:** add the new grid folder to an existing, but not yet active study case.

After the **Ok** button in the second dialog has been pressed, the new grid is created in the Network Data folder and a reference in the Summary Grid object of the selected study case is created. Normally, the second option (from the list above) is preferred because this creates a new study case which is dedicated to the new grid only. This means that the new grid may be tested separately using a load flow or other calculation. To analyse the combination of two or more grids, new study cases may be created later, or the existing ones may be altered.

As indicated in Chapter 13 (Study Cases), grids can be later added or removed from the active study case by right-clicking and selecting *Activate/Deactivate*.

8.3 Project Overview

The Project Overview is illustrated in Figure 8.3.1. It is a dockable window, displayed by default on the left side of the main application window between the main toolbar and the output window. It displays an overview of the project allowing the user to assess the state of the project at a glance and facilitates easy interaction with the project data. The window is docked by default, but can be undocked by the user and displayed as a floating window that can be placed both inside and outside of the main application window.

To dock an undocked Project Overview window, it should be moved all the way to the left border of the main window, until the mouse reaches the border.

If required, the window can be closed by the user. To close or reopen the window, toggle the option *Window* → *Project Overview* from the main menu.

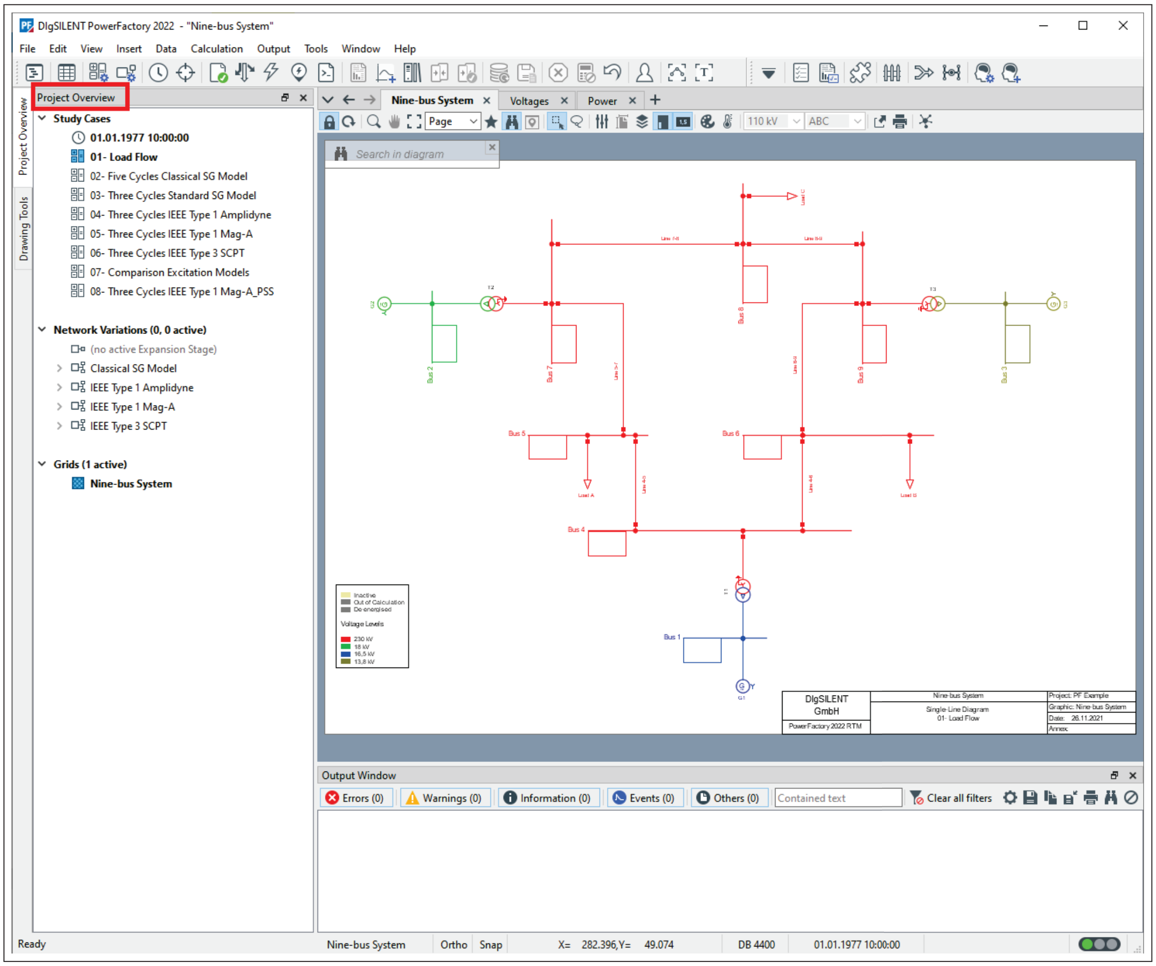


Figure 8.3.1: Project overview

The following objects and information can be accessed via the Project Overview.

- Study Cases
  - Active Study Case
  - Inactive Study Cases
  - Current Study Time

- Operation Scenarios
  - Active Scenario Schedulers
  - Active Scenarios
  - Inactive Scenarios
- Variations
  - Recording Expansion Stage
  - List of active Variations with active and inactive Expansion Stages as children
  - List of inactive Variations with inactive Expansion Stages as children
- Grid/System Stages
  - List of active Grids or System Stages
  - List of inactive Grids or System Stages
- Trigger
  - Active triggers

Entries for active objects are displayed with bold text, entries for inactive objects are displayed as disabled/grey.

A context-sensitive menu can be accessed by right-clicking on each of the tree entries. The following actions are available for each of the entries:

- Change active item(s): Activate, Deactivate, change active
- Show all available items
- Edit (open dialog)
- Edit and Browse
- Delete
- Save (for Operation Scenario only)

Chapter 9

Network Graphics

9.1 Introduction

PowerFactory works with three different classes of graphics which constitute the main tools used to design new power systems, controller block diagrams and displays of results:

- Single Line Diagrams (described in this chapter)
- Block Diagrams (described in Sections 30.2 and 30.6)
- Plots (described in Section 19.7: Plots)

Diagrams are organised in Graphic Boards for visualisation (see Section 9.2.2 for more information).

9.2 Graphic Windows and Database Objects

In the *PowerFactory* graphic windows, graphic objects associated with the active study case are displayed. Those graphics include single line diagrams, station diagrams, block diagrams and plots. Many commands and tools are available to edit and manipulate symbols in the graphics. The underlying data objects may also be accessed and edited from the graphics, and calculation results may be displayed and configured.

Many of the tools and commands are found in the drop down menus or as buttons in the toolbars, but by far the most convenient manner of accessing them is to use the right mouse button to display a “context menu”. The menu presented is determined primarily from the cursor position.

9.2.1 Network Diagrams and other graphics

Four types of graphical pages are used in *PowerFactory*:

1. Single Line Diagrams (schematic and geographical network diagrams) for entering power grid definitions and for showing calculation results.
2. Detailed graphics of sites, substations or branches (similar to network diagrams) for showing busbar (nodes) topologies and calculation results.
3. Block Diagrams for designing logic (controller) circuits and relays.
4. Plot Pages for designing graphs, e.g. for the results of a time domain simulation.

The icon *Diagrams* () can be found inside the Data Manager. Grids, substations, branches, sites and controller types (*DSL Model Type* and *Composite Model Frames* in *PowerFactory* terminology) each have a graphical page. In order to see the graphic on the screen, open a Data Manager and locate the graphic page object you want to show, click on the icon next to it, right-click and select *Show Graphic*. The *Show Graphic* option is also available directly by right-clicking on the object. The graphic pages of grids and substations are to be found in the subfolder *Diagrams* () under the *Network Model* folder.

Note that it is also possible to store Diagrams within the Grid, although this is generally not recommended.

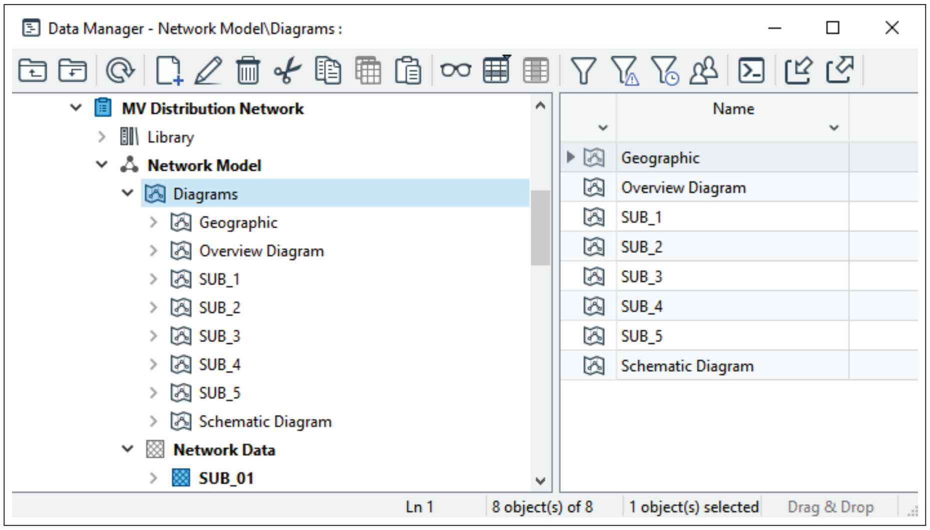


Figure 9.2.1: The Diagrams folder inside the Data Manager

9.2.2 Active Graphics, Graphics Board and Study Cases

The graphics that are displayed in an active project are determined by the active study case. The study case folder contains a folder called the *Graphics Board* folder (*SetDesktop*) in which references to the graphics to be displayed are contained. The Graphics Board folder is automatically created and maintained and should generally not be edited by the user.

Consider the project shown in figure 9.2.2. There are several diagrams in the *Diagrams* folder, but the graphics board folder in the study case has a reference to only some of them and thus only these graphics will be shown when the study case is activated.

The page tab, which is normally at the top of the graphic, offers options for handling the graphics in the graphics board. See section 9.2.5 for details.

The study case and graphics board folder will also contain references to any other graphics that have been created when the study case is active, such as plot pages.

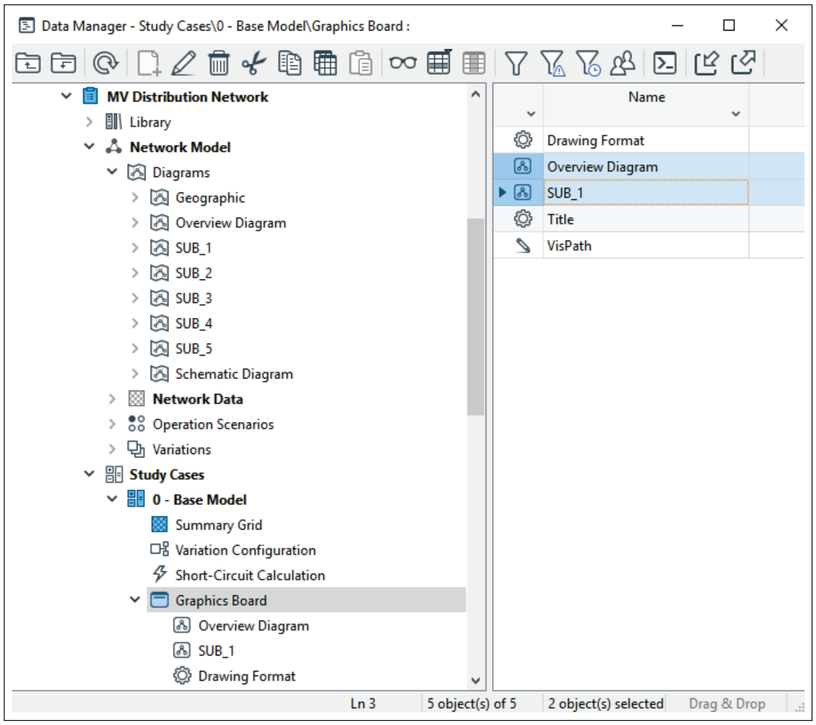


Figure 9.2.2: Relationship between the study case, graphics board and single line diagrams

9.2.3 Single Line Graphics and Data Objects

When building a new network, it is usually recommended that this is done from a single-line diagram. As each objects is created, the associated graphical representation is created too. For more information about building a network, see Chapter 11.

In a simple network there may be a 1:1 relationship between data objects and their graphical representations, i.e. every load, generator, terminal and line is represented once in the graphics. However, *PowerFactory* provides additional flexibility in this regard. Data objects may be represented graphically on more than one graphic, but only once per graphic. Thus a data object for one terminal can be represented graphically on more than one graphic. All graphical representations contain the link to the same data object.

Furthermore, graphical symbols may be moved without losing the link to the data object they represent. Likewise, data objects may be moved without affecting the graphic.

The graphics themselves are saved in the database tree, by default in the Diagrams folder of the Network Model. This simplifies finding the correct Single Line graphic representation of a particular grid, even in the case where there are several graphic representations for one grid.

When the drawing tools are used to place a new component (i.e. a line, transformer, etc.) a new data object is also created in the database tree. A Single Line Graphic object therefore has a reference to a grid folder. The new data objects are stored into the 'target' folders that the graphics page are associated with. This information may be determined by right-clicking the graphic → *Graphic Options* or using the *Graphic Options* button ().

Since data objects may have more than one graphic representation the deletion of a graphic object should not mean that the data object will also be deleted. Hence the user may choose to delete only the graphical object (right-click menu → *Delete Graphical Object only*). In this case the user is warned that the data object will not be deleted. This suggests that a user may delete all graphical objects related to a data object, with the data object still residing in the database and being considered for calculations.

When an element is deleted completely (right menu option → *Delete Element*) a warning message will confirm the action. This warning may be switched off in the User Settings dialog, *General* page, *Always confirm deletion of Grid Data*.

9.2.4 Creating New Graphic Windows

A new graphic window can be created by using *Insert* on the main menu. Four options relating to graphics are offered:

- ***Single Line Diagram***: Creates a Single Line Diagram graphic in the target folder. Before the graphic is inserted, the user is prompted to select the relevant grid.
- ***Geographic Diagram***: Creates a Geographic Diagram of the network.
- ***Block / Frame Diagram***: Creates a new (empty) Block Diagram Graphic object, which is then displayed.
- ***Plot***: Presents a dialog box where the user selects the required plot type. Another dialog box enables parameters to be entered. Once these are confirmed, the plot is then displayed.

Graphic pages can only be shown as pages in a so-called graphics board, so this will automatically be created if required. Creating a new page while in a graphics board can be done by clicking on the icon on the graphics board toolbar.

9.2.5 Page Tab

The page tab of the graphic window shows the name of the graphic. By default the tab is at the top of the window, but a user setting (Window Layout page) enables the user to change this if required. Another user setting allows the user to show icons on the page tab, to indicate the type of graphic. The order of the page tabs is easily changed by using drag-and-drop, and new graphics can be added using the + icon.

A number of options are offered when right-clicking on the page tab. Some of these relate to split-screen working and the options offered will depend on whether the screen is split horizontally or vertically, and also on how many tabs are already open in a group. The options offered to the user also depend on the type of graphic, but this is the complete list of options available:

- *Move to group above/below* or *Move to left/right group*: For split-screen working, move to an existing group. (See section 4.7 for more details)
- *Move rightward/downward to new group*: For split-screen working, start a new group (See section 4.7 for more details)
- *Rename page...* displays a dialog to rename the graphic.
- *Close page* closes the graphic page, removing it from the graphic board in the study case, but not deleting the graphic object itself, if one exists.
- *Delete page* (for plots) will close the graphic page *and* delete the page from the study case.
- *Delete diagram* (for single-line diagrams and geographic diagrams) will close the graphic page *and* delete the diagram itself.
- *Duplicate page*: Used for creating copies of plot pages.
- *Convert to permanent diagram*: If a new detailed diagram of a substation or site has been created, this option enables the user to save it as a permanent graphic.

The name of the active (i.e. currently-presented) graphic is shown in bold. But the above options are available whether the graphic tab is active or not. Tabs can be multi-selected by using the CTRL key (to select individual tabs) or the Shift key (to select a range of tabs), and the selected tabs will be coloured blue; then appropriate options can be applied to the selected tabs.

9.2.6 Tab Group

The tab group offers a drop-down menu consisting of a list of options that allows the opening of diagrams and plots. The icon ⌄ can be used to access this list as shown in Figure 9.2.3, where two tab groups are shown with the drop-down menu selected and the following options illustrated:

- *Open Recently Closed*: This option assists the user to open the recently closed diagrams and plots.
- *Open Diagram*: To open existing diagrams, this option can be used.
- *Open Plot Page*: Used for opening the existing plot pages.

All open plots and diagrams in the tab group are also individually listed and these items can be selected to navigate from one diagram or plot page to the other. Moreover, there is the possibility to move a whole tab group using the options below:

- For docked groups:
  - *Move All tabs to Left/Right* or *Move All tabs Up/Down*: This option can be used to simultaneously move all the tabs of one tab group to another tab group.
  - *Move All tabs to New Floating Group*: The selection of this option enables the user to simultaneously move all the tabs of a particular group to a floating window.
- For floating groups:

  Move All tabs to Main Window: This allows the user to simultaneously move all the tabs in a floating window to the main window.
- For all groups:

  Close All tabs: Simultaneously closes all the tabs of a particular group.

Figure 9.2.3 shows an example

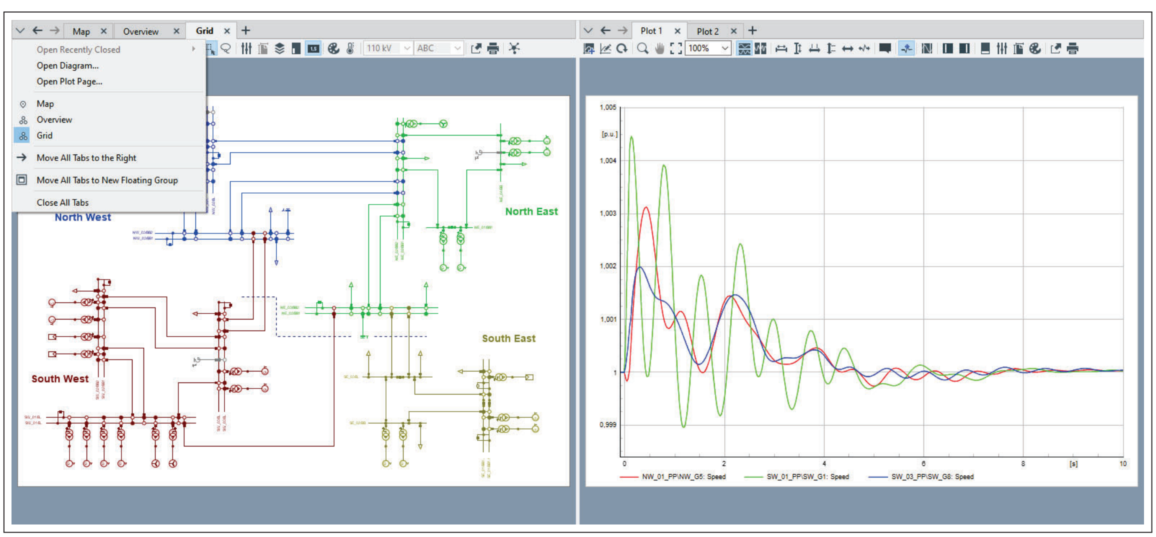


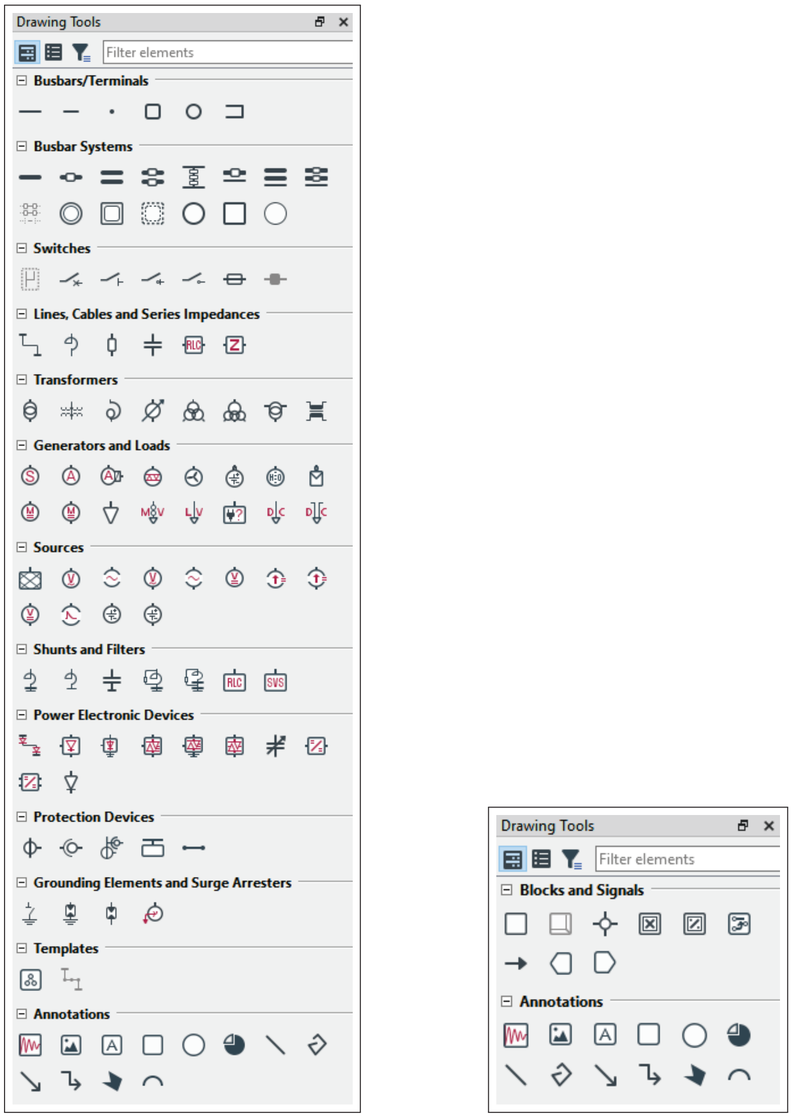
Figure 9.2.3: Drop-down menu for a tab group

9.2.7 Drawing Tools

The Drawing Tools tool-window appears by default on the left-hand side of the graphic window, where it shares the space with the Project Overview. The tool icons shown in the Drawing Tools window form a tool-box which is specific to the type of graphic that is currently selected.

See figure 9.2.4 for two examples.

The Drawing Tools are only available for use if the graphic is not in freeze-mode. A graphic may be unlocked from freeze mode using the 🔒 icon on the graphic tool bar or the similar icon at the top of the Drawing Toolbox itself.



(a) Single Line Diagrams

(b) Block Diagrams

Figure 9.2.4: Drawing Toolbox examples

The user can customise the Drawing Tools window in several ways:

- Group headers can be shown or not.
- Element labels can be shown or not.

- A group filter allows the user to show only certain groups. Alternatively, if the group headers are shown, the plus and minus icons can be used to expand or collapse individual groups.
- There is a text filter for filtering by element label. This is case-insensitive and will find any occurrence of the given text string in the element labels, but only in the groups which are currently visible.

In addition, the Drawing Tools tool-window features a dynamically generated category, called *Recently Used*, which contains the most newly used icons. Once produced, it is automatically situated at the top of the Drawing Toolbar toolbox and, as any other grouping category, can be collapsed (minimised) and/or hidden. To avoid cluttering, it only shows at most one full row of icons, i.e. the number of shown icons depends on the width of the Drawing Tools tool-window.

Note: An icon is considered as used when it is selected and unselected by the user.

9.2.7.1 Annotations

As well as tools for adding elements to a graphic, the drawing toolbox also includes a number of annotation options. By default, these are placed into a separate Annotation layer. See section 9.6 for more information on annotations.

9.2.8 Active Grid Folder (Target Folder)

On the status bar of *PowerFactory* (figure 9.2.5), the active grid folder is displayed on the left-most field, indicating the target folder (grid) that will be modified when changes are made in the network diagram. The active target folder can be changed double-clicking this field and then selecting the desired target folder. This can be useful if the user intends to place new elements on a single line diagram, but have the element stored in a different grid folder in the Data Manager.

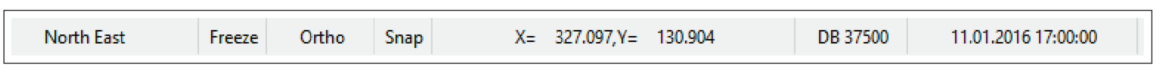


Figure 9.2.5: The Status Bar

9.3 Graphic Commands, Options, and Settings

In this section the commands, options and settings that are available in *PowerFactory* to configure and use the graphic windows are introduced. The sub-sections of this chapter are divided as illustrated in figure 9.3.1. These commands are also available from the main menu under *View*.

Further commands available from the context-sensitive menus of elements are also listed towards the end of this section (9.3.13).

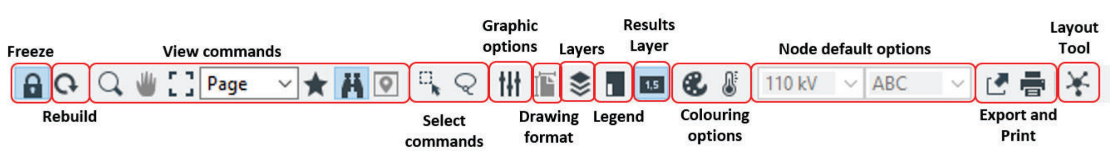


Figure 9.3.1: Categories of graphic commands, options, and settings

9.3.1 Freeze Mode

locks the diagram from graphical changes, no network elements can be added or deleted. Note that the status of switches can still be modified when freeze mode is on.

9.3.2 Rebuild

The drawing may not be updated correctly under some circumstances. The rebuild function updates the currently visible page by updating the drawing from the database.

9.3.3 View commands

9.3.3.1 Zoom In

Press the *Zoom In* to change the cursor to a magnifying glass. The mouse can then be clicked and dragged to select a rectangular area to be zoomed. When the frame encompasses the area you wish to zoom into release the mouse button.

Alternatively, by pressing **Ctrl** and using the mouse scroll wheel it can be zoomed in and out with the mouse cursor as reference point. Using the **Ctrl+-** and **Ctrl++** keys, zooming is also possible referenced to the centre of the visible area. If in addition **Shift** is pressed, the reference changes to the mouse cursor.

Note: The Acceleration Factor for zooming and panning can be changed on the Advanced tab of the *Graphic Window* page in the User Settings dialog.

9.3.3.2 Zoom All

zooms to the page extends.

9.3.3.3 Zoom Level

Zooms to a custom or pre-defined level.

9.3.3.4 Hand Tool

Use the hand tool to pan the single line diagram (when not at the page extends). The hand tool is activated with pressed middle mouse button, too. Alternatively, the mouse scroll wheel can be used to scroll vertically, and **Ctrl+**$\rightarrow / \uparrow / \leftarrow / \downarrow$ used to scroll vertically and horizontally.

9.3.3.5 View Bookmarks

The *View Bookmarks* allows to save the current view and restore that view at a later date. The bookmarks may be used with different network diagrams (single line, geographic, detailed substation graphic) of the same or different grids. In big networks this feature allows to switch very quickly between diagram details to check e.g. the impact of operational changes in loads/feed-in at different places in the network.

By clicking *View Bookmarks* → *Add Bookmark...* the name will be asked, under which the current view is stored and displayed in the list of the View Bookmarks. To edit, delete already existing or even create manually new bookmarks, click on *View Bookmarks* → *Manage Bookmarks...*. An object browser with all existing bookmarks appears. They can directly be changed using the object browser or by opening the Edit-dialog for single bookmarks. The *IntViewBookmark*-objects contain the reference to the diagram, the position and size of the *View Area*. To further accelerate the workflow Hotkeys are set automatically for the bookmarks, which can be changed, too. If the current view should be assigned to the opened bookmark, the button **« From View** may be pressed.

9.3.3.6 Search in Diagram

The icon is used to search for elements in the graphic, as described in section 9.9 below.

9.3.3.7 Navigation Pane

The icon is used to open or close the navigation pane, described in section 9.8 below.

9.3.4 Select commands

9.3.4.1 Rectangular Selection

is used to select a rectangular section of the diagram. This icon is by default pressed, however the *Hand Tool* or *Free-form Selection* may also be used.

9.3.4.2 Free-form Selection

is used to select a custom area of the diagram.

9.3.5 Graphic Options

Each graphic window has its own settings, which may be changed using the *Graphic Options* button . The available settings of the dialog are described in the following sections.

9.3.5.1 Basic Attributes page

The *General*-tab offers the following settings:

- **Name:** the name of the graphic
- **Target folder for network elements:** the reference to the database folder in which new power system elements created in this graphic will be stored.
- **Default view area:** when the user has zoomed into part of the graphic, the icon can be used to zoom out again. By default, this will be to the full area of the graphic, but by selecting a bookmarked area here, as the default view area, this “zoom all” can be customised. This is useful for

users of large networks who are only interested in a specific region. (See section 9.3.3.5 for more information about bookmarks.)

- **Write protected:** if enabled, the single line graphic can not be modified. The drawing toolboxes are not displayed and the *freeze* icon becomes inactive.
- **Line style for cables:** is used to select a line style for all cables.
- **Line style for overhead lines:** is used to select a line style for all overhead lines.
- **Node width factor:** the width of points and lines for nodes and busbars.
- **Offset factor when drawing one-port devices:** defines the length of a connection when a one port device (e.g. load, shunt) is drawn by clicking on the busbar/terminal. This is the default distance from the busbar in grid points.
- **Allow individual line style:** permits the line style to be set for individual lines. The individual style may be set for any line in the graphic by right-clicking the line → *Set Individual Line Style*. This may also be performed for a group of selected lines/cables in one action, by first multi selecting the elements.
- **Allow individual line width:** as for the individual line style, but may be used in combination with the "Line Style for Cables/Overhead Lines" option. The individual width is defined by selecting the corresponding option in the right mouse menu (may also be performed for a group of selected lines/cables in one action).
- **Diagram colouring:** by default, changes of the active Colouring Scheme take effect on every diagram (Default). By setting the option to *Colouring scheme*, the scheme of the current diagram can be configured separately. Press **Manage...** to open an object browser with a list of the available colouring scheme settings. Copy the existing or create a new one and alter it to the wished scheme. Close the object browser and choose the new colouring scheme out of the drop down list.

The *Advanced*-tab offers the following settings:

- **Allow navigation pane to be shown:** if checked, the *Navigation Pane* can be activated by *Window* → *Navigation Pane* or the context sensitive menu in the diagram *Navigation Pane*.
- **Show tooltip on network elements:** if this is selected, information about network elements will be shown if the user hovers the mouse over the element.

The *Additional Attributes* and *Coordinates Space* pages should generally only be configured with the assistance of *DIgSILENT* support staff. Note that if *Use Scaling Factor for Computation of Distances* is selected on the Coordinates Space page, it is possible to calculate the length of lines on the Single Line Graphic by right-clicking and selecting *Measure Length of Lines*. In geographic diagrams, this option is activated by default.

9.3.5.2 Schematic Diagram page

When a schematic diagram (overview, single line or detailed) is active, the *Schematic Diagram* page will be available with the following options:

Drawing Tools

- **Snap:** snaps the mouse onto the drawing raster.
- **Grid:** shows the drawing raster using small points.
- **Ortho-Type:** defines if and how non-orthogonal lines are permitted:
  - **Ortho Off:** connections will be drawn exactly as their line points were set.
  - **Ortho:** allows only right-angle connections between objects.

- **Semi Ortho:** the first segment of a connection that leads away from a busbar or terminal will always be drawn orthogonally.

Size Factors for

Defines the size of the symbols in the diagram for sites, substations, edge elements and line end symbols. The *connection circles on simplified substations* is a width factor used in single line diagrams: In single line diagrams multiple busbar substations are only represented by their main busbars. Connected elements may be connected to each of the busbars by changing the state of the internal switches. The currently active connection is shown in the diagram by a filled circle, the not connected ones by hollow circles. The width of the circles is defined in this field.

Note: The settings for the cursor type for the graphic windows (arrow or tracking cross) may be set in the User Settings dialog, see section 7.3 (Graphic Windows Settings). This is because the cursor shape is a global setting, valid for all graphic windows, while all graphic settings described above are specific for each graphic window.

9.3.5.3 Text Boxes page

The following options are available:

- **Allow individual fonts for text boxes:** self-explanatory.
- **Use fonts from project settings:** this option allows the user to use the font configuration (i.e. font type, style, size, effects) defined in the Project Settings. When activated, a button is available to access the corresponding Project Settings page.
- **Labels**
  - **Nodes:** in case the option *Use fonts from project settings* is not active, the user can customize the font characteristics. The option *Show frame* shows a frame in the text boxes corresponding to node labels.
  - **Branches:** in case the option *Use fonts from project settings* is not active, the user can customize the font characteristics. The option *Show frame* shows a frame in the text boxes corresponding to branch labels.
  - **Background:** specifies the transparency of label boxes:
    - **Opaque:** means that objects behind the label box cannot be seen through the label box.
    - **Transparent:** means that objects behind the label box can be seen through the label box.
- **Results**
  - **Nodes:** in case the option *Use fonts from project settings* is not active, the user can customize the font characteristics. The option *Show frame* shows a frame in the text boxes corresponding to node results.
  - **Branches:** in case the option *Use fonts from project settings* is not active, the user can customize the font characteristics. The option *Show frame* shows a frame in the text boxes corresponding to branch results.
  - **Background:** specifies the transparency of result boxes:
    - **Opaque:** means that objects behind the result box cannot be seen through the result box.
    - **Transparent:** means that objects behind the result box can be seen through the result box.
  - **Always show result boxes of detailed couplers:** self-explanatory.
  - **Space saving representation of result boxes on connection lines:** self-explanatory.
- **Title and legends:** in case the option *Use fonts from project settings* is not active, the user can customize the font characteristics corresponding to the title and legends.

- **Show line from general text boxes to referenced objects:** may be disabled to unclutter the graphic.

9.3.5.4 Switches page

- **Switch state symbol at connection end:** selects the switch representation (see figure 9.3.2):
  - **Permanent box:** shows a solid black square for a closed and an frame line for an open switch (left picture).
  - **Old style switch:** shows the switches as the more conventional switch symbol (right picture).

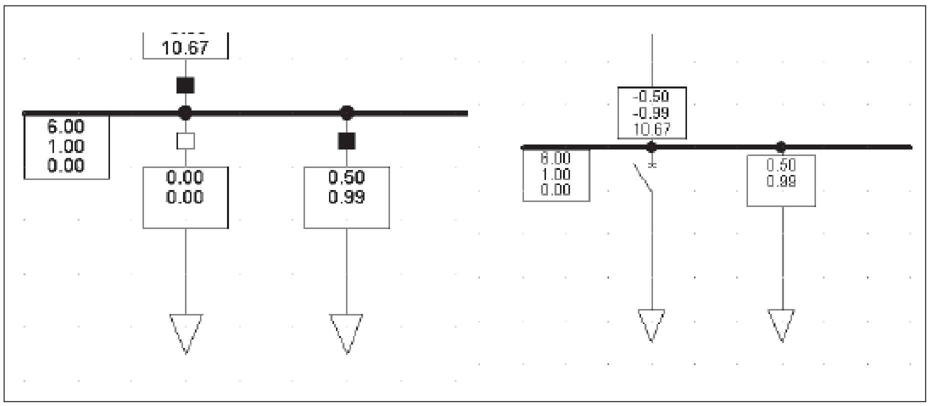


Figure 9.3.2: Cubicle representations

- **Display frame around switches:** draws a frame around the switch itself (breakers, disconnectors, etc.). This only applies to user-drawn breakers and disconnectors.
- **Create switches when connecting to busbar:** self-explanatory.
- **Show connected busbars as small dots in simplified substation representation:** defines how the connection points on busbars are represented in busbar systems.

9.3.5.5 Geographic Diagram page

The settings on this page define the appearance of the graphical representation of network elements in the geographic diagrams. This page is only visible when a geographic diagram is active.

- **Size factors for:** defines the size of the symbols in the diagram for sites, substations, terminals, edge elements, text, line loads and section transitions and line end symbols.
- **Scale level threshold for visibility of:** in extensive networks with a high scale level, edge elements (except lines), switch state boxes at line ends, text labels and annotation objects are hidden at a specified scale level to improve the clarity of the diagram.
- **Line width:** sets the width of all the lines in the geographic diagram.
- **Distance factor for one-port devices:** defines the distance of all drawn one-port-devices (e.g. load, shunt) to their connected nodes. This is the default distance from the busbar in grid points.
- **Margin at full zoom:** since in geographic diagrams there is no border, this value defines the margin shown if *Zoom All* is pressed.
- **Show coordinates in latitude/longitude:** shows the coordinates of the current cursor position in latitude and longitude in the Status Bar. Otherwise the position is displayed as X/Y values representing UTM-coordinates. The border values of the area represented by the diagram are listed in the tab *Coordinate Space* of the *Basic Attributes* page (9.3.5.1).

- **Prefer branch coordinates:** this option affects elements which are grouped to branches (*Elm-Branch*). If the branch itself has geographic coordinates, they will be used in the geographic diagram, otherwise the coordinates of the elements contained in the branch are taken into account.
- **Show Scale:** shows or hides the scale in the diagram.
- **GPS projection:** defines the projection used in the geographic diagram. This is only active when local maps are selected.

Substation Types tab

The settings in this tab are related to the graphical representation of substations in geographic diagrams. The dialog offers the possibility to distinguish graphically different types of substations and improve the clearness of the diagram by adding additional data through the substation symbol. A possible use of this feature can be seen in the geographic diagram 'Overview Diagram' of the Application Example 'MV Distribution Network' (*File* → *Examples*).

The column 'Substation Type' of the *Assignment Table* is the list, of which one element may be chosen in the drop down list *Type* in the *Description*-page of the Substation-dialog, opened by right-clicking on a graphical substation element and choosing *Edit Substation*. The list contains the Substation Types defined in the according page of the Project Settings dialog.

The column 'Symbol' is storing the name of the symbol, which is searched by default in the subfolder *Database/System/Library/Graphic/Symbols/SGL/Composites*. As additional source, the project's sub-folder *Settings/Additional Symbols* is taken into account.

9.3.6 Layers

The single line, geographic and block diagrams use transparent layers of drawing sheets on which the graphical symbols are placed. Each of these layers may be set to be visible or not, and layer depth functionality is built in, meaning that the user can select the order in which layers sit on top of one another.

Which layers are visible and exactly what is shown on a layer is defined in the *Graphical Layers* dialog, accessed through the graphic toolbar ( ), by right-clicking on an empty spot of the graphic area → *Layers*, or selecting *View* → *Layers* from the main menu.

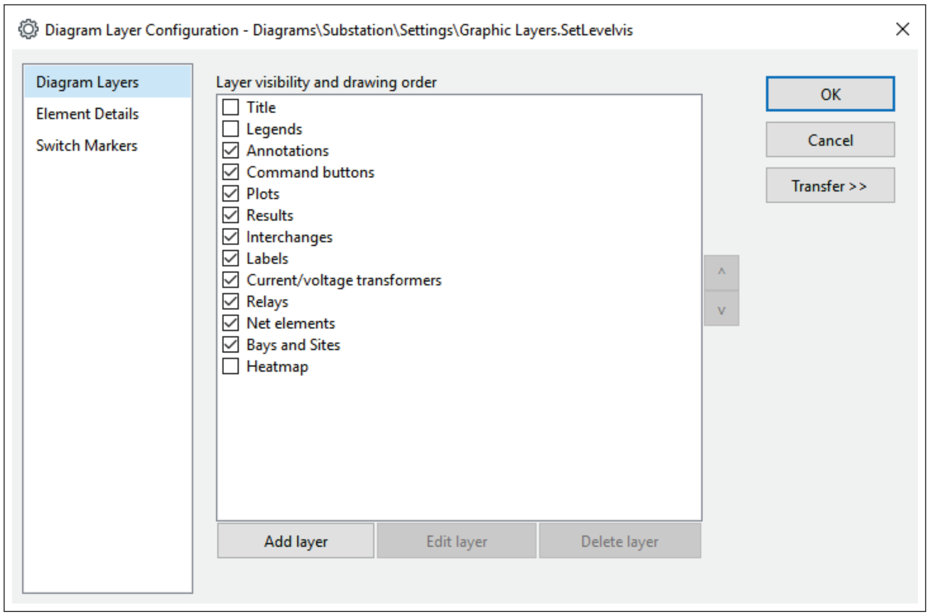


Figure 9.3.3: Graphical layers dialog (*SetLevelvis*)

As shown in Figure 9.3.3, the Graphic Layers dialog has three pages:

- **Diagram Layers**, where default and user-defined layers are created, modified, selected and ordered
- **Element Details**, where symbols related to specific elements are selected to be visible or not
- **Switch Markers**, where layers specifically to switch annotation are selected visible or not

These are described in more detail in the following subsections.

Note: Prior to *PowerFactory* 2019, all annotation graphical objects were held in a dedicated Annotation Layer. Now, it is possible to mix annotation graphical objects with other graphical objects in any user-defined layer. Nevertheless, there is still an "Annotation Layer" where such objects will by default be placed when created. See Section 9.6 for more details.

9.3.6.1 Diagram Layers

On this page, the default and user-defined layers are listed. The visibility of any one layer is determined by the check-box. New layers can be created using the **Add layer** button and deleted using the **Delete layer** button. The detailed configuration of any layer can be changed using the **Edit layer** button, or by double-clicking on the layer name. The order of the layers can be changed using the up and down arrows to the right of the list (see Figure 9.3.3 above). Depending upon the type of graphic, the following layers are available as a default:

Layer	**Content**	**Configuration Options**	**Diagram Type: SL Single Line, GEO Geographic, B Block**
Title	Graphic title	Text/Box Format	SL/GEO/B

Layer	Content	Configuration Options	Diagram Type: SL Single Line, GEO Geographic, B Block
Legends	Results boxes legend and colour legend	Text/Box Format	SL/GEO
Net elements	Symbols for the elements of the grid	Text/Box Format	SL/GEO/B
Labels	Boxes with names and additional data description, if configured	Text/Box Format	SL/GEO/B
Results	Boxes with calculation results. See **Note** below	Text/Box Format	SL/GEO/B
Interchanges	Boxes and arrows, together with associated results boxes	Separate visibility and configuration of boxes and arrows	SL/GEO
Bays and Sites	Bay representation and Site frames	Line colour, width and style	SL
Device data	Additional Text explanation given in the device symbol	Text/Box Format	SL/GEO/B
Background image	Graphic used as the background (“wallpaper”) to allow easier drawing of the diagram or to show additional information (map information)	Name of file with graphics (WMF, BMP, JPEG, PNG, GIF, TIF)	SL/B
Geographic map	Geographical map used as the background to allow easier drawing of the diagram	Map Provider (see section 9.10), map type and graphic settings	GEO
Load/generation distribution	Shows circles for load and generation around substations	Selection of S, P or Q, colour-settings	GEO
Additional text	Additional text labels for elements	Text/Box Format	SL/GEO
Relays	Graphically represented relays	Text/Box Format	SL/GEO
CTs and VTs	Contains the drawn current and voltage transformers	Text/Box Format	SL/GEO
Plots	Plots placed in the diagrams	Text/Box Format	SL/GEO/B
Commands buttons	Commands buttons used mostly for DPL scripts	None	SL/GEO
Block Definition	Definition each block is based on	Text/Box Format	B
Connection numbers	Index of each possible block connection point	Text/Box Format	B
Connection names	Name of each unused connection of a block	Text/Box Format	B
Signals	Name of the signal transmitted	Text/Box Format	B

Table 9.3.1: Predefined diagram layers in the layers dialog

Note: The Results layer can be easily toggled on and off without going into the Layers command by using the  icon on the graphics toolbar.

9.3.6.2 Element Details

For some elements shown on diagrams, additional details can optionally be shown, such as the vector groups for transformers or symbols to indicate the number of phases of a line. On the Element Details page, such details are turned on or off, with additional configuration for the Power flow direction arrows option.

Layer	Content	Configuration Options	Diagram Type: SL Single Line, GEO Geographic, B Block
Connection points	Dots at the connections between edges and buses/terminals and signal connections to blocks	Text/Box Format	SL/GEO/B
Connection Arrows	Double-Arrow at connections where the end point is not represented in the current diagram.	Text/Box Format	SL/GEO
Phases	Number of phases of a line/cable, shown as parallel lines	Text/Box Format	SL/GEO
Sections and Line Loads	Symbols at lines consisting of sections and/or where line loads are connected	Text/Box Format	SL/GEO
Line compensations	Symbols at the ends of the lines, indicating the presence of line compensation	Text/Box Format	SL/GEO
Vector groups	Vector group for rotating machines and transformers	Text/Box Format	SL/GEO
Tap positions	Positions of taps for shunts and transformers	Text/Box Format	SL/GEO
Power Flow direction arrows	Arrows that can be configured for active and reactive power flow representation	Active/Reactive Power for direct/ inverse/ homopolar system	SL/GEO

Table 9.3.2: Element Details in the Layers dialog

9.3.6.3 Switch Markers

On the Switch Markers page, markers specific to operation and/or status of switches (ElmCoup or StaSwitch) can be selected and configured:

Remotely controlled substations

This type of marker indicates if sites (ElmSite), substations (ElmSubstat) or secondary substations (ElmTrfstat) host remote controlled switches. This switch attribute can be defined in the field Fault Separation/Power Restoration, on the page Reliability of the switch edit dialog.

If checked, an user-defined colored circle is displayed behind sites, substations and secondary substations, where at least one switch within the aforementioned grouping object is set to be remote-controlled.

This switch mark is only visible if the grouping object has a Composite Node (Beach Ball) graphical representation. Additional information about node graphical representation can be found in section 11.2.7.

Tie open points

The marker suggests that a feeder (ElmFeeder) terminates at the open switch.

If this option is selected, an user-defined colored circle is shown around open switches, if different feeders are found on either side, also if no feeder is detected at all on one side.

In addition, in the case that the switch at which the feeders terminate is located within a site, substation or secondary substation, the circle mark will be also displayed on diagrams where a composite node representation of the grouping object exists. It should be noted that the mark will be only presented at the composite node representation of a grouping object, if none of the connected branch objects already shows a tie open point circle at its line end.

Normally open switches

This is self-explanatory. A circle mark is depicted around switches, if the normal state of a switch is open. The attribute that specifies whether the normal state of a circuit breaker is open can be defined in the Tie Open Point Optimisation page of the switch edit dialog.

For those cases where normally open switches are located within a site, substation or secondary substation, the circle mark will be also displayed on diagrams where a composite node representation of the aforementioned grouping object exists. It should be noted that the mark will be only presented at the composite node representation of a grouping object, if none of the connected branch objects already shows a normally open switch at its end.

Open standby switches

It shows if despite opening a switch within a substation or secondary substation, elements on both sides of the switch remain supplied. In other words, it suggests that the aforementioned elements are still connected to a source (standby). The annotation mark works only for devices which belong to a substation or secondary substation (Bus couplers are excluded).

9.3.6.4 Working with Layers

Each graphic symbol in a single line, geographic or block diagram is assigned by default to the corresponding layer at first. All busbar symbols, for example, are drawn on the *Net elements* layer by default, their name boxes on the layer *Labels*. Graphic symbols may be shifted onto other layers by right-clicking them in the single line graphic and selecting the option *Shift to Layer* from the context sensitive menu, then selecting the layer from the list presented. Should an object disappear when it has been re-assigned to a layer, the visibility of the target layer should be checked. Moving symbols from one layer to another is normally needed when only a few symbols from a certain group should be made visible (for instance the result boxes of one or two specific junction nodes), or when user defined layers are used. This allows to hide some elements or text boxes to improve the clarity of the diagram, or to show additional information for e.g. printing purposes.

Note: Certain names and result boxes are, by default, not visible. An example is the names and result boxes for internal nodes. This is done to reduce clutter on the graphic. To display such labels, simply right-click on the object and select *text boxes* → *Show labels*.

The default layer for annotations is called Annotations Layer, which is empty until the first annotation object is inserted. If the user creates new layers, annotation objects can easily be shifted from one to another by right clicking on them, selecting *Shift to layer* and select the destination layer from the list. More information about annotations can be found in section 9.6.

For some layers, for example Text Boxes or Results, when the layer is edited to change the configuration, a *target* may be set; this target will be the focus of the performed configuration command. Various actions or settings may be performed, such as e.g. changing the font using the **Change Font** button. The configuration page may also be used to mark (select/highlight) the target objects in the graphic using the **Mark** button.

The options available to configure a layer depend on the type of Layer. Table 9.3.1 shows for each layer in which way its content can be changed in format.

As an example, suppose that a part of the single line graphics is to be changed, for instance, to allow longer busbar names. To change the settings, the correct graphical layer is first selected. In this example, it will be the *Labels* layer. In this layer, only the busbar names are to be changed, and the target must therefore be set to *All Nodes*. When the layer and the target has been selected, the width for object names may be set in the *Settings* area. The number of columns may be set using the **Visibility/Width** button. Alternatively, the **Adapt Width** will adapt all of the object name placeholders to the length of the name for each object. Changing a setting for all nodes or all branches at once will overwrite the present settings.

In the detailed diagram of the substations, it is possible to display the names of neighboured substations/sites next to the lines leaving the substation. In order to achieve that, edit the *Labels* layer and on the *Text Boxes* page, the option "Show jump-to labels at graphically half-connected lines" should be checked.

The visibility of the layer can be selected within the edit dialog of that layer (as well as in the main Layers dialog), and - as additional options - users can choose to set maximum and/or minimum zoom-dependent visibility levels, so that objects are not shown outside these zoom levels. For geographic diagrams the limits are expressed in terms of a map-scale such as 1:5000, for schematic diagrams as a percentage zoom level.

Many layers have an option to make the objects selectable or not. This can be useful if the layer contains annotation objects, for example.

For certain layers for example the Results layer, the Labels layer and user-defined layers, there is the possibility to define dedicated colours. The colours can be set using the colour palettes within the object dialog of each layer. See section 4.7.5.1 for more information about the use of colour palettes. After selection of the colour, the colouring mode for layers can be activated using the *Diagram Colouring* icon. This allows the user to easily distinguish between the different layers present in the drawing and to easily identify which elements belong to each respective layer.

9.3.7 Colouring Options

9.3.7.1 Diagram Colouring

The single line and geographic diagrams have an automatic colour representation mode. The *Diagram Colouring* icon on the diagrams toolbar will open the diagram colouring representation dialog (alternatively, select *View* $\rightarrow$ *Diagram Colouring* on the main menu). This dialog is used to select different colouring modes and is dependent if a calculation has been performed or not. If a specific calculation is valid, then the selected colouring for that calculation is displayed.

As described below, colours are selected via the *Colour Settings* button found in the *Diagram Colouring* dialog. Clicking on this button brings up a new dialog, which has two pages. On the first page called "General", the user has the option to select a colour palette. This is not essential, but if a palette is selected, the user will find that colours from this palette will be offered as a default selection when configuring colours for elements in diagrams or in a Network Model Manager or Data Manager. For general information about configuring colours and the use of colour palettes see Section 4.7.5.

The *Diagram Colouring* has a 3-priority level colouring scheme implemented, allowing colouring elements according to the following criteria: 1st Energising status, 2nd Alarm and 3rd "Normal" (Other) colouring.

> ***Energising Status:*** if this check box is enabled "De-energised" or "Out of Calculation" elements are coloured according to the settings in the "Project Colour Settings". The settings of the "De-energised" or "Out of Calculation" mode can be edited by clicking on the *Colour Settings* button.

Alarm: if this check box is enabled a drop down list containing alarm modes will be available. It is important to note here that only alarm modes available for the current calculation page will be listed. If an alarm mode is selected, elements “exceeding” the corresponding limit are coloured. Limits and colours can be defined by clicking on the *Colour Settings* button.

“Normal” (Other) Colouring: here, two lists are displayed. The first list contains all available colouring modes. The second list contains all sub modes of the selected colouring mode. The settings of the different colouring modes can be edited by clicking on the *Colour Settings* button.

Every element can be coloured by one of the three previous criteria. Also, every criterion is optional and will be skipped if disabled. Regarding the priority, if the user enables all three criteria, the hierarchy taken into account will be the following:

- “Energising Status” overrules the “Alarm” and “Normal Colouring” mode. The “Alarm” mode overrules the “Normal Colouring” mode.

The graphic can be coloured according to the following list. Availability of some options will depend on the function that is selected (e.g. ’Voltage Violations’ does not appear when the ’Basic Data’ page is selected, but does when the ’Load Flow’ page is selected) and on the licence (e.g. Connection Request is only available if the advanced function Connection Request Assessment is part of the licence).

Energising Status:

- De-energised
- Out of Calculation
- De-energised, Planned Outage

Alarm:

- Feeder Radiality Check (Only if “Feeder is supposed to be operated radially” is selected).
- Outages
- Overloading of Thermal/Peak Short Circuit Current
- Voltage Violations/Overloadings

“Normal” (Other) Colouring:

- Results
  - Fault Clearing Times
  - Voltage Angle (colouring according to absolute or relative voltage angles and angle differences along branch elements; relative voltage angles do not reflect transformer vector groups, while absolute voltage angles include the angle shift caused by transformer vector groups)
  - Voltages / Loading
  - Loading of Thermal / Peak Short-Circuit Current
  - Incident Energy
  - PPE-Category
  - Connection Request: Approval Status
  - Contribution to EIC
  - Contribution to ENS
  - Contribution to SAIDI
  - Contribution to SAIFI
  - Loads: Average Interruption Duration
  - Loads: Load Point Energy Not Supplied
  - Loads: Yearly interruption frequency

  - Loads: Yearly interruption time
  - Optimal Manual Restoration
  - Probabilistic Analysis
  - State Estimation
- Topology
  - Boundaries (Definition)
  - Boundaries (Interior Region)
  - Connected Components
  - Connected Components, Voltage Level
  - Connected Grid Components
  - Energising Status
  - Feeders
  - Missing graphical connections
  - Outage Check
  - Station Connectivity
  - Station Connectivity (Beach Balls only)
  - Supplied by Secondary Substation
  - Supplied by Substation
  - System Type AC/DC and Phases
  - Voltage Levels
- Primary Equipment
  - Cross Section
  - Year of Construction
  - Forced Outage Duration
  - Forced Outage Rate
- Secondary Equipment
  - Measurement Locations
  - Power Restoration
  - Relays, Fuses, Current and Voltage Transformers
  - Switches, Type of Usage
- Groupings (Grids, Zones, Areas...)
  - Areas
  - Grids
  - Layers
  - Meteo Stations
  - Operators
  - Owners
  - Paths
  - Routes
  - Zones
- Variations / System Stages
  - Modifications in Recording Expansion Stage
  - Modifications in Variations / System Stages
  - Original Locations

- User-defined
  - Individual

The list *User-defined* may be used to define own colouring schemes. Pressing **Manage Filters...** opens an object browser with the list of all available user-defined filters, found in the subfolder Settings/Colouring/Colouring Scheme. New filter sets (*IntFiltset*) can be created, containing several General Filter objects (*SetFilt*) with an assigned colour and the conditions, under which an element is coloured. This allows to implement very specific filters to identify graphically elements in the diagram with certain properties or results.

9.3.7.2 Heatmaps

In *PowerFactory*, heatmaps can be used to illustrate the state of a grid by colouring the area around network elements. The colour definition is carried out as described in section 9.3.7.1.

To use the colour definition for heatmaps, click on the *Heatmap* button. On the *General* page of the dialog, the basic settings for the creation of the heatmap can be selected. The colour settings dialog (explained in section 9.3.7.1) is accessible from this page. The *Mode* shows which type of colouring is used.

The resolution of the Heatmap can be:

- Low
- Medium
- High
- User-defined (in pixels)

Note: The amount of time required to generate each heatmap increases with the specified resolution. Since the optimal settings for heatmaps vary for each grid, the process of finding this optimum might take a few iterations. Therefore it is advised to start with a small or medium resolution.

A background colour for the heatmap may additionally be set.

The *General* page defines the general settings for the Heatmap and the *Advanced* page defines specifics regarding the colouring. Five different parameters can be set; the first two being:

- **Number of closest influence points:** defines the number of reference points taken into account when colouring a certain point of the Heatmap.
- **Contour sharpness:** defines the smoothness of the transition between differently coloured areas.

The other three parameters define the *Fading Area*, i.e. the orthogonal transition from the element colouring to the background colour:

- **Begin:** defines how far away from the centre of the element on the lateral axis the colouring begins to fade to the background colour.
- **Extent:** defines how far away from the centre of the element on the lateral axis the colouring ends to fade to the background colour.
- **Fading exponent:** defines how fast the colour between *Begin* and *Extent* will fade to the background colour.

Figure 9.3.4 shows an example of a Heatmap, which is coloured according to loading, over- and under-voltage.

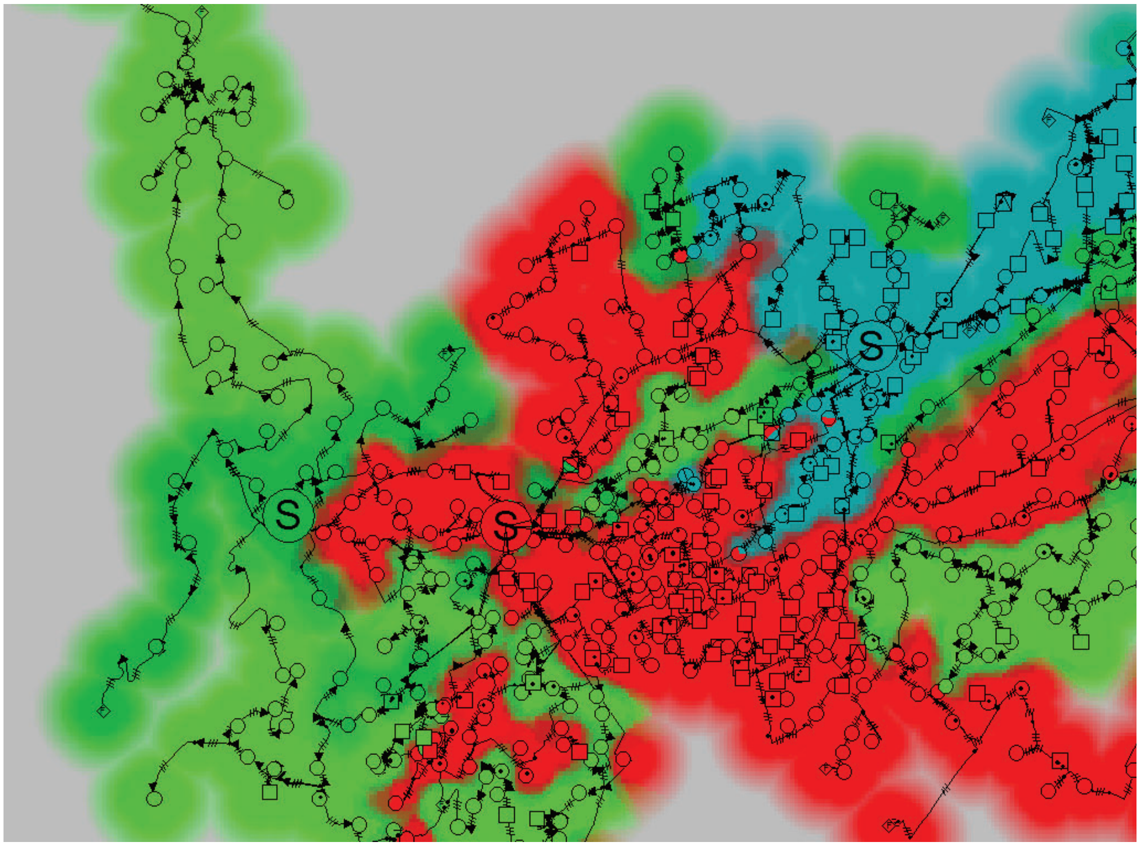


Figure 9.3.4: Example of a heatmap coloured according to loading, over- and under-voltage

9.3.8 Graphic Legends

9.3.8.1 Show Title Block

The title block can be turned on and off from the Layers dialog (see 9.3.6). The title block is placed in the lower right corner of the drawing area by default.

The contents and size of the title mask can be changed by right-clicking the title block and selecting the *Edit Data* option from the context sensitive menu. The Select Title dialog that pops up is used to scale the size of the title block by setting the size of the block in percent of the default size. The font used will be scaled accordingly. To edit the text in the title block press the edit button (→) for the 'Title Text' field.

All text fields have a fixed format in the title block. The data and time fields may be chosen as automatic or user defined. Most text fields are limited to a certain number of characters. When opening a new graphic the title will appear by default.

9.3.8.2 Show Legend Blocks

The legend blocks can be turned on and off from the Layers dialog (see 9.3.6), or from the single line diagram toolbar ( ).

The results legend block describes the contents of result boxes (for information about result boxes see 9.5).

Because more than one type of result box is normally used in the Single line graphic, for instance, one for node results and another one for branch results, the legend box normally shows more than one column of legends. After changing the result box definitions, it may be necessary to manually resize the legend box in order to show all result box legends.

The Legend Box definition dialog is opened by right-clicking the legend block and selecting *Edit Data* from the context menu. The font and format shown may be configured. When opening a new graphic the legend will appear by default.

The other block is the colour legend block, which updates automatically based on the colouring options selected.

9.3.9 Node Default Options

Figure 9.3.5 shows the commands available for setting node default options. These are discussed in further detail in this section.

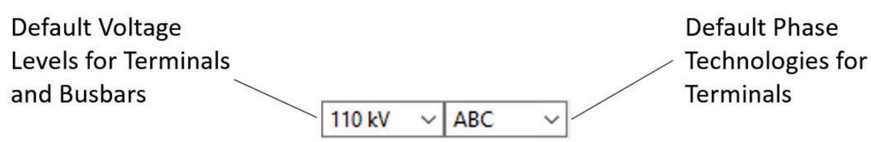


Figure 9.3.5: Node default options

Default Voltage Levels for Terminals and Busbars:

The default voltage level for terminals can be set in this field. New terminals placed on the single line diagram will have this voltage (e.g. 110 kV, 0.4 kV).

Default Phase Technologies for Terminals:

The default phase technology for terminals can be set in this field. New terminals placed on the single line diagram will be of this type (e.g. three-phase ABC, single-phase, DC, etc.).

9.3.10 Page, Print and Export Options

9.3.10.1 Drawing Format

The drawing area for single line diagrams, block diagrams and plots is modified in the *Drawing Format* dialog, accessed using the button. A predefined paper format can be selected as-is, edited, or a new format can be defined. The selected paper format has 'Landscape' orientation by default and can be rotated by 90 degrees by selecting 'Portrait'. The format definitions, which are shown when an existing format is edited or when a new format is defined, also show the landscape dimensions for the paper format.

It is not possible to draw outside the selected drawing area. If a drawing no longer fits to the selected drawing size, then a larger format should be selected. The existing graphs or diagrams are repositioned on the new format (use **Ctrl+A** to mark all objects and then grab and move the entire graphic by left clicking and holding the mouse key down on one of the marked objects; drag the graphic to a new position if required).

9.3.10.2 Print

This function is accessed via the button, the menu *File* → *Print* or via the hotkey **Ctrl+P**. It opens a print preview, showing the diagram to be printed. The dialog offers all the options that the user would expect, including a Print Setup dialog, accessed via the icon.

If only part of the diagram is to be printed, the selection of the area can be done within the print preview dialog. There is also a *Pages* option, which allows the user to print selected graphic pages.

9.3.10.3 Exporting Graphics

Graphics can be exported in a wide range of formats, using the *Export Diagram* function. This is accessed either from the main toolbar, using *File* → *Export Diagram...*, or by using the icon on the graphic toolbar.

If only part of the diagram is to be exported, the selection of the area can be done within the preview dialog. For PDF exports, there is also a *Pages* option, which allows the user to select graphic pages to be exported.

9.3.11 Diagram Layout Tool

The *Diagram Layout Tool* is a powerful feature to create graphical representations of network topologies. The tool offers a manual, semi- and fully automatic creation of nodes and branch elements, which are not yet graphically represented in the current Network Diagram. The options and the dialog are described in detail in section 11.6.

9.3.12 Insert New Graphic

Pressing the button opens the *New* command dialog described in section 9.2.4.

Note: The Page Tab menu is opened by right-clicking a page tab, shown just below the single line diagram. Existing graphics can be opened by selecting *Show Graphic* of the context sensitive menu of the graphic object in the subfolder Network Model/Diagrams or by choosing it from the list, which opens after selecting *Insert Page* → *Open Existing Page* from the context sensitive menu of the page tab.

9.3.12.1 Other Page Commands

Other page commands accessed through the page tab are as follows:

Remove Page: this function will remove the selected graphic from the Graphics Board. The graphic itself will not be deleted and can be re-inserted to the current or any other Graphics Board at any time.

Rename Page: this function can be used to change the name of the selected graphic.

Move/Copy Page(s): this function can be used to move a page/s to modify the order of graphics. Also accessed through:

- Mouse Click: Left-click and select a single page (optionally press control and select multiple pages) and drag the page/s to change the order graphics are displayed.
- Data Manager: (Advanced) Modify the order field of Graphics Pages listed within the Study Case Graphics Board. To reflect the changes, the study case should be deactivated and then re-activated.

9.3.13 Element Options

In addition to the options available from the graphics toolbar, there are many edit options which are available for elements in the graphic, using the context-sensitive menu (i.e. right-click).

Edit and Browse Data: this option lets the user edit the data of the selected object. The object will be selected in its project folder within the list of the other objects and can be double-clicked to open its edit dialog. See chapter 10 (Data Manager) for more information.

Delete: this function deletes both the database and graphical objects for the selected element(s). A warning message will appear first - this may be switched off in the "User Settings" dialog; see section 7.3 (Graphic Windows Settings)). Also accessed through the keyboard: **Del** key.

Note: To delete graphical objects only, right click the selected element/s and select *Delete Graphical Object only*.

Cut: this function cuts the selected objects. Objects can then later be pasted as discussed below. Also accessed through the keyboard:**Ctrl+X** key.

Copy: copies the selected objects to the clipboard. Also accessed through the keyboard: **Ctrl+C** key.

Paste: pastes all objects from the clipboard into the current drawing. The objects are pasted at the current graphical mouse position. Objects that are copied and pasted create completely new graphic and data objects in the graphic that they are pasted into. Also accessed through the keyboard: **Ctrl+V** key.

Note: To copy and paste just the graphic, *Paste Graphic Only* should be chosen from the right-click menu. Similar results are obtained when using the "Draw Existing Net Elements" tool (see Section 11.6: Drawing Diagrams with Existing Network Elements).

Note: The undo command undoes the last graphic action and restores deleted elements, or deletes created elements. The undo command is accessed through the *undo* icon (), by right-clicking and selecting 'Undo', or by pressing **Ctrl+Z**.

Rotate: rotates symbols clockwise, counter-clockwise, or 180 degrees. It is generally preferable to disconnect an element before rotating it.

Disconnect: disconnects the selected element/s. When right-clicking at the end of a connection element a different/reduced menu is shown which allows disconnecting just the selected side (*Disconnect Side*)

Connect: right-click and select *Connect Element* to connect an element.

Reconnect: used to disconnect a selected element and then re-connect it. The branch to be connected will be 'glued' to the cursor. Left clicking a bar or terminal will connect the element. When right-clicking at the end of a connection element a different/reduced menu is shown which allows reconnecting just the selected side (*Reconnect Side*)

Redraw: right-click and select *Redraw Element* to redraw a selected element.

Move: marked objects can be moved by left clicking them and holding down the mouse button. The objects can be moved when the cursor changes to an arrowed cross (). Hold down the mouse button and drag the marked objects to their new position. Connections from the moved part of the drawing to other objects will be adjusted.

Edit Line Points: right-click and select *Edit Line Points* will show the black squares ('line points') that define the shape of the connection. Each of these squares can be moved by left clicking and dragging them to a new position (see figure 9.3.6). New squares can be inserted by left clicking the connection in between squares. Line points are deleted by right-clicking them and selecting the *Delete Vertex* option from the case sensitive menu. This menu also presents the option to stop (end) the line point editing, which can also be done by left clicking somewhere outside the selected lines.

Push to Back: to select a particular object when multiple objects are lying on top of each other in network graphics, there is a possibility to temporarily push the selected elements back. For this purpose, press the **CTRL** key and then right-click on the element, which is to be pushed to the back. The "Push to Back" option appears at the bottom of the context-sensitive menu that appears. Subsequently, right-clicking on the overlapping objects will select the element which has not been pushed back, the user can then either move the element graphically or choose any of the element edit options from the context-sensitive menu. Alternatively, the combination of pressing **ALT** and the left mouse button can be used to select the top level element.

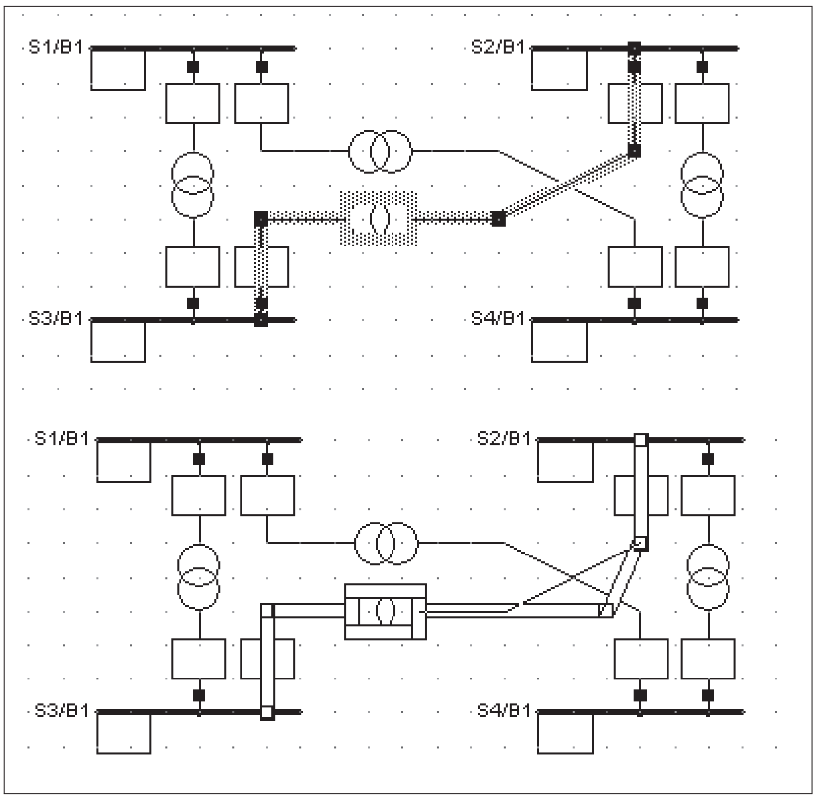


Figure 9.3.6: Editing line points

9.4 Editing and Changing Symbols of Elements

You can edit or change the symbols, which are used to represent the elements in the single line graphic. Right click with on a symbol of an element in the single line graphic, and select *Change Symbol* from the context sensitive menu in order to use a different symbol for the element.

PowerFactory supports user-defined symbols as Windows-Metafile (*.wmf) and Bitmap (*.bmp) files.

9.5 Result Boxes, Text Boxes and Labels

PowerFactory uses result boxes, text boxes, and labels in the single sine diagram to display calculation results and other useful information. Figure 9.5.1 illustrates an example of this.

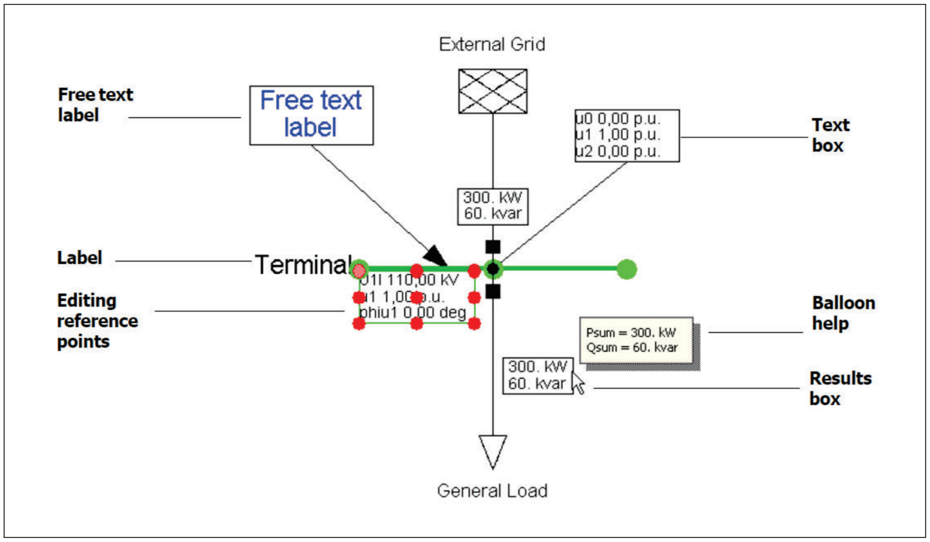


Figure 9.5.1: Results boxes, text boxes, and labels available in *PowerFactory*

9.5.1 Result Boxes

General

Result boxes are generally set up so that there are a series of different formats for each calculation function, with variables appropriate to that function. In addition, the format differs for the objects class and/or for individual objects. For example, following a load flow calculation, branch and edge elements will have different formats compared to nodes, and an external grid will have an individual, different, format as compared to the branch and edge elements.

Although the result boxes in the single line graphic are a very versatile and powerful way for displaying calculation results, it is often not possible to display a large (part of a) power system without making the result boxes too small to be read. *PowerFactory* solves this problem by offering balloon help on the result boxes. Positioning the mouse over a result box will pop up a yellow text balloon with the text displayed in a fixed size font. This is depicted in figure 9.5.1. The result box balloon always states the name of the variable, and may thus also be used as a legend.

Reference points

A result box is connected to the graphical object for which it displays the results by a "reference point". Figure 9.5.1 shows the default reference points for the result box of a terminal. A reference point is a connection between a point on the result box (which has 9 optional points), and one of the "docking" points of the graphical object. The terminal has three docking points: on the left, in the middle and on the right. The reference point can be changed by:

- Right-clicking the result box with the graphics cursor (freeze mode off), and selecting *Change Reference Points*.
- The reference points are shown: docking points in green, reference points in red. Select one of the reference points by left-clicking it.
- Left-click the selected red reference point, and drag it to a green docking point and drop it.

- An error message will result if you drop a reference point somewhere else than on a docking point.

Result boxes can be freely moved around the diagram. They will remain attached to the docking point, and will move along with the docking point. A result box can be positioned back to its docking point by right-clicking it and selecting *Reset Settings* from the menu.

If the option “Reset textboxes completely” is set in the graphical settings, then the default reference and docking points will be selected again, and the result box is moved back to the default position accordingly.

Editing Result Boxes

PowerFactory uses separate result boxes for different groups of power system objects, such as node objects (i.e. busbars, terminals) or edge objects (i.e. lines, loads). For each type of result box, a different result box definition is used.

A number of these predefined formats are available for display; they may be selected by right-clicking a result box to get the *Format for Edge Elements* (in this example) option, which then presents a number of formats that may be selected. The active format is ticked (✓) and applies for all the visualised edge elements.

It is also possible to select predefined formats for a specific element class. If the edge element is for example an asynchronous machine, in the context sensitive menu it will be also possible to get the option *Format for Asynchronous Machine*, which shows the predefined formats for the element class Asynchronous Machine (*ElmAsm*). The selected format will in this case apply only to the visualised asynchronous machines.

If the user wants to create a specific format that is different from the pre-defined ones, the *Edit Format for Edge Elements (or Node Elements)* option should be used. Note that the new format will be applied to the entire group of objects (edge or node objects).

If a created format is expected to be used for just one specific element, then the *Create Textbox* option should be used. An additional result box/ textbox will be created, using the current format for the object. This may then be edited. Information about text boxes is given in 9.5.2.

When the *Edit Format* option has been selected, the user can modify the variables and how are they showed as described Chapter 19: Reporting and Visualising Results, Section 19.2.1: Editing Result Boxes.

Formatting Result Boxes

Result boxes can be formatted by means of the context sensitive menu (right-clicking the desired result box). The available options include:

- Shift to layer (see 9.3.6).
- Rotate
- Hide
- Change the font type and size of the text
- Change the width
- Change colour
- Set the text alignment
- Adapt width
- Change reference points
- Reset Settings, only available after changes have been made).

Resetting Calculation Results

When pressed, the *Reset Calculation* icon ( ) will clear the results shown on the Single Line Diagram. By default, *PowerFactory* will also clear the calculation results when there is a change to network data or network configuration (such as opening a switch). However, if *Retention of results after network change* is set to *Show last results* in the User Settings (see Section 7.1: General Settings), results will appear in grey on the Single Line Diagram and on the Flexible Data page until the calculation is reset, or a new calculation performed. Reset Calculation can also be accessed from the main Calculation menu or using **F12**.

9.5.2 Text Boxes

As mentioned before, text boxes are used to display user defined variables from a specific referenced object within the single line graphic. To create a text box, right-click on the desired object (one end of the object when it is a branch element) and select *Create Textbox*. By default a text box with the same format of the corresponding result box will be generated.

The created text box can be edited, to display the desired variables, following the same procedure described in 9.5.1. In this case after right-clicking the text box, the option Edit Format should be selected. By default the text boxes are graphically connected to the referred object by means of a line. This “connection line” can be made invisible if the option *show line from General Textboxes....* from the *Result Boxes* page of the Graphic Option dialog (9.3.5) is disabled.

9.5.3 Labels

In the general case, a label showing the name of an element within the single line graphic is automatically created with the graphical objects (see figure 9.5.1). The label can be visualised as a text box showing only the variable corresponding to the name of the object. As for text boxes, the format of labels can be set using the context sensitive menu.

9.5.4 Free Text Labels

Free Text Labels (see figure 9.5.1) can be anchored to an element on the single line diagram, and used to display custom text. The are created by right-click and selecting ’Create Free Text Label’.

9.6 Annotations

The Annotations feature offers the user the opportunity to include additional graphical information in one or more configurable layers in the single line, geographic or block diagrams. Examples include:

- Built-in graphical annotation elements
- Text
- Icons (bitmap files)
- Plots

Note: The Annotation Layer is the default layers for new annotation objects, but users can choose to move such objects into their own user-defined layers, which can also contain other graphical objects.

The annotation elements are as follows:

- Graphical annotation
  - Line:
  - Polyline:
  - Arrow:
  - Polyline with arrow:
  - Polygon:
  - Rectangle:
  - Circle:
  - Pie:
  - Arc:
- Text:
- Icons (bitmap files):
- Plots:

Except the icons and plots, all the annotation elements can be drawn directly in the diagram. Before placing an icon in the diagram, an available icon-object has to be selected or if not yet existing, created. To insert a plot into the diagram, an already existing plot can be selected from the list in the object browser, which opens after pressing the button.

It is possible have multiple layers that contain annotations. To do this, the user should click on the button and then use the **Add layer** button to make a user-defined layer, selecting *Net elements, annotations, text boxes* from the drop-down menu for Layer Type.

Export graphical layer

To export a graphical layer the user should press the **Edit layer** button, then go to the Annotations page of the dialog. Using the **Export** button, the user can export the layer as an *.svg file.

Import graphical layer

To import a graphical layer, the user should first create the new layer as described above, then use the **Import** button on the Annotations page.

9.7 Annotation of Protection Device

Adding a protection device into the single line diagram is described in section 33.2.2.

9.8 Navigation Pane

The navigation pane provides the user an overview of the whole network in a small window. It is available for all graphics but plots. When zooming-in on a part of the grid, the navigation pane provides an overview of the whole network and highlights the part of the network that is currently being shown in the diagram. This is illustrated in figure 9.8.1.

The navigation pane can be turned on or off using the icon in the graphics toolbar. The frame within the navigation pane can be moved around in order to see different parts of the network.

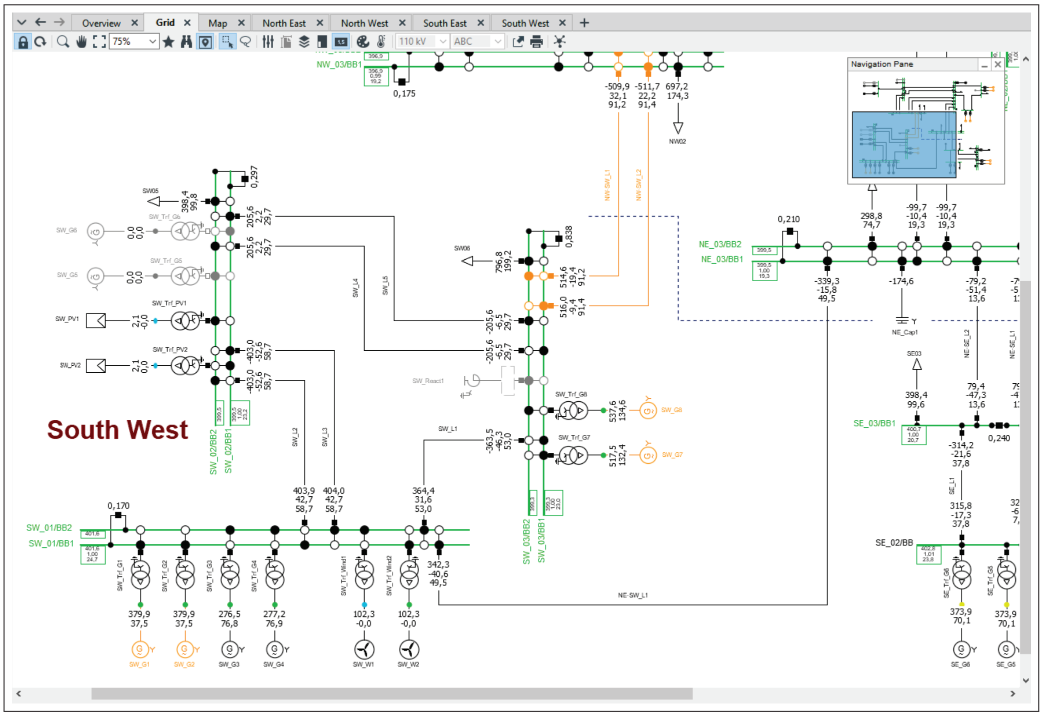


Figure 9.8.1: Navigation pane

The navigation pane is enabled for every diagram by default, but can be disabled for specific diagrams. This is done by first clicking on the *Graphic Options* icon (), then the dialog, go to the *Advanced* tab within the *Basic Attributes* and disable the option “Allow Navigation Pane”.

9.9 Graphic search facility

It is possible to search for network elements within a graphic, using the icon in the graphic toolbar. This is illustrated in figure 9.8.1 above: the search facility automatically lists possible objects as the user types.

On geographic diagrams (see section 9.10 below), it is also possible to search for places such as towns or streets.

If the user wishes to restrict the search to geographic places rather than network elements, the prefix “geo:” can be used.

9.10 Geographic Diagrams

In *PowerFactory* it is possible to specify terminal GPS coordinates, and automatically generate geographic diagrams. GPS coordinates (latitude and longitude) are entered on the *Description* page of terminals and lines, on the *Geographical Coordinates* tab. One geographic diagram can be created per project by either:

- Opening the Data Manager, right-clicking on the active project or active grid and selecting *Show Graphic → Geographic Diagram*.
- On the main menu, under *Insert → Geographic Diagram*.

The geographic diagram provides a visual representation of the network and includes all terminals and lines for which GPS coordinates have been entered.

One port elements (e.g. loads, shunts, generators) can also be represented in the geographic diagram. The Diagram Layout Tool can be used to automatically draw all the edge elements in the diagram (see section 11.6.1.2).

The settings for the geographic diagram are defined in the Graphic Options, *Geographic Diagram* page (see section 9.3.5.5).

An additional layer called *Load/Generation Distribution* is available for geographic diagrams, in order to illustrate the magnitude of network load and generation (apparent power), as illustrated in Figure 9.10.1. The *Layers* dialog is accessed using the icon, and for information about working with layers, see Section 9.3.6 (Layers). If the *Load/Generation Distribution* layer is edited, the colours and other settings can be changed on the Load/Generation page. (For general information about configuring colours, see Section 4.7.5.

Note that the displayed size of circles does not change as the user zooms in and out of the diagram.

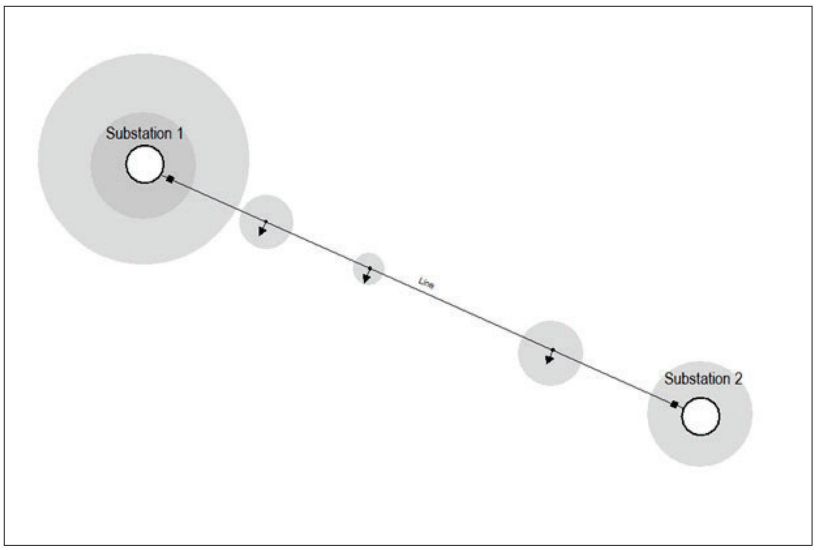


Figure 9.10.1: Load/Generation Distribution example

Maps can be used as background images and can be specified on the *Configuration* page of *Layers* ( ). Maps from the following providers are pre-configured:

- OpenStreetMap (OSM), featuring free-of-charge mapnik-style maps
- Esri ArcGIS©, including road maps, satellite, and hybrid maps
- Google Maps©[1], including road maps, satellite/aerial, hybrid, and topographic maps
- Bing Maps, including road and satellite maps
- IGN Géoportail, including road maps, satellite and special maps
- Local map files, stored in plain text-files

To use the map data of some providers, special licence keys are necessary, which can be stored in the Geographic Maps page of the configuration dialog accessed via *Tools* → *Configuration*.

[1]requires Google Maps for Business account: http://www.google.com/enterprise/mapsearth

9.10.1 Using an External Map Provider

If an external map provider from the internet is used, the *Map layer* can be chosen from (depending on which map layers the provider offers):

- Roadmap
- Satellite/Aerial
- Hybrid
- Topographic

The following parameters are available:

- Saturation adjustment
- Brightness adjustment

These parameters are valid in the range -100 % and +100 % and can be used to highlight either the map or the network elements.

Figures 9.10.2 and 9.10.3 illustrate small distribution grids where OpenStreetMap, and Esri ArcGIS© satellite maps, respectively, are used as the background image providers.

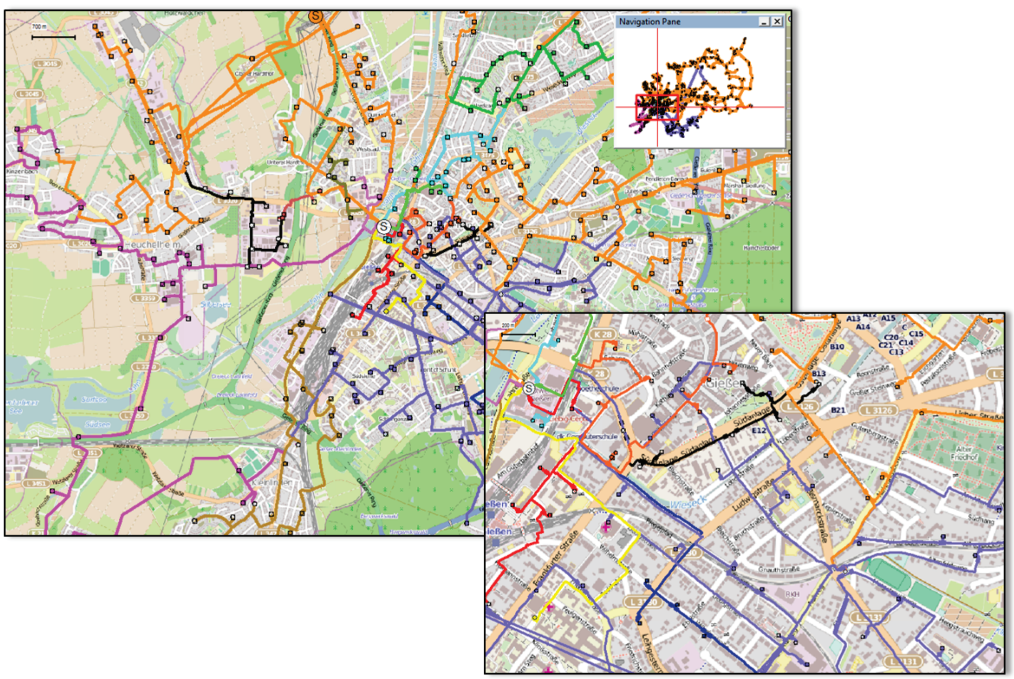


Figure 9.10.2: Network example with OpenStreetMap data

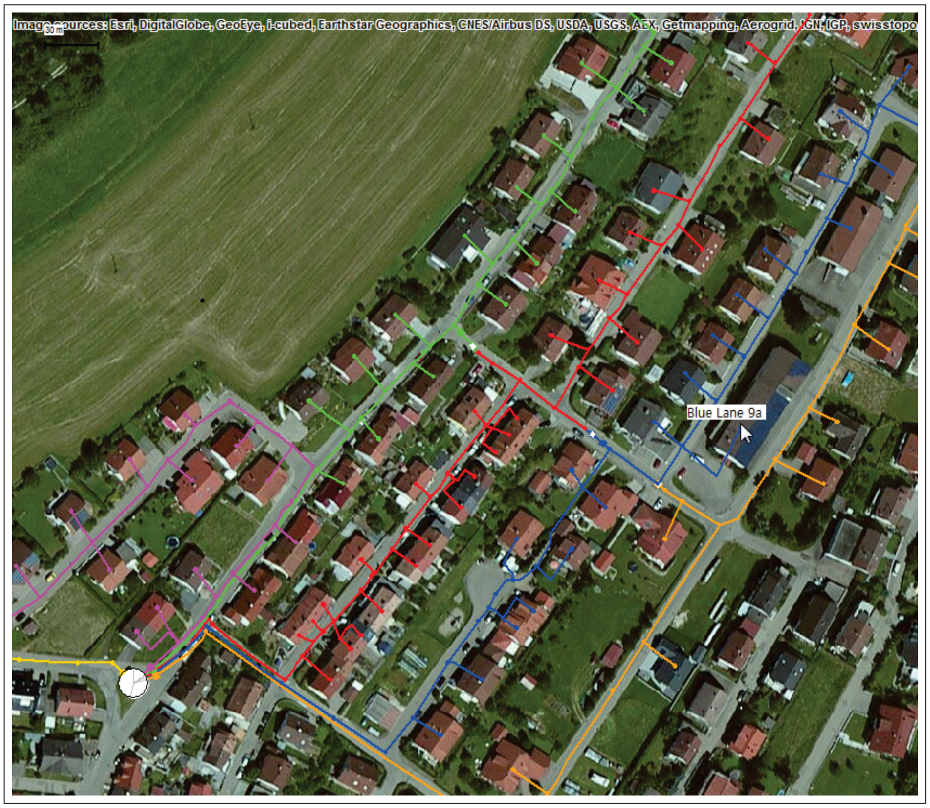


Figure 9.10.3: Network example with satellite background map (ESRI ArcGIS©)

Besides usage of pre-configured built-in map services, *PowerFactory* supports the use of user-configured map services based on the standardised WMS/WMTS protocol. The WMS are defined by the *Administrator* in the *Configuration* folder as shown in figure 9.10.4

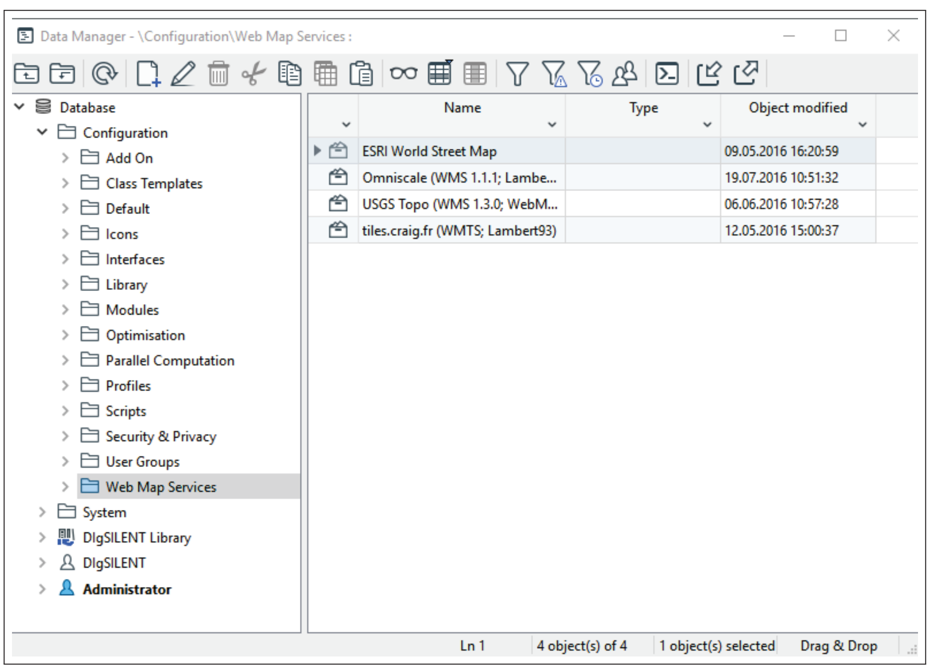


Figure 9.10.4: Web Map Services folder location

If the “Web Map Services” folder doesn’t exist, it can be created by right clicking on the *Configuration* folder and selecting *New* → *Folder*. The folder should be a *System (*DIgSILENT*)* folder with the key “MapServices” as shown in figure 9.10.5.

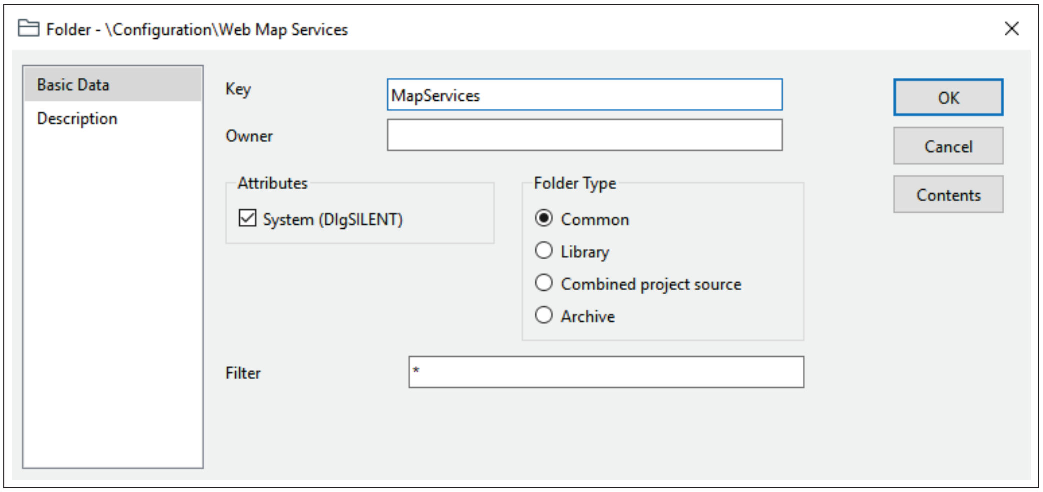


Figure 9.10.5: Web Map Services folder

A new Web Map Service can be created by clicking on the button *New* ( ) in the Data Manager and selecting *Others* → *Other Elements (Int*)* → *Hyperlink (IntUrl)*. In the edit dialog of the hyperlink, the field *Resource type* must be set to *Map Service*.

Once *Address*, *Map server protocol* and the rest of the fields of the edit dialog are set, it is possible to verify the connection to the Map Service by clicking on **View**.

Once a Web Map Service is defined by the Administrator, the user can select the desired map by configuring the *Background* layer of the geographic diagram, as shown in figure 9.10.6

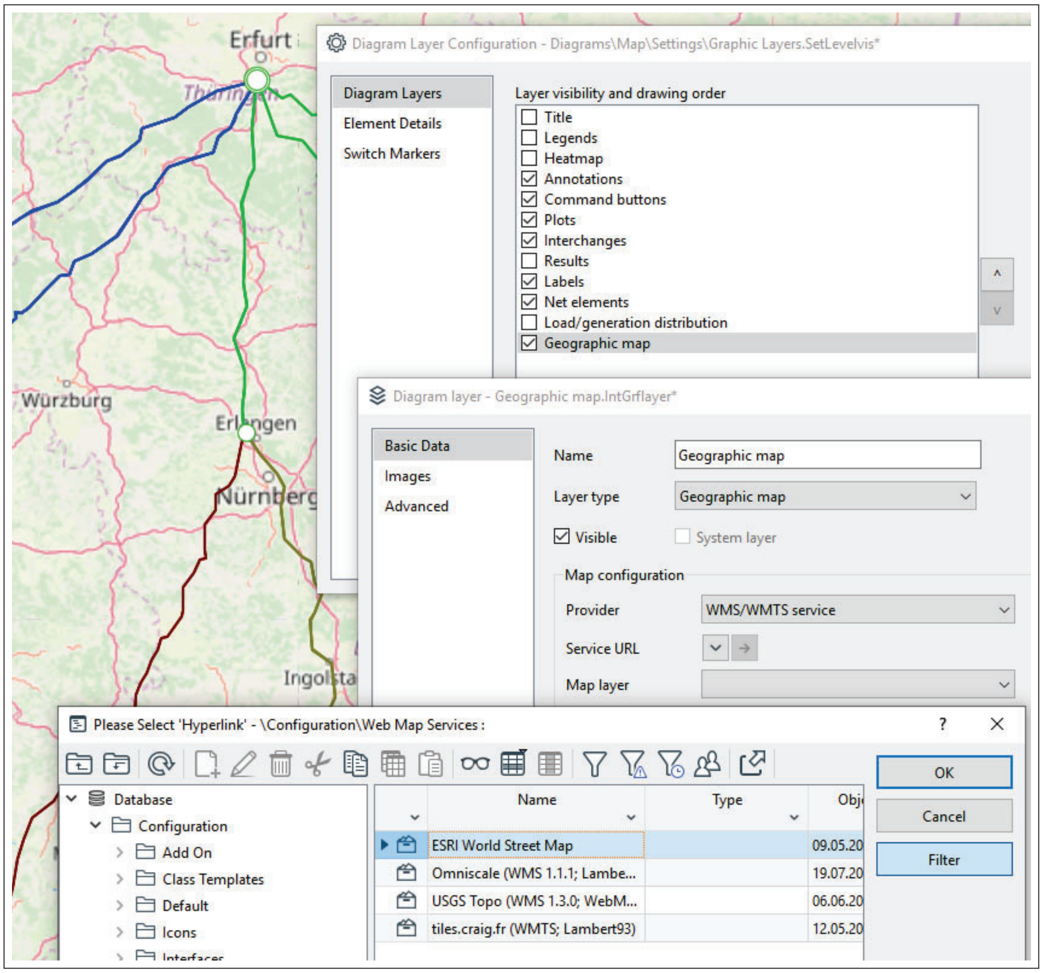


Figure 9.10.6: Background layer configuration

9.10.2 Using Local Maps

To display background images (e.g. maps) on the geographic diagram, the map provider must be selected as *Local map files* (on the Layers dialog, *Configuration* page). A *File* for reading background images must be selected. This facilitates 'tiling' of multiple images in the background of the GPS graphic if required.

The *File* is simply a text file with semicolon delimited entries, as follows:

Image_filename; X1; Y1; X2; Y2

Where:

- *Image_filename* is the name of the image file. If it is not in the same directory as the *File*, it should include the file path.
- *X* is the latitude and *Y* is the longitude.
- *(X1,Y1)* are the bottom-left coordinates of the image.
- *(X2,Y2)* are the top-right coordinates of the image.
- The # symbol can be used to comment-out entries.

Chapter 10

Data Manager

10.1 Introduction

To manage/ browse the data in *PowerFactory*, a Data Manager is provided. The objective of this chapter is to provide detailed information on how this Data Management tool. Before starting, users should ensure that they are familiar with Chapter 4 (*PowerFactory* Overview).

10.2 Using the Data Manager

The Data Manager provides the user with all the features required to manage and maintain all the data from the projects. It gives both an overview over the complete data base as well as detailed information about the parameters of single power system elements or other objects. New case studies can be defined, new elements can be added, system stages can be created, activated or deleted, parameters can be changed, copied, etc. All of these actions can be instituted and controlled from a single data base window.

The Data Manager uses a tree representation of the whole database, in combination with a versatile data browser. To initially open a Data Manager window press the icon from the main toolbar. The settings of this window can be edited using the 'User Settings' dialog (Section 10.2.7: Data Manager Settings).

The Data Manager window has the following parts (see Figure 10.2.1):

- The Data Manager local tool bar **[1]**.
- The address bar, which shows the name and path of the folder currently selected in the database **[2]**. This feature is described in Section 10.2.1
- In the left upper area the database window, which shows a symbolic tree representation of the complete database **[3]**.
- In the left lower area the input window. It may be used by more experienced users to enter commands directly, instead of using the interactive command buttons/dialogs. By default it is not shown. For further information see Section 10.7 (The Input Window in the Data Manager) **[4]**. The input window is opened and closed by the clicking on the *Input Window* button ().
- On the right side is the database browser that shows the contents of the currently selected folder **[5]**.
- Below the database browser and the input window is the status bar, which shows the current status and settings of the Data Manager (for further information see Section 10.2.7).

There are some special features of the database browser which can be accessed at any time when the content of a folder is shown:

- Balloon text: this is not only available for the buttons in the tool bar and the active parts of the status bar or the browser window, but also for the data fields **[a]**.
- Active Title buttons of each column; click on any title button to sort the items in the column; first click- items are sorted in ascending order; second click - items are sorted in descending order **[b]**.
- Object buttons showing the object standard icon in the first column of the database browser: each object is represented by a button (here a line object is shown). One click selects the object and a double-click presents the edit dialog for the object **[c]**.

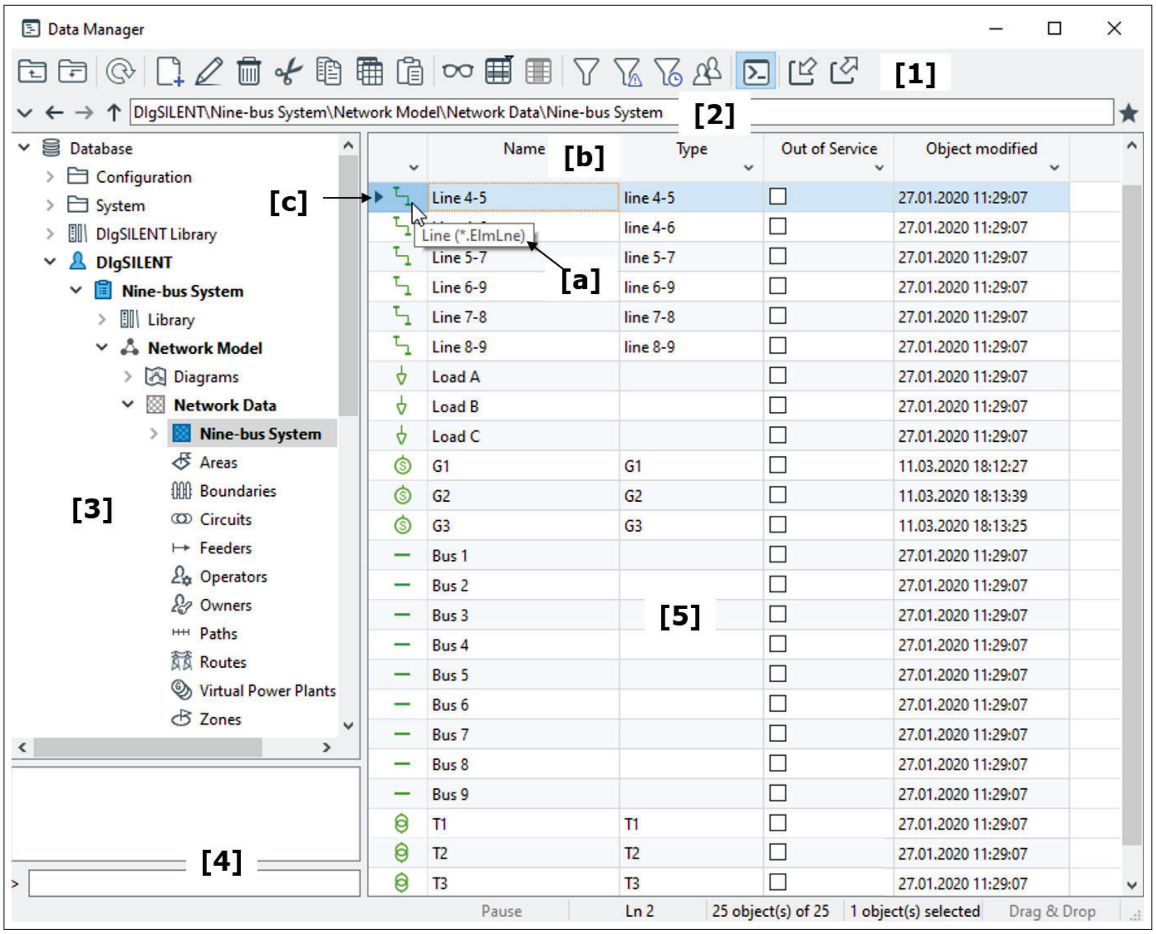


Figure 10.2.1: The Data Manager window

PowerFactory makes extensive use of the right mouse button. Each object or folder may be 'right-clicked' to pop up a context sensitive menu. For the same object the menu presented will differ depending on whether the object is selected in the left or right hand side of the Data Manager (this is known as a 'context sensitive' menu). Generally, the left hand side of the Data Manager will show object folders only. That is, objects that contain other objects inside them. The right hand side of the Data Manager shows object folders as well as individual objects.

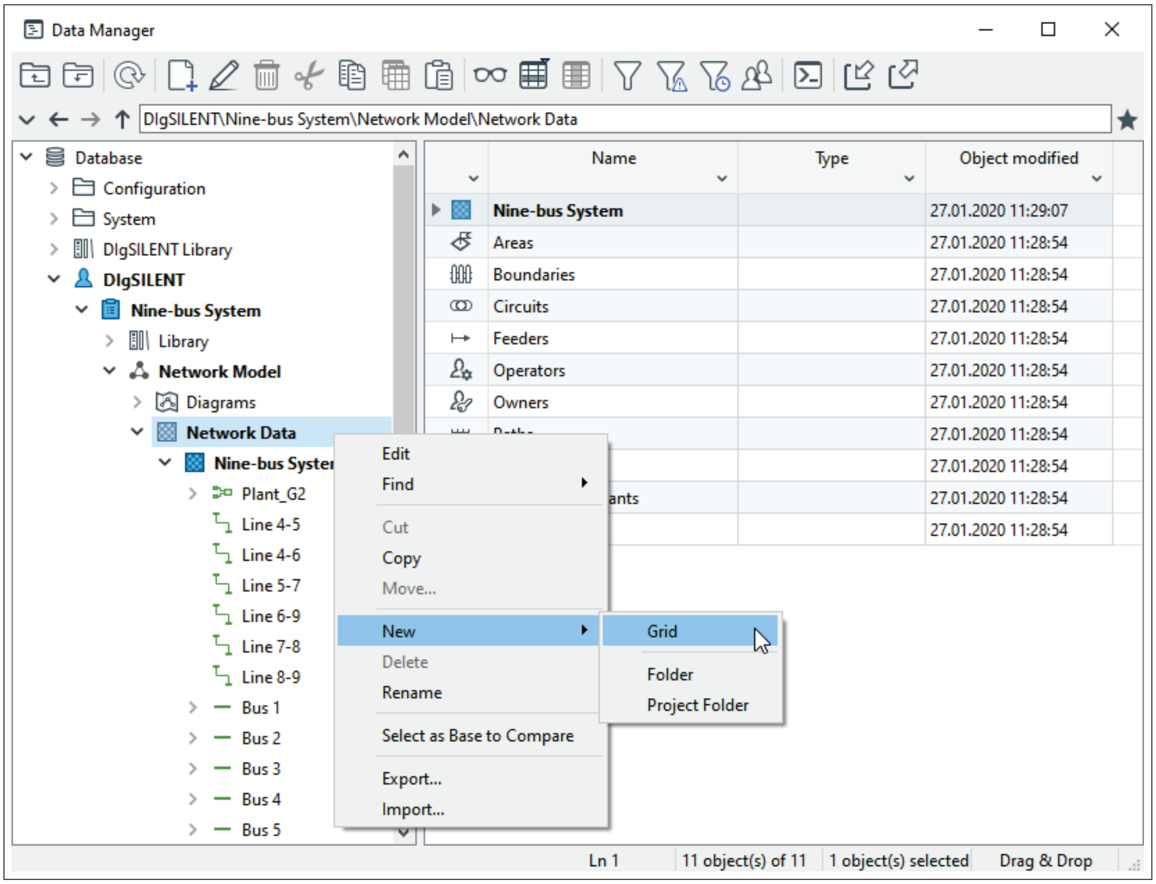


Figure 10.2.2: Context sensitive menus in the Data Manager

Using the right mouse button to access menus is usually the most effective means of accessing features or commands. Figure 10.2.2 shows an Illustration of a context-sensitive right mouse button menu.

The symbolic tree representation of the complete database shown in the database window may not show all parts of the database. The user settings offer options for displaying hidden folders, or for displaying parts that represent complete stations. Set these options as required (Section 10.2.7: Data Manager Settings).

10.2.1 Using the Data Manager address bar

The Data Manager address bar (seen in Figure 10.2.1 and labelled **[2]**) displays the file-path of the location currently selected on the left-hand side of the Data Manager, but also offers a number of features to help the user easily navigate the data structure.

Autocomplete

If the user starts typing a location in the address bar, matching locations will be listed, from which the user can select the one that is wanted.

Switch to...

On the left-hand side of the address bar there are left- and right-arrows, which can be used to switch to the previous or next selected folder, whilst the up-arrow is used to switch to the parent folder.

Favourites

A selected location can be added to (or removed from) the “favourites” list by clicking on the star-icon to the right of the address bar.

Selecting common locations

On the far left-hand side of the address bar there is a down-arrow, which gives access to several features:

- **History**: This lists the locations viewed since the Data Manager was opened.
- **Favourites**: This presents the list of favourite locations
- **Current user**
- **Active project**
- **Active study case**

10.2.2 Navigating the Database Tree

There are several ways to “walk” up and down the database tree:

- Use the mouse: all folders that have a “+” sign next to them may be expanded by double-clicking on the folder, or by single clicking the “+” sign.
- Use the keyboard: the arrow keys are used to walk up and down the tree and to open or close folders (left and right arrows). The **Page Up** and **Page Down** keys jump up and down the tree in big steps and the “-” and “+” keys may also be used to open or close folders.
- Use the toolbar in combination with the browser window. Double-click objects (see “c” in Figure 10.2.1) in the browser to open the corresponding object. This could result in opening a folder, in the case of a common or case folder, or editing the object dialog for an object. Once again, the action resulting from your input depends on where the input has occurred (left or right side of the Data Manager).
- The buttons *Up Level* () and *Down Level* () on the Data Manager tool bar can be used to move up and down the database tree.

10.2.3 Adding New Items

Generally, new network components are added to the database via the graphical user interface (see Section 11.2: Defining Network Models with the Graphical Editor), such as when a line is drawn between two nodes creating, not only the graphical object on the graphics board, but also the corresponding element data in the relevant grid folder. However, users may also create new objects “manually” in the database, from the Data Manager.

Certain new folders and objects may be created by right-clicking on folders in the Data Manager. A context sensitive menu is presented, offering a choice of objects to be created that will “fit” the selected folder. For example, right-clicking a grid folder will allow the creation (under the *New* menu) of a Graphic, a Branch, a Substation, a Site or a Folder object. The new object will be created in the folder that was selected prior to the new object button being pressed. This folder is said to have the ’focus’ for the commanded action. This means that some objects may not be possible to create since the focused folder may not be suited to hold that object.

For instance: A synchronous machine should not go into a line folder. A line folder should contain only line routes, line sections and cubicles. The cubicles in their turn should contain only switches or protection elements.

To access the whole range of objects that may be created, the icon must be pressed (*new object* icon). This is found the Data Manager toolbar and presents the dialog shown in Figure 10.2.3.

To simplify the selection of the new objects, a filter is used to sort the object list. This filter determines what sort of list will appear in the drop-down list of the 'Element' field. If "Branch Net Elements" is first selected, the selection of, for instance, a 2-winding transformer is accomplished by then scrolling down the element list.

The Element field is a normal edit field. It is therefore possible to type the identity name of the new element, like *ElmTr3* for a three-winding transformer, or *TypLne* for a line type directly into the field.

The possible list of new objects is therefore context sensitive and depends on the type or class of the originally selected folder.

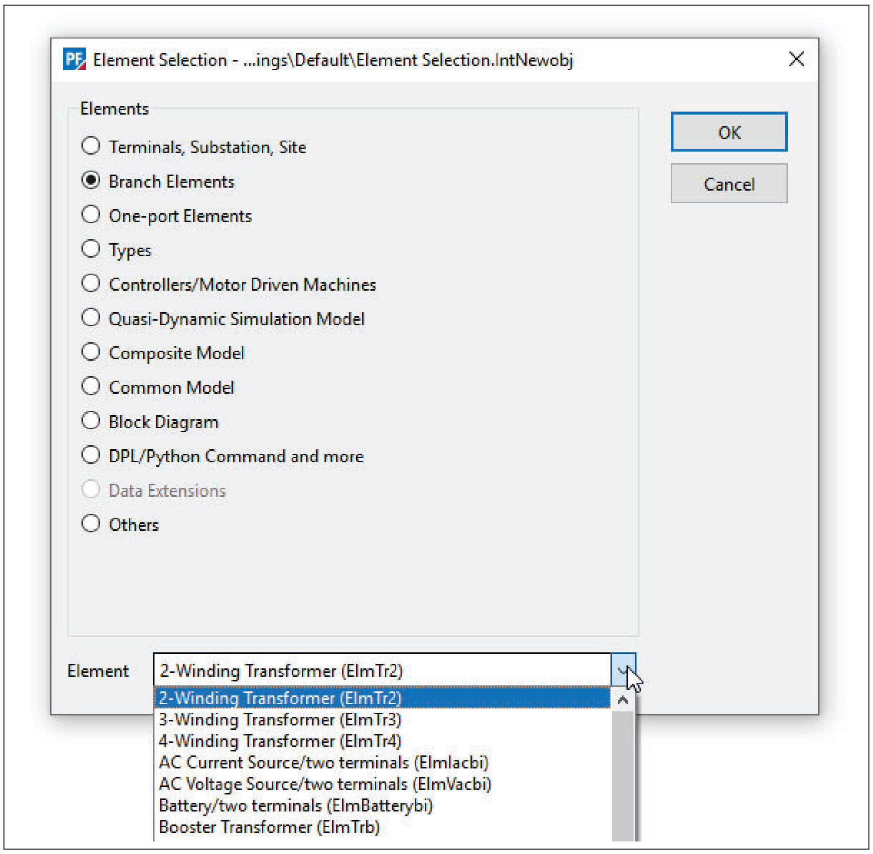


Figure 10.2.3: The element selection dialog

After the selection for a new object has been confirmed, the "Element Selection" dialog will close, the new object will be inserted into the database and the edit dialog for the new object will pop up. If this dialog is closed by pressing the **Cancel** button, the whole action of inserting the new object will be cancelled: the newly created object will be deleted from the active folder. The dialog for the new object may now be edited and the **OK** button pressed to save the object to the database.

As any other object, folders can be created either by using the context sensitive menu or by using the icon. Common folders (*IntFolder* objects) may have an owner name entered, for documentation or organisational purposes. In this way it should be clear who has created the data. Descriptions may also be added. An existing folder may be edited by using the *Edit* icon on the toolbar or by using the right mouse button.

Each folder may be set to be read-only, or to be a *PowerFactory* system folder. The folder may be a

"Common" or "Library" folder. These attributes can be changed in the edit-folder dialog. These settings have the following meaning:

- Common folders are used for storing non-type objects: electric elements, command objects, settings, projects, etc.
- Type folders are used as 'libraries' for type objects.
- System folders, which are read only folders

The use of read-only folders is clear: they protect the data. In addition, folders containing data that is not normally accessed may be hidden. Selecting the kind of folders that the user/administrator wants to be hidden is done in the user settings dialog see Chapter 7 (User Settings).

10.2.4 Deleting an Item

A folder or object which is selected may be deleted by pressing the **Delete** key on the keyboard, or by clicking the icon on the toolbar of the Data Manager.

When deleting an object on the Data Manager or in the Single Line diagram, this object will be deleted immediately from the database. Only the Undo button or **Ctrl-Z** can restore the element and its references to the original location.

Because most power system objects that are stored in the database are interconnected through a network topology or through type-element relationships, deleting objects often causes anomalies in the database consistency. Of course, *PowerFactory* knows at any moment which objects are used by which others and could prevent the user from creating an inconsistency by refusing to delete an object that is used by others.

10.2.5 Cut, Copy, Paste and Move Objects

Cut, Copy and Paste

Cutting, copying and pasting may be achieved in four different manners:

1. By using the Data Manager tool bar buttons.
2. By using the normal 'MS Windows' shortcuts:
   - **Ctrl-X** will cut a selection,
   - **Ctrl-C** will copy it,
   - **Ctrl-V** will paste the selection to the active folder.

   Cutting a selection will colour the item-icons gray. The cut objects will remain in their current folder until they are pasted. A cut-and-paste is exactly the same as moving the object, using the context sensitive menu. All references to objects that are being moved will be updated. Cancelling a cut-and-paste operation is performed by pressing the **Ctrl-C** key after the **Ctrl-X** key has been pressed.
3. By using the context sensitive menu. This menu offers a *Cut*, a *Copy* and a *Move* item. The move item will pop up a small second database tree in which the target folder can be selected. When the selected objects have been Cut or Copied, the context sensitive menu will then show a *Paste, Paste Shortcut* and a *Paste Data* item.
   - **Paste** will paste the selection to the focused folder.
   - **Paste Shortcut** will not paste the copied objects, but will create shortcuts to these objects. A shortcut object acts like a normal object. Changes made to the shortcut object will change the original object. All other shortcuts to this original object will reflect these changes immediately

- **Paste Data** is only be available when just one object is copied, and when the selected target object is the same kind of object as the copied one. In that case, Paste Data will paste all data from the copied object into the target object. This will make the two objects identical, except for the name and the connections.

4. By dragging selected objects to another folder. The 'Drag & Drop' option must be enabled first by double-clicking the 'Drag & Drop: off' message on the Data Manager's status bar. When the Drag & Drop option is on, it is possible to copy or move single objects by selecting them and dragging them to another folder. Dragging is done by holding down the left mouse button after an object has been selected and keeping it down while moving the cursor to the target/destination folder, either in the database tree or in the database browser window.

Note: When dragging and dropping a COPY of the object will be made (instead of moving it) if the **Ctrl** key is held down when releasing the mouse button at the destination folder. To enable the 'Drag & Drop' option double click the 'Drag & Drop' message at the bottom of the Data Manager window.

10.2.6 Display of multidimensional attributes

Multidimensional attributes, i.e. variables that are not defined by a single parameter but by a set or matrix of values, can be displayed from the Data Manager (and the Network Model Manager) in an additional window via double click on the variable. Figure 10.2.4 shows an example of such a multidimensional attribute.

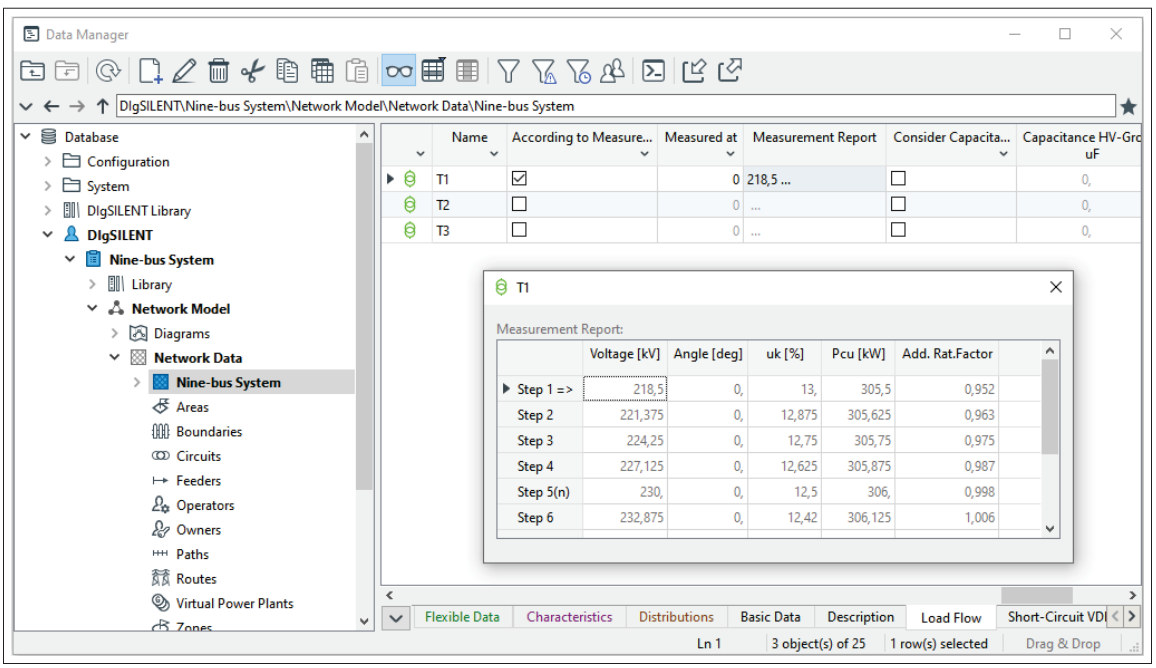


Figure 10.2.4: Displaying multidimensional attributes in the Data Manager

For the particular case of complex matrices, as depicted in Figure 10.2.5, it is also possible to select which of the variables should be displayed in the additional window.

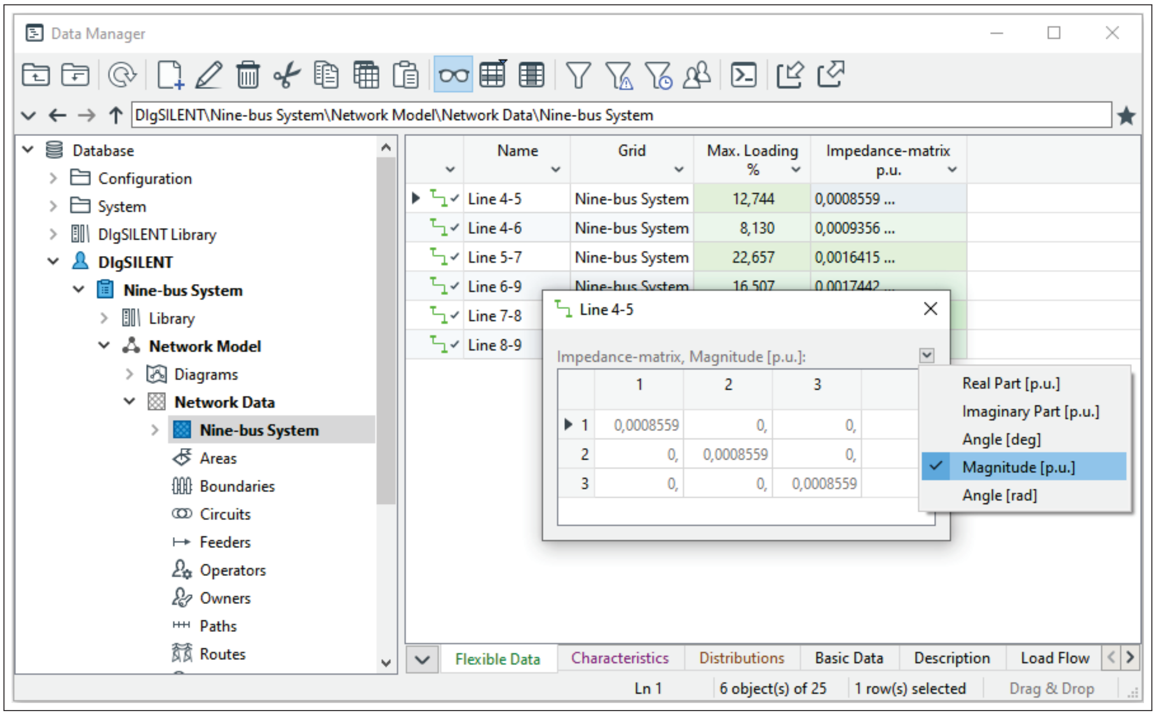


Figure 10.2.5: Displaying complex matrices in the Data Manager

10.2.7 The Data Manager Status Bar

The status bar shows the current status and settings of the Data Manager. Some of the messages are in fact buttons which may be clicked to change the settings.

The status bar contains the following messages.

- “Pause: on/off” (visible only in case of an opened input window) shows the status of the message queue in the input window. With pause on, the command interpreter is waiting which makes it possible to create a command queue. The message is a button: double-clicking it will toggle the setting.
- “N object(s) of M” shows the number of elements shown in the browser window and the total number of elements in the current folder.
- “N object(s) Selected:” shows the number of currently selected objects.
- “Drag & Drop: on/off” shows the current drag & drop mode. Double clicking this message will toggle the setting.

10.2.8 Additional Features

Most of the Data Manager functionality is available through the context sensitive menus (right mouse button).

The following items can also be found in the context sensitive menus:

Show Reference List (*Output...* → *Reference List*) Produces the list of objects that have links, or references (plus the location of the linked object), to the selected object. The list is printed to the output window. In this manner for example, a list of elements that all use the same type can be produced. The listed object names can be double- or right-clicked in the output window to open their edit dialog.

Select All Selects all objects in the database browser.

Mark in Graphic Marks the highlighted object(s) in the single line graphic. This feature can be used to identify an object.

Show → ***Station*** Opens a detailed graphic (displaying all the connections and switches) of the terminal to which the selected component is connected. If the component, is connected to more than one terminal, as might be in the case of lines or other objects, a list of possible terminals is shown first.

Goto Busbar Opens the folder in the database browser that holds the busbar to which the currently selected element is connected. If the element is connected to more than one busbar, a list of possible busbars is shown first.

Goto Connected Element Opens the folder in the database browser that holds the element that is connected to the currently selected element. In the case of more than one connected element, which is normally the case for busbars, a list of connected elements is shown first.

Calculate Opens a second menu with several calculations which can be started, based on the currently selected objects. A short-circuit calculation, for example, will be performed with faults positioned at the selected objects, if possible. If more than one possible fault location exists for the currently selected object, which is normally the case for station folders, a short-circuit calculation for all possible fault locations is made.

Other useful features:

- Relevant objects for calculations are tagged with a check-mark sign (this will only be shown following a calculation). Editing one of these objects will reset the calculation results.

10.3 Searching for Objects in the Data Manager

There are three main methods of searching for objects in the data base: Sorting, searching by name and filtering.

10.3.1 Sorting Objects

Objects can be sorted according to various criteria, such as object class, name, rated voltage,..., etc. Sorting according to object class is done using the *Open Network Model Manager...* icon on the toolbar (). A browser window will open, in which the user may select a particular class of calculation-relevant object (e.g. synchronous machine, terminal, general load, but not graphics, user settings etc.).

Further sorting can be done according to the data listed in a table- either in the Data Manager or in a browser obtained using the procedure described above. This is done by clicking on the column title. For example, clicking on the column title 'Name' in a data browser sorts the data alphanumerically (A-Z and 1-9). Pressing it again sorts the data Z-A, and 9-1.

Tabulated data can be sorted by multiple criteria. This is done by clicking on various column titles in a sequence. For example, terminals can be sorted alphanumerically first by name, then by rated voltage and finally by actual voltage by pressing on the titles corresponding to these properties in reverse-sequence (actual voltage... rated voltage... name). A more detailed example follows:

Suppose that you have executed a load flow calculation and that, for each rated voltage level in the network, you want to find the terminal with the highest voltage. These terminals could be identified easily in a table of terminals, sorted first by rated voltage and then by calculated voltage. Proceed as follows:

- Perform the load flow calculation.
- Select the *ElmTerm* class in the Network Model Manager.
- Include, in the *Flexible Data* page tab, the terminal voltage and nominal voltage (see 10.6).
- In the table (*Flexible Data* page tab), click on the title 'u, Magnitude p.u' to sort all terminals from highest to lowest calculated voltage.
- Then click on the title 'Nom.L-L Volt kV' to sort by nominal voltage level.
- Now you will have all terminals first sorted by voltage level and then by rated terminal voltage.

10.3.2 Searching by Name

Searching for an object by name is done either in the right-hand pane of the Data Manager or in a data browser. To understand the procedure below, notice that the first column contains the symbols of the objects in the table. Clicking on such a symbol selects all columns of that row, i.e. for that object. The procedure is as follows:

- Select an object in the table by clicking on any object symbol in the table (if one object was already selected then select a different one).
- Now start typing the object name, which is case sensitive. Notice how the selection jumps as you type, For example, typing 'T' moves the selection to the first object whose name starts with T, etc.
- Continue typing until the selection matches the object that you are looking for

10.3.3 Using Filters for Search

Advanced filtering capability is provided with the *Find* function. A filter is normally defined to find a group of objects, rather than individual objects (although the latter is also possible). Advanced search criteria can be defined, e.g. transmission lines with a length in the range 1 km to 2.2 km, or synchronous machines with a rating greater than 500 MW etc.

The function is available in both the Data Manager and a data browser. Clicking on the *Find* () in the Data Manager allows the user to apply a predefined filter or to define a new filter, called 'General filter'. If a new filter is defined, the database folder that will be searched can be defined.

General Filters defined by the user are objects stored in the Changed Settings \ Filters folder.

The options in the General Filter dialog window are now explained:

Name: Name of filter.

Object filter: This field defines either the complete or a part of the search criteria, and is optional. Examples are as follows:

- ***.ElmSym**: Include element objects of the class synchronous machines.
- ***.TypSym**: Include type objects of the class synchronous machines.
- **Lahney.***: Include all objects with the name Lahney.
- **Lahney.Elm***: Include all element objects with the name Lahney.
- **D*.ElmLod**: Include all load element objects whose names start with D.
- A drop down list providing various object classes can be accessed with .

Look in: This field is available if a filter id defined within the Data Manager. It allows the user to specify the folder in the database that will be searched.

Check boxes:

- *Include Subfolders* will search the root folder specified as well as the subfolders in the root folder. The search can be stopped at the matching folder.
- *Relevant Objects for Calculation* will include only those objects considered by the active study case (if no study case is active the search is meaningless and no search results will be returned).
- *Area Interconnecting Branches* will search for branch elements that interconnect grids.

The **OK** button will close the search dialog, but save the filter object to the Changed Settings\Filters folder. This makes it available for further use. The **Cancel** button will close the dialog without saving the changes. This button is useful if a search criterion (filter) will only be used once.The **Apply** button starts the actual search. It will scan the relevant folders and will build a list of all objects that match the search criteria.

Once the search is complete a list of results is returned in the form of a new data browser window. From this browser, the returned objects can be marked, changed, deleted, copied, moved, etc. . . .

Advanced search options allow more sophisticated expressions as search criteria. These are specified in the *Advanced* page of the General Filter dialog (Figure 10.3.1). The filter criterion is defined in terms of a logical expression, making use of parameter names. Objects will be included in the data browser if, for their parameters, the logical expression is determined to be true. An example of a logical expression is $dline > 0.7$. The variable *dline* refers to the length of a transmission line, and the effect of such a filter criterion is to limit the data in the browser to transmission lines having a length exceeding 0.7 km. The logical expressions can be expanded to include other relations (e.g. $>=$), standard functions (e.g. **sin()**), and logical operators (e.g. **.and.**).

Note: Parameter names can be object properties or results. The parameter names for object properties are found, for example, by letting the mouse pointer hover over an input field in an object's dialog window. Parameter names for result variables are found from variable sets, which are described in Section 19.3 Variable Selection.

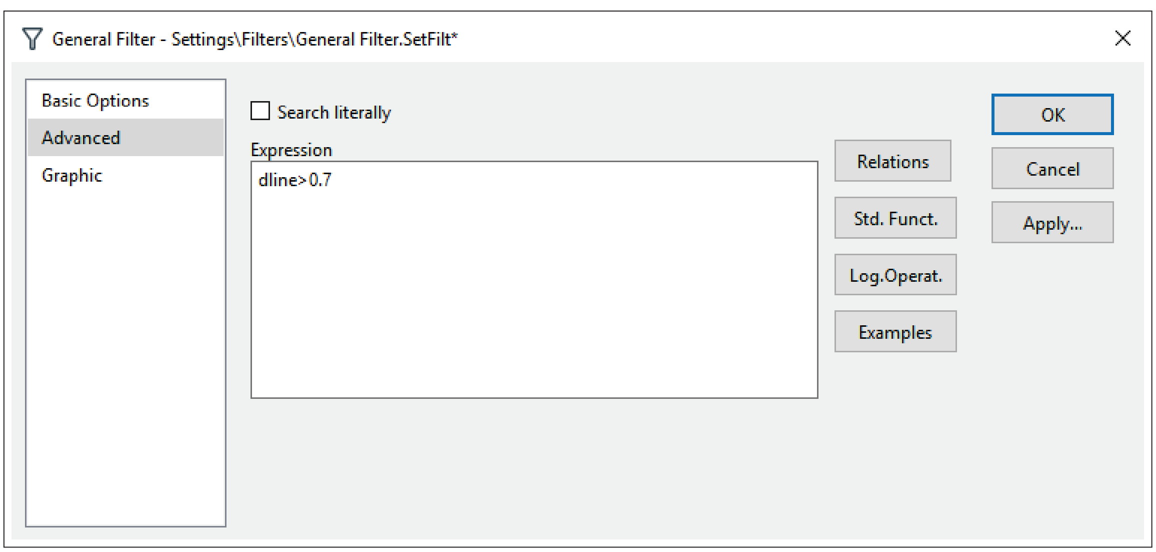


Figure 10.3.1: Filter dialog - Advanced page

Search literally is used to search for user defined strings 'inside' parameter fields. The user can specify if the search is done in a specific parameter, if the field *in Parameter* is left blank, all parameter fields will be searched for this string.

As stated before, the objects matching the filter criteria are displayed in a data browser. They may also be highlighted in the graphic using the 'Colour representation' function described in Chapter 9: Network Graphics (Single Line Diagrams). The colour to be used in this case can be specified under the page *Graphic* of the General Filter dialog.

Note: New filters are saved to the Project\Changed Settings\Filters folder in the project and are available for use directly, using the right mouse menu. If a search is to be performed in a particular grid simply proceed as follows: right-click the *grid folder* → *Find*→ *Local Filters*→ *Filter Name* (e.g. Lines longer than 700 m). Remember to press the **Apply** button to perform the search. If you unchecked the *Show Filter Settings before Application* box under *User Settings* → *General* then the filter will be applied as soon as it is selected from the menu. This is useful when you have already defined several filters for regular use.

10.4 Auto-Filter functions in Data Manager and browser windows

Columns within the Data Manager, browser windows or the Network Model Manager can be filtered. To add a filter for a column proceed as follows:

- Click on the down arrow in the column header of the table. A window will open.
- A list of all entries that differ from each other within that column, will be shown.
- Option: *Selection*
  - Select/deselect desired/undesired entries from that list. Only objects, which contain marked entries will later be shown in the table. All other ones will be hidden.
  - Sometimes the list of entries can be very long. To reduce this list, the user may enter a text into the search field. The list will be filtered with every typed character.
- Option: *Custom...*
  - In addition to the possibility of selecting existing entries, also custom filters can be used by choosing the radio button *Custom...* and confirming with **OK**.
  - A new window will open, in which up to two filter can be combined via an AND or an OR relation. For each of both filters one of the following criterions has to be chosen:
    - None
    - Equals
    - Is not equal to
    - Contains
    - Does not contain
    - Starts with
    - Does not start with
    - Ends with
    - Does not end with
    - Special
  - After one criterion has been selected (except: None), a drop-down box will appear, from which one of the entries may be chosen or an own text may be entered.

Filtered columns are indicated by a blue column heading and a filter symbol in the lower right corner of the column header. By holding the mouse cursor still over the heading of a filtered column, a balloon help appears and shows the applied filter settings.

The auto filters in the Data Manager and browser windows are temporary. They will be lost, if for example another path is chosen.

10.5 Editing Data Objects in the Data Manager

The Data Manager offers several ways to edit power system components and other objects stored in the database, regardless they appear graphically or not.

The basic method is to double-click the object icons in the database browser. This will open the same edit dialog window obtained, when double clicking the graphical representation of an element in the graphic window.

An open edit dialog will disable the Data Manager window from which it was opened. The edit dialog has to be closed first in order to open another edit dialog.

However, it is possible to activate more than one Data Manager (by pressing the icon on the main toolbar) and to open an edit dialog from each of these Data Managers. This can be useful for comparing objects and parameters.

Using the edit dialogs has one major drawback: it separates the edited object from the rest of the database, making it impossible to copy data from one object to the other, or to look at other object parameter values while editing.

PowerFactory brings the big picture back in sight by offering full scale editing capabilities in the Data Managers browser window itself. The browser window in fact acts like a spreadsheet, where the user can edit and browse the data at the same time. The browser window has two modes in which objects can be edited,

- Object mode
- Detail Mode

which are described in the following sections.

10.5.1 Editing in Object Mode

In the general case the icon, the name, the type and the modification date (with its author) of the objects are shown in the 'object' mode. Certain objects, for example network components, show additional fields like the "Out of Service" field.

The title buttons are used to sort the entries in the browser. The visible data fields can be double-clicked to edit their contents, or the **F2** button can be pressed. The object will show a triangle in its icon when it is being edited.

After the data field has been changed, move to the other fields of the same object using the arrow-keys or by clicking on these data fields, and alter them too.

The new contents of a data field are confirmed by pressing the **Return** key, or by moving to another field within the same object. The triangle in the icon will change to a small star to show that the object has been altered. The object itself however has not been updated. Updating the changes is done by pressing **Return** again, or by moving to another object in the browser. By default, *PowerFactory* will ask to confirm the changes. See Section 10.2.7 (Data Manager Settings) to disable these conformation messages.

10.5.2 Editing in "Detail" Mode

If the icon on the browse window of the Data Manager is pressed, the browser changes to 'detail' mode. It will display only the objects from the same class as the one which was selected when the button was pressed.

In 'detail' mode, the browser shows all data fields for the selected calculation function data set, which can be selected by clicking on a tab shown at the bottom of the table view.If a page tab is out of reach, then the page tab scrollers will bring it within the browser window again. As depicted in Figure 10.5.1, the Flexible Data, Scenarios, Characteristics and Distribution tabs are highlighted by colouring. The colours reflect those selected in the User Settings (refer to section 7.8) for these types of data.

The list of objects may be sorted by any column by pressing the title field button. The widths of the data fields can be adjusted by pointing the mouse on the separation line between two title fields and dragging the field border by holding a mouse button down.

As with the browser in 'object' mode, the data fields can be edited by double-clicking them. In the example the active power settings are being edited, but from the star in the object icon it is clear that another field of the same object has been edited too, but not confirmed, because this star would otherwise be a triangle.

It is possible to change a parameter field for more than one object simultaneously. This is, for instance, useful to raise a certain limit for a range of objects, in order to get a better load-flow result i.e. by alleviating line overloads. An example is shown in Figure 10.5.1 where the derating factor for a range of lines is changed at once.

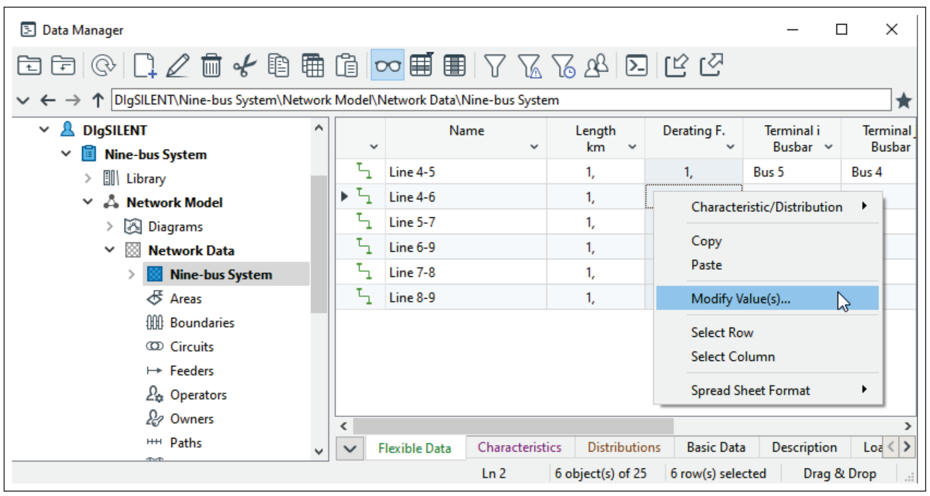


Figure 10.5.1: Modify values dialog

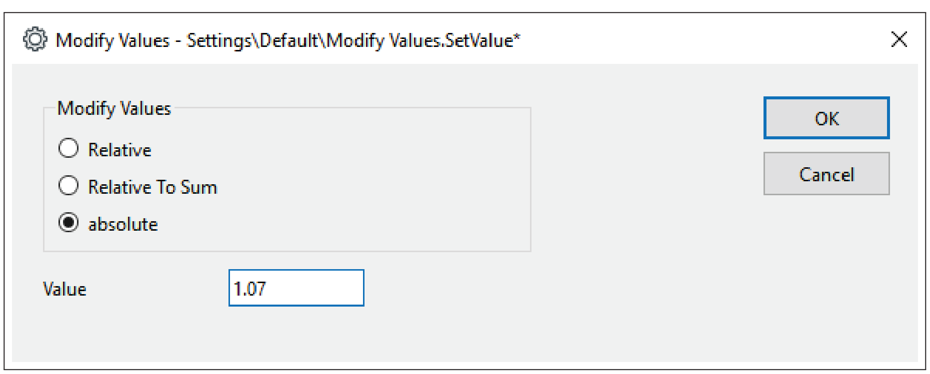


Figure 10.5.2: Modify values dialog

The parameter fields which have to be changed have to be multi-selected first. Right-clicking the selection will pop up a case sensitive menu from which the *Modify Value(s)* option opens the *SetValue* dialog, see Figure 10.5.2.

This dialog can be used to:

- increase or decrease them by multiplication with a scale factor (“Relative”).
- increase or decrease them by multiplication with a scale factor with respect to the sum of values selected (“Relative to Sum”).
- Set all the selected parameter fields to a new fixed (“absolute”) value.

It is not possible to simultaneously alter parameter fields from more than one column, i.e. to change rated currents and nominal frequencies simultaneous, even if they would happen to take the same value or would have to be raised with the same percentage.

10.5.3 Copy and Paste while Editing

One of the great advantages of editing data fields in the Data Manager’s browser window is the possibility to copy data from one object to another. This is done by selecting one or more objects or object fields, copying this selection to the clipboard, and pasting the data back in another place.

To copy one or more objects,

1. Open the Data Manager and select the grid folder where you find the objects to be copied.
2. Select the objects.
3. Press **Ctrl-C** to copy or use the icon on the Data Manager toolbox.
4. Press **Ctrl-V** to paste or use the icon on the Data Manager toolbox. The objects will be copied with all the data. Their names will automatically be altered to unique names.

Copying data fields from one object to another is done just like for any spreadsheet software you may be familiar with. To copy one or more data fields,

1. Select them by clicking them once. Select more data fields by holding down the **Ctrl** key.
2. Copy the fields to the clipboard by pressing **Ctrl-C** or the icon.
3. Select one or more target objects data fields. If more than one field was copied, make sure that the target field is the same as the first copied data field.
4. Press **Ctrl-V** or the icon. The contents of the data fields will be copied to the target objects.

10.6 The Flexible Data Page

The data browser (this will be seen in the Data Manager when the ’Detail Mode’ has been engaged) has page tabs for all calculation functions. These tabs are used to view or edit object parameters which are categorised according to a calculation function and have a fixed format.

The ’Flexible Data’ tab, normally used to display calculation results, allows the user to define a custom set of data to be displayed.

The default format for the calculation results displayed in the flexible page depends on the calculation performed: Following a load-flow calculation, the default variables for terminals are line-to-line voltage, per unit voltage and voltage angle. Following a short-circuit calculation the default variables are initial short-circuit current, initial short-circuit power, peak current etc. Figure 10.6.1 shows an example of the flexible data page tab.

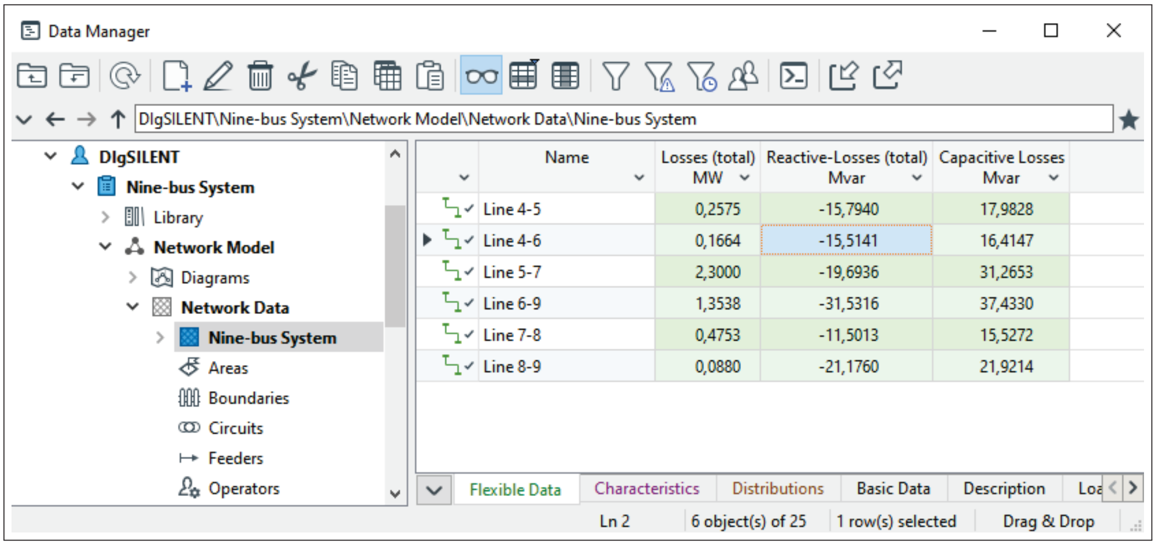


Figure 10.6.1: The Flexible Data page tab

10.6.1 Customising the Flexible Data Page

The displayed variables are organised in 'Variables Sets' that are, in turn, organised according to the calculation functions. For example, an object class *ElmTr2* (two-winding transformer) has a variable set for symmetrical load flow calculation, a variable set for short-circuit calculation etc. There may also be more than one variable set for any calculation function. For example, the object *ElmTr2* may have two variable sets for symmetrical load flow calculation.

The Flexible Page Selector allows the user to specify the variable set to use, or to define new variable sets. Furthermore, the Flexible Page Selector allows the user to access and edit the variable sets, i.e. to specify which variables to display in the Flexible Data page.

The 'Flexible Page Selector' dialog is shown in Figure 10.6.2. This dialog is opened by pressing the ( ) icon on the Data Manager toolbar. The Flexible Page Selector has a menu with all the different calculation functions. It opens in the page corresponding to the most recent calculation.

The selection of variables within Variable Sets is presented in detail in Section 19.3 Variable Selection.

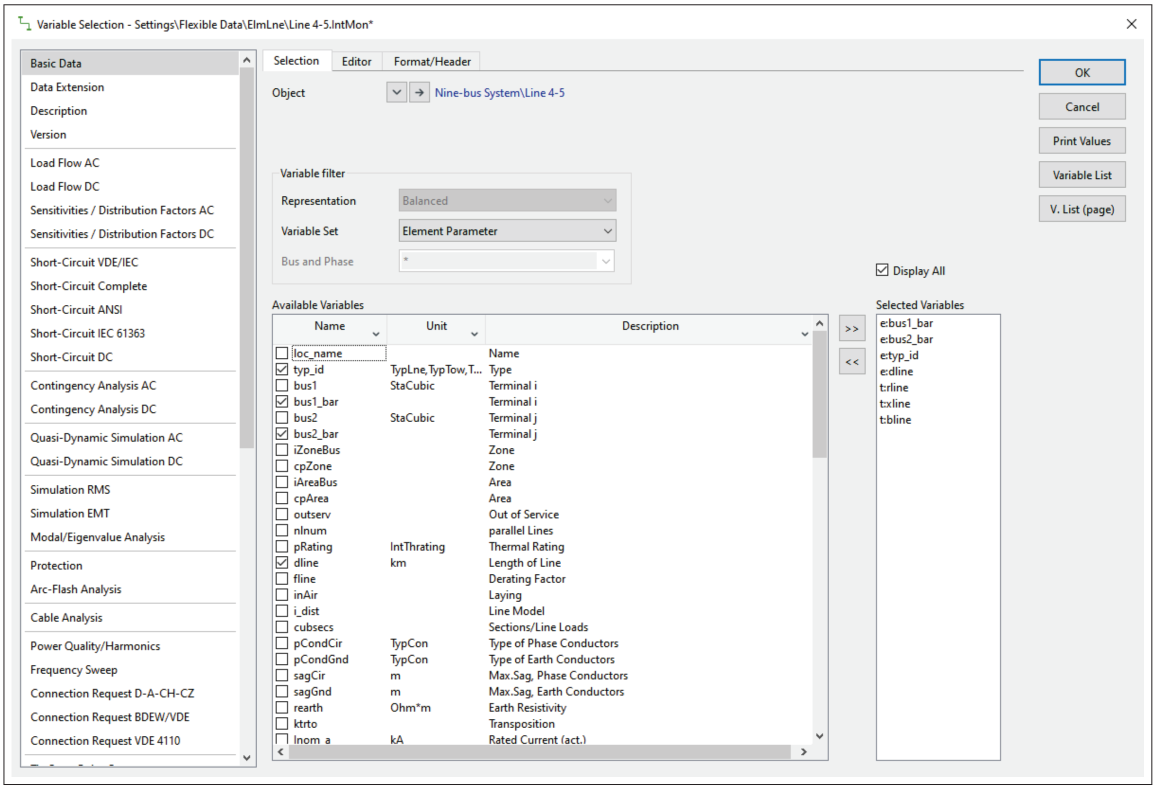


Figure 10.6.2: The Flexible Page Selector

The Format/Header tab (Figure 10.6.3) allows the user to customise the formats or column headers of the Flexible Data page.

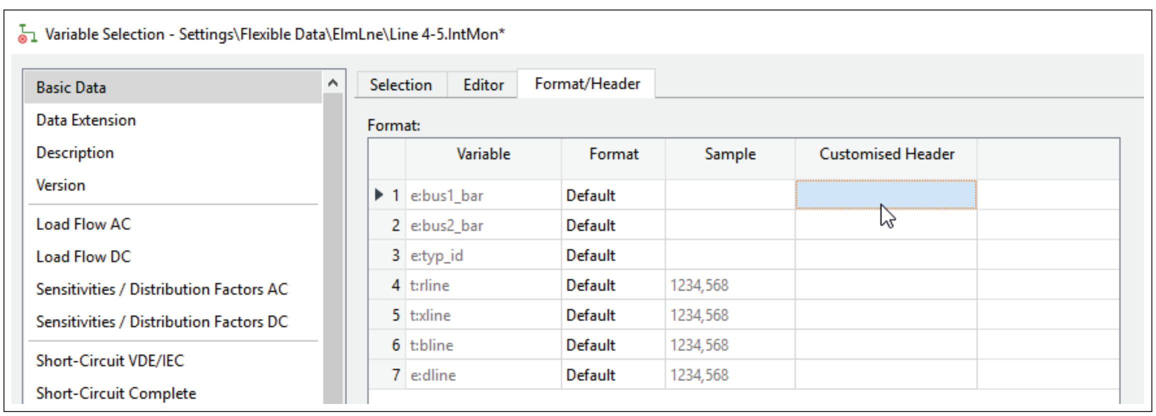


Figure 10.6.3: Customising flexible data page formats and column headers

---

Note: Variable Sets are objects of class *IntMon*, within *PowerFactory* they have multiple uses. This section only presents their use in conjunction with Flexible Data. For further information refer to Section 19.3 Variable Selection.

The number format per column in the Flexible Data Page can also be modified by right clicking on the column header of the variable and selecting *Edit Number Format . . .* . A new window will appear and the user may define the number representation.

The order of the columns (except: *Name*, *In Folder* and *Grid*) on the Flexible Data page can be changed. Therefor, the header of a column has to be clicked and while holding the left mouse button pressed, the column can be moved to the desired position. To illustrate, where the column will be placed, when the mouse button is released, an arrow between the actual and the possible new position of the column is shown during this process.

10.7 The Input Window in the Data Manager

The input window is for the more experienced users of DIgSILENT *PowerFactory*. It is closed by default. Almost all commands that are available in *PowerFactory* through the menu bars, pop-up menus, icons, buttons, etc., may also be entered directly into the input window, using the *PowerFactory* commands.

The contents of the input window can be saved to file, and commands can be read back into the window for execution.

PowerFactory also has special command objects which carry one single command line and which are normally used to execute commands. In this way, complex commands can be saved in the same folder as the power system for which they were configured.

10.7.1 Input Window Commands

In principle, everything that can be done in DIgSILENT *PowerFactory* can be done from the command line in the input window. This includes creating objects, setting parameters, performing load-flow or short-circuit calculations.

Some commands that are available are typically meant for command line use or for batch commands. These commands are rarely used in another context and are therefore listed here as "command line commands", although they do not principally differ from any other command.

Cd Command Moves around in the database tree by opening another folder at a relative position from the currently open folder.
Example:
`cd...\gridB\Load1`

Cls Command Clears the output or input window.

cls/outclears output window
cls/inpclears input window completely
cls/inp/doneclears only previously executed commands
.../y asks for confirmation

Dir Command Displays the contents of a folder.
Example:
`dir Study Case`

Ed Command Pops up the dialog of a default command, i.e. "ldf", "shc", etc.
Example:
ed ldf

Exit Command Queries or sets a variable.
Example:
`man/set obj=Load_1.elmlod variable=plini value=0.2`

Pause Command Interrupts the execution of the command pipe until a next pause command is executed.

Pr Command Prints either the contents of the output window or the currently active graphics window.

Rd Command Opens and reads a file.

Stop Command Stops the running calculation.

Wr Command Writes to a file.

10.8 Save and Restore Parts of the Database

A selected part of the database can be written to a ".pfd" file with the button **Export Data...** . This will bring a 'File Save' dialog where a filename must be specified.

Alternatively, the folder or object that is to be exported can be right-clicked in the database tree, after which the option *Export...* is selected.

The exported part of the database may be a complete project, a library, or a specific object in the browser window. Exporting a folder (i.e a project, grid, library, etc.) will export the complete content of that folder, inclusive subfolders, models, settings, single line graphics, etc.

It is even possible to export a complete user account. However, only the administrator is able to ***import*** an user-account. Exporting the user-account on a regular basis is a practical way to backup your data.

It is even possible to export data from another user account, or even to export another user-account completely. However, only the shared, visible, data will be exported.

The exported data file can be imported into the database again in any desired folder by pressing the *Import Data...* button. This will bring a 'File Open' dialog where the ".pfd" data-file can be selected.

The ".pfd"-file will be analysed and error messages will be displayed when the file is not a genuine *PowerFactory* data file, or if it is corrupted. If the file format has been found to be correct, a dialog will appear which shows the data and version of the file. The default target folder is shown also, which is the original folder of the saved data. If this is not desired, another target folder can be selected by pressing the **Drop Down** button. This button will bring a small version of the database tree. A new target folder can be selected from this tree.

10.8.1 Notes

By exporting a folder from the database, only the information in that folder and all its subfolders will be stored. If the exported objects use information (e.g. power system types like line or transformer types) that is saved somewhere else, then that information will not be stored. Make sure that the used power system types and all other referenced information is exported too.

When importing a file that contains objects which use data outside the import-file, a search for that data is started.

For instance, assume a project is exported. One of the line-models uses a type from a library outside the project. When exporting, the path and name of this type is written in the export-file, but the type itself is not exported, as is does not reside in the exported project.

At importing, the stored path and name of the 'external' type is used to find the type again and to restore the link. However, if the 'external' type is not found, then it will be created, using the stored path and name. Of course, the created object has default data, as the original data was not exported. Additionally, an error message is written to the output window.

Suppose that you are working with a large library, which is stored in a special user-account to make it read-only. The library is made accessible by sharing it to all users.

When export the projects, the objects from the external library are not exported. However, a colleague which has access to the same library may still import your projects without problems. The external objects used in your projects will be found in the same location, and the links to these objects will be correctly restored.

10.9 Spreadsheet Format Data Import/Export

The *PowerFactory* data browser in the Data Manager's window looks and acts like a spreadsheet program as far as creating and editing power system objects is concerned. To enable and simplify the use of power system element data which is stored in spreadsheet programs such as the Microsoft Excel or the Lotus 123 programs, the data browser offers 'Spreadsheet Format' import and export facilities.

10.9.1 Export to Spreadsheet Programs (e. g. MS EXCEL)

All data visible in the data browser may be exported as it is. The export format is such that most common spreadsheet programs can read in the data directly (space separated ASCII). Exporting data is performed as follows.

- Select a range of data in the data browser. Such a range may contain more than one column and more than one row.
- Right-click the selected range.
- Now you have different options:
  - If you want to copy the content of the marked cells only, simply select Copy from the context-sensitive menu.
  - If you want to copy the content of the marked cells together with a description header, select the *Spread Sheet Format* option. This opens a second menu which offers the choice between writing the Spreadsheet export to a file (*Write to File*), or to put it on the Windows Clipboard (*Copy (with column headers)*). See Figure 10.9.1.
- The exported data can now be pasted into a spreadsheet program.

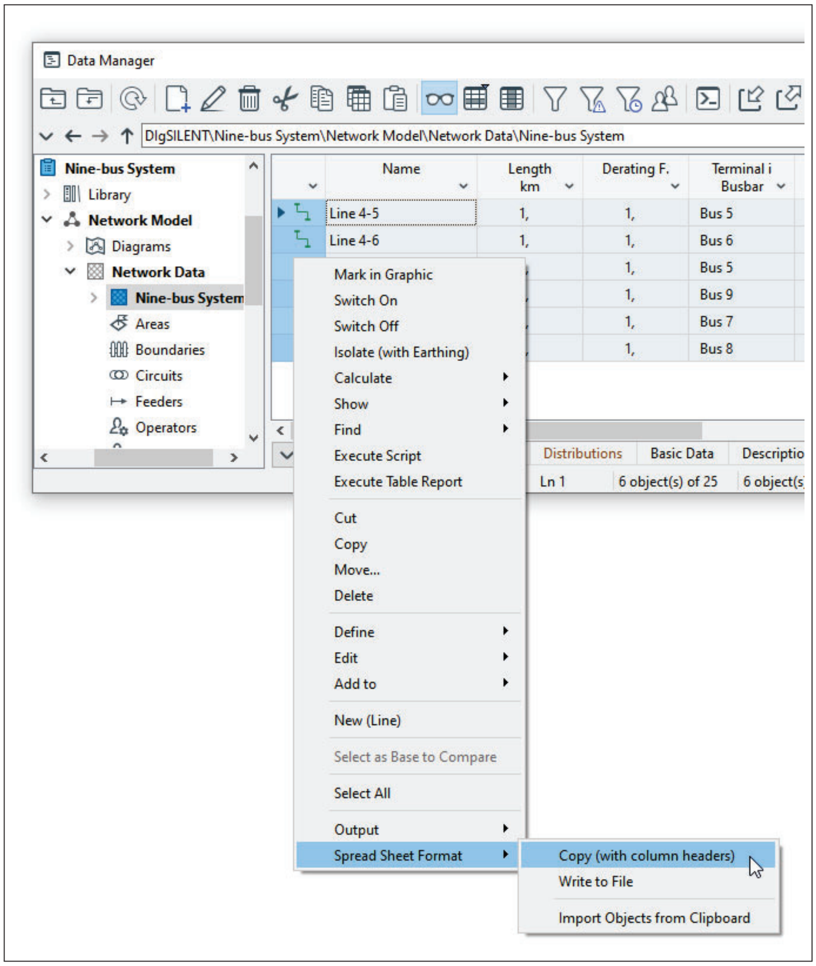


Figure 10.9.1: Exporting a range of data

10.9.2 Import from Spreadsheet Programs (e. g. MS EXCEL)

There are two methods available for importing data from a spreadsheet program. The first method uses a direct import of 'anonymous' numerical data, i. e. of the values stored in the cells of the table. This method is used to change parameter of existing objects by importing columns of parameter values.

The second method can be used to create new objects (or replace whole objects) by importing all the data from a spreadsheet.

Any range of parameter values can be copied from a spreadsheet program and imported into the Data Manager. The import is performed by overwriting existing parameter values by 'anonymous' values. The term 'anonymous' expresses the fact that the imported data has no parameter description. The size of the imported value range and the required data are tested. Importing invalid values (i.e. a power factor of 1.56) will result in an error message.

Spreadsheet Import of Values

The import of values (anonymous variables), i. e. cells of a table, is explained by the following example.

In Figure 10.9.2, a range of active and reactive power values is copied in a spreadsheet program. In Figure 10.9.3, this range is pasted to the corresponding fields of 6 load objects by right-clicking the upper left most field which is to be overwritten.

In contrast to the import of whole objects, the anonymous import of data does not need a parameter description. This would complicate the import of complete objects, as the user would have to enter all parameters in the correct order.

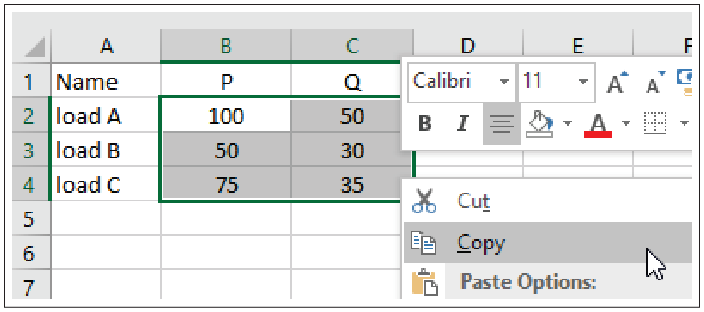


Figure 10.9.2: Copying a range of spreadsheet data

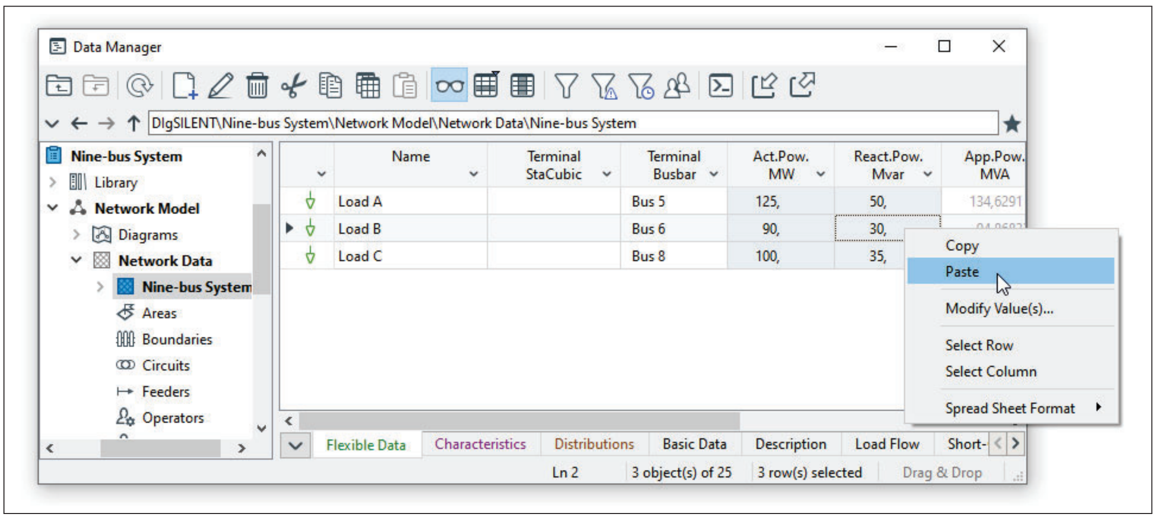


Figure 10.9.3: Pasting spreadsheet data from clipboard

Spreadsheet Import of Objects and Parameters

With this kind of import, it is possible to import whole objects (in contrast to the import of pure values, which is described above). The object import uses a header line with the parameter names (which is necessary in addition to the cells with the pure values). This header must have the following structure:

- The first header must be the class name of the listed objects.
- The following headers must state a correct parameter name.

This is shown in Figure 10.9.4.

	A	B	C	D	E
1	**Class Name**	**Param. Name 1**	**Param. Name 2**		**Param. Name N**
2	Name1	value	value	...	value
3	Name2	value	value	...	value
4	...				
5	NameM	value	value	...	value

Figure 10.9.4: Excel required format

Figure 10.9.5 shows an example of valid spreadsheet data of some line types and some 2-winding transformer types.

	A	B	C	D	E	F	G	H	I
3	**Line Types**								
4	**TypLne**	**uline**	**sline**	**frnom**	**aohl_**	**rline**	**xline**	**rlin0**	**xline0**
5	NKBA 3x120/70 1.00 kV	1	0.32	50	cab	0.157	0.083	0.261	0.97
6	NKBA 3X 25/ 16 1kV-TT	1	0.133	50	cab	0.724	0.092	0.292	1.672
7	NAYY 4x95 1.00 kV	1	0.211	50	cab	0.321	0.082	0.261	1.284
8	NAYY 4x70 1.00 kV	1	0.176	50	cab	0.444	0.082	0.261	1.776
9	NAYCWY 4x95/50 1.00 kV	1	0.211	50	cab	0.321	0.082	0.261	1.284
10	NAYCWY 4x70/35 1.00 kV	1	0.176	50	cab	0.444	0.082	0.261	1.776
11	NAYCWY 4x50/25 1.00 kV	1	0.142	50	cab	0.642	0.083	0.264	2.568
12									
13	**2-Winding Transformer Types**								
14	**TypTr2**	strn	utrn_h	utrn_l	uktr	pcutr	uk0tr		
15	0.4MVA 110/10kV	0.4	110	10	12	6	14		
16	10MVA 110/10kV	10	110	10	12	6	14		
17	TR_LV- 800/10	0.8	10	0.4	4	7	6		
18	TR_MV- 40/10	40	110	10	12	150	14		

Figure 10.9.5: Example of valid spreadsheet data

The import of the spreadsheet data into *PowerFactory* is performed as follows.

- Select the header line and one or more objects lines.
- Copy the selection. See Figure 10.9.6 for example.
- Right-click the folder browser in the Data Manager to which the objects are to be imported. Select *Spread Sheet Format* → *Import Objects from Clipboard*. See Figure 10.9.7 for example.

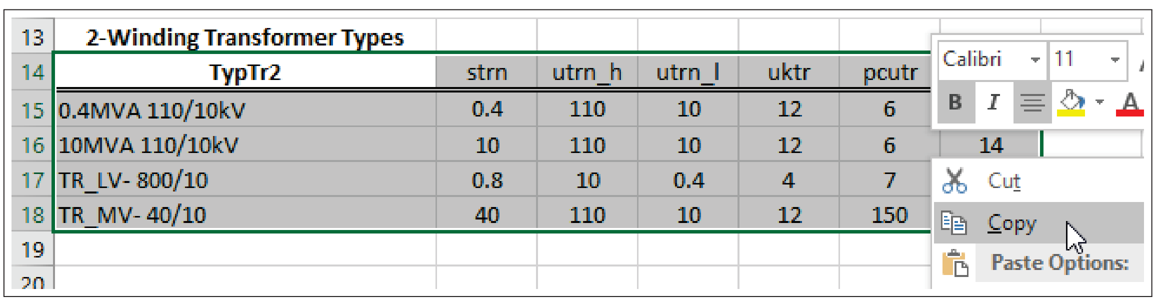


Figure 10.9.6: Selecting object data in spreadsheet

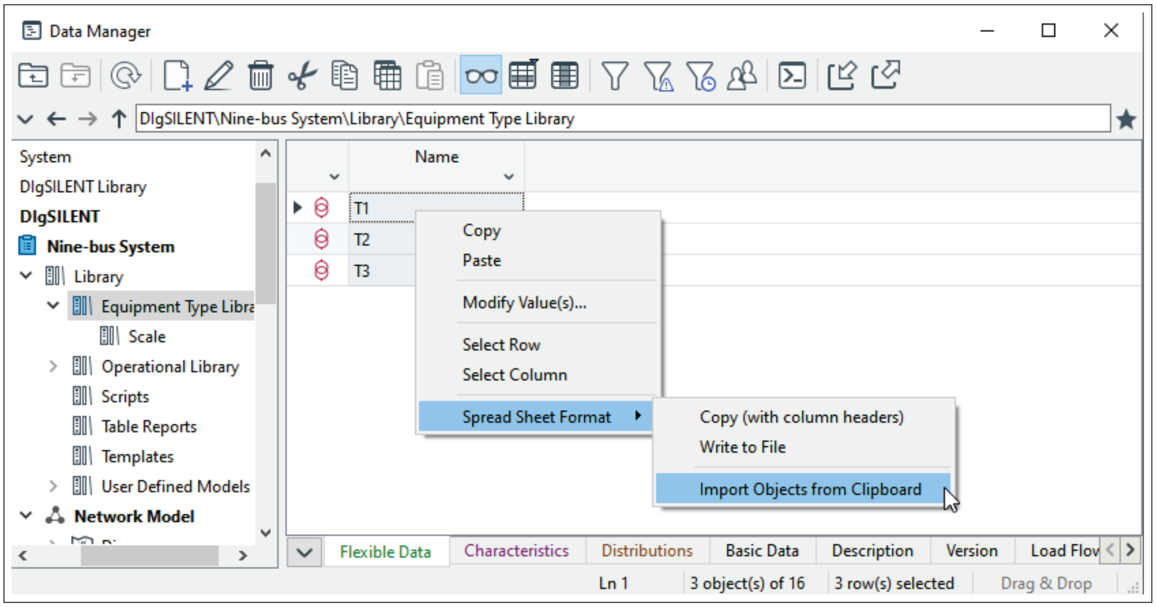


Figure 10.9.7: Importing objects from clipboard

The result of the object import depend on whether or not objects of the imported class and with the imported names already exist or not in the database folder. In the example of Figure 10.9.8, none of the imported objects existed in the database an all were created new therefore. The example shows the database in detail mode.

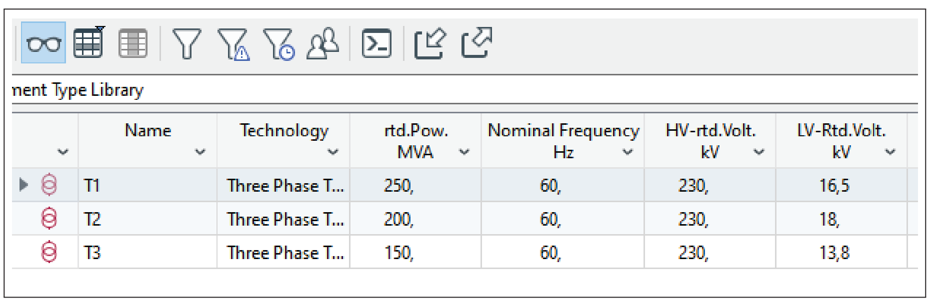


Figure 10.9.8: Result of spreadsheet object import

Note: New objects are created in the *PowerFactory* database folder only when no object of the imported class and with the imported name is found in that folder. If such an object is found then its data will be overwritten by the imported data

Because new objects are only created when they do not exist already, and only the imported parameters are overwritten when the object did exists already, the import is always a save action.

Remarks

Object Names

Object names may not contain any of the characters

*? = " , \ ~ |

Default Data

When an imported object is created newly, the imported data is used to overwrite the corresponding default data. All parameters that are not imported will keep their default value.

Units
The spreadsheet values are imported without units. No conversion from MW to kW, for example, will be possible. All spreadsheet values therefore have to be in the same units as used by *PowerFactory*.

Chapter 11

Building Networks

11.1 Introduction

This Chapter describes basic processes for setting up a network model in *PowerFactory*. Network models are usually constructed via a network graphic, or the Data Manager of the project. Therefore it is useful to have some understanding of these two concepts before starting. See Chapters 9 and 10.

11.2 Defining Network Models using the Graphical Editor

This section explains how the tools of the Graphical Editor are used to define and work with network models. Some basic terms to understand are:

- Node: a node is another name for a terminal, which is an object of class *ElmTerm*. Other objects such as loads and lines are connected to the nodes.
- Edge element: any element connected to a terminal (e.g. load, shunt, line, switch, transformer). It can be a single-port element or have more than one port.
- Branch element: an edge element that is connected between two or more nodes (e.g. switch, line, transformer). It has more than one port.
- Cubicle: the cubicle is not an element represented on the diagram; it is internal to a terminal and can be thought of as the point where an object is connected to the terminal.

11.2.1 Adding New Power System Elements

When new elements are created via a diagram graphic, they will by default be stored in the folder of the grid associated with that graphic ("Target folder for network elements").

PowerFactory provides a Drawing Toolbox from which elements can be selected. This toolbox is only visible to the user when a project and study case is active and the open graphic is made editable by deselecting the *Freeze Mode* button ( ). The Drawing Toolbox will then be seen on the right-hand side of the GUI. The process is that elements are first created and then their parameters subsequently edited through the element and type dialogs. Information about the element and type parameters are given in the Technical References Document.

To create a new power system element, left-click once on the corresponding icon in the toolbox. Then the cursor will have this symbol "attached" to it. Then a left-click on the graphic will create a new element of the selected class. The **Esc** key, or right mouse-click, can be used to stop this process.

Power system elements are placed and connected in the single line graphic by left clicking on empty places on the drawing surface (places a symbol), and by left clicking on nodes (makes a connection). It is therefore recommended to start by creating at least some of the nodes (terminals) in the network first.

The connection between edge elements and terminals is carried out by means of cubicles. When working with the graphical editor, the cubicles are automatically generated in the corresponding terminal.

Note: When connections to terminals are defined with switch elements of the class *ElmCoup* (circuit breakers), cubicles without any additional switches (*StaSwitch*) are generated.

11.2.2 Nodes

When starting to build a network, it is usual to first place the required terminals (*ElmTerm*) on the graphic. There are several representations of terminals available in the Drawing Toolbox. Note that terminals have a parameter `e:iUsage`, which is set to Busbar, Internal Node or Junction Node; by default this will be set to Busbar unless the “point” representation is selected.

Busbar. This is the most common representation of a node.

Busbar (Short). Looks the same as a Busbar but is shorter and the results box and name is placed on the *Invisible Objects* layer by default. Typically used to save space or reduce clutter on the graphic.

Junction / Internal Node. Typically used to represent a junction point, say between an overhead line and cable. The results box and name is placed on the *Invisible Objects* layer by default.

Busbar (rectangular). Typically used for reticulation and / or distribution networks.

Busbar (circular). Typically used for reticulation and / or distribution networks.

Busbar (polygonal). Typically used for reticulation and / or distribution networks.

Busbars (terminals) should be placed in position and then, once the cursor is reset, dragged, rotated and sized as required. Re-positioning is performed by first left clicking on the terminal to mark it, then clicking once more so that the cursor changes to , and then holding the mouse button down and dragging the terminal to a new position. Re-sizing is performed by first left clicking on the terminal to mark it. Sizing handles appear at the ends.

11.2.3 Edge Elements

Edge elements are elements which connect to nodes.

Single port elements (loads, machines, etc.) can be positioned in two ways. The simplest method is to select the symbol from the toolbox and then left click the busbar where the element is to be placed. This will draw the element at a default distance under the busbar. In case of multi busbar systems, only one of the busbars need be left-clicked. The switch-over connections to the other busbars will be made automatically.

The “free-hand” method first places the element symbol wherever desired, that is, first click wherever you wish to place the symbol. The cursor now has a “rubber band” connected to the element (i.e. a dashed line), left-clicking on another node will connect it to that node. To create corners in the joining line left click on the graphic. The line will snap to grid, be drawn orthogonally, as determined by the “Graphic Options” that have been set.

If a single port element is connected to a terminal using the first method (single left click on busbar), but a cubicle already exists at that position on the busbar, the load or machine symbol will be automatically positioned on the other side of the terminal, if possible.

Note: By default all power system elements are positioned "bottom down". If the element has already been placed and one wishes to flip it to the other side of the terminal, it can be done by selecting the element and the *right-click* → *Flip At Node*.

Once drawn, an element can be rotated by right-click and selecting from the *Rotate* commands.

Double port elements (lines, transformers, etc.) are positioned in a similar manner to single port symbols. By left-clicking the first busbar, the first connection is made. The second connection line is now held by the cursor. Again, left-clicking the drawing area will create corners. Double-clicking the drawing area will position the symbol (if not a line or cable - e.g. a transformer). The second connection is made when a node is left clicked.

Triple port elements (e.g. three-winding transformers) are positioned in the same manner as two port symbols. Clicking the first, and directly thereafter the second node, will place the symbol centred between the two nodes, which may be inconvenient. Better positioning will result from left clicking the first busbar, double-clicking the drawing space to position the element, and then making the second and third connection.

The 'free-hand' method for two and triple port elements works the same as for one port elements.

Note: Pressing the **Tab** key after connecting one side will leave the second leg unconnected, or jump to the third leg in the case of three port elements (press Tab again to leave the third leg unconnected). Pressing **Esc** or right-click will stop the drawing and remove all connections. If the element being drawn seems as if it will be positioned incorrectly or untidily there is no need to escape the drawing process; make the required connections and then right-click the element and Redraw the element whilst retaining the data connectivity.

It is recommended that the connections for a transformer are always made in order of voltage, starting with the highest voltage connection.

It is possible to insert a terminal into an existing line in the single line diagram by placing the terminal on the line itself. This splits the line into two, defaulting at 50 %. If the terminal is then moved, the adjacent line sections will automatically be redrawn. If the terminal needs to be moved (graphically) along the line, this can be done by holding the **Ctrl+Alt** keys whilst moving the terminal. Note that both these adjustments are just graphical and do not change the actual lengths of the two lines.

Annotations are created by clicking one of the annotation drawing tools. Tools are available for drawing lines, squares, circles, pies, polygons, etc. To draw these symbols left click at on an empty space on the single line diagram and release the mouse at another location (e.g. circles, lines, rectangles). Other symbols require that you first set the vertices by clicking at different positions and finishing the input mode by double-clicking at the last position.

For further information on defining lines, see Section 11.3 (Lines and Cables).

11.2.4 Cubicles

A cubicle (*StaCubic*) in *PowerFactory* is an object which stores information about the connection between an edge element and a node element. Whenever an edge element is connected to a node element it must be connected via a cubicle. However, the cubicle is created automatically when an edge element and a node are connected and the user does not generally need to take special measures to facilitate the creation of the cubicle.

In the data manager cubicles are stored within node elements. A node element can contain cubicles which do not have an edge element associated with them. This can happen if for example an edge element is connected to a node and then disconnected. A cubicle is automatically created during the connection but is not automatically deleted upon disconnection and therefore remains in the node. If alternatively the edge element was deleted instead of disconnected, in this case, the cubicle would also be deleted. If an attempt is made to connect an edge element to a node containing such unassigned cubicles then *PowerFactory* will give the user a choice of unassigned cubicles to which they can connect.

In addition to storing information about the associated connection, a cubicle is also used as a storage location for certain objects. For example, relays, switches, circuit breakers and measurement devices can all be stored inside cubicles. Only one switching device (*StaSwitch*) can be stored inside a cubicle and this device can be used to toggle the connection between the edge element and the node element. By default a switching device is always created in a cubicle, but the user can also choose to remove the switching device if required.

11.2.5 Marking and Editing Power System Elements

A left-click on an element selects it and it then becomes the “focus” of the next action or command. For branch elements, the parts near their connection to nodes are treated differently and show specific context sensitive menu options regarding the marked side of the element (e.g. to insert a new device at the line end or to disconnect the line). To get all the menu options anyway, hold down the **Ctrl**-key while clicking the right mouse button.

The element can be un-marked or de-selected by clicking on another element, by clicking onto some free space in the graphic or just by pressing the **Esc** key.

There are different ways to select several objects at once:

- Pressing the *Mark All Elements* button ( ), or using **Ctrl+A** to mark all graphical elements.
- If the *Rectangular Selection* button ( ) is pressed (default condition), a set of elements can be selected by clicking on a free spot in the drawing area, holding down the left mouse button, moving the cursor to another place, and release it. All elements in the so defined rectangle will now be marked.
- One or more objects can be marked holding down the **Ctrl** key whilst marking the objects.
- Holding down the **Alt**-key while clicking on the same object again marks all the adjacent objects. Doing this several times marks more and more connected objects.
- If the area to be selected cannot be covered with a rectangular form, the *Free-form Selection* button ( ) can be used to select a custom area of the diagram.

The data of any element (its edit dialog) may be viewed and edited by either double-clicking the graphic symbol, or by right-clicking it and selecting *Edit Data*. If multiple objects are selected, *right-click* → *Edit Data* will bring up a data browser.

The option *Edit and Browse Data* will show the element in a Data Manager environment. The object itself will be selected (highlighted) in the Data Manager and can be double-clicked to open the edit dialog. A new Data Manager will be opened if no Data Manager is presently active. The edit dialogs for each element may be opened from this data browser one by one, or the selected objects can be edited in the data browser directly.

Note: The position of an object in the database tree can be found by:

- Opening the edit dialog. The full path is shown in the header of the dialog.
- Using the keyboard shortcut **Ctrl+E**, which opens the Data Manager with the element marked in the folder hierarchy.
- Right-clicking the object and selecting Edit and Browse. This will open a new database browser when required, and will focus on the selected object.

11.2.6 Interconnecting Power Subsystems

Interconnections between two different graphics can be done in one of two ways:

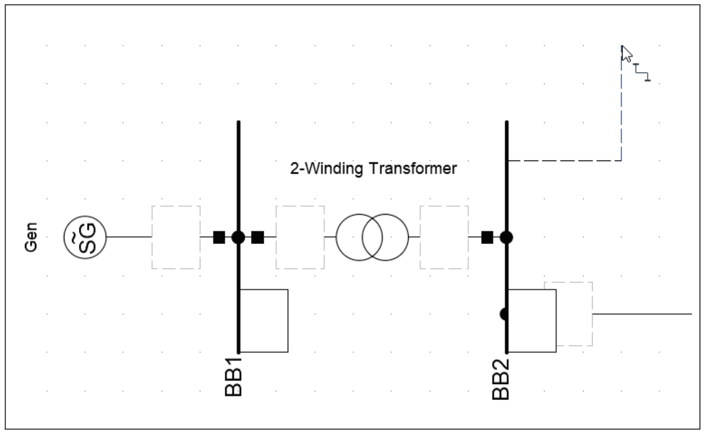
1. Representing a node in another diagram by copying (*right-click* → *Copy*) the node in the first graphic and pasting just the graphic object (*right-click* → *Paste Graphic Only*) into the second diagram. Both graphical objects are then associated with the same element; no new element is created.
2. Ensure that there is a node to connect to in the graphics that are to be interconnected. Then connect an edge element between the two graphics.

Example

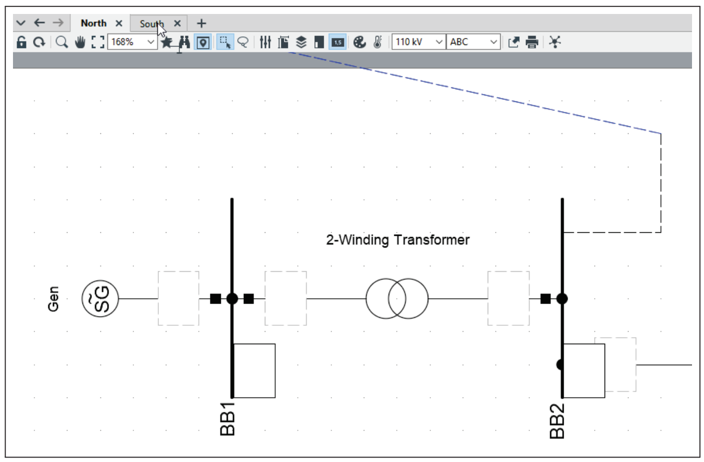
In this example a line will be used to interconnect two regions using the second method. See figure 11.2.1.

1. Select a line drawing tool from the toolbox and create the first connection as normal by left clicking a node (see figure 11.2.1a).
2. Double-click to place the symbol. Your cursor is now attached to the line by a "rubber band". Move the cursor to the bottom of the drawing page and click on the tab of the graphic that the interconnection is to be made to (see figure 11.2.1b).
3. Once in the second graphic left click to place the line symbol (see figure 11.2.1c) and then left click on the second node.

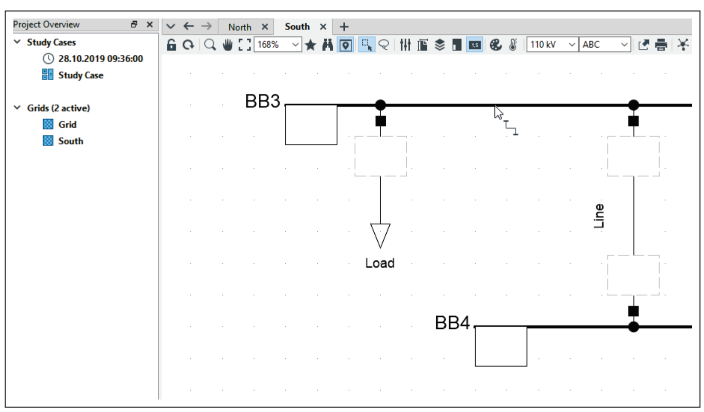
The interconnected leg is shown by a $\gg$ symbol.



(a) First step



(b) Second step



(c) Third step

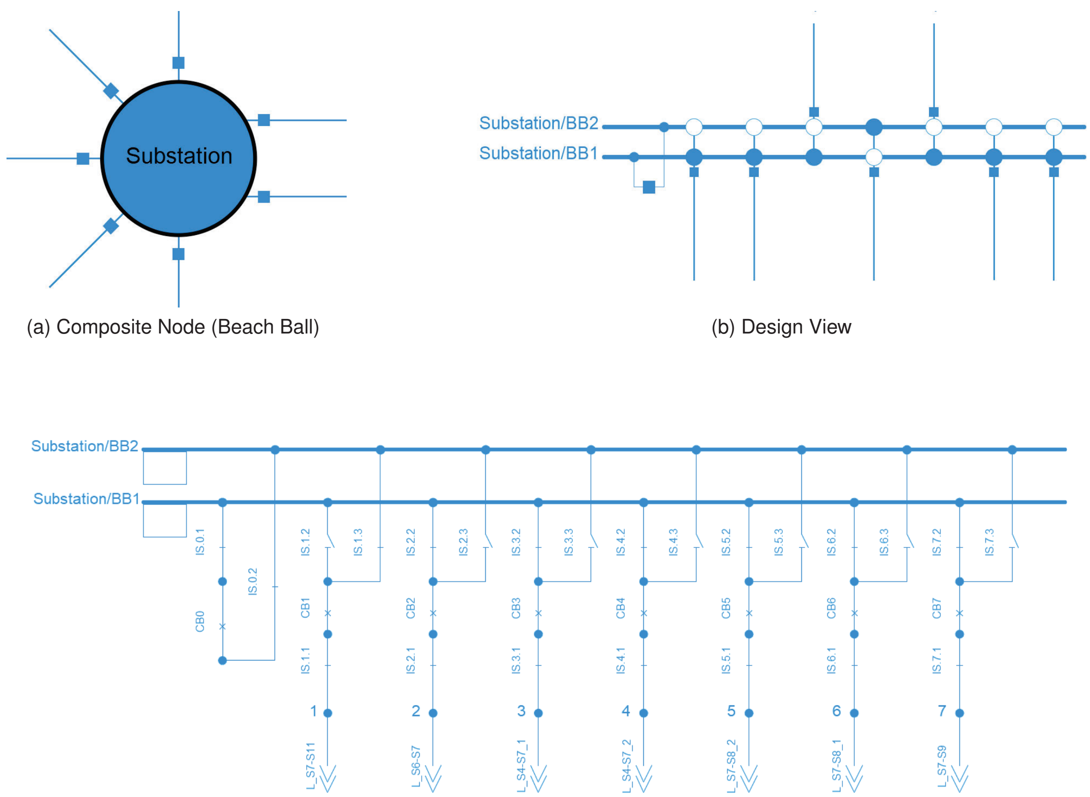
Figure 11.2.1: Interconnecting Power Subsystems

Note: The first method of interconnection, that of representing a node in two, or more, different graphics, may lead to confusion at a later point as the 'inflow' and 'outflow' to the node will not appear correct when just one graphic is viewed - especially if a user is not familiar with the system. The node may be right-clicked to show all connections in what is known as the "detailed diagram" (menu option *Show Detailed Graphic of Substation*). Thus, the second method may be preferred. To check for nodes that have connections on other graphics the *Topology* → *Missing graphical connections* diagram colouring may be employed.

11.2.7 Substations

Substations (*ElmSubstat*) and Secondary Substations (*ElmTrfstat*) can be created from predefined templates or from templates previously defined by the user by means of the icons located in the Drawing Tools toolbox, as explained below.

Depending on the user's preferences and taking into account the degree of detail required in the graphical representation of the network, the substations can be displayed as Composite Nodes, which is suitable for overview diagrams, or have a so-called Design View, which provides a simplified representation with additional information on connectivity. A detailed representation of the substation topology is also offered. The different graphical representations are depicted in Figure 11.2.2.



(a) Composite Node (Beach Ball)

(b) Design View

(c) Detailed Graphic

Figure 11.2.2: Substation graphical representations

Note: Before starting to create new substations, the user may wish to consider whether it would be useful to have Bays (*ElmBay*) created within the substation. If so, these can be automatically created if the necessary project setting is selected. See section 8.1.2.4. More information about Bays is given in section 11.2.7.3.

11.2.7.1 Creating a Substation from predefined templates

Substation representation: Composite Node

To create a substation from a predefined template with a graphical representation as a "Composite Node" (also "Beach Ball" or "Single Node"), the following steps should be taken:

- From the Drawing Tools toolbox, select one of the following *Busbar Systems*:
  - Click on ◯ for a substation represented by a circle
  - Click on ☐ for a substation represented by a rectangle
  - Click on ◯ for secondary substation
- Select the required substation template from the browser that appears.
- Click on the drawing area to place the symbol. The substation is automatically created in the Grid set as target folder for network elements.
- Close the window with the templates. Alternatively press **Esc** or right click on the mouse to get the cursor back (browser will close automatically).
- To name the substation, right click its symbol, select *Edit Substation...* from the context-sensitive menu and rename it appropriately.
- Resize the substation symbol as required, by clicking on it and dragging one of the corners or sides.

Note: The nominal voltage of the substation can be set before drawing it, via the Node default option on the graphics toolbar (see 9.3)

To highlight the connectivity within a substation represented by means of a composite node, the user can use *Diagram Colouring*. For this, press the button to open the diagram colouring scheme dialog. Choose the "function" for which the colouring mode is relevant (for example, the *Basic Data* page) and then select *Other* → *Topology*→ *Station Connectivity*.

Substation representation: Design View

To create a substation from a predefined template with a "Design View" representation, the following steps should be taken:

- From the Drawing Tools toolbox, select one of the following *Busbar Systems*:
  - Click on for a Single Busbar
  - Click on for a Single Busbar with Tie Breaker
  - Click on for a Double Busbar
  - Click on for a Double Busbar with Tie Breaker
  - Click on for a 1 1/2-Busbar
  - Click on for a Single Busbar with Tie Breaker and Bypass
  - Click on for a Double Busbar with Bypass
  - Click on for a Double Busbar with Tie Breaker and Bypass
- Click on the drawing area to place the symbol. The substation is automatically created in the Grid set as target folder for network elements.
- Press **Esc** or right click on the mouse to get the cursor back.

- To name the substation, right click its symbol, select *Edit Substation...* from the context-sensitive menu and rename it appropriately.

Note: The nominal voltage of the substation can be set before drawing it via the Node default option on the graphics toolbar (see 9.3). Alternatively, the user can change the nominal voltage of one of the terminals forming the busbar system and update the nominal voltage of all the terminals inside the station.

Switching substation representation

It is possible to change the graphical representation of the substation, i.e. to switch from a composite node representation to a design view and vice versa. To do this, select the composite node representation of the substation with a right click and choose the option *Convert to design view*, or select one or more of the busbars in design view and *Convert to beach ball*.

Substation representation: Detailed Graphic

Predefined substation templates have a detailed graphical representation. To access it:

- If the substation is represented via a composite node, there are two ways to open its detailed graphic page. The first is to double-click on the corresponding composite node in the overview diagram. The second is right click its beach ball symbol and select *Show Graphic...* from the context-sensitive menu.
- If the substation has a design view representation, its detailed diagram can be opened by selecting the option *Show Detailed Graphic of Substation...* from the context-sensitive menu.

On the detailed graphic of the substation, it is possible to display the names of neighboured substations/sites next to the lines leaving the substation, as illustrated in figure 11.2.3. In order to achieve that, edit the *Labels* layer and on the *Text Boxes* page, the option "Show jump-to labels at graphically half-connected lines" should be checked.

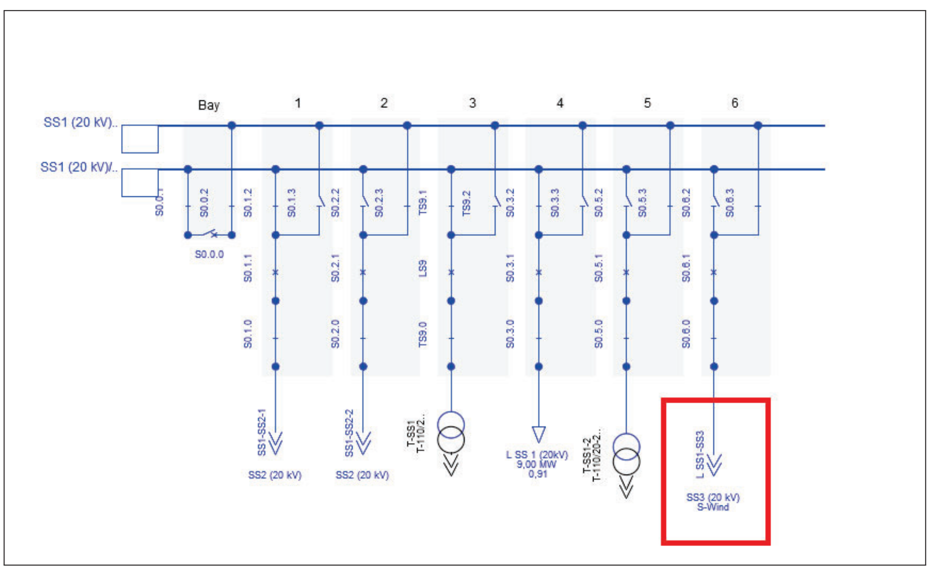


Figure 11.2.3: Displaying the names of neighboured substations in detailed graphic of substation

---

Note: Detailed graphics of substations can be generally accessed from the Data Manager. For this, the user has to go to the *Diagrams* folder (in *Network Model*), find the graphic object of the substation and open it by selecting *Show Graphic* from the context-sensitive menu.

11.2.7.2 Creating a Substation from templates previously defined by the user

The following steps describe the general procedure for inserting a previously defined user template:

- From the Drawing Tools tool-window select the *General Busbar System* symbol ( )
- Select the required substation template from the browser that appears. If the *Templates* folder in the project library contains just one template, no window is opened and the user can place the substation in the drawing area as described in the following step.
- Click on the drawing area to place the symbol. The substation is automatically created in the Grid set as target folder for network elements.
- Close the window with the templates. Alternatively press **Esc** or right click on the mouse to get the cursor back (browser will close automatically).
- To name the substation, right click its symbol, select *Edit Substation...* from the context-sensitive menu and rename it appropriately.
- Resize the substation symbol as required, by clicking on it and dragging one of the corners or sides.

Note: To make use of user-defined substation templates, such templates have to be created first. Information about creating user-defined templates can be found in sections 14.4.2 and 14.4.3.

11.2.7.3 Substation Bays

The Bay object *ElmBay* is used to group together the network elements that normally constitute a standard bay connection of a circuit to a busbar within a substation. This grouping is useful for visualisation but is also used by the Load Flow Calculation option *Calculate max. current at busbars*: see section 25.3.3.

If the detailed diagram of the substation is viewed, the bays are highlighted by rectangular blocks (by default pale grey). These blocks are not just annotation - users can move additional elements into the area and they will also be moved into the Bay object in the project hierarchy. The Bay representation can be manually resized but will also expand automatically to encompass all objects that form part of the bay.

Additional bays may be added using the Bay icon in the toolbox.

Note that the bay representation on the graphic is held within a layer called “Bays and Sites”, so may be made visible or invisible as required. The colour can also be configured.

11.2.7.4 Substation Switching Rules

Switching Rules (*IntSwitching*) store switching actions for a selected group of switches that are defined inside a substation. The different switching actions (no change, open or close) are defined by the user considering different fault locations that can occur inside a substation. By default, the number of fault locations depends on the number of busbars and bay-ends contained inside the substation; although the user is allowed to add (and remove) specific fault locations and switches belonging to the substation. The switch actions will always be relative to the current switch positions of the breakers.

The selection of a *Switching Rule* for a substation is independent of the selection of a *Running Arrangement* and if required, the reference to the switching rule in a substation can be stated to be operational data; provided the user uses the *Scenario Configuration* object. For more information on the scenario configuration refer to Chapter 16 (Operation Scenarios).

The typical application of Switching Rules is in contingency analysis studies or reliability analysis studies, where the predefined switching rules could be immediately applied after a fault. For example, a busbar fault in a double-busbar system could be followed by switching the connections to the other healthy bus bar. The Switching Rules are composed of a matrix, which defines the required switch actions for several fault locations in the substation. Please refer to 27.4.6.1 for the application in contingency analysis.

Switching Rules are also considered during Reliability Analysis (see Chapter 45)

To create a switching rule

- *Edit a Substation*, either by right-clicking on the substation busbar from the single line graphic, and from the context-sensitive menu choosing *Edit a Substation*, or by clicking on an empty place in the *substation graphic*, and from the context-sensitive menu choosing *Edit Substation*. This will open the substation dialog.
- Press the *Select* button ( ) in the *Switching Rule* section and select *New. . .*
- The new *Switching Rule* dialog pops up, where a name and the switching actions can be specified. The switching actions are arranged in a matrix where the rows represent the switches and the columns the fault locations. By default the fault locations (columns) correspond to the number of busbars and bay-ends contained inside the substation, while the switches correspond only to the circuit breakers. The user can nevertheless add/remove fault locations and/or switches from the *Configuration* page. The switch action of every defined breaker in the matrix can be changed by double clicking on the corresponding cell, as illustrated in figure 11.2.4. Press afterwards **OK**.
- The new switching rule is automatically stored inside the substation element.

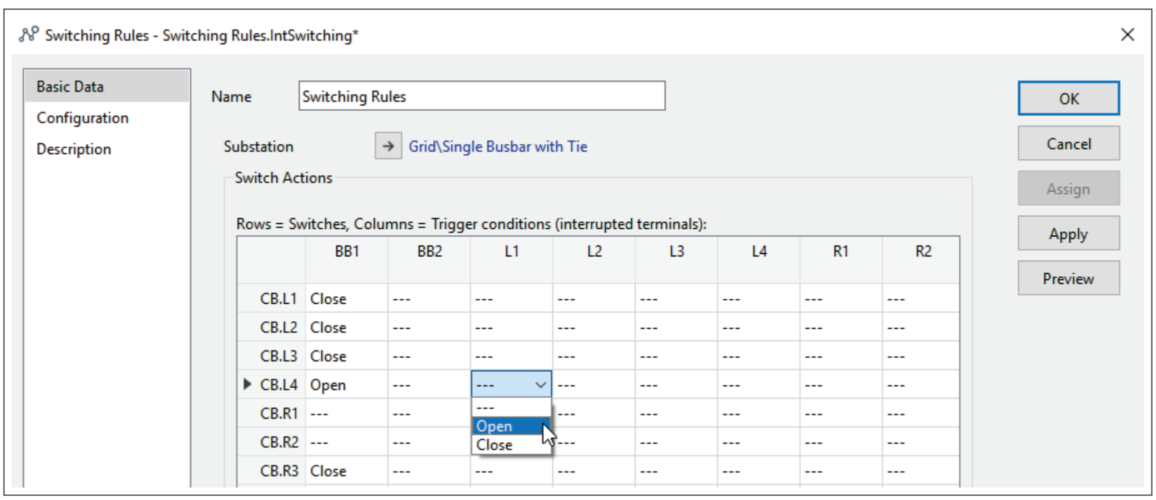


Figure 11.2.4: Switching Rule Dialog

To select a Switching Rule

A *Switching Rule* can be selected in the *Basic Data* page of a substation dialog (*ElmSubstat*) by:

- Opening the substation dialog.
- Pressing the *Select* button ( ) in the *Switching Rule* section. A list of all Switching Rules for the current substation is displayed.
- Selecting the desired Switching Action.

To apply a Switching Rule

A *Switching Rule* can be applied to the corresponding substation by pressing the **Apply** button from within the switching rule dialog. This will prompt the user to select the corresponding fault locations (busbars) in order to copy the statuses stored in the switching rule directly in the substation switches. Here, the user has the option to select either a single fault location, a group or all of them.

The following functional aspects must be regarded when working with switching rules:

- A switching rule can be selected for each substation. By default the selection of a switching rule in a substation is not recorded in the operation scenario. However, this information can defined as part of an operational scenario by using the *Scenario Configuration* object (see Chapter 16: Operation Scenarios).
- If a variation is active the selection of the Switching Rule is stored in the recording expansion stage; that is considering that the *Scenario Configuration* object hasn't been properly set.

To assign a Switching Rule

The **Assign** button contained in the switching rule dialog allows to set it as the one currently selected for the corresponding substation. This action is also available in the context-sensitive menu in the Data Manager (when right-clicking on a switching rule inside the Data Manager).

To preview a Switching Rule

The **Preview** button contained in the switching rule dialog allows to display in a separate window the different switch actions for the different fault locations of the corresponding substation.

11.2.8 Sites

As noted in section 4.6.8, a site is normally used to group network components, for example, substations of different voltage levels at the same location. Due to this particular characteristic, site elements do not have predefined templates inside the software.

The site element can be represented in overview and/or geographic diagrams; a detailed representation can also be defined.

11.2.8.1 Creating a New Site in Overview and Geographic Diagrams

Site elements can be represented by a square or a circle using the buttons and from the Drawing Toolbar. For geographic diagrams, only the circular representation is available.

To draw a new site:

- Click on one of the site symbols ( )
- Click on the overview diagram to place the symbol. The site is automatically created in the active grid folder.
- Press **Esc** or right click on the mouse to get the cursor back.
- Resize the site symbol as required.
- Right click on the site and select *Edit Site* to open the edit dialog of the element.

Once the site is defined, a detailed diagram is automatically created. It is possible then to draw all the elements directly inside the Site diagram, using detailed substation diagram templates as explained in section 11.2.7.1.

If the site already exists it is possible to use the *Diagram Layout Tool* to generate its detailed representation automatically. See section 11.6 for more information about the Diagram Layout Tool.

The resizing and colouring according to connectivity of the site can be done as explained in section 11.2.7.1.

11.2.8.2 Site Frames

In some overview diagrams, it may not be sufficient to use site objects for all sites; and users may want to see more detail at certain sites. This can be done by using a representation of the site as a frame within which the substations can also be seen.

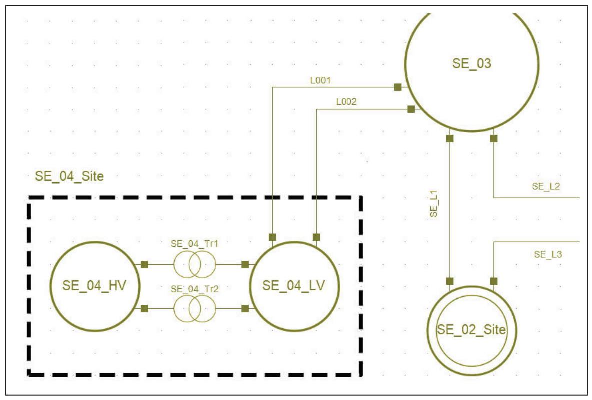


Figure 11.2.5: Substations within a site frame in an overview graphic

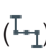
The site can be introduced into the graphic using the graphic object , which draws the site as a rectangular frame. The user can then create new substations within this frame and they will become part of the *ElmSite*. The Site frame will automatically resize to accommodate the new substation representations.

This new graphical option gives the user the flexibility to show sites as single symbols or in more detail, as required, and the site frame representation is held within a graphical layer called "Bays and Sites", so may be made visible or invisible as required.

11.2.9 Composite Branches

New composite branches (*ElmBranch*) can be created in the Data Manager using the procedure described in Chapter 10, Section 11.5.4 (Defining Composite Branches in the Data Manager). The definition and connection of the branch components can then be done in the single line diagram that is automatically generated upon creation of a new branch.

Branches are created in single line diagrams using previously defined templates. To create a new branch from a template:

- Click on the *Composite Branch* symbol () in the Drawing Toolbox. If there is more than one branch template (in the Templates library), a list will appear, so that the correct one can be selected.
- If the branch is to be connected to two terminals of the same single line graphic, simply click once on each terminal.
- If the branch is to be connected to a terminal from another single line diagram, you have to 'Paste graphically' one of the terminals on the diagram where you want to represent the branch, or connect across pages as discussed in Section 11.2.6 (Interconnecting Power Subsystems).
- If the branch is to be connected to terminals from a substation, click once on each composite node to which the branch is to be connected. You will be automatically taken inside each of those

composite nodes to make the connections. In the substation graphic click once on an empty spot near the terminal where you want to connect the branch end, and then on the terminal itself.

A diagram of the newly created branch can be opened by double clicking on its symbol. In the new diagram it is possible to rearrange the branch configuration and to change the branch connections.

Details of how to define templates can be found in Chapter 14 (Project Library).

11.2.10 Single and Two Phase Elements

It is possible to define the phase technology of elements such as terminals, lines, and loads. In instances where the number of phases of a connecting element (e.g. a circuit breaker or line) is equal to the number of phases of the terminal to which it connects, *PowerFactory* will automatically assign the connections. However, when connecting single-phase elements to a terminal with greater than one phase, or two-phase elements to terminals with greater than three phases, it is sometimes necessary to adjust the phase connectivity of the element to achieve the desired connections. The phase connectivity can be modified as follows:

- Open the dialog window of the element (by double-clicking on the element).
- Press the **Figure** >> button to display a figure of the elements with its connections on the bottom of the dialog window.
- Double-click on the dark-red names for the connections inside this figure.
- Specify the desired phase connection/s.

Alternatively, click the right arrow (→) next to the terminal entry and specify the desired phase connection/s.

Note: It is possible to colour the grid according to the phases (System Type AC/DC and Phases). For more information about the colouring refer to Section 9.3.7.1 (Diagram Colouring).

11.3 Lines and Cables

This section describes specific features and aspects of line and cable data models used in *PowerFactory*. Detailed technical descriptions of the models are provided in Technical References Document.

In *PowerFactory*, lines and cables are treated alike, they are both instances of the generalised line element *ElmLne*. A line may be modelled simply as a point-to-point connection between two nodes and will refer to a line (*TypLne*), tower (*TypTow*), a tower geometry (*TypGeo*), a line coupling (*ElmTow*), or a cable system coupling (*TypCabsys*, *TypCabmult*) type. Alternatively, lines may be subdivided into sections referring to different types.

Note: Anywhere that 'line' is written in this section, 'lines and/or cables' may be read, unless otherwise specified.

The three basic line configurations are shown in figure 11.3.1:

1. Top line: The simplest line is a single line object (*ElmLne*) connected between two terminal objects via two cubicle objects.
2. Middle Line: Line (*ElmLne*) objects can also be cascaded or subdivided. Again, with each *ElmLne* connected between two terminal objects via two cubicle objects.

3. Bottom line: Line (*ElmLne*) objects can also be subdivided into line section objects (*ElmLnesec*). The *ElmLne* object is again connected via cubicle objects between two terminal objects, but here the line section (*ElmLnesec*) objects constituting the line are not specifically connected between terminals themselves meaning the number of cubicles and terminals associated with such a configuration could be substantially reduced.

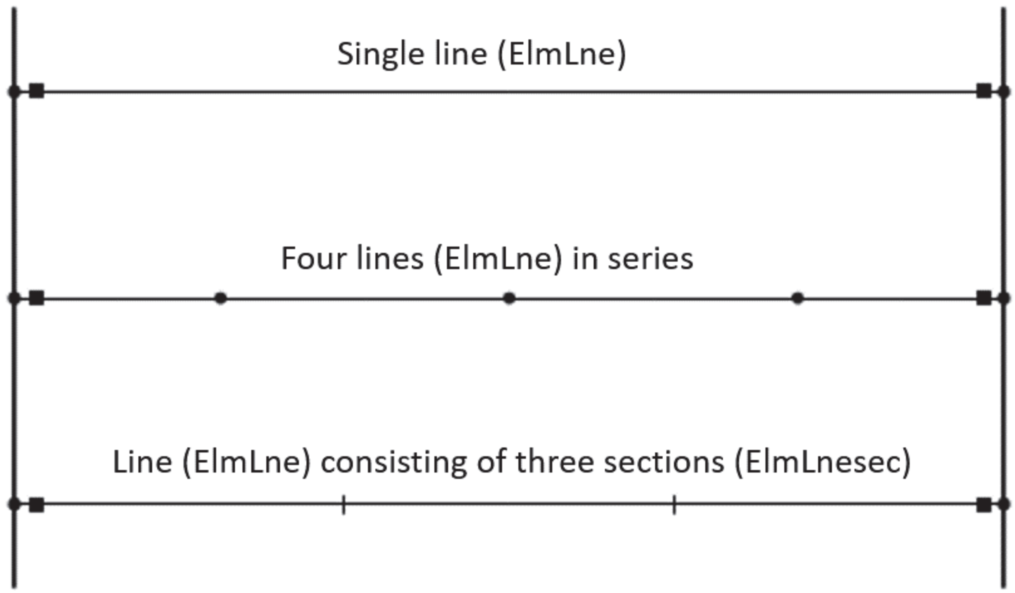


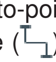
Figure 11.3.1: Basic line configurations

Cascading line objects together can be useful where it is necessary to explicitly represent the transposition of conductors. It is also possible to give each separate cascaded line a different conductor/cable type or tower geometry. Further, it is possible to branch off towards loads, generators and other substations (for example) at the intermediate terminals and include additional switching devices in the line. It may also be a useful representation to approximate the behaviour of a distributed parameter model using lumped parameter models.

An arrangement of ElmLnesec objects is generally used to represent circuits or parts of circuits consisting of multiple conductor/cable types. Such arrangements being typical of low voltage and medium voltage distribution networks.

In addition to the configurations described above, objects known as branch (*ElmBranch*) objects can be defined to organise and simplify the handling of complex composite arrangements of lines. Handling of these objects is described in sections 11.2.9 and 11.5.4.

11.3.1 Defining a Line (*ElmLne*)

The simplest line model is a point-to-point connection between two nodes. This is normally done in the single line graphic by selecting the () icon and by left clicking the first terminal, possibly clicking on the drawing surface to draw a corner in the line and ending the line at the second terminal by left clicking it. This will create an *ElmLne* object in the database. When this object is edited, the following dialog will appear.

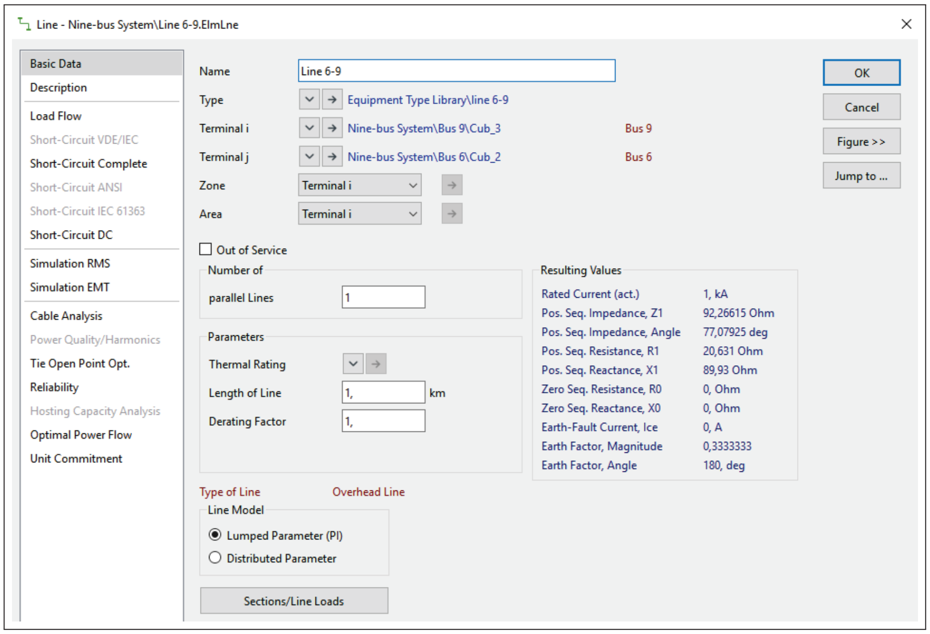


Figure 11.3.2: Editing a transmission line

The dialog shows the two cubicles to which the transmission line is connected (*terminal i* and *terminal j*). The line edit dialog shows the name of the node (in brown) in addition to the name of the cubicle (in blue). The actual connection point to the node is the cubicle and this may be edited by pressing the edit button (→). The cubicle may be edited to change the name of the cubicle, add/remove the breaker, or change phase connectivity as discussed in Section 11.2.10 (Single and Two Phase Elements).

The type of the line is selected by pressing the (⌄) next to the type field. Line types for a line are:

- The *TypLne* object type, where electrical parameters are directly written (the user can select if the type is defined for an overhead line or a cable).
- Tower types (*TypTow* and *TypGeo*), where geometrical coordinates and conductor parameters are specified, and the electrical parameters are calculated from this data. Selection of the tower type will depend on the user's requirement to link conductor type data to the line element as in *TypGeo* (for re-use of the one tower geometry with different conductors), or to link conductor type data to the tower type as in *TypTow* (for re-use of one tower geometry with the same conductors).
- Cable definition types (*TypCabsys*), used to complete the definition of a cable system. It defines the coupling between phases, i.e. the coupling between the single core cables in a multiphase/multi-circuit cable system.

Once the lines (or cables) have been created it is possible to define couplings between the circuits that they are representing by means of line coupling elements *ElmTow* (for overhead lines) and cable system coupling elements *ElmCabsys* (for cables).

Details of how to create Line Sections, Cable Systems, and Line Couplings are provided in the following sections, and further information about line/cable modelling is given in the Technical References Document.

11.3.2 Defining Line Sections

To divide a line into sections:

- Press the **Sections/Line Loads/Compensation** button in the line dialog. This will open a data browser showing the existing line sections (if any).
- Click on the new object icon () and select the element *Line Sub-Section* (*ElmLnesec*).
- The edit dialog of the new line section will appear, and the type and length of the new section can be entered.

11.3.3 Defining Line Compensation

It is possible to define the Line Compensations at one or both ends of a line element. In order to achieve that, following steps should be followed:

- Press the **Sections/Line Loads/Compensation** button in the line dialog. If there are existing line compensations then they will be listed in the opened data browser.
- Next, click on the new object icon () and select the element *Line Compensation* (*ElmLnecomp*).
- The edit dialog of the new line compensation will appear. Based on the selected *Input Mode*, the parameters can be updated.

11.3.4 Defining Line Couplings

The Line Couplings element (*ElmTow*) is used to represent electromagnetic coupling between transmission lines. In order to define a line coupling, a tower type (*TypTow* or *TypGeo*) determining the geometrical characteristics is required, along with the conductor type (*TypCon*) of the circuits.

Since line coupling occurs between lines on the same tower or between lines running approximately parallel to each other, the lines should be the same length; if they are not, a warning message will be displayed when calculations are executed and the shorter length will be considered for the coupling. To facilitate this Line objects can divided into two objects with one of the divided parts assigned a length as well as a coupling to other line objects of the same length. The other divided part can be assigned the remainder of the length of the circuit and no coupling.

The line coupling can be directly defined in the Data Manager; however it is easier to do it from the single line diagram as follows:

1. Select the lines drawn in the single line diagram, right-clicking and select *Define* → *Line Couplings* from the context sensitive menu.
2. A dialog pointing to the *Equipment Type Library* will open. At this point you have to select the tower type, either a *TypTow* or a *TypGeo*. If none of them are yet available, the button *New Object* () can be used to define a new tower type. In this example a *TypTow* will be used.
3. On the edit dialog of the tower type, shown in figure 11.3.3, the number of circuits and earth wires should be defined. Then the conductor types should be selected, by double clicking on the *TypCon* field.

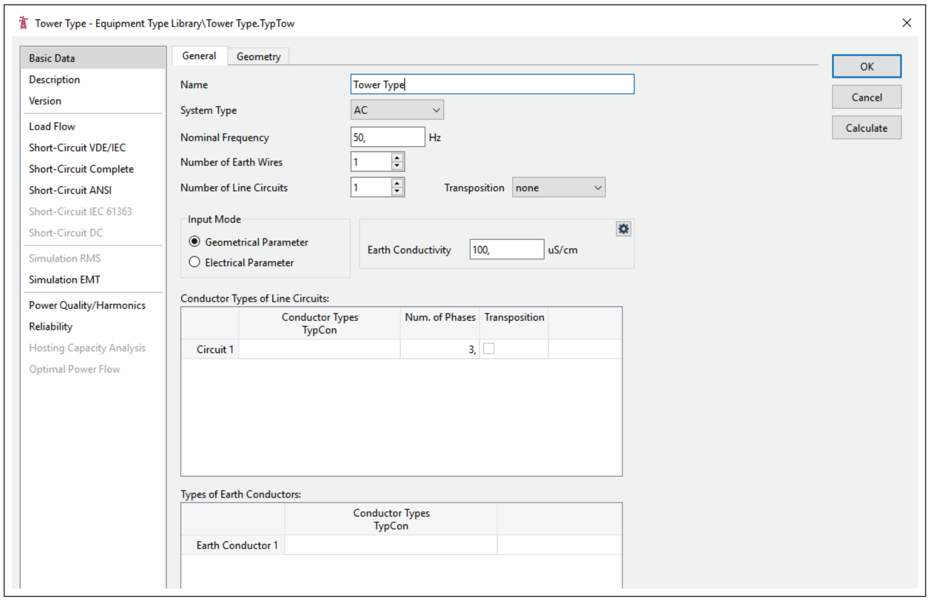


Figure 11.3.3: Tower Type (*TypTow*)

4. Once again, a dialog pointing to the *Equipment Type Library* will open to select the conductor type. If no type is yet available, the button *New Object* ( ) can be used to define a new conductor type (*TypCon*).

5. On the edit dialog of the conductor type, shown in figure 11.3.4, the rated voltage and current, number of subconductors, model and measurements should be defined.

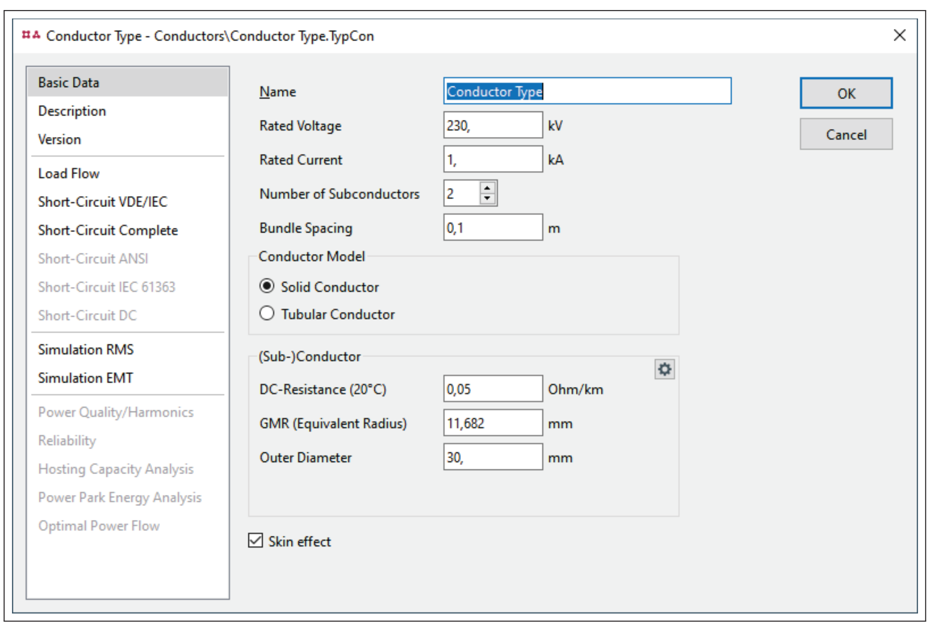


Figure 11.3.4: Conductor Type (*TypCon*)

6. Separate conductor types can be used for each circuit and earth wires. Note that earth wires are only to be entered if the *Input Mode* is set to *Geometrical Parameters*.

7. Once the conductors are defined, the rest of the parameters of the tower type should be entered. The transposition can be selected as:

- None
- Circuit-wise
- Symmetrical
- Perfect

More information about these options can be found in the Technical References Document (Overhead Line Constants).

8. On the *Geometry* page of the tower type edit dialog, the disposition of the conductor can be defined by inserting the coordinates in metres, on the right side of the dialog, an image of the location of the phases is shown. The button **Calculate** can be used to get the matrix of impedances; more information about the calculation and values obtained is also available in the technical reference for the Overhead Line Constants.

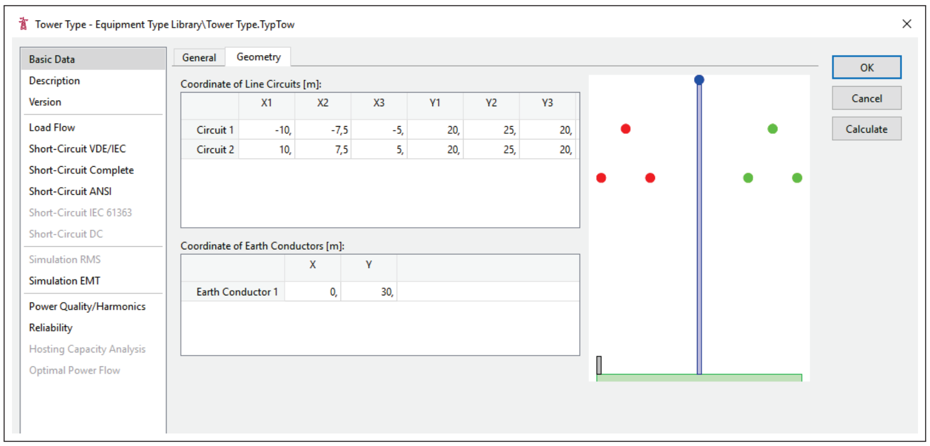
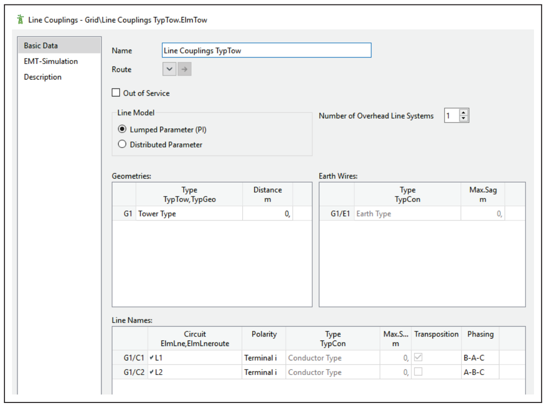


Figure 11.3.5: Geometry of the Tower Type (*TypTow*)

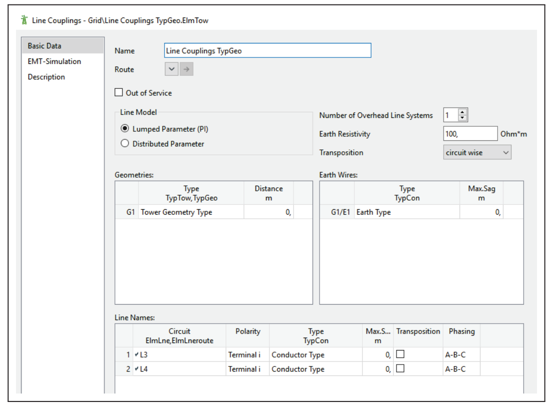
9. Once the tower type is defined, click on the **OK** button. A new dialog will open, where the lines (*ElmLne*) are assigned to the circuits. Select the line for each circuit and click **OK**. Now the line coupling element (*ElmTow*) is complete. Note that once a line coupling has been assigned to a line element, the type of the line changes to line coupling.

The example above uses a *TypTow* tower type, but line couplings can also be defined using the tower geometry type *TypGeo*. The main difference is that within the tower type (*TypTow*) the geometry of the tower is associated with the corresponding conductor types of each circuit and therefore the tower type contains all data of the overhead line transmission system as required for the calculation of the electrical parameters. The tower geometry type (*TypGeo*), however, does not contain a reference to the conductor type, so that the definition is not complete and the conductor types are added later on in the line (*ElmLne*) or coupling (*ElmTow*) elements. This makes the tower geometry type (*TypGeo*) more flexible, and it is therefore the preferred option when combining the same tower geometry with different conductor types.

The following figure presents a comparison of line couplings using different tower types.



(a) Using Tower Type



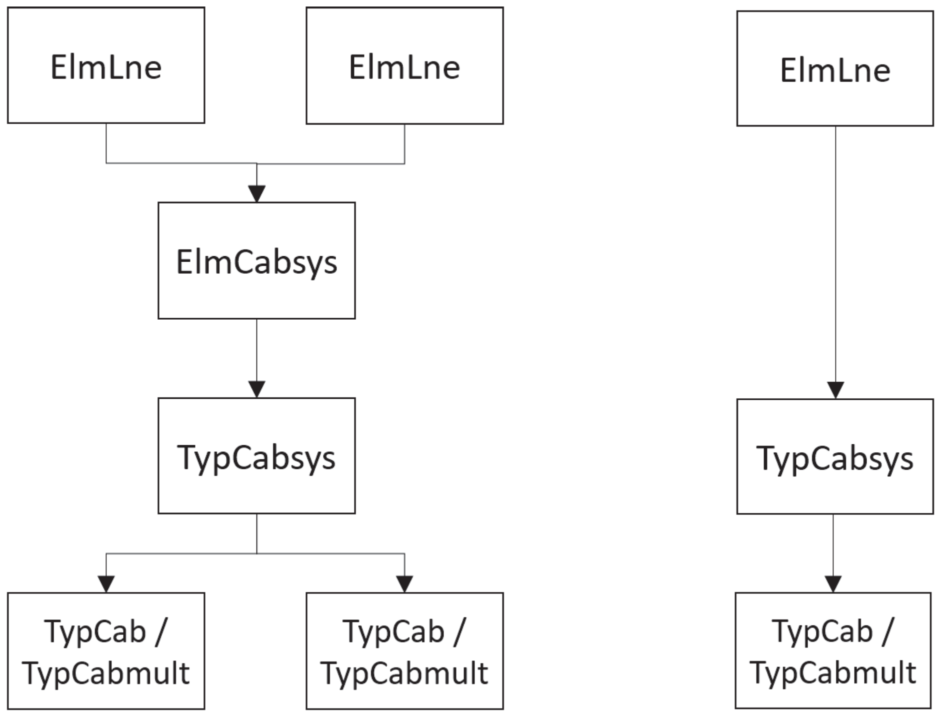
(b) Using Tower Geometry Type

Figure 11.3.6: Line Couplings Element

11.3.5 Defining Cable Systems

A cable system can be used to calculate and represent the impedance of a cable or group of cables including the calculation of mutual impedances developed between all conductors comprising the system. Unlike the simple *TypLne* representation of a cable the input data does not include sequence impedance data but rather includes geometrical data e.g. the relative positions of individual cables and individual cores as well as data about the construction of the cables for example the core material, the core cross section, the insulation type, or the presence of a sheath and armour. It is from this data that *PowerFactory* calculates the impedances and admittance matrices of the arrangement, which are subsequently used to represent the system in the various calculations.

Figure 11.3.7 illustrates that there are essentially two different kinds of cable system that can be constructed in *PowerFactory*.



(a) Cable system with coupling between *ElmLne* objects (b) Cable system based on a single *ElmLne* object

Figure 11.3.7: Cable System Overview

The cable system shown on figure 11.3.7a illustrates two line objects (*ElmLne*) which are coupled together using a Cable System Element object (*ElmCabsys*). Although only two line objects are illustrated it is important to realise that practically there is no limit to the number of *ElmLne* objects which can be coupled. Further, the *ElmLne* objects being coupled can have 1,2 or 3 phases and they can represent many different types of conductor for example a phase conductor, a neutral, a sheath or an armour with the type of conductor they represent likely to correspond with the designation of the node on which they are terminated. They can also be connected at different voltage levels. Each *ElmLne* representing one or more phase conductor is referred to as a circuit in the *ElmCabsys* and *TypCabsys* objects. Each circuit must be assigned a Single Core Cable type (*TypCab*) or a Multicore/Pipe Cable type (*TypCabmult*). If sheaths or armours of the cables are to be considered then this is specified in the *TypCab*or *TypCabmult* objects.

Since coupling generally occurs between cables sharing a route over the part of the route where the cables run approximately parallel to each other, the cables coupled in this cable system arrangement should be specified to have the same length; if they are not, a warning message will be displayed when calculations are executed and the shorter length will be considered for the coupling. To facilitate this Line objects can divided into two objects with one of the divided parts assigned a length as well as a coupling to other line objects of the same length. The other divided part can be assigned the remainder of the length of the circuit and no coupling.

The line coupling can be directly defined in the Data Manager; however it is easier to do it from the single line diagram as follows:

1. Multi-select the cables to be coupled which are drawn in the single line diagram. Right-click and select *Define* → *Cable System* from the context sensitive menu.
2. A dialog pointing to the *Equipment Type Library* will open. At this point you are asked to select a *TypCabsys* object. If no appropriate existing types are available, the button *New Object* ( ) can be used to define a new Cable System Type.
3. On the basic data page of the dialog of the tower type, shown in figure 11.3.8, the number of

circuits should be defined. Then the cable types should be selected, by double clicking on the *TypCab*, *TypCabmult* field.

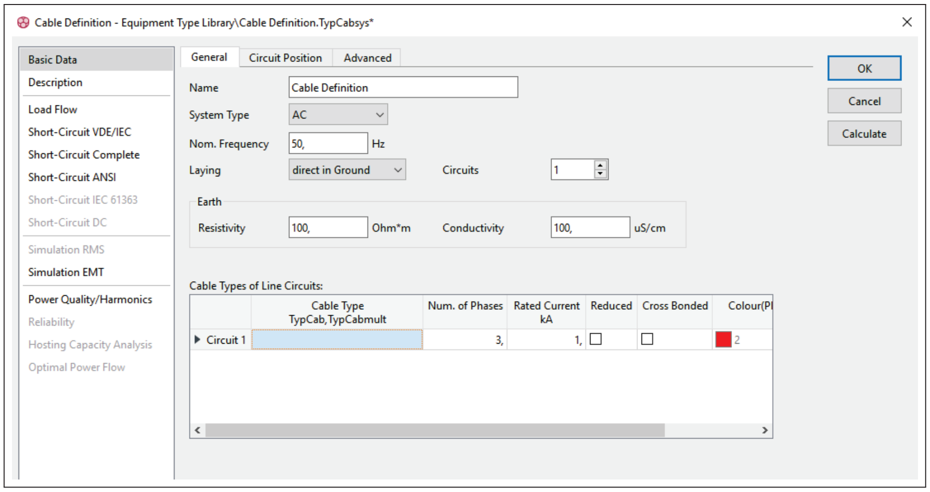


Figure 11.3.8: Cable Definition Type (*TypCabsys*)

4. Once again, a dialog pointing to the *Equipment Type Library* will open for selection of a Single Core or Multicore/Pipe Cable Type. If no appropriate type is yet defined, the button *New Object* ( ) can be used to define a new type.
5. On the Basic Data page of the dialog of the Single Core Cable type, shown in figure 11.3.9, the rated voltage, and details of the core construction should be defined along with the characteristics of the various conducting, insulating and semi-conducting layers from which the cable is constructed. If a sheath or armour is to be considered then the relevant checkbox should be selected to confirm that it exists.

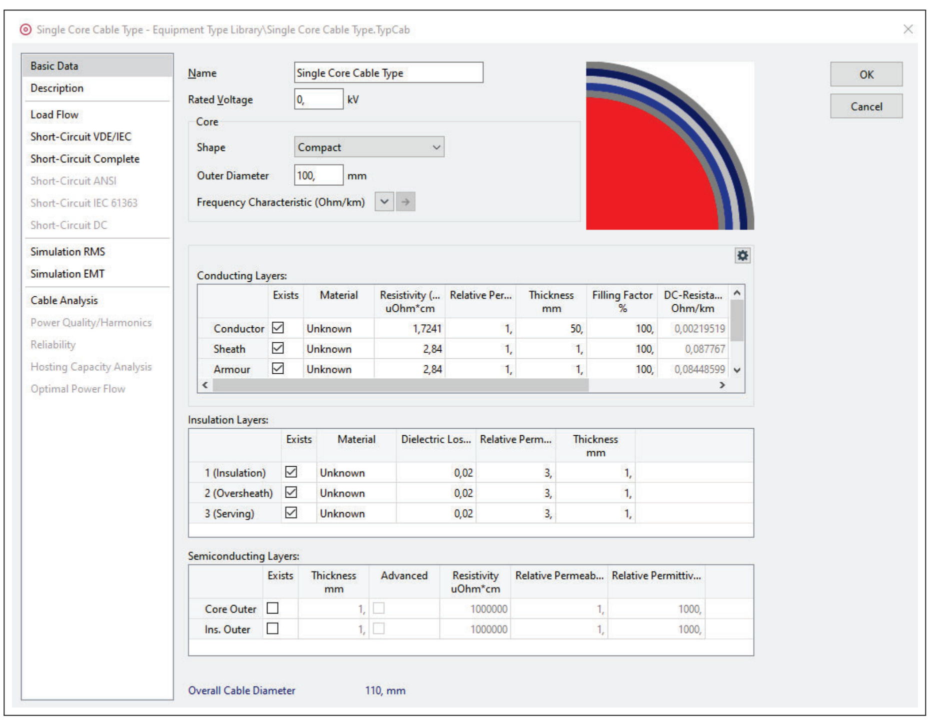


Figure 11.3.9: Single Core Cable Type (*TypCab*)

6. On the Basic Data page of the dialog of the Multicore/Pipe Cable Type, shown in figure 11.3.10, just as for the Single Core Cable type the rated voltage, and details of the core construction should be defined along with the characteristics of the various conducting, insulating and semi-conducting layers from which the cable is constructed. Again, if a sheath or armour is to be considered then the relevant checkbox should be selected to confirm that it exists. Note that additionally here it is possible to select whether the Pipe Type model or the Multicore model is to be used. Note also that depending on which model is selected, different data may be entered in the additional tabs defining either the multicore common outer layers or the pipe. Finally the geometry of the conductors within the arrangement should be defined using the Conductor coordinates tab.

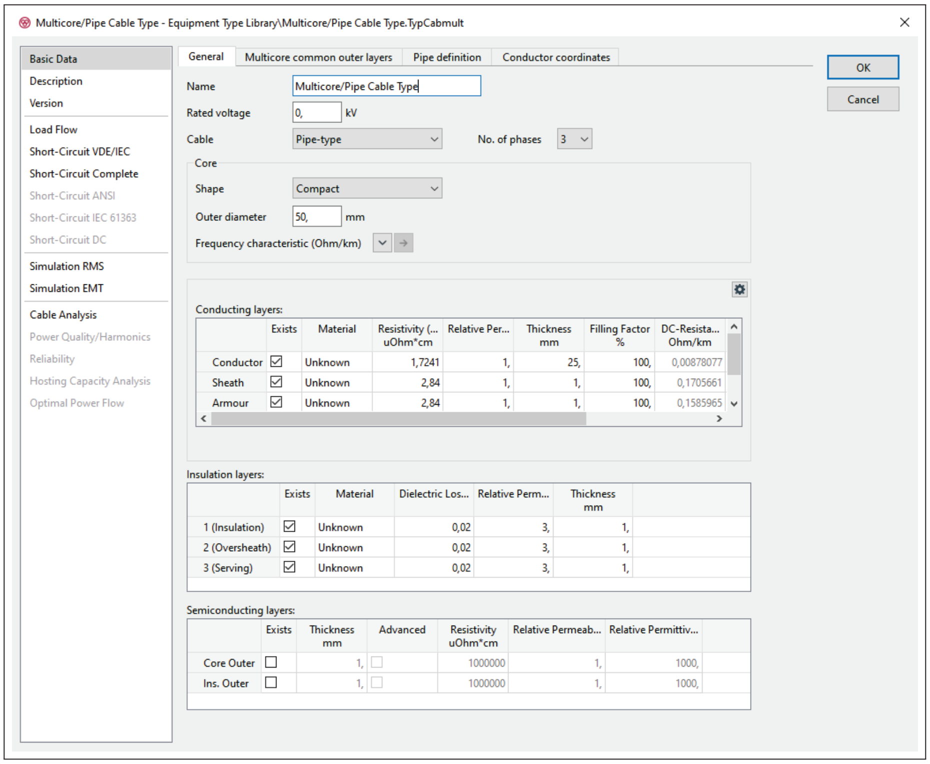


Figure 11.3.10: Multicore/Pipe Cable Type (*TypCabmult*)

7. Separate Single Core Multicore/Pipe Cable types can be used for each circuit.
8. Once the types are defined, the rest of the parameters of the Cable Definition should be entered.
9. On the *Circuit Position* tab of the Cable Definition Basic Data dialog, the positions of the circuits can be defined by inserting the coordinates, on the right side of the dialog, an image of the location of the phases is shown. The button **Calculate** can be used to get the matrix of impedances; more information about the calculation and values obtained is also available in the technical reference for Cable Systems.

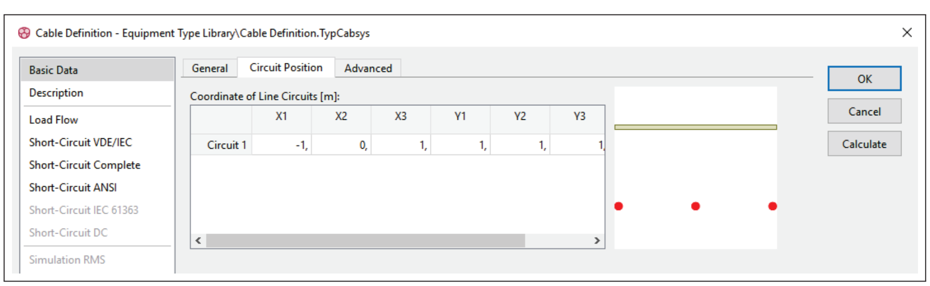


Figure 11.3.11: Geometry of the Cable Definition (*TypCabsys*)

10. Once the Cable Definition is defined, click on the **OK** button. A new dialog will open, where the lines (*ElmLne*) are assigned to the circuits. Select the line for each circuit and click **OK**. Now the Cable System element (*ElmCabsys*) is complete. Note that once a line coupling has been assigned to a line element, the type of the line changes to line coupling.

The cable system on figure 11.3.7b illustrates a more simple cable system configuration where coupling between *ElmLne* objects is not to be considered. In this case a Cable definition (*TypCabsys*) is assigned directly to an individual line object. This line object can also represent 1,2 or 3 phases and can also represent many different types of conductor for example a phase conductor, a neutral, a sheath or an armour with the type of conductor they represent again likely to correspond with the designation of the node on which they are terminated.

The *ElmLne* objects associated with this configuration can actually be used to represent multiple circuits and for considering the coupling between them. However, there is one limitation in that each circuit must be identical. Each circuit must be assigned the same Single Core Cable type (*TypCab*) or a Multicore/Pipe Cable type (*TypCabmult*). If sheaths or armours of the cables are to be considered then this is specified in the *TypCab* or *TypCabmult* objects. However, in this case these cannot be considered explicitly by representation with their own *ElmLne* object as they could be with the coupled cable system.

For this configuration it is easiest to define the cable system via the associated *ElmLne* dialog. In this case the type parameter of the line should be selected or defined with selection of the class *TypCabsys*. The *TypCabsys* itself is defined exactly as described for the coupled cable system.

11.4 Neutral Winding Connection in Network Diagrams

PowerFactory offers the user the option to explicitly represent the neutral connections and interconnections of the following commonly used elements:

- Power transformers (*ElmTr2*, *ElmTr3* and *ElmTr4*)
- Shunt elements (*ElmShunt*)
- External grids (*ElmXnet*)
- Synchronous (*ElmSym*) and asynchronous machines (*ElmAsm*)
- Static generators (*ElmGenstat*)
- PV systems (*ElmPvsys*)
- Neutral earthing elements (*ElmNec*)
- Harmonic Filter (*ElmFilter*)
- Step-Voltage Regulator (*ElmVoltreg*)

The interconnection of separate neutral wires is illustrated with the help of the Synchronous Generator.

A separate neutral connection can be activated by choosing the option N-Connection on the Zero Sequence/Neutral Conductor tab on the basic data page of the element as shown in figure 11.4.1, the graphical symbol of the object will change. An illustration for the Synchronous Generator element is shown in figure 11.4.2. Please note, once the N-Connection via a separate terminal option is selected, the Vector Groups layer can no longer be hidden in the single line diagram.

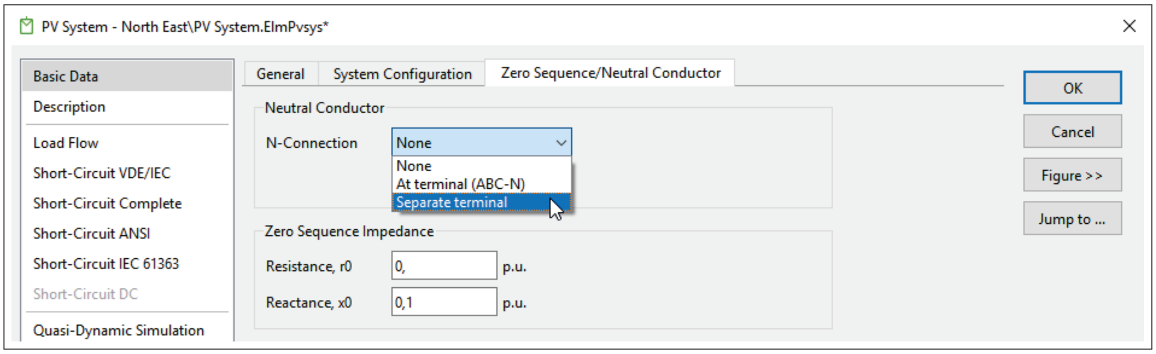


Figure 11.4.1: Zero Sequence/Neutral Connection Tab

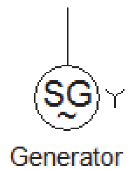


Figure 11.4.2: Generator with N-Connection via separate terminal

To connect the neutral of the Element to a neutral busbar, right click on the element and select Connect Element. An example of a single line diagram with the interconnection of neutral wires is shown in figure 11.4.3. A Neutral terminal is configured by ensuring that the Phase Technology of the terminal is set to N as shown in figure 11.4.4.

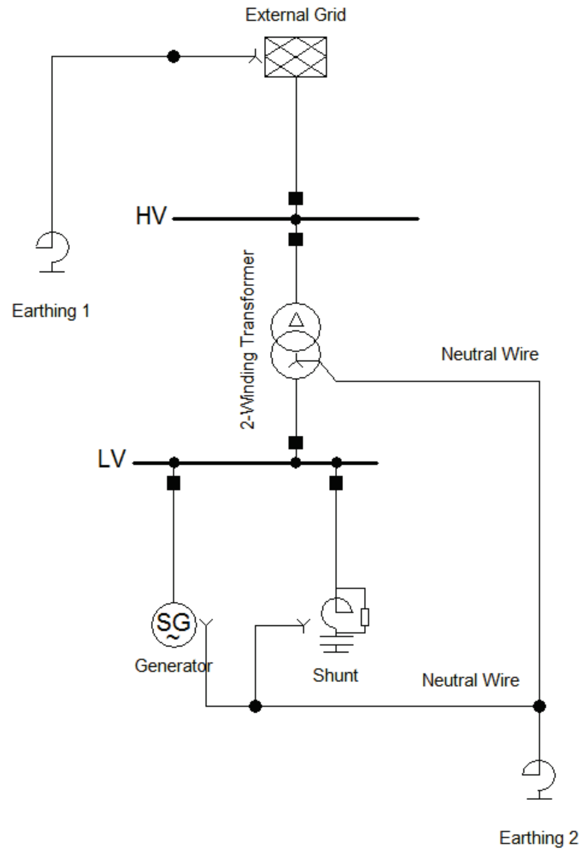


Figure 11.4.3: Grid with neutral winding connection

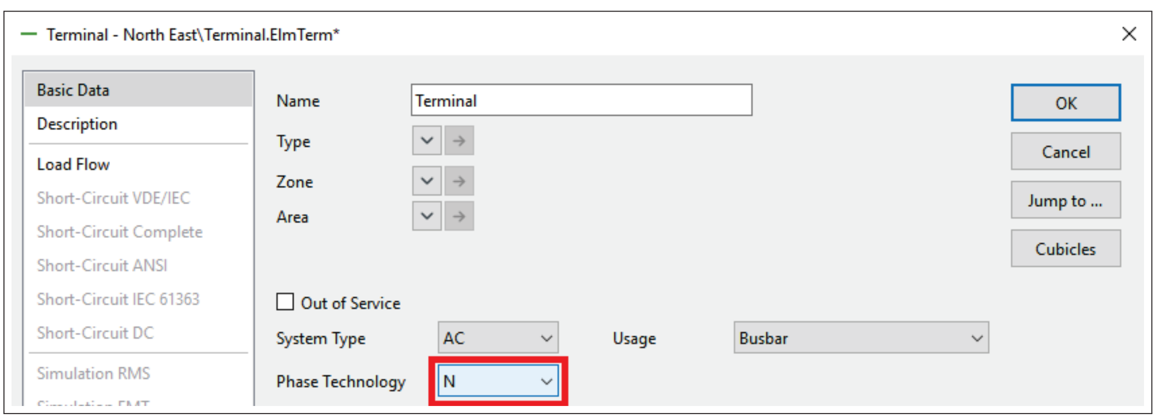


Figure 11.4.4: Set neutral Terminal

11.5 Defining Network Models using the Data Manager

In this section it is explained how the tools of Data Manager are used to define network models.

11.5.1 Defining New Network Components in the Data Manager

This section deals with defining and connecting network components using the Data Manager. General information about editing data objects in the Data Manager can be found in section 10.5.

New network components can be directly created in the Data Manager. This is done by clicking on the target grid/expansion stage (left pane) to display its contents in the browser (right pane). Then the New Object icon is used; the required object class is selected or the class name typed in directly.

11.5.2 Connecting Network Components in the Data Manager

To connect newly created branch elements to a node, a free cubicle must exist in the target terminal. In the 'Terminal' field (Terminal i and Terminal j for two port elements, etc.) of the edge element you have to click on the ( ) arrow to select (in the data browser that appears) the cubicle where the connection is going to take place.

To create a new cubicle in a terminal you have to open its edit dialog (double click) and press the **Cubicles** button (located at the right of the dialog). A new browser with the existing cubicles will appear, press the *New Object* icon and in the 'Element' field select *Cubicle (StaCubic)*. The edit dialog of the new cubicle will appear; by default no internal switches will be generated. If you want a connection between the edge element and the terminal through a circuit breaker, you have to press the **Add Breaker** button. After pressing **OK** the new cubicle will be available for connecting new elements.

Note: New users are recommended to create and connect elements directly from the single line graphics. The procedures described above are intended for advanced users.

11.5.3 Defining Substations in the Data Manager

The concept and the application context of substations is presented in Section 4.6 (Project Structure). A description of the procedure used to define new substations with the Data Manager is given as follows. For information about working with substations in the graphical editor refer to Section 11.2 (Defining Network Models using the Graphical Editor).

To define a new substation from the Data Manager:

- Display the content of the grid where you want to create the new substation.
- Right click on the right pane of the Data Manager and select *New* → *Substation* from the context sensitive menu.
- The new substation edit dialog will appear. There you can change the name, assign running arrangements and visualise/edit the content of the substation (directly after creation it is empty).
- After pressing **OK** the new substation and an associated diagram (with the same name of the substation) will be created.

The components of the new substation can be created and connected using the associated single line diagram or using the Data Manager; the first option is recommended. For the second option, the Contents button is used to bring up a browser, where the new components can be created using the New Object icon.

Components of a substation can of course be connected with components of the corresponding grid or even with components of other networks. The connection in the Data Manager is carried out following the same procedure discussed in the previous section.

For information about working with substations in the graphical editor refer to Section 11.2 (Defining Network Models using the Graphical Editor). For information about the definition of Running Arrangements refer to Section 14.3.10 (Running Arrangements).

11.5.4 Defining Composite Branches in the Data Manager

The concept and the application context of composite branches (*ElmBranch*) is discussed in Section 4.6 (Project Structure), and a description of how to define branches from within the diagram is provided in Section 11.2 (Defining Network Models using the Graphical Editor). This section explains how to define new branches from within the Data Manager.

Branches can be defined in the Data Manager as follows:

1. To create a Branch template, navigate to the *Library* → *Templates* folder in the Data Manager.
2. Right-click on the right pane of the Data Manager and select *New* → *Branch* from the context sensitive menu.
3. In the branch edit dialog, define the name of the branch and press **OK**.
4. Now navigate back to the branch edit dialog (right-click and 'edit', or double click), and select **Contents** to add terminal and line elements etc. to the template as required. The internal elements can be connected as discussed in Section 11.5.2.
5. Use the fields 'Connection 1' and 'Connection 2' to define how the branch is to be connected to external elements.
6. To create an instance of the Branch from the created Branch template, either:
   - Select the Composite Branch icon and connect the branch to existing terminals on the Single Line Diagram.
   - Select the Composite Branch icon and place the branch on the single line diagram, press **Tab** twice to place the branch without making any connections. Then connect the branch to external elements by right-clicking and selecting 'Connect', or double-clicking the branch and selecting external connections for the relevant internal elements (e.g. lines). Select **Update** on in the Branch dialog to update the external connections.

Alternatively, for a single Branch (i.e. not using Templates) the branch can be defined in the grid folder.

11.5.5 Defining Sites in the Data Manager

The concept and the application context of sites are presented in the Section 4.6 (Project Structure).

To define a new site from the Data Manager do the following:

- Display the content of the grid where you want to create the new site.
- Right click on the right pane of the Data Manager and select *New* → *Site* from the context sensitive menu.
- The new Site edit dialog will appear.
- After pressing **OK** the new site will be created.

Note: It is possible to move objects from a grid to a Substation, Branch, Site, etc. and vice versa.

11.6 Drawing Existing Elements using the Diagram Layout Tool

This section provides information about how to draw network components from existing objects. *PowerFactory* separates strictly the electrical (and therefore for calculations relevant) data of network elements from their graphical representation in the diagrams. Calculations of networks without any graphical representation is possible.

Designing new (extensions to) power system grids, is preferably done graphically. This means that the new power system objects may be created in a graphical environment. After the new components are added to the design, they can be edited, either from the graphical environment itself (by double-clicking the objects), or by opening a Data Manager and using its editing facilities.

It is however possible, to first create objects in the Data Manager (either manually, or via data import using e.g. the DGS format), and subsequently draw these objects in one or more single line diagrams. If the imported data contains geographical coordinates, a geographic diagram can be created automatically by right clicking on the Grid in the Project Overview window and choosing *Show Graphic* → *Geographic Diagram*.

If no geographic coordinates are given or if a single line diagram should be created, *PowerFactory* provides the *Diagram Layout Tool* to do that.

The following sections describe the options and possibilities of the *Diagram Layout Tool*, located in the graphic icon bar. It replaces the *Draw Existing Net Elements* tool of previous versions and enhances its functionality by a semi- and fully automated creation of network diagrams.

11.6.1 Action

11.6.1.1 Generate new diagram for

When this option is selected from the *Action mode* part of the Diagram Layout Tool dialog, it is possible to create graphical representations of grids and network elements. It's a quick way to get a graphical overview of a network, offering visualisation of, for example, results or topology (colouring schemes for feeders, zones, etc.). The options in the *Generate new diagram for* part of the dialog are:

- **entire grid:** with this option a complete new diagram of the selected grid is automatically drawn. It is possible to select more than one grid; in this case one diagram showing all the selected grids will be created.
  Additional settings when using the option *Generate new diagram* → *Entire grid* are set in pages *Node Layout* (section 11.6.2), *Edge Elements* (section 11.6.3) and *Protection Devices* (section 11.6.4)
- **detailed representation of:** this option can be used for substations, branches and sites. It creates a detailed diagram with all the elements contained inside the original element. No additional settings are needed.
- **k-neighbourhood:** if this option is used, a complete new diagram will be generated, starting with the selected elements and extending as far as specified. The "k-neighbourhood" logic is described below in section 11.6.1.2.
- **feeder:** with this option a complete new schematic feeder diagram is created. It is possible to select more than one feeder; in this case a separate diagram will be created for each feeder.
  This option replaces the previous option *Show* → *Schematic visualisation* by Distance or Bus Index of the feeder. See Section 15.6 (Feeders) for further information on how to define feeders.
  Additional settings when using the option *Generate new diagram* → *Feeder* are set in pages *Node Layout* (section 11.6.2) and *Edge Elements* (section 11.6.3).
- **area interchange type:** this option generates an interchange diagram, used to show the power flows between defined parts of the network. Grids, Areas or Zones may be selected. The diagram

will consist of summary boxes for those regions, and arrows showing power flows between the regions, held in a separate, configurable graphic layer. The thickness of each power arrow indicate the relative size of the power interchange.

11.6.1.2 Auto-insert elements into current diagram

When this option is selected from the *Action mode* part of the Diagram Layout Tool dialog, it is possible to insert additional elements into an existing diagram. This option is only available if the diagram is not in “freeze” mode.

The options in the *Insert elements into current diagram* part of the dialog are:

- **K-neighbourhood expansion:** this action creates graphical elements starting from a selection of elements already graphically represented in the diagram. A selection of elements is therefore necessary. If some or all graphic elements are selected before opening the Diagram Layout Tool, these elements are automatically inserted into the Start elements selection. Alternatively the Start elements can be selected directly from the dialog using *Select...* or in the *Neighbourhood Expansion* settings.
Starting from every selected element, the connected and not yet graphically represented neighbours are created and subsequently also their neighbours. The depth of this recursive algorithm is defined by the K-factor, which can be configured in the *Neighbourhood Expansion* settings.
This approach offers a step-by-step creation of a diagram, where an intervention after each step is possible to adapt the final appearance of the network diagram.
Additional settings when using the *Auto-insert element into current diagram* → *K-neighbourhood expansion* option are set in pages *Node Layout* (section 11.6.2), *Edge Elements* (section 11.6.3) and *Protection Devices* (section 11.6.4)
- **Edge elements:** this action automatically completes the current diagram with the branch elements which are not yet graphically represented. It is only available for diagrams which already contain some existing graphical node elements. Additional settings when using the option *Auto-insert element into current diagram* → *Edge elements* are set in pages *Edge Elements* (section 11.6.3) and *Protection Devices* (section 11.6.4)
- **Protection devices:** when this option is selected, protection devices are included into the current diagram according to the options set in the *Protection Devices* page described in section 11.6.4.
- **Bays and Sites:** this option enables the graphical representation of sites and bays to be added to existing diagrams, taking into account the options on the Bays and Sites page.
- **Area interchange type:** the graphical representation of the power interchanges between network regions (Grids, Areas or Zones) is added to currently-active graphic. This consists of summary boxes for those regions, and arrows showing power flows between the regions, held in a separate, configurable graphic layer. The thickness of each power arrow indicate the relative size of the power interchange.

11.6.1.3 Assisted manual drawing

This action replaces the earlier *Drawing existing Net Elements* tool. Upon execution, a window will appear, listing all the elements which are not yet graphically represented in the diagram. This option is only available if the diagram is not in “freeze” mode.

Drawing Existing Busbars

Click on the symbol for busbars (—) in the drawing toolbox. The symbol of the busbar (terminal) is now attached to the cursor.

If the list is very large, press the button *Adjacent Element Mode* (). This activates the selecting of distance (number of elements) from elements in the selection of the *Neighbourhood Expansion*. Select the Distance of 1 in order to reduce the number of busbars (terminals) shown.

If the button *Use drawn nodes as starting objects* ( ) is also selected, the list will be filtered based on all drawn nodes (not just a single starting node).

If *Show elements part of drawn composite nodes* ( ) is selected, elements internal to already drawn composite nodes will be shown in the list. However, since they are already drawn as part of the composite node, they should not be re-drawn.

The marked or selected element can now be visualised or drawn by clicking somewhere in the active diagram. This element is drawn and disappears from the list.

Note that the number of elements in the list can increase or decreases depending on how many elements are a distance away from the element lastly drawn. Scroll down the list, in case only certain elements have to be visualised.

Close the window and press **Esc** to return the cursor to normal. The drawn terminals (busbars) can be moved, rotated or manipulated in various ways.

Drawing Existing Lines, Switches, and Transformers

Similar to the busbars, elements like lines and transformers connecting the terminals in the substation can be drawn.

Execute the *Assisted manual drawing* action of the *Diagram Layout Tool*. For lines select the line symbol ( ) from the drawing toolbox, for transformers select the transformer symbol ( ), and so on.

Similar to terminals, a list of all the lines (or transformers, or elements which have been chosen) in the network, that are not in the active diagram, is shown.

For each selected line (or transformers...) a pair of terminals, to which the line is connected, are marked in the diagram. Click on the first terminal and then on the second. The selected line is drawn and removed from the list of lines.

Continue drawing all lines (or transformers...), until the list of lines is empty or all the lines to be drawn have been drawn. If a branch cannot be completely drawn (for example, when the terminal at only one end of a line is shown on the diagram), it is possible to draw a first line section, then press **Tab** or double click on the diagram and arrows will appear to indicate that the line connects to a terminal that is not shown.

Note: Before placing elements onto the graphic users may find it useful to configure and display a background layer. This will be an image of an existing single line diagram of the system. It may be used to 'trace' over so that the *PowerFactory* network looks the same as current paper depictions; see Section 9.3.6 for more information on layers.

11.6.2 Node Layout

The settings regarding the Node Layout take effect on the following actions:

- Generate new diagram for
  - entire grid
  - k-neighbourhood
  - feeder
- Auto-insert element into current diagram
  - K-neighbourhood expansion

The following options are available for node insertion:

- **Node spacing:** this option defines the distance between the newly created nodes in the diagram and can be set to *low*, *medium* or *high*.
- **Draw each composite as single node:** this check box only has an impact if the corresponding grid contains composite elements (e.g. *ElmSubstat*, *ElmTrfstat*, *ElmBranch*) which graphically combine internal nodes, switches, transformers, lines, etc. If checked, the graphical representation of the composite elements are created, otherwise each of the internal elements of the composite elements is created separately in the diagram.
- **Consider physical line length:** with this option checked, the length of the graphical representation is based on the corresponding line length. The graphic object length is not strictly proportional to the actual line length, but nevertheless gives a good view in the diagram of the relative line lengths.
- **Adjust diagram size:** The size of the diagram defined in the *Drawing Format* is ignored and overwritten by the algorithm, which uses as much space as is needed. To get clearer outputs, this option should be selected. The new drawing size is saved and can be reused in other diagrams.

If the option *Generate complete diagram* → *Feeder* is selected, the options of the Node Layout page include:

- **Layout Style:** this option defines the layout of the feeder; the options are *Rectangular* and *Tree*. *Rectangular* is usually recommended, since it provides the best overview of the topology of the feeder. For very large feeders, however, the rectangular layout may become too large. In this case the tree-like layout may be better, since it produces a narrower layout.
- **Horizontal/Vertical node spacing:** this option defines the distance between the feeder nodes, can be set to *low*, *medium* or *high*.
- **Draw each composite as single node:** as explained in the options for node insertion.
- **Consider backbones:** if selected, backbones (if available) are emphasised by laying them out strictly vertically (straight line from top to bottom). Otherwise, the longest path within the feeder will be laid out vertically.

11.6.3 Edge Elements

The settings on the Edge Elements page take effect on the following actions:

- Generate new diagram for
  - entire grid
  - k-neighbourhood
  - feeder
- Auto-insert element into current diagram
  - K-neighbourhood expansion
  - Edge elements

The following options are available in the Edge Elements page:

- **Insert edge elements:** if this is not checked, the Diagram Layout Tool only creates graphical representations of nodes (or composite elements, if the *Draw each composite as single node* option is selected in the Node Layout page). If the option *Auto-insert element into current diagram* → *Edge elements* is selected, the edge elements are always inserted and so the setting of this option is ignored in that case.
- **Ortho Type:** if set to *Ortho*, all inserted branch elements will consist of only vertical or horizontal sections. The opposite option is *Ortho Off*, where the branch elements show a direct point-to-point connection between the according start and end nodes. With the option set to *Semi-Ortho*, the branches have a orthogonal part near the start and end node and in between a direct connection.

- **Insertion of one-port devices connected to substations:** this option should be checked if the option *Draw each composite as single node* is selected from the *Node Layout* page, but the user still wishes to show the one port elements (e.g. loads, shunts) connected to the composite node of the substation in the diagram.
- **Insert classes only:** this option can be used as a filter to insert elements of particular classes. To achieve that, the class name of the element can be mentioned in the provided field. Also, there is a possibility for the user to insert the elements of different classes and for that purpose a comma operator "," should be used to separate the class names.

11.6.4 Protection Devices

The settings on the Protection Devices page take effect on the following actions:

- Generate new diagram for
  - entire grid
  - k-neighbourhood
- Auto-insert element into current diagram
  - K-neighbourhood expansion
  - Edge elements
  - Protection devices

The following options are available in the Protection Devices page:

- **Insert protection devices:** if checked, the Diagram Layout Tool inserts graphical representations of the protection devices. If the option *Auto-insert element into current diagram* → *Protection devices* is selected, the protection devices are inserted anyway.
- **Relays:** to insert graphical representations of relays (*ElmRelay*).
- **CTs and VTs:** to insert graphical representations of current transformers (*Stat*) and voltage transformers (*StaVt*) .

An example of automatically inserted protection devices is shown in figure 11.6.1

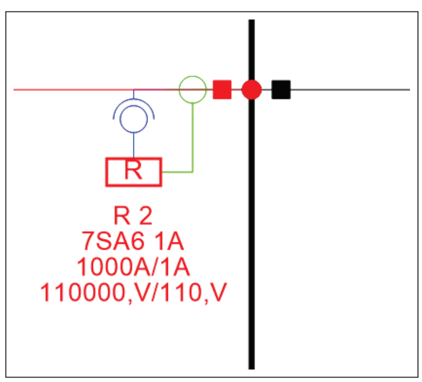


Figure 11.6.1: Protection devices in the single line diagram

11.6.5 Bays and Sites

The settings on the Bays and Sites page take effect on the following actions:

- Generate complete diagram
  - Entire grid

  - k-neighbourhood

- Auto-insert element into current diagram
  - K-neighbourhood expansion
  - Edge elements
  - Bays and sites

On the Bays and Sites page, the options to show the Bay and Site representations can be selected independently. It should be noted that if *Auto-insert element into current diagram* and *Bays and sites* are selected on the Actions page, this will override the main *Insert bays and sites* option on the Bays and Sites page; however, the two individual sub-options Bays and Sites will be observed.

11.6.6 Interchanges

The settings on the Interchanges page take effect on the following actions:

- Generate new diagram for
  - area interchange type

The options on the Interchanges page give the user some control over the positioning of elements on the graphic:

- **consider positions from current diagram** is typically used if an overview diagram is currently active, for example.
- **prefer geographic positions** will use the Geographical Coordinates of the network elements.

If nothing is selected, the layout will be determined automatically.

11.7 Drawing Existing Elements using Drag & Drop

As an alternative to using the Diagram Layout tool, it is possible to create graphical objects for existing network elements by using drag & drop:

1. Enable the *Drag & Drop* feature in a Data Manager window by double-clicking the *Drag & Drop* message in the status bar.
2. Select the data object in the Data Manager by left clicking on its icon.
3. Hold down the left mouse button and move the mouse to the graphic drawing area (drag it).
4. Position the graphical symbol and release the mouse button to drop the object.
5. A new graphical symbol is created, which is representing the selected element in the diagram. No new data object is created.

This approach may lead to problems and should therefore be used carefully.

Chapter 12

Network Model Manager

12.1 Introduction

The Network Model Manager is a browser for all calculation relevant objects. The objective of this chapter is to provide detailed information about this data management tool. Before starting, users should ensure that they are familiar with Chapter 4 (*PowerFactory* Overview).

12.2 Using the Network Model Manager

The Network Model Manager shows all objects relevant for the calculation. It can be accessed by clicking the button *Open Network Model Manager...* in the main icon bar.

On the left hand side of the Network Model Manager window the classes of all calculation relevant objects are displayed with their names and symbols. To give a good overview, they are sorted into groups, such as *Substations/Terminals/Switches* or *Lines/Series Impedances/Transformers*. By double-clicking on the group's name, its contents can be hidden or shown. If one of the classes is selected, all calculation relevant objects will be listed on the right side of the browser window and *Detail Mode* automatically activated. This makes it easy to edit the parameters of an object, without using the object dialog.

Figure 12.2.1 shows the Network Model Manager window, where some of the groups are collapsed and the class *Busbar* is selected.

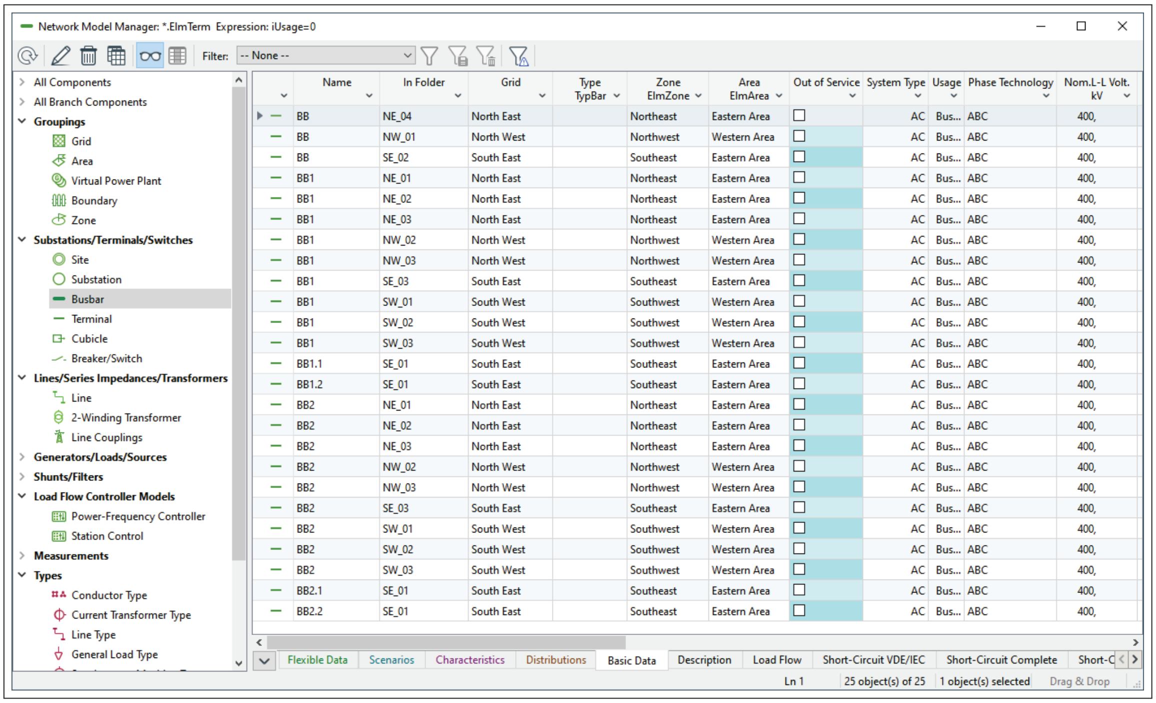


Figure 12.2.1: The Network Model Manager window

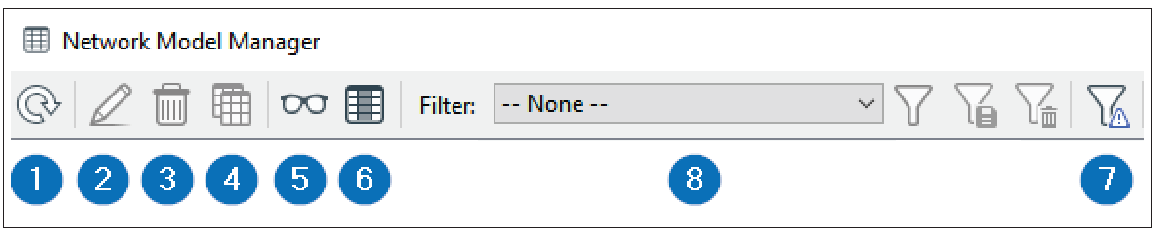


Figure 12.2.2: Icon bar of the Network Model Manager

The functions of the buttons of the icon bar of the Network Model Manager shown in Figure 12.2.2 are:

1. The *Refresh* button can be used to update the Network Model Manager. This is necessary, for example, when objects are deleted in the Data Manager or the Single line diagram while the Network Model Manager is open. The deleted objects will still be displayed in the Network Model Manager until the refresh button is pressed.

2. A mouse click on the *Edit Object* button will open the dialog window of the object selected on the right hand side. A selected object is indicated by a marked row and an arrowhead next to the object symbol. Another possibility to open the edit dialog of an object is to double click on the corresponding symbol of the object.

3. One or more objects can be deleted by marking the objects to be removed and then pressing the *Delete Object* button. A query window will appear to request confirmation of the deletion (unless this has been suppressed via the relevant user setting).

4. The button *Copy (with column headers)* copies the data of a selection with the corresponding column header(s) into the Windows Clipboard. The content can then, for example, be pasted into a spreadsheet.

5. The *Detail Mode* can be activated and deactivated.
   - If the *Detail Mode* is activated (button is pressed), the table on the right will display all the object parameters of the class (e.g. Busbars, as shown in Figure 12.2.1). The tabs at the bottom of the table give access to the same pages that are available via the object dialog window. The Flexible Data, Scenarios, Characteristics and Distribution tabs are highlighted by colouring. The colours reflect those selected in the User Settings (refer to section 7.8) for these types of data.
   - If the *Detail Mode* is deactivated, a table will appear, in which columns with predefined parameters are shown. This table and the Flexible Data page (available in *Detail Mode*) can be extended, by pressing the *Variable Selection* button.
6. By clicking on *Variable Selection* , variables/parameters to be displayed in the table can be chosen. For more information about how to handle the selection dialog, refer to Section 19.3 (Variable Selection).
7. Objects which are Out-of-Service or non-relevant for the calculation can be hidden by clicking the button *Hide Out-of-Service-Objects and Objects non-relevant for Calculation* .
8. The Network Model Manager provides the following filter functions, which are available when at least one column is filtered. How to filter columns is described in section 10.4 (Auto-Filter functions in Data Manager and browser windows).
   - If one or more columns are filtered, the button *Edit Filter* will become accessible. After the button has been pressed, an edit dialog will open, in which a filter name can be entered. Furthermore the filters for each parameter can be modified by double clicking the relevant cell in the Filter column.
   - The Current Working Filter is temporary. That means, if the Network Model Manager window is closed, the applied filters will all be discarded. However, one or more column filters can be consolidated to one common filter, which can then be saved under any name to reuse it, by clicking the *Save Filter* button. A window will appear, to enable the user to name the filter.
   - Unwanted filters can also be deleted from the list of saved filters, by selecting a Filter from the drop-down list in the icon bar and clicking on the *Delete Filter* button.

Chapter 13

Study Cases

13.1 Introduction

The concept of study cases was introduced in Chapter 4 (*PowerFactory* Overview). Study cases (*IntCase*,) define the studies to be performed on the system being modelled. They store everything created by the user to perform calculations, allowing the easy reproduction of results even after deactivation/reactivation of the project. By means of the objects stored inside them, the program recognises:

- Parts of the network model (grids and expansion stages) to be considered for calculation;
- Calculations (and their settings) to be performed on selected parts of the network;
- Study time;
- Active variations;
- Active operation scenario;
- Calculation results to be stored for reporting;
- Graphics to be displayed during the study.
- The last-selected function toolbox (i.e. which function toolbox was last selected using on the main toolbar)

A study case with a reference to at least one grid or expansion stage has to be activated in order to enable calculations. A project that contains more than one grid, which has several expansion stages for design alternatives, or which uses different Operation Scenarios to model the various conditions under which the system should operate, requires multiple study cases. All of the study cases in a project are stored inside the study cases folder () in the project directory.

Note: Only one study case can be active at a given time. When activating a study case, all the grids, Variations and Operation Scenarios that it refers to will also become active.

Without study cases, it would be necessary to manually activate the relevant grid and/or expansion stage multiple times in order to analyse the resulting power system configuration. Similarly, it would be necessary to define over and over again the same calculation command setup used to analyse the behaviour of the selected network.

In addition to storing the objects that define a network study, study cases set the units used for the output of calculation results, and allow the definition of specific options for the calculation algorithms.

The following sections describe the main objects stored inside study cases.

13.2 Creating and Using Study Cases

When a new project is created, an empty study case is automatically created and activated. This new study case has default settings. The user can later modify these settings using the study case dialog.

The user may define several study cases to facilitate the analysis of projects containing more than one grid, several Expansion Stages, different Operation Scenarios or simply different calculation options. To create a new study case:

- Open the Data Manager and go to the study cases folder. Right-click on the folder and select *New* → *Study Case* from the context-sensitive menu. Enter the name of the new study case in the dialog that pops up and (if desired) modify the default settings.

Only one study case can be active at any given time. To activate or deactivate a study case:

- Open the Data Manager. The active study case and the folder(s) where it is stored are highlighted. Right-click on the active study case and choose *Deactivate* from the context-sensitive menu. To activate an inactive study case place the cursor on its name, right-click and choose *Activate*. Study cases may also be activated via the Project Overview Window (see Figure 13.2.1).

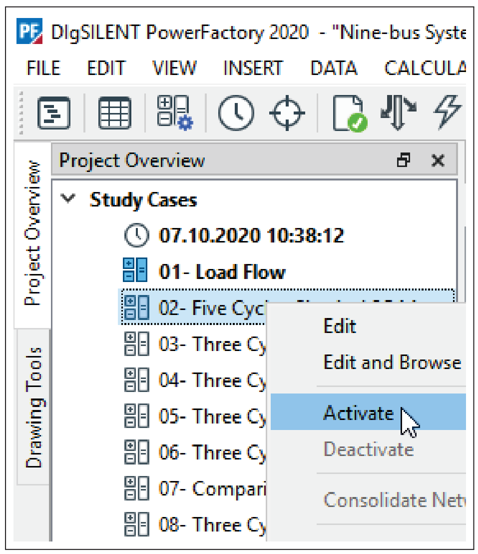


Figure 13.2.1: Activating a study case from the Project Overview window

A study case can have more than one grid. Only the objects in the active grids will be considered by the calculations. To add an existing grid to the active study case:

- Open the Data Manager and go to the Network Data folder. Right-click the desired grid and select *Activate* from the context-sensitive menu. The grid will be activated and any relevant graphics will be opened (following a user selection). To remove an active grid, select *Deactivate*.

Variations are considered by a study case when they are activated. The Expansion Stages are applied according to the study case time, which is set by the time trigger stored inside the study case folder. More than one variation can be active for a given study case. However there can be only one recording stage. For further information, refer to Chapter 17 (Network Variations and Expansion Stages). To add (activate) a Variation to the active study case:

- Right-click on the Variation and select *Activate* from the context-sensitive menu. The Variation will be activated and stages will be highlighted depending on the study time.

An Operation Scenario can be activated or deactivated via the context-sensitive menu, or by using the option *File* → *Activate Operation Scenario/Deactivate Operation Scenario* from the main menu.

Upon activation, a completeness check is performed (i.e. a check that operational data is available for all components). This is reported in the *PowerFactory* output window. If an Operation Scenario is active, all operational data attributes in property sheets or in the Data Manager are highlighted in blue. This indicates that changes to these values will not modify the base component (or Variation) but are recorded by the active Operation Scenario. Upon deactivation, previous operational data is restored. If the Operation Scenario was modified, user confirmation is requested regarding the saving of changes. For further information about working with Operation Scenarios, refer to Chapter 16 (Operation Scenarios).

Note: Only one study case can be activated at a time. Although network components and diagrams can be edited without an active study case, calculations cannot be performed. Variations and Operation Scenarios used by a study case are automatically activated upon activation of the corresponding study case.

13.2.1 The Study Case Manager

The Study Case Manager simplifies the management of study cases, by providing an overview of all existing study cases with all active Operation Scenarios, Variations, Grids and Triggers. There is a column for each study case, with the component objects listed on the left, as shown in Figure 13.2.2.

The Study Case Manager can be accessed by clicking on the icon on the main toolbar or from *Tools* → *Study Case Manager* in the main menu. Upon opening the Study Case Manager the active study case will be deactivated, but will be reactivated when the Study Case Manager window is closed.

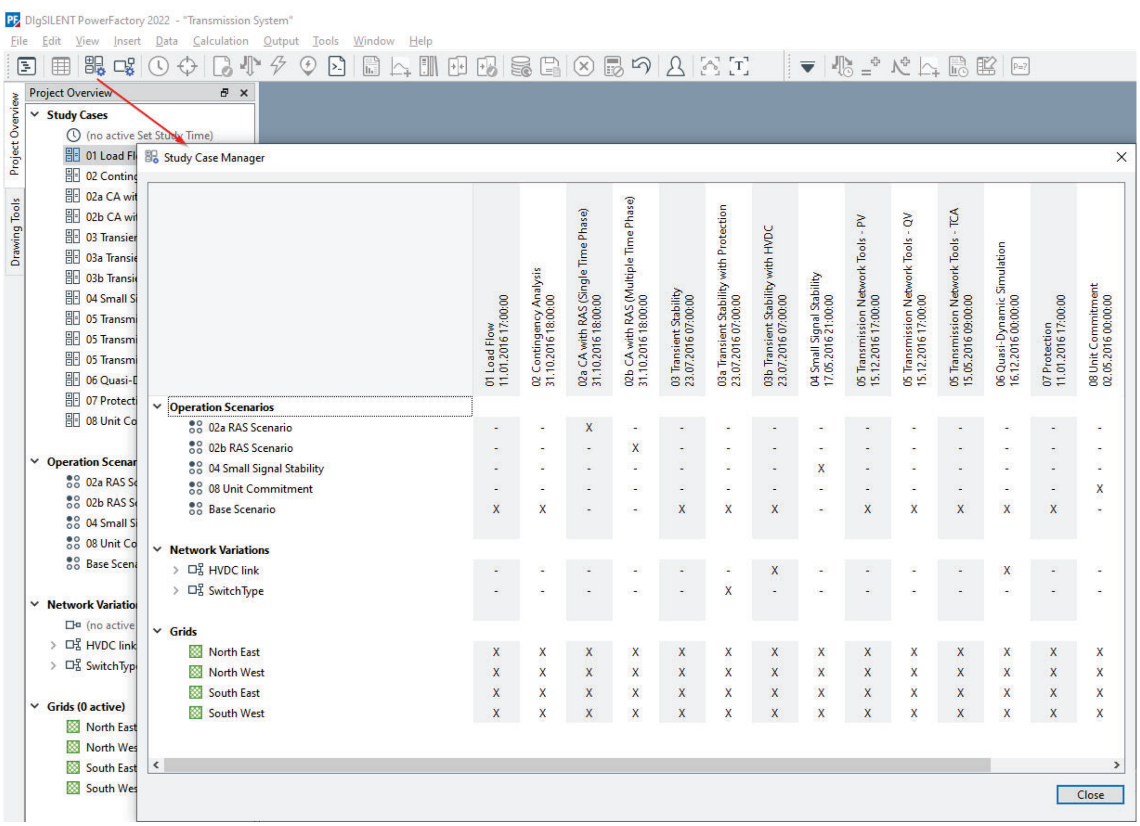


Figure 13.2.2: Study Case Manager

The Study Case Manager not only provides an overview but can also be used to manage the setup of individual study cases, as it allows the activation/deactivation of:

- Operation Scenarios
- Variations
- Grids
- Triggers

simply by double-clicking on the cell entries - without the need to activate the study cases themselves. Since the time of the active study case defines which Expansion Stage is active, it is only possible to activate or deactivate Variations, but not Expansion Stages. Depending on the study time, the recording Expansion Stage will be marked in bold.

Note: When folders are used to store study cases, only the study cases in the folder selected (within the Study Case Manager) will be shown.

13.3 Summary Grid

The primary task of a study case is to activate and deactivate a calculation target, which is a combination of grids and optionally expansion stages from the network model. The Summary Grid object holds references to the grids which are considered in the calculation (i.e. the active grids). Grids may be added to or removed from the currently active study case by right-clicking on them in the database tree or the project overview window and *Activate* or *Deactivate* them. A reference to the activated/deactivated grid is automatically generated in or deleted from the Summary Grid.

A grid cannot be activated without an active study case. With no study case active, the *Activate* action from the context-sensitive menu of a grid will show a dialog, where a new study case can be created or an existing one can be chosen in order to activate the grid.

13.4 Study Time

Study cases have a study time which defines the point in time to analyse.

The study time must be inside the Validity Period of the project, which specifies the time span for which the project is valid (see Chapter 8: Basic Project Definition, Section 8.1.2 (Project Settings)). *PowerFactory* will use the study time in conjunction with time-dependent network expansions (see Chapter 17: Network Variations and Expansion Stages) to determine which network data is applicable at that point in time. The study time may be changed in order to analyse a different point in time. The Expansion Stages will be activated/deactivated in accordance with the study time.

The status bar at the bottom of the *PowerFactory* program window shows the currently-set study time.

The simplest way to change the study time is:

- Double-click on the *Study Time* shown in the status bar of *PowerFactory*.
- Enter the date and time or press the **Date** and **Time** buttons in order to set the study time to the current time of the computer on which *PowerFactory* is being run.
- Press **OK** and close the window.

There are several alternative ways to edit the study time.

Alternative 1: Edit the study time like a trigger:

- Press the button **Date/Time of Calculation Case** in the main toolbar of *PowerFactory*.
- Enter the date and time or press the **Date** and **Time** buttons in order to set the study time to the current time of the computer on which *PowerFactory* is being run.
- Press **OK** to accept the changes and close the window.

Alternative 2: Edit the study case from within the study case dialog:

- Activate the project and browse for the study case in the Data Manager.
- Right-click on the study case and select *Edit* from the context-sensitive menu.
- On the *Basic Data* page use the drop-down arrow to bring up a standard calendar menu.
- Set the study time.
- Press **OK** and close the window.

13.5 The Study Case Dialog

To edit the settings of a study case, select *Edit* → *Project Data*→ *Study Case* from the main menu, or alternatively right-click on the study case in the Data Manager and select *Edit* from the context-sensitive menu. A dialog will appear.

13.5.1 Basic Data

On the *Basic Data* page, the user can define the name and an owner of the study case. The grids that are linked to a study case may be viewed by pressing the **Grids/System Stages** button.

The study time can be edited by using the drop-down arrow to bring up a standard calendar menu.

Please note that the study time can also change if the recording expansion stage is set explicitly (see Chapter 17: Network Variations and Expansion Stages).

Note: To edit the study time one can alternatively press on the "Date/Time of Calculation Case" button . This will open the study case time trigger window. In addition, the time of the simulation case is displayed in the lower-right corner of the program window. Double-clicking on this field provides access to the same window.

13.5.2 Calculation Options

The *Calculation Options* page is used to configure the basic algorithm for the study case calculations. The following options are available:

- **General** tab:
  - **Topological processing:** The *Breaker reduction* mode determines the internal calculation topology of the grid. In particular, electrically equivalent areas of a detailed substation are identified and merged for an efficient internal treatment. If the check box *Calculate results for all breakers* is ticked, results of reduced elements may then be post-calculated.
  - **Solution of linear equations:** The solution of linear equation systems is an intrinsic part of most calculations in *PowerFactory*, such as load flow, short-circuit or the RMS/EMT simulation. Since version 15.2, these equation systems can either be solved by a direct factorisation method or an iterative method. The latter method has been developed to meet the increasing demands of modern applications where interconnected, large-scale networks must be

analysed. In contrast to traditional direct methods, the implemented iterative solver is able to solve even very large systems with controlled precision.

  – The button **Set all calculation options to default** will restore all default options on the *Calculation Options* tab.

- The other tabs (**Jacobian Matrix**, **Optimisation Matrix**, **Simulation Matrix** and **Advanced**) include additional calculation options to tune the performance and the robustness of the linear system solver. Please note that alteration of default options is only recommended under the supervision of the *DIgSILENT* support experts.

The *Description* page is used for user comments.

13.6 Variation Configuration

Similar to the Summary Grid, the Variation Configuration object (*IntAcscheme*) contains references to the active variations.

13.7 Operation Scenarios

A reference to the active Operation Scenario (if any) is always stored in the study case. Similar to Variation Configurations and Summary Grids, when a study case is activated, the Operation Scenario (if any) whose reference is contained, will be automatically activated. The reference to the active Operation Scenario will be automatically updated by the program.

13.8 Commands

In *PowerFactory* a calculation (i.e load flow, short-circuit, initial conditions of a time-domain simulation, etc.) is performed via 'Calculation Commands', which are the objects that store the calculation settings defined by the user. Each study case stores its own calculation commands, holding the most recent settings. This ensures consistency between results and calculation commands and enables the user to easily reproduce the same results at a later date. When a calculation is performed in a study case for the first time, a calculation command is automatically created inside the active study case. Different calculation commands of the same class (i.e different load flow calculation commands: objects of the class *ComLdf* or different short-circuit calculation commands: objects of the class *ComShc*) can be stored in the same study case. This allows the user to repeat any calculation with identical settings (such as fault location, type, fault impedance, etc.) as last performed in the study case. The calculations are only performed on active grids (expansion stages).

Figure 13.8.1 shows a study case, 'Study 1' which contains two load flow calculation commands (, 'Ldf 1' and 'Ldf 2'), one command for an OPF calculation, one command for the calculation of initial conditions, and one transient simulation. The dialog of each of calculation command in *PowerFactory* is described in the chapter corresponding to that calculation function.

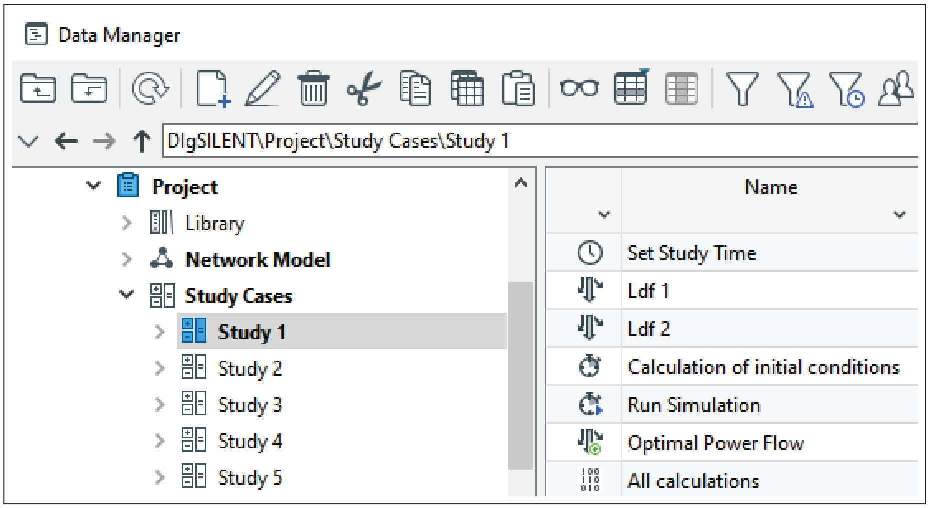


Figure 13.8.1: Calculation commands in a study case

Actions such as generating a report of the actual calculation results or the state of the defined network components are carried out via commands (in this case *ComSh* and *ComDocu*, respectively). For information about reporting commands refer to Chapter 19 (Reporting and Visualising Results).

Note: As with any other object, calculation commands can be copied, pasted, renamed and edited.

13.9 Events

Simulation Event objects are used to define simulation events. For time-domain simulations, events are stored within the *Study Case* → *Simulation Events/Fault* folder (see Chapter 29: RMS/EMT Simulations, Section 29.5 for a general description). For short-circuit studies, they are stored in the *Study Case* → *Short Circuits* folder. For other steady-state calculations that utilise *Simulation Events*, they are stored within the *Operational Library* → *Faults* folder. *PowerFactory* offers several kinds of events:

- Broken Conductor Event (*EvtBrkcond*)
- Dispatch Event (*EvtGen*)
- External Measurement Event (*EvtExtmea*)
- Intercircuit Fault Events (*EvtShcll*)
- Events of Loads (*EvtLod*)
- Message Event (*EvtMessage*)
- Outage of Element (*EvtOutage*)
- Parameter Events (*EvtParam*)
- Save Results (*EvtTrigger*)
- Save Snapshot (*EvtSave*)
- Short-Circuit Events (*EvtShc*)
- Stop Events (*EvtStop*)
- Switch Events (*EvtSwitch*)

- Synchronous Machine Event (*EvtSym*)
- Tap Event (*EvtTap*)
- Power Transfer Event (*EvtTransfer*)

13.9.1 Broken Conductor Event

This event type can be used to simulate a broken conductor in a line element. The user can specify which phase is broken and whether specific short-circuit conditions apply on either or both sides of the break. The broken conductor event can be used in the complete short-circuit calculation (if the multiple faults option is enabled in the short-circuit command) and in dynamic simulations.

- **Fault location**: defines at which point of the line the event is applied (seen from terminal i).
- **Conductor interruption**: specifies the phase(s) to be interrupted. Either all phases can be selected or, if this option is disabled, individual phases can be checked/unchecked.
- **Fault at side 1**: check this option to apply a fault at side 1 (towards from terminal i) and configure it accordingly.
- **Fault at side 2**: check this option to apply a fault at side 2 (towards from terminal j) and configure it accordingly.

13.9.2 Dispatch Event

The user specifies the point in time in the simulation for the event to occur, and a generation element (*ElmSym*, *ElmXnet* or *ElmGenstat*) or Ward-Equivalent element (*ElmVac*).

There is the option to define either an incremental change or a move to a specified set-point.

13.9.3 External Measurement Event

External measurement events can be used to set and reset values and statuses of external measurements.

13.9.4 Inter-Circuit Fault Events

This type of event is similar to the short-circuit event described in Section 13.9.11 (Short-Circuit Events (*EvtShc*)). Two different elements and their respective phases are chosen, between which the fault occurs. As for the short-circuit event, four different elements can be chosen:

- Busbar (*StaBar*)
- Terminal (*ElmTerm*)
- Overhead line or cable (*ElmLne*)

13.9.5 Events of Loads

The user specifies the point in time in the simulation for the event to occur, and a load or set of load element(s) (*ElmLod*, *ElmLodlv*, *ElmLodmv* or *ElmLodlvp*). Optionally a set of loads *SetSelect* can be also selected. The value of the load (s) can then be altered using the load event. The power of the selected load(s) can be changed as follows:

- **Step** Changes the current value of the power (positive or negative) by the given value (in % of the operating power of the load) at the time of the event.
- **Ramp** Changes the current value of the power by the given value (in % of the operating power of the load), over the time specified by the *Ramp Duration* (in seconds). The load ramping starts at the time of the event.

13.9.6 Message Event

A message will be printed to the output window at the specified time in the simulation.

13.9.7 Outage of Element (*EvtOutage*)

The *Outage of Element* event can be used to take an element out of service at a specified point in time. It is intended for use in steady-state calculations (e.g. short-circuit calculations and reliability assessment) and dynamic simulation.

Out of service: check/uncheck this flag in order to enable/disable this event.

Execution Time: defines the simulation/calculation time when this event is to be executed.

Element: the target element of this event is to be assigned here.

Type of Outage Event:

- *Take element out of service* - the *Element* is set to out of service at the defined *Execution Time*;
- *Bring element back into service* - the *Element* is put back in service at the defined *Execution Time* (this option is not supported in dynamic simulation);
- *Reinsert all outaged elements* - all elements outaged by previously executed outage events are put back in service (this option is not supported in dynamic simulation);
- *Take element and its controls out of service* - the *Element* is set to out of service at the defined *Execution Time* along with all associated controls. Associated controls are all DSL elements (and only those ones) which are referenced into a slot of a composite model whose *Main slot* is occupied by the target *Element*.

Note: The event can be used to take elements out of service in time-domain simulations, however it is not possible to bring an outaged element back into service using this event during a transient simulation. This is only possible in steady-state calculations. The following message will be displayed if the user attempts to bring a previously-outaged element back into service using *Outage of Element*:

```
t=000:000 ms - Outage Event in Simulation not available.
Use Switch-Event instead!
```

13.9.8 Parameter Events

With this type of event, an input parameter of any element or DSL model can be set or changed. First, a time specifying when the event will occur is specified. An element or set of elements *SetSelect* must then be specified/selected using the *down-arrow* button. Choose *Select...* from the context-sensitive menu, and insert the name and the new value of the element parameter. In case a selection is used all elements within the selection have to share the same element parameter.

13.9.9 Save Results

This event is only used for *PowerFactory* Monitor applications. It cannot be used during time-domain simulations.

13.9.10 *Save Snapshot* event

This event is used in a time-domain simulation whenever the current simulation state (a simulation snapshot) is to be saved. Further information on how to configure, save and load a simulation snapshot can be found in Section 29.9.

Event options:

- *Out of service* - Check this flag in order to disable its execution. Un-check this flag in order to activate it.
- *Execution time* - define the Absolute/Relative execution time of this event. The Absolute time reference is 0 seconds. The relative time reference is the current simulation time (if the simulation is reset/initialised, then the relative time fields are not available).

13.9.11 Short-Circuit Events

This event applies a short-circuit on a busbar, terminal or specified point on a line. The fault type (three-phase, two-phase or single-phase fault) can be specified, as can the fault resistance and reactance and the phases which are affected. The duration of the fault cannot be defined. Instead, to clear the fault, another short-circuit event has to be defined, which will clear the fault at the same location.

13.9.12 Stop Events

Stops the simulation at the specified time within the simulation time frame.

13.9.13 Switch Events

To create a new switch event, press the icon on the main menu (if this icon is available), which will open a browser containing all defined simulation events. Click on the icon in the browser, which will show the Element Selection dialog (*IntNewobj* as shown in Figure 13.9.1). This dialog can be used to create a new switching event.

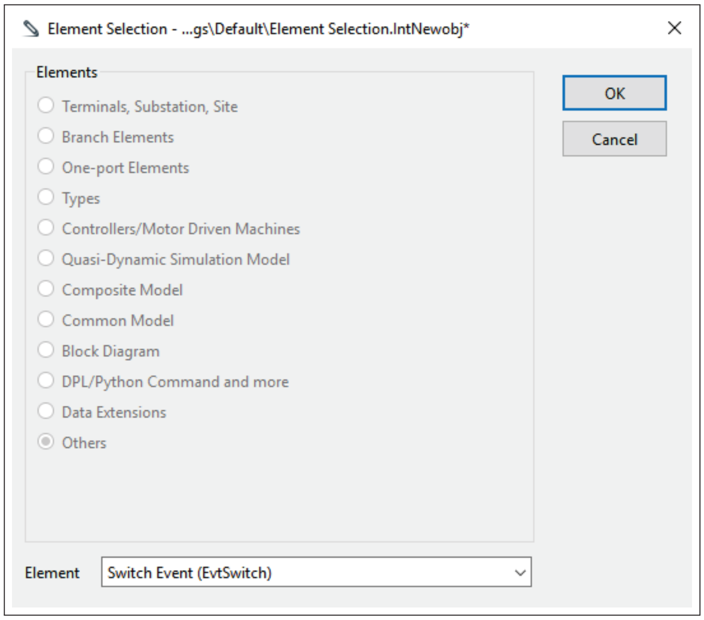


Figure 13.9.1: Creation of a new switch event (*IntNewobj*)

After pressing **OK**, the reference to the switch (labelled *Breaker* or *Element*) must be manually set. Any switch in the power system may be selected, thus enabling the switching of lines, generators, motors, loads, etc. The user is free to select the switches/breakers for all phases or only those for one or two phases.

It should be noted that more than one switching event must be created if, for instance, a line has to be opened at both ends. These switch events should then have the same execution times.

13.9.14 Synchronous Machine Event

The *Synchronous Machine Event* is used to change the mechanical torque of a synchronous machine (*ElmSym*) in a simple manner. The user specifies the point in time in the simulation for the event to occur, and an active synchronous machine. The user can then define the additional mechanical torque supplied to the generator. The torque can be positive or negative and is entered in per-unit values.

13.9.15 Tap Event

The user specifies the point in time in the simulation for the tap event to occur, and a shunt or transformer element (*ElmShnt*, *ElmTr2*, etc). The *Tap Action* can then be specified.

13.9.16 Power Transfer Event

The Power Transfer event (*EvtTransfer*) is used to transfer demand from a load object, or output from static generator, to other load objects or static generators respectively. The user specifies the source object (*ElmLod* or *ElmGenstat*) and the destination object or objects, which must be of the same class. Then separate percentage figures are input for active and reactive power. If more than one Power Transfer event is to be used, it is important to consider the order in which they will be executed, as this could affect the final outcome. Power Transfer events may be used in RMS simulations, Outage Planning and Faults Cases for contingency analysis.

13.10 Simulation Scan

For details of Simulation Scan modules, refer to Chapter 29: RMS/EMT Simulations, Section 29.8.

13.11 Results Objects

The results object (*ElmRes*) is used to store tables with the results obtained after the execution of a command in *PowerFactory*. Results Files are described in chapter ch:ReportingResults: Reporting and Visualising Results, section 19.6.

For Contingency Analysis, the results object can optionally contain a filter (*SetFilt*), to restrict the recording of results to a specified part of the network. The use of such a filter is described in the Contingency Analysis chapter, in section 27.10.1.

13.12 Triggers

As described in Chapter 18 (Parameter Characteristics, Load States, and Tariffs), parameter characteristics are used to define parameters as ranges of values instead of fixed amounts. The parameter characteristics are set over user defined scales. The current value of the parameter is at the end determined by a trigger object (*SetTrigger*,), which sets a current value on the corresponding scale. For example if the value of a certain parameter depends on the temperature, a characteristic over a temperature scale is set. The current value of the temperature is defined by the trigger. The current value of the temperature determines the current value of the parameter, according to the defined characteristic.

Once a parameter characteristic and its corresponding scale are set, a trigger pointing to the scale is automatically created in the active study case. The user can access the trigger and change its value as required.

PowerFactory offers different types of characteristics and scales, and each scale points to a trigger from the active study case. By default, scales are stored in the Scales folder within the *Characteristics* folder in the *Operational Library*. Information regarding the use and definition of characteristics, scales and triggers is given in Chapter 18 (Parameter Characteristics, Load States, and Tariffs).

13.13 Graphics Board

The study case folder contains a folder called the Graphics Board folder (*SetDesktop*,). This folder contains references to the graphics which are to be displayed. This folder, similar to the Summary Grid folder, is automatically created and maintained and should generally not be edited by the user.

The references in the graphics board folder are created when the user adds a grid to a study case. *PowerFactory* will ask the user which graphics pertaining to the grid should be displayed. At any time, the user may display other graphics in the grid by right-clicking the grid and selecting *Show Graphic*.

Graphics may be removed by right-clicking the tab at the top of the page and selecting *Close Page*.

The study case and graphics board folder also contain references to any other graphics that were created when the study case was active.

While working with multiple graphics boards, the currently active graphics board is ticked () in context menu of the study case as shown in the Figure 13.13.1). The user can switch between the existing graphics boards by selecting them.

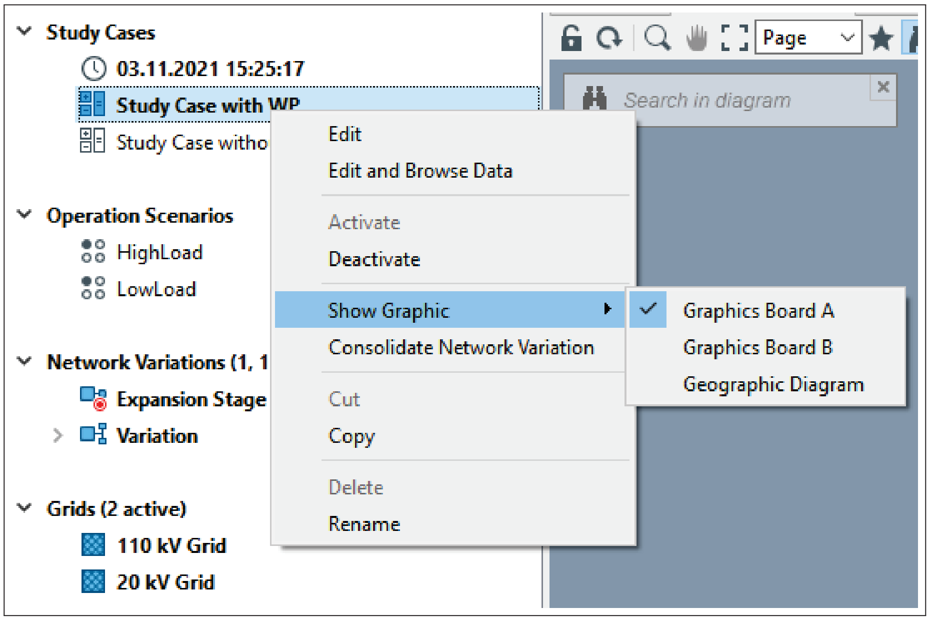


Figure 13.13.1: Multiple graphics boards in a study case

Chapter 14

Project Library

14.1 Introduction

The project library stores the following categories of data:

- **Equipment Types** (Section 14.2: Equipment Type Library)
- **Operational Data** (Section 14.3: Operational Library)
- **Scripts** (See Chapter 23: Scripting)
- **Table Reports** (See section Table Report Methods of the DPL Reference)
- **Templates** (Section 14.4: Templates Library).
- **Dynamic Models** (See Section 30.1: System Modelling Approach)

A range of pre-defined types, models, templates, and scripts are available in the *DIgSILENT* Global Library, described in Section 4.5.1.

Unlike the global libraries (located directly in the Database folder), which are accessible to all users, the local project library is used to store data that are to be used in the specific project. It can only be used by the project owner, and users with which the project is shared.

This chapter describes the *Equipment Type Library, Operational Library,* and *Templates* library.

14.2 Equipment Type Library

The *Equipment Type Library* is used to store and organise type data for each class of network component. Once a new project is created, an *Equipment Type Library* is automatically set by the program within the Library folder.

To create or edit a folder in the *Equipment Type Library*:

1. Right-click on the *Equipment Type Library* folder in the left pane of the *Data Manager* and select *New* → *Project Folder* from the context sensitive menu (or to edit an existing folder, right-click the folder and select *Edit*). The project folder edit dialog is displayed.
2. In the *Name* field, enter the name of the new folder.
3. In the *Folder Type* field, select *Generic*.

4. In the *Class Filter* field, write the name of the type class(es) to be allowed in the folder (case sensitive). If more than one class is to be allowed, write the class names separated by commas. An asterisk character (*) can be used to allow all classes.

5. In the *Icon* field, select *Library*.

To create new type objects in these folders select the *New Object* icon and select the appropriate type class. Alternatively, types can be copied from other projects or the *DIgSILENT* library. If the type class does not match the folder filter, an error message is displayed.

Note: By default new *DSL Model Types* (used by dynamic models) are also stored in the Equipment Types Library. Chapter 30 (Models for Dynamic Simulations) provides details related to dynamic modelling and *DSL Model Types*.

Figure 14.2.1 shows the equipment library of a project containing generator, load, and transformer types, sorted using library sub-folders.

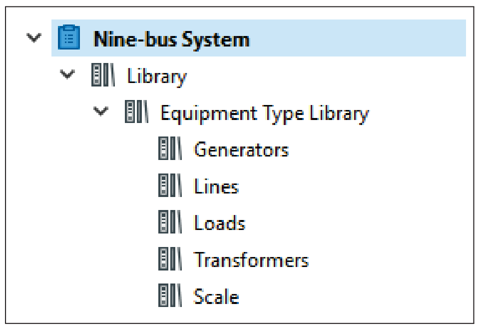


Figure 14.2.1: The Equipment Library

There are two options available for defining the type data for network components:

1. *Select Type*: the *Data Manager* pointing to the *DIgSILENT* global library, shown in Figure 14.2.2, is launched. The **Project Library** button can be used to quickly switch between the global and local libraries.

Note: If an additional library is defined in the global area (see Section 4.5.2) and set in the User Settings (see Section 7.1), an additional button will be available in the dialog

---

2. *New Project Type*: a new type will be defined and automatically stored in the local *Equipment Type Library*.

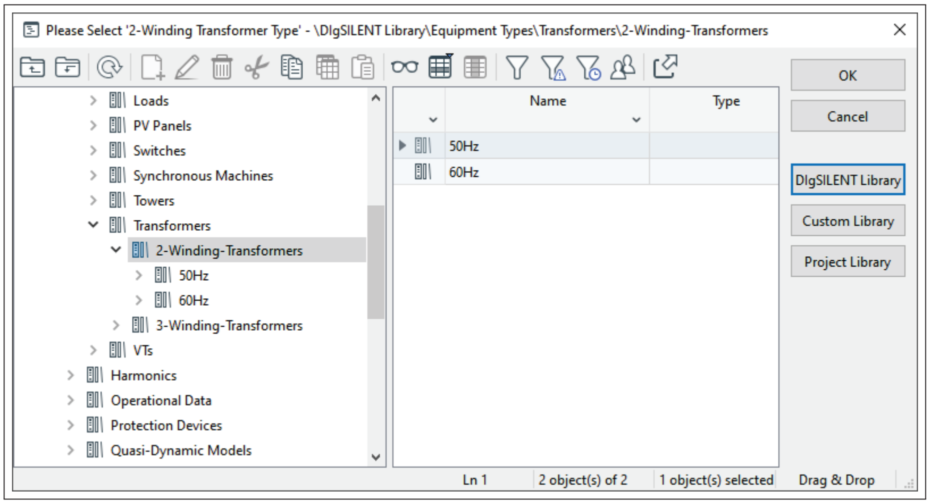


Figure 14.2.2: Libraries buttons shown when assigning a type

14.3 Operational Library

The *Operational Library* is used to store and organise operational data for application to a number of elements, without the need to duplicate operational information.

To illustrate, consider an example where there are two generators, “G1” and “G2”. The units have slightly different Type data, and thus unique Type models, “G 190M-18kV Ver-1” and “G 190M-18kV Ver-2”. The Capability Curves for these units are identical, and so the user wishes to create only a single instance of the capability curve. By defining a *Capability Curve* in the *Operational Library*, a single Capability Curve can be linked to both generators.

Similarly, various circuit breakers may refer to the same short-circuit current ratings. A *Circuit Breaker Rating* object can be defined in the *Operational Library* and linked to relevant circuit breakers

Within the *Characteristics* folder in the *Operational Library*, the *Scale* folder is used to store the scales used by the parameter characteristics. Refer to Chapter 18 (Parameter Characteristics, Load States, and Tariffs) for details.

This section describes the definition and application of operational data objects.

14.3.1 Circuit Breaker Ratings

Circuit Breaker Ratings objects (*IntCbrating*) contain information that define the rated short-circuit currents of circuit breakers (objects of class *ElmCoup*). They are stored inside the *CB-Rating* folder in the *Operational Library*. Any circuit breaker (*ElmCoup*) defined in the Network Model can use a reference to a Circuit Breaker Rating object in order to change its current ratings.

The parameters defined by a circuit breaker rating are:

- Three phase initial peak short circuit current.
- Single phase initial peak short circuit current.
- Three phase peak break short circuit current.
- Single phase peak break short circuit current.

- Three phase RMS break short circuit current.
- Single phase RMS break short circuit current.
- DC time constant.

To create a new circuit breaker rating in the operational library:

- In the Data Manager open the *CB Ratings* folder.
- Click on the *New Object* icon .
- In the *Element Selection* dialog select Circuit Breaker Rating (*IntCbrating*) and press **Ok**.
- The new circuit breaker rating dialog will then be displayed. Set the corresponding parameters and press **Ok**.

To assign a circuit breaker rating to a circuit breaker (*ElmCoup* object) from the network model:

1. Go to the *Complete Short-Circuit* page of the element's dialog.
2. In the Ratings field click on the button to select the desired rating from the *CB Ratings* folder.

The parameters defined in the circuit breaker ratings can be made to be time-dependant by means of variations and expansion stages stored inside the *CB Ratings* folder.

For information regarding short-circuit calculations, refer to Chapter 26 (Short-Circuit Analysis). For further information about variations and expansion stages, refer to Chapter 17(Network Variations and Expansion Stages).

Note: Variations in the CB Ratings folder act 'locally', they will only affect the circuit breaker ratings stored within the folder. Similarly, the variations of the Network Model will only affect the network components from the grids.

Note: Circuit breaker elements (*ElmCoup*) must be distinguished from Switch objects (*StaSwitch*); the latter are automatically created inside cubicles when connecting an edge element (which differs to a circuit breaker) to a terminal. Ratings can also be entered in the *StaSwitch* Type object.

Example Time-Dependent Circuit Breaker Rating

Consider an example where a substation circuit breaker "CB" operates with different ratings depending on the time of the year. From 1st January to 1st June it operates according to the ratings defined in a set of parameters called "CBR1". From 1st June to 31st December it operates with the ratings defined in a set of parameters called "CBR2".

This operational procedure can be modelled by defining a circuit breaker rating "CBR" in the *CB Ratings* folder, and a variation "CB_Sem_Ratings" containing two expansion stages. The first expansion stage should activate on the 1st January and the second on the 1st June. The first task is the definition of the time-dependant circuit breaker rating "CBR". To set the parameters of "CBR" for the first period:

1. Set a study time before the 1st June to activate the first expansion stage (the Variation "CB_Sem_-Ratings" must be active).
2. Edit the parameters of "CBR" (previously defined) according to the values defined in "CBR1". The new parameters will be stored in the active expansion stage.
3. To set the parameters of "CBR" for the second period:
4. Set a study time after the 1st June to activate the second expansion stage;
5. Edit "CBR" according to the values of "CBR2". The new parameters will be stored in the active expansion stage.

Once the ratings for the two expansion stages have been set, and the circuit breaker rating "CBR" has been assigned to the circuit breaker "CB", the study time can be changed from one period to the other to apply the relevant ratings for "CB" (note that the variation must be active).

14.3.2 Characteristics

Characteristics can be assigned to many element parameters. The use of Characteristics is described in detail in Chapter 18. Such characteristics are normally held in this folder of the Operational Library.

14.3.3 Demand Transfers

Note that Demand Transfers make use of the *IntOutage* object, which has now been superseded by the new *IntPlannedout* object described in section 14.3.7. Therefore, users wishing to create *IntOutage* objects will need to enable a project setting: on the Project Settings, Miscellaneous page select *Create IntOutage (obsolete)*.

The active and reactive power demand defined for loads and feeders in the network model can be transferred to another load (or feeder) within the same system by means of a *Demand Transfer* (objects class *IntOutage*). This transfer only takes place if it is applied during a validity period defined by the user (i.e. if the current study time lies within the validity period).

To create a new load demand transfer:

1. In the Data Manager, open the *Demand Transfer* folder.
2. Click on the *New Object* icon.
3. In the Element Selection dialog select *Planned Outage* (*IntOutage*) and press **Ok**.
4. Set the validity time, the source and target loads/feeders and the power transfer.

Note: If there is a demand transfer, which transfers load between two loads (*ElmLod*) belonging to different feeders (*ElmFeeder*), then the same MW and Mvar value is transferred from one feeder to the other.

A demand transfer is only possible if an active operation scenario (to record the changes) is available. The **Apply all** button will automatically apply all transfers that are stored in the current folder and which fit into the current study time. Before execution, the user is asked if the current network state should be saved in a new operation scenario. The same demand transfers can be applied as many times as desired during the validity period.

If a non-zero power transfer has been executed and the source's power is less than zero, a warning is printed to the output window indicating that the power limit has been exceeded. The applied transfers can be reverted by using the **Reset all** button.

When the current operation scenario is deactivated, all load transfers executed while the operation scenario was active will be reverted.

For information about operation scenarios refer to Chapter 16 (Operation Scenarios).

14.3.4 Fault Cases and Fault Groups

This section discusses the data structure of the Faults folder, and the objects contained within it. The functionality of Event objects is described in Section 29.5: Events (*IntEvt*).

The *Faults* folder stores two types of subfolders:

1. *Fault Cases* folders, which in turn store objects that represent *Simulation Events*. *Simulation Events* may contain a number of individual *Events* (*Evt**), e.g. short-circuits events, switching events.

2. *Fault Groups* folders store *Fault Groups* (IntFaultgrp) objects, which in turn reference *Fault Cases* (*Simulation Events* or individual *Events*).

Note: The use of *IntEvt* objects extends beyond *PowerFactory*'s reliability analysis functions. Time domain simulations (EMT/RMS) make reference to *IntEvt* objects, in order to include simulation events which take place during a time-domain simulation. In this case the execution time sequence of the events must be defined. In the case of fault representations in the Operational Library by means of fault cases, only short-circuit and switching events are relevant.

Note that the calculation commands provided by the reliability assessment function of *PowerFactory* use Contingencies objects (*ComContingency* and *ComOutage*) to simulate the outage (and subsequent recovery) of one or more system elements. To avoid duplication of data, these objects can refer to previously defined *Simulation Events* (*IntEvt*). For information regarding the functionality of fault cases and fault groups in contingency analysis tools refer to Chapter 27 (Contingency Analysis). For the use of fault cases to create outages for the contingency analysis tools refer to Chapter 45 (Reliability Assessment).

The following sections provide a details of how to define *Fault Cases* and *Fault Groups*.

Fault Cases

A fault case can represent a fault in more than one component, with more than one event defined. There are two types of Fault Cases:

1. **Fault cases without switch events (Type 1)**: Independent of the current topology and only stores the fault locations. The corresponding switch events are automatically generated by the contingency analysis tools. For further information refer to Chapter 45 (Reliability Assessment).

2. **Fault Case with at least one switch event (Type 2)**: A Fault Case of Type 2 predefines the switch events that will be used to clear the fault. No automatic generation of switch events will take place. For further information refer to Chapter 45 (Reliability Assessment).

To create new *Fault Cases* or new *Fault Groups* folders, open the *Faults* project folder from the *Operational Library* and use the *New Object* icon (select *Fault Cases*(*IntFltcases*) or *Fault Groups* (*IntFltgroups*) respectively).

To create new fault case (object of class *IntEvt*):

1. Multi-select the target components on a single line diagram.

2. Right-click and select *Define* → *Fault Cases* from the context-sensitive menu.

3. Select from the following options:
   - ***Single Fault Case***: This creates a single simultaneous fault case including all selected elements. A dialog box containing the created fault case is opened to allow the user to specify a name for the fault case. Press **Ok** to close the dialog and saves the new fault case.
   - ***Multi fault Cases, n-1***: This creates an n-1 fault case for each selected component. Therefore the number of fault cases created is equal to the number of components selected. This menu entry is only active if more than one component is selected. The fault case is automatically created in the database after selection.
   - ***Multi fault Cases, n-2***: This creates an n-2 fault case for each unique pair among the selected components. Therefore the number of fault cases is $(b \cdot (b-1)/2)$ where "b" is equal to the number of selected components. This menu entry is only active if more than one

component is selected. If only one component is selected, then no fault case will be created. The fault case is automatically created in the database after selection.

- ***Mutually Coupled Lines/Cables, n-k***: This creates fault cases considering the simultaneous outage of each coupled line in the selection.

The fault cases created will consist of short-circuit events applied to the selected components. All breakers (except for circuit breakers, which are used to model a circuit breaker failure) will be ignored.

- If only breakers are included in the selection, an error message will be issued.
- If a simple switch (not a circuit breaker) is included in the selection, a warning message will be issued that this switch will be ignored.
- If a circuit breaker is contained in the selection, then an *Info* message will be issued, that the CB will be used for modelling a CB failure and will not be handled as a fault location.

Note: In the case that a branch is selected, the short-circuit event is generated for a (non-switch device with more than one connection) component of the branch. The component used in the event is: "Connection 1" if suitable, otherwise "Connection 2" if suitable, otherwise a suitable random component of the branch (line, transformer . . .).

Fault Groups

New *Fault Groups* are created in the Data Manager as follows:

1. Open the target *Fault Groups* folder and select the *New Object* icon.
2. In the edit dialog, specify the name of the *Fault Group*, and **Add Cases** (*IntEvt*) to the *Fault Group*.

14.3.5 Capability Curves (Mvar Limit Curves) for Generators

Reactive Power operating limits can be specified in *PowerFactory* through definition of *Capability Curves* (*IntQlim*). They are stored in *Operational Library*, within the *Mvar Limit Curves* folder. Synchronous generators (*ElmSym*) and static generators (*ElmGenstat*) defined in the Network Model can use a pointer to a *Capability Curve* object from the *Load Flow* page of their edit dialog. When executing a *Load Flow* (with *Consider Reactive Power Limits* selected on the *Basic Options* page) generator Reactive Power dispatch will be limited to within the extends of the defined capability curve. For information about the dispatch of synchronous generators, refer to the synchronous machine technical reference in the Technical References Document. For information about Load Flow calculations and reactive power limits, refer to Chapter 25 (Load Flow Analysis).

Note: If *Consider active power limits* is selected on the *Basic Options* page of the Load Flow Calculation command, active power is limited to the lesser of the Max. Operational Limit and the Max. Active Power Rating specified on the Synchronous Machine *Load Flow* page.

Defining Capability Curves

To define a new generator *Capability Curve*:

1. Open the folder *Mvar Limit Curves* from the *Operational Library*.
2. Click on the *New Object* icon and select *Capability Curve*. The new capability curve dialog will be displayed.
3. Enter data points to define the generation limits, and *Append Rows* to add the required number of rows to the table.

4. To apply a *Capability Curve* to a generator:
   - Locate the *Reactive Power Limit* section on the *Load Flow* page of the synchronous machine's or static generator's dialog.
   - Press next to *Capability Curve*.
   - Choose *Select* and then select the required curve in the *Mvar Limit Curves* folder of the *Operational Library* (the required curve can also be created at this step by selecting the *New Object* icon .
5. Select a capability curve and press **OK**.

Capability curves are included in operation scenario subsets; meaning that if a capability curve is selected/reset from a generator when an operation scenario is active, the change will be stored in the operation scenario. Once the operation scenario is deactivated, the assignment/reset of the curve is reverted. For information on working with operation scenarios, refer to Chapter 16 (Operation Scenarios).

To enter a capability curve for information purposes only (i.e. a capability curve which is not to be considered by the calculation), enter it on the Advanced tab of the *Load Flow* page. Then select *User defined Capability Curve* and enter the curve as a series of points in the table. Right-click on the rows to append, delete or insert new rows.

Defining a Variation of a Capability Curve

Similar to circuit breaker ratings (see Section 14.3.1 (Circuit Breaker Ratings), *Capability Curves* can become time-dependant by means of variations and expansion stages stored inside the *Mvar Limit Curves* folder.

To create a time-dependent variation for a *Capability Curve*, navigate to the*Mvar Limit Curves* folder in the left pane of a Data Manager window. Right-click on the folder and select *New* → *Variation*. Name the variation, press **OK**, name the Expansion Stage, and press **OK**. Changes to Capability Curves are recorded in the active expansion stage.

To activate a variation of a *Capability Curve*, open the Data Manager. Right-click the *Variation object* in the *Mvar Limit Curves* folder and select *Activate*.

For general information about variations and expansion stages refer to Chapter 17 (Network Variations and Expansion Stages).

14.3.6 Planned Outages

Planned Outage objects (*IntPlannedout*) are normally stored in the Outage folder of the Operational Library. They can be applied to an active study case to model expected outages of network elements for maintenance, network expansion etc. Figure 14.3.1 shows the dialog box of a Planned Outage object, illustrating the following features:

- Start and End date of the period for which the Planned Outage is valid.
- Outaged components.
- Buttons to apply and reset the outage, view the events and record additional events.

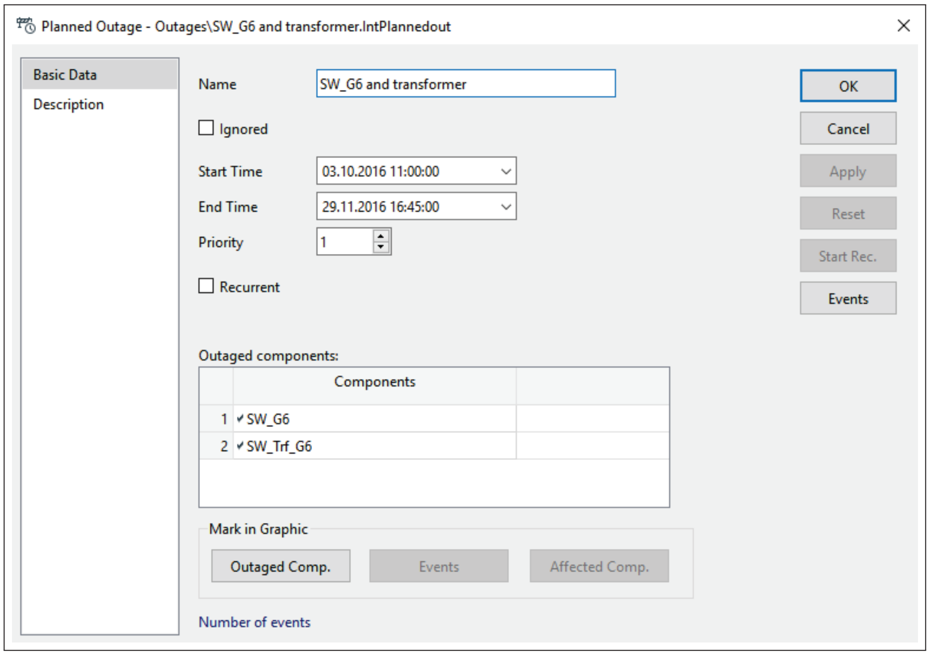


Figure 14.3.1: Planned Outage object dialog

Changes to switch positions and other parameters resulting from the application of Planned Outages will be taken into account for all calculations but are only effective as long as the study case is active. A new toolbar has been provided for the handling of Planned Outages. Please see chapter 42 for how to create Planned Outage objects and handle them via the toolbar.

14.3.7 Planned Outages *IntOutage*

Note that this subsection refers to the original *IntOutage* object, which is now superseded by the *IntPlannedout* object described in section 14.3.6. Users wishing to create *IntOutage* objects will need to enable a project setting: on the Project Settings, Miscellaneous page select *Create IntOutage (obsolete)*.

A *Planned Outage* is an object used to check and/or apply an *Outage of Element* or *Generator Derating* over a specified time period. *Planned Outages* are stored within the *Operational Library* in the *Outages* folder.

- For the *Outage of Element type*, *PowerFactory* automatically isolates the referenced components. The switches connecting the target elements with the other network components are open and the terminals connected to the elements are earthed (if the *Earthed* option in the terminal (*ElmTerm*) dialog is checked). Note that the target element can only be earthed if it is directly connected (without switches in the cubicle) to terminals, which are then connected through switches to the network terminals.
- For a *Generator Derating*, a reference to the generator which is to be derated and the magnitude of the *MW reductions* is specified. For the *Generator Derating*, the maximum active power that can be dispatched (defined on the *Load Flow* page of the generator element dialog, in the section *Operational Limits*) is recalculated as the difference between the maximum active power (section *Active Power: Ratings*) and the *MW reductions*.

Note: If a Planned Outage object is defined in the Outages folder of the Operational Library, only the outage types Outage of Element and Generator Derating are enabled. Similarly if outage objects are defined in the Demand transfer folder, only the outage type Demand Transfer is enabled.

Defining Outages and Deratings

To create a new *Element Outage* or *Generator Derating*:

1. In the Data Manager, open the *Outages* folder.
2. Click on the *New Object* icon, select *Planned Outage* and press **Ok**.
3. The *Planned Outage* dialog will be displayed. In the *Outage Type* frame of the dialog, the options *Outage of an Element* and *Generator Derating* will be enabled. Set the desired *Outage Type, Start Time* and *End Time*.
4. The definition of a *Planned Outage* requires reference(s) to relevant network components. To create a reference:
   - Press the **Contents** button of the outage object.
   - In the data browser that is displayed, create a reference to the target element by selecting the *New Object* icon (*IntRef*).
   - Press the button in the *Reference* field to select the target element.
   - Press **Ok** to add the reference.
5. (*Generator Derating* only) Specify the *MW Reduction* (see previous section for details) for the generator derating.
6. To apply the *Planned Outage*, press the **Apply** button (the **Apply** button is only available if the study time lies within the outage period, and an *Operation Scenario* is active).

Applied outages and generator deratings can be reset using the **Reset** button.

Checking Outages and Deratings

The **Check All** button in the *Planned Outage* dialog is used to verify if the actions defined for the target element(s) have been performed (right-click a *Planned Outage* and select *Check* to perform an individual check). Only the outages within a valid period are considered. Outages marked as *Out of Service* are not regarded (even if the study time lies within the outage period).

For an *Outage of Element*, the energising state is always determined by a connectivity analysis. Any component that is connected to an External Grid or a reference Generator is considered to be energised. All other components are considered to be de-energised (if circuit breakers are open). A de-energised component is earthed if a topological connection to a grounding switch or an earthed terminal exists (terminal with the *Earthed* option checked).

Note: If the outaged element is a Branch Element (*ElmBranch*), all contained elements are checked. If any of these elements is not correctly outaged, the whole branch is reported as not correctly outaged.

The fulfilment of programmed outages can also be checked via the use of the colour representation function available within the single line graphic by setting the *Colouring* option to *Outage Check* from the colour representation dialog. The following states are coloured, according to user preferences:

- Components that are energised, but should be outaged.
- Components that are de-energised and not earthed, but should be outaged.
- Components that are de-energised and earthed, but should NOT be outaged.
- Components that are de-energised, not earthed and should be outaged.
- Generators that are not derated, but should be outaged.
- Generators that are derated, but should NOT be outaged.

14.3.8 QP-Curves

Q(P)-curves (*IntQpcurve*) are used by generators (*ElmSym*, *ElmGenstat* and *ElmAsm*) when the Controller mode Q(P)-Characteristic is selected. In this mode, the reactive power control follows a user-specified characteristic, so that the reactive power setpoint is adapted according to the active power output of the machine.

14.3.9 Remedial Action Schemes (RAS)

Remedial Action Schemes (*IntRas*) and Remedial Action Scheme groups may be used during contingency analysis. If the user creates Remedial Action Schemes and groups, they will be stored here in the Operational Library. See chapter 27, Section 27.11 for more information.

14.3.10 Running Arrangements

Running Arrangement objects store operational data (switch status) for a single substation. As shown in Figure 14.3.2, a *Running Arrangement* uses a reference to the substation object (*ElmSubstat*) whose switch statuses are stored. *A Start Time* and *End Time* is used to specify the validity period of the *Running Arrangement*. Running arrangements are stored in the *Running Arrangements* folder in the Operational Library.

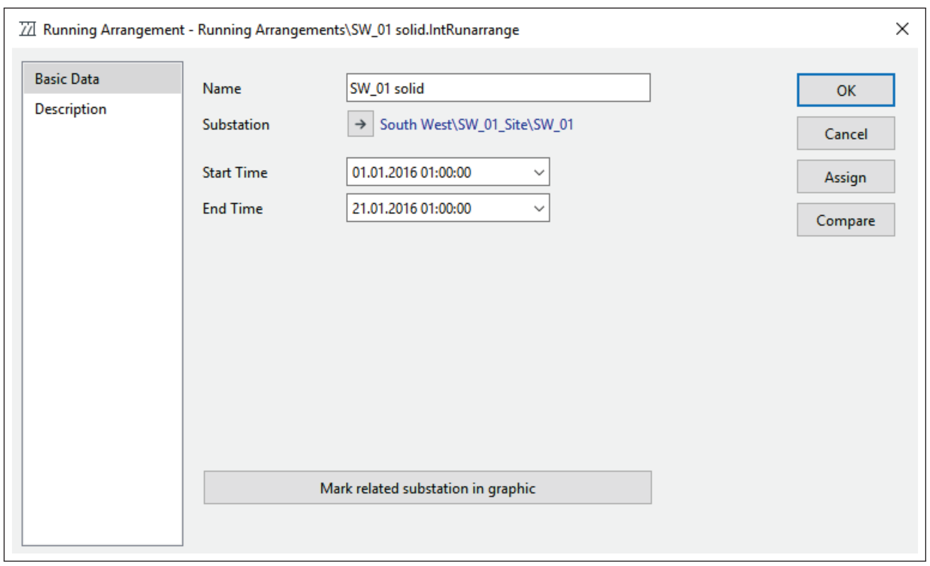


Figure 14.3.2: RA object dialog

Different configurations of the same substation can be defined by storing the corresponding switch statuses in *Running Arrangements*. Different *Running Arrangements* can then be easily selected during a study. If a running arrangement is selected for a substation, the status of the substation switches cannot be modified (i.e. they become read-only). If there is no setting for a switch in a *Running Arrangement* (i.e. the *Running Arrangement* is incomplete), the switch will remain unchanged but its status will also be set to read-only. If the current *Running Arrangement* is deselected, switch status will be reverted to the status prior to application of the *Running Arrangement*, and write-access will be re-enabled. Running arrangements are defined and selected in the substation object dialog *Basic Data* page.

Note: Running arrangements store only the status of switches of class *ElmCoup*. The status of switches which are automatically created in a cubicle following the connection of a edge element (*StaSwitch* objects) are not considered in a running arrangement.

Further details of how to create, select, apply, and assign Running Arrangements are provided in the following sections.

Creating a Running Arrangement

To store the current status of the switches in a substation, a *Running Arrangement* object must be created. To create and save a new *Running Arrangement* (RA):

1. Click on an empty place in the substation graphic, and from the context-sensitive menu choose *Edit Substation*. Open the substation dialog.
2. Click **Save as** to store the switch settings of the substation as a new RA. This button is only available if there is currently no RA selection active.
3. In the new RA dialog is displayed, specify a name and time period, and press **Ok**. The new RA is automatically stored in the *Running Arrangements* folder in the *Operational Library*.

An **Overwrite** button is available in the substation dialog (if no RA is selected), to store current switch statuses to an existing RA.

Selecting a Running Arrangement

A Running Arrangement (RA) can be selected in the Basic Data page of a substation dialog:

1. Open the substation dialog.
2. In the *Running Arrangement* frame of the Substation dialog, select ⌄ from a list of previously defined RA's.
3. Select the desired RA. This selection is immediately reflected in the substation graphic.

While an RA is selected, the switch statuses of a substation are determined by this RA and cannot be changed by the user (i.e. they are read-only).

If there is no setting for a switch in an RA (i.e. the RA is incomplete), such a switch will remain unchanged but its status is also set to read-only.

Furthermore, there is a button **Select by Study Time** (also available via the context-sensitive menu when right-clicking on the Data Manager), which selects a valid RA automatically according to the study time. If there are multiple RAs valid for the current study time, or if there is no valid one, a warning is printed to *PowerFactory*'s output window (nothing is selected in this case).

Applying and Resetting a Running Arrangement

An active *Running Arrangement* (RA) can be applied to a substation by pressing the **Apply and Reset** button from within the substation dialog. This action copies the statuses stored in the RA directly in the substation switches. It is only available only if an RA is selected. The RA will be deselected afterwards. An RA can be directly set as the substation's selected RA, using the **Assign** button (from within the RA dialog).

The following functional aspects must be regarded when working with running arrangements:

- An RA can be selected for each substation. If an operation scenario is active, the selection of an RA in a substation is recorded in the operation scenario (i.e. the RA selection is part of the operational data included in the operation scenario subset).
- If a variation is active (and there is no active operation scenario), the selection of the RA is stored in the recording expansion stage.

- While an RA is selected, the switch statuses of the corresponding substation are determined by the RA and can not be modified. Any attempt to change such a switch status will be rejected and a warning message will be printed to the output window. The switch statuses preceding the activation of an RA remain unchanged and are restored when deselecting the RA.
- The switch statuses stored in the RA could be incomplete due to the activation of a variation or a modification made to the network model. For example, if an RA was defined and then deactivated, and then later new switches were added to a substation. In this case if the RA is re-activated, a warning would be printed to the output window and the current switch statuses, which depend on the base network, active variations and active operation scenario, remain unchanged. Missing switch statuses will be added only when performing the **Save as** or **Overwrite** functions (available in the substation dialog).
- Switch statuses stored in the RA, and which are currently not required (depending on expansion stages) are ignored and remain unchanged. In this case a summary warning is printed during the RA activation.
- It is not possible to add a new switch to a substation while a running arrangement is selected. Additionally, it is not possible to delete an existing switch from this substation. In both cases the action is blocked and an error message is issued.

For information regarding operation scenarios and their application refer to Chapter 16 (Operation Scenarios).

Assigning a Running Arrangement

The **Assign** button contained in the *Running Arrangement* (RA) dialog facilitates the selection of the RA as the one currently selected for the corresponding substation. This action is also available in the context-sensitive menu in the Data Manager (when right-clicking on an RA inside the Data Manager). It should be noted that assignment is executed immediately and cannot be undone by pressing the cancel button of the dialog.

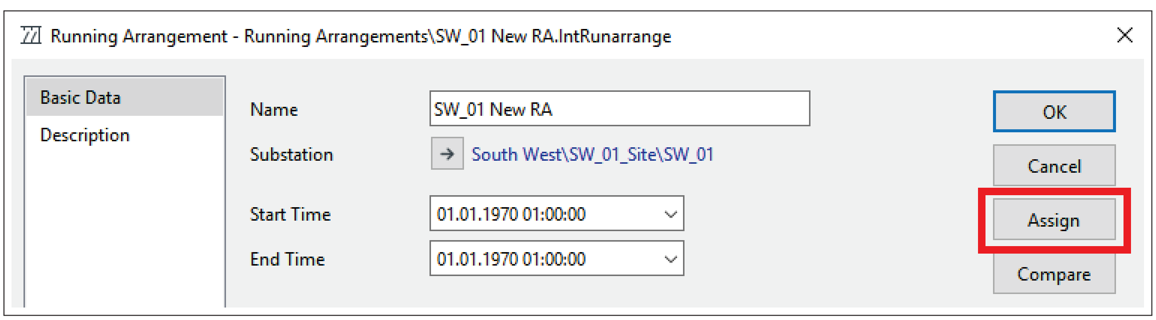


Figure 14.3.3: Running Arrangement Dialog

Marking Running Arrangements in Graphic

A **Mark related substation in graphic** button is provided on the Running Arrangement object. This can be used to display the related substation diagram or find the related substation in an overview graphic.

It is also possible to do this using the *Mark in Graphic* option in the context-sensitive menu displayed when right-clicking on a Running Arrangement in a Data Manager.

14.3.11 Thermal Ratings

Thermal Ratings objects (*IntThrating*) allow the definition of post-fault operational ratings for certain branch elements, depending on the fault duration and the loading prior to the fault. *Thermal Ratings* objects are stored in the *Thermal Ratings* folder in the *Operational Library*. They are two-dimensional matrices, with cells that contain the "short time" post-fault ratings (in MVA), according to the pre-fault loading (defined in the first column) and the duration of the fault/overloading (defined in the first row).

References to *Thermal Ratings* are defined on the *Basic Data* page of the dialog of the target branch elements. Elements that can use references to *Thermal Ratings* are:

- Transmission lines (*ElmLne*).
- 2- and 3-winding transformers (*ElmTr2*) and (*ElmTr3*).
- Series reactors (*ElmSind*).
- Series capacitors (*ElmScap*).

Note that the rating table given on the Ratings page of the *Thermal Rating* object (when option *Consider short term ratings* is enabled) is used solely for the contingency analysis command in *PowerFactory*. In this calculation, the pre-fault loading conditions of the network components are determined after a base load flow calculation. The contingency analysis is then performed using a load flow command, where the post-contingency duration is specified.

To create a new *Thermal Ratings* object:

1. Open the folder *Thermal Ratings* from the Operational Library.
2. Click on the *New Object* icon and select *Thermal Ratings.*
3. The new object dialog is displayed. To configure the table for the short-term ratings (only visible if the option *Consider short term ratings* is checked), go to the *Configuration* page and:
   - Introduce the increasing values for the pre-fault loading axis (Prefault %). By default, values between 0% and 80%, with increments of 5%, up to 84% are set.
   - Introduce the fault duration in minutes. Default values are: 360 min, 20 min, 10 min, 5 min, 3 min).

The pre-fault continuous rating (used as the base to calculate the loading before the fault) and the post-fault continuous rating (assumed as the branch element post-fault rating if the fault duration is larger than the largest duration time defined in the table) are defined on the *Ratings* page.

The values of a thermal rating object can be edited at any time by double-clicking on it to open the *Thermal Ratings* dialog. Similar to *Circuit Breaker Ratings* and *Capability Curves*, *Thermal Ratings* objects can be made to be time-dependant by means of variations and expansion stages stored inside the *Thermal Ratings* folder (refer to the *Circuit Breaker Ratings* section for an explanation on how to define time-dependant operational objects).

When a contingency analysis (*ComSimoutage*) is configured, the user can define a post-contingency time. According to the pre-fault loading found by the load flow used to calculate the base case, and the post-contingency time (if specified), the ratings to be used in the contingency load flow are determined (based on the referred *Thermal Ratings* object). The loading of the branch elements after the contingency load flow are calculated with respect to the new ratings.

For information about contingency analysis refer to Chapter 27 (Contingency Analysis).

14.3.12 V-Control-Curves

The curves in this folder, i.e. V-Control-Curve (*IntVctrlcurve*) are used by transformers, if the user wishes to define the voltage setpoint according to the power or current flow through the transformer.

14.4 Templates Library

The *Templates* folder is used to store and organise templates of network components (or groups of components) for re-use in a power system model. Components from templates are created using the graphical editor. Five kinds of templates are supported in *PowerFactory*:

1. Element template for single network elements: New single network elements with the same parameters as the original element are created.
2. Group template for non-composite graphic objects: New groups of objects (including graphical attributes) are created.
3. Substation template (composite node): New substations with the same configuration as the original substation (including its diagram).
4. Secondary Substation template: New secondary substations.
5. Branch template (composite branch): New branches with the same configuration as the original branch (including its diagram).

Templates are normally stored in the *Templates* folder, in the *Library*. When a template for a single network element is defined, a copy of the original element is automatically created in the *Templates* folder. New templates of substations and branches will copy the objects together with all of their contents (including the diagram) to the *Templates* folder. New templates for groups of objects will copy the corresponding objects, together with their graphical information to a subfolder for groups of class *IntTemplate* within the *Templates* Library.

For further information about working with templates, refer to Section 11.2 (Defining Network Models with the Graphical Editor).

Substation (*composite node*) templates (or), secondary substation ( ), busbar templates ( ), branch templates ( ), and general templates ( ) can be selected from the Drawing Toolbox on the right-hand pane of the *PowerFactory* GUI. To apply an element template:

- Select the symbol for a substation, secondary substation, busbar, branch, or general template as required.
- Select the required template.
- Insert the new element in the single line graphic.

Note: The use of Substation templates is recommended for diagrams of networks, where components are grouped in branches and substations. In this case the composite nodes can be graphically connected with the composite branch, forming an overview diagram of the complete network.

This section explains how to create and use templates.

14.4.1 General Templates

General Templates are used for grouping and packaging objects into a data container that can be stored within the project *Library* for future use. *General Templates* are intensively used when a fully packaged power equipment model is defined, with one such model containing various functional components, for example:

- Built-in models (e.g. static generators, PWM converters);
- User defined load flow / Quasi-dynamic simulation models (e.g. QDSL models);
- Measurement devices for dynamic simulation (e.g. voltage and current measurement objects);
- User-defined dynamic controller models (e.g. power and voltage controllers, fault-ride through controllers).

Furthermore, all external references of the contained objects can be additionally packed, thus making sure that all required elements are stored in the *General Template*. Consequently, *General Templates* enable *PowerFactory* users to group and pack all of the above objects into a single data container that can later on be easily and safely shared across *PowerFactory* project boundaries.

In order to create a *General template*, do the following:

- The source data for a *General Template* is an already operational and configured model. Therefore, once a power equipment model / *User-defined* model is configured and operational in a project, go to the single line diagram of the grid in which this model is existing.
- Select all graphically defined elements shown in the grid's single line diagram which belong to the model.
- Right-click on the selected elements and choose "Define Template". As a result, a copy of each selected element is created in a newly created *General Template object* (*IntTemplate*) located within the project library (*Library* → *Templates*).
- If the selected elements are part of a dynamic simulation model (i.e. they have a referenced *Composite Model* to parameter *c_pmod* (if existing), then a copy of the referenced *Composite Model* is added to the template. Furthermore, all contained objects within the selected elements and the referenced *Composite Model* are copied. Lastly, note that the copied elements maintain their references to other objects within the template (i.e. the references and connections to objects within the template are not lost).

Note: The template contains by default only the directly copied elements (as previously detailed) but without storing any possibly existing library type objects.

- If the library objects shall be included in the template (recommended action, in order to completely separate the contained model from other source project folders), then open the Edit dialog of the *General Template* and click the **Pack** button. A subfolder is created within the template containing all referenced library objects.
- Click on **Check** button to verify whether any contained objects have references to objects located outside the template.
- Click on **Contents** button to display the contents of the template. If needed, further elements/objects can be manually added to the template e.g. DPL scripts, further library objects etc.
- Once the template has been created, it can be used within the existing project or exported outside *PowerFactory* (as a .pfd file) or to other projects within the same *PowerFactory* database.

In order to export a *General Template* outside *PowerFactory*, do the following:

- Open the *Data Manager* and deactivate the current project (if any).
- Navigate and locate the *General template* of interest (by default, found in the *Library* → *Templates* folder of a *PowerFactory* project.
- Select the *General Template* and click the **Export Data** button. Alternatively, from the file menu click *File* → *Export*→ *Data*.
- Follow the normal export procedure. The template is now saved in a *.pfd* file which can be shared outside *PowerFactory*.
- The *.pfd* file can now be imported in other *PowerFactory* programs. When importing the template, it should always be imported into the *Library* → *Templates* subfolder of the destination project.

In order to copy a *General Template* from one project to another one located in the same *PowerFactory* database, do the following:

- Open the *Data Manager* and deactivate the current project (if any).

- Locate and copy the *General Template* of interest
- Activate the project in which the template is to be added.
- Navigate to the project's *Library* → *Templates* and paste the template there.

In order to make use of a *General template* (e.g. deploy the contained model into a network), do the following:

- Show the single line diagram of the Grid in which the model shall be deployed. Make sure that the diagram is un-frozen, such that the *Drawing Tools* toolbar is shown.
- Click on the *General templates*
- A selection window with all available project templates is shown (the "Project Templates" option on the left side pane of the window, corresponds to the *Library* → *Templates* project folder).
- Alternatively, it is possible to choose a template located in the Global Library (i.e. DIgSILENT *Library* → *Templates*) by selecting the "Library Templates" option on the left side pane of the window.
- Click once on the *General Template* of interest in order to select which template shall be deployed.
- Click once in the single line diagram in order to activate the *General Template* drawing mode. If done so, the cursor will change appearance by overlaying the template's graphic. The template is not deployed at this stage.
- The mouse cursor can now be moved to an area in the single line diagram of the grid in order to conveniently deploy the *General Template*.
- When appropriately positioned, click once to deploy the template into the grid at the current location, containing all template's elements (inlcuding the non-graphical ones).

14.4.2 Substation Templates

Existing substations can be used as "models" to define templates, which may be used later to create new substations. A new substation template is created by right clicking on one of the busbars of the detailed substation single line diagram and selecting *Define substation template* from the context sensitive menu. This action will copy the substation together with all of its contents (including its diagram even if it is not stored within this substation) in the Templates folder.

Note: In case of creating templates which contain graphical information the default settings of the names and result boxes defining their graphical representation (font, frame, size,...) are copied into the template diagram so that they appear as in the source object(s).

14.4.3 Busbar Templates

Similar to creating substation templates, existing busbars can be used as a "models" to create user-defined templates, which may be used later to create new busbars. A new busbar template is created by right clicking on the detailed single line diagram or simplified diagram of busbar and selecting 'Define substation template' from the context sensitive menu. This action will copy the busbar together with all of its contents (including detailed and simplified diagrams) in the Templates folder. If the detailed busbar configuration has been modified, it is possible to right-click the (existing) simplified representation in the main single line diagram and select 'Update representation'.

Busbars that have been created by the user in this way can be added to the single line diagram by selecting the 'General Busbar System' icon ( ). Note that for a busbar to be accessible from this icon, both detailed and simplified diagrams must be included within the busbar template, as in the previously described method.

14.4.4 Composite Branch Templates

Composite Branch templates can be defined as follows:

1. To create a Branch template, navigate to the *Library* → *Templates* folder in the Data Manager.
2. Right-click on the right pane of the Data Manager and select *New* → *Branch* from the context sensitive menu.
3. In the branch edit dialog, define the name of the branch and press **Ok**.
4. A new (empty) single line diagram will be displayed. Draw the required elements (for example, a terminal with two lines connected, with each line connected at one end only).
5. To create an instance of the Branch from the newly created Branch template, navigate back to the main grid diagram, then select the Composite Branch ( ) icon and connect the branch to existing terminals on the Single Line Diagram.

Alternatively, composite branches can be defined in the Data Manager as described in Chapter 10: Data Manager, Section 11.5.4 (Defining Composite Branches in the Data Manager).

14.4.5 Example Power Plant Template

Consider the following example, where there is a power station with multiple transformers, generators, and control systems of the same type. The model can be created using templates as follows:

1. Firstly, define type data for the transformer, generator, and control system.
2. Add a single instance of the generating unit (including generator transformer) to the network model.
3. Define a *Template* by selecting the generator, generator bus, and transformer, then right-click and select *Define Template*. Optionally include the control system model with the template.
4. To create another instance of the newly created template, select the *General Templates* icon ( ) and place it on the single line graphic.

14.4.6 Wind Turbine Templates according to IEC 61400-27-1

There are predefined Templates for Wind turbine models according to IEC 61400-27-1 in the Templates Library of *PowerFactory*. More information is available in section Templates of the Technical References Document.

Chapter 15

Grouping Objects

15.1 Introduction

This chapter describes the management and functionality of the objects used to group network components.

Among the various applications of grouping objects is their use for visualisation in network diagrams. For this reason, many of these grouping object classes include the possibility of associating colours with the objects. For general information about configuring colours, see Section 4.7.5.

15.2 Areas

To facilitate the visualisation and analysis of a power system, elements may be allocated into areas (*ElmArea*). The single line graphics can then be coloured according to these areas and special reports after load flow calculations ('Area summary report' and 'Area interchange report') can be generated. Area objects are stored inside the *Areas* folder ( ) in the *Network Data* directory.

To define a new area:

- Multi select the components belonging to the new area (in the Data Manager or in a single line diagram).
- Right click on the selection and select *Define* → *Area* from the context sensitive menu.
- After the area has been defined, terminals can be added to it by selecting *Add to...* → *Area...* in their context sensitive menu.

In the edit dialog of the new area you must select a colour to represent the area in the single line diagrams. Using the **Edit Elements** button you can have access to all the element belonging to that area in a data browser, then you can edit them. The **Mark in Graphic** button may be used to locate the components of an Area in a single line diagram.

Note: Areas that are created/deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time.

For information concerning the visualisation of areas within the single line Graphic refer to Chapter 9: Network Graphics, section 9.3.5.1 (Basic Attributes). For information about reporting Area results refer to Chapter 19 (Reporting and Visualising Results).

15.3 Virtual Power Plants

Virtual Power Plants are used to group generators in the network, in such a way that the total dispatched active power is set to a target value. The dispatch of each generator (the *Active Power* field available in the *Dispatch* section of the *Load Flow* page in the generator element dialog) is scaled according to the Virtual Power Plant rules (must run, merit of order, etc.), in order to achieve the total target value.

Virtual Power Plant objects (*ElmBmu*) are stored inside the *Virtual Power Plants* folder () within the *Network Data* directory.

15.3.1 Defining and Editing a New Virtual Power Plant

A new Virtual Power Plant is created by:

- Multi selecting in a single line diagram or in a data browser an initial set of generators to be included in the Virtual Power Plant;
- Then pressing the right mouse button and selecting *Define* → *Virtual Power Plant* from the context sensitive menu.

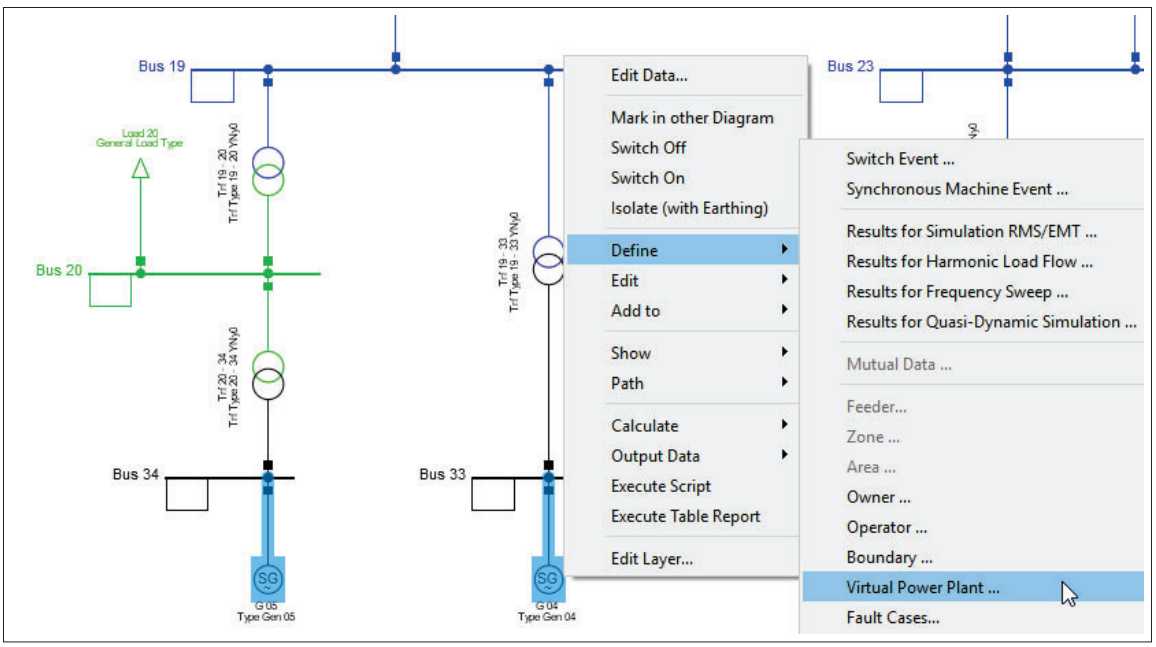


Figure 15.3.1: Defining a Virtual Power Plant

Alternatively you can create a new empty Virtual Power Plant by using the Data Manager:

- Open a Data Manager.
- Find the Virtual Power Plant folder () and click on it.
- Press the icon for defining new objects ().
- select *Others*.
- Then select *Virtual Power Plant* (*ElmBmu*) in the list box.
- Assign a suitable name to the Virtual Power Plant.

- Press **OK**.

The rules which determine the dispatch of the selected generators are set in the Virtual Power Plant dialog. The total active power to be dispatched is set in the field 'Active Power'. The dispatch of the belonging generators (variable "pgini" from the Load Flow tab of the generator) is set by pressing the **Apply** button. If the 'Maximal active power sum' of the included generators (sum of the maximal active power operational limit of the generators) is smaller than the active power to be dispatched, an error message pops up. Otherwise the dispatch is set according the user defined 'Distribution Mode':

According to merit order Distribution of the dispatched active power is done according to the priorities given to each generator in the Merit Order column of the 'Machines' table (this value can also be set in the Optimisation tab of the generators dialog). Lower values have higher priority. Generators with the option 'Must Run' checked are dispatched even if they have low priority (high value). It is assumed that the merit of order of all generators in the Virtual Power Plant is different. If not an error message appears after the 'Apply' button is pressed.

According to script The rules for the dispatch are set in user defined DPL scripts, which are stored inside Virtual Power Plant object. To create new scripts or to edit the existing ones you must open a data browser with the 'Scripts' button.

Note: The Virtual Power Plant active power is part of the operation scenario subsets and therefore is stored in the active operation scenario (if available). The active power is stored in the active expansion stage (if available) if no active operation scenario is active. Virtual Power Plants that are created/deleted when a recording expansion stage is active; become available/non available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time.

15.3.2 Applying a Virtual Power Plant

Check that the active power set for the Virtual Power Plant is less than or equal to the maximum power. Press the **Apply** button.

15.3.3 Inserting a Generator into a Virtual Power Plant and Defining its Virtual Power Plant Properties

Generators are added to an existing Virtual Power Plant by adding a reference in the 'Optimisation' tab of their edit dialog. Notice that a generator can belong to at most one Virtual Power Plant. Define the Merit Order and must run properties as required.

You also can add a generator to a Virtual Power Plant by clicking with the right mouse button on the element in the network graphic and choose *Add to...* → *Virtual Power Plant...* from the context sensitive menu.

15.4 Boundaries

Boundaries are used in the definition of network reductions and to report the interchange of active and reactive power after a load flow calculation. Boundary objects (*ElmBoundary*) may define topological regions by specifying a topological cut through the network.

Boundaries are defined by the cubicles that determine the cut through the network, these cubicles together with the orientations define the corresponding "Interior Region". Topologically, the interior region is found searching through the network starting at each selected cubicles towards the given direction. The topological search continues until either an open switch or a cubicle that is part of the boundary list is found. Any open switch that is found by this search is considered to be part of the interior region.

Boundaries can be defined using the *Boundary Definition Tool* or directly on the branch elements by right clicking on them and selecting *Define* → *Boundary. . . .*

The Boundaries are stored in the project folder *Boundaries* ( ) within the *Network Data*.

Note: Boundaries that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time.

15.4.1 Boundary Definition Tool

The Boundary Definition Tool ( ) is located within the *Additional Tools* toolbox as shown in figure 15.4.1.

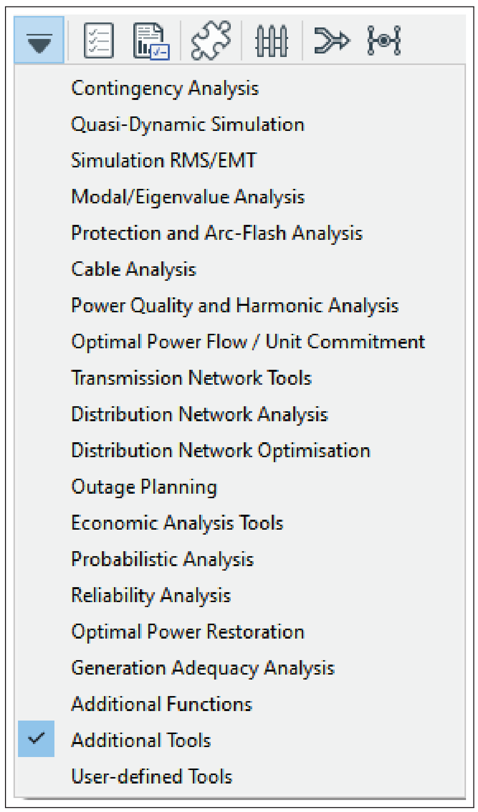


Figure 15.4.1: Additional Tools toolbox

The following options are available when using the Boundary Definition Tool command:

15.4.1.1 Basic data

Based on regional elements

Zones, areas, grids and even existing boundaries can be used to define a boundary. The selection supports multiple elements of the same type.

When regional elements are used, some additional options are available for the user:

- One common boundary: single boundary containing all the interior elements of the composing regions.
- Separate boundary for each region: a number of boundaries corresponding to the number of regions will be defined with corresponding interior elements.
- All boundaries between neighbouring regions: each combination between selected regions is considered and corresponding boundary is defined.

Based on branch elements

Branch elements (e.g. lines, transformers) can be used to define a boundary, *PowerFactory* will perform a topological search to define the interior elements. To finishing defining the interior region, the user can check the *Assign selected branch elements to interior region* checkbox on the *Basic Data* page of the command dialog.

In addition the *Boundary Definition Tool* offers the possibility to define the boundary only if it splits the network into two separated regions.

15.4.1.2 Advanced

The options available on this page depend on whether the boundary is based on regional elements or branch elements.

Use fictitious border network

This option is available when the boundary is being created based on regional elements. It is intended specifically for networks that have a "fictitious border grid", which contains the nodes marking the borders between adjacent grids. This is a typical configuration in European networks to facilitate CIM (CGMES) data interchange. The option is relevant when one common boundary is to be created. If this option is checked, the fictitious border grid must also be selected. Then, those fictitious border grid elements which connect adjacent regions that have been selected to define the common boundary will also be included.

Choice of the border inside a branch (*.ElmBranch)

The option *Prefer the longest line modelled as distributed parameters* is available when the boundary is being created based on branch elements, and is relevant when the user is defining a boundary for the purpose of carrying out a cosimulation calculation (see Section 29.10). It gives the user control over the positioning of the boundary border within an *ElmBranch* so as to ensure that the cosimulation regions are connected via lines of sufficient length.

15.4.2 Element Boundary

The element boundary *ElmBoundary* edit dialog is accessible by double clicking on the boundary element, using either the Data Manager or the Network Model Manager.

To add cubicles to an existing Boundary:

- In the Boundary dialog, right click on the table (on the number of a row) that lists the included cubicles.
- Select *Insert rows, Append rows* or *Append n rows* from the context sensitive menu.

- Double click on the *Boundary Points* cell of the new line.
- Select the target cubicle using the data browser that pops up.

After selecting the desired cubicle, the terminal and the branch element connected to it are added to the *Busbar* and *Branch* cells on the table. By default the *Orientation* (direction used to determine the interior region) is set to the branch; you can change it in order to direct the definition of the internal region to the connected terminal.

Cubicles can be retired from a boundary by selecting *Delete rows* from the context sensitive menu of the table in the element dialog.

The selected colour underneath the boundary name is used when representing the boundary in single line diagrams (). Each element in the graphic is coloured according to the following criteria:

- If it uniquely belongs to one interior region of a boundary to be drawn, its colour will be assigned to that specific boundary colour.
- If it belongs to exactly two of the interior regions of the boundaries to be drawn, its will be represented with dashed lines in the specific boundary colours.
- If it belongs to exactly more than two of the interior regions of the boundaries to be drawn, its will be represented with dashed lines in black and the colour selected for multiple intersections.

15.4.2.1 Boundary object buttons

The **Edit Interior Elements** button can be used to list in a data browser all the components included in the internal region. The **Mark Interior Region** button marks all the components of the interior region in the selected network diagram.

Topological changes in the network that affect the defined interior regions are automatically detected by the program.

The **Check Split** button can be used to check whether or not the boundary is a closed boundary which splits the network into two parts.

Related to the Check Split is an option *Topological search: stop at open breakers*. Some boundaries may only split the network because particular breakers are open, i.e. they effectively rely on these breakers to ensure that they are "splitting" boundaries. By selecting the *Topological search: stop at open breakers* option, this ensures that they are taken into account. In some cases, a boundary may be "splitting" only if the *Topological search: stop at open breakers* option is selected; in such a case the user can find out which switches are critical by using the **Report open switches making boundary split** button to get a list of these switches.

The **Colour graphic according to this boundary** will set the colouring option of the currently active graphic according to the Boundary Definition of the boundary in question. This is to help users visualise large boundaries in particular, as they create or modify them. (Note that if the original colouring scheme needs to be restored subsequently, this will have to be done manually.)

15.5 Circuits

Circuits are objects of class *ElmCircuit* (), and are used to group branches in order to clarify which branches are connected galvanically. Each branch (*ElmBranch*) can have a reference to any defined circuit object. This feature allows branches to be sorted according to the circuit to which they belong.

To create a new Circuit:

- In the Data Manager open the Circuits folder from the Network Model.
- Click on the New Object icon.
- The edit dialog of the new Circuit pops up. Give a name to the new object and press **Ok**.

Branches are added to a circuit using the pointer from the 'Circuit' field of the branch dialog. The button **Branches** in the Circuit dialog opens a data browser listing the branches that refer to that circuit.

Note: Circuits that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time.

15.6 Feeders

When analysing a system it is often useful to know where the various elements are receiving their power supply from. In *PowerFactory* this is achieved using Feeder Definitions (*ElmFeeder* ↦).

A feeder is defined at a line or transformer end, and then the feeder definition algorithm searches the system from the definition point to determine the extent of the feeder. The feeder ends when:

- An open breaker is encountered; or
- The end of a line of supply is encountered; or
- 'Terminate feeder at this point' is enabled in a cubicle (optional); or
- A higher voltage is encountered (optional).

Once a feeder has been defined it may be used to scale the loads connected along it according to a measured current or power, to create voltage profile plots or to select particular branches and connected objects in the network. Following load flow calculations, special reports can be created for the defined feeders. To distinguish the different feeder definitions, they can be coloured uniquely in the single line graphic. All feeder objects are stored in the Feeders folder (↦) in the *Network Data* folder. A new feeder is created by right-clicking on a cubicle (that is, when the cursor is held just above the breaker in the single line diagram) and selecting *Define* → *Feeder...*. Once the option Feeder has been selected, the Feeder dialog pops up. There you can define the desired options for the new object. After pressing **Ok**, the new Feeder is stored in the Feeders folder of the Network Model.

Any existing Feeder can be edited using its dialog (double click the target Feeder on a data browser). The Feeder dialog presents the following fields:

Name

Cubicle Is a reference to the cubicle where the Feeder was created. It is automatically set by the program once the Feeder is created.

Zone Reference to the Zone (if any) to which the feeder belongs. A Feeder is assigned to the zone of the local busbar/terminal.

Colour Sets the colour be used when the Feeder Definitions colouring mode () is engaged in the single line diagram.

Terminate feeder when... A feeder will, by default, terminate when a higher voltage level is encountered, however, this may not always be desirous. This may be prevented by un-checking this option. The feeder will now continue 'past' a higher voltage level and may be terminated at a user defined cubicle if desired. To manually terminate a feeder right-click a branch element above the breaker (to select the desired cubicle where the feeder is going to end) and select Edit *Cubicle*. The cubicle dialog will be presented, and the 'Terminate feeder at this point' option may be checked.

Orientation The user may select the direction towards the feeder is defined. 'Branch' means that the feeder starts at the cubicle and continues in the direction of the connected branch element. 'Busbar' means that the Feeder is defined in the direction of the connected Terminal.

Load Scaling In any system some loads values may be accurately known whilst others are estimated. It is likely that measurement points exist for feeders in the system as well, and thus the power that is drawn through this feeder is also known. The load scaling tool assists the user in adjusting these estimated load values by scaling them to match a known feeder power or current that has been measured in the real system. More information about the use of the Load Scaling Function is given below.

Elements The **Mark in Graphic** button may be used to select all the elements of a Feeder in the desired single line diagram. The **Edit** button is used to list all the elements belonging to a Feeder in a data browser.

To use the Load Scaling tool first define which loads may be scaled by enabling the 'Adjusted by Load Scaling' option on the Load-Flow tab of the load dialog. All of the loads in a feeder may also be quickly viewed by editing the feeder from the feeders folder. Load scaling is now performed by the load-flow calculation function when:

- At least one feeder is defined with load scaling according to a current or power.
- The option 'Feeder Load Scaling' is enabled in the load-flow command dialog (basic options).
- At least one load exists in the feeder area for which
  - A change in operating point affects the load-flow at the feeder position
  - The option 'Adjusted by Load Scaling' has been enabled.

The load-flow calculation will then adjust the scaling of all adjustable loads in the feeder areas in such a way that the load-flow at the feeder equals the current or power set-point.

The feeder setpoint is influenced by the zone scaling. This means that the current or power flow as calculated by the load-flow could differ from the setpoint in the feeder dialog when the busbar where the feeder is defined is part of a zone.

For instance, a feeder has a set-point of 1.22 MVA. The busbar is in a zone and the zone-scale is set to 0.50. The flow at the feeder position will thus be 0.61 MVA.

For information about colouring the single line graphic according to feeder definitions refer to Chapter 9: Network Graphics. For information about voltage profile plots, refer to Chapter 19 (Reporting and Visualising Results).

Defining Feeders from a Terminal Element

Often it is useful to be able to quickly setup a feeder or many feeders from a 'source' bus within the system. There is a specific methodology within *PowerFactory* for this purpose. The procedure is as follows:

1. Right-click the target terminal where the feeder/s should be defined from.
2. Choose the option *Define* → *Feeder...* from the context sensitive menu that appears. This step is illustrated in Figure 15.6.1.
3. *PowerFactory* will automatically create Feeder objects for each of the connected two terminal elements, for example lines and transformers. The list of created feeders is displayed in a pop-up window. The default name for each feeder is the concatenation of the terminal name and the connected object.
4. Adjust the feeder colours and definitions as required and remove any unwanted feeders.

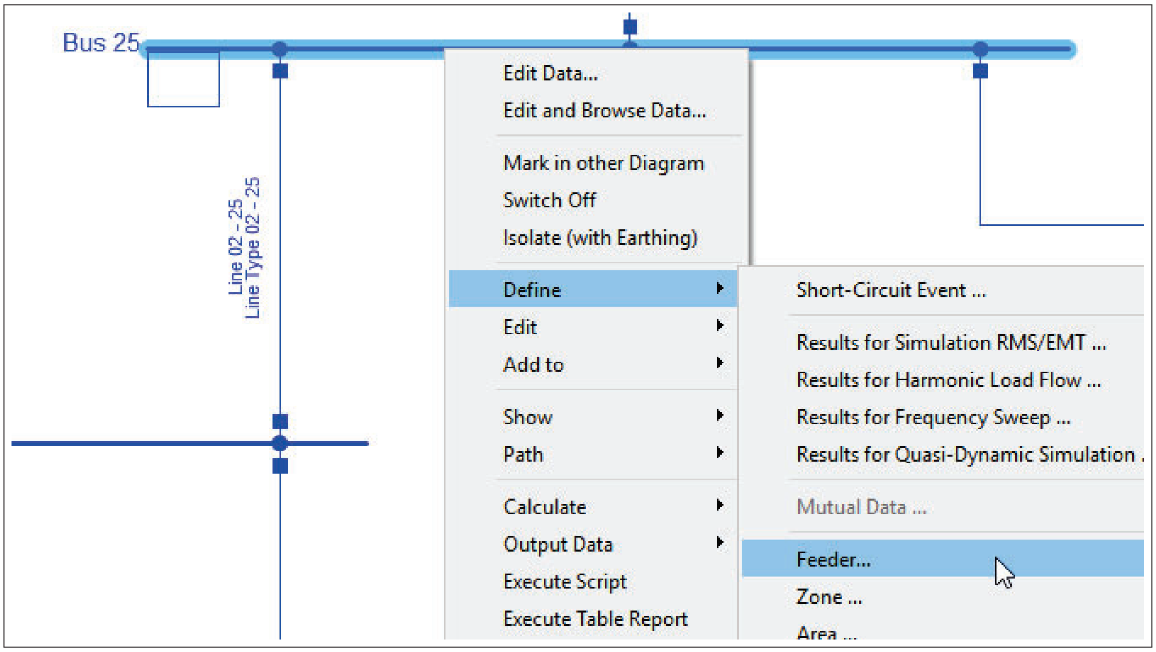


Figure 15.6.1: Definition of Feeders from a terminal by right-clicking the terminal

Note: The Load Scaling options are part of the operation scenario subsets; therefore they are stored in the active operation scenario (if available). The Load Scaling options are stored in the active expansion stage (if available) if no active operation scenario is active. feeders that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding Variation is active and the expansion stage activation time is earlier than the current study time.

15.6.1 Feeder Tools

Feeder Tools is a set of three tools that can be used only in radial systems to change voltage, technology from a particular point downwards.

Note: Additional functions for feeders like *Backbone Calculation* or *Phase Balance Optimisation* are available in the module *Distribution Network Tools*, described in chapter 41

15.6.1.1 Voltage Change Tool

The Voltage Change Tool automatically changes type data (for transformers, lines, loads and motors) and element data such that the primary voltage can be changed to a specified voltage value. The tool will change the voltage from a particular point downwards but is limited to the HV side. This will enable the voltage level of a network to be changed for planning studies, taking into account all downstream equipment.

15.6.1.2 Technology Change Tool

The Technology Change Tool automatically changes type data (for transformers, lines, loads, motors) and element data such that the primary number of phases or neutrals (commonly referred to as 'technology') can be changed to a specific number of phases/neutrals. The tool will change the technology from a particular point downwards but is limited to the HV side.

Note: If a device such as a transformer or shunt device is no longer compatible (number of phases and/or phasing is not supported) then the device is set out of service and is reported to the user.

15.6.1.3 Feeder Tool Command

Feeder Tools is a built-in command (*ComFeedertool*) in *PowerFactory* and can be started via the right-mouse context-sensitive menu, by clicking on an element of a feeder as shown in Figure 15.6.2. A radial feeder must be defined prior to using the command.

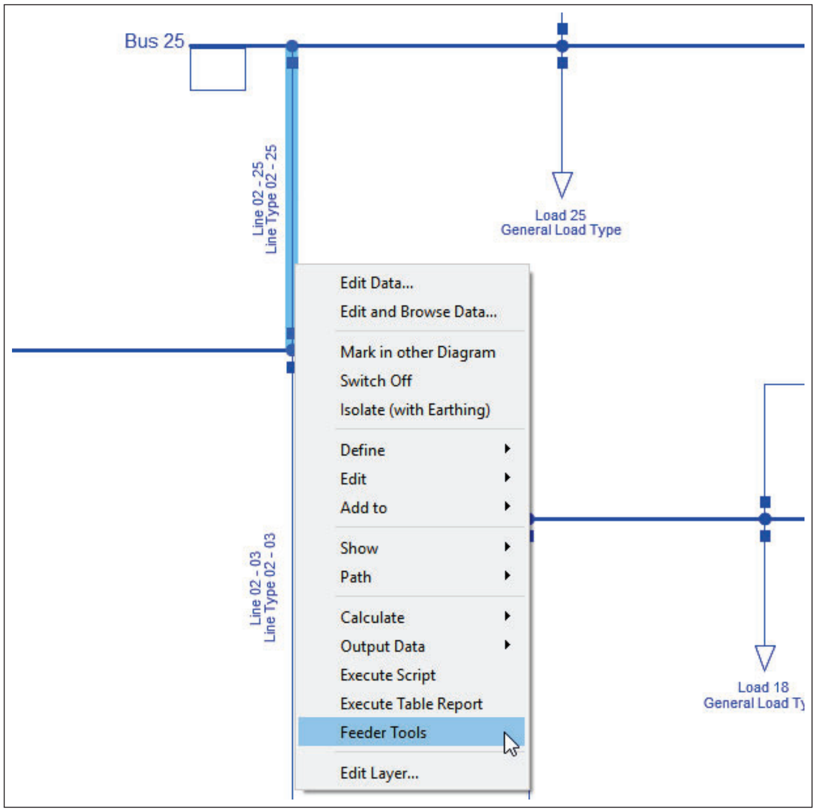


Figure 15.6.2: Feeder Tool

The voltage, technology and balancing tools are all related and are integrated in *PowerFactory* as one command having different options for enabling/disabling each individual tool. Any combination of the three tools can be used. For example, a user may want to evaluate the alternative where an existing 19 kV SWER line is to be changed to a 22 kV three-phase line. In this case, the line type voltage, phasing and technology will all need to change. The transformers should then be changed to equivalent single- or dual-phase transformers (depending on their original secondary technology) with 22 kV phase-to-phase connected primary windings.

Since Voltage and Technology Tools are more intrinsically related than the Auto Balancing Tool, the first tools are meshed into one algorithm. The Auto Balancing Tool runs independently of Voltage and Technology Tools but requires a convergent load flow. If the user wishes to apply all tools in one run (Voltage, Technology and Balancing), then the algorithm of Voltage and Technology Tools is performed followed by execution of the Auto Balancing Tool.

15.6.1.4 How to use the Voltage and Technology Tool

When selecting the Voltage Change Tool, the user should specify the voltage level in kV (*Previous Voltage*) that will be replaced, and the *New Voltage*. Both voltages should be specified as phase-phase voltages, even if there is no phase-phase voltage available; for example when the previous or new technology is 1 PH or 1 PH-N. When selecting the Technology Change Tool, the user should specify the *New Technology* from the drop-down list and then proceed as follows:

1. A radial feeder must be defined.
2. A *Start Element* (terminal or line) must be selected:
   - If the *Start Element* is a terminal, then this is defined as the *Start Terminal*.
   - If the *Start Element* is a line, then the *Start Terminal* is defined as:
     - For the Voltage Tool: the line terminal nearest to the feeder definition point
     - For the Technology Tool: the line terminal more distant from the feeder definition point.

Note: The algorithm uses a top-down approach: working from the *Start Terminal* downwards to the *Stop Point*

---

3. Definition of *Stop Point* for Voltage/Technology Tools: The voltage/technology changes will stop at transformers or open points. For transformers, only the primary voltage/technology is changed. This means that the transformer secondary voltage/technology and secondary network remains unchanged. If the transformer technology (three phase, single phase or SWER) is not compatible with the new primary technology, then the transformer will be disconnected. This will occur when a three-phase primary network supplies a three-phase transformer and the primary technology is changed to a non-three-phase technology. In this case, the transformer will be disconnected. Likewise, three-phase machines cannot be connected to a non-three-phase technology. (Note: Out-of-service elements are not Stop Points for Voltage/Technology Tools.)
4. Setting the new type/element voltage: If *Voltage Change Tool* is selected, the new voltage is equal to the *New Voltage* specified by the user. A voltage change can be performed independent of a technology change. If *Technology Change* is disabled, the voltage change will be associated with the existing technology.
5. Setting the new type/element technology: If *Technology Change Tool* is selected, the new technology is that of the *New Technology* specified by the user. A technology change can be performed independent of the voltage (the voltage will not be changed).
6. A *Linking Object* must be provided.
   The selection of a new Type is not automated as there may be several types that could be compatible with a particular scenario. The solution for this is a linking database object. This linking object stores the relationships between old and new types for different new voltage and/or technology changes. This linking object can be saved in a project or library. It should be added to and modified each time a technology/voltage change is performed.
   If for any network element a new type that should match a specific new voltage/technology is not found, the user can choose how the program should proceed by selecting one of the following Linking Object options:
   - Prompt for new type selection: the user should manually select which type should be used. If the selected type is still not valid (see item 7: Validation rules for types), the program will present new options: (i) the user can select a new type again, (ii) ignore replacing this type, or (iii) interrupt algorithm execution. Otherwise, if the selected type is valid (see item 7: Validation rules for types), the existing record in Linking Object is updated or in the event that it does not exist, a new one will be created.
   - Automatically creates new type: a new type that matches the required voltage/technology is automatically created and the existing record in the Linking Object is updated, or in the event that it does not exist, a new one will be created.
   - Do not change the old Type: the old type is not replaced and the corresponding element is put out-of-service. Changes, if necessary, should be manually performed after the command execution.

An example of a *Linking Object* is shown in Figure 15.6.3. The voltage tolerance (parameter *vtol*) for comparison between type voltage and new voltage can be optionally specified. The default value is 30 %. Records in *Linking Object* should be unique for each combination of Old Type, New Voltage and New Technology. Validation rules (see item 7) are applied when the user presses the **OK** button or/and automatically (i.e. within the algorithm).

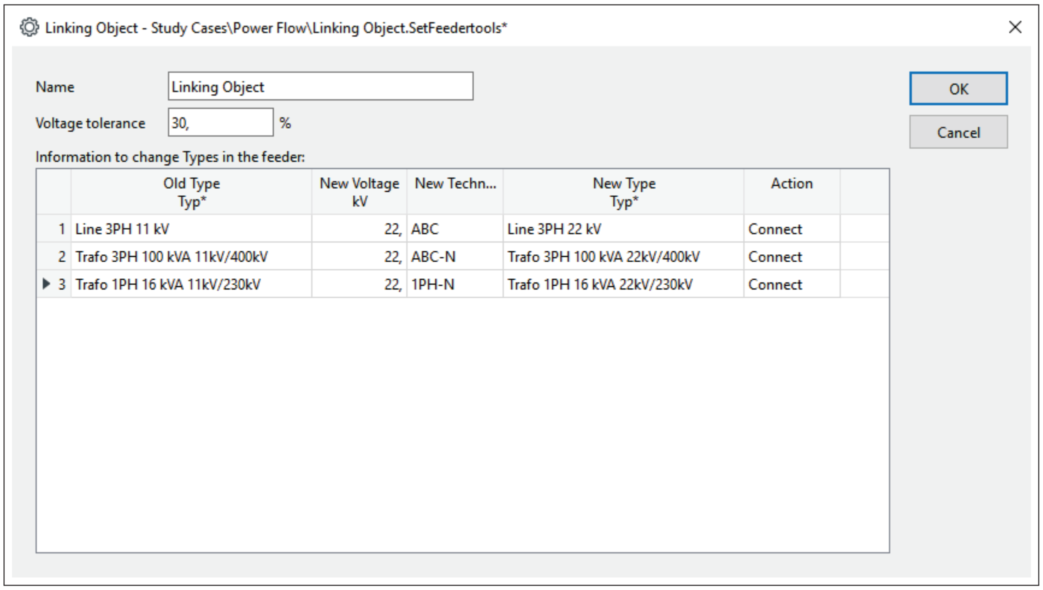


Figure 15.6.3: Linking Object

7. Validation rules for types:
   a. The new type voltage must be equal to the new voltage (item 4) or inside its tolerance specified by parameter *vtol* in the Linking Object (item 6). If this rule fails, the message “The Type selected voltage is incompatible with the new voltage” is shown. This rule does not apply to the load type (*TypLod*).
   b. For the motor type (*TypAsmo*, *TypAsm*), the number of phases must be equal to or greater than the old number of phases. If this condition is not met the message “Cannot supply a three-phase motor from a single or bi/dual phase” is shown.
   c. For the 2-winding transformer type (*TypTr2*), the number of phases must be equal to or greater than the old number of phases. If this condition is not met the message “Cannot supply a three-phase secondary network from a single or bi/dual phase primary” is shown.
   d. As stated in item 3, the voltage/technology changes will stop at the primary of a transformer so that the secondary network remains unchanged. Despite this, the algorithm does not limit the selected 2-winding transformer type to the same secondary voltage/technology.
   e. For the line type (*TypLne*) the number of phases and neutrals must be equal to the one required by the new technology specified by the user. For the 2-winding transformer type (*TypTr2*), motor type (*TypAsmo*, *TypAsm*) or load type (*TypLod*), the number of phases and neutrals must be less than or equal to the one required by the new technology. If this condition is not met the message “The type selected technology is incompatible with the new technology” is shown.
   f. The algorithm will change a type object when:
      - The actual type voltage/technology is not compatible with the new voltage/technology;
      - The actual type voltage/technology is compatible with the new voltage/technology but the change operation is specified in the Linking Object.

      If the type is incompatible with the new voltage/technology and a new type has not yet been specified in the *Linking Object* for a particular old type, *New Voltage* and *New Technology* combination, then one of the following predefined actions will occur:

- Prompt for new type selection: during this process the user can select or create a new type, where the default parameters are the same as those for the old type. Once a new type has been selected or created, it will be checked to ensure that it is compatible with the new voltage/technology. If it is still incompatible, the user will be asked whether to select again or to put this element out-of-service. The changes/additions are stored in the Linking Object so that they are available for future reuse.
- Automatically creates a new type: a compatible type is created where the default parameters are the same as those for the old type. The changes/additions are stored in the Linking Object so that they are available for future reuse.
- Do not change the type: no new type is selected and the element is set out-of-service.

15.7 Meteo Stations

It is often the case that *groups* of wind generators have a wind speed characteristic that is correlated. *PowerFactory* can represent such a correlation through the *Meteo Station* (*ElmMeteostat* object. The meteorological stations are stored within the folder *Network Data*.

Meteorological stations can be defined either via the element that is to be part of the meteorological station (from any of the generator elements described in Section 47.4), or via the single line diagram by right-clicking on an appropriate element and selecting *Define* → *Meteo Station* (or *Add to* → *Meteo Station*) from the context-sensitive menu. Note that the ability to define a *Meteo Station* is dependent upon whether at least one of the 'member' generators has the options *Generator* and *Wind Generator* selected on its *Basic Data* page. If these options are not selected, the context menu entry is not visible.

Note: A graphical colouring mode exists for Meteorological Stations, so that they can be visualised in the single line graphic.

15.8 Operators

For descriptive purposes, it is useful to sort network components according to their operators. Additionally, system operators may find it advantageous to generate summary reports of the losses, generation, load, etc. according to their designated region(s). *PowerFactory* allows the definition of operators, the assignment of network components to these operators, and the identification of operators on single line diagrams by means of Operator objects. The Operator objects (*ElmOperator*,) are stored in the *Operators* folder () in the *Network Model* directory.

To create a new operator:

- In the Data Manager open the Operators folder from the Network Model.
- Click on the 'New Object' icon.
- The edit dialog of the new operator pops up:
  - Give a name to the new object.
  - Select a colour to represent the operator in the corresponding colouring mode of the single line diagram.
  - Press **Ok**.

Network elements (class name *Elm**) such as terminals, switches, lines, generators, transformers, relays or composite models (*ElmComp*), Substations (*ElmSubstat*) and Branches (*ElmBranch*) can be assigned to an operator by means of the reference 'Operator' from the Description tab of their dialog.

Note: Operators that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time

---

15.9 Owners

For descriptive purposes it is useful to sort network components according to their owners. Additionally, for network owners it may prove advantageous to generate summary reports of the losses, generation, load, etc. for their region(s). Similar to Operators, *PowerFactory* allows the definition of network owners, and the assignment of network components to them, by means of Owner objects.

The Owner objects (*ElmOwner*,) are stored in the 'Owners' folder () in the *Network Model* directory. They are created following the same procedure described for operators. Network elements (class name *Elm*)* such as terminals, switches, lines, generators, transformers, relays or composite models (*ElmComp*), Substations (*ElmSubstat*) and Branches (*ElmBranch*) can be assigned to an operator by means of the reference 'Operator' from the Description tab of their dialog.

Note: Operators that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time

---

15.10 Paths

A path (*SetPath*,) is a set of two or more terminals and their interconnected objects. This is used primarily by the protection module to analyse the operation of protection devices within a network.

The defined paths can be coloured in a single line graphic using the colouring function. New paths are stored inside the *Paths* folder () in the *Network Data* directory.

To create a new Path:

- In a single line diagram select a chain of two or more terminals and their inter-connecting objects.
- Right click on the selection.
- Select the option *Path* → *New* from the context sensitive menu.
- The dialog of the new path pops up, give a name and select the desired colour for the corresponding colour representation mode in the single line diagram. The references to the objects defining the Path (First/Last Busbar First/Last Branch) are automatically created by the program, according to the selection.
- After pressing **Ok** the new path is stored in the Paths folder of the Network Model.

Alternatively, the path can be created as follows:

- In the single line diagram, select the first element to be included in the path definition by pressing left mouse button and select the last element in combination with CTRL+Shift keys. As a result, the shortest path between the first and last element will be highlighted.
  - In order to constrain the route, additional elements can also be selected in combination with CTRL key.
  - The paths with multiple end terminals can be created by selecting the multiple end terminals using left mouse button in combination with CTRL+Shift keys.

   - In order to consider the state of switches and to define a path consisting of closed switches only, select the last element of the path using left mouse button in combination with CTRL+Shift+A keys.

- Next, right click on the selection.
- Select the option *Path* → *New* from the context sensitive menu.
- Consequently, a new path is created. The names of the first and the last elements are used for defining the corresponding path name. However, the name can be further edited, if required.

By using the **Elements** button of the Path dialog you can have access to all the element belonging to the path in a data browser, there you can edit them. The **Select** button may be used to locate the components of the path in a single line diagram. With the **Toggle** button you can invert the order of the objects limiting the path (First/Last Busbar First/Last Branch). This order is relevant when evaluating directional protective devices.

In cases where a path forms a closed ring the **First Busbar** button of the SetPath dialog can be used to specify at which busbar the path should be considered to begin and end. This can be particularly useful when displaying the path on a time distance diagram.

New objects can be added to a path by marking them in a single line diagram (including one end of the target path and a busbar as the new end) right clicking and selecting *Path* → *Add* to from the context sensitive menu. Objects can be removed from a Path (regarding that the end object of a Path must be always a busbar) by marking them in the single line diagram, right clicking and selecting *Path* → *Remove Partly* from the context sensitive menu. The Remove option of the Path context sensitive menu will remove the firstly found path definition of which at least one of the selected objects is a member.

For information about the colouring function refer to Chapter 9: Network Graphics. For information about the use of the path definitions for the analysis of the protective devices, refer to Chapter 33 (Protection).

Note: Paths that are created or deleted when a recording expansion stage is active; become available/not available only if the corresponding variation is active and the expansion stage activation time is earlier than the current study time

---

15.11 Routes

Routes are objects which are used to group line couplings (tower elements). Each coupling (*ElmTow*) can have a reference to any defined route (*ElmRoute*, ). Each route has a colour that can be used to identify it in single line diagrams, when the corresponding colouring function is enabled.

For information regarding line couplings refer to the technical reference for the transmission line model (see Technical References Document).

15.12 Zones

Components of a network may be allocated to a zone object (*ElmZone*, ) in order to represent geographical regions of the system. Each zone has a colour which can be used to identify the elements belonging to it in the single line graphic if the colouring is set to *Groupings* → *Zones*. These elements can be listed in a browser format for group editing; additionally all loads belonging to the zone can be quickly scaled from the zone edit dialog. Using the 'Scaling Criteria'-Button, the total active or apparent power demand can be set as absolute values in MVA or MW. The zone scaling factor is then calculated accordingly. Another special scaling factor is provided for all generators, its plant category is set to 'Wind' and which are not part of other controllers: The Wind Generation Scaling Factor.

Reports for the defined zones can be generated following calculations.

Upon being defined, zones are by default stored inside the Zones folder () in the *Network Data* folder.

Zones are created by selecting a node or multi-selecting elements (at least one node-element has to be among them), right-clicking and choosing *Define* → *Zone...* from the context sensitive menu. The option *Add to* → *Zone...* can be selected when a zone(s) have already been defined. Single-port elements are directly assigned to the zone, its connected node is part of. For multi-port elements (like lines or transformers) one of the available terminals has to be chosen, from which the zone assignment is inherited.

Chapter 16

Operation Scenarios

16.1 Introduction

Operation Scenarios are used to store operational data such as generator dispatch, load demand, and network line/switch status. Individual Operation Scenarios are stored within the **Operations Scenarios** folder, and can be easily activated and deactivated. This Chapter describes *PowerFactory* operation scenarios.

Note: Parameter Characteristics can also be used to modify network operational data - see Section 18.2 (Parameter Characteristics) for details.

16.2 Operation Scenarios Background

Operation Scenarios are used to store network component parameters that define the operational point of a system. Examples of operational data include generator power dispatch and a load demand. Operational data is typically distinguished from other component data because it changes frequently. Compare for instance, how often a generator changes its power set-point, with how often the impedance of the generator transformer changes.

Storing recurrent operation points of a network and being able to activate or deactivate them when needed accelerates the analyses of the network under different operating conditions. *PowerFactory* can store complete operational states for a network in objects called operation scenarios (*IntScenario*,).

Operation scenarios are stored inside the operation scenarios folder () in the project directory. You can define as many operation scenarios as needed; each operation scenario should represent a different operational point.

Figure 16.2.1 shows a project containing four operation scenarios, and the contents of the 'High Load - No Infeed' scenario (i.e. its subsets) are shown in the right pane of the Data Manager.

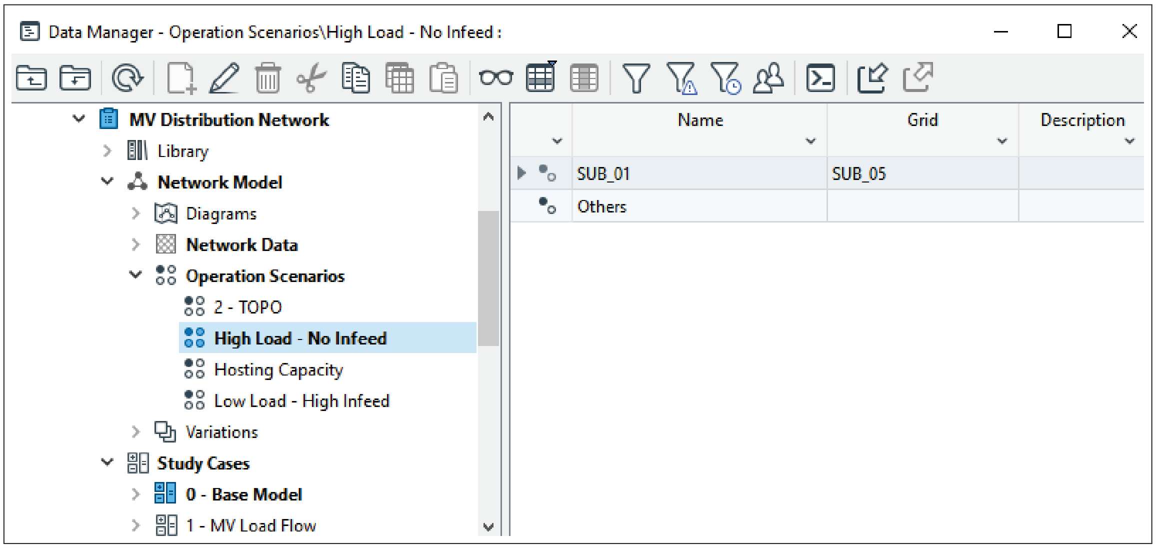


Figure 16.2.1: Operation Scenarios and Subsets

A new operation scenario is defined by saving the current operational data of the active network components. Once they have been created, operation scenarios can be activated to load the corresponding operational data. If an operation scenario is active and certain operational data is changed, these changes are stored in the active operation scenario (if you decide to save the changes). If the current operation scenario is deactivated, the active network components revert to the operational data that they had before the activation of the operation scenario (this is the 'default' operational data). Changes made to the 'default' operational data do not affect data within existing operation scenarios.

Operation scenario data stored within each operation scenario is separated into subsets, with one subset of operational data created for every grid in the network model. It is possible to 'exclude' the operational data for individual grids. This prevents the operation scenario from saving the operational data for any subset where this option is active. For example, you might be working with a network model with four grids, say North, South, East and West. Perhaps you do not wish to store operational data for the 'West' grid because the models in this grid have fixed output regardless of the operational state. By excluding the operational data subset for this grid, the default data can be used in all cases, even though the operational data is different in the other three grids.

When working with active operation scenarios and active expansion stages, modifications on the operational data are stored in the operation scenario whereas the expansion stage keeps the default operational data and all other topological changes. If no operation scenarios are active and new components are added by the current expansion stage, the operational data of the new components will be added to the corresponding operation scenario when activated.

Note: When an operation scenario is active, the operational data can be easily identified in network component dialogs and in a Network Model Manager because it is shown with a blue background. The colouring is configurable by the user: see Section 7.8

16.3 How to use Operation Scenarios

This sub-section explains how to complete the common tasks you will need when working with operation scenarios. The most common tasks are creating a new operation scenario, saving data to an operation scenario, Activating an existing operation scenario, Deactivating an operation scenario and identifying parameters stored within an operation scenario.

16.3.1 How to create an Operation Scenario

There are two ways to create an operation scenario.

Method 1

Follow these steps:

1. In the Data Manager, right-click on the *operation scenarios* folder in the active project.
2. Select *New* → *Operation Scenario* from the context-sensitive menu as shown in Figure 16.3.1. The dialog of the new operation scenario pops up.

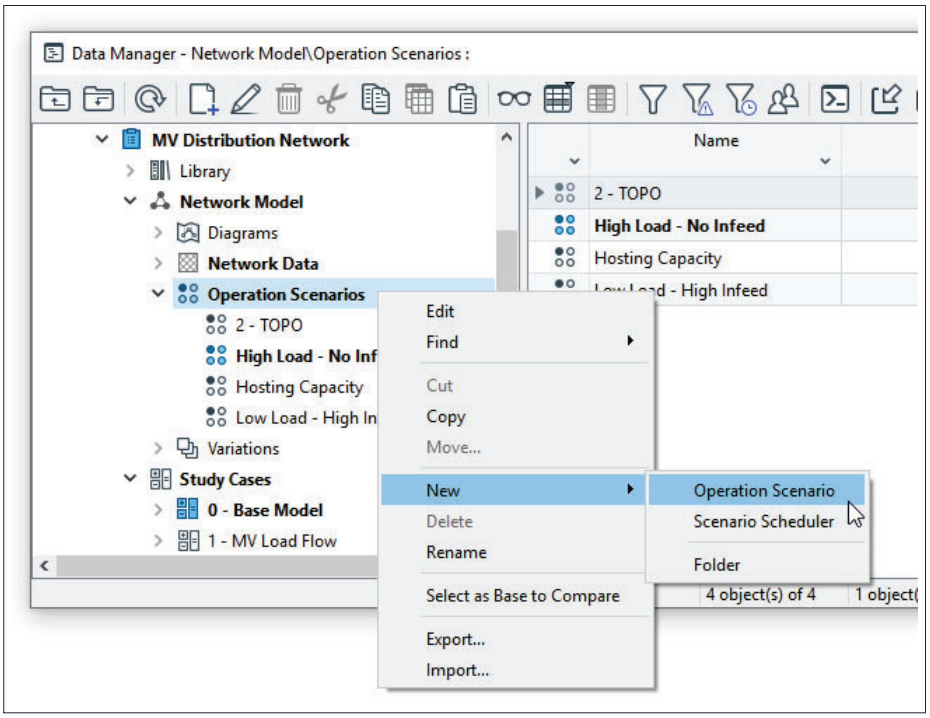


Figure 16.3.1: Creating a new operation scenario object using the Data Manager.

3. Enter the name for the operation scenario in the name field.
4. Press **OK**. The operation scenario will appear as a new object within the Operation Scenarios folder.

Method 2

Follow these steps:

1. From the main *PowerFactory* menu go to the File menu and select *File* → *Save Operation Scenario as. . .* . The dialog of the new operation scenario pops up.
2. Enter the name for the operation scenario in the name field.
3. Press **OK**. The new operation scenario is created within the operation scenarios' project folder and automatically activated and saved.

16.3.2 How to save an Operation Scenario

Why do you need to save Operation Scenarios?

Unlike all other *PowerFactory* data, changes to operational data are not automatically saved to the database if an operation scenario is active. So, after you update an operation scenario (by changing some operational data) you must save it. If you prefer automatic save behaviour, you can activate an automatic save option setting - see Section 16.6.1.

How to know if an Operation Scenario contains unsaved data

If any operational data (of a network component) is changed when an operation scenario is active, the unsaved status of it is indicated by an asterisk (*) next to the icon for the operation scenario as shown in Figure 16.3.2. The other situation that causes an operation scenario icon to appear with an asterisk is when new network components are added to the model. Any operational parameters from these models are not incorporated in the active operation scenario until it is saved.

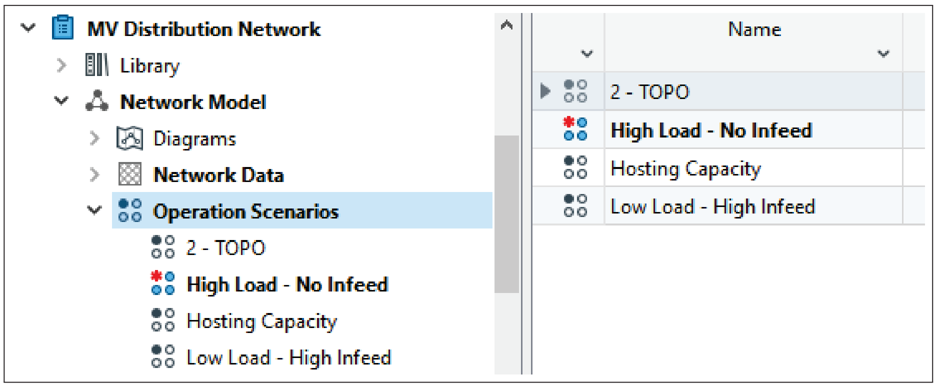


Figure 16.3.2: An asterisk indicates unsaved changes in operation scenarios

Options for saving an Operation Scenario

There are four ways to save a modified operation scenario to the database. They are:

- The menu entry *Save Operation Scenario* in *PowerFactory*'s main file menu.
- The button **Save** in the dialog window of the operation scenario.
- The button *Save Operation Scenario* ( ) in the main icon bar.
- The context-sensitive menu (right mouse button) entry Action -> Save of the operation scenario (see Figure 16.3.3).

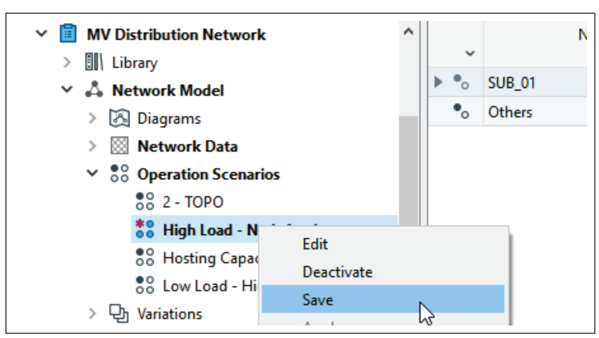


Figure 16.3.3: Saving an operation scenario using the context-sensitive menu

Note: The button **Save as** from the operation scenario dialog (only available for active operation scenarios) can be used to save the current operational data as a new operation scenario. The new operation scenario is automatically activated upon being created.

16.3.3 How to activate an existing Operation Scenario

Switching between already available operation scenarios is a common task. There are two methods for activating an existing operation scenario.

Method 1

Follow these steps:

1. Go to the operation scenarios' folder within your project using the Data Manager.
2. Right-click the operation scenario that you wish to activate. The context sensitive menu will appear.
3. Choose the option *Activate* from the menu. If a currently active operation scenario contains unsaved data, you will be prompted to save or discard this information.

Method 2

Follow these steps:

1. From the main file menu choose the option *Activate Operation Scenario*. A pop-up dialog will appear, showing you the available operation scenarios.
2. Select the operation scenario you wish to Activate and press **OK**. If a currently active operation scenario contains unsaved data, you will be prompted to save or discard this information.

Note: The active operation scenario can be displayed in the status bar. To do this right-click the lower right of the status bar and choose *display options* → *operation scenario*.

16.3.4 How to deactivate an Operation Scenario

There are two ways to deactivate an active operation scenario.

Method 1

Follow these steps:

1. Go to the 'operation scenarios' folder within your project using the Data Manager.
2. Right-click the operation scenario that you wish to deactivate. The context sensitive menu will appear.
3. Choose the option deactivate from the menu. If the operation scenario contains unsaved data, you will be prompted to save or discard this information.

Method 2

From the main file menu choose the option *Deactivate Operation Scenario*. If the operation scenario contains unsaved data, you will be prompted to save or discard this information.

Note: On deactivation of an operation scenario, previous operational data (the 'default' operational data) is restored.

16.3.5 How to identify operational data parameters

Because the operation scenario only stores a subset of the network data, it is useful to know exactly what data is being stored by the operation scenario. This is relatively easy to see when you have an active scenario. Data that is stored in the operation scenario is shown with a blue background. This appears in both the object dialogs and the Data Manager browser as shown in Figures 16.3.4 and 16.3.5.

The colouring is configurable by the user: see Section 7.8.

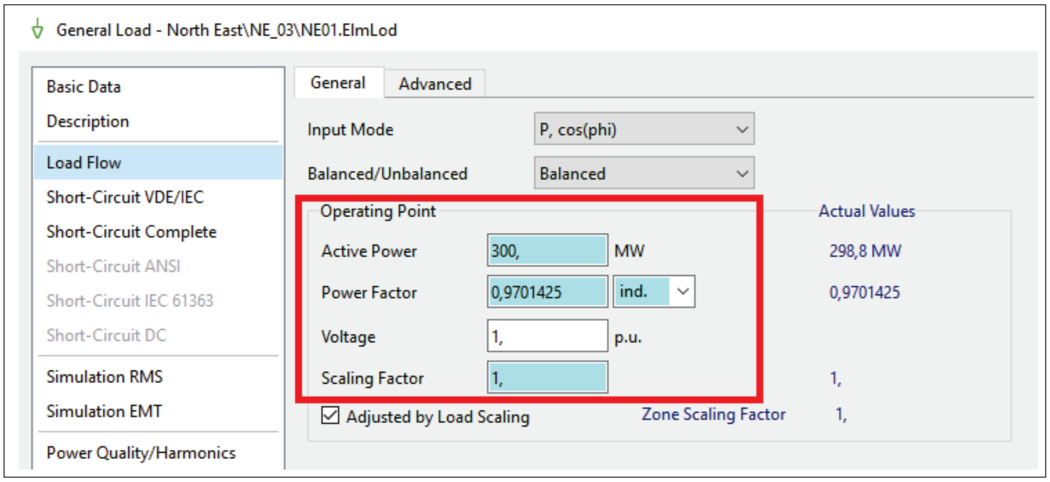


Figure 16.3.4: Blue highlighted operational data in an element dialog

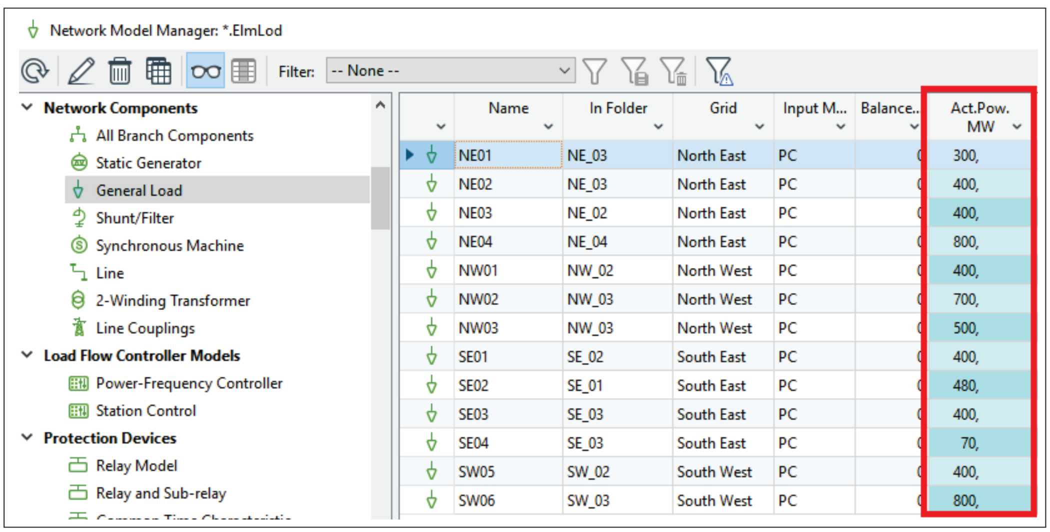


Figure 16.3.5: Blue highlighted operational data in a browser window

16.4 The Operation Scenario Manager

The Operation Scenario Manager is a tool which has been introduced to enable users who work with operation scenarios to compare data between scenarios and easily copy data values from one scenario to another.

It will be noted that some of the "How to" tasks covered in section 16.5 also enable the user to view and copy data. In other words, there can be different ways of achieving the same result, but the choice of method will in the end depend on the precise task at hand. The Operation Scenario Manager is particularly useful for viewing the same data attributes for various scenarios together.

16.4.1 Accessing the Operation Scenario Manager

The Operation Scenario Manager can be accessed via the Network Model Manager (see Chapter 12), presenting itself as a special *Scenarios* tab (the second of four "special" tabs) within the network model manager itself. Alternatively, the Operation Scenario Manager can be started from the context sensitive menu accessed by right-clicking on the title heading of the "Operation Scenarios" section in the Project Overview window.

When the Scenarios tab is used for the first time in a project, there will be little to see, because no scenarios have been selected to be viewed, and the configuration of variables to be shown for each element class has not yet been set up.

16.4.2 Selecting scenarios and setting up variable configurations

In the descriptions below, the references such as "(1)" refer to this figure, 16.4.1:

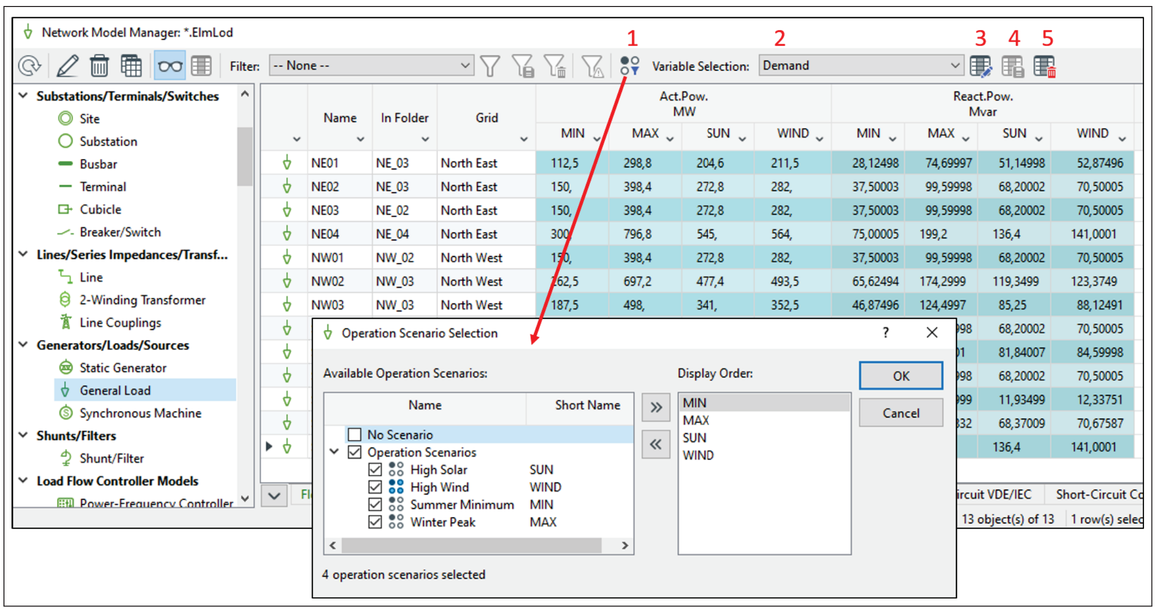


Figure 16.4.1: The Operation Scenario Manager

16.4.2.1 Selecting scenarios

The *Select Operation Scenarios* button (1) is used to specify for which scenarios the data shall be displayed. An option to see the "No Scenario" data i.e. the underlying values, is also available.

It can be seen in the *Operation Scenario Selection* dialog that a *Short Name* is listed in addition to the full scenario name. This scenario attribute is used for the column headings, as a way of avoiding the practical difficulties of having many scenarios with long names only differing by characters at the end of the name. If the user has not populated this field, the full name will be used instead.

16.4.2.2 Defining variable configurations

The variables to be displayed must also be selected. The concept is that the Operation Scenario Manager can be used with a number of different customised Variable Selection configurations, which the user first sets up and can then select as required from the drop-down Variable Selection menu (2).

When the Operation Scenario Manager tab is opened for the first time, the Variable Selection field will show "- - None - -". The user can then select an element class from the left hand pane of the network model manager and click on the *Edit Variable Selection* button (3) to start setting up the configuration. Variables are selected via standard variable selection (**.IntMon*) dialogs, and the configuration can include one or more element classes.

When the variable selection (**.IntMon*) dialog is displayed using the operation scenario manager, variables can be sorted in two separate categories by selecting from the *Variable Set* drop down menu. The following selections are available:

1. **Element Parameter**: Selection of this option will display all the available variables for the respective element. This includes both operational variables (variables which are stored in operation scenarios)and non-operational variables (variables which are not stored in operation scenarios). If non-operational variables are selected, then the operation scenario manager will display an additional column for each selected variable with the column header *Flexible Data* assigned as highlighted in figure 16.4.3.

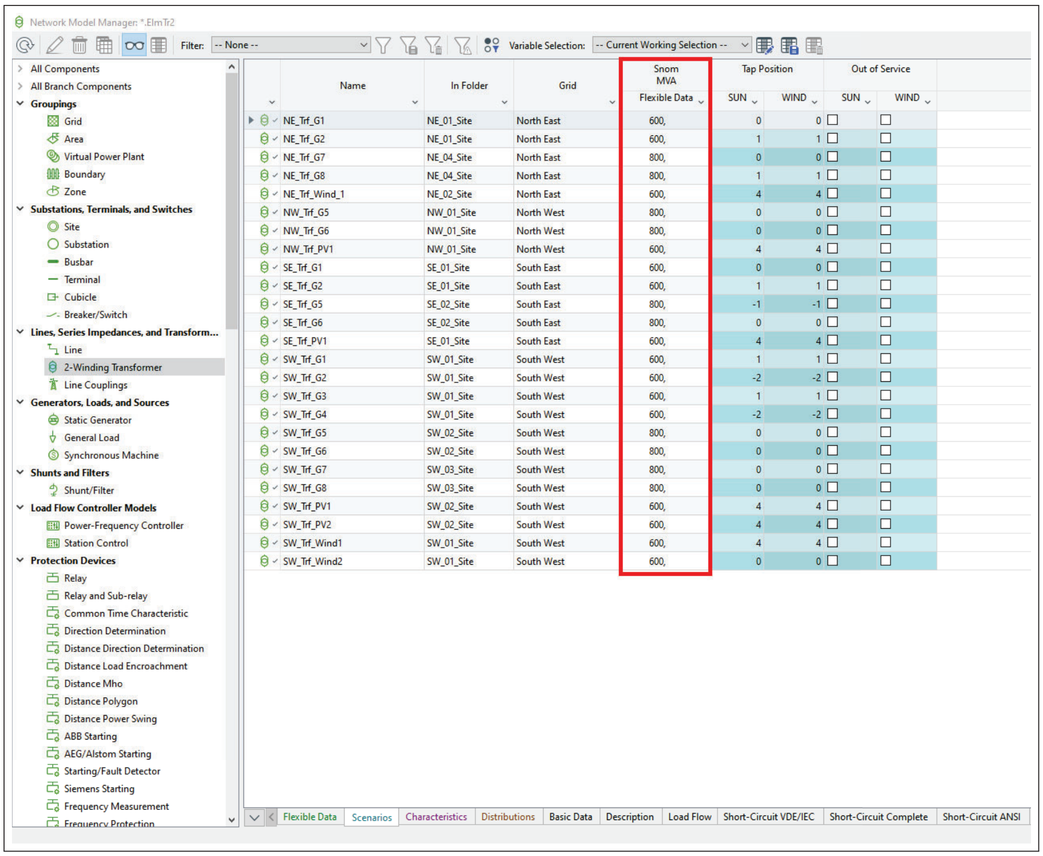


Figure 16.4.2: Flexible Data Column in Operation Scenario Manager

2. **Scenario Parameter**: If this option is selected then only operational variables are available for selection.

Once the user starts setting up the configuration, this then becomes the "- - Current Working Selection - -". When all the required variables have been selected, the configuration can be saved (4) with a user-specified name.

Configurations can be edited at any time, or deleted (5) as required.

16.4.2.3 Location of variable configurations within the project

The user can set up a number of variable selection configurations, and these are stored within the project Settings folder. Within this folder, there is a *Scenario Manager Selection* Set, and this can contain any number of folders, each of which will then have one or more **.IntMon* variable definitions, as shown in Figure 16.4.3 below. It is these folders which are the named configurations that are selected via the drop-down menu seen in Figure 16.4.1 (2).

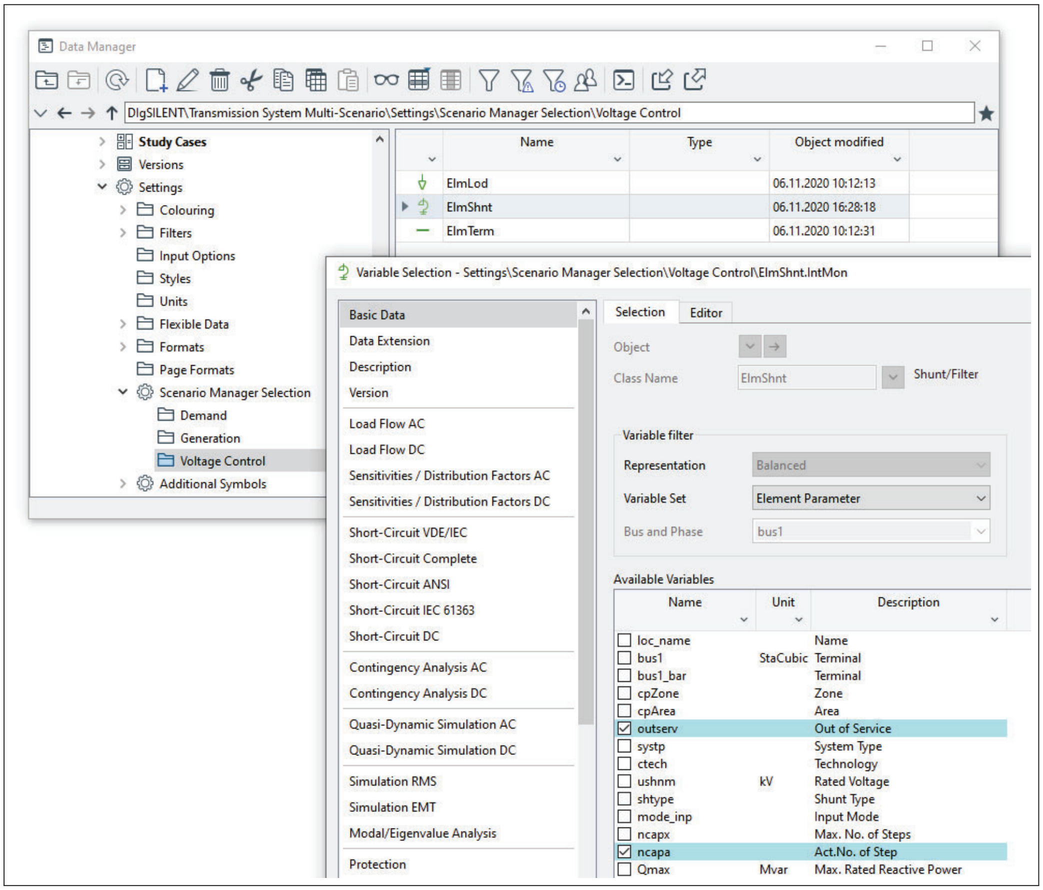


Figure 16.4.3: Variable Selection configurations for the Operation Scenario Manager

16.4.3 Viewing and editing data within the Operation Scenario Manager

When scenario data is viewed in the Operation Scenario Manager, the data is grouped into blocks, one for each data attribute selected in the configuration. Within each block, the values for each of the selected scenarios are seen. The usual sorting and filtering options are available.

In addition to being able to view the data, the user can also modify values as expected in the Network Model Manager, including copying and pasting from one column to another.

Note: It is very important for users to realise that when scenario data is changed in any scenario which is not currently active, that change will be automatically saved in the scenario. Only in the currently-active scenario will the normal option remain, to save or discard data changes. Care must be taken therefore, when making any changes to data in the Operation Scenario Manager.

16.5 Working with Operation Scenarios

In this section the operation scenario administrative tasks are explained. This includes reporting operational scenario data status, comparing operation scenarios, viewing the non-default running arrangements, applying data from one operation scenario to another (copying), updating the base network model, excluding grids from the operation scenario and creating a time based operation scenario.

16.5.1 How to view objects missing from the Operation Scenario data

When you add a component to a network, the data is not automatically captured in the active operation scenario until you save the scenario. The operation scenario appears with an asterisk next to its name in the Data Manager. If you want to get a list of all the objects that have operational data that is missing from the active scenario, then you need to print the operation scenario report. To do this, follow these steps:

1. Open the active operation scenario dialog by finding the operation scenario in the Data Manager right-clicking it and selecting edit from context sensitive menu.
2. Press the **Reporting** button. A list of objects with data missing from the operation scenario is printed by *PowerFactory* to the output window.

Note: If you double click a listed object in the output window the dialog box for that object will open directly allowing you to edit the object. You can also right click the name in the output window and use the function 'Mark in Graphic' to find the object.

16.5.2 How to compare the data in two operation scenarios

It is sometimes useful to compare data in two separate operation scenarios so that key differences can be checked. To compare two operation scenarios:

1. Deactivate all operation scenarios that you wish to compare. Only inactive operation scenarios can be compared.
2. Open the first operation scenario dialog by finding the operation scenario in the Data Manager right-clicking it and selecting edit from context sensitive menu.
3. Press the **Compare** button. A data window browser will appear.
4. Choose the second operation scenario and press **OK**. A report of the operation scenario differences is printed by *PowerFactory* to the output window.

16.5.3 How to view the non-default Running Arrangements

Any running arrangements that are assigned to substations will be stored as part of the operational data. The operation scenario has a function that allows you to view any substations with active running arrangements that are different from the default running arrangement for that substation. The default running arrangement is determined by the running arrangement that is applied to the substation when no operation scenarios are active. To view all the non-default Running Arrangements follow these steps:

1. Open the active operation scenario dialog by finding the operation scenario in the Data Manager, right-clicking it and selecting edit from context sensitive menu.
2. Press the **Reporting RA** button. *PowerFactory* prints a report of the non-default Running Arrangements to the output window.

Note: Most of these actions are also available in context-sensitive menu when right-clicking on an operation scenario (*Action* → . . .).

16.5.4 How to transfer data from one Operation Scenario to another

As explained in the chapter introduction, within each operation scenario there is a subset of operation scenario data for each grid in the network model. Therefore, there are two options when transferring data from one operation scenario to another, either copying all the operation scenario data at once, or only copying a subset of data for an individual grid. Both methods are explained within this section. Furthermore, whether operational data is to be transferred for the whole scenario or one grid only, it is also possible to be selective about which data is transferred, by setting up and using *Scenario Apply Configurations*.

16.5.4.1 Transferring operational data from one grid only

To transfer the operational data from a single grid subset to the same grid subset of another operation scenario follow these steps:

1. Activate the target operation scenario.
2. Right-click the source operation scenario subset.
3. From the context sensitive menu select *Apply*. A pop-up dialog will appear asking you if you really want to apply the selected operational data to the active operation scenario.
4. Click **OK**. The data is copied automatically by *PowerFactory*. Warning, any data saved in the equivalent subset in the active scenario will be overwritten. However, it will not be automatically saved.

16.5.4.2 Transferring operational data from a complete operation scenario

To transfer the operational data from a complete operation scenario to another operation scenario follow these steps:

1. Activate the target operation scenario.
2. Right-click the source operation scenario.
3. From the context sensitive menu select *Apply*. A pop-up dialog will appear asking you if you really want to apply the selected operational data to the active operation scenario.
4. Click **OK**. The data is copied automatically by *PowerFactory*. Warning, any data saved in the active scenario will be overwritten. However, it will not be automatically saved.

16.5.4.3 Transferring selective operational data using *Scenario Apply Configurations*

If the user wants to be selective about which data is transferred, this can be done by setting up *Scenario Apply Configurations* within the project, then using these in conjunction with the *Apply* command.

Creating *Scenario Apply Configurations*

If creating *Scenario Apply Configurations* for the first time in a project, first carry out this step in the *Settings* folder of the project:

- Click on the *Settings* folder and use the new object icon to create an object *Scenario Apply Configurations (SetOpdselection)*.

Once this is created, one or more scenario apply configuration folders may be created within it. Each will define a set of data items to be copied when that folder is selected in the *Apply* command. To create and populate each folder, follow these steps:

- First use new object icon to create a folder within the *Scenario Apply Configurations (SetOpdselection)*. Give it a meaningful name such as "Generator MW".
- Within the folder, use new object icon to create one or more *Variable Selection (IntMon)* objects. These are used to specify an element class (e.g. *ElmSym*) and which variables (e.g. *pgini*) are to be copied for this element class when this folder is selected.

Using *Scenario Apply Configurations*

Once the above configurations have been created, the folder names will appear as options when the *Apply* command is used to copy data from one scenario to another, so instead of applying all the data, the user would have option, using the example above, of just copying the generator MW set points by selecting "Generator MW".

16.5.5 How to update the default data with operation scenario data

As a user, sometimes you need to update the default operational data (the operational data parameters that exist in the network when no operation scenario is active) with operational data from an operation scenario within the project. To do this:

1. Deactivate any active operation scenario.
2. Right-click the operation scenario that you want to apply to the base model.
3. From the context sensitive menu select Apply. A pop-up dialog will appear asking you if you really want to apply the selected operational data to the base network data
4. Click **OK**. The data is copied automatically by *PowerFactory*. Warning, any data saved in the base network model will be overwritten.

16.5.6 How exclude a grid from the Operation Scenario data

Background

By default, each operation scenario contains several subsets, one for each grid in the network model. For example, you might be working with a network model with four grids, say North, South, East and West. In such a case each operation scenario would contain four subsets. Now it might be the case that you do not wish to store operational data for the 'West' grid because the models in this grid have fixed output etc. regardless of the operational state. By excluding the operational data subset for this grid, the default data can be used in all cases, even though the operational data is different in the other three grids.

How to exclude a Grid from the Operation Scenario

1. Select an operation scenario using the Data Manager.
2. Double-click the subset of the grid that you wish to exclude (you can only see the subsets in the right panel of the Data Manager). A dialog for the subset should appear.
3. Check the 'Excluded' option and the operational data from this grid will not be included within the operation scenario the next time it is saved.

16.5.7 How to create a time-based Operation Scenario

Background

By default, operation scenarios do not consider the concept of time. Therefore, when you activate a particular operation scenario, the operational parameters stored within this scenario are applied to network model regardless of the existing time point of the network model. However, sometimes it is useful to be able to assign a 'validity period' for an operation scenario, such that if the model time is outside of the validity period, then the changes stored within the operation scenario will be ignored and the network model will revert to the default parameters.

The concept of validity periods can be enabled in *PowerFactory* by using the *Scenario Scheduler*. There are two tasks required to use a 'Scenario Scheduler'. Firstly, it must be created, and secondly it must be activated. These tasks are explained below.

How to create a Scenario Scheduler

To create a Scenario Scheduler follow these steps:

1. Go to the operation scenarios' folder within your project using the Data Manager.
2. Click the *New Object* icon. A object selection window will appear.
3. From the *Element* drop down menu choose the 'Scenario Scheduler' (IntScensched).
4. Press **OK**. The scenario scheduler object dialog will appear as shown in Figure 16.5.1. Give the scheduler a name.

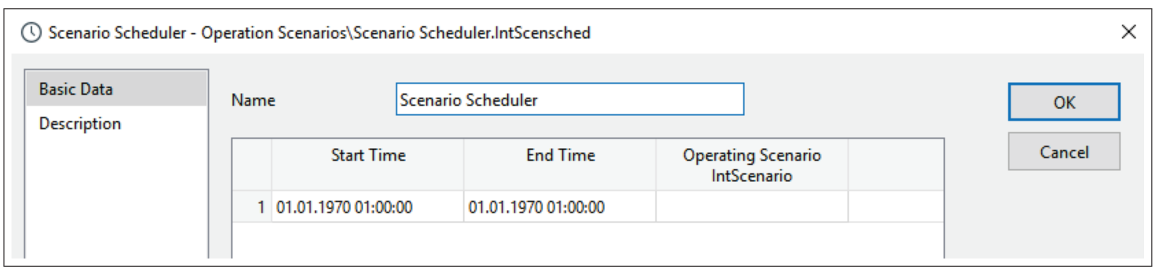


Figure 16.5.1: The Scenario Scheduler (IntScensched) dialog

5. Double-click on the first cell within the operation scenario. A scenario selection dialog will appear.
6. Choose an operation scenario to schedule.
7. Adjust the start time of the schedule by double clicking the cell within the *Start Time* column.
8. Adjust the end time of the schedule by double clicking the cell within the *End Time* column.
9. Optional: To add more scenarios to the scheduler, right-click an empty area of the scheduler and *Append Rows*. Repeat steps 5-9 to create schedules for other operation scenarios.

How to Activate a Scenario Scheduler

When first created, a scenario scheduler is not automatically activated. To activate it, follow these steps:

1. Go to the operation scenarios' folder within your project using the Data Manager.
2. Right-click the scenario scheduler object that you wish to activate and choose the option *Activate* from the context sensitive menu. The operation scenario validity periods defined within the scenario scheduler will now determine whether an operation scenario is activated automatically based on the study case time.

Note: It is possible to create more than one scenario scheduler per project. However, only one may be active. Also, if you have defined overlapping validity periods for operation scenarios within the scenario scheduler, then the operation scenario listed first (lowest row index) in the scenario scheduler will be activated and all other scenarios ignored.

16.6 Advanced Configuration of Operation Scenarios

This sub-section describes some advanced configuration options for the operation scenarios. This includes adjusting the automatic save settings and modifying the data that is stored within the operation scenarios. Note for new users, it is recommended to use the default settings.

16.6.1 How to change the automatic save settings for Operation Scenarios

As mentioned in Section 16.3.2, by default operation scenarios do not automatically save your modifications to the network data operational parameters at the time the changes are made. As a user, you can enable automatic saving of operation scenario data and you can alter the automatic save interval. It is also possible to change the save interval to 0 minutes so that all operational data changes are saved as soon as the change is made. To change the save interval for operation scenarios, follow these steps:

1. Open the *PowerFactory* User Settings by clicking the (icon on the main toolbar).
2. Select the Data Manager page.
3. In the operation scenario section of the page, enable the option *Save active Operation Scenario automatically*.
4. Change the *Save Interval* time if you would like to alter the automatic save interval from the default of 15 minutes. Setting this value to 0 minutes means that all operation scenarios will be saved automatically as soon as operational data is modified.

Note: If an operation scenario is active any changes to the network model operational parameters are stored within such an scenario. If no operation scenario is active, then the changes are stored within the network model as usual, within a 'grid' or within a 'recording expansion stage'. A changed operation scenario is marked by a "*" next to the operation scenario name in the status bar. In the Data Manager the modified operation scenario and operation scenario subset are also marked ().

16.6.2 How to modify the data stored in Operation Scenarios

Background

PowerFactory defines a default set of operational data for each object within the network model. This is the information that is stored within the operation scenarios. However, it is possible to alter the information that is stored to a limited extent by creating a *Scenario Configuration*. The procedure is divided into two tasks. Firstly, a special *Scenario Configuration* folder must be created and then the object definitions can be created within this folder.

Task 1: Creating a Scenario Configuration Folder

To create a scenario configuration folder follow these steps:

1. Go to the *Settings* folder within the project using the Data Manager.

2. Click the *New Object* icon . A object selection window will appear.
3. Choose the Scenario Configuration (*SetScenario*). A scenario configuration dialog will appear. You can rename it if you like.
4. Press **OK**.

Task 2: Defining the Operational Data Parameters

Once you have created the scenario configuration folder (task 1 above), then you can create the object definitions that determine which parameters are defined as operational data. Follow these steps:

1. Deactivate any active operation scenario.
2. Open the Scenario Configuration folder object using the Data Manager.
3. Press the **Default** button. *PowerFactory* then automatically creates the object definitions according to the defaults.
4. Open the object definition that you would like to change by double clicking it. The list of default operational data parameters is shown in the *Selected Variables* panel of the dialog box that appears.
5. You can remove an operational parameter of this object by double clicking the target parameter from the *Selected Variables* panel. Likewise, a variable can be added to this list by clicking the *black triangle* underneath the cancel button and then adding the variable name to the list of parameters.
6. Once you have altered the defined parameters, click **OK**.
7. Repeat steps 4-6 for as many objects as you would like to change.
8. Open the scenario configuration folder object again (step 2) and press the **Check** button. *PowerFactory* will notify you in the output window if your changes are accepted.

Note: Some variables cannot be removed from the default operational parameters due to internal dependencies. If you need to remove a certain variable but the *check* function doesn't allow you to, it is suggested that you contact *DIgSILENT* support to discuss alternative options.

Chapter 17

Network Variations and Expansion Stages

17.1 Introduction

As introduced in Chapter 4 (*PowerFactory* Overview), Variations and Expansion Stages are used to store changes to network data, such as parameter changes, object additions, and object deletions. This Chapter describes how to define and manage Variations, and presents an example case. The term “Variation” is used to collectively refer to Variations and Expansion Stages.

The use of Variations in *PowerFactory* facilitates the recording and tracking of data changes, independent of changes made to the base Network Model. Data changes stored in Variations can easily be activated and deactivated, and can be permanently applied to the base Network Model when required (for example, when a project is commissioned).

The concept of having a “permanent graphic” in *PowerFactory* means that graphical objects related to Variations are stored in Diagrams folders, and not within Variations. When a Variation is inactive, its graphic (if applicable) is shown on the Single Line Graphic in yellow. Turning on *Freeze Mode* ( ) hides inactive variations graphics.

When a project uses Variations, and the user wants to make changes to the base network model directly, Variations should be deactivated, or the Study Time set to be before the activation time of the first Expansion Stage (so that there is no recording Expansion Stage).

In general there are two categories of data changes stored in Variations:

1. Changes that relate to a future project (e.g. a potential or committed project). The changes may be stored in a Variation to be included with the Network Model at a particular date, or manually activated and deactivated as required by the user.
2. Changes that relate to data corrections or additions based on the current (physical) network. The changes may be stored in a Variation in order to assess the model with and without the changes, to track changes made to the model, and to facilitate reversion to the original model in case the changes are to be revised.

Notes regarding Variations and Expansion Stages:

- General:
  - The user may define as many Variations and Expansion Stages as required.
  - Variations and Expansion Stages cannot be deleted when active.
  - Variations may also be used to record operational data changes, when there is no active Operation Scenario.

    - Expansion Stages are by default sorted according to their activation time in ascending order.
    - To quickly show the recording Expansion Stage, project name, active Operation Scenario, and Study Case, hover the mouse pointer over the bottom right corner of the *PowerFactory* window, where (by default) the project name is shown. To change this to display the recording Expansion Stage, choose *Display Options* → *'Recording' Expansion stage*.

- Activating and deactivating Variations:
    - Active Variations and Expansion Stages are shown with blue icons in the Data Manager and Project Overview.
    - The Activation Time of Expansion Stages can only be modified when the parent Variation is inactive.
    - To activate or deactivate single or multiple Variations in the Data Manager, navigate to the "Variations" folder, select and right-click on the Variation(s) and choose to activate or deactivate the selected Variation(s).
    - In the active Study Case, the "Variation Configuration" object stores the status of project Variations. It is automatically updated as Variations are activated and deactivated.

- Recording changes:
    - Whenever network data changes are recorded, an information box will appear for 10 seconds. This tells the user where (in which Variation and Expansion Stage) the changes are being recorded.
    - Elements in *PowerFactory* generally include references to Type data. Changes to Type data <u>are not</u> recorded in Expansion Stages. However, changes to Element Type references <u>are</u> recorded.
    - When there are multiple active Expansion Stages, only the 'Recording' Expansion Stage stores changes to Network Data (shown with a red circle icon and bold text). There can be only one recording Expansion Stage per study case.
    - With the exception of objects added in the active 'Recording' Expansion Stage, objects (e.g. Terminals in the base network model) cannot be renamed while there is a 'Recording' Expansion Stage.

- DPL:
    - Deleted objects are moved to the *PowerFactory* Recycle Bin, they are not completely deleted until the Recycle Bin is emptied. If a DPL script is used to create an Expansion Stage, and Expansion Stage objects are subsequently deleted, re-running the DPL script may first require the deleting of the Expansion Stage objects from the Recycle Bin. This is to avoid issues with references to objects stored in the Recycle Bin.

17.2 Variations

To define a new Variation (*IntScheme*):

1. First, either:
    - From the Main Menu, select *Insert* → *Variation*.
    - In a Data Manager, right-click on the *Variations* folder ( ) and from the context-sensitive menu select *New* → *Variation*.
    - In a Data Manager, select the *Variations* folder and click on the *New Object* icon . Ensure that the *Element* field is set to *Variation (IntScheme)*, and press **<u>O</u>k**.
2. Define the Variation *Name*.
3. Optionally set the Variation *Colour*. This is used to highlight modifications introduced by the Variation in the Single Line Graphic.

4. On the second page of the *Basic Data* tab, optionally select *Restricted Validity Period* for the Variation. If a restricted validity period is defined, the variation will effectively be ignored outside this time range. This option is not normally used, as the same effect can be achieved by having an expansion stage which changes the network and a second which restores the original state.

The “starting” and “completed” *Activation Time* are set automatically according to the Expansion Stages stored inside the Variation. The “starting” time is the activation time of the earliest Expansion Stage, and the “completed” time is the activation time of the latest Expansion Stage. If no Expansion Stages are defined, the activation time is set by default to 01.01.1970.

To activate a previously defined Variation, in the Data Manager, right-click on the Variation and from the context-sensitive menu select *Activate*. The Variation and associated Expansion Stages will be activated based on their activation times and the current study case time.

In the Variation dialog, the **Contents** button can be used to list the Expansion Stages stored within the Variation.

17.3 Expansion Stages

To define a new Expansion Stage (*IntSstage*):

1. First, either:
   - Right-click on the target Variation and select *New* → *Expansion Stage*.
   - Select the target Variation and click on the *New Object* button in the Data Manager's icon bar. Set the 'Element' field to *Expansion Stage (IntStage)* and press **Ok**.
2. Define the Expansion Stage *Name*.
3. Set the Expansion Stage Activation Time.
4. Optionally select to *Exclude from Activation* to put the Expansion Stage out of service.
5. Optionally enter Economical Data on the *Economical Data* page (see Section 43.2 (Techno-Economical Calculation) for details).
6. Press **OK**.
7. Select whether or not to set the current Study Time to the Activation Time of the defined Expansion Stage. See Section 17.5 for details.

From the Expansion Stage dialog, the following buttons are available:

- Press **Contents** to view changes introduced by the Expansion Stage.
- Press **Split** to assign changes from the recording Expansion Stage to a target (see Section 17.9.3).
- Press **Apply** to apply the changes of an Expansion Stage (only available if the parent Variation is inactive). Changes are applied to the *Network Model*, or to the recording Expansion Stage (see Section 17.9.1).

17.4 The Study Time

The study case Study Time determines which Expansion Stages are active. If the Study Time is equal to or exceeds the activation time of an Expansion Stage, it will be active (provided that the parent Variation is active, and provided that “Exclude from Activation” is not selected in the Expansion Stage or an active Variation Scheduler). The Study Time can be accessed from:

- The *Date/Time of Calculation Case* icon .

- Clicking on the lower right corner of the *PowerFactory* window, where the time of the active Study Case is displayed.
- The *Main Menu* under *Edit* → *Project Data*→ *Date/Time of Study Case*, or *Edit* → *Project Data*→ *Study Case* and then use the drop-down arrow to open up a standard calendar dialog.
- The Data Manager in the active Study Case folder, object “Set Study Time”.

17.5 The Recording Expansion Stage

When a Variation is activated for a study case, the active Expansion Stage with the latest activation time is automatically set to the recording Expansion Stage. If there are multiple Expansion Stages with this same activation time, the stage that previously set to the recording stage will remain as the recording Expansion Stage. Changes made to the network data by the user are saved to this stage.

As changes are made, an information box will appear for 10 seconds. This tells the user where (in which Variation and Expansion Stage) the changes are being recorded.

As discussed previously, the Study Time can be changed in order to set the active Expansion Stages, and as a consequence, set the “recording Expansion Stage”. To simplify selection of the recording Expansion Stage, in the Data Manager it is possible right-click an Expansion Stage, and select *Set ’Recording’ Expansion stage* to quickly modify the Study Time to set a particular Expansion Stage as the recording Expansion Stage.

As noted in 17.1, unless an Operation Scenario is active, changes made to operational data are stored in the recording Expansion Stage.

17.6 The Variation Scheduler

As an alternative to setting the activation time of Expansion Stages individually, *Variation Schedulers* (*IntSscheduler*) may be used to manage the activation times and service status of each Expansion Stage stored within a Variation. Multiple Variation Schedulers can be defined within a particular Variation, but only one may be active at a time. If there is no active Variation Scheduler, the Expansion Stage activation times will revert to the times specified within each individual Expansion Stage.

To define a Variation Scheduler:

1. Open a Data Manager, and navigate to the Variation where the Scheduler is to be defined. Then, either:
   - Right-click on the Variation and select *New* → *Variation Scheduler*.
   - Click on the *New Object* button and select *Variation Scheduler* (***IntScheduler***).
2. Press the **Contents** button to open a data browser listing the included stages with their activation times and service statuses, and modify as required.

The activation time and status of Expansion Stages referred to be a Variation Scheduler can only be changed when the Variation is active, and the Variation Scheduler is inactive. Note that Expansion Stage references are automatically updated in the scheduler.

Note: If the parent Variation is deactivated and reactivated, the Variation Scheduler must be re-activated by the user, if required.

17.7 Variation and Expansion Stage Example

Figure 17.7.1 shows an example project where there are two Variations, “New Connection” and “New Line”.

With the current study case time, some expansion stages are active and some are not. Active stages are indicated by their icon being coloured blue. The recording stage is also indicated:

- Expansion Stage “Ld1” is shown with a blue icon, so is active. But it also has its name shown in bold, and this, together with the marker against the icon, shows that it is the recording Expansion Stage.
- Expansion Stage “Ld2” has no colouring; it is inactive.
- Expansion Stage “Line and T2” is shown with a blue icon; it is active.

The Variation Scheduler “Scheduler1” within the “New Connection” Variation, shown with a blue icon and bold text, is active.

Therefore, the activation time and service status of each Expansion Stage within the Variation “New Connection” is determined from the activation times specified in this Variation Scheduler. The alternative Variation Scheduler “Scheduler2” is inactive (only one Variation Scheduler can be active at a time).

Also shown in Figure 17.7.1 on the right-side pane are the modifications associated with Expansion Stage “Ld1”. In this stage, a load and an associated switch and cubicle has been added.

Note that graphical changes are not included in the Variation.

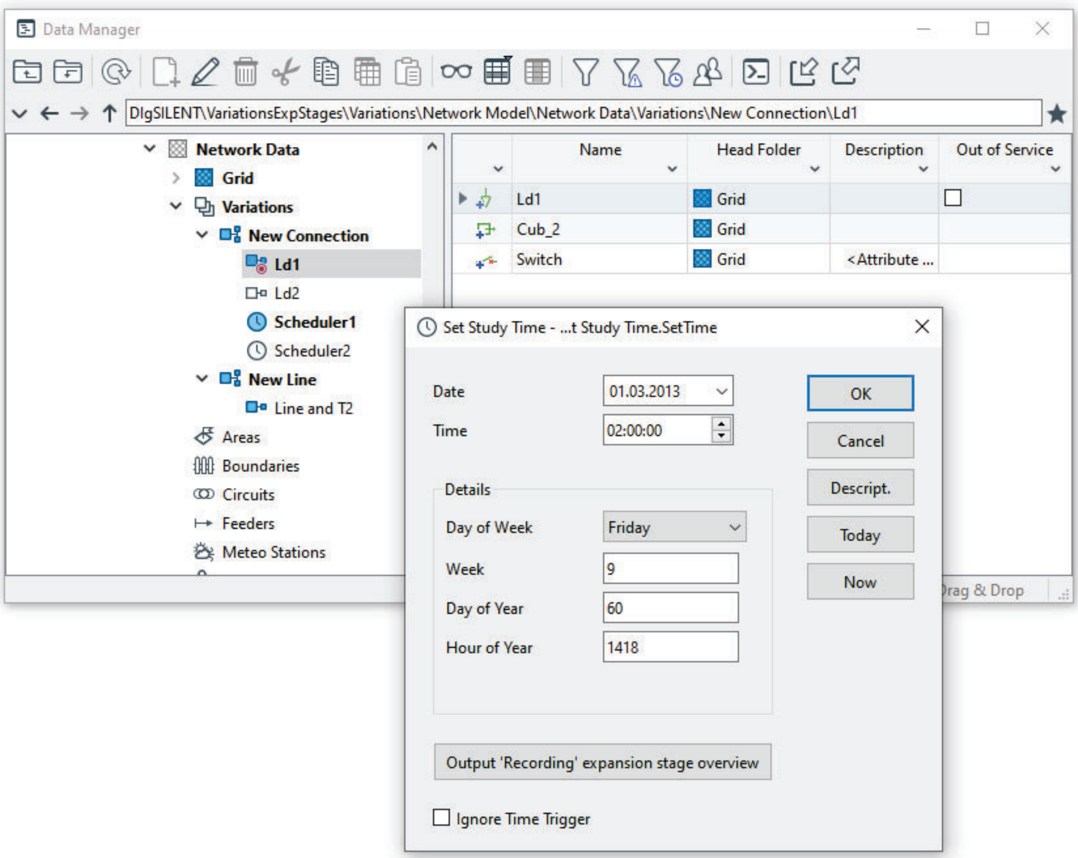


Figure 17.7.1: Example Variations and Expansion Stages - Data Manager

Figure 17.7.2 shows the Single Line Graphic of the associated network. Since the Expansion Stage "Ld2" is inactive, the Load "Ld2" is shown in yellow.

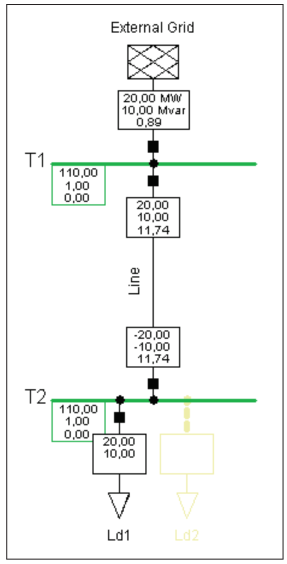


Figure 17.7.2: Example Variations and Expansion Stages - Single Line Graphic

17.8 The Variation Manager

The Variation Manager is a tool which enables users to have an overview of their Network Variations and Expansion Stages in a Gantt-chart like format. It can be accessed from the main tool bar, using the icon. In Figure 17.8.1 below, the Variation Manager is used to show several Variations in a project. In this project, the Network Variations have been arranged into subfolders, according to the year. Each Variation is represented by a blue horizontal bar in the chart, and each Expansion Stage by a short vertical bar. The sections following describe the options and features available in the tool, with reference to this example.

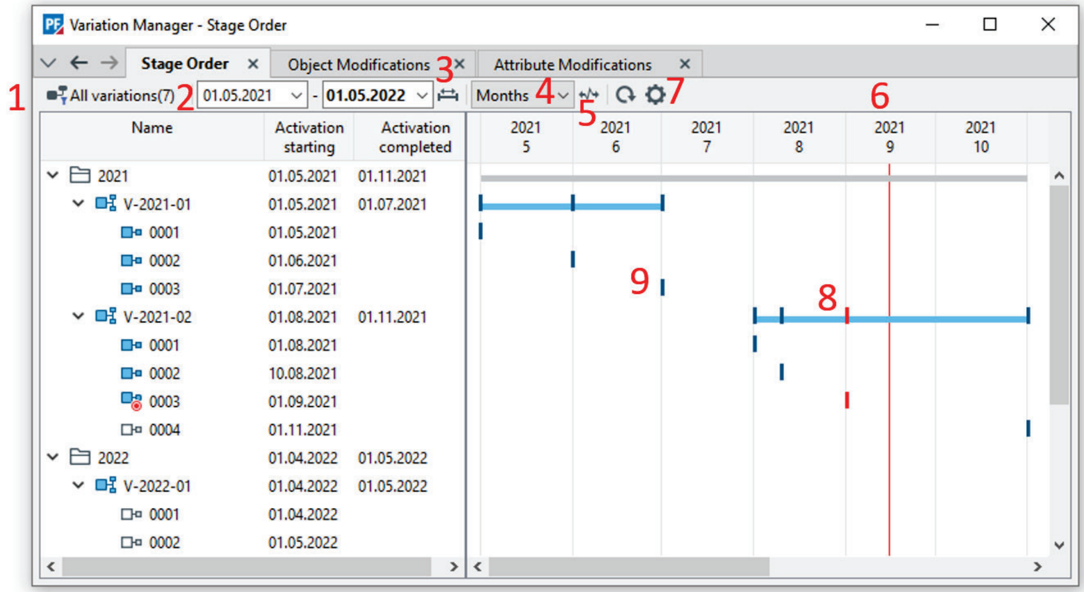


Figure 17.8.1: Variation Manager, Stage Order tab

17.8.1 Stage Order

17.8.1.1 Selecting the Variations to be shown

By default, all the Variations will be shown in the Variation Manager, but the user can restrict what is shown by clicking on the icon (1). Then, Variations and/or folders containing Variations can be selected as required. Several options for selecting Variations are offered, including an option to select Variations relating to a particular network element. The Variations are listed on the left-hand side, with two columns (“Activation starting” and “Activation completed”) to indicate when each Variation and Expansion Stage is active. If the project contains Variation Schedulers, another column will appear to indicate which Variations include a Variation Scheduler.

17.8.1.2 Timeframe and resolution

By default, the period of time covered in the Variation Manager view will be set automatically according to the first and last Expansion Stage activation time of the selected Variations. This time period can be adjusted using the drop-down calendar from the start and end date fields (2). If the time period is restricted by the user such that not all selected Variations can be displayed in full, this is indicated by the start and/or date being highlighted in bold, and by arrows within the chart indicating that not everything is displayed.

If the time period has been restricted, the full time period is easily restored using the icon (3).

In addition to managing the time period shown, the user is able to select the time resolution from a drop-down menu (4).

The “Fit time scaling into window” icon (5) will adjust the view so that everything can be seen without the need for a scroll-bar (note that this can result in the resolution being changed). There is also a “Refresh” icon. To further adjust the view, the user can change the column widths manually (in the same way as in a spreadsheet).

17.8.1.3 Study case time and recording stage

If the current study case time is within the period shown in the Variation Manager, it will be shown as a red line (6). By double-clicking on this line, the user can change the study case time, which may in turn affect which Expansion Stage is the recording stage. Note that the current recording stage is indicated on the left-hand side of the view, by the symbol , and highlighted in red in the chart.

17.8.1.4 Looking at the Variations in detail

On the right-hand side of the Variation Manager, each blue horizontal bar (8) represents a network Variation, which can be edited by double-clicking on the bar. Likewise, each short vertical bar (9) is linked to an Expansion Stage, and these are also editable in order to see the contents.

The amount of information shown on the left-hand side of the Variation Manager can be controlled by the user, via the Settings icon (7), which also offers options for the format of the times and dates shown.

The Variations on the left-hand side can be expanded to show the individual Expansion Stages (as in Figure 17.8.1) or not, as required.

Using right-click on the Variations or Expansion Stages on the left-hand side brings up a context menu. As well as other expected options, there are two options specific to the Variation Manager:

- When only a portion of the chart is visible, **Show in Chart** can be used to scroll the chart automatically so as to see the selected Variation or Expansion Stage.
- **Include in displayed time period** can be used on Variations or Expansion Stages which are not shown because they lie outside the currently-specified time period. The displayed time period will be automatically expanded so that they can be seen.
- Selecting **Show Object Modifications** for a Variation will open up an Object Modifications tab (see 17.8.2) for that particular Variation.

17.8.2 Object Modifications

The Object Modifications tab provides detail of the modifications recorded in the selected Variation(s), at an object level. Most of the options at the top are the same as described above in section 17.8.1 for the Stage Order tab.

As well as the button for selecting Variations, there is a button for selecting objects (the default being "All Objects"). This button will bring up a dialog box for the selection of objects to be viewed. There is a column of checkboxes for selecting objects, and column showing the element class, which can be filtered. Objects can be selected/deselected individually, but also these four options may be used:

- **Select All**
- **Select Visible**: Elements that are visible (depending on the filter applied to the element class column) are added to the selection.
- **Deselect Visible**: Elements that are visible (depending on the filter applied to the element class column) are removed from the selection.
- **Deselect All**

In addition, the icon allows the user to toggle between displaying the objects in a list or in a hierarchical view.

On the left-hand side, a number of columns provide information about the object modifications listed. Any of these columns can be used to filter the data: for example, filtering on the Conflicts column provides an easy way of checking for potential errors in the use of Variations in the project.

- **Number of references (#)**: The number of times the object is referenced in the selected Variations
- **Existing in base grid (B)**: Shows whether the object already exists in the underlying grid or is only added in a Variation
- **Added (+)**: Indicates that the object is added in a Variation
- **Modified (*)**: Indicates that the object is modified in a Variation
- **Deleted (x)**: Indicates that the object is deleted in a Variation
- **Scheduler(s)**: If Variation Schedulers are present in the project, this column will be visible, to indicate affected objects
- **Conflicts (!)**: Possible conflicts are identified. For example, modifiying a non-existent object will flag up an error, whereas deleting a non-existent object or adding an already-existing object will flag up a warning. It should be noted that the conflict identification only takes account of the selected Variations.

Right-clicking on any of the objects brings up a context menu, with a number of options for viewing data related to this object. In particular, the option *Show Attribute Modifications* enables the user to examine in detail the changes made to the attributes of the selected object. This option will bring up a new Attribute Modifications tab, as described in the next section.

The Gantt chart on the right-hand side is similar to that for the Variations, described in 17.8.1.4 above. It shows information about the creation, modification and deletion of network elements, and each element can be accessed directly by double-clicking on the blue horizontal bar.

17.8.3 Attribute Modifications

The Attribute Modifications tab does not appear by default when the Variation Manager is first opened, but is generated for an object using the *Show Attribute Modifications* context menu in the Object Modifications tab, or from the a Data Manager or Network Model Manager.

17.8.3.1 Viewing attribute modifications

This view shows a history for all the attributes which are modified in any of the selected Variations. Each column shows the Variation and Expansion Stage in which a change occurs, the changed values being shown in bold. The activation time of the Expansion Stage is also shown in the column heading, and the tool tip for this activation time also considers and lists schedulers and the times with and without scheduler.

It might be that some attribute changes are of little interest to the user. The Settings dialog, accessed via the Settings icon, includes an *Ignored attributes* button, which allows the user to specifiy (by element class) attributes which should not be considered relevant by the Variation Manager.

17.8.3.2 Editing attribute modifications

As well as looking at the attributes that have been modified via the Network Variations, the user is also able to make changes directly within the expansion stages, by editing the data in the table. This can be used, for example, to make corrections if an erroneous value is spotted. The edits will directly change the relevant Expansion Stage. It is not possible, however, to change attributes which are not already modified within any of the selected/listed Variation(s).

As well as the option to edit the data in the table directly, right-clicking on a value brings up a context menu for editing, including options to apply the selected value to all the relevant Expansion Stages, or a selection of them.

17.9 Variation and Expansion Stage Management

17.9.1 Applying Changes from Expansion Stages

Changes stored in non-active Expansion Stages can be applied to the *Network Data* folder, or if there is an active recording Expansion Stage, to the recording Expansion Stage. To apply the changes, either:

- In the Data Manager, right-click the Expansion Stage and select *Apply Changes*, or in the Expansion Stage dialog press **Apply** (only available if the Expansion Stage is within a non-active Variation).
- In the Data Manager, select item(s) within an inactive Expansion Stage, right-click and select *Apply Changes*. If required, delete the item(s) from the original Expansion Stage.

17.9.2 Consolidating Variations

Changes that are recorded in a projects active Variations can be permanently applied to the *Network Data* folder by means of the *Consolidation* function. After the consolidation process is carried out, the active (consolidated) Expansion Stages are deleted, as well as any empty active Variations.

To consolidate an active Variation(s):

1. Right-click on the active study case and from the context-sensitive menu select *Consolidate Network Variation*.
2. A confirmation message listing the Variations to be consolidated is displayed. Press **Yes** to implement the changes.
3. View the list of consolidated Variations and Expansion Stages in the Output Window

---

Note: Variations stored within the Operational Library must be consolidated in separate actions. To consolidate a Variation stored in the Operational Library, right-click and from the context-sensitive menu select *Consolidate*.

17.9.3 Splitting Expansion Stages

Changes stored in the recording Expansion Stage can be split into different Expansion Stages within the same Variation using the Merge Tool.

To split an Expansion Stage:

1. Open the dialog of the recording Expansion Stage and press **Split**. Alternatively, right-click and from the context-sensitive menu select *Split*.
2. A data browser listing the other Expansion Stages from the parent Variation is displayed. Double-click on the target Expansion Stage.
3. The Merge Tool window is displayed, listing all the changes from the compared Expansion Stages. Select the changes to be moved to the “Target” stage by double-clicking on the *Assigned from* cell of each row and selecting *Move* or *Ignore*. Alternatively, double-click the icon shown in the “Target” or “Source” cell of each row.

4. Press **Split**. All the changes marked as *Move* will be moved to the target Expansion Stage, and the changes marked as *Ignore* will remain in the original “Base” stage. Once completed, the Variation is automatically deactivated.

17.9.4 Comparing Variations and Expansion Stages

Variations and Expansion Stages can be compared, as can any other kind of object in *PowerFactory*, using the Merge Tool. To compare objects using the Merge Tool, a “base object” and an “object to compare” must be selected. The comparison results are presented in a data browser window, which facilitates the visualisation, sorting, and possible merging of the compared objects. Comparison results symbols indicate the differences between each listed object, defined as follows:

- − The object exists in the “base object” but not in the “object to compare”.
- + The object exists in the “object to compare” but not in the “base object”.
- △ The object exists in both sets but the parameters’ values differ.
- = The object exists in both sets and has identical parameter values.

To compare two Variations:

1. In an active project, right-click on a non-active Variation and from the context-sensitive menu select *Select as Base to Compare*.
2. Right-click on the (inactive) Variation to compare and from the context-sensitive menu select *Compare to “Name of the base object”*.
3. The Merge Tool dialog (*ComMerge*) is displayed. By default, all of the contained elements are compared. The *Compare* fields can be configured however, to compare only the objects or selected subfolders.
4. Once the *Compare* options are set, press the **Execute** button.
5. When prompted, select **Yes** to deactivate the project and perform the comparison.

Figure 17.9.1 shows an example comparison of two Variations (based on the example presented in Section 17.7), where the Variation “New Line” is set as the “Base” for comparison. The “Assigned from” options are set such that all Expansion Stages from both “New Line” and “New Connection” Variations will be merged into a single Variation, which will retain the name of the “Base” Variation “New Line”.

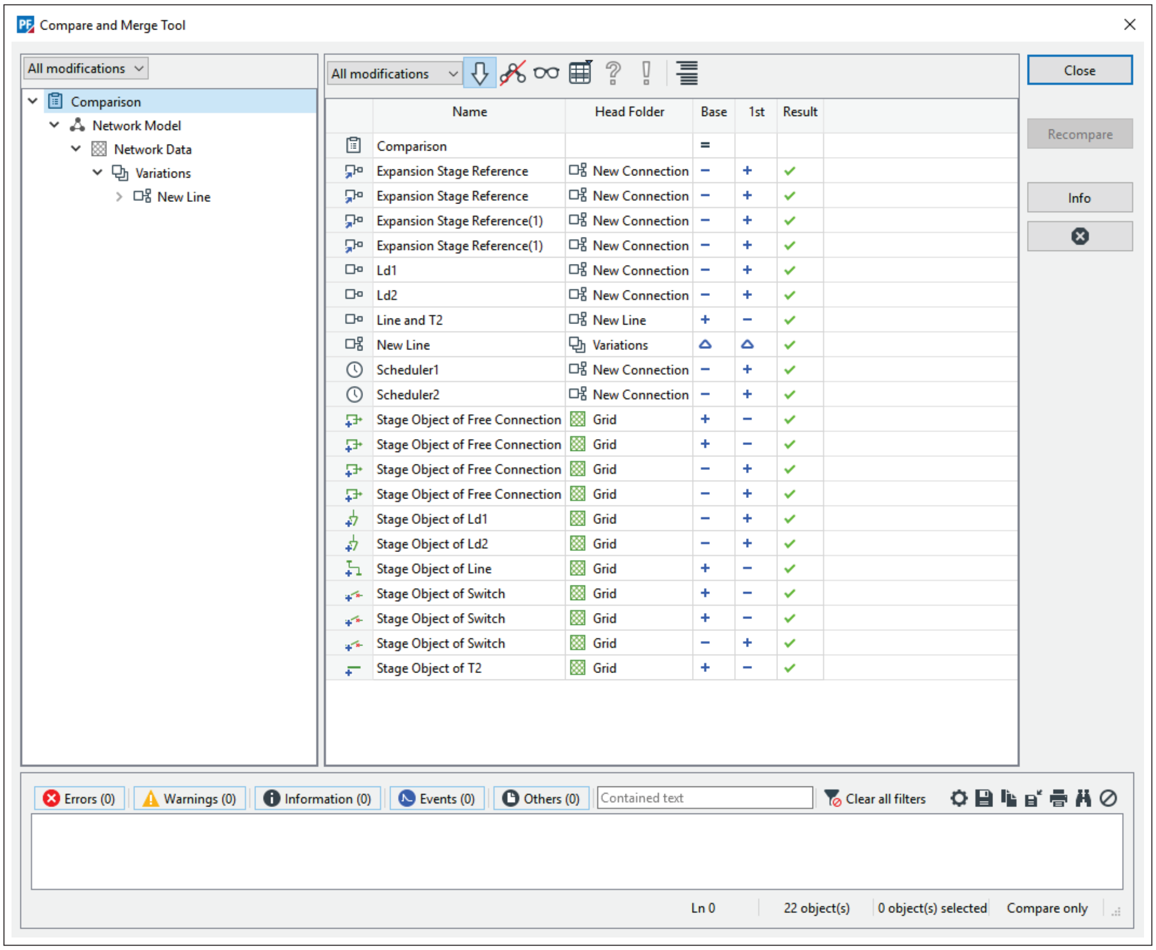


Figure 17.9.1: Merge Tool Window

Refer to Chapter 21: Data Management, Section 21.4 (Comparing and Merging Projects) for further details on use of the Merge Tool.

17.9.5 Colouring Variations the Single Line Graphic

The single-line graphic colouring function offers three modes which may be used to identify changes from Variations and Expansion Stages. To set the colouring mode, go to *Diagram Colouring*, and under *Other* select *Variations / System Stages*, and the desired mode from the following:

- *Modifications in Recording Expansion Stage*. Colours can be defined for *Modified*, *Added*, and *Touched but not modified* components.
- *Modifications in Variations / System Stages*. Objects are shown in the colour of the Variation in which the object is last added or modified.
- *Original Locations*. Objects are shown in the colour of the grid or the Variation in which the object is added.

The use of colouring in network diagrams is described in Section 9.3.7.1, and more detail about the use of colours and colour palettes can be found in Section 4.7.5.

17.9.6 Variation Conflicts

Active Expansion Stages with the same activation time must be independent. This means that the same object can not be changed (modified, deleted, or added) in active Expansion Stages with the same activation times. If there are dependent Expansion Stages, when the Variation is activated *PowerFactory* will display an error message to the Output Window and the activation process will be cancelled. Other conflicts that may arise during the activation of a Variation:

- The same object is added by more than one Expansion Stage. In this case the latest addition is applied and a warning message is displayed in the Output Window.
- A previously deleted object is deleted. In this case the deletion is ignored and a warning message is displayed in the Output Window.
- An object is changed or deleted in a Expansion Stage but it does not exist. In this case the change is ignored and a warning message is displayed in the Output Window.
- A deleted object is changed in a Expansion Stage. In this case the change is applied to the deleted target object and a warning message is displayed in the Output Window.

17.9.7 Error Correction Mode

As well as recording the addition and removal of database objects, variations also record changes to database objects. Human error or the emergence of new information can result in a need to update a change. Suppose that at some time after the change has been introduced, the user wishes to update the change. If additional variations have been created since the change was introduced, this will be hard to achieve. The user must first remember in which Expansion Stage the change was introduced, then they must make this Expansion Stage the Recording Stage before finally updating the change or rectifying the error. The Error Correction mode is intended to simplify this procedure. The following example illustrates use of the Error Correction Mode.

Suppose that a project is planned consisting of a base case and 2 Variations, namely Variation 1 and Variation 2. Suppose that the base case network contains a line object (*ElmLne*) of length 1km. When Variation 1 is recorded, the length of the line is updated from the base case value to a new value of 10km. This change is recorded in the Expansion Stage associated with Variation 1. Subsequently, the user creates Variation 2 and records a new set of changes in the Expansion Stage of Variation 2. The user makes no changes to the line object in Variation 2, but suddenly realises that the length of the line is incorrect. The length should be 15km not 10km. If the user makes a change to the line length while Variation 2 is recording this change will be recorded and applied while Variation 2 is activated. However, as soon as Variation 2 is deactivated, providing Variation 1 is activated, the line length will return to the 10km value. This is incorrect and the error is therefore still present in the project.

Instead of recording the change in the Recording Expansion Stage of Variation 2, the user should turn on the Error Correction Mode. This can be achieved by first ensuring that the Project Overview Window is visible. (If not, select *Window* → *Show Project Overview Window*). Then, by Right clicking in the Project Overview Window on the title line of the Network Variations section. A contextual menu as illustrated in Figure 17.9.2 will appear. The option Error Correction Mode should be selected from the contextual menu.

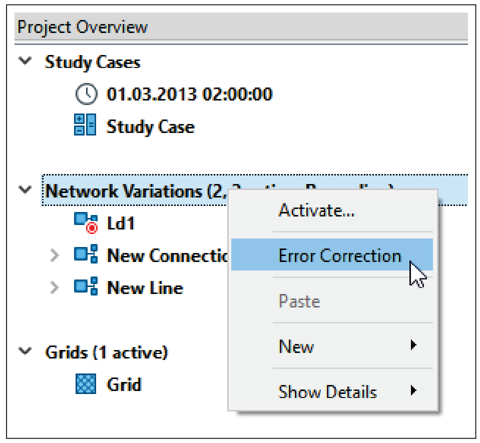


Figure 17.9.2: Activating Error Correction Mode

Once the Error Correction Mode has been switched on, any changes introduced will now, not automatically be stored in the Recording Expansion Stage. Instead, they will be stored in the Expansion Stage containing the record of the last change to the object in question. For the example described, this will be in the Expansion Stage associated with Variation 1, where the length was updated from 1km to 10km. The 10km value will be updated to 15km. If the Error Correction Mode is now switched off, again by right clicking in the Project Overview Window, the user can proceed knowing that the error has been eliminated from the project.

Please note, if any change to the line had been recorded during Variation 2 prior to the application of the Error Correction Mode, not necessarily a change to the length of the line, but a change to any *ElmLne* parameter, then with Error Correction Mode active, the change would be recorded in the Recording Expansion Stage of Variation 2. This is because the Expansion Stage containing the record of the last change to the object in question would in fact be the one in Variation 2. In this case, the error would still be present in the project.

17.10 Compatibility with Previous *PowerFactory* Releases

17.10.1 General

Prior to *PowerFactory* v14, “System Stages” where used to analyse design alternatives as well as different operating conditions. They recorded model changes (addition/removal of equipment, topology changes, etc.), operational changes (switch positions, tap positions, generator dispatch, etc.), and graphical changes. Since version 14.0, the System Stage definition has been replaced by Variations and Operation Scenarios, which provides enhanced flexibility and transparency.

When importing (and then activating) a project that was implemented in a previous *PowerFactory* version, the activation process will automatically make a copy of the project, rename it (by appending _v14 or _v15 to the project name) and migrate the structure of the copied project.

The migration process creates new *Project Folders* (such as Network Data, Study Cases, Library folders, etc.) and moves the corresponding information to these project folders. Additionally, existing *Stations* and *Line Routes* elements are migrated to their corresponding definition in v14 and v15 (i.e. Substations and Branches).

If the project contains *System Stages*, they **will not be converted automatically**. They will be still be defined, and functions related to their handling will still be available. If the user wishes to take full advantage of the *Variation* and *Operational Scenario* concepts, then the System Stages must be converted manually. The procedure is described in the following section.

17.10.2 Converting System Stages

The conversion process of System Stages is described with reference to an example project opened in *PowerFactory* v14, with the structure shown in Figure 17.10.1. The project contains three grids "Grid 110 kV", "Grid 220 kV" and "Grid 33 kV". Each Grid contains a "2010 Base Case" System Stage with three System Stages "2010 MAX", "2010 MIN", and "2011 Base Case". The "2011 Base Case" stage in-turn contains two stages, "2011 MAX" and "2011 MIN". The Study Cases are configured so that the "2011 MAX" Study Case and the "2011 MAX" stages are active.

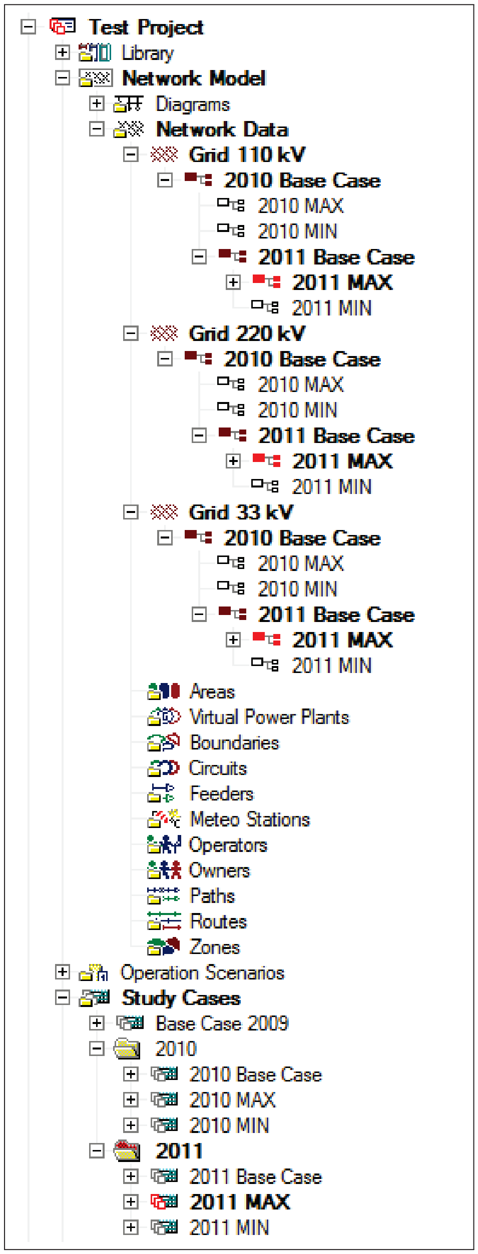


Figure 17.10.1: Example Project - System Stage Structure

To convert the System Stages to Variations / Operation Scenarios:

1. Activate the *Study Case* that uses the base grids (in this example "Base Case 2009"), so that no System Stage is active.

2. Create a *Variations* folder inside the *Network Data* folder by opening the *Data Manager* window and from the left pane select the *Network Data* folder (located inside the *Network Model* folder), right-click and select *New* → *Project Folder*. In the dialog window that appears, type in a name (for example “Variations”) and select “Variations” as the folder type. Press **OK**.

3. Define a *Variation* inside the Variations folder. From the *Data Manager* window select the Variations folder, right-click and select *New* → *Variation*. In the dialog window that appears, type in a name (for example “2010”). Press **OK**, and select **Yes** to activate the new Variation.

4. The Expansion Stage dialog will be displayed. Type in a name and set the activation time as appropriate (in this case, it is set to 01.01.2010). Press **OK**, and select **Yes** to set the stage as recording. After this step, the Variation should be active and the Expansion Stage be recording.

5. From the Data Manager, select a Study Case that uses System Stages (it should not be active), right-click and select *Reduce Revision*. This will copy both network data and operational data from the System Stages used by the study case into the recording Expansion Stage, and will delete the System Stages (to copy operational data to an Operation Scenario, an Operation Scenario must be active at this step). In this example, the “2010 Base Case” is reduced, followed by the “2011 Base Case” - this is because the complete System Stage branch, containing all System Stages between the selected stage and the target folder are reduced. Figure 17.10.2 shows the result of reducing the “2010 Base Case” and “2011 Base Case” to Variations.

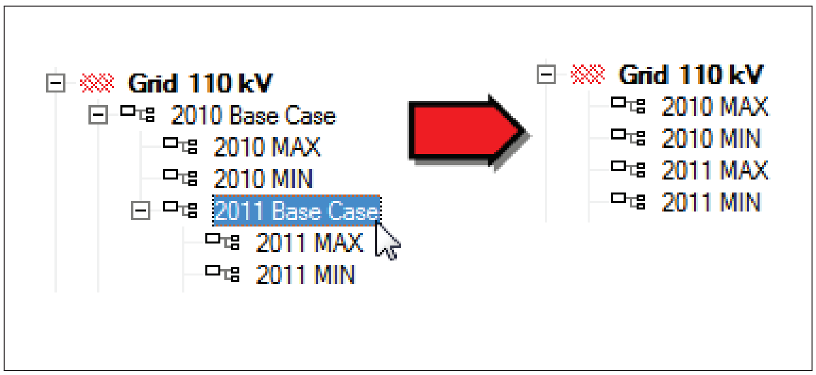


Figure 17.10.2: Reduce Revision performed for the 2011 Base Case

6. After converting System Stages “2010 Base Case” and “2011 Base Case” (with Network Data modifications) to Variations, and System Stages “2010 MAX”, “2010 MIN”, “2011 MAX”, and “2011 MIN” (with operational modifications) to Operation Scenarios, the Variations and Operation Scenarios are assigned to Study Cases. Figure 17.10.3 shows the resulting project structure for this example, where all System Stages have been converted to Variations and Operation Scenarios.

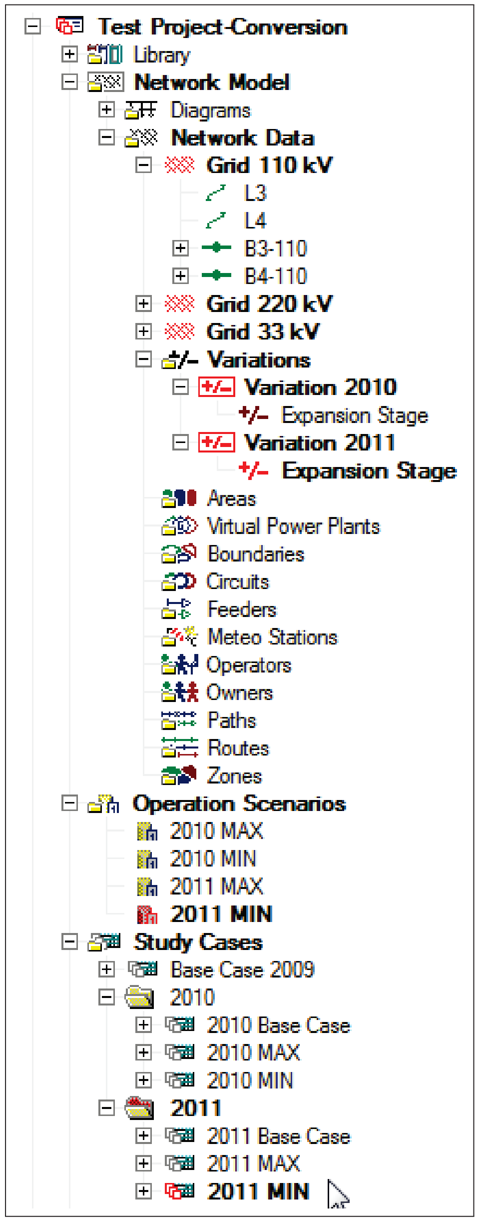


Figure 17.10.3: Resulting Project Structure

Chapter 18

Parameter Characteristics, Load States, and Tariffs

18.1 Introduction

This chapter provides details on how to define and use characteristics, load states, load distribution states, and tariffs.

It is useful to be aware that when element parameters have characteristics applied to them, they appear differently coloured in both the element dialogs and in a network model manager. By default, the colouring for characteristics is lilac (pale purple). Note that both the colour and its priority can be changed in the User Settings (see Section 7.8).

18.2 Parameter Characteristics

General Background

In *PowerFactory* any parameter may be assigned a range of values (known as a Characteristic) that is then selectable by date and time, or by a user-defined trigger. The range of values may be in the form of a scaling factor, a one-dimensional vector or a two-dimensional matrix, such as where:

- Load demand varies based on the minute, day, season, or year of the study case.
- Generator operating point varies based on the study being conducted.
- Line/transformer ratings, generator maximum power output, etc. vary with ambient temperature.
- Wind farm output varies with wind speed, or solar farm output varies with irradiance.

The assignment of a characteristic may be made either individually to a parameter or to a number of parameters. New characteristics are normally defined in either:

- The *Characteristics* folder of the *Operational Library*.
- The Global *Characteristics* folder within *Database* → *Library*.

Studies which utilise characteristics are known as 'parametric studies'.

Scales and Triggers

The value of the characteristic is defined by the value of the scale. New scales are normally defined in the *Scales* folder within the *Characteristics* folder in the *Operational Library*.

When a scale is created, a means to 'set' the scale, and thereby to set the parameter to the corresponding value, is required. This is called a trigger (*SetTrigger*,). After a new scale has been defined, a trigger is automatically created in the active study case folder (see also Chapter 13, Section 13.12: Triggers). When a trigger is edited and a 'current' value is set the scale is set and the parameter value is changed. When a different study case is activated, or a new study case is created, and a load-flow is performed, all relevant triggers are copied into the study case folder and may be used in the new study case. Triggers for characteristics may be created at any time in the Data Manager within the *Library* → *Operational Library*→ *Characteristics*→ *Scale* folder, or at the time the Characteristic is created. Triggers for characteristic can generally be accessed from either:

- The Date/Time of Study Case icon ().
- The Trigger of Study Case icon ().

Figure 18.2.1 illustrates an application of scales and triggers, where the study case time is used to set the output of a load based on the hour of the day.

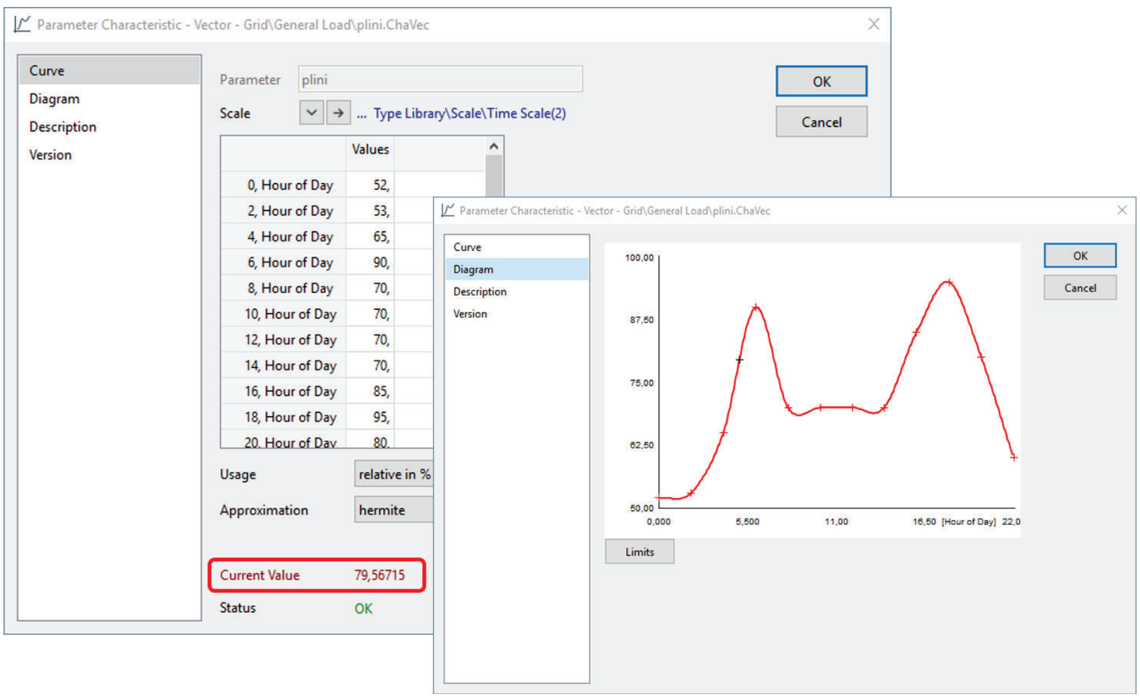


Figure 18.2.1: Illustration of Scales and Triggers

Available Characteristics

Table 18.2.1 shows a summary of the Parameter Characteristics available in *PowerFactory*. Note: Click on Characteristic description to link to the relevant section.

Characteristic	Description of Application
18.2.1: Time Characteristics	Parameter(s) are to be modified based on the day, week, or month set in the Study Time. Parameter states may be interpolated between entered values.
18.2.2: Profile Characteristics	Parameter(s) are to be modified according to seasonal variation and the day, week and month set in the Study Time.
18.2.4: Scalar Characteristics	Parameter(s) are to be manually modified by a scaling factor.
18.2.5: Vector Characteristics with Discrete Scales	Discrete parameter states are to be selectable.
18.2.5: Vector Characteristics with Continuous Scales	Parameter states may be interpolated between entered values.
18.2.5: Vector Characteristics with Frequency Scales	Parameter(s) are to be modified with Frequency.
18.2.5: Vector Characteristics with Time Scales	Parameter(s) are to be modified based on a user-defined scale referencing the Study Time.
18.2.6: Matrix Parameter Characteristics	Parameter states are based on two variables, and may be interpolated between entered values.
18.2.7: Parameter Characteristics from Files	Parameter states and the trigger (optional) is to be read from a file.
18.2.8: Characteristic References	Reference link between a parameter and a Characteristic

Table 18.2.1: Summary of Parameter Characteristics

Usage

With the exception of the Scalar Characteristic, the “Usage” field at the bottom of the characteristic dialog can be used to specify how “Values” are applied to the parameter that the characteristic is associated with:

- Relative in % will multiply the parameter by the percentage value.
- Relative will multiply the parameter by the value.
- Absolute will replace the current parameter with the absolute value entered.

Characteristic Curves

For continuous characteristics, various approximation methods are available to interpolate and extrapolate from the entered Values:

- Constant: holds the Y-value in between X-values.
- Linear: uses a linear interpolation.
- Polynomial: uses a polynomial function with a user defined degree.
- Spline: uses a spline function.
- Hermite: uses Hermite interpolation.

The approximation curve will be shown in the diagram page of the Characteristic dialog. The interpolated Y-value may vary considerably depending on the entered data and the approximation function applied.

Figure 18.2.2 highlights the difference between interpolation methods for an example characteristic with a continuous scale (shown on the horizontal axis from -20 to +45). For instance, at a trigger value of 25, linear interpolation will give an output value of 60, whereas constant interpolation will give an output value of 40.

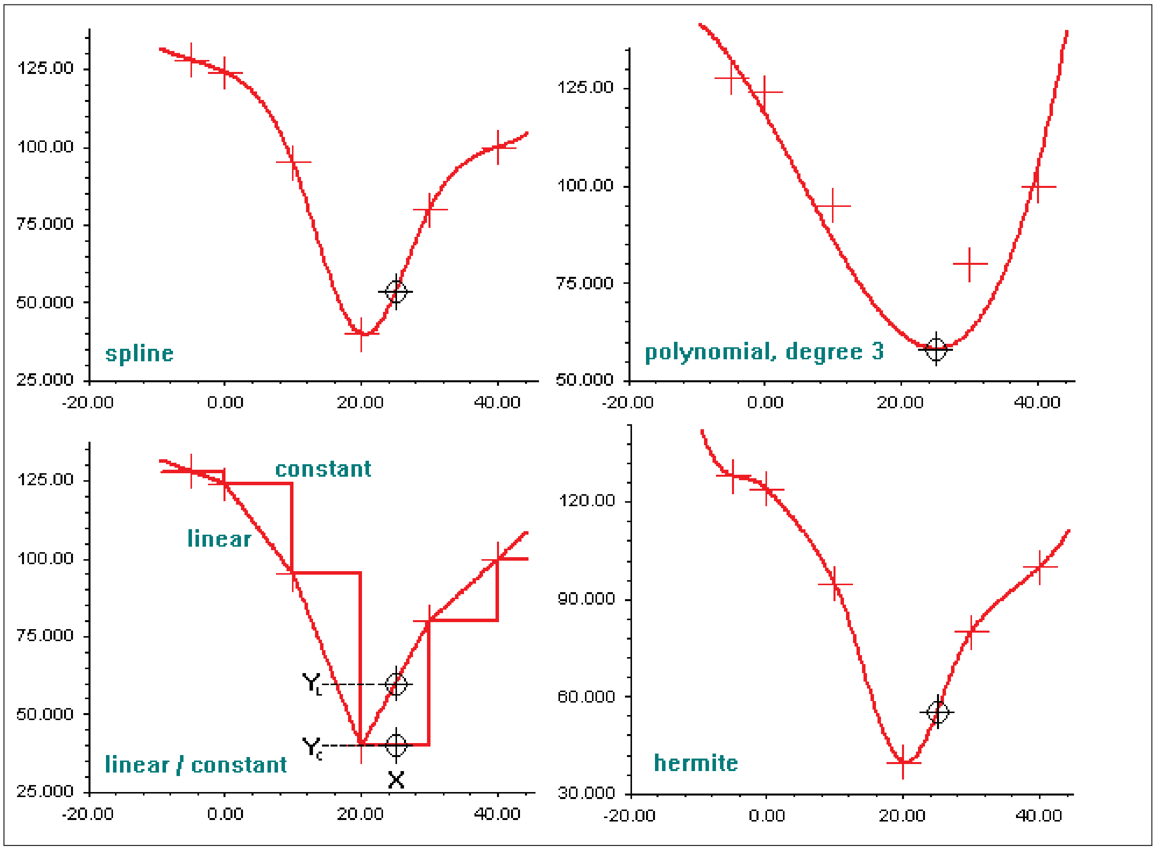


Figure 18.2.2: Approximated characteristics

Note that Approximation methods are not available for discrete characteristics.

Creating a Characteristic

To create a Characteristic, right-click on the desired parameter (e.g. 'Active Power'), right-click and select *Add Project Characteristic* or *Add Global Characteristic* and create the desired characteristic. It is also possible to edit the existing characteristic by selecting the option *Edit Characteristic*. Details of how to create the different types of characteristics are provided in the following sub-sections, including an example application of characteristics.

18.2.1 Time Characteristics

General background on characteristics and their properties is provided in Section 18.2. The time characteristic determines the value of the parameter according to the *study time* (*SetTime*).

Recurrence

When using the time characteristic (*ChaTime*), the user has the option to set a *Recurrence* period for the characteristic values. The available options are:

- Daily
- Weekly
- Monthly
- Yearly
- None

If the values are to be supplied locally via a table (see below for the descriptions of different data sources), the user will also see a field called *Resolution*. The resolution that can be selected depends upon the recurrence period.

A special case is the selection of yearly recurrence with a resolution of “Seasons”. If this option is selected, a new *Seasons* page name appears on the left-hand side of the dialog. On this page, the user is able to configure the required seasons. Once this is done, the values can be entered on the *Curve* page.

There are four options for defining the data source of values used in a time characteristic: *Table*, *File*, *Result File* and *Database*. The *Table* data is stored internally within *PowerFactory*. The *File* data is stored externally to *PowerFactory* in a *Comma Separated Values* (*.csv) file or *User Defined Text File*. For the *Result File* option, the characteristic is created from data in an existing results file, and the *Database* option enables the data to be taken from a database outside the *PowerFactory* database, for which access information has to be provided by the user, as described below.

Time characteristic using internal table

To define a project time characteristic for a parameter using a table:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *Time Characteristic . . .*
- Click the *New Object* button
- The edit dialog of the Time Characteristic will be displayed. Define the parameter name and select ’Data Source’ *Table*.
- Select the desired ’Recurrence’ and the ’Resolution’.
- Define the ’Usage’ and ’Approximation’ and enter the characteristic values in the table.
- Press **OK**.

Time characteristic using an external file

To define a project time characteristic for a parameter using an external file:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *Time Characteristic . . .*
- Click the *New Object* button
- The edit dialog of the Time Characteristic will be displayed. Define the parameter name and select ’Data Source’ *File*.
- Select the desired ’Filename’ and file ’Format’.
- Define the file configuration including the ’Unit’ of time or ’Time Stamped Data’ format, ’Time Column’ and ’Data Column’ and ’Column separator’ and ’Decimal separator’.

- Define the 'Usage' and 'Approximation'.
- Press **OK**.

Time characteristic using a Result File

To define a project time characteristic for a parameter using a result file:

- In the edit dialog of the target network component right-click on the parameter for which the characteristic is to be defined.
- Select *Add Project Characteristic* → *Time Characteristic ...*
- Click the *New Object* button
- The edit dialog of the Time Characteristic will be displayed. Define the parameter name and select 'Data Source' *Result File*.
- Use the drop-down arrow to select the Result File.
- Select the element whose results are to be used.
- Select the relevant parameter.
- Define the 'Usage' and 'Approximation'.
- Press **OK**.

Time characteristic using a Database

When defining characteristics which use data from an external database, the user has to set up an ODBC Database Configuration object, or select an existing Database Configuration. These database configurations can exist within the project, in a folder, or in a configuration area, for example.

If defining a new configuration, this can be done from the *ChaTime* dialog, using the drop-down arrow next to *Database*. This is described below, as part of the process of setting up the characteristic.

To define a project time characteristic for a parameter using an external database:

- In the edit dialog of the target network component right-click on the parameter for which the characteristic is to be created.
- Select *Add Project Characteristic* → *Time Characteristic ...*
- Click the *New Object* button
- The edit dialog of the Time Characteristic will be displayed.
- Define the parameter name and select "Data Source" *Database*.
- Using the drop-down arrow next to *Database* in the Database panel of the dialog, the database configuration details are either selected or a new configuration created like this:
  - Navigate to the chosen location for the new object.
  - Click the *New Object* button
  - Select the database system and the ODBC driver. Oracle, PostgreSQL and SQL Server are offered explicitly as database system options; a fourth option *Generic* enables the user to select any available ODBC driver, but in this case the user must determine which of the remaining fields needs to be populated.
  - Enter the server name.
  - Enter the access details (Username and password) for the database system.
  - Enter the database name.
- Two *Table mode*s are offered, as illustrated in figure 18.2.3 below:

- "Value Column" is used when individual database tables exist for individual element characteristics. In this case the user must provide the Table name, the Time Column name and the Value Column name.
- "Id Column" is used if there is a common table, where data for more than one characteristics is stored. In this case, the table will have an additional column for id-references, and so the column name and id must be supplied.

- The *Time offset*: The values in the time column are normally assumed to be UTC times, which are therefore independent of the local time zone or daylight saving regime. If the user wishes to enter local times instead, the *Time offset* is used to make the necessary adjustment to these values to convert them to UTC times. For example, if the values in the time column are in Central European Time (CET i.e. UTC+01:00), the *Time offset* should be "-1.0".
- Select the relevant parameter.
- Define the 'Usage' and 'Approximation'.
- Press **OK**.

Value Column

Timestamp	Active Power	Reactive Power
2019-01-01 00:00:00	11.167	3.350
2019-01-01 00:15:00	11.145	3.344
2019-01-01 00:30:00	11.132	3.340
…….		

Id Column

Timestamp	Id	Active Power	Reactive Power
2019-01-01 00:00:00	Load01	11.167	3.350
2019-01-01 00:00:00	Load02	2.449	0.735
2019-01-01 00:00:00	Load03	30.488	9.146
…….			
2019-01-01 00:15:00	Load01	11.145	3.344
2019-01-01 00:15:00	Load02	2.300	0.690
2019-01-01 00:15:00	Load03	28.034	8.410
…….			
2019-01-01 00:30:00	Load01	11.132	3.340
2019-01-01 00:30:00	Load02	2.130	0.639
2019-01-01 00:30:00	Load03	27.001	8.100
…….			

Figure 18.2.3: Database table modes

Note: When setting up the database, users should be aware that the read performance for large tables can be vastly improved by ensuring that there are appropriate table indexes. The *value column*-based table should have an index on the Timestamp column (e.g. with "CREATE INDEX IndexName ON Table (Timestamp)"); the *id*-based table should have an combined index on both the

Timestamp and the Id column (e.g. with “CREATE INDEX IndexName ON Table(Timestamp,Id)”).

Note: When defining characteristics with time stamped data, be aware of **daylight savings**.

- In summer when the clocks are put forward 1 hour, the time stamps from 02:00 AM to 02:59 AM do not occur this day. *PowerFactory* expects that these time stamps do not occur in the time characteristic either. Time stamps within this time range will be ignored.
- In winter when the clocks are put back 1 hour, the time stamps from 02:00 AM to 02:59 AM occur twice this day. *PowerFactory* expects that these time stamps only occur once in the time characteristic. Multiple definitions for the same time stamp will be ignored.

Discrete Time Characteristics

The discrete time characteristic (*ChaDisctime*) is provided for backward compatibility with previous versions of *PowerFactory*. It is more restricted than the time characteristic and hence its use is limited since *PowerFactory* version 15.1. Similar to the time characteristic, the discrete time characteristic uses an internally defined series of time scales that are convenient to use to define the characteristic. The user simply selects a scale (e.g. day of the week) and enters the corresponding values.

18.2.2 Profile Characteristics

General background on characteristics and their properties is provided in Section 18.2.

The profile characteristic is used to select a time characteristic (*ChaTime*) corresponding to individual days or group of days and each season. The profile characteristic can also be used to select a time characteristic for certain holiday days.

To define a project profile characteristic for a parameter:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *Profile Characteristic . . .*
- Click the *New Object* button
- The edit dialog of the Profile Characteristic will be displayed.
- Select the ’Seasons’ page and define one or more seasons with a ’Description’, ’Start Day’, ’Start Month’, ’End Day’ and ’End Month’. Note that Seasons can not overlap with each other.
- Select the ’Groups of Days’ page and define grouping for each day and holiday.
- Select the ’Holidays’ page and define one or more holidays with a ’Description’, ’Day’, ’Month’, if it is ’Yearly’ or select a holiday ’Year’.
- Select the ’General’ page, Right Click and Select ’Select Element/Type . . . ’ or Double-Click on each relevant cell and select or create a time characteristics for each group of days, holiday and season.
- Press **OK**.

Yearly Growth Characteristic

In addition to seasonal characteristic variation, a yearly growth characteristic can also be defined. A yearly growth characteristic is defined using a time characteristic (*ChaTime*) with a recurrence value of “None”, for the specified years.

Note: All daily and yearly characteristics must be relative. No absolute-value characteristics are permitted.

18.2.3 Scaling Factor

General background on characteristics is provided in Section 18.2.

Scaling factors are used when a parameter should be multiplied by a certain value or percentage. For example, a scaling factor could be used to multiply the *Active Power* value of one or more static generators by 0.5. If a parameter is assigned several scaling factors, it will be multiplied by their product.

To define a project scaling factor for a parameter:

- In the edit dialog of the target network component right-click on the desired parameter (e.g. 'Active Power').
- Select *Add Project Characteristic* → *Scaling Factor...*
- Click the *New Object* button
- The edit dialog will be displayed. Set the value of the factor. The associated trigger is automatically created in the current study case.
- Define the 'Usage' "relative" or "relative in %".
- Press **OK**.

18.2.4 Linear Functions

General background on characteristics and their properties is provided in Section 18.2.

Linear Functions are used when a parameter should vary according to a mathematical relationship, with reference to a scale value "x". For example, a linear function may reference a *Scalar and Trigger* (*TriVal*) with a *Unit* of 'Temperature'. Then, if the temperature is set to, say, 15 deg, the parameter that this characteristic is applied to will thus be multiplied by the value of the linear function $2 \cdot 15 + 3 = 33$.

To define a project linear function for a parameter:

- In the edit dialog of the target network component right-click on the desired parameter (e.g. 'Active Power').
- Select *Add Project Characteristic* → *Linear Function...*
- Click the *New Object* button
- The edit dialog will be displayed. Click 'Select' from the drop down menu next to 'Scale' and select an existing scale and press **OK**, or create a new scale:
  - Click on the 'New Object' button to create a *Scalar and Trigger* (*TriVal*) and set the desired units of the scale. The associated trigger is automatically created in the current study case.
  - Press **OK**.
- Define the 'Usage' and enter the parameters 'A' and 'b' of the linear function $A \cdot x + b$.
- Press **OK**.

18.2.5 Vector Characteristics

Vector Characteristics may be defined with reference to *Discrete Scales, Continuous Scales, Frequency Scales,* and *Time Scales.*

Vector Characteristics with Discrete Scales (TriDisc)

General background on characteristics and their properties is provided in Section 18.2.

A discrete parameter characteristic is used to set the value of a parameter according to discrete cases set by the trigger of a discrete scale. A discrete scale is a list of cases, each defined by a short text description. The current value is shown in the characteristic dialog in red, according to the case that is currently active.

To define a new project discrete parameter characteristic:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *One Dimension Vector...*
- Click the *New Object* button
- The edit dialog of the one dimension vector characteristic (generic class for one dimensional characteristics) will be displayed. Click 'Select' from the drop down menu next to 'Scale' and select an existing scale and press **OK**, or create a new scale:
  - Click on the *New Object* button and select *Discrete Scale and Trigger* (*TriDisc*).
  - Write the name of the scale cases (one case per line).
  - Press **OK** twice.
- Define the 'Usage' and enter the characteristic values.
- Press **OK**.

The diagram page for the discrete characteristic shows a bar graph for the available cases. The bar for the case that is currently active (set by the trigger) is shown in black.

Vector Characteristics with Continuous Scales (TriCont)

General background on characteristics and their properties is provided in Section 18.2.

A continuous parameter characteristic is used to set the value of a parameter ('Y' values) according to the 'X' values set in the continuous scale.

To define a new project continuous parameter characteristic:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *One Dimension Vector...*
- Click the *New Object* button
- The edit dialog of the one dimension vector characteristic (generic class for one dimensional characteristics) will be displayed. Click 'Select' from the drop down menu next to 'Scale' and select an existing scale and press **OK**, or create a new scale:
  - Click on the *New Object* button and select *Continuous Scale and Trigger* (*TriCont*).
  - Enter the unit of the 'X' values.
  - Append the required number of rows (right-click on the first row of the Scale table and select *Append n rows*) and enter the 'X' values.
  - Press **OK**.
- Define the 'Usage', enter the characteristic 'Y' values, and define the 'Approximation' function.
- Press OK.

Vector Characteristics with Frequency Scales (TriFreq)

General background on characteristics and their properties is provided in Section 18.2.

A frequency characteristic is a continuous characteristic with a scale defined by frequency values in Hz. The definition procedure is similar to that of the continuous characteristics, although the Frequency Scale (*TriFreq*) is selected.

Vector Characteristics with Time Scales (TriTime)

General background on characteristics and their properties is provided in Section 18.2.

Time parameter characteristics are continuous characteristics using time scales. A time scale is a special kind of continuous scale that uses the global time trigger of the active study case. The unit of the time trigger is always a unit of time but may range from seconds to years. This means that changing the unit from minutes to hours, for instance, will stretch the scale 60-fold. The units 's', 'm', and 'h' are respectively, the second, minute and hour of normal daytime. A Time Scale may be used, for example, to enter four equidistant hours in a year (1095, 3285, 5475, and 7665).

The definition procedure is similar to that of the continuous characteristics, although the Time Scale (*TriTime*) scale is selected.

18.2.6 Matrix Parameter Characteristics

General background on characteristics and their properties is provided in Section 18.2.

When defining a matrix parameter characteristic, two scales must be defined. The first scale, that for columns, must be a discrete scale. The scale for rows may be a discrete or continuous scale.

To define a new project matrix parameter characteristic:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *Add Project Characteristic* → *Two Dimension - Matrix...*
- Click the *New Object* button
- The edit dialog of the matrix characteristic will be displayed. Click 'Select' from the drop down menu next to each 'Scale' and select an existing scale and press **OK**, or create a new scales. Scales can be defined as discussed in previous sections.

A column calculator can be used to calculate the column values, as a function of other columns. This is done by pressing the **Calculate...** button. Once the values have been entered and the triggers have been set, the 'Current Value' field will show the value to be used by the characteristic.

18.2.7 Parameter Characteristics from Files

General background on characteristics and their properties is provided in Section 18.2.

When a series of data is available in an external file, such as an Excel file, or tab or space separated file this data may be utilised as a characteristic if the "Parameter Characteristic from File" (*ChaVecfile* object) is used. The external file must have the scale column for the data series in column 1.

To define a new parameter characteristic from file:

- In the edit dialog of the target network component right-click on the desired parameter.
- Select *New Characteristic* → *Characteristic from File...*
- Complete the input data fields, including:
  - Define (or select) a scale and trigger. Scales can be defined as discussed in previous sections.

- Generally the 'Column' should be set to the default of '1'. The field is used for specialised purposes.
- Set the 'Factor A' and 'Factor B' fields to adjust or convert the input data. The data contained in column 2 of the external file will be adjusted by $y = ax + b$ where "x" is the data in the external file and "y" is what will be loaded into the characteristic.
- Set the 'Usage' and 'Approximation'.
- Once the file link has been set, press the Update button to upload the data from the external file to the characteristic.

18.2.8 Characteristic References

When a characteristic is defined for an objects parameter, *PowerFactory* automatically creates a characteristic reference (*ChaRef* object). The characteristic reference is stored within the *PowerFactory* database with the object. The characteristic reference acts as a pointer for the parameter to the characteristic. The characteristic reference includes the following parameters:

Parameter the name of the object parameter assigned to the characteristic. This field cannot be modified by the user.

Characteristic the characteristic which is to be applied to the parameter.

Inactive a check-box which can be used to disable a characteristic reference.

The ability to disable the characteristic for individual objects using the object filter and the *Inactivate* option makes data manipulation using characteristics quite flexible.

18.2.9 Edit Characteristic Dialog

Once a parameter has a characteristic defined, then an option to *Edit characteristic(s)* becomes visible on the parameters context sensitive menu, i.e. select parameter and *right-click* → *Edit characteristic(s)*. Once selected, the *Edit characteristics* dialog appears which lists all the characteristics referenced by the parameter. The *Edit characteristics* dialog provides a graphical representation of the characteristic and allows characteristics to be inserted, appended and deleted. The *Edit characteristics* dialog also allows modification of individual characteristics values, triggers and characteristic activation and deactivation.

Note: By default the value of the first active characteristic is assigned to the parameter.

18.2.10 Characteristics Tab in Data Filters

When viewing elements in a Data Manager or Network Model Manager, parameter characteristics information can be seen by selecting the Characteristics tab. For this tab to be visible, it must be enabled in the User Settings, on the "Functions" page. An example of a Network Model Manager showing the Characteristics tab is shown in Figure 18.2.4 (remember that the browser must be in "detail" mode to see these tabs). Note also that the data colouring indicates that characteristics are applied.

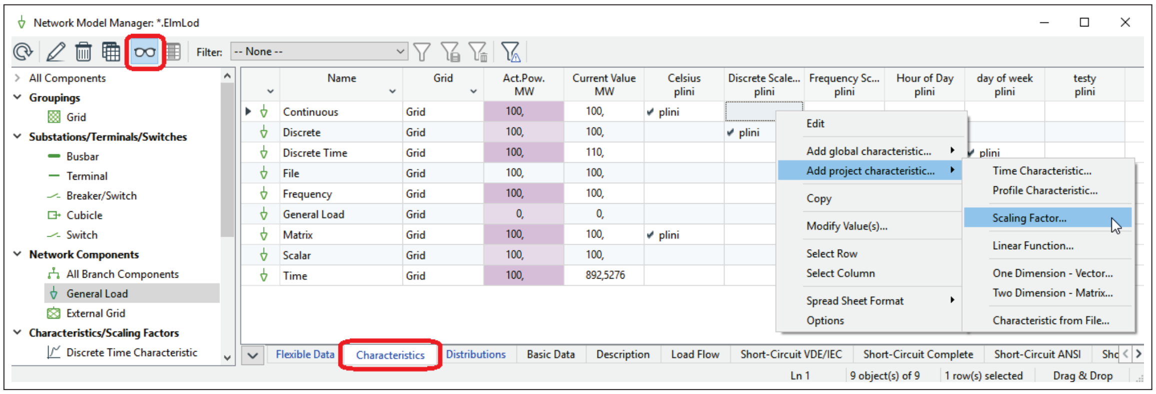


Figure 18.2.4: Network Model Manager Characteristics tab

The Characteristics tab shows all characteristics defined for the displayed objects, together with the original value and the current value as determined by the characteristic. In the example, various scales are applied to modify the active power from 100 MW to the “Current Value”. The current values will be used in all calculations. New characteristics for individual or multiple elements can be defined by selecting the relevant fields and doing *right-click* → *Add project characteristic...*

The Characteristics tab will only show a particular characteristic column when at least one of the objects has that characteristic defined for a parameter. It is thus necessary to define a characteristic for one object prior to using the browser, when the user would like to assign characteristics, for the same parameter, for a range of other objects. To define a Project “High-Low” loading characteristic for all loads, for instance, can thus be done by performing the following steps.

- Create a discrete scale in the grid folder.
- Create a vector characteristic using this scale in the grid folder.
- Edit one of the loads, right-click the active power field and assign the vector characteristic to the relevant parameter.
- Open a browser with all loads, activate the “detail” mode and select the Characteristics tab.
- Select the characteristic column (*right-click* → *Select Column*) and then right-click the selected column.
- Use the *Select Project Characteristic...* option and select the vector characteristic.

18.2.11 Example Application of Characteristics

Consider the following example, where the operating point of a generator should be easily modified by the user to predefined values within the capability limits of the machine.

Firstly, the *Active Power* of the synchronous generator is set to the maximum capability of 150 MW. Then, a vector characteristic is added to the *Active Power* parameter. To create a new *Project Vector Characteristic*, right-click on the *Active Power* parameter (pgini) and select *Add Profile Characteristic* → *One Dimension - Vector...*. Click on the *New Object* icon and define a characteristic called “Active Power” in the *ChaVec* dialog.

A new discrete scale is required. To create the scale, click on the arrow next to *Scale* and select *Select...*. Click on the *New Object* icon and create a new *Discrete Scale and Trigger (TriDisc)*. The *Discrete Scale and Trigger* is named “Output Level”, with three cases as shown in Figure 18.2.5.

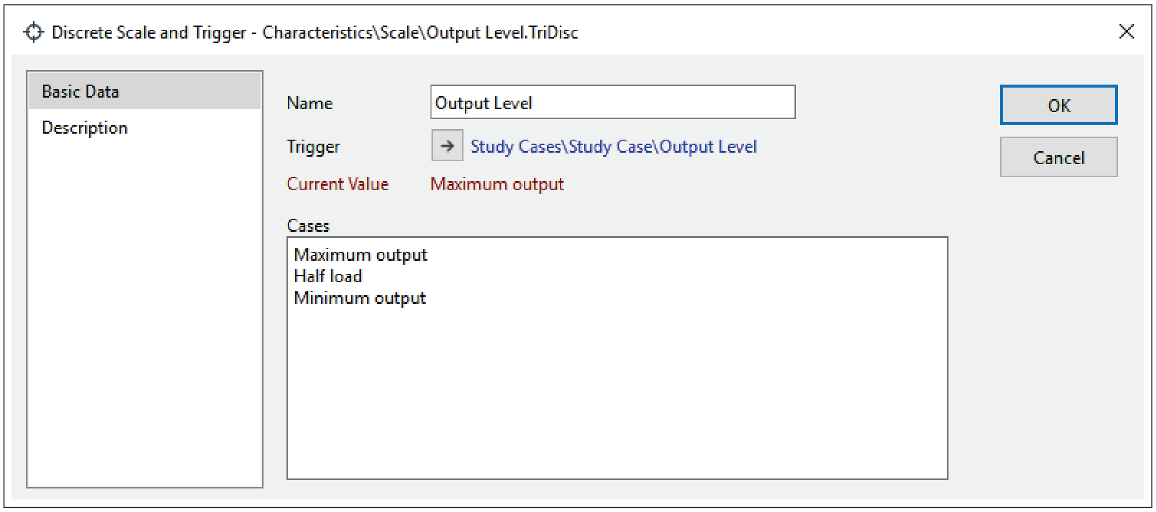


Figure 18.2.5: Active Power Discrete Scale and Trigger

Click on **OK** to return to the *Vector Characteristic*. Define the values for the different loading scenarios. Values are entered in %, and thus *Usage* is set to 'relative in %'. Figure 18.2.6 shows the resultant vector characteristic, including a reference to the Scale "Output Level" and the current parameter value.

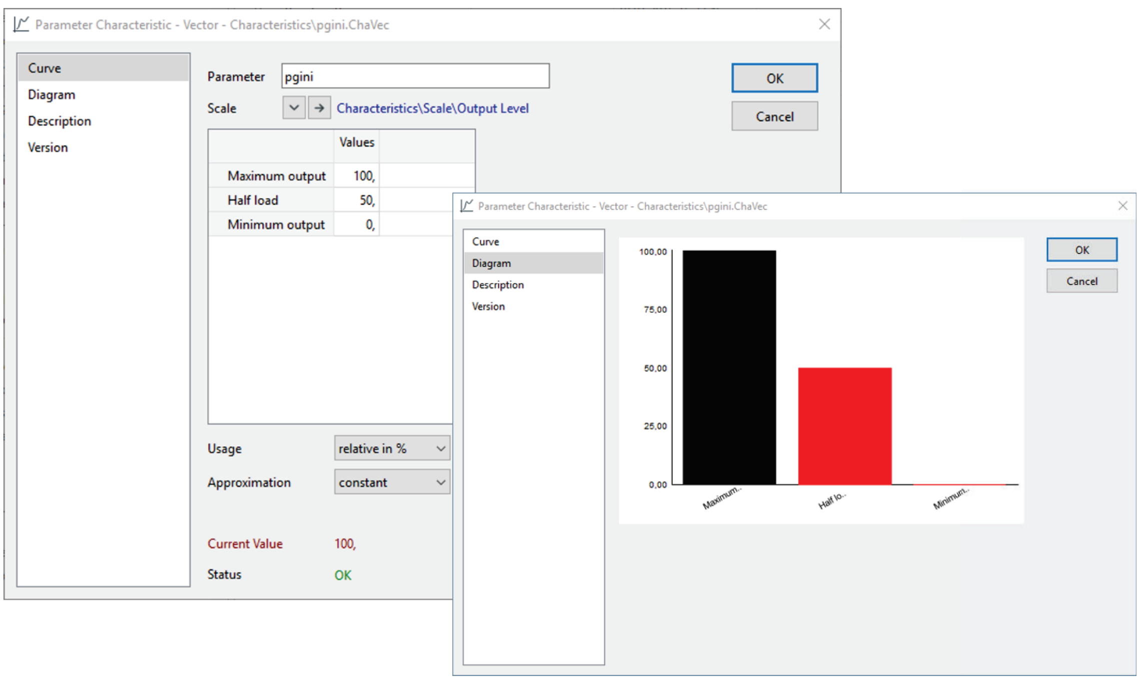


Figure 18.2.6: Active Power Parameter Characteristic

Next, a matrix characteristic is added to the *Reactive Power* parameter of the generator in a similar fashion to the *Active Power* characteristic. A new discrete scale named "Operating Region" is created (for the Columns) and the three operating regions "Underexcited", "Unity PF" and "Overexcited" are defined.

The scale "Operating Region" is linked to the *Scale for Columns*, and the previously defined scale "Output Level" is selected for the *Scale for Rows*. Absolute Mvar values are entered in the Matrix Characteristic as shown in Figure 18.2.7.

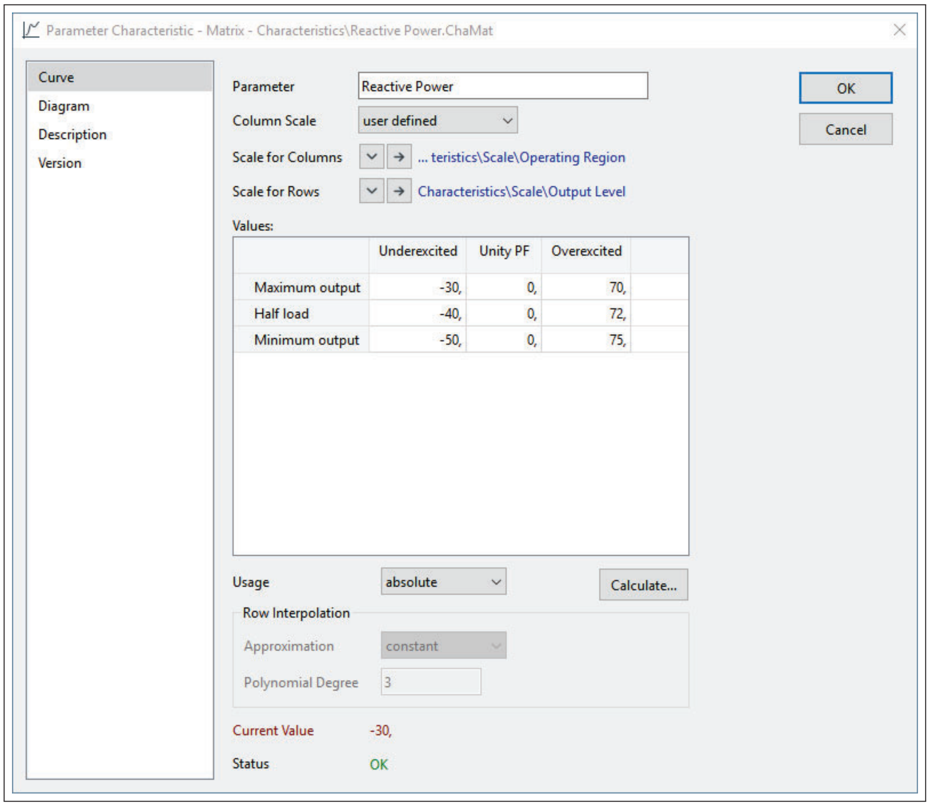


Figure 18.2.7: Reactive Power Matrix Characteristic

Now that the characteristics and triggers are defined, the “Operating Region” and “Output Level” triggers can be used to quickly modify the operating point of the generator (see Figure 18.2.8).

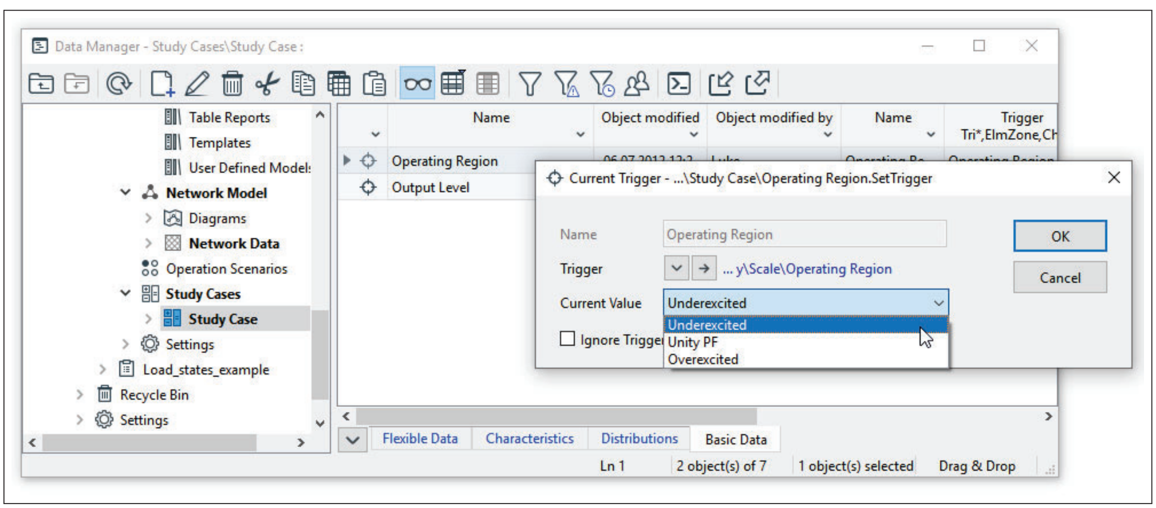


Figure 18.2.8: Setting of Discrete Triggers

18.3 Load States

This section describes Load States, as used in Reliability and Optimal Capacitor Placement calculations.

18.3.1 Creating Load States

Pre-requisites:

Prior to creating load states, a time-based parameter characteristics must be defined for at least one load in the network model. See Time Characteristics (*ChaTime*) in Section 18.2.1 and Vector Characteristics with Time Scales (*TriTime*) in Section 18.2.5 for more information on parameter characteristics, as well as the example later in this section.

Follow these steps to create the load states:

1. For calculation of load states:
   - (Reliability) click the 'Create Load States' icon ( ) on the reliability toolbar and select 'Load States'. Optionally inspect or alter the settings of the Reliability Calculation and Load Flow commands.

   Note: Optionally choose the option to "Consider Generators" to consider time varying power feed-in of generators. If selected, the generator power state will additionally be contained within the clusters. Please note, that the Load and Generator States can only be applied for Reliability assessment.

   - (Optimal Capacitor Placement) Click on 'Load Characteristics' page of the Optimal Capacitor Placement command and select 'Create Load States'.
2. Enter the time period for calculation of load states:
   - (Reliability) Enter the year.
   - (Optimal Capacitor Placement) Enter Start Time and End Time. The time period is inclusive of the start time but exclusive of the end time.
3. Enter the Accuracy. The lower accuracy percentage, the more load states are generated.
4. Optional: Limit the number of load states to a user-defined value. If the total number of calculated load states exceeds this parameter then either the time period of the sweep or the accuracy should be reduced.
5. Optional: Change the threshold for ignoring load states with a low probability by altering the 'Minimum Probability'. If selected, states with a probability less than this parameter are excluded from the discretisation algorithm.
6. Click Execute to generate the load states.

18.3.2 Viewing Existing Load States

After you have generated the load states as described above, or if you want to inspect previously generated load states follow these steps:

1. Using the Data Manager, select the 'Reliability Assessment' or 'Optimal Capacitor Placement' command within the active Study Case.
2. Use the filter ( ) (in the Data Manager window) to select the 'load states' object. There should now be created load states visible in the right panel of the Data Manager.
3. Locate the 'load states' object and double-click to view the load states.

18.3.3 Load State Object Properties

The load states object properties are as follows:

Basic Data

- **year:** The Year used to create the load states.
- **Number of loads:** Number of loads and generators considered in the load cluster object.
- **Number of states:** This equals the number of columns in the “Clusters” table.
- **Loads:** Table containing each load considered by the load states creation algorithm and their peak demand.
- **Clusters:** Table containing all load clusters. The first row in the table contains the probability of the corresponding cluster. The remaining rows contain the power values of the loads. Every column in the table contains a complete cluster of loads with the corresponding power.

Diagram Page

- **Displayed Load:** Use the selection control to change the load displayed on the plot.

The plot shows the cluster values (P and Q) for the selected load where the width of each bar represents the probability of occurrence for that cluster in the given year.

18.3.4 Example Load States

The example below shows how load states can be generated for a network model with four Loads (Ld1, Ld2, Ld3, and Ld4).

1. The Vector Characteristic shown in Figure 18.3.1 is applied to both Active Power and Reactive Power of load Ld4 only, with the associated Time Scale shown in Figure 18.3.2 Ld4 is initially set to 3.1 MW, 0.02 Mvar.

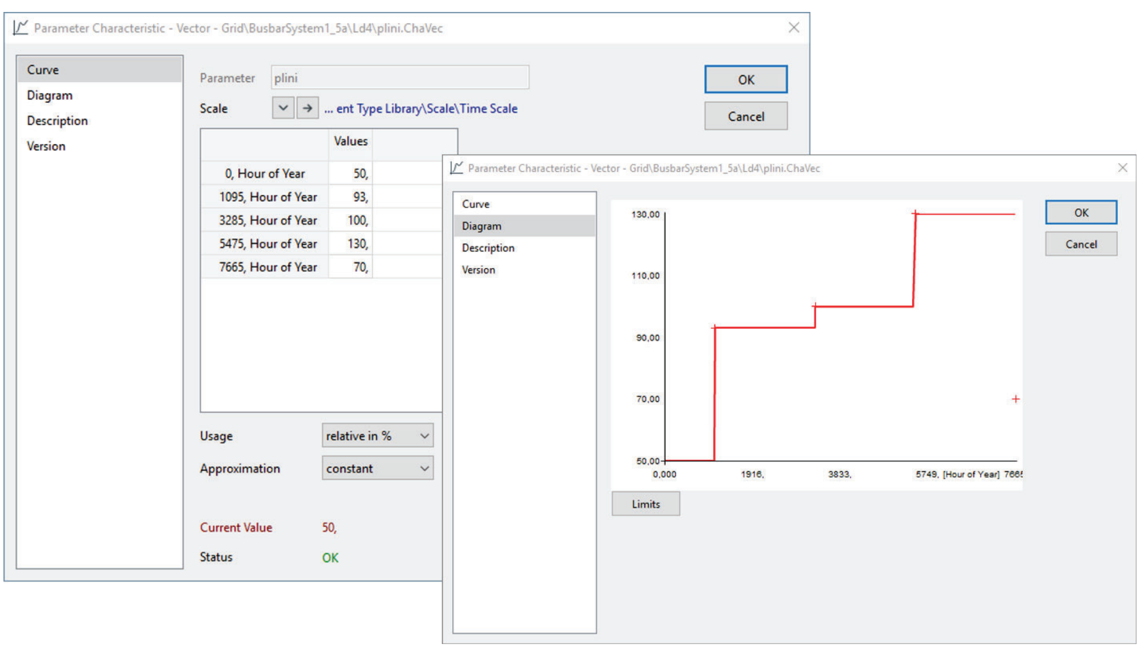


Figure 18.3.1: Load State Vector Characteristic

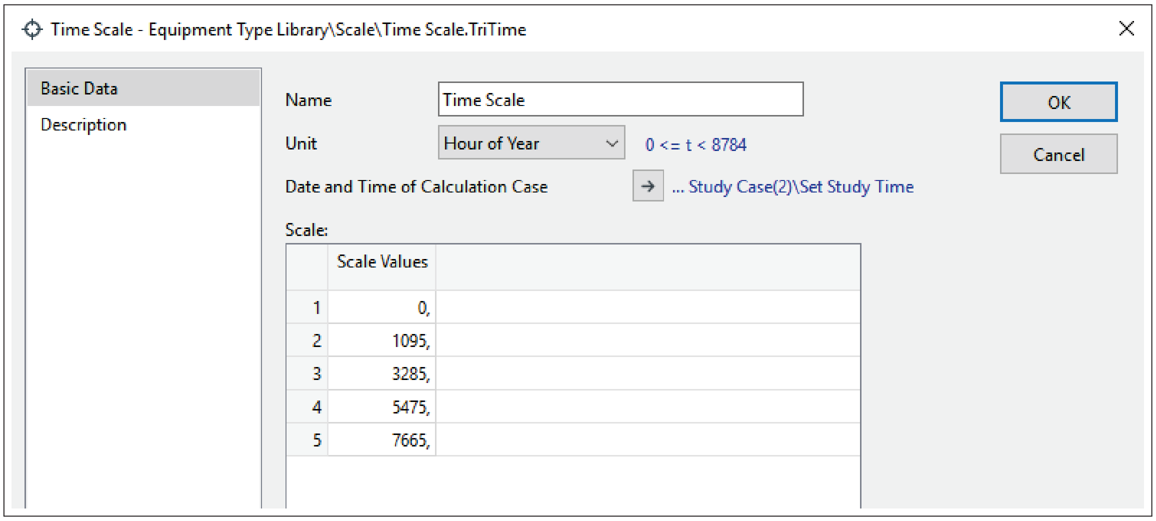


Figure 18.3.2: Time Scale for Load State Characteristic

2. Load States are generated by clicking *Create Load States* (as discussed in the preceding section).
3. *PowerFactory* calculates the resultant Load States:
   - The maximum value of each load L_p is determined for the time interval considered. In the example, Ld4 has a peak load of 4.03 MW.
   - The 'load interval size' (Int) is determined for each load, where $Int = L_p \cdot Acc$ and 'Acc' is the accuracy parameter entered by the user. For the example above using an accuracy of 10 %, the interval size for Active Power is 0.403 MW.
   - For each hour of the time sweep and for each load determine the Load Interval: $LInt = Ceil\left(\frac{L_i}{Int}\right)$ where L_i is the load value at hour 'i'.
   - Identify common intervals and group these as independent states.
   - Calculate the probability of each state based on its frequency of occurrence.

The independent states and their probabilities are shown in Figure 18.3.3. Load states for Ld4 vary according to the characteristic parameters, where the states from characteristic values of 93 % and 100 % have been combined due to the selection of 10 % accuracy in the calculation. Load states for Ld1, Ld2, and Ld3 do not vary, since characteristics were not entered for these loads.

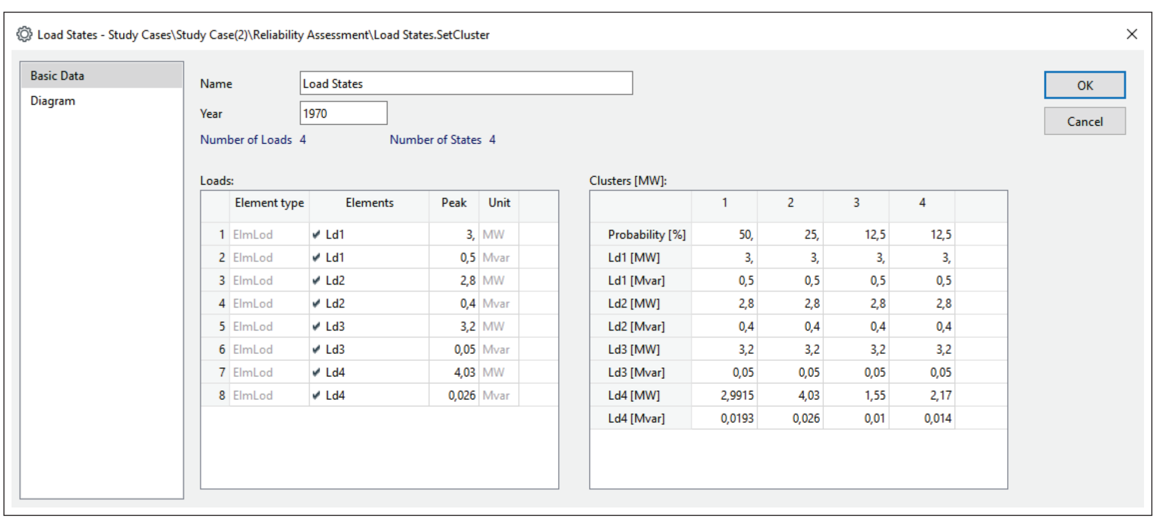


Figure 18.3.3: Load States (SetCluster) dialog box

18.4 Load Distribution States

This section describes how to create load distribution states, as used by the Reliability calculation.

18.4.1 Creating Load Distribution States

Pre-requisites:

Prior to creating load distribution states a substation/s must have been defined within the model. A distribution curve must have also been defined (accessed from the reliability page of the substation/s).

Follow these steps to create the load distribution states:

1. Click the 'Create Load States' button ( ) on the reliability toolbar. The load states creation dialog will appear.
2. Optional: Use the Reliability Assessment selection control to inspect or alter the settings of the Reliability Calculation command. This selection control points to the default reliability command within the active Study Case.
3. Optional: Use the Load Flow selection button to inspect and alter the settings of the load flow command. This selection control points to the default load-flow command within the active Study Case.
4. Enter the Minimum Time Step in hours (suggested to be the minimum step size on the load distribution curve).
5. Enter the Maximum Power Step (0.05pu by default).
6. Optional: Force Load State at S = 1.0 p.u. so that a state is created at P = 1.0 pu, irrespective of the load distribution curve data and step sizes entered.
7. Click Execute to generate the load distribution states.

18.4.2 Viewing Existing Load Distribution States

After you have generated the load states as described above, or if you want to inspect previously generated load states follow these steps:

1. Using the Data Manager, select the 'Reliability Assessment' Command within the Active Study Case.
2. Optional: Use the filter ( ) (in the Data Manager window) to select the 'load distribution states' object. There should now be created load distribution states visible in the right panel of the Data Manager.
3. Locate the 'load distribution states' object and double-click to view the load states.

18.4.3 Load Distribution State Object Properties

The distribution load states object properties are as follows:

Basic Data

Year The Year used to create the load states.

- **Clusters:** Table containing all substation clusters. The first row in the table contains the probability of the corresponding cluster. The remaining rows contain the power values of the substations. Every column in the table contains a complete cluster of substations with the corresponding power.
- **Number of substations:** Number of substations considered in the Distribution State object.
- **Number of states:** This equals the number of columns in the Distribution State table.

Diagram Page

Displayed Station: Use the selection control to change the load displayed on the plot

The plot shows the cluster values (Apparent power in pu with reference to the substation load) for the selected substation where the width of each bar represents the probability of occurrence for that cluster.

18.4.4 Example Load Distribution States

In this example, a Load Distribution Curve is entered for a substation.

1. The Load Distribution Curve shown in Figure 18.4.1 is entered for the substation (Apparent power in pu of substation load).

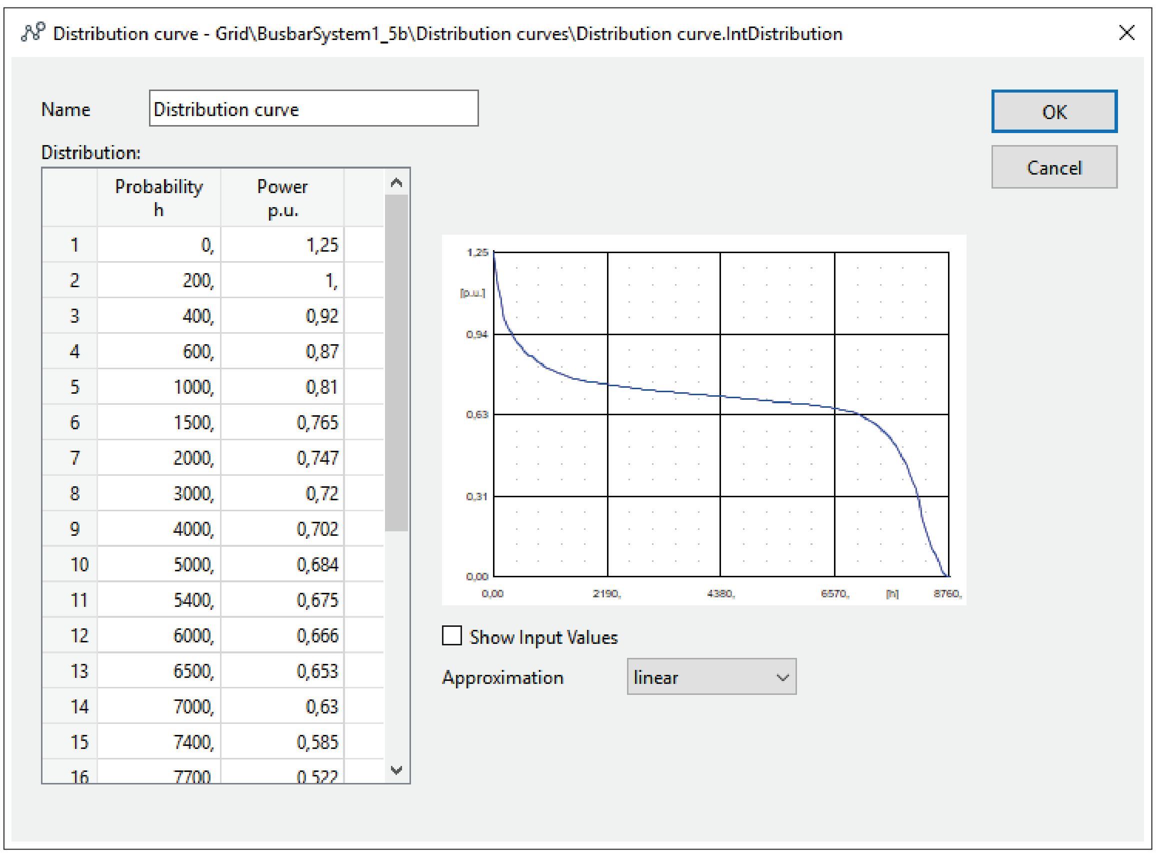


Figure 18.4.1: Substation Load Distribution Curve (*IntDistribution*)

2. Load States are generated by clicking *Create Load Distribution States* (as discussed in the preceding section).
3. The resultant Load Distribution States are shown in Figure 18.4.2. 'Force Load State at S = 1.0 p.u.' has not been selected in this instance.

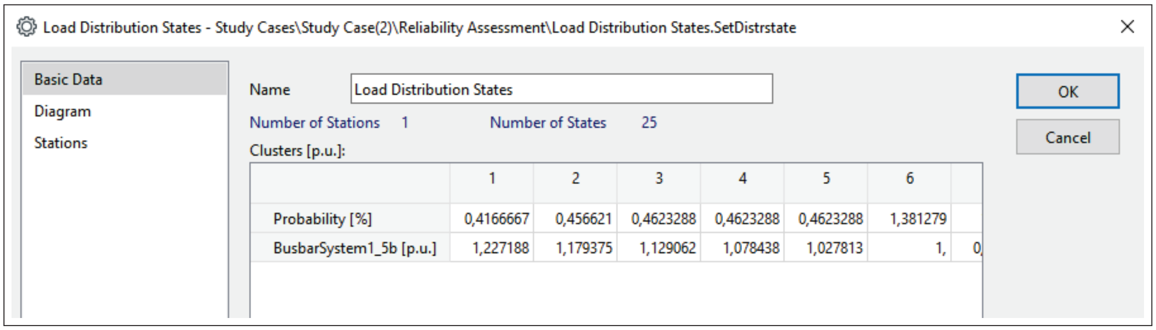


Figure 18.4.2: Load Distribution States (*SetDistrstate*)

18.5 Tariffs

This section describes the definition of Time Tariffs (as used in Reliability calculations), and Energy Tariffs (as used in Reliability calculations and Optimal RCS Placement calculations, and Techno-Economical calculations).

18.5.1 Defining Time Tariffs

A time tariff characteristic can be defined by taking the following steps:

1. Choose the *Select* option from the 'Tariff' selection control on the reliability page of the load element. A Data Manager browser will appear with the 'Equipment Type Library' selected.
2. Optional: If you have previously defined a 'Tariff' characteristic and want to re-use it, you can select it now. Press **OK** to return to the load element to reliability page.
3. Create a time tariff object by pressing the *New Object* button from the data browser toolbar. A type creation dialog should appear.
4. Select *Time Tariff* and press **OK**. A 'Time Tariff' dialog box will appear.
5. Select the unit of the interruption cost function by choosing from the following options:

   $/kW Cost per interrupted power in kW, OR

   $/customer Cost per interrupted customer, OR

   $ Absolute cost.
6. Enter values for the Time Tariff (right click and 'Append rows' as required).
7. Press **OK** to return to the load element reliability page.
8. Optional: enter a scaling factor for the Tariff.

Example Time Tariff

An example Time Tariff characteristic is shown in Figure 18.5.1. In this example, 'Approximation' is set to 'constant', i.e. no interpolation between data points, and 'Unit' is set to $. An interruption to a load for a duration of 200 minutes would lead to a cost of $20, irrespective of the active power consumption.

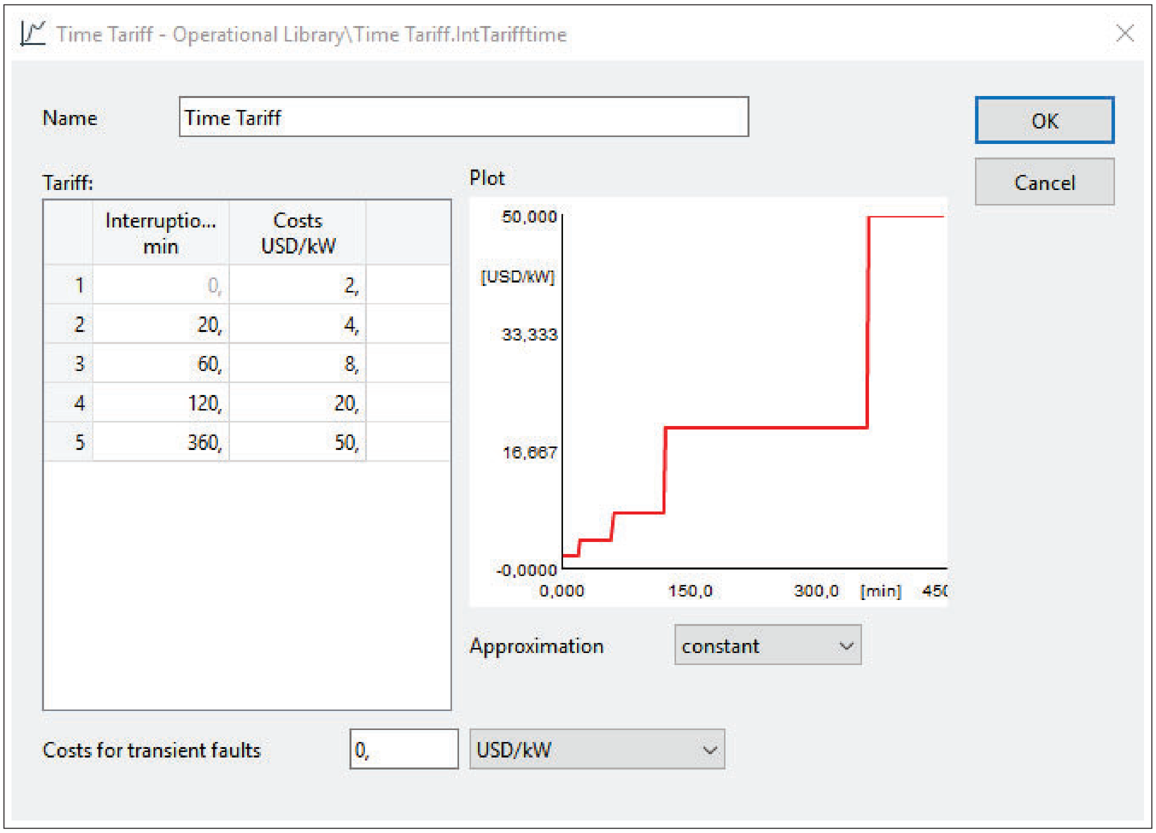


Figure 18.5.1: Example Time Tariff

18.5.2 Defining Energy Tariffs

An energy tariff characteristic can be defined by taking the following steps:

1. Choose the 'Select' option from the 'Tariff' selection control on the reliability page of the load element. A Data Manager browser will appear with the 'Equipment Type Library' selected.
2. Optional: If you have previously defined a 'Tariff' characteristic and want to re-use it, you can select it now. Press **OK** to return to the load element to reliability page.
3. Create an energy tariff object by pressing the *New Object* button from the data browser toolbar. A type creation dialog should appear.
4. Select 'Energy Tariff' and press OK. An 'Energy Tariff' dialog box will appear.
5. Enter Energy and Costs values for the Energy Tariff (right click and 'Append rows' as required).
6. Press **OK** to return to the load element reliability page.
7. Optional: enter a scaling factor for the Tariff.

Example Energy Tariff

An example Energy Tariff characteristic is shown in Figure 18.5.2. In this example, 'Approximation' is set to 'constant', i.e. no interpolation between data points. A fault which leads to energy not supplied of 2.50 MWh would result in a cost of

$$\$9{,}20 \cdot 2{,}50 \cdot 1000 = \$23000 \tag{18.1}$$

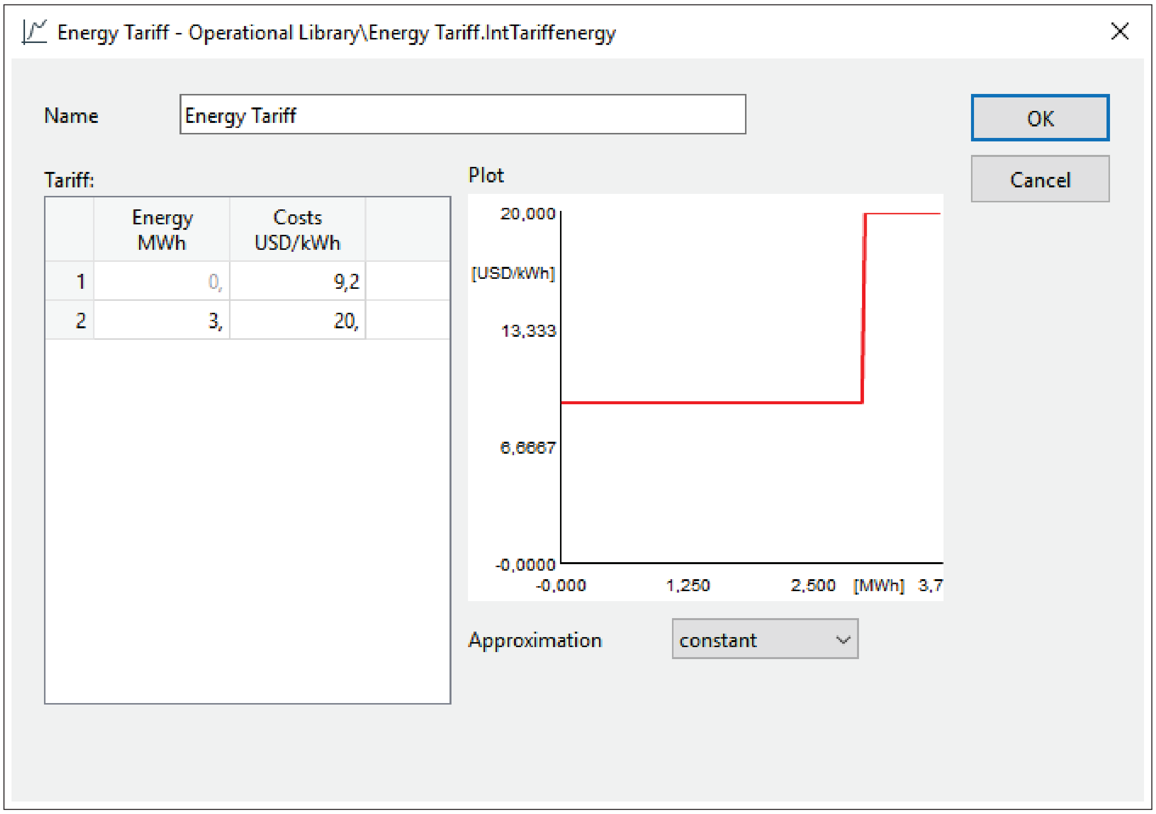


Figure 18.5.2: Example Energy Tariff

Chapter 19

Reporting and Visualising Results

19.1 Introduction

This chapter introduces the tools and options in *PowerFactory* for presenting the calculation results. Key concepts in this topic are Result Boxes, Output Reports, Results Objects, Variable Selection and Plots. The structure of this chapter is as follows:

- Section 19.2 provides the instructions for customising the result boxes displayed in the single-line, overview and detailed diagrams. Instructions about selecting the predefined formats are given in chapter Network Graphics, section 9.5.
- Section 19.3 describes the *Variable Selection* object, which is used to define the variables to be presented, either in the Result Boxes, *Flexible Data* page or Results Files.
- Section 19.4 describes the predefined reports available in *PowerFactory* to present data in the output window.
- Section 19.5 describes the option to compare steady state calculations results.
- Section 19.6 describes the *Results File* object to store results or selected variables.
- Section 19.7 lists and describes all the plot types available in *PowerFactory* and the tools used to modify/customise them.

19.2 Result Boxes

Results are displayed with help of result boxes in the single line diagrams. Several predefined formats can be selected, as described in Chapter 9, Section 9.5 (Result Boxes, Text Boxes and Labels).

The result box itself is actually a small output report, based on a form definition. This form definition is used to display a wide range of calculated values and object parameters, and can be also be used to specify colouring or user defined text.

19.2.1 Editing Result Boxes

To edit result boxes the so-called “Format” dialog is used. In this dialog, text reports can be defined, from very small result boxes to more complex and comprehensive reports within DIgSILENT *PowerFactory*.

The Format object (*IntForm*), shown in Figure 19.2.1, will be used in most cases to change the contents of the result boxes in the single line graphic; the Format dialog is accessed by right clicking on a result box and selecting the option *Edit format for. . .* .

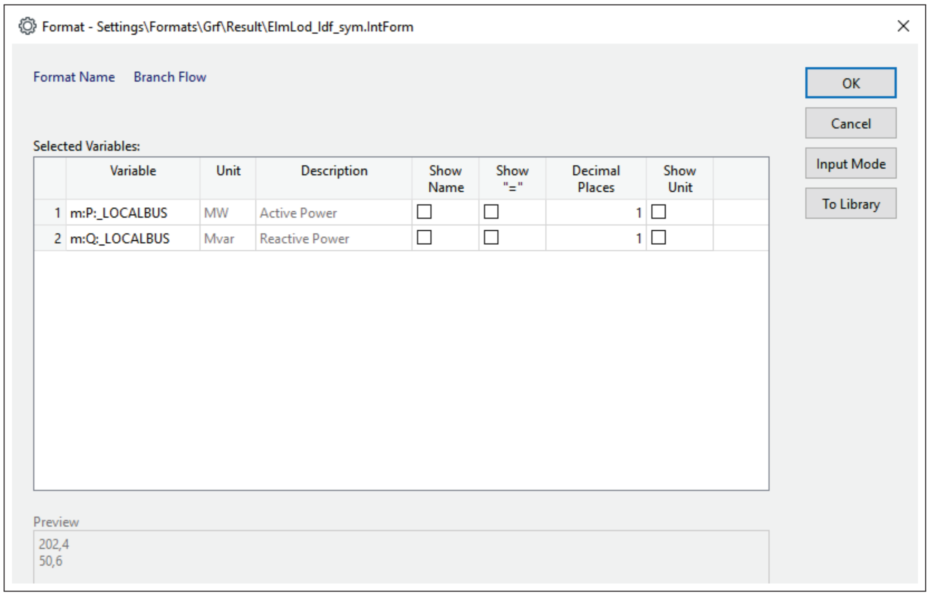


Figure 19.2.1: The Format dialog

The format defined in this dialog can be saved for later use by clicking on the button **To Library** and defining a user-specific name for it.

Such saved formats are stored in the user's own settings folder, and can therefore be selected for use in any of the user's projects.

This Format dialog has a page to change the format by selecting variables and a page to manually define a format. What is displayed on this page depends on the input mode; that can be changed using the button **Input Mode**. Both options are described in the following sections.

19.2.1.1 Input Mode - User Selection

When using this input method it is possible to select any number of parameters out of all available parameters for the selected object or class of objects. This includes model parameters as well as calculated values.

Different variables can be added by appending new rows. By double clicking on the corresponding row in the column *Variable*, a Variable Selection showing the list of all available variables will appear. More information about Variable Selection is available in section 19.3.

It is also possible to define how the variable will be shown by selecting the columns *Show Name*, *Show "="*, *Decimal Places* and *Show Unit*. A preview of the result box is shown in the *Preview* field.

19.2.1.2 Input Mode - Format Editor

This is the most flexible, but also the most difficult mode. In this mode, any text and any available variable, in any colour, can be entered in the Form. The highly flexible *DIgSILENT* output language allows for complex automatic reports. The **User defined** button acts like the input mode *User Selection* with one important difference: where the *User Selection* mode is used to redefine the complete form text, the **User defined** button appends a line for each set of variables to the existing form text.

For example if the active and reactive power of an element have been selected using the input mode *User Selection*, when switching to *Format Editor* the variables will be shown in the *DIgSILENT* output language code like this:

```
#.## $N,@:m:P:_LOCALBUS
#.## $N,@:m:Q:_LOCALBUS
```

This example shows the basic syntax of the *DIgSILENT* output language:

- The '#' sign is a placeholder for generated text. In the example, each line has a placeholder for a number with two digits after the decimal point ('#.##'). The first '#'-sign stands for any whole number, not necessary smaller than 10.
- The '$N' marks the end of a line. A line normally contains one or more placeholders, separated by non-'#' signs, but may also contain normal text or macro commands.
- After the '$N', the list of variable names that are used to fill in the placeholders have to be added. Variable names must be separated by commas. Special formatting characters, like the '@:'-sign, are used to select what is printed (i.e. the name of the variable or its value) and how.

The Format Editor offers options for the unit or name of the selected variable. If the *Unit-show* option is enabled, a second placeholder for the unit is added:

```
#.## # $N,@:m:P:_LOCALBUS,@:[m:P:_LOCALBUS
#.## # $N,@:m:Q:_LOCALBUS,@:[m:Q:_LOCALBUS
```

The '['-sign encodes for the unit of the variables, instead of the value.

The same goes for the variable name, which is added as

```
# #.## $N,@:~m:P:_LOCALBUS,@:m:P:_LOCALBUS
# #.## $N,@:~m:Q:_LOCALBUS,@:m:Q:_LOCALBUS
```

where the "∼" -sign encodes for the variable name. With both options on, the resulting format line

```
# #.## # $N,@:~m:P:_LOCALBUS,@:m:P:_LOCALBUS,@:[m:P:_LOCALBUS
```

will lead to the following text in the result box:

```
 P -199,79 MW
```

Other often-used format characters are '%', which encodes the full variable description, and '&', which encodes the short description, if available.

For a detailed technical description of the report generating language, see Appendix D (The *DIgSILENT* Output Language).

19.3 Variable Selection

Variable Selection (*IntMon*) objects are used to select and monitor variables associated with objects in the data model. The variable selection object can be used to select the variables to be recorded during a calculation (e.g. RMS/EMT Simulation, Quasi-dynamic Simulation, Harmonic Analysis) and to define the variables to be displayed in the result boxes and in the *Flexible Data* (see section 10.6).

The variable selection dialog is shown in Figure 19.3.1. The object for which the variables are defined is marked in red, the calculation to which these variables belong to in green, and the selected variables in blue.

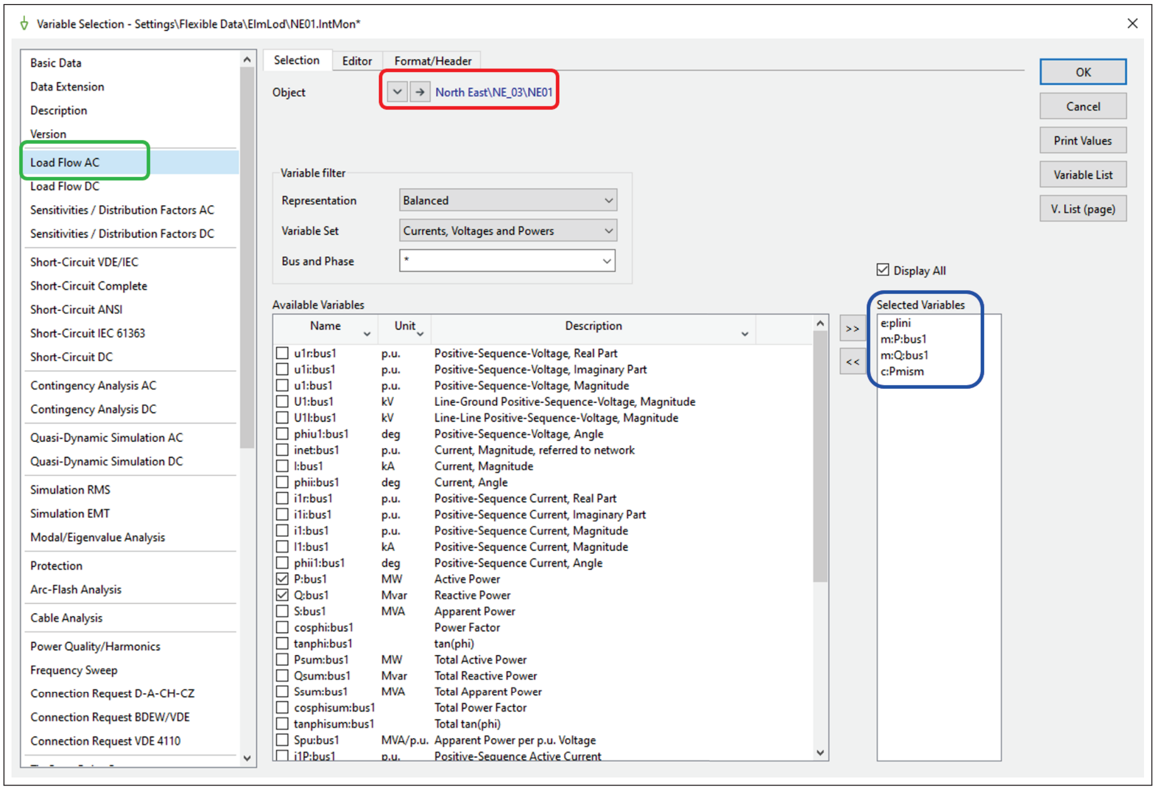


Figure 19.3.1: Example of a variable selection dialog

The variable selection object contains the following fields:

- **Object:** the object (normally a network component), whose variables are going to be monitored.
- **Class Name:** if no object has been selected the *Class Name* field becomes active. The use of the class name instead of the object for variable definition is only valid for some calculation types (e.g Quasi-dynamic simulation).
- **Display Values during simulation in output window:** this is only visible when the variable selection is being done for a time domain simulation (RMS/EMT). By checking this box and selecting the option *Display results variables in output window* in the simulation command, the values calculated for the selected variables during a simulation will be displayed in the output window.
- **Page:** the first step is to select the page on the left-hand side of the dialog, because the variables are sorted by calculation function (e.g. Basic Data, Load Flow AC, Short-Circuit DC, etc.).
- **Variable Filter:** within this panel further filtering options are available:
  - **Representation**: depending on the type of calculation to be monitored (balanced or unbalanced), it is possible to toggle between balanced and unbalanced variable selections.
  - **Variable Set**: the variables are further sorted into variable sets, which are described below in the subsection 19.3.1.
  - **Bus and Phase**: this filter allows the selection of individual phase information where appropriate, or the selection of bus for multi-port elements.
- **Available Variables:** all the variables available for display, according to the above selections, are listed here. To help the user to find the required variables, a drop-down arrow on each column header allows the list to be filtered further.

- **Selected Variables:** the selected variables. Variables are placed here by double clicking on them on the *Available Variables* side, by selecting their checkbox, or by selecting them and then pressing the (») button. Variables can be removed from the *Selected Variables* area by double-clicking on them or by selecting them and then pressing the («) button.
- **Display All:** if this box is checked then all of the selected variables are shown in the *Selected Variables* area. If not checked, the filter selected in the *Filter for* field will also apply to the *Selected Variables* area and only those selected variables in the filtered set will be shown.

The following buttons are available on the right side of the dialog:

- **Print Values**: the current values of all the selected variables are displayed in the output window.
- **Variable List**: a list of all available variables is printed in the output window.
- **Variable List (Page)**: a list of all the available variables for the current page (e.g. *Basic Data*) is displayed in the output window.

The second tab of the Variable Selection dialog goes to the *Editor*, where variables can be manually input. If the variable selection dialog is used to define the *Flexible Data* page, an additional tab called *Format/Header* is visible; more information about this tab is available is section 10.6.1: Customising the Flexible Data Page.

19.3.1 Variable Selection Filter

The first sorting of the variables is by calculation function. Within these sets variables are sorted into sub-sets. The desired subset can be selected using the drop down menu on the *Variable Set* field. These are the available subsets:

- **Currents, Voltages and Powers:** almost self explanatory - these are the outputs as calculated by a calculation function. The variable is preceded by “m:” (representing ’monitored’ or ’measured’) as in `m:P:bus1` for the active power drawn by the load.
- **Bus Results:** variables for the bus/es where the element is connected (usually preceded by “n:” for ’node’). An element having only one connection to a bus, will obviously only have results for “Bus1.” An element having two connections will have “Bus1” and “Bus2”. This means that the results of objects connected to the object whose variable list is compiled can be accessed.
- **Signals:** variables that can be used as interface between user defined and/or *PowerFactory* models (inputs and outputs). They are preceded by “s:”. These should be used when creating a controller or in a DPL script. These variables are accessible whilst an iteration is being calculated, whereas the other variables sets are calculated following an iteration.
- **Calculation Parameter:** variables that are derived from the primary calculations (i.e. currents, loading, power, losses, etc.), from input data (i.e. the absolute impedance of a line, derived from $impedance/km * linelength$), or that have been transformed from input data to a format useful for calculation (actual to per unit), or that are required for such transformation (e.g. rated power). The parameters that actually are available depend on the object type. Calculation parameters are preceded by a “c:”.
- **Element Parameter:** input parameters that belong directly to the object selected (preceded by “e:”).
- **Type Parameter:** input parameters from the corresponding type object that are linked to the element object under consideration; for example, the current rating of a line type that a line element is using.
- **Reference Parameter:** these are variables from objects that are linked or connected to the object under consideration (preceded by “r:”). For example, a line element may be part of a line coupling and the reference parameter will allow us to display the name of the coupling element.

Array variables: certain models within *PowerFactory* may support arrays (for more information, refer to the Technical Reference documentation of a specific model or to the Modelica dynamic models described in Section 30.8). In such cases, vector variables use 0-based indexing, with the exception of *Modelica Models* which internally use 1-based indexing. Without exception, *PowerFactory* always uses 0-based indexing with respect to storing variable results (i.e. *Results* objects). The syntax is: name of signal suffixed by colon suffixed by index e.g. "s:gate:0" (the first index of signal "s:gate") or "s:Ucap:20"(the twenty-first index of signal "s:Ucap").

For general use it is sufficient to simply select the variables required and transfer them to the selected variables column. To find a particular variable requires some knowledge of where the variables are stored in the object under consideration.

Additional information about how the result variables are calculated is available in section Technical References of Result Variables of the Technical References Document.

User defined variables are shown when the page *Data Extensions* is selected. Additional information about Data Extensions is available in Chapter 20.

19.4 Output Reports

PowerFactory offers two types of reports which are printed in the output window. The *Documentation of Device Data* prints either all or a part of the data entered in *PowerFactory* the *Output of Calculation Analysis* prints the results of a previously executed calculation in the output window.

19.4.1 Documentation of Device Data

The *Output of Device Data* command (*ComDocu*) can be accessed by clicking on the icon on the main tool menu.

19.4.1.1 Documentation of Device Data - Settings

The Short Listing

The "Short Listing" reports only the most important device data, using one line for each single object, resulting in concise output. Like the "Output of Results", the "Short Listing" report uses a form to generate the output. This form can be modified by the user. When the report form is changed, it is stored in the "Settings" object of the active project, so does not influence the reports of other projects. The output of objects without a defined short listing will produce warnings like:

```
Short Listing report for StoCommon is not defined.
```

The Detailed Report

The detailed report outputs all device data of the elements selected for output. In addition, type data can be included ("Print Type Data in Element"). Device Data is split into the different calculation functions like "Load-Flow" or "Short-Circuit". The "Basic Data" is needed in all the different calculations. "Selected Functions" shows a list of the functions whose data will be output. To report the device data for all functions, simply move all functions from left to right. If "Selected Functions" is empty no device data will be output.

Device Data

- **Use Selection:** the set of reported elements depends on the *Use Selection* setting. If *Use Selection* is checked one element or a set object must be chosen for output. If *Use Selection* is not checked, the *Filter/Annex* page specifies the set of elements for the report. Another way to select object for the report is to right-click on the objects from the Data Manager or the single line graphics and select *Output Data* → *Documentation*, this will open the *Documentation of Device Data* command.

- **Annex:** each class uses its own annex. There is either the default annex or the individual annex. To use the default annex check *Use default Annex*. Changes of the annex are stored in the *Settings* of the active project. The local annex is stored in the *Documentation of Device Data* command. To modify the local annex press the **Change Annex** button.
- **Title:** most reports display a title on top of each page. The reference *Title* defines the contents of the header.

19.4.1.2 Documentation of Device Data - Filter/Annex

If one wants to report elements without defining a set of objects, *Use Selection* on the *Device Data* page must not be checked. The objects in the list *Selected Objects* will be filtered out of the active projects/grids and reported. *Available Objects* shows a list of elements which can be added to the *Selected Objects* list. The list in *Available Objects* depends on the *Elements* radio button. Elements in the left list are moved to the right by double-clicking them. The text in the *Annex* input field will be set as default annex for the selected class.

The Annex for Documentation

The *Annex for Documentation* stores the annex for the documentation of results. The annex number and the page number for the first page are unique for each class.

- **Objects:** this column shows the different classes with their title.
- **Annex:** this column stores the annex number shown in the Annex field of the report.
- **First Page:** this column defines the start page for the class in the report. The first page number depends on the class of the first element output in your report. The page number of its class is the page number of the first page.

19.4.2 Output of Results

The command *Output of Results* (*ComSh*) is used to produce an output of calculation results. The output can be used in reports or may help in interpreting the results and is accessed by clicking on the icon from the main tool menu.

Several different reports, depending on the actual calculation, can be created. The radio button on the upper left displays the different reports possible for the active calculation. Some reports may be inactive, depending on the object(s) chosen for output.

On the Advanced tab, the *Used Format* gives access to the format(s) used for the report.

Some reports are a set of different outputs. For these reports more than one form is shown. If the form is modified it will be stored automatically in the *Settings* folder of the active project. The changed form does not influence the reports of other projects. If *Use Selection* is active, a set of objects (selection) or a single object must be chosen. The report is generated only for these elements. All relevant objects are used if *Use Selection* is not selected. The relevant objects depend on the chosen report. Most reports display a title on top of each page. The reference *Title* defines the contents of the header.

For some reports additional settings are required. These settings depend on the chosen report, the selected objects for output and the calculation processed before. The calculation (left top) and the used format(s) (right top) are always shown.

One option for Load Flow calculations is the *Power Interchange* report. If this option is selected, a tabular report is generated, showing the power interchange between various parts of the network, i.e. Grids, Areas or Zones. When reporting Power Interchange for Grids, a further feature is available: If the several connected networks are modelled, and the connections are represented by boundary nodes held in a dedicated boundary grid, the user may not be interested in power interchanges with the boundary grid itself but rather in the power interchanges between the grids either side of it. Therefore there is now an option provided for the ElmNet object, to mark it as a “Fictitious border grid”. If this is done, then it becomes transparent from the point of view of this Power interchange report, which then reports the interchange between the grids of interest.

19.5 Comparisons Between Calculations

At many stages in the development of a power system design, the differences between certain settings or design options become of interest. For a single calculation, the “absolute” results are shown in the single line graphics and in the flexible data page of the elements.

When pressing the *Comparing of Results on/off* button (), the results of the calculation are “frozen”. Subsequent calculations results can then be shown as deviations from the first calculation made. The subsequent calculation results are stored together with the first result. This allows the user to re-arrange the comparisons as desired by pressing the icon.

The differences between cases are coloured according to the severity of the deviation, making it possible to recognise the differences between calculation cases very easily.

The set of calculated comparisons may be edited to select the cases which are to be compared to each other or to set the colouring mode. When the icon on the main toolbar is pressed, the *Compare* dialog will open.

With the *Compare* dialog, the two cases which are to be compared can be selected. Furthermore, a list of colours may be set which is then used to colour the results displayed in the result boxes, according to certain levels of percentage change.

19.6 Results Objects

The results object (*ElmRes*,) is used by *PowerFactory* to store tables of results. The typical use of a results object is in writing specific variables during a transient simulation, or during a data acquisition measurement. Results objects are also used in scripts, contingency analysis, reliability calculations, harmonic analysis, etc.

The results object edit dialog shows the following fields:

- **Name:** the name of the results object
- **File path:** is the path where the results file is saved inside the database
- **Last Modification:** date when the results file was changed the last time
- **Default for:** the default type of calculation
- **Info:** information about the currently stored data including:
  - the time interval
  - the average time step
  - the number of points in time
  - the number of variables

  - the size of the database results file

- **Trigger-Times:** trigger times (in case of a *Triggered* default use)

The **Clear Data** button will clear all result data.

Note: Clearing the data will delete all calculated or measured data in the results file. It will not be possible to restore the data.

The default type settings are used for two purposes:

1. Creating a new results object and setting the default type to Harmonics, for instance, will cause the harmonics command dialog to use this results object by default.
2. Setting the Default type to *Triggered* will cause the calculation module to copy and temporarily store signals in that copied results object, every time a Trigger Event becomes active. The *Triggered* default type enables the trigger time fields.

When the **Output Protocol** is pressed, all events that happened during the simulation, recorded by the results object, will be written again into the output window. So one can check which events took place during the last simulation.

The contents of a results object are determined by one or more monitor Variable Selection (*IntMon*) objects. These monitor objects can be edited by pressing the **Variables** button. This will show the list of monitor sets currently in use by the results object.

Selecting a set of result variables, using monitor objects is necessary because otherwise all available variables would have to be stored, which is practically impossible.

By clicking on the **Variables** button, the list of recorded variables is displayed, if the list is empty a new variable selection can be added by clicking on the *New Object* icon (). More information about the definition of variable selections is available in section 19.3.

19.6.1 Exporting Results

The stored results for the monitored result variables can be exported by pressing the **Export** button in the results object. This will activate and open the *ASCII Result Export* command, which enables the definition of the format and the file type used to export the results.

19.6.1.1 Results Export - Basic Options

On this page the Results File and its information is displayed, and the type of export to be executed can be defined.

Export to

The following options are available:

- Output window
- Windows clipboard
- Measurement file (*ElmFile*)
- ComTrade
- Textfile

- PSSPLT Version 2.0
- Comma Separated Values (*.csv)
- Database

If the last option (Database) is used, there is a requirement to configure the access to the database via an ODBC Database Configuration object (*.SetDatabase), or select an existing database configuration. These database configurations can exist within the project, in a folder, or in a configuration area, for example. If a new database configuration is to be defined, these are the steps:

- Navigate to the chosen location for the new object.
- Click the *New Object* button
- Select the database system and the ODBC driver. Oracle, PostgreSQL and SQL Server are supported.
- Enter the access details (Username and password) for the database system.
- Enter the database name.

Variable selection

By default, the option *Export all variables* is selected, which mean that all the results for all monitored variables are exported. But also a selection of variables can be made by selecting the option *Export only selected variables*.

19.6.1.2 Results Export - Advanced Options

On this page, additional options such as the individual step size and the columns headers of the results file for the export can be defined.

Export

- Values: the results values will be exported
- Variable description only: the description of the recorded variables is exported. This is useful for reviewing the stored data.
- Object header only: also useful for reviewing the recorded data; will only export the columns headers.

Interval

A *User defined interval* for the time/x-scale can be set as the minimum and maximum value of the first recorded variable (in time domain simulations this is of course the time).

Shift time

When this box is checked, a *new start time* can be defined. This will “move” the results to the starting time.

Column header

Here is possible to customise the column header to be exported not only for the element (e.g. name, path, foreign key), but also for the variable (e.g. parameter name, short or long description)

19.7 Plots

Plots are used for displaying results graphically. The most common use of a plot is to show the results of a time-domain simulation such an EMT or RMS simulation, but there are various other applications, for example to graphically display voltage profiles, results of a harmonic analysis, results of modal analysis, etc. These could be in the form of a bar graph, a plotted curve, single displayed variables, tables of values, etc.

All signals, parameters, variables or other values from *PowerFactory* can be shown in a plot. The variables are normally floating point numbers, but it is also possible to show discrete variables and binary numbers, for example an *out of service* flag or the switching operation of a circuit-breaker.

The plots are inserted using the *Insert Plot* icon from the main menu ( ), which will open the insert plot dialog, shown in figure 19.7.1.

Once a plot has been created, it is held in the Graphics Board of the active study case. If the user closes the plot using the x on the tab, or by right-clicking on the tab and selecting *Close page*, the plot is still retained. It can be re-opened from the Window menu of the main toolbar, by selecting the option *Open Plot Page*. If the user wants to delete the plot completely, this is done by right-clicking on the tab and selecting *Delete page*.

There are various designs of plot available. The plots can be filtered by the functions where they are normally used. Some plots are typically used for more that one category (e.g. curve plots) and some are meant to be used for specific functions (e.g. correlation plot, time-overcurrent plot). All the plots are listed under the category *(All)* and the recently used in category *(Recent)*.

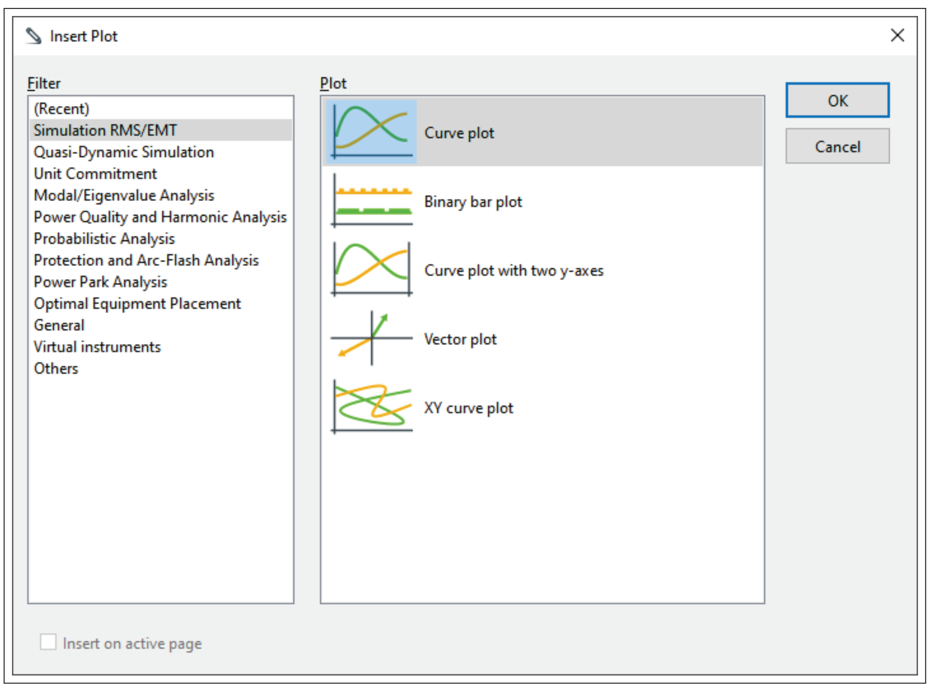


Figure 19.7.1: Insert Plot dialog

All the available plots are described in section 19.7.9 or in the corresponding chapter (for calculation-specific plots).

The plots have several areas that should be edited separately:

- The curve area (*Data Series*) that can be edited either by right-clicking on the plot and selecting

Edit Shown Data..., or by double-clicking on the area where the curve is displayed. The functions, common to all the plot types, are described in the Data Series section (19.7.2).

- The plot area, described in section 19.7.1, that can be edited by double-clicking on an empty area of the plot page (e.g. close to the border).
- The axes and gridlines, accessible by right-clicking on an axis and selecting *Edit Axis...*, or by double-clicking on it. These are described in section 19.7.4.
- The legend, described in section 19.7.5 and accessible by right-clicking on an axis and selecting *Edit Legend...*, or by double-clicking on it.

The tools available for modifying plots, such as labels and constants, can be applied equally to most plot types and are described in section 19.7.6.

The plots can be exported by selecting the option *File* → *Export*→ *Diagram...* from the main menu or using the *Diagram Export* icon on the plots toolbar.

The following formats are supported:

- Portable Document Format (*.pdf)
- Enhanced Windows Metafile (*.emf)
- Scalable Vector Graphics (*.svg)
- Portable Network Graphics (*.png)
- Tag Image File Format (*.tif, *.tiff)
- File Interchange Format (*.jpg, *.jpeg, *.jpe, *.jfif)
- Windows Bitmap (*.bmp)
- Windows Metafile (*.wmf)
- Graphics Interchange Format (*.gif)

Use of colouring in plots

The use of colouring in *PowerFactory* is described in section 4.7.5. This includes a description of colour palettes, which are particularly relevant for plots. Although the user has the option of selecting the colour of each curve individually, the simpler option is to select a colour palette. Then, *PowerFactory* will automatically assign colours to the curves in a plot using the colours from the assigned palette, making it easy to achieve the required overall appearance.

The colour palette can be changed by right-clicking on the plot area and selecting the option *Apply Colour Palette...* from the context menu. When a new colour palette is selected, the colours of the curves will all be changed automatically.

19.7.1 Plot Area

The plot dialog summarises all the information on the plot, providing the links to all the parts of the plot. In addition, the style and layout of the area outside the curve can be personalised. It can be edited by double-clicking on an empty area of the plot page (e.g. close to the border).

19.7.1.1 Basic

On this page, the name of the plot can be modified. In addition, the following links are available:

- ***Curves:*** a link to the Data Series, described in section 19.7.2.

- ***Title:*** clicking on the select arrow opens the edit dialog of the plot title with the following fields:
  - Title: to define visibility and displayed name
  - Positioning and Layout: to define the location of the title of the plot
  - Text Format: to define the font size and colour
  - Border and Background: to define the style of the title
  - Floating position: if the title is moved from the default position, a floating position can be set in this field.
- ***Legend:*** a link to the plot legend dialog, described in section 19.7.5.

19.7.1.2 Axes

The plot axes often needs to be synchronised for all plots in the Study Case or for all plots on one plot page, for instance to show the same time-scale in all plots.

In this page, the sharing of the axis can be defined. This can also be done by right-clicking on the axis and selecting → *Axis sharing*.

The sharing options are:

- **Local:** the axis scale, labelling and format are only valid for the local plot.
- **Page:** all the plots on the plot page will use the same axis scale, labelling and format.
- **Graphics Board:** the axis settings will apply to all the plots in the study case.

The scale, labelling and format options, described in section 19.7.4, can be accessed by clicking on the → button by the *Used Axis* field.

Additional axes can be added using the button **Create**.

19.7.1.3 Style and Layout

On the *Style and Layout* page, the following display options can be selected:

Border and Background

- ***Draw border:*** to define outside border and colour of the plot area, i.e. the “non-gridded area”.
- ***Fill background:*** to define the background colour of the plot area.

Plot Position on Page (mm)

Defines the position of the plot on the page. Usually this is set using the *Automatic Arrangement Commands* described in section 19.7.6.4, but it is also possible to define the position on the page using these fields.

Element Padding (mm)

Defines the size of the plot area outside the curve; this is particular useful when using a coloured background.

Axis Visibility

In this panel, the user can set the visibility of the axes.

19.7.2 Data Series

The Data Series in the base for almost all the plot types. To edit the Data Series, right-click on the plot and select *Edit Shown Data. . .* , or double-click on the area where the curve is displayed. The pages of the edit dialog are described in the following sections.

19.7.2.1 Curves

The data in the curves page is entered in the following panels:

Data Source

- ***Calculation type:*** the calculation type is set automatically when the plot is inserted, based on the category selected on the *Insert Plot* dialog. This setting automatically determines other settings for the plot (e.g. scales).
- ***Auto-search results:*** this is a reference to the currently active results file (*ElmRes*). With this selection the currently used result file of the last calculation is chosen. More information about results objects is available in section 19.6
- ***Select results individually per curve:*** this option gives the user the possibility to choose another result file. The result file should be specified in the *Curves* table.

Curves

The definition table for the curves is used to specify the results file (optional), element and variable for each curve as well as its representation. Each line in the *Curves* table is used to define one curve.

- The first column allow the visibility of the curve on the plot to be set.
- If the option *Select results individually per curve* is selected, the next column is for the results object from which the data to plot the curve will be read.
- There then follows a column for the power system element, which is selected from the available elements in the results object.
- The next column is for the actual variable for the curve, selected from the variables in the results object, belonging to the selected element.
- The following columns allow the user to specify the style of the individual curve.
- The last column, named *Label*, can be used to add additional information for the curve legend.

Note: As well as the option to edit colours individually by double-clicking on the colour square, the user has two further options which are offered in the context menu by right-clicking on the colour box. These are:

- **Apply Colour Palette**: this will apply the selected colour palette and change all curve colours accordingly
- **Auto-Assign Subsequent Colours**: if a different standard colour has been selected for the first curve (for example), the remaining curves will be assigned the subsequent colours from same palette.

Only the elements and variables stored in the results file can be plotted. Additional curves can be added by right clicking and selecting *Insert Rows* or *Append (n) Rows*. Similarly, to delete a marked curve definition from the list, *Delete Rows* can be selected.

Several elements can be selected and *PowerFactory* will automatically insert the corresponding number of rows. In the same way, several variables of the same element can be added in one step by selecting them together.

Note: The position of the curve (i.e. front or back) is automatically assigned by the row numbering, however, it is possible to change it by right clicking on the curve (on the plot) and selecting *Move Curve → Backwards/Forwards*

Plot Features

In this panel, additional features for the curves, including shape, transformation and stacking of curves are defined.

- ***Additional curve shapes:*** when this option is checked, two additional columns are shown in the *Curves* table:
  - On the column *Shape*, one of these options can be selected: Curve, Steps, Filled curve, Filled steps and Bars.
  - If one of the filled shapes is selected in the *Shape* column, the filling style can be changed in the column *Fill Style*
- ***Data transformation:*** this option allows the transformation of the data on the curve according to the following options:
  - Normalisation: the values of the variable can be normalised to a nominal value (specified in the column *Nom. Value*).
  - Binarisation: transforms the values of the curve into binary data: if a value is bigger than 0.5, it is set to 1, otherwise is to 0.
  - Functions: probabilistic functions can be used to convert the data. More information can be found in section 19.7.9.6.

Note: The options for data transformation might vary depending on the calculation type.

- ***Curve stacking:*** this option can be used if more than one curve is shown and will "pile up" the values of the curves as:
  - Values
  - Absolute values
  - Relative values
  - Percentage

Export Button

When clicking on the **Export...** button on the right of the plot, the *ASCII Result Export* command, described in section 19.6.1, with a time interval set to the time displayed on the plot can be executed to export the result values of the plotted variables.

Filter Button

It is possible to add additional filters to the curves presented in the plot; it should be noted that when a filter is defined it is applied to all the curves displayed in the plot. The *Curve Filter* command specifies the type of filter applied to the data read from the results object. The following filter settings are available:

- ***Disabled:*** no filtering will be performed.
- ***Moving average:*** the filtered curve is the running average of the last n points. The first n-1 points are omitted.
- ***Moving balanced average:*** the filtered curve is the running average of the last (n-1)/2 points, the current point and the next (n-1)/2 points. This filter thus looks ahead of time. The first and last (n-1)/2 values are omitted; n must be an odd number.
- ***Average:*** the filtered curve contains the averages of each block of n values; every n-th value is shown.

- ***Subsampling:*** the filtered curve only contains every n-th value. All other values are omitted.
- ***Deviation from initial value:*** the curves are shifted, so that for each curve the values are all relative to the first value.

Note: A curve filter can only be applied at the end of the simulation or measurement. Points added during a simulation or measurement are not filtered.

Processed and Aggregated Results

The curve plots have the option to define a user defined signal. This option allows calculation of additional results based on the arithmetic manipulation of one or more results calculated by *PowerFactory* and recorded in a results object (*ElmRes*). A new user defined signal, can be defined by clicking on the **Create...** button on the edit dialog of the curve plot. An example of the calculated result dialog is shown in Figure 19.7.2.

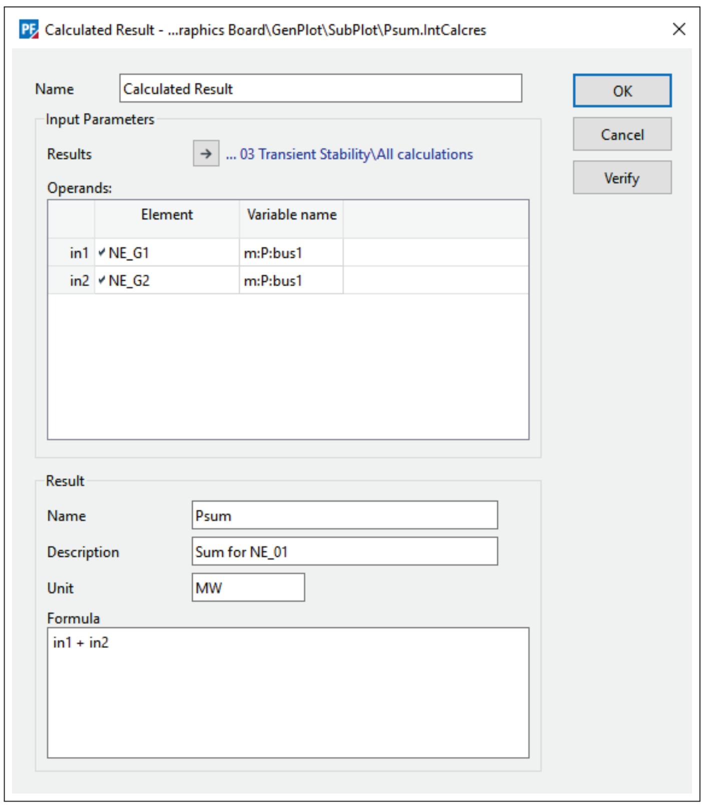


Figure 19.7.2: The calculated results object

The calculated results object dialog includes the following fields:

- ***Name:*** the name of the calculated results object
- ***Input Parameters***
  - Results: defines the results object in which the arithmetic operands are located.
  - Operands: defines the elements and variable names of the operands within the results object. Additional operands can be inserted or appended by *Right-Click* → *Insert Row(s)* or *Append (n) Row(s)*.
- ***Result***
  - Name: defines the name of the user defined curve

- Description: a free text field for description of the curve
- Unit: user defined variable unit
- Formula: DSL expression for arithmetic calculation; operands are defined in accordance with the naming convention in the *Input Parameters* field i.e. in1, in2, in3 etc.

More information about the DSL syntax is available in section 30.4.

The buttons **Manage...** and **Add to table...** can be used for editing the defined variables and adding them to the *Curves* table.

19.7.2.2 Style and Layout

On the *Style and Layout* page, the following additional display options can be selected:

Curve Area Position (mm)

If *Auto-Position Curve Area* is selected, the position of the curve is automatically set to fit the page size. If it is deselected, the border of the plot will appear as shown in figure 19.7.3 and the user can set the size and position of the plot.

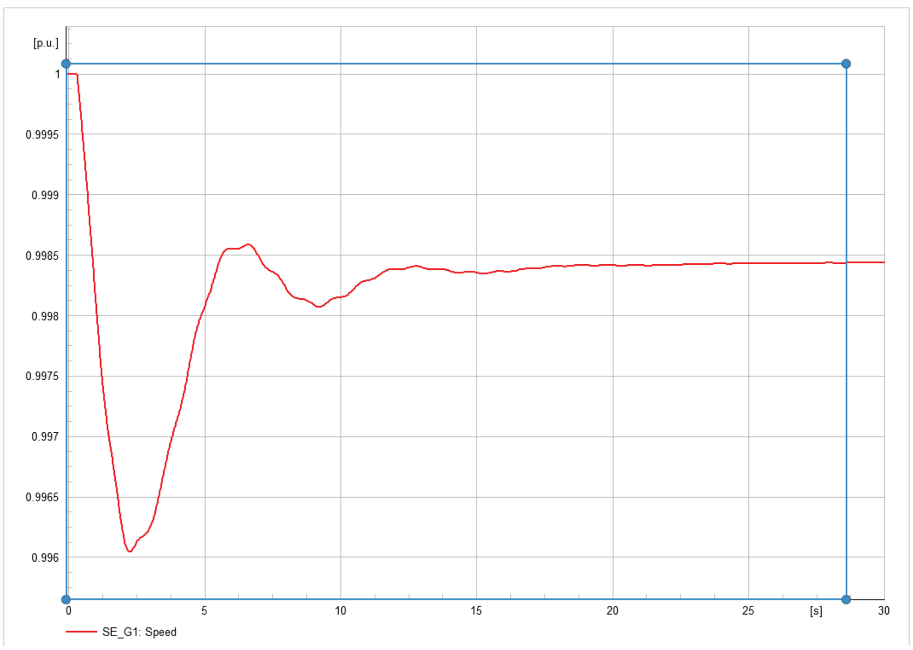


Figure 19.7.3: Resizing a Curve Plot

Border and Background

- ***Draw border:*** to define outside border and colour of the curves area.
- ***Fill background:*** to define if the curves area should be filled, and select the colour.

19.7.3 Complex Data Definition

The complex data definition is mainly used for the vector plot. In order to edit the complex data definition, right-click on the plot and select *Edit Shown Data...*, or double-click on the area where the curve is displayed. The pages of the edit dialog are described in the following sections.

19.7.3.1 Variables

The data in the curves page is entered in the following panels:

Data Source

- ***Element Variables:*** if the plot is inserted by right clicking on the element, the element and complex variable are automatically defined on the Variables table. Otherwise the element and complex variable can be selected by double clicking on the corresponding fields.
- ***Result File:*** this option enables the user to add the results file (*ElmRes*). More information about results objects is available in section 19.6.
- ***Select results individually per curve:*** this option gives the user the possibility to choose another result file. The result file should be specified in the *Variables* table.

Variables

The Variables definition table for the curves is used to specify the results file (optional), element and complex variable for each plot as well as its representation. Each line in the *Variables* table is used to define one plot.

- The first column allow the visibility of the curve on the plot to be set.
- If the option *Select results individually per curve* is selected, the next column is for the results object from which the data to plot the curve will be read.
- There then follows a column for the power system element, which is selected from the available elements in the results object.
- The next column is for the complex variable for the plot, belonging to the selected element.
- The following columns allow the user to specify the style of the individual plot.

Note: The assignation of colours and styles can be done automatically by right clicking on the colour/style and selecting *Auto Assign Subsequent Colours /Line Styles/Line Widths*.

- The last column, named *Description*, can be used to add additional information for the plot.

Additional curves can be added by right clicking and selecting *Insert Rows* or *Append (n) Rows*. Similarly, to delete a marked curve definition from the list, *Delete Rows* can be selected.

Several elements can be selected and *PowerFactory* will automatically insert the corresponding number of rows. In the same way, several variables of the same element can be added in one step by selecting them together.

Time Point Selection

Once the result file is added to the plot, *Time Point Selection* option can be used to investigate the results at different points in time. Furthermore, the cursor can be added to the plots in order to keep a track of the result variables in both the vector and dynamic simulation plots simultaneously.

Plot Features

- ***Vector labels:*** the vector labels provides the user the possibility to update the labels of the vectors in the plot according to the requirements. If *None* is selected then no labels will be displayed. The *Automatic* option automatically detects the labels based on the complex variables representation. Moreover, the user can explicity set the labels in polar or cartesian system using *Coordinates, Polar* or *Coordinates, Cartesian* respectively. The *Phase Only* option can be used to display the labels based on the respective phases of each vectors.
- ***Draw arrow heads:*** the arrow heads can be drawn on the vectors by enabling this option.

- ***Vector transformation:*** using this feature, it is possible to transform a complex variable using another variable or a constant. As required, various operations such as addition, subtraction, multiplication and division in addition to shift can be applied to the complex variable. The information about these operations has to be provided in the *Variables* table. Also, the information about the operation being applied to the complex vector is visible in the plot legend. Figure 19.7.4 shows an example of the implementation of vector transformation feature in the edit dialog of the plot.

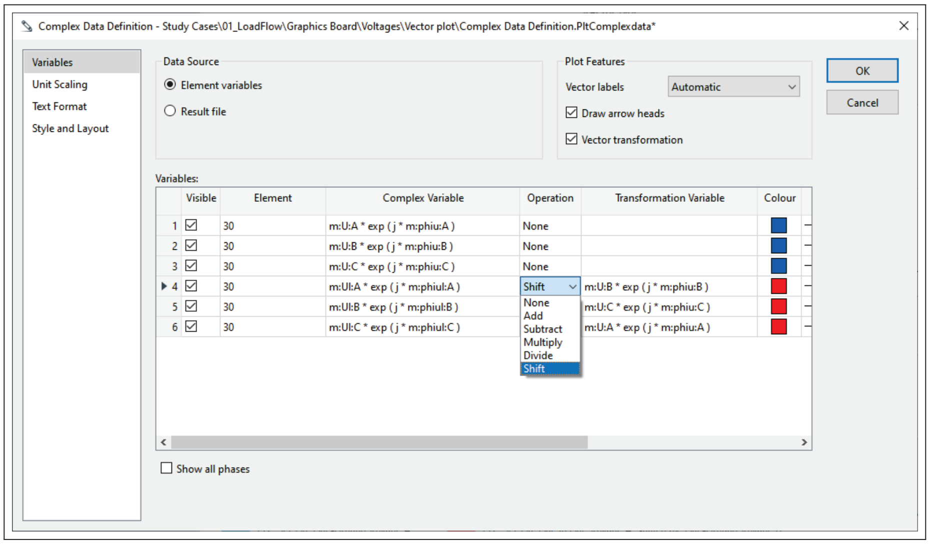


Figure 19.7.4: Different operations for vector transformation

19.7.3.2 Unit Scaling

This page allows the user to set the scaling of the units of multiple vectors being displayed in the plot. There is a possibility to determine the unit scales automatically by enabling the corresponding option. And if required, the users can define their own scales for the units to normalise the individual vectors by disabling the *Determine unit scales automatically* option.

19.7.3.3 Text Format

The label fonts and number formats can be updated using this page. In the *Label Font* panel, the following options can be set:

- Font: font size, type and effects.
- Colour: the colour of the label text.

In the *Number Format* panel, the following options can be set:

- Format: the user can select between concise, fixed decimals and scientific.
- Digits: if the option *concise* is selected, the user can set the maximum number of digits to be displayed.
- Decimals: if the option *fixed decimals* or *scientific* is selected, the user can set the number of decimal places to be shown.
- Exponent character: the user can select between E and e for the *scientific* option.

19.7.3.4 Style and Layout

On the *Style and Layout* page, the following additional display options can be selected:

Curve Area Position (mm)

If *Auto-Position Curve Area* is selected, the position of the curve is automatically set to fit the page size. If it is deselected, then the user can set the size and position of the plot.

Border and Background

- ***Draw border:*** to define outside border and colour of the curves area.
- ***Fill background:*** to define if the curves area should be filled, and select the colour.

See section 4.7.5.1 for more information about the use of colour palettes.

19.7.4 Axes and Gridlines

Each of the axes can be edited either by double click or by *Right click* → *Edit Axis*. The plot axis dialog options are described in this section.

19.7.4.1 Scale

The options presented in the *Scale* page depend on whether the edited axis is a x- or a y-axis.

Axis mode

This option is only available for the x-axis and is automatically set depending on the plot type. Alternatively, can be selected by the user. The options are:

- Default: depends on the type of simulation and the results object created during the previous simulation.
- Time: the x-axis is set to time and an additional field for the time unit is displayed.
- Date and time: used by default in quasi dynamic simulation plots, the format for the date and hours and the time period can be specified.
- Frequency: the x-axis is set to frequency and an additional field for the frequency unit is displayed.
- Discrete (net elements): used in bar plots, where network elements are shown in the x-axis

Scale Type

For the x-axis, for all the modes, except the *Date and time*, the scale can be set to linear or logarithmic. The y-axis can additionally be set to dB.

Range

The *Minimum and Maximum* limits of the axes can be set manually.

Scale to Contents

The scaling of the axis is done according to the data available, for the x-axis is by default 0 % and for the y-axis 10 %. The option *Limit minimum to origin if possible* will set the minimum to zero. The option *Scale when calculation data changes* will automatically update the scale of the axis after the calculation is executed.

19.7.4.2 Axis Labelling

On this page, the grid lines and styles are defined.

Tick Mark Positions

- Step size in data space: *PowerFactory* tries to find the ticks that would best fit with the data. For this, a reference value can be set to force the origin.
- Fixed number of tick marks: the number of ticks a fixed number of subdivisions between the minimum and maximum limits.

For both options, the option *Determine tick positions automatically* can be selected. This option determines the number of ticks depending on what would look best on the plot page. If the option is unchecked, the user can select either the step size (for the step size in data space option) or the number of ticks (for the fixed number of tick marks option).

Line Style

Options to alter the width, line style and colour of the ticks and grid lines.

19.7.4.3 Text Format

In the *Font Style* panel, the following options can be set:

- Font: font size, type and effects.
- Label colour and offset: the colour of the axis values and the distance between the axis and the text.
- Show unit: to display the unit of the axis.

In the *Number Format* panel, the following options can be set:

- Format: the user can select between concise, fixed decimals and scientific.
- Digits: if the option *concise* is selected, the user can set the maximum number of digits to be displayed.
- Decimals: if the option *fixed decimals* or *scientific* is selected, the user can set the number of decimal places to be shown.
- Exponent character: the user can select between E and e for the *scientific* option.

19.7.5 Plot Legend

The edit dialog of the legend can be accessed either by double-clicking on it or by *Right click* → *Edit Legend*. These are the options available:

Show legend

Defines the visibility of the legend.

Positioning and Layout

- ***Position:*** the position of the legend. Options are:
  - Floating
  - Top left
  - Top centre
  - Top right

  - Bottom left
  - Bottom centre
  - Bottom right
  - Left
  - Right

- ***Layout:*** the layout of the legend. Options are:
  - Automatic: based on the size of the plot and number of variables
  - Columns: organise in columns; an additional field for the number of columns is displayed
  - Horizontal
- ***Margin:*** the distance in mm to the axis/curve area.
- ***Padding:*** defines the size of the box around the legend, particulary useful when defining a background colour for the legend.
- ***Include only drawn curves:*** only the curves with results will be shown on the legend, otherwise the legend of those curves is shown greyed-out.
- ***Reverse item order:*** change the order of the legend items.

Text Format

- ***Font:*** font size, type and effects.
- ***Always show units:*** displays the units of the variables.
- ***Use short variable description:*** short description of all the variables.
- ***Line spacing:*** defines the spacing of the legend text.

Border and Background

- ***Draw border:*** to define outside border of the legend and the colour of it.
- ***Fill background:*** to define if the legend should be filled and the colour to be used.

Floating position

If the position of the legend is set to *floating*, the location of the legend in mm can be set in this field.

19.7.6 Plots Toolbar

There are numerous tools which help the user interpret and analyse data and calculation results. Most of the tools are accessible directly through plot toolbar, which is displayed when a plot is inserted. Each of the icons of the plot toolbar, shown in figure 19.7.5 and the additional context sensitive menu options are described in the following sections.

Figure 19.7.5: Plots Toolbar

19.7.6.1 Insert Plot

The icon  inserts a plot in the existing page. Clicking on this button opens the *Insert Plot* dialog, described in section 19.7.

19.7.6.2 Edit Plots on Page

The icon opens the dialog for defining curves of several plots. If the variables of only one plot are to be changed, it is suggested to edit the dialog of the plot itself by double-clicking it. This procedure is more convenient.

This dialog gives a very good overview over the diagrams on the plot page and the variables, axis and curve styles. Figure 19.7.6 shows an example of the dialog.

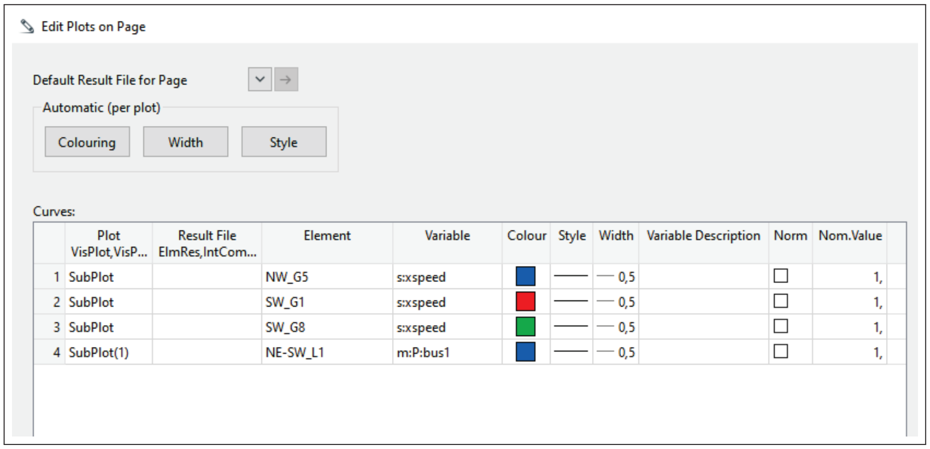


Figure 19.7.6: Editing all plots on the plot page

Each line of the table named *Curves* defines a variable shown on the panel. The variables definition applies to the plot shown in the first column. When the dialog is opened the plots are sorted from left to right and from top to bottom and are numbered accordingly.

All data and settings of each variable are displayed in the table, and the columns are used exactly like the columns in the table of a plot.

The *Default Result File for Page* can be used to set the default result file for all the plots on the page.

The buttons **Colouring**, **Width** and **Style** set those values automatically per plot.

The **Colouring** button will set the colours according to the "*PowerFactory* Standard" palette regardless of which palette the user has selected as a default for plots.

19.7.6.3 View and Select Commands

- **Rebuild:** updates the currently visible page by updating the drawing from the database.
- **Zoom In:** changes the cursor to a magnifying glass. The mouse can then be clicked and dragged to select a rectangular area of the plot to be zoomed.
- **Hand Tool:** if a zoom is applied, can be used to pan the plot.
- **Zoom All:** zooms to the page extends.
- 100% **Zoom Level:** zooms to a custom or pre-defined level.

Note: Ctrl+ mouse scrolling can be used to zoom in and out

19.7.6.4 Automatic Arrangement Commands

A plot's size and position is usually set automatically. There are two different modes for automatically arranging the plots in the plot page:

- *Arrange plots on top of each other*
- *Arrange plots automatically*

The modes can easily be changed by pressing the one or the other button. The relative positions of plots can also easily be changed: mark the plot by clicking it, then 'drag' the plot across another plot.

Note: This option of exchanging the plots by dragging is only possible when one of the arrangement buttons are active. If you deactivate both buttons by unselecting them in the toolbar, the plots can freely be moved by dragging them on the panel

---

19.7.6.5 Scale Buttons

- **Scale x-axes automatically:** scales all the x-axes according to the start and end of the results file.
- **Scale y-axes automatically:** scales all the y-axes according to the maximum and minimum values of the variables in the results file.
- **Zoom x-axis:** zooms in a certain range of the x-axis (if the axis is shared, the zoom is applied to all plots sharing the axis definition).
- **Zoom y-axis:** zooms in a certain range of the y-axis (if the axis is shared, the zoom is applied to all plots sharing the axis definition).
- **Move x-scale:** moves the position of the x-axis (is applied to all plots sharing the axis definition).
- **Stretch/compress x-scale:** modifies the x-axis scales in order to compress or stretch the shown curve (is applied to all plots sharing the axis definition).

Note: The scale buttons are inactive if there are no plots shown at all or if the x or y axes cannot be scaled automatically.

19.7.6.6 Add Curve Label

A number of *Curve Labels* tools are available in the Drawing Toolbox, which will be displayed if the user clicks on the *Add Curve Labels* button.

Figure 19.7.7: Curve Labels Tools

There are different styles of labels available for labelling curves and graphics. Setting labels is possible in most of the different plots, although some of the labels are not available in all plot types. Labels are all created in the same way.

Most of the label buttons are only active after clicking on the curve. After selecting the appropriate label from the sub-option of label, a rubber band from the cross to the mouse is shown. A click with the left mouse button sets the label, the right mouse button cancels. The following labels are available:

- **Text Label:** displays user-defined text above and below a line connected to the curve.
- **Label with Curve Value:** displays the x/y coordinates of selected point.
- **Label with Definable Format:** displays the name of the element.
- **Gradient Label:** displays the difference in x and y values (dx and dy) between two points, and also the gradient (dy/dx) and the value of 1/dx.
- **Statistic Label:** helps to analyse a curve, by labelling, for example, its extrema.
- **Delete Statistic Label:** delete existing statistic label.

Text, Curve Value and Gradient labels

The text, value and gradient labels are defined using the same object type. The *VisValue* edit dialog contains the following fields:

- *Value:* displays the connected curve position of the label. For labels created as a value-label this position is displayed automatically as label text. “x-Axis” displays the x axis value and “y-Axis” the y axis value. “Time” is only visible for plots showing a trajectory.
- *Text on Top and on Bottom:* text written above and below the horizontal line.

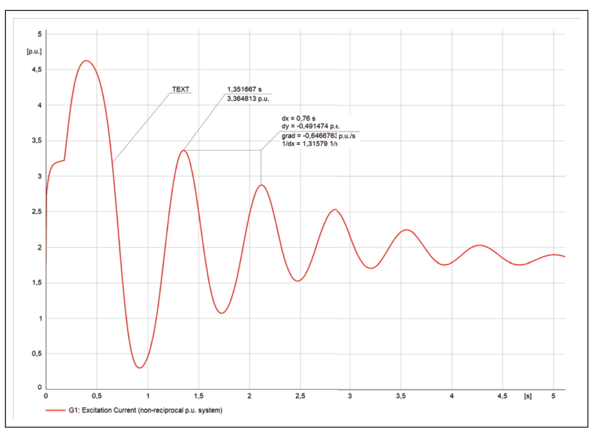


Figure 19.7.8: Text, Value and Gradient labels

Label with Definable Format

The format-label displays text printed using a form. It is typically used to show the name of the object whose variable is shown in the curve. It is useful when several curves with the same colour are plotted.

The form is different for each type of diagram. It is either defined locally per label or defined for all diagrams of the same type in the activated project. The format-label dialog is shown in the following figure.

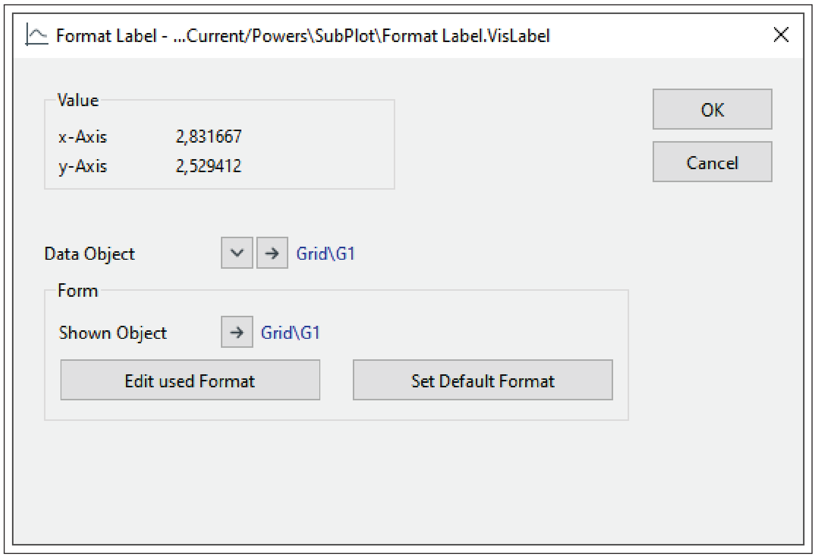


Figure 19.7.9: Edit dialog of the label with definable format

- *Value:* displays the connected curve position of the label. *x-Axis* displays the x axis value and *y-Axis* the y axis value.
- *Data Object:* reference to the object of which the plotted curve parameter is derived. If *Data Object* is not set the label itself is taken as the *Shown Object*.
- *Shown Object:* object output by the form, see *Data Object* described above.
- *Edit Used Format:* shows the used *Format Manager*. The used format is either the local format or the one defined for all plots of the same type in the active project.
- *Set Default Format:* sets the used format as the one used for all plots of the same type in the active project.

Statistic Labels

The statistic label function provides the possibility to label the following values of the curves:

- Minimum or Maximum in the visible area of the plot
- Global Minima or Maxima
- Local Minima or Maxima
- Global Average
- Average of the visible area of the plot
- Integral of the visible area of the plot
- Energy (Integral calculation available specifically for the load duration plot)
- Global statistical mean value of the probabilistic data
- Global confidence interval for mean value of the probabilistic data

To remove statistic labels from a curve, the button *Delete Statistic Labels* can be used.

19.7.6.7 Multi-Curve Tracking

Enabling “Multi-Curve Tracking” mode means that when the mouse is hovered over a point in the plot, a tooltip will appear, giving information about the current value on the x-axis as well as all y-axis values for the curves. The tooltip follows the mouse while it is moved over the plot and therefore makes it very easy to see individual results without zooming in, even if lines are on top of each other, or very close together.

19.7.6.8 Frequency Analysis

When the button is clicked, the frequency of an area of the curve can be analysed. More information about the frequency analysis is available in section 29.13.

19.7.6.9 Cursors

The buttons and can be used to add a vertical line in all the plots of the page. These cursor can be used to compare plots instead of defining separate x-constants in every plots.

19.7.6.10 Title Block

The icon shows or hides the title block of the page.

The title can be defined or changed by double-clicking on it.

For details about the settings of the title object refer to Chapter 9: Network Graphics.

19.7.6.11 Edit Plot Page

Whenever a plot is inserted, a Plot Page is automatically created, the Plot Page being one of the possible page types on a Graphics Board. The edit dialog can be opened by clicking on the icon and the pages are described below.

Basic Data

- *Name:* name of the page, also modifiable by right clicking on the tab and selecting → *Rename page*.
- *Page auto layout:* defines the layout of the plots on the page. This option is the same as using the plot arrangement buttons, described in section 19.7.6.4.
- *Curve area alignment:* aligns the plots on the page, options are:
  - Off
  - Shared axes only
  - All plots on page
- *Results:* to define the Results File object used for all the plots in the plot page. The result column of the plots need not be set for most calculations: the plot itself will look for the results element to display automatically.

Advanced

Defines the order on the Graphic Board and the group where the page belongs.

19.7.6.12 Page Format

The page format is modified using the icon. In the Page Format, the drawing size and the page format are defined. The plot page uses the page format set in the graphics board.

In addition a local page format can be created for each Plot Page by selecting the option *Create local Page Format* from the context sensitive menu.

19.7.6.13 Export Diagram

The *Export Diagram* icon opens a dialog offering options for exporting the graphic. A range of file formats is supported.

19.7.6.14 Print

The *Print Diagram* icon opens the print preview page. The printer and margins to be used can be selected in this dialog.

19.7.7 Context Sensitive Menu Tools

As well as the tools of the plot toolbar, the following additional tools are available via the context sensitive menu, displayed by right clicking on the plot.

Copy Plot to New Page

This option will create a new Plot Page containing only the selected plot.

Add Intersection Line

The intersection line is used to display a straight line, which can be the y-values for a constant x-quantity, the x-values for a constant y-quantity or a line with a slope in the form of $y = mx + b$.

The appearance of constant labels can be modified using the following settings:

- ***Style:*** changes the representation of the constant label as follows:
  - *Line Only*: displays only the solid line and the related label.
  - *Line with Intersections*: shows a solid line including label and indicates the values when intersections with the curves of the plot.
  - *Short Line Only (Left/Right/Top/Bottom)*: indicates the constant value at the bottom/top respectively at the right/left side of the plot.
  - *Short Line/Intersection (Left/Right/Top/Bottom)*: indicates the constant value at the bottom/-top respectively at the right/left side of the plot and the intersections with curves.
  - *Intersection Only*: shows only the intersection points with the curves.
- ***Label:*** either the default value (name of the variable in the axis) or defined by the user.
- ***Position:*** defines the position of the constant value label as follows:
  - *None*: displays no label at all.
  - *Outside of Diagram*: creates the label between the border of the Plot and the diagram area. Labels of constant x values are created above the diagram area, labels of constant y values to the right of the diagram area.
  - *Above Line (right)*: shows a label above the line if y is constant; the label will be on the right hand side.

- *Below Line (left)*: shows the label below the line on the left hand side.
- *Left of Line (top)*: shows a label on the left side of the line if x is constant; the label will be on the top end.
- *Right of Line (bottom)*: shows the label right of the line on the bottom end.

- ***Value:*** defines the constant value, either X or Y. The dialog shows if either an X or Y is set. Also the actual position of the cross will be shown as an x- or y-value. It is not possible to change a constant X into a constant Y label other than by removing the old label and creating the new one.
- ***Colour:*** specifies the colour of the line and the labels/intersections.
- ***Linestyle and Width:*** specifies the line style and line width for the line shown. Invisible if *Show Values* is set to *Intersections Only*.
- ***m and b:*** if the intersection line is a *Line Equation*, the m and b values should be defined in these fields.

19.7.8 User-Defined Styles

The user-defined styles are stored in a folder called *Plot Style Templates* within the *Settings* folder of the active project. A new style is created by right-clicking on the plot or on the plot page and selecting *Style* → *Save Style as...*.

The style can be changed by right clicking on the plot or on the plot page and selecting *Change Style* → *...*.

Inside the user-defined style, a template of the plot page, described in section 19.7.1, is stored and can be modified.

19.7.9 Plot Types

All the available plots are listed below, grouped by category, and described either in the following sections or in the corresponding chapter (for calculation-specific plots).

1. **Simulation RMS/EMT**
   - Curve plot (section 19.7.9.1)
   - Curve plot with two y-axes (section 19.7.9.2)
   - XY curve plot (section 19.7.9.3)
   - Vector plot (section 19.7.9.10)
2. **Quasi-Dynamic Simulation**
   - Curve plot (section 19.7.9.1)
   - Complete Generation (section 19.7.9.8)
   - Plant Categories (section 19.7.9.7)
   - Renewable and Fossil (section 19.7.9.9)
   - Duration curve (section 19.7.9.6)
   - Curve plot with two y-axes (section 19.7.9.2)
   - XY curve plot (section 19.7.9.3)
3. **Unit Commitment**
   - Curve plot (section 19.7.9.1)
   - Complete Generation (section 19.7.9.8)
   - Plant Categories (section 19.7.9.7)

- Renewable and Fossil (section 19.7.9.9)
- Duration curve (section 19.7.9.6)

4. **Modal/Eigenvalue Analysis**
   - Eigenvalue Plot (section 32.4.2.1)
   - Mode bar plot (section 32.4.2.2)
   - Mode polar plot (section 32.4.2.3)
5. **Power Quality and Harmonic Analysis**
   - Curve plot (section 19.7.9.1)
   - Harmonic distortion (section 36.5.4)
   - Curve plot with two y-axes (section 19.7.9.2)
   - Waveform Plot (section 36.5.5)
   - XY curve plot (section 19.7.9.3)
6. **Probabilistic Analysis**
   - Convergence of statistics (section 44.3.8.3)
   - Distribution estimation (section 44.3.8.5)
   - Distribution fitting (section 44.3.8.6)
   - Correlation plot (section 44.3.8.4)
7. **Protection and Arc-Flash Analysis**
   - Current comparison differential plot (section 33.10.1)
   - Curve-input (section 19.7.9.11)
   - Phase comparison differential plot (section 33.10.2)
   - R-X plot (section 33.6)
   - Relay operational limits plot (P-Q diagram) (section 33.7)
   - Short-circuit sweep plot (section 33.12)
   - Time-distance plot (section 33.8)
   - Time-overcurrent plot (section 33.4)
8. **Power Park Energy Analysis**
   - Annual load duration curve
   - Curve plot (time series)
   - Generation curves (basic analysis)
   - Losses curves (basic analysis)
   - Wind speed cumulative probability (basic analysis)
   - Wind speed probability (basic analysis)
9. **Optimal Equipment Placement**
   - Curve plot (section 19.7.9.1)
   - Complete Generation (section 19.7.9.8)
   - Plant Categories (section 19.7.9.7)
   - Renewable and Fossil (section 19.7.9.9)
   - Duration curve (section 19.7.9.6)
10. **General**
   - Curve plot (section 19.7.9.1)
   - Bar plot (section 19.7.9.4)
   - Binary bar plot (section 19.7.9.5)
   - Curve plot with two y-axes (section 19.7.9.2)

- Curve-input (section 19.7.9.11)
- Vector plot (section 19.7.9.10)
- XY curve plot (section 19.7.9.3)

11. **Virtual Instruments** (section 19.7.9.12)
    - Digital display
    - Horizontal scale
    - Measurement instrument
    - Vertical scale

12. **Others**
    - Button
    - Command button
    - Network graphic
    - Schematic path
    - Text
    - Voltage profile (along feeder) (section 19.7.9.13)
    - Voltage sag plot

19.7.9.1 Curve Plot

Curve plots are the “basic” diagrams and are typically used to display one or more plotted curves from the results of a simulation (EMT, RMS, Quasi-dynamic). The Data Series of this plot type is described in section 19.7.2. The following figure shows an example of a curve plot.

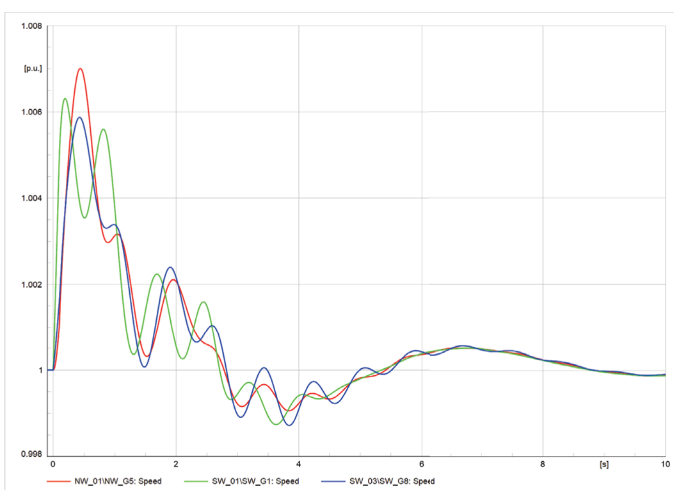


Figure 19.7.10: Curve Plot

19.7.9.2 Curve Plot With Two Y-Axes

A curve plot with two y-axes is typically used for displaying together quantities which have very different scales. The Data Series of this plot type is as described in section 19.7.2, the main difference being the additional y-axis.

The following figure shows an example of a curve plot with two y-axes.

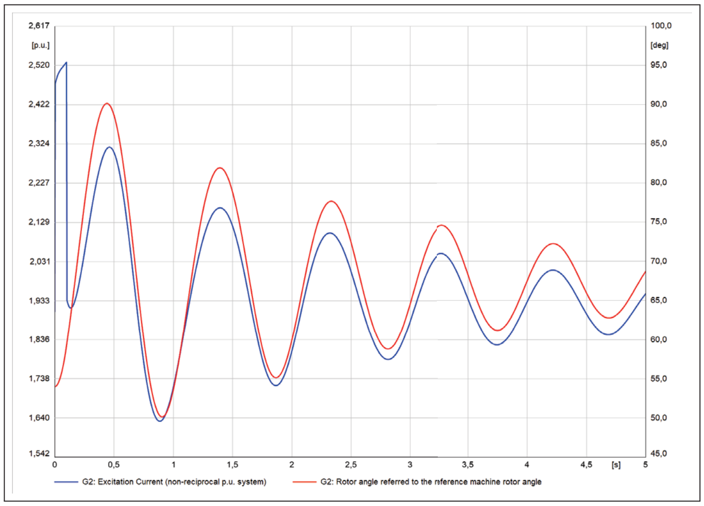


Figure 19.7.11: Curve Plot (two y-axes)

19.7.9.3 XY Curve Plot

This plot shows one variable plotted against a second variable. The two variables can be completely independent from each other and do not have to belong to the same element. The differences in the Data Series of this plot type compared with the one described in section 19.7.2 are:

- Additional columns for the element and variable on the x-axis.
- A specific time range to be displayed can be defined.
- Is it not possible to add processed and aggregated results.

The following figure shows an example of a XY curve plot.

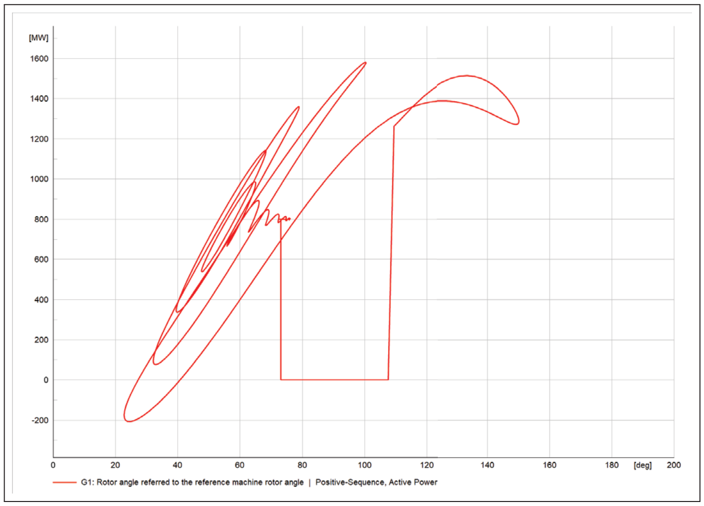


Figure 19.7.12: XY Curve Plot

19.7.9.4 Bar Plot

The bar plot is used to visualise steady-state values such as voltages, currents and power. A bar plot can be inserted after a calculation has been executed using the *Insert Plot* dialog or directly by right clicking on an element(s) and selecting the option *Show* → *Bar Plot*→ *"variable"*. An example of the bar plot is shown in the following figure.

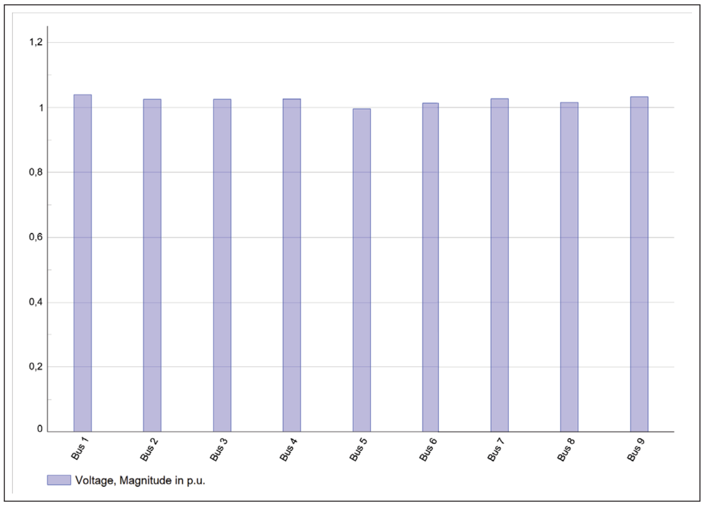


Figure 19.7.13: Bar Plot

The *Style and Layout* page of the edit dialog of the bar plot is the same as that of the Data Series, described in section 19.7.2.2. The *Curves* page is different, as described below.

x-Axis Elements

The *x-Axis Elements* table consists of the elements whose variables are shown in the plot. If the plot is inserted by right clicking on the element, the elements are automatically defined. Otherwise the element can be selected by double clicking on the corresponding field.

Curves

On the *Curves* table, the user can select the variables to be displayed. The default shape is *Bars* and the fill style can be specified by the user.

19.7.9.5 Binary Bar Plot

The Binary Bar Plot can be used to illustrate digital signals. The Data Series is as the one described in section 19.7.2 and the following options are set by default:

- Plot Features:
  - Additional curve shapes
  - Data transformation: Binarisation
  - Curve stacking: Values
- Curves table:
  - Shape: Filled steps

An example of the binary bar plot is shown at the bottom of figure 19.7.14.

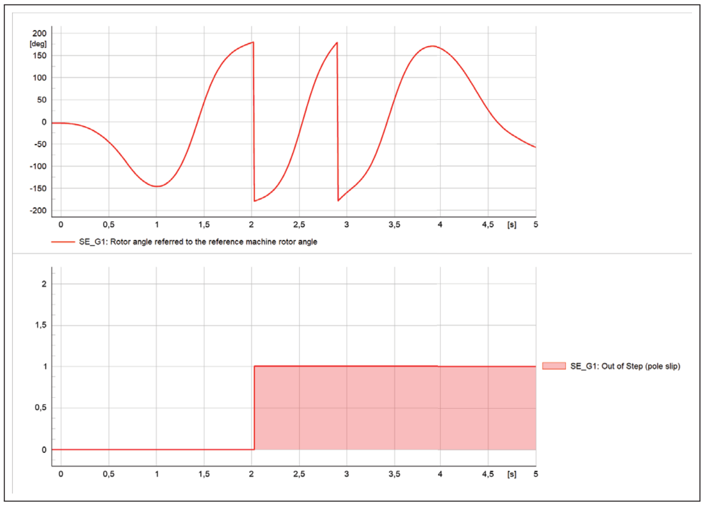


Figure 19.7.14: Binary Bar Plot

19.7.9.6 Duration curve

With the *Duration curve* plot it is possible to evaluate the utilisation of specific elements such as transformers or different power plants over a given time period. After a *Quasi-Dynamic Simulation* is executed, for example, the loading of a transformer can be illustrated.

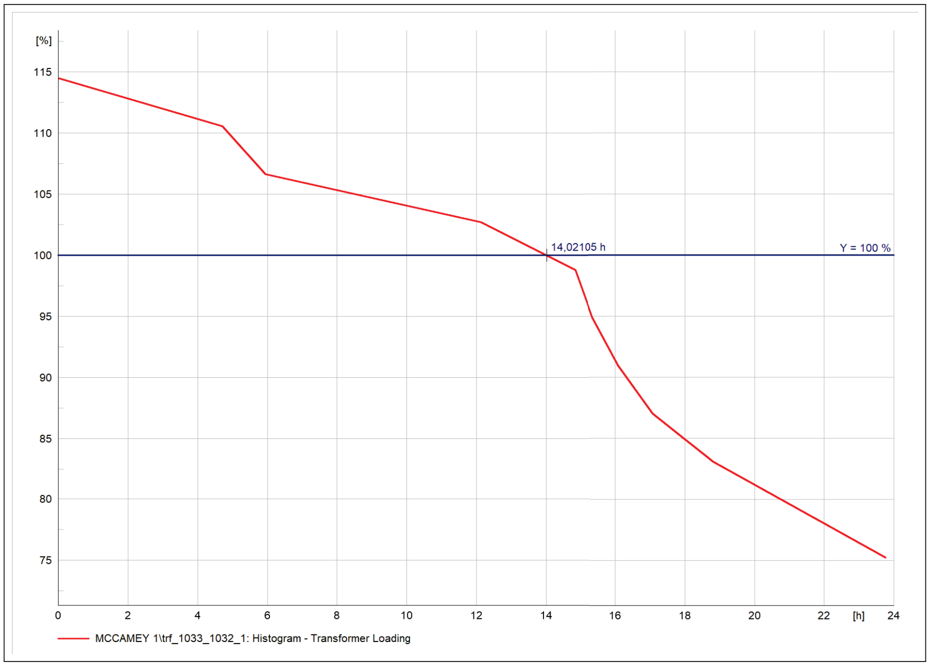


Figure 19.7.15: Duration curve of the loading of a transformer over the duration of a day

The edit dialog of the Data Series of this plot type is the same as the one described in section 19.7.2. The following options are set by default:

- Plot Features:
  - Additional curve shapes
  - Data transformation: Functions

- Curves table:
  - Shape: Steps
  - Column *Function* set. See description below.

Double clicking on the column *Function* opens the *Distribution estimation* dialog, which contains the following options:

- For the ***Method*** that is used to determine the duration curve, *Histogram* or *Bootstrapping* can be selected. A detailed description on the methods can be found in section 44.2.5.
- The ***Curve*** can be defined through several different functions: *Cumulative distribution function*, *Probability density function*, *Quantile function* and *Mirrored quantile function*. More details can be found in section 44.2.1.1.
- The different methods that are used for the ***Estimation*** are *User-defined*, *Automatic parameter deduction*, *Rice Rule*, *Freedman-Diaconis choice* and *Scotts Rule*. Detailed descriptions of the estimation methods can be found in section 44.2.5.2.

19.7.9.7 Plant Categories

The *Plant Categories* plot is one of the energy plots and used for displaying the proportion of all defined plant categories from the total generation in a network.

An example of the plot is shown here:

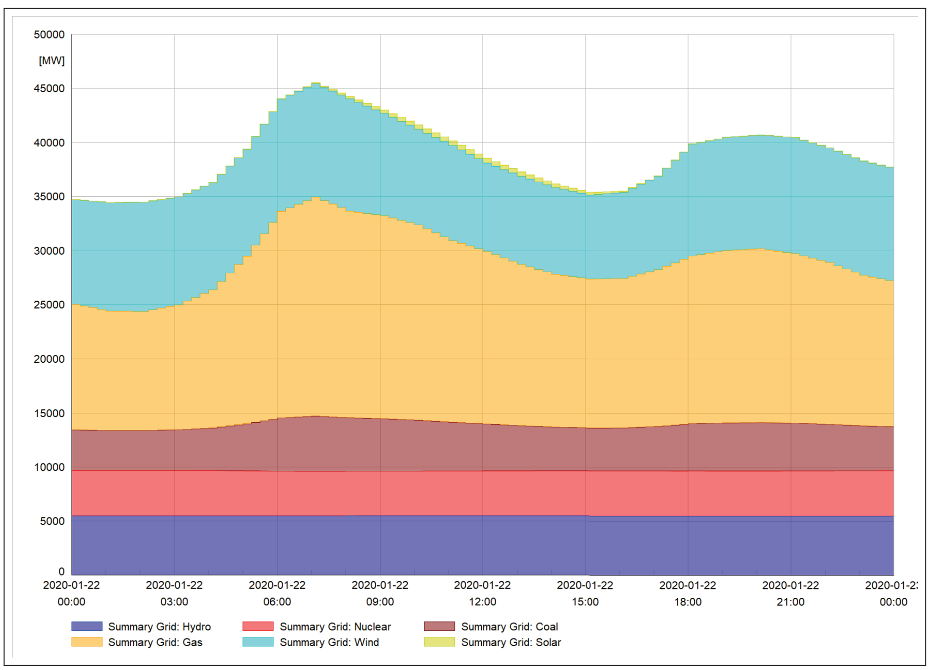


Figure 19.7.16: Total generation in a grid divided into different plant categories

The edit dialog of the Data Series of this plot type is the same as the described in section 19.7.2. The following options are set by default:

- Plot Features:
  - Additional curve shapes
  - Data transformation: Energy plot
  - Curve stacking: Values

- Curves table:
  - Shape: Filled steps

The Plant Categories plot can be customised as follows:

- In the *Curves* table all available *Plant Categories* are shown by default; however, the user can use the checkbox on the *Visible* column to show only some categories.
- In the column *Element*, the defined grouping object for each category is used. By default the *Summary Grid* is selected to represent all calculated generation in the network. It has to be noted that different grouping objects can be defined for the same category, but no element is allowed to be defined in several grouping objects.
- The order of the stacking of the curves is defined as the first entry is displayed in the plot on the bottom, while the last entry that is defined is displayed on the top. This however, can be modified on the plot by right clicking on a curve and selecting *Move Curve* → *Upwards/Downwards*.
- Changing the stacking to *Percentage* or *Relative values* standardises the values to 100 % or 1 to illustrate the share of the shown variables.

The following options available for the curve plots are not available for the energy plots:

- Filter... button
- Export... button

19.7.9.8 Complete Generation

The *Complete Generation* plot is one of the energy plots and displays the total generation in a network. Therefore it can be used to show, for example, the total generation in different areas of the grid, as shown here:

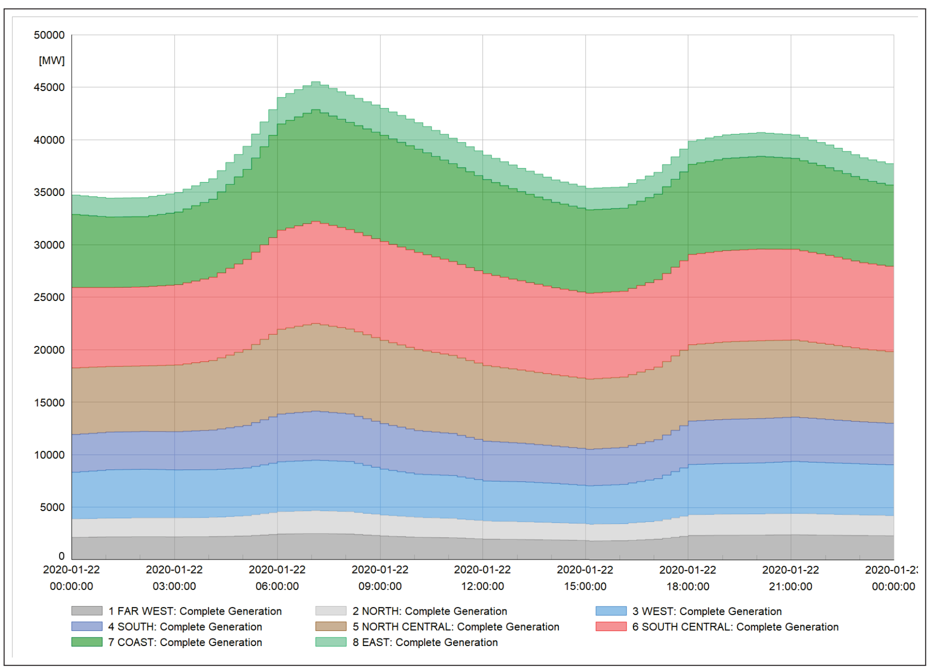


Figure 19.7.17: Total generation in a grid divided into different areas

The pages on the edit dialog of the plot are the same as for the energy plot *Plant Categories* described in section 19.7.9.7.

The only difference is the default entry for the ***Curves***, since all plant categories are summed up as *Group Generators* for the whole summary grid.

19.7.9.9 Renewable and Fossil

The *Renewable and Fossil* plot is one of the energy plots and displays the renewable and fossil shares of the generation in a network.

An example of the plot is shown here:

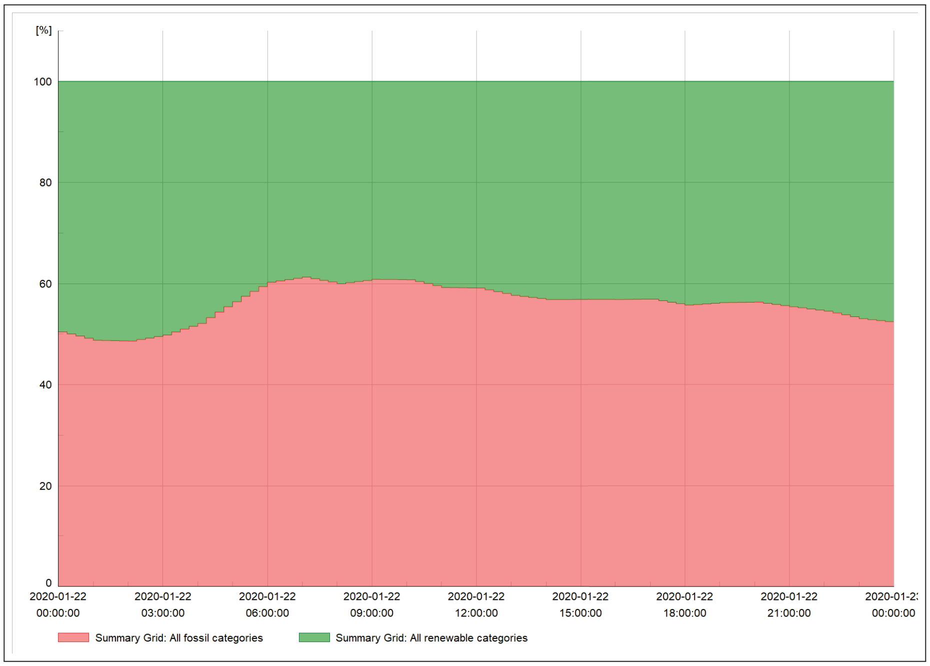


Figure 19.7.18: Total generation in a grid divided into renewable and fossil categories

The pages on the edit dialog of the plot are the same as for the energy plot *Plant Categories* described in section 19.7.9.7.

The only difference is the default entry for the ***Curves*** since the plant categories in the grid are divided into fossil and renewable categories.

19.7.9.10 Vector Plot

A vector plot is used to visualise complex values such as voltages, currents and apparent power as vectors. A complex variable can be defined and shown in one of two different representations:

- Polar coordinates, e.g. magnitude and phase of the current
- Cartesian coordinates, e.g. active-and reactive power

Note: A vector plot can be shown after a load flow calculation or before and after a transient simulation.

A vector plot can be inserted using the *Insert Plot* dialog or directly by right clicking on the element and selecting the option *Show* → *Vector Plot*→ *“variable”*

The Complex Data Definition of this plot type is described in the section 19.7.3.

Figure 19.7.19 shows an example of a vector plot with vector transformation feature enabled.

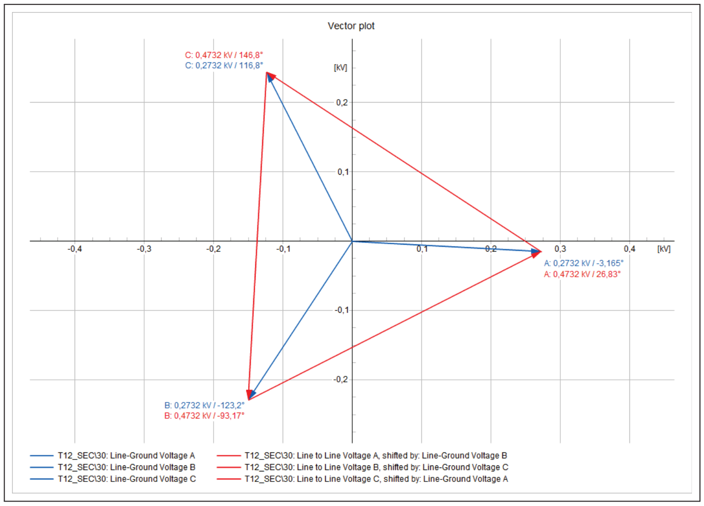


Figure 19.7.19: Vector Transformation in vector plots

Apart from all the options already described in the section 19.7.3, which are also accessible via the context sensitive menu of the plot, the following additional options are available for the vector plot (only available by right click):

- ***Edit Data Element:*** opens the edit dialog of the element whose variables are displayed in the plot.
- ***Select Colour Palette:*** See section 4.7.5.1 for more information about the use of colour palettes.

19.7.9.11 Curve-Input Plot

The curve input command is used for measuring printed curves. The original curves must be available in one of the supported formats and are displayed as a background in the curve input plot as shown in figure 19.7.20. This plot then allows plot points to be defined by successive mouse clicks.

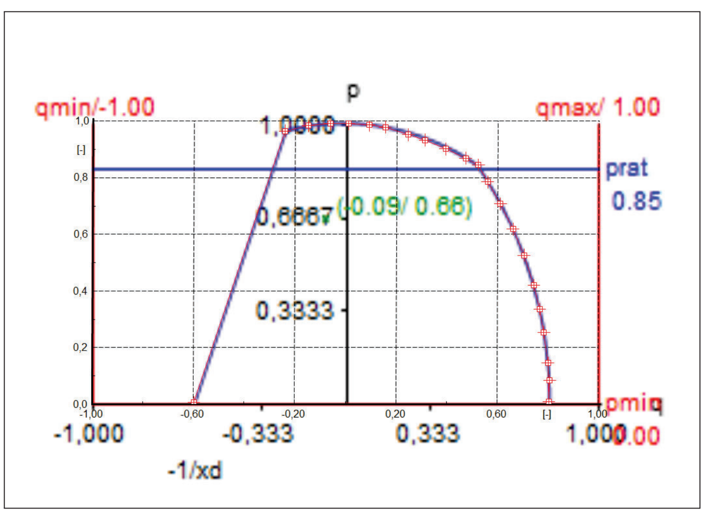


Figure 19.7.20: Curve-Input Plot

The curve input plot allows the measurement and editing of single curves or group of curves at once. The measured curve points will be stored in a Matrix object, which is why, before inserting this plot, it is necessary to define or select the corresponding matrix.

The matrix object should be created inside the project, for example in the Study Case folder, by opening the Data Manager and once inside the Study Case folder (or selected folder), clicking on the *New* icon ( ). From the elements list, the *Matrix (IntMat)* should be selected as shown in figure 19.7.21. The matrix should have at least two columns and one point (inside the curve) has to be manually defined.

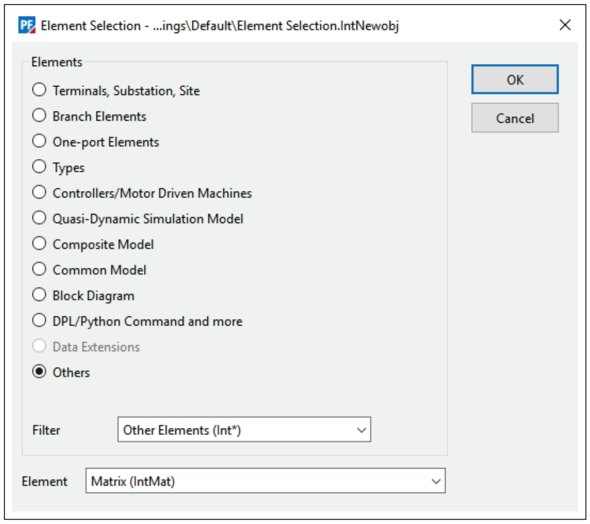


Figure 19.7.21: Defining new matrix

The fields of the curve-input plot edit dialog are:

- ***Background:*** by clicking on the ... button the location of the graphic file to be used as background image can be selected; several formats are supported.
- ***Limits:*** this is used to set the range of the axes of the curves as they are in the graphics file.
- ***Scale:*** the options *Linear* and *Log.* (logarithmic) can be selected and should be as per the graphic file.
- ***Curves:*** two different types of curves can be input:
  - Single: each matrix input defines a single curve. The first column in the matrix holds the x-values, the second one the y values. Other columns are ignored.
  - Set of Curves: only the first matrix is used for input. The first column in the matrix holds the x-values, the other columns hold the y-values of each curve in the group of curves.
- ***Interpolation:*** the measured curve is drawn between the measured points by interpolation. The available modes of interpolation are:
  - Linear
  - Cub. Spline
  - Polygon
  - Hermite
- ***Curves Tables:*** the matrix or list of matrices to be used has to be set in this table.

The rest of the setting of the plot are done using the context sensitive menu, which is accessed by right clicking on the plot. The settings are described below in the order they should be executed.

- ***Set Axis:*** with this option the origin of the axes and the length of the axes can be adjusted according to the figure imported.

  - Origin: sets the origin of the graph to be inserted
  - x-Axis (y=Origin): sets the x-axis dependent on the y-axis origin.
  - x-Axis: sets the x-axis independent of the y-axis.
  - y-Axis (x=Origin): sets the y-axis dependent on the x-axis origin.
  - y-Axis: sets the y-axis independent of the x-axis

- ***Active Curve:*** sets the curve to modify
- ***Input:*** specifies the input mode:
  - x/y-Pairs: each left mouse click adds a point to the curve.
  - Drag & Drop: turns on the "edit mode": all points can be dragged and dropped to change their y-position or left click and delete the point with the **Del** key.
  - Off: switches off the measurement mode
- ***Interpolate All:*** interpolates undefined y values for all curves for all defined x-values
- ***Interpolate N:*** interpolates undefined y values of curve N for all defined x-values

19.7.9.12 Virtual Instruments

The virtual instruments are basically measurement instruments that can be inserted into the plot to present steady state values. The variable can be displayed with one of the following instruments:

- Digital display
- Horizontal scale
- Vertical scale
- Measurement instrument

An example of all the available measurement instruments is shown on the right side of the following figure; the maximum and minimum limits, as well as the element and the variable presented should be defined in the edit dialog of the instrument.

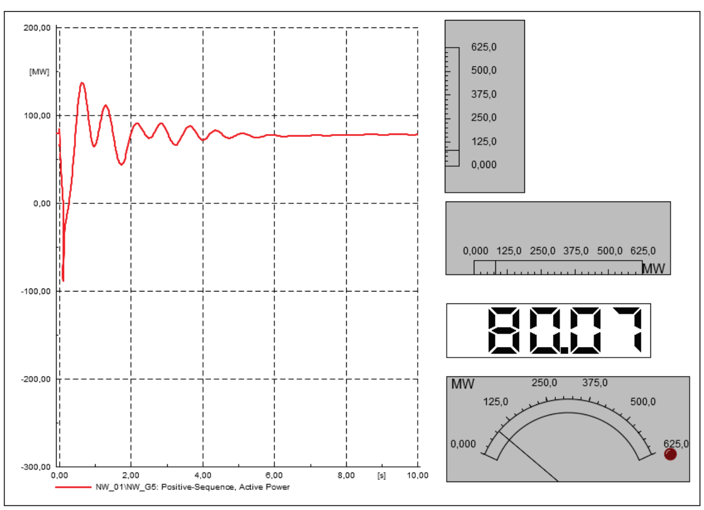


Figure 19.7.22: Measurement Instruments

19.7.9.13 Voltage Profile Plot

The *Voltage Profile Plot (along feeder)* shows the voltage profile of a radial network based on the load flow calculation results. The *Voltage Profile Plot* is directly connected to a feeder object defined in the network, so it can only be created for parts of the system where a feeder is assigned.

The *Voltage Profile Plot* requires a successful load flow calculation before it can display any results. The voltage profile plot can be inserted, as all the plots, using the *Insert Plot* dialog, however, since it is linked to one or more feeders, in this case it is recommended to create the plot directly from the context sensitive menu of the feeder element, selecting *Show* → *Voltage Profile*.

An example of a voltage profile plot is shown here:

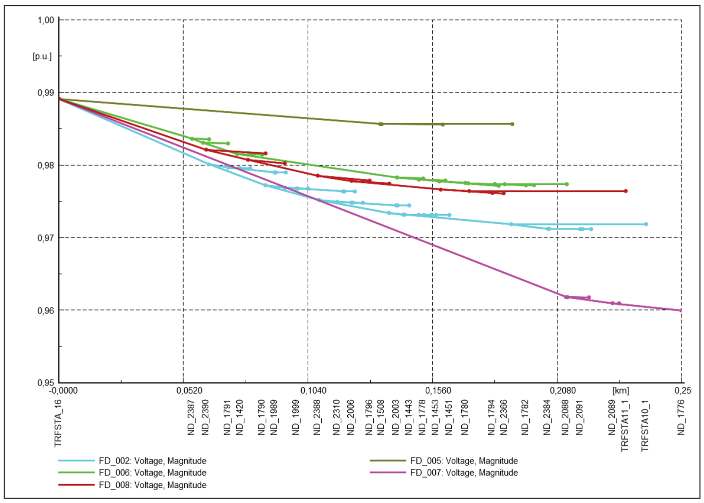


Figure 19.7.23: Voltage profile along feeders

Customising the Voltage Profile Plot

Scales Page

The x-axis variable can be set to one of the following options:

- ***Distance:*** shows the distance from the beginning of the feeder in km.
- ***Bus Index:*** each bus is numbered sequentially from the beginning of the feeder and all of the buses are displayed equidistantly on the plot.
- ***Other:*** this option allows plotting against a user defined variable. Only variables available at all terminals in the feeder can be used.

By default, any branch with a loading greater than 80 % will appear red on the voltage profile plot and any branch loaded less than the *Lower Limit* will be coloured blue. These colours and limits can be adjusted in the *Branch Colouring* field.

The *Parallel Branches* option is required because the voltage profile plot only shows a single connection line between nodes, regardless of how many parallel branches connect the two nodes. If there is a situation where one of these parallel lines is below the *Lower Limit* and another is above the *Upper Limit*, then the parallel branches option determines whether the single line in the voltage profile plot is either the line with the maximum loading or the line with the minimum loading.

Curves Page

On the *Curves* page, the colour and style of the displayed feeders can be modified; also, a display filter can be configured to show only the nodes with a nominal values between the specified values and to ignore nodes with voltages below a specified limit.

Advanced Page

On this page the frame of the plot and the visibility of the legend can be defined. The colour of the busbar (terminal) names on the voltage profile plot can also be modified as follows:

- ***Off:*** does not display any bus names
- ***Black:*** shows all names in black
- ***Coloured acc. to Feeder:*** colours the bus names according to the colour of the different feeders.

The context sensitive menu of the plot shows additional functions regarding the voltage profile plot including:

- ***Edit Feeder:*** opens the *edit* dialog of the feeder related to the plot.
- ***Edit Data:*** opens the *edit* dialog of the selected line, transformer or other element.
- ***Edit and Browse Data:*** shows the selected element in the Data Manager.
- ***Mark in Graphic:*** marks the selected element in the single line graphic(s).

Chapter 20

Data Extensions

20.1 Introduction

Introduced in *PowerFactory* 2018, the Data Extensions functionality allows users to extend their data models by adding user-defined attributes for elements and other objects in a *PowerFactory* project. Furthermore, if the users find that defining additional attributes for existing objects is not sufficient to address their needs, it is also possible to define entire new classes of object.

The definitions of the attributes, which are done on an object class basis, include the data type (such as integer, double, string), description, unit and default value. Once defined, the new attributes are treated by *PowerFactory* in the same way as the in-built attributes, available to scripts, DGS interface etc. Data Extensions are specific to the project, but the configuration can be copied from one project to another, where the new Data Extensions will be aggregated with any existing Data Extensions.

20.2 Data Extension Configuration

20.2.1 Creating Data Extensions

To create Data Extensions within a project, select from the main toolbar *Tools* → *Data Extensions*→ *Configuration*. The Data Extension Configurations are stored in the project Settings folder, and the required **.SetDataext* will be automatically created as required.

When the *Tools* → *Data Extensions*→ *Configuration* command is used, a dialog box appears, which allows the user to create new Data Extensions or modify or delete existing ones.

To create a new Data Extension variable definition, follow these steps:

- Use the *New Object* icon ; this will then create a new *IntAddonvars* object, where the attributes are configured.
- Select or type in the name of the class for which the attributes are to be configured. “Wildcards” (e.g. Elm*) may be used if the attributes are to be made available to multiple classes. Note that if the class name is not recognised as an existing class, it will be assumed that the user wishes to create an entirely new class (see section 20.2.2 below)
- Give the definition a meaningful name.
- Use right-click to append rows in the table below.

- Populate the rows as required. The Name is the actual attribute name and the Description will appear, for example, as a column heading if the attribute is used in a flexible data page.
- The attribute Type is selected from a drop-down list, and the Unit and Initial Value can also be specified as required.
- By clicking on *More...* an *Additional Options* dialog opens, in which the previous described attributes as well as the following three additional attributes can be defined:
  - With the attributes *Minimum value* and *Maximum value* the permissible range for the values can be defined. Note: These fields are only available for data types “int” and “double”.
  - The attribute *Allowed values* can be used to define which values are to be selectable from a drop-down menu. Multiple values must be separated by a semicolon (e.g. 0;1;3;5). Note: This field is only available for data types “int”, “double” and “string”. Additionally, the fields for the minimum and maximum value will be disabled.
- Click on **OK** to save the new definition.

Note that once Data Extensions have been created, further additions or changes will result in a need to migrate the data within the project, and a version will be created in case there is a need to roll back to the state prior to the change.

Modifications to Data Extensions must be done using the *Tools* → *Data Extensions*→ *Configuration* command. Modifications cannot be done by going to the Settings folder of the project and accessing the Data Extensions directly there. It should also be noted that it is currently not possible to change an attribute’s type in an existing Data Extension. The attribute must be deleted, the configuration saved and then the attribute can be redefined with a new type.

20.2.2 User Defined Classes

In an enhancement to the above process, the user can choose to define a completely new class of object, then ascribe the attributes to this new object class.

To define a new object class, follow the above steps but instead of using an existing class name enter a new class name. PowerFactory will create the new class, prefixing it with “Ext” to indicate that it is a user-defined class.

Once the new class is defined, objects of that class can be created and handled like other objects in the project, for example being populated with data and accessed via DPL or Python scripts.

20.3 Using Data Extensions

The new attributes can be treated in much the same way as existing attributes, for example added to flexible data pages or accessed by scripts; the use of characteristics with such parameters, however, is not possible.

To add a new attribute to a flexible data page (see 10.6 for more information about the use of flexible data pages), open up the Variable Definition dialog and select Data Extension on the left-hand side.

They have a prefix p, rather than the e used for the built-in attributes, so for example if Data Extension for synchronous machines includes a new attribute Size, the parameter will be shown as p:Size.

The value of a Data Extension attribute can be modified within the object’s edit dialog (select the Data Extension page) or in a Flexible data page, as with other parameters. Attributes of primitive data types, i.e. integer and double, can be edited in place. For more complex attributes including strings, double-clicking into the cell opens a dedicated edit value dialog. Although it is not possible to type such values directly in a flexible data page, it is still possible to copy and paste them from one object to another.

As mentioned in the introduction, Data Extensions can be used by scripts, DGS imports etc. Recording in scenarios is also supported; however, this is restricted to attributes of type string, integer, double or object.

20.4 Sharing Data Extensions

Having created Data Extensions in one project, users may wish to use them in other projects. It is possible to fetch Data Extension configurations from another project using the command *Tools → Data Extensions→ Copy settings from project*. This process is additive, i.e. the fetched Data Extensions will be added to any existing Data Extensions in the project. However, the user will be required to resolve any conflicts such as duplicate attribute names for the same object class.

Another option open to users if they have Data Extensions which they want to use routinely is to create them in the default project (in the Configuration, Default folder) so that every new project created in the database will automatically have them.

As the Data Extensions are part of the project, they are therefore retained when moving or copying projects. Likewise, they are also retained when projects are exported as .pfd or snapshot export .dzs files, but will not be retained if the older .dz export is used.

Chapter 21

Data Management

21.1 Introduction

The basic elements of project management within the *PowerFactory* environment were introduced in Chapter 4 (*PowerFactory* Overview). They allow the user to generate network designs and administer all input information and settings related to *PowerFactory* calculations and analyses. The project itself is much more than a simple folder which stores all objects which comprise a power system model; it allows the user to do advanced management tasks such as: versioning, deriving, comparing, merging and sharing. These advanced features simplify data management in multi-user environments.

The following sections explain each of the data management functions in more detail:

- Project Versions;
- Derived Projects;
- Comparing and Merging Projects;
- How to update a Project;
- Sharing Projects;
- Combining Projects; and
- Database Archiving.

21.2 Project Versions

The section explains the *PowerFactory* concept of a version. The section first explains what a version is and when it can be used. Next the procedure for creating a version is explained. Specific procedures related to versions such as rolling back to a version, checking if a version is the basis for a derived project and deleting a version are then explained.

21.2.1 What is a Version?

A *Version* is a snapshot of a project taken at a certain point in time. Using versions, the historic development of a project can be controlled. Also, the previous state of a project can be recovered by rolling back a version. From the *PowerFactory* database point of view, a version is a read-only copy of the original project (at the moment of version creation), which is stored inside a Version (*IntVersion*,). Versions are stored inside the original project in a special folder called *Versions*.

The concept of versions is illustrated in Figure 21.2.1. At time $t0$, the project 'SIN' is created. After a time, $t1$, when the owner has made several changes they decide to make a copy of the project in its current state by creating the version 'V1'. After more time, $t2$, and after more changes with respect to 'V1', another version 'V2' is created by the owner. The version control can continue with time like this, with versions accumulating with a periodicity of t.

After versions are created, the owner can revert the project to the state of the version by using the *Rollback* function. This *destroys* all modifications implemented after a version was created (including all versions created after the *rolled*-back version.

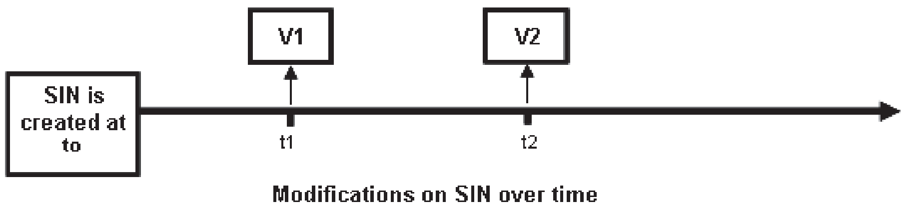


Figure 21.2.1: Project versions

21.2.2 How to Create a Version

This sub-section describes the procedure for creating a version. To create a version of the active project follow these steps:

1. Right-click on the active project.
2. Select *New* → *Version* from the context-sensitive menu. Alternatively, use the option *File* → *New Version* from the main *PowerFactory* menu. The dialog for the new version appears.
3. Set the desired options (explained in the next section) and press **OK**. *PowerFactory* automatically creates and stores the version in the Versions folder (which is automatically created if it does not yet exist).

21.2.2.1 Options in the Version Dialog

Point in Time By default this is set to the system clock time when the version was created. However, it is also possible to enter an earlier time (back to the beginning of retention period of the project).

Note: Setting a *Point in Time* earlier than the clock time means that the version is created considering the state of the project at the time entered. This can be used for example, to revert the project to a previous state, even though other versions have not yet been created.

Notify users of derived projects If this option is enabled, when a user of a project that is derived from the active project activates their derived project, they are informed that the new version is available. Thereafter, updates of the derived project can be made (for further information about derived projects refer to Section 21.3).

Complete project approval for versioning required If this option is enabled, *PowerFactory* checks if all the objects in the active project are approved. If *Not Approved* objects are found, an error message is printed and the version is not created.

Note: The *Approval Status* is found on the *Description* page in the dialog of most grid and library objects.

21.2.3 How to Rollback a Project

This sub-section describes the use of the *Rollback* function to revert a project to the state of a version of that project. For example, consider a project called 'V0', created at a point in time, t. If a *Rollback* to 'V0' is completed, the project returns to its state at the creation of 'V0'. After the *Rollback*, all changes implemented after 'V0' (i.e. after V0's point in time) are deleted. Also, all versions newer than 'V0' are removed. This concept is illustrated in Figure 21.2.2.

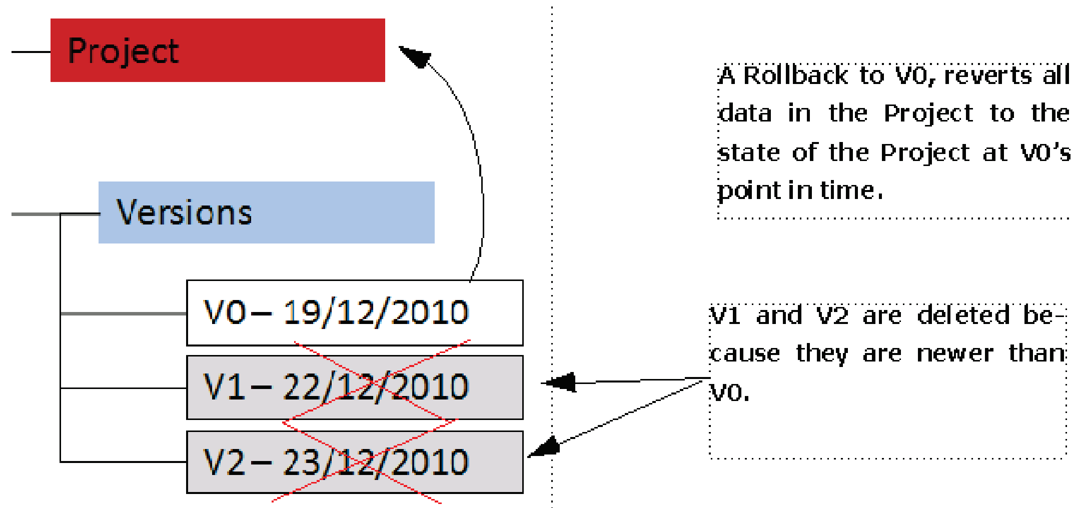


Figure 21.2.2: Example of a rollback

To complete a rollback

1. Deactivate the target project.
2. Right-click on the version that you wish to rollback to and select the option *Rollback to this version* from the context-sensitive menu.
3. Press **OK** when the confirmation message appears.

Note that a *Rollback* is not allowed (and therefore not enabled in the context-sensitive menu) if a newer version of the project exists and this version is the base of a derived project. ***A rollback cannot be undone!***

Note: A version can only be deleted if it does not have derived projects.

21.2.4 How to Check if a Version is the Base for a Derived Project

The following steps should be followed to check if a version is the base for a derived project:

1. Activate the project.
2. Go to the Versions folder inside the project.

3. Right-click on the *Version* that should be checked. This should be done via the right window pane in the Data Manager, not the main Data Manager tree.
4. Select the option *Output...* → *Derived Projects*.
5. A list of derived projects will be shown in *PowerFactory*'s output window.

21.2.5 How to Delete a Version

To delete a version:

1. Activate the project containing the version.
2. Go to the *Versions* folder inside the project.
3. Right-click on the Version that should be deleted.
4. Select the option *Delete*.

21.3 Derived Projects

This section explains the concept of a derived project. For background information regarding the use of derived projects, see sub-Section 21.3.1. In addition, sub-Section 21.3.2 describes the procedure for creating a derived project.

21.3.1 Derived Projects Background

As is often the case, several users might wish to work on the same project. To avoid large amounts of data duplication that would be required to create a project copy for each user, *DIgSILENT* has developed a *virtual copy* approach called *derived* projects. From the user's point of view, a derived project is like a normal copy of a project version. However, only the differences between the original project version (the *base project*) and the virtual copy (the *derived project*) are stored in the database. Because the derived project is based on a version, changes made to the base project do not affect it. Like 'normal' projects, derived projects can be controlled over time by the use of versions, but these *derived* versions cannot be used to create further derived projects.

Note: A derived project is a local 'virtual copy' of a version of a base project (master project):
- It behaves like a "real copy" from the user's point of view.
- Internally, only the data differences between the *base project* and the *derived project* are stored in the database.
- This approach reduces the data overhead.

In a multi-user database, the data administrator might publish a *base* project in a public area of the database. Every user can subsequently create their own derived project and use as if it is the original base project. Changes made by individual users are stored in their respective derived projects, so that the base project remains the same for all users.

In a single-user database, the data administrator must export the *base* project. The user of the derived project must always have this project imported. However, different users of the same *base* project can exchange their derived project. Therefore the derived project should not be exported with option *Export derived project as regular project* enabled. See Section 8.1.4 for further details.

The purpose of a derived project is that all users work with an identical power system model. The derived project always remains connected to the base project.

The concept of derived projects is illustrated in Figure 21.3.1; here version 'Version3' of the base project ('MasterProject') was used to create 'DerivedProject'. After 'DerivedProject' was created, two versions of it were created.

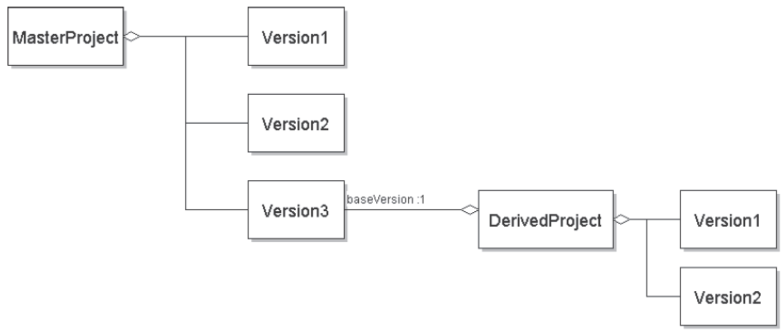


Figure 21.3.1: Principle of derived projects

At any stage, the data administrator might create a version of a base project which has derived projects from other versions of the base project. The user might wish to update their derived project with one of these new versions. Alternatively, the data administrator might like to incorporate changes made in a derived project to the base project. All of these features are possible, by using the Compare and Merge Tool, explained in Section 21.4.

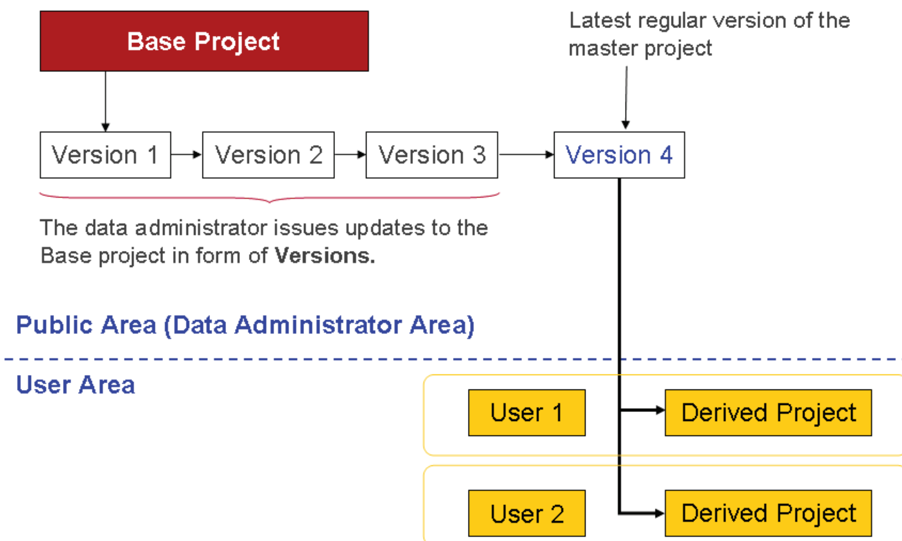


Figure 21.3.2: Derived projects in a multi-user database

In the Data Manager, a derived project looks like a normal project. The *Derived Project* page of its dialog has a reference where the user can see the base project and the version used to derive the project.

Users are notified of changes in the base project if: there is a new version of the base project (newer than the currently-used version) which has the option *Notify users of derived projects* enabled (the user/administrator enables this option when creating a new version), and the option *Disable notification at activation* disabled (found on the *Derived Project* page of the project dialog).

The user may update a derived project when they next activate it, provided that the conditions stated above are met. The newest version that can be used to update a derived project is referred to (if available) in the *Most recent Version* field of the dialog. Users can compare this new version with their own derived project and decide which changes to include in the derived project. For comparing and accepting or refusing individual changes, the Compare and Merge Tool is used. For information about the Compare and Merge Tool refer to Section 21.4.

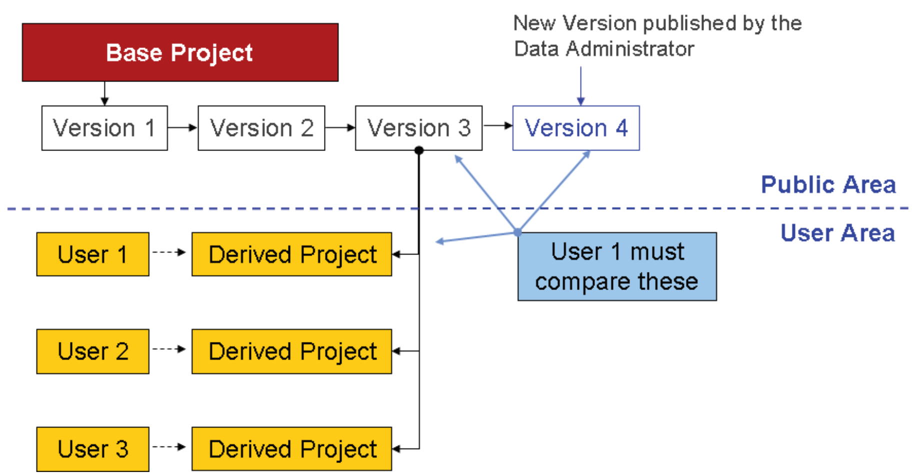


Figure 21.3.3: New version of base project in a multi-user database

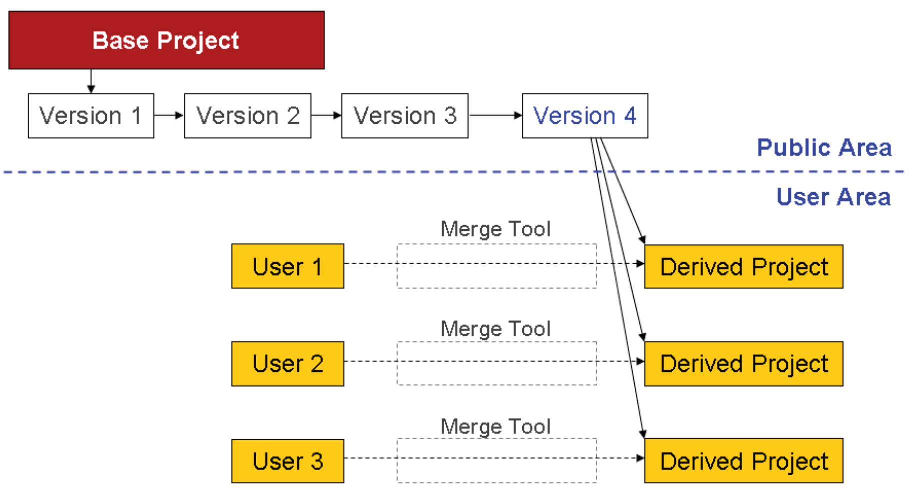


Figure 21.3.4: Merging the new version of the base project into derived projects

21.3.2 How to Create a Derived Project

The most direct and intuitive method for creating a Derived Project is by means of the Data Manager, as explained below:

1. Right-click on the desired base project in the left pane of the Data Manager,
2. Select the option *Create Derived Project* from the context-sensitive menu,
3. Select the source Version of the base project using the data browser that appears automatically,
4. Edit, if required, the predefined settings of the Derived Project and
5. Press **OK**.

After the derived project is created, it can be used in a similar way as a regular project.

Note:
- The base or master project has to have at least one version before other projects can be derived from it.
- A derived project cannot be used as a base project, i.e. it is not possible to create a derived project from another derived project.
- Whether a project is derived or not can be checked on the *Derived Project* page of the project dialog.
- To create a derived project from a base project stored in another user's account, read access is required. See Section 21.6 for further details.

The derived project can be exported as a "Regular Project" or with the base project. This option can be selected from the Export dialog.

21.4 Comparing and Merging Projects

This section describes the procedure for comparing and merging projects within the *PowerFactory* database. There are many circumstances whereby it may be desirable and/or necessary to merge data from multiple projects. For example, when the data administrator updates a master project that is the base project for a derived project that a user is working with. The Compare and Merge Tool (CMT) can be used to update the user's project with the data changes, yet also allows the user control over which changes are implemented.

This section is separated into six sub-sections. Firstly, the background of the CMT is presented. The following sub-section explains the procedure needed for merging together or comparing two projects. Sub-Section 21.4.3 explains the procedure for merging or comparing three projects. In sub-Section 21.4.4, the advanced options of the CMT are explained. The CMT uses a *diff browser* for showing the differences and conflicts between compared projects and also for allowing data assignments. This is explained in sub-Section 21.4.5.

21.4.1 Compare and Merge Tool Background

When working collaboratively in a multi-user environment, a data administrator might often need to update the *master* project to create a version based on updates completed by one or more users to projects derived from the master project. The Compare and Merge Tool (CMT) is used for this purpose. This tool can be used for project comparison in addition to the merging of project data. It is capable of a *two-way comparison* between two projects and also a *three-way comparison* for three projects.

Internally, *PowerFactory* refers to each of the compared projects according to the following nomenclature:

- <Base> Project - the base project for comparison.
- <1st> - the first project to compare to the <Base> project.
- <2nd> - the second project to compare to the <Base> project and to the <1st> project (three-way comparison only).

The CMT compares the chosen projects and generates an interactive window known as the CMT *diff browser* to show the differences. For a two-way merge, the changes found in the <1st> project can be implemented in the <Base>, provided that the user selects <1st> as the *source* (<Base> is by default the *target*). When merging three projects together, the *target* is either the <1st> or <2nd> project.

21.4.2 How to Merge or Compare Two Projects Using the Compare and Merge Tool

This section describes the procedure for merging together or comparing two projects using the Compare and Merge Tool (CMT) (*ComMerge*). Note that the comparison is completed using a similar procedure but with slight differences that will also be explained here.

To merge or compare two projects:

1. In the Data Manager, right-click on an inactive project and choose *Select as Base to Compare*.
2. Right-click on a second (inactive) project and select *Compare to [Name of Base Project]*. The CMT options dialog will appear. The <Base> and the <1st> project are listed in the *Compare* section of the dialog.
3. *Optional*: If a third project should be included in the comparison, the box next to <2nd> must be checked. The third project can then be selected with a data browser by using the icon. See Section 21.4.3 for a more detailed explanation of the 3-way comparison.
4. *Optional*: If the base and compare projects should be swapped around, press the button. This would be the case if Project A should be the <1st> project and Project B should be the <Base>.
5. Select one of the options *Compare only*, *Manually* or *Automatically*. The differences between these three choices are:
   - *Compare only*: If the two projects should only be compared and no merge is desired, then select *Compare only*. This disables the merge functionality and only the differences between the two projects will be shown.
   - *Manually*: When this option is selected, the user will be asked to make assignments (i.e. to choose the source project data for common objects that are merged together). For this option, the target project can also be selected. Selecting <Base> will merge changes into the <Base> project, whereas selecting <1st> will instead merge changes into the <1st> comparison project.
   - *Automatically*: When this option is selected, *PowerFactory* will attempt to automatically merge the two projects together, by automatically making data assignments. In a two-way comparison, merging will be automatically into the base project (the base is automatically assumed to be the 'target' for the merging procedure). Note that if *conflicts* are detected during an automatic merge, the CMT will automatically switch to manual mode.
6. Press **Execute** to run the compare or merge. The CMT *diff browser* will appear (unless an automatic merge was selected and no conflicts were identified by *PowerFactory*). Interpreting and using the *diff browser* is described in Section 21.4.5.

Note: It is possible to assign user-defined names to each of the compared projects. This makes it easier to recognise which project is being referred to by the CMT later on in the diff browser (see Section 21.4.5). For example, the user might wish to name two compared projects 'Master' and 'User', respectively. User-defined names can be implemented by typing the desired name in the *as ...* field in the CMT options dialog. These user-defined names are limited to a maximum of 10 characters.

21.4.3 How to Merge or Compare Three Projects Using the Compare and Merge Tool

This section describes the procedure for merging or comparing three projects using the Compare and Merge Tool (CMT). The comparison procedure is completed using a similar method to that used for a two-way merge or compare, but with minor differences that will be explained here.

To merge or compare three projects:

1. In the Data Manager, right-click an inactive project and choose *Select as Base to Compare*.
2. In the window on the right of the Data Manager, hold the **CTRL** key to multi-select a second and third inactive project.
3. Right-click the multi-selection and select the option *Compare to "<project>"*. The CMT options dialog will appear as shown in Figure 21.4.1. The <Base>, the <1st> and the <2nd> project are listed in the *Compare* section of the dialog.

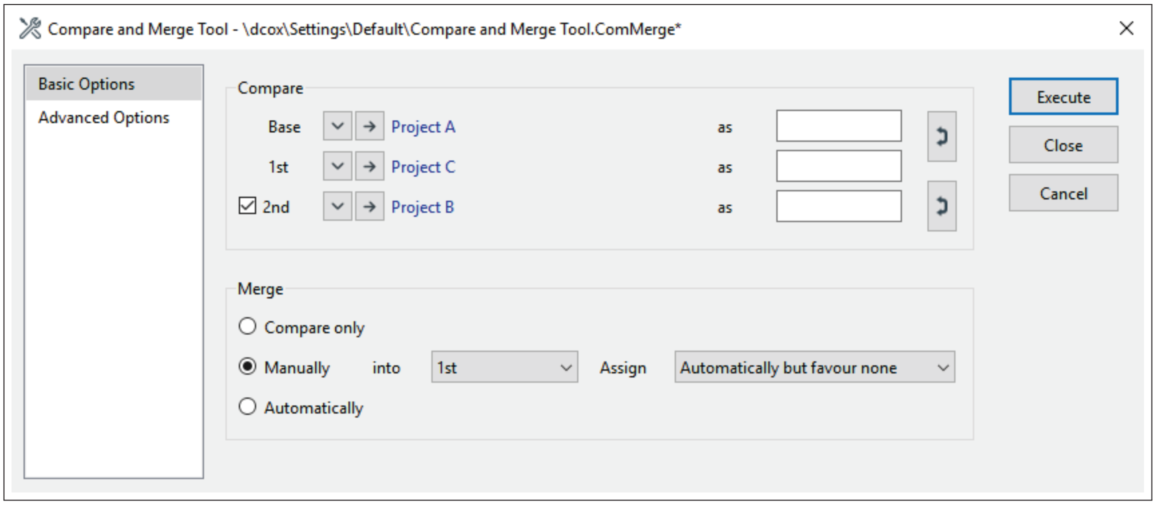


Figure 21.4.1: Compare and Merge Tool options dialog for a three-way merge

4. Select one of the options *Compare only*, *Manually* or *Automatically*. The differences between these three choices are:
   - *Compare only*: If only two projects should be compared and no merge is required, then select the radio button *Compare only*. This disables the merge functionality and only the differences between the two projects will be shown.
   - *Manually*: When this option is selected, the user will be asked to make assignments (to choose the source project data for common objects that are merged together). Using this option, the target project can also be selected. For a three-way merge, merging cannot be done into the <Base>, meaning that either the <1st> or the <2nd> project must be selected.
   - *Automatically*: When this option is selected, *PowerFactory* will attempt to merge the three projects together, via automatic data assignments. As for the option *Manually*, the target can be either the <1st> or <2nd> project. Note that if 'conflicts' are detected during an automatic merge, the CMT will automatically switch to manual mode.
5. If using options *Manually* or *Automatically*, the assignment priority must also be selected, by choosing an option from the *Assign* drop-down menu. This defines the default assignment in the CMT diff browser (or automatic merge) when *PowerFactory* identifies conflicts. For example, say the CMT identifies that the load 'L1' has an active power of 10 MW in <Base>, 12 MW in <1st> and 13 MW in <2nd>. By choosing the option *Automatically and favour 1st*, the default assignment for 'L1' would be <1st>, and a power of 12 MW would be assigned to this load in the target project (provided that the user did not manually alter the assignment).
6. Press **Execute** to run the compare or merge. The CMT *diff browser* will appear (unless an automatic merge was selected and no conflicts were identified by *PowerFactory*). Using the *diff browser* is described in Section 21.4.5.

Note: It is possible to assign user-defined names to each of the compared projects. This makes it easier to recognise which project is being referred to by the CMT later on in the diff browser (see Section 21.4.5). For example, the user might wish to name two compared projects 'Master' and

'User', respectively. User-defined names can be implemented by typing the desired name in the *as ...* field in the CMT options dialog. These user-defined names are limited to a maximum of 10 characters.

21.4.4 Compare and Merge Tool Advanced Options

Identify correspondents by foreign key

When this option is selected *PowerFactory* aligns objects with same foreign key and same class automatically irrespective of their name and location in the Data Manager. This can be useful when combining grid models from different sources when different names are used and the location in the Data Manager may not fit perfectly.

Search correspondents for added objects

This option is only available for a three-way merge and is enabled by default. If enabled, *PowerFactory* can automatically align two independently added objects as being the same object. This option can be useful when completing a comparison on projects where users have added the same object (same name) in each of their respective projects, and the user would like to ensure that *PowerFactory* identifies this object as being the same object. Note that this option is only considered when the *Identify correspondents always by name/rules* option is also enabled.

Consider approval information

By default this option is disabled, which means that information on the *Description* page under *Approval Information* is not compared. For example, if this option is disabled and an object's *Approval status* changes from *Not Approved* to *Approved* or vice versa, then this modification would not be registered by the CMT comparison engine.

Depth

This option controls whether the CMT compares only the selected objects or also all objects contained within the compared objects. By default, *Chosen and contained objects* is enabled which means the CMT compares all objects within the selected comparison objects. This is generally the most appropriate option when merging projects.

Ignore differences $<$

This field defines the tolerance of the comparison engine when comparing numerical parameters. If the difference between two numerical parameters is less than the value entered into this field, then the comparison will show the two values as equal, $=$.

21.4.5 Compare and Merge Tool 'diff browser'

After the CMT options have been set, press the **Execute** button to start the CMT comparison. The comparison and assignment results are then presented in a data browser window (the CMT *diff browser* window shown in Figure 21.4.2). The *diff browser* is divided into three parts:

- Data Tree window on the left;
- Comparison and Assignment window on the right; and
- Output window at the bottom.

These features are explained in the following sections.

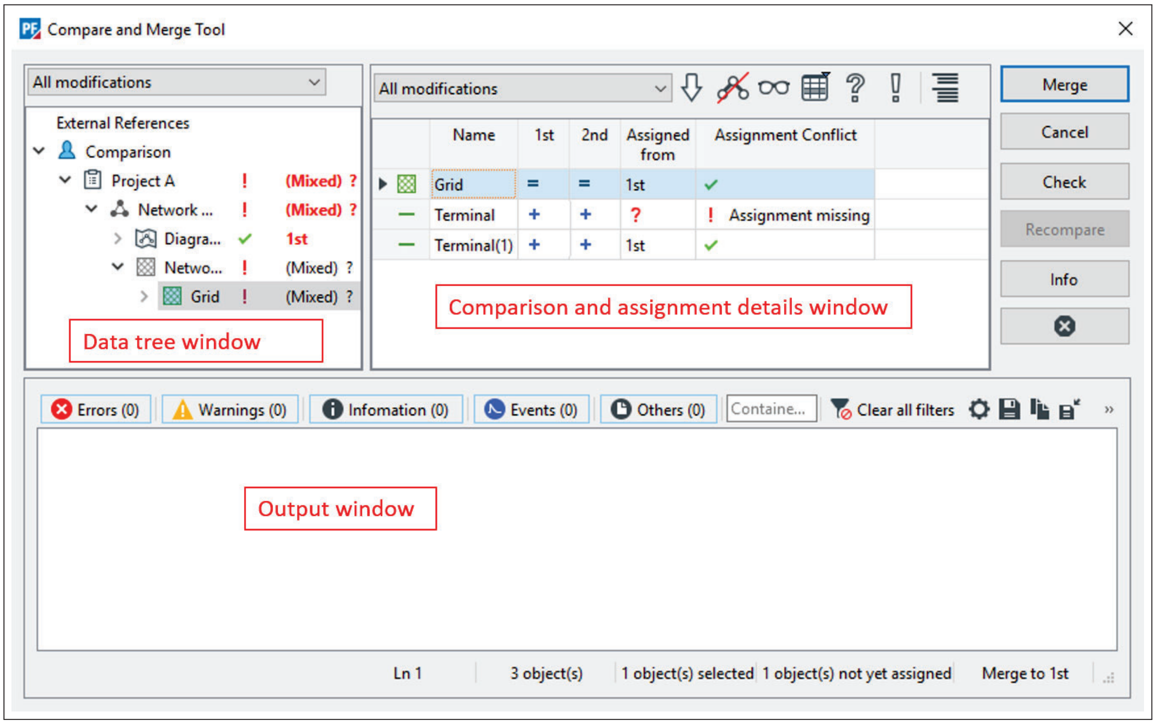


Figure 21.4.2: Compare and Merge Tool *diff browser* after a three-way merge

Output Window

The output window displays reports from the context-sensitive (right-click) menu, and other error information.

How to use the Comparison and Assignment window

In the CMT Comparison and Assignment Window, a list of the compared objects is shown. The window appears slightly different depending on whether a two-way merge, a three-way merge or a comparison has been performed. For instance, after a comparison, the *Assigned from* and *Assignment Conflict* columns are not shown. After a two-way merge, the columns with the project names will show <Base> and <1st> (or user-defined names), whereas after a three-way merge they will show <1st> and <2nd>. A comparison result symbol, indicating the differences found for each object from the list, is displayed in the columns <Base> and <1st> after a two-way merge and in columns <1st> and <2nd> after a three-way merge. The possible combinations of these symbols are shown and explained in Tables 21.4.1 and 21.4.2.

Base	1st	Comment
+	–	The object has been removed from the <1st> project
–	+	The object has been added to the <1st> project
△	△	A parameter of the object has been modified in the <1st> project
=	=	The object is identical in both projects

Table 21.4.1: Possible results after a two-way comparison or merge

1st	2nd	Result	Comment
=	=	=	Objects are identical in all projects
–	△	✓	A parameter of the object is modified in the <2nd> project
△	=	✓	A parameter of the object is modified in the <1st> project
⁞⁞	+	✓	A new object in the <2nd> project
+	⁞⁞	✓	A new object in the <1st> project
=	–	✓	Object removed from the <2nd> project
–	=	✓	Object removed from the <1st> project
△	△	✓	Modified in both projects but the same modifications in both
△	△	!	Modified in both projects but the modifications are different
△	–	!	Modified in the <1st> project and removed from the <2nd> project
–	△	!	Modified in the <2nd> project and removed from the <1st> project
+	+	✓	Identical object added to both projects
+	+	!	Object added to both projects but parameters are different
–	–	✓	Object removed from both projects

Table 21.4.2: Possible results after a three-way comparison or merge

For a project merge (i.e. the *Merge* option was enabled in the command dialog), the *Assigned from* must define the source project of the changes to implement in the target project. All listed objects must have an *Assignment*. If a certain change should not be implemented in the target; then the 'target' project must be selected as the source.

Special attention should be paid to all results indicated by the 'conflict' symbol ! . This symbol shows that the objects are different in both compared projects or that another error has occurred. In the case of conflicts, the user must always indicate the source project for the data.

In a two-way merge, the only available sources for assignment are the <Base> (which is also the target) and <1st>. In a three-way merge, the possible sources are <Base>, <1st> and <2nd>. The assignment can be made manually by double-clicking on the corresponding cell in the *Assigned from* column and selecting the desired source, or double-clicking the <Base>, <1st> or <2nd> cell that the user wishes to assign. However, this task can be tedious in large projects where there are many differences. To rapidly assign many objects, the objects can be multi-selected and then *Assign from ...* or *Assign with Children from ...* can be selected from the context-sensitive (right-click) menu.

Following the assignment of all the objects, the projects can be merged by pressing the **Merge** button. The changes are then automatically implemented in the target project.

Note: The Comparison and Assignment window always shows the selected object in the Data Tree window in the first row.

Data Tree Window

The window on the left side of Figure 21.4.2 shows the *Data Tree*, which is similar in appearance to *PowerFactory*ś Data Manager tree. This window shows the compared objects in a normal project tree structure. At each level of the tree, there is an indication on the right showing the status of the comparison of the contained objects (and the object itself).

The legend for the comparison indication is shown in Table 21.4.3.

Icon/Text	Meaning
✓	Assignments/comparison is okay
!	Conflicts exist
Mixed/<Base>/<1st>/<2nd>	The text indicates the assignments within by indicating the assigned project. If assignments within are from multiple different sources, then 'Mixed' will be shown.
?	Assignments missing
Bold red font	Three-way merge - information will be lost during the merge Two-way merge - information could be lost during the merge

Table 21.4.3: Data Tree window legend

Diff Browser Toolbar

As previously mentioned, the objects displayed in the CMT window can be sorted and organised by the toolbar as shown in Figure 21.4.3. The buttons available are explained in this section.

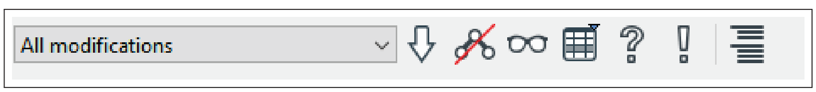


Figure 21.4.3: Compare and Merge Tool 'diff browser' toolbar

Modifications to be shown The *Modifications to be shown* drop-down menu allows the results in the comparison windows to be filtered according to their comparison status. Possible filter options for a three-way comparison are:

- All objects
- All modifications (default)
- All modifications in <1st> (show all modifications, additions and deletions in the <1st> project)
- All modifications in <2nd> (show all modifications, additions and deletions in the <2nd> project)
- All modifications in both (show only those objects which exist in both projects and have been modified in both projects)
- All modifications in both but different (show only those objects which exist in both projects and have been modified in both projects to different values)
- Added in <1st> (show only objects added to the <1st> project)
- Modified in <1st> (show only objects modified in the <1st> project)
- Deleted in <1st> (show only objects deleted from the <1st> project)
- Added in <2nd> (show only objects added to the <2nd> project)
- Modified in <2nd> (show only objects modified in the <2nd> project)
- Deleted in <2nd> (show only objects deleted from the <2nd> project)

The following options are available for a two-way comparison:

- All objects
- All modifications
- Added in <1st>
- Modified in <1st>
- Deleted in <1st>

Only one option can be selected at a time.

Show all objects inside chosen object This button will list all compared objects and also all contained objects (at every level of the tree).

Show graphical elements Pressing this button will prevent graphical differences from appearing in the Comparison window. Because graphical changes often occur, and are usually trivial (i.e. a slight adjustment to the x-axis position of an object), this button is extremely useful for organising the data.

Detail mode and Detail mode class select The functionality of these two buttons is identical to their function in the Data Manager.

Show only not assigned Filters the display to show only objects not yet assigned. This filter is only available when the merge option is used. By default all assigned and unassigned objects are displayed.

Show only Objects with assignment conflicts Only objects with assignment conflicts are displayed. This filter is only available when the merge option is used. By default objects with and without assignment conflicts are displayed.

Group dependent objects If this option is enabled, dependent objects are listed indented underneath each listed comparison object. A dependent object is defined as an object that is referenced by another object. For example, a line type (*TypLne*) is a dependent object of a line element (*ElmLne*), as are the cubicles that connect the line element to a terminal. If the objects are grouped and not filtered otherwise, every object has to be listed at least once but can be listed several times as a dependency. Non-primary objects (such as graphical elements) are only listed separately if they are not listed as a dependency for another object.

Dependent objects are not filtered. By default, the grouping of dependent objects is not displayed because this type of display can quickly expand to become unusable (because in a typical project there are many dependencies).

Diff window right-click menu options

A context-sensitive menu can be accessed by right-clicking on a cell or an object in the Data Tree window or in the Comparison and Assignment window. The following options are available:
Show Object ... A project selection window will appear so that the user can select to show specific object data. After the reference project has been selected, the dialog of the selected object is then displayed. The dialog is read-only.

Output Modification Details This prints a report to the output window showing the details of the differences for the selected objects. The format of the report is an ASCII table with the modified parameters as rows and the parameter values in each compared project as columns. The date and time of the last modification along with the database user who made the last change are always shown in the first two rows.

Output Non-OPD Modification Details This option is similar to the *Output Modification Details* option, but it only shows the modifications that are not classed as *Operational Data*.

Align Manually This option allows the compared objects to be *realigned* across the compared projects. What this means is that *disparate* objects can instead be compared directly. This could be useful for example when two different users have added an object to their derived projects but each has given it a slightly different name, even though the objects are representing the same 'real world' object. The CMT would see these objects as different objects by default. In this case, the data administrator might wish to tell *PowerFactory* that these two *different* objects are the same object. This can be achieved using the *Align Manually* function.

Ignore Missing References For every compared object, missing references can be optionally ignored. The assignment check will then not check the references of the object. Missing references can also be considered again by using the *Consider Missing References* option. By default missing references are not ignored.

Set Marker in Tree A right-click in the Data Tree window allows the user to set a marker within the Data Tree. This has the functionality similar to a bookmark and the user can return to this point in the Data Tree at any time by using the *Jump to Marker "..." in Tree*. Note that it is only possible to set one marker at a time; setting a new marker will automatically overwrite the last marker.

Diff window buttons

The various diff window buttons (as highlighted in Figure 21.4.4) will now be explained.

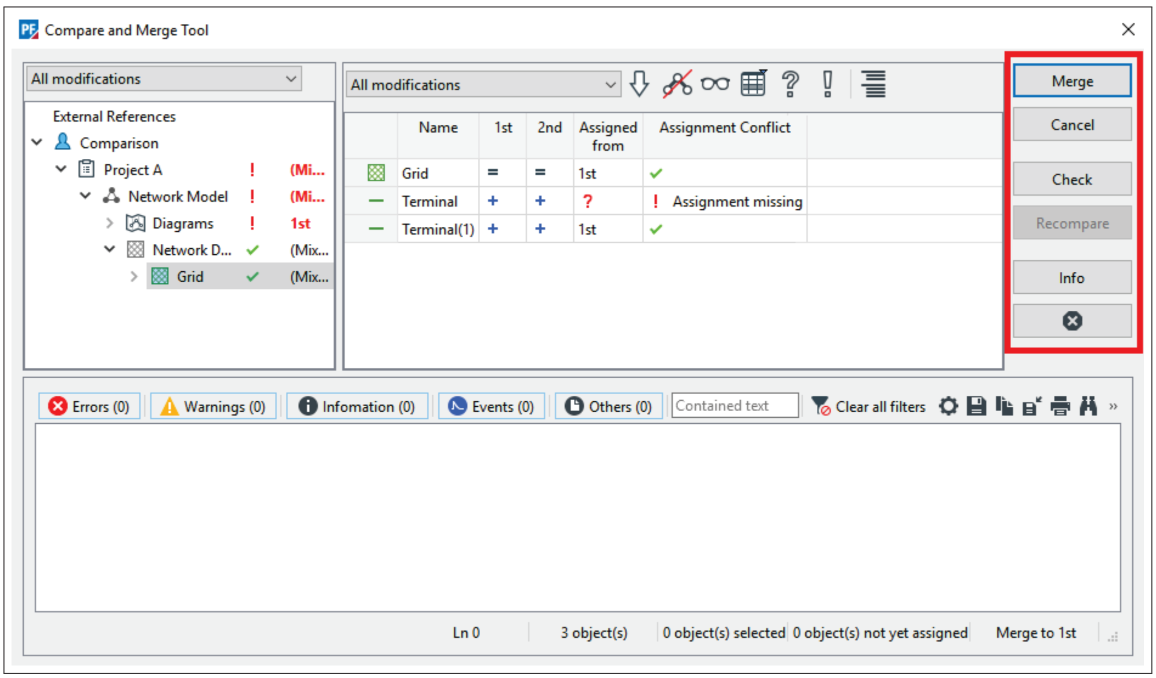


Figure 21.4.4: Compare and Merge Tool 'Diff window' with buttons highlighted

Check This button checks that all assignments are okay. The following conflicts are checked for all compared objects:

- Missing assignment;
- Missing parent (parent object of an assigned object will not exist in the target after merge.)
- Missing reference (referenced object of an assigned object will not exist in the target after merge.)

All conflicts are printed as errors to the output window of the CMT. Conflicts are listed in groups and with the **!** icon in the Data Tree and in the Comparison and Assignment window.

Recompare After a *realignment*, it is necessary to run the CMT again using this button to update the comparison results.

Merge The merge procedure updates the target by copying objects or parameters or by deleting objects according to the assignments. Before the merge procedure is started, an assignment check is done. The merge procedure is cancelled if the check detects conflicts. If no conflicts are detected, the diff browser is closed and then the merge procedure is started. After the merge procedure is complete, all data collected by the CMT is discarded.

Info The Info dialog called by the *Info* button shows more information about the comparison:

- Database path of the top-level projects/objects that are being compared;
- Target for merge (only if merge option is active);
- Selected comparison options;
- Number of objects compared;
- Number of objects modified; and
- Number of objects with conflicts (only if merge option is active).

21.5 How to Update a Project

There are two common procedures that users and data administrators need to complete when working with master projects and other user projects that are derived from versions of this master project:

- Updating a derived project with information from a new version; and
- Updating a master project with information from a derived project.

This section explains these two procedures and also provides *tips* for working with the CMT.

21.5.1 Updating a Derived Project from a new Version

When a derived project is activated after a new version of the *Base* project has been created (provided that the flag *Notify users of derived projects* was checked when the version was created and that the derived project option *Disable notification at activation* is unchecked), then the user will be presented with the dialog shown in Figure 21.5.1.

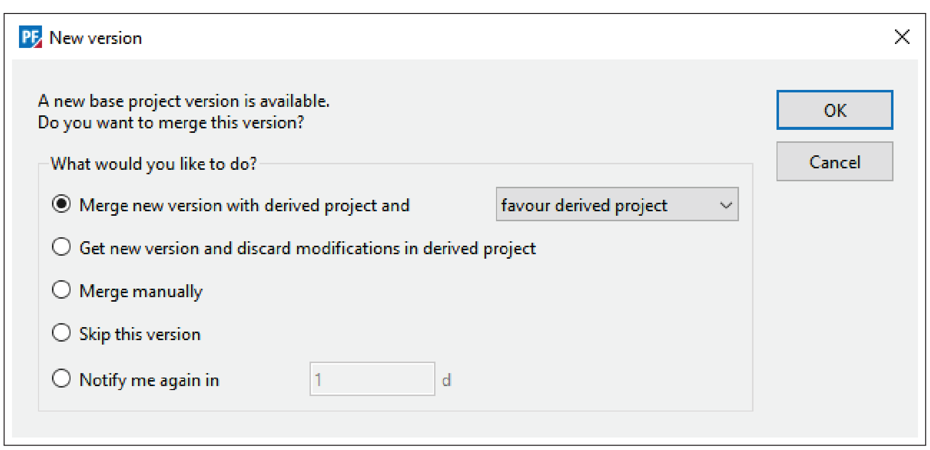


Figure 21.5.1: New version available dialog

The options offered in the notification dialog are:

- **Merge new version with derived project and.** *PowerFactory* automatically generates a temporary copy derived from the new version. It then executes a 3-way comparison with the base version of the user's project (as the base), the derived project (as <1st>) and the temporary copy (as <2nd> and target). In the case of a conflict, one of the following actions will be taken:
  - ***favor none:*** The CMT diff browser is displayed, and the user can then resolve the conflict(s) by defining how the changes should be assigned.
  - ***favor derived project:*** Conflicts are resolved automatically by favouring the user's modifications, thereby discarding modifications in the base.
  - ***favor new version:*** Conflicts are resolved automatically by favouring the base's modifications, thereby discarding the user's modifications.
- **Get new version and discard modifications in derived project.** The derived project is automatically replaced by the new version. All user modifications will be lost.
- ***Merge manually.*** Use the CMT to merge the modifications manually. The results of the comparison are displayed in a CMT diff browser, where the user defines how the changes should be assigned. After these assignments have been defined, the new version and the derived project are merged to the temporary copy, when the user clicks on the *Merge* button. The derived project is then automatically replaced by the temporary copy (now containing information from the new version), which is deleted.
- **Notify me again in. . . .** The user enters the desired time for re-notification, and the derived project is activated according to how it was left in the previous session. The notification is deactivated for the indicated number of days.

Note: In a multi-user environment, updated versions of the *base* project can be released regularly and the user will often be presented with the new version notification in Figure 21.5.1. In many cases, the user will not want to apply the updated version because they will be in the middle of a project or other calculation and may not want to risk corrupting or changing their results. Therefore, the option *Notify me again in. . .* is the recommended choice because it will leave the user's project unchanged.

If the **Cancel** button is used, the project is activated as it was left in the previous session. The notification will appear following the next activation.

An alternative way to manually initiate the above procedure is to right-click on the derived project and select the option *Merge from base project*. This feature is only possible with deactivated projects.

21.5.2 Updating a Base Project from a Derived Project

Changes implemented in derived projects can also be merged to the base project. In this case, the option *Merge to base project* must be selected from the context-sensitive menu available by right-clicking on the derived project. As in previous cases, the CMT is started and conflicts can be manually resolved using the *diff browser*.

21.5.3 Tips for Working with the Compare and Merge Tool

One of the most common uses of the CMT is for merging changes made by users to their derived projects back into the *master* project to create an updated version for all users. This kind of task is often performed by the data administrator. For this task it can help to follow the steps outlined below:

1. Check the user's modifications with a 2-way merge (derived vs. base; What changes were made? Are all changes intended? Modifications which were made by mistake should be corrected in the user's derived model before continuing with the merge procedure.). The check of the modifications should be done by the user and the data administrator.

2. The data administrator creates a new derived project based on the most recent version of the ’master’ model.
3. A three-way merge is performed, selecting the version on which the user’s derived project is based on as ’base’, the derived project created in the previous step as <1st> and the user’s derived project as <2nd>. The changes are merged into <1st> (target).
4. The resulting model is then validated. Conflicts which could not be resolved automatically by the CMT are corrected manually.
5. The validated model (derived project in data administrator account) is merged to the base model by using the context-sensitive menu entry *Merge to Base Project*. This will not cause problems if the master model has not been changed since deriving the model in step 2.
6. A new version is created by the data administrator and the users are informed.

Note: The Compare and Merge Tool can be used to compare any kind of object within a *PowerFactory* project. The functionality and procedure to follow is similar to that explained in this section for project comparison and merging.

21.6 Sharing Projects

In *PowerFactory*, any project can be shared with other users according to the rules defined by its owner. Projects are shared with groups of users and not directly with individuals. Therefore, users must be part of a group (created and managed by the data administrator) in order to access shared projects.

Depending on the access level that the owner assigns to a group, other users can get:

- Read-only access to the shared project, which allows the copying of objects and the creation of derived projects from versions within the shared project;
- Read-write access, which allows users full control over all objects within the project.

  This includes project activation, but does not include the creation of versions.
- Full access, which allows the user to modify the sharing properties and create versions.

Each access level includes the rights of the lower levels.

Deletion of a project is only possible by the project owner.

To share a project:

1. Open the project dialog by right-clicking on the project name and selecting the option *Edit*.
2. Select the *Sharing* page;
3. Right-click within the *Groups* or *Sharing access level* columns on the right side of the *Sharing information* table to insert (or append) a row(s);
4. Double-click in the *Groups* cell of the new line and select the group with whom the project is shared using the data browser;
5. Double-click on the *Sharing access level* to select the desired access level.

A shared project is marked with the symbol in the Data Manager. To display all the available users on the Data Manager, click on the *Show All Users* icon (). Only the shared projects of the other users will be displayed.

For information regarding user groups and the data administrator, refer to Chapter 6 (User Accounts and User Groups).

21.7 Combining Projects

In version 2017 of *PowerFactory*, a new tool was introduced which enables two or more projects to be combined. It is a two-stage process: first, the Project Combination Assistant is used to bring the two networks and all associated data into one project, then the Project Connection Assistant can be used to make connections between the two networks at known common points. Buttons which give easy access to these functions can be found in the “Additional Tools” toolbar.

Note: Within any individual *PowerFactory* project, if a foreign key is defined for any element (on the Description page of the element), that foreign key must be unique. However, when projects are to be combined it is quite possible that there will be duplication of foreign keys. This can be deliberate, to facilitate the connection process (see section 21.7.2.2) but may also be coincidental. In either case, the situation is managed by prefixing all foreign keys with characters which make them unique. For example, a foreign key of “LA234” from the first project would be changed to “001:LA234”.

21.7.1 Project Combination Assistant

The Project Combination Assistant can be used either to combine two or more projects into one new project, or to incorporate further projects into an already active project.

21.7.1.1 New Combined Project

Before starting the combination process, it is first necessary to ensure that the source projects all have a Version defined; the user will need to specify which Version is to be used when the combination process is executed. Please see section 21.2 for more information about versions. It may also be worth thinking about how the networks will be connected in the next step, to ensure that the necessary data configuration has been made, although this can also be done after the two projects are combined.

To start the project combination, first of all ensure that there is no project active, then bring up the Project Combination Assistant tool, either via the icon on the Additional Tools toolbar, or via the *File* → *New*→ *Combined Project . . .* option from the main menu, or by right-clicking on the user name in data manager, then *New* → *Combined Project*.

A list of projects is then made by adding project and version references. Once this has been done, the process is run by pressing the Execute button. The new project is created and activated.

21.7.1.2 Structure of Combined Project

The resultant structure of the combined project can be seen in figure 21.7.1, below. In this example two projects have been combined.

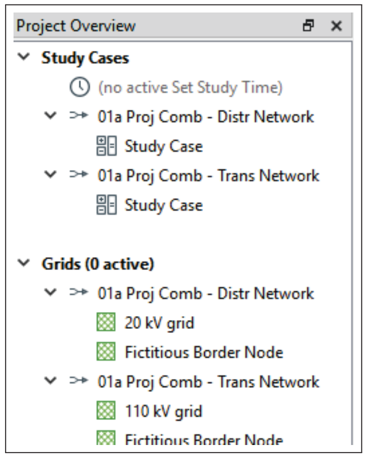


Figure 21.7.1: Structure of combined project

The folders seen in the figure, which are automatically created, take the names of the source projects and allow the user to clearly identify where each object came from. Initially no study cases or grids are active. The user can now use the Project Connection Assistant (21.7.2) to establish the links between the two networks.

21.7.1.3 Incorporating additional Projects

Another possibility for combining projects is to start with an active project and incorporate an additional project or projects into it. In this case, the target project must be active but there should be no study case active. The Project Combination Assistant is then launched using the button on the Additional Tools toolbar. As described above, the source project is then specified, including the required version.

21.7.2 Project Connection Assistant

The Project Connection Assistant offers two methods for automatically making the connections between two networks at the common points in their grids: connection via terminals, or connection via switches.

Following the process of creating a new project described in section 21.7.1.1, a new active study case must first be created before the connection process can start. To do this, right-click on one of the source study cases and select the option to “Apply network state”. The same can be done on the other source study case(s), in which case all the settings of those study cases, i.e. the active scenarios, variations and the summary grid will also be copied to the currently active Study Case. These settings might affect the network topology so it can make a difference for the connection algorithm if they are not added, but it is optional.

At this point it is still possible to make any adjustments to the data (e.g. foreign keys or node names) to ensure that the next step runs smoothly.

21.7.2.1 Connection Using Common Terminals

With this approach, the common points in the two networks are identified as nodes, i.e terminal elements *ElmTerm*. In each of the source projects, all the relevant *ElmTerm* objects should be separated into a designated grid; in the example in figure 21.7.1, the grid is called Fictitious Border Node. There is no restriction on the name of the grid but the same name must be used in each source project. The default method by which the nodes within the connecting grids are matched up is to use the node names (loc_name). However the user can specify an alternative parameter such as the CIM RDF id (*cimRdfId*).

The Project Connection Assistant is launched from the Additional Tools toolbar. It should be noted that a new Variation will be created during the process, which will record all the changes made. The user then selects the connection method "by virtual nodes" from the drop-down menu and specifies the virtual nodes grid. It doesn't matter which of the virtual nodes grid is selected; the tool searches for all grids of this name.

When the Execute button is pressed, the matching nodes in the various virtual node grids are consolidated into new terminals in a new virtual node grid. The source virtual node grids are deactivated.

21.7.2.2 Connection Using Elements with Foreign Keys

As an alternative to specifying terminals as connection points, it is possible to identify the connection points as elements with identical foreign keys. Currently, only switch elements are permitted for this process. Such switches must only be connected to one terminal, indicating that the other side is available for connection. If necessary, the terminal on the outer side of the switch can simply be deleted.

The connection tool searches from switches which have the same foreign key in the two models and will then execute a connection process using any switches connected only on one side. The process involves the removal of one switch and connecting the other switch in its place, as shown in figure 21.7.2. The switch is always left open after this process.

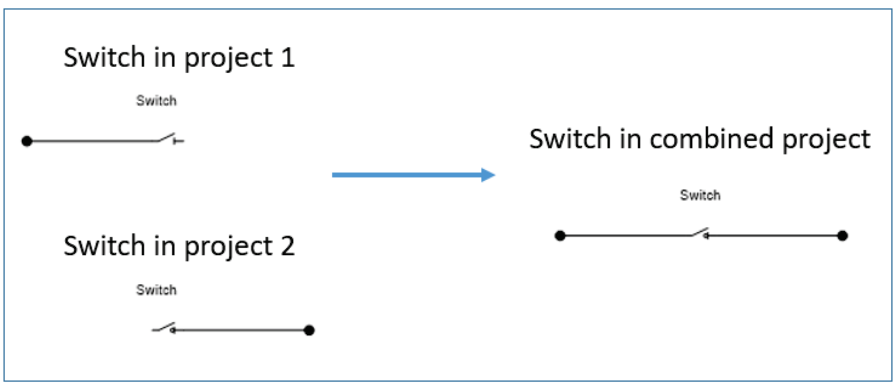


Figure 21.7.2: Network connection using paired switches

To carry out the connection process, the Project Connection Assistant is launched from the Additional Tools toolbar. The user should select the connection method "by foreign key" from the drop-down menu, then press Execute.

Once the connection process is complete, a report is presented to the user in tabular format, listing the matching switches. Any switches found which had identical foreign keys but could not be connected because they were already connected to terminals at both sides will be highlighted.

21.7.3 Final Project State

Once the project combination and connection processes are complete, the result will be a useable project for the whole network. The project structure, with separate folders for components originating from the different source projects, will remain. The connection activities are captured in variations, which can be deactivated if the user wishes to see the state before connection. As mentioned above, foreign keys will have been modified to avoid duplication.

The user should review the connected network to ensure that it correctly represents the desired state. For example, it may be necessary to remove load objects that previously represented lower voltage networks which are now modelled. If connections were made using terminals, the user should consider whether changes need to be made as a result of the original terminals being deleted, for example check that station controllers which referenced terminals in the virtual node grid are now pointing to the new terminals at the interface points.

If it is found that the networks have not been connected as expected, the user should check carefully that the names or foreign keys of the matching elements are precisely the same (remembering that these names/keys are case-sensitive).

Before executing any calculations, the user should be aware that when projects are combined no command objects are copied from the source projects (as there would be no way of knowing which are the preferred settings). Any calculations will therefore initially take the default settings.

21.7.4 Project Normalisation

As described in Section 21.7.1.2, the contents of the project are organised into dedicated folders, making it easy to identify the original sources of the different data items. In some cases, users would like to take a further step of completely integrating the data into a single folder structure. Although it is possible to do this "normalisation" manually, there is also a tool provided, to do this very easily:

- Deactivate the combined project
- Edit the project and go to the Combined Project page
- Click on the **Normalise** button

21.8 Database Archiving

An archiving function for decreasing the used database storage space and increasing performance of large multi-user databases is available. Older projects that are currently not used, but which are still important for possible future use can be archived.

Archiving describes the process of automatically exporting and deleting projects from the database and storing them in a restorable way in the file system. The actual workload is shifted to the housekeeping job which can be run overnight, where export and delete operations do not interfere with the users. Archiving can either be done by the user selecting a project for archiving, or by using DPL scripts.

In multi-user database environments, the user can easily send projects to the archive folder via the context-sensitive (right-click) menu for each project, and selecting "Archive". The archived projects are exported from the database and are placed in a separate folder ("Archived Projects") for long-term storage. The user thereby increases system performance and the speed of general database operations (e.g. project loading/closing). All information regarding the initial project location is also saved allowing the user to restore projects to the exact location from which they originated.

Projects can be restored into the active database by executing the "Restore" command in the context-sensitive (right-click) menu of each project.

For more information on this topic, see Chapter 5 Program Administration, Section 5.6: Housekeeping.

Chapter 22

Task Automation

22.1 Introduction

The Task Automation command (*ComTasks*) enables the *PowerFactory* user to run a list of various tasks ranging from specific *PowerFactory* power system analysis calculation functions up to generic data handling/processing jobs (via scripts) in parallel or sequentially. Using this command it is possible to execute tasks defined in multiple study cases (with any number of calculation commands per case) or multiple independent calculation commands (organised within a single study case). The *Parallel Computation* feature makes full use of a host machine with multi-core processor architecture.

To successfully execute the *Task Automation* command the user first needs to configure a list of calculation functions (e.g. *ComLdf*, *ComSim*) or scripts (e.g. *ComDpl*, *ComPython*) for every designated study case and then *PowerFactory* processes automatically the assigned tasks. Depending on the selected configuration options, a task may represent one study case or one calculation command within a study case (refer to Section 22.2.2 for more information). Most calculation commands can be used within the Task Automation tool given that the specific command actions are acting on the same *PowerFactory* project data. Generally speaking, a calculation command is designated in *PowerFactory* by the object class prefix “Com”.

Task Automation offers enhanced possibilities for power network studies execution, with examples such as:

- Already developed *PowerFactory* projects containing complete power grid analyses which are organised in various study cases, as is usually the case, can be directly used for parallel computation;
- Calculation intensive dynamic simulations can be configured with individual simulation events / operation scenarios by creating multiple study cases. Then, the list of study cases can be passed to the Task Automation command for parallel computation.

For information on how to configure the *Task Automation* command, refer to Section 22.2.

For information on the *Parallel Computing Manager* object, refer to Section 22.4.

For information on how to locate and manage the results generated by the *Task Automation* command, refer to Section 22.3.

22.2 Configuration of Task Automation

A new Task Automation command can be created via:

- the Menu Bar: Click on *Calculation* → *Task Automation...*
- the Main Toolbar:
  - Click on *Change Toolbox* icon ( ) and select the *Additional Tools* toolbox
  - Click on *Task Automation* icon ( )
- the Data Manager:
  - With a project active, open the Data Manager and click on the *Study Cases* project folder
  - From the Data Manager toolbar click on the *New Object* icon ( )
  - Select the elements category *Others* and type in "ComTasks". Click the **OK** button.
- Scripting, by creating a *Task Automation* object (*ComTasks*). Special care needs to be taken when configuring the *Task Automation* object using a script in the sense that assigning references (e.g. study cases or specific commands) to the *Basic Options* page fields (e.g. *vecCases* and *curTasks*) is done via the dedicated DPL functions described in the DPL Reference document.

The Task Automation command dialog has the following pages:

- Basic Options
- Parallel Computing
- Output.

22.2.1 Basic Options Page

Selection of study cases: This dialog pane contains a list of existing study cases that may be considered for the Task Automation command. Study Cases can be added to the list via the **Add** button. The **Remove all** button removes all items within the current list. The checkboxes in the *Ignore* column exclude a specific study case from the cases being considered by the calculation without removing it from the list.

Selection of commands/additional results: This dialog pane stores information on the calculation commands (*Com**) to be executed by the Task Automation for each study case that is added to the *Selection of study cases* list as previously described. The commands list is unique for each study case. The currently shown list is valid for one specific study case as selected via the drop down menu *Study case*. Calculation commands (*Com**) can be added to the list via the **Add** button. As a prerequisite, each selected command must be located within the referenced study case folder. The **Remove all** button removes all items within the current list. The *Ignore* checkbox excludes a specific command from the list without removing its entry.

Additional Results: Several commands generate results files during their execution, such a case being for example the Contingency Analysis command (*ComSimoutage*). Others, like a conventional load flow calculation (*ComLdf*), do not, while the results are stored temporarily in the memory. To address this latter case, the user can choose to write an additional results file per command (by ticking the checkbox in the *Additional results* column). Variables that shall be recorded in that results file after the execution of the command can be configured by double-clicking/right-clicking on the corresponding cell of the *Result variables* column.

Results: This field reference defines the folder where the additional results files of the currently selected study case will be located after the Task Automation command is executed. Moreover, this folder will contain references to all results files that have been generated by a calculation command within this study case.

Note: There is one Results-folder per study case. The shown field reference corresponds to the currently selected study case as shown in the drop down menu *Study case*.

22.2.2 Parallel Computing Page

The Task Automation command can be executed sequentially, thereby processing command after command, or in parallel mode, using the built-in process parallelisation algorithm.

The following settings are subject to user configuration:

Parallel computation: By ticking this checkbox, the user switches from the sequential execution to the parallel task processing; by unchecking it, the sequential execution of tasks is adopted. If parallel computation is selected, a minimum number of tasks can be specified via the setting *Minimum number of packages*. If the user selects fewer tasks than this number, the ComTasks will be executed sequentially.

Parallel Computing Manager: A reference to the Parallel Computing Manager which administrates the parallelisation settings is added. The Parallel Computing Manager is described in Section 22.4. If the *Parallel Computation* checkbox is ticked, the number of processor cores that are practically used by *PowerFactory* (as configured in the Parallel Computing Manager and dependent on the local machine characteristics) is displayed. Clicking the *Edit* button (→) next to the *Parallel Computing Manager* reference will open the parallel computing configuration object. Further details on the settings used in this object are presented in Section 22.4.

The use of the Parallel Computing feature is dependent on the particular *PowerFactory* user settings as defined via the *PowerFactory* Administrator account. Enabling *Parallel Computing* for a particular user is achieved by following the procedure below:

- Log in *PowerFactory* using the Administrator account;
- Within the Data Manager, open the specific *PowerFactory* user account Edit Dialog;
- Click on the *Parallel Computing* page;
- Tick the *Allow Parallel Computing* checkbox.

Note: If a user is not allowed to perform a parallel computation an info-message is displayed in the *Parallel Computing* page.

Distribute packages: The radio-button *Distribute packages* determines the definition of a task (package) in the context of distribution of tasks to the parallel processes:

- If *By study case* is selected, a task is defined as a study case and all commands configured for a specific study case are processed sequentially by the Task Automation command within a single parallel process. This setting is handy when commands belonging to one study case list depend on each other.
- If *By command* is selected, a task is defined as an individual command. Every command is executed by the Task Automation independently of the other commands within the same study case. Furthermore, commands within the same study case may be queued for execution in different parallel processes in order to maximise performance. This option can be used when commands are independent of each other.

Note: The *Distribute packages* option is disabled for sequential execution of the Task Automation. In this case, the option *By study case* is always chosen and commands are executed in the defined order (as specified in the *Selection of commands/additional results* list). If the option *By study case* is chosen for parallel computation, different study cases are assigned to different parallel processes. In particular, the execution of a command in a later study case in the list should not rely on the execution of commands in a previous study case.

Database changes of parallel processes: the radio-button *Database changes of parallel processes* defines, which data shall be transferred from the parallel process to the master process and merged into the database:

- If *Merge all changes to master process* is selected, all changes, which have been made within the parallel process, are transferred to the master and merged into the database. In this case the changes of the database correspond to a sequential execution.
- If *Transfer only results files to master process* is selected, the parallel processes will run in read-only mode. That means, all modifications are temporarily stored in the internal memory of the computer. After a parallel process has finished, only the results (i.e. pointers to results files) are transferred to the master process, to be written back into the database. In addition to the advantage that the database is not changed by the parallel processes in this way, the amount of data, which has to be transferred to the master process, is significantly reduced. This results in a performance increase.

Note: The *Database changes of parallel processes* option will only be available if a parallel calculation is possible (*Parallel computation* box is checked and settings of the Task Automation allow a parallel execution $\rightarrow$ emphasised by the blue text in the *Parallel computation* field). For a sequential execution (emphasised by the red text in the *Parallel computation* field) the *Database changes of parallel processes* option will be disabled.

22.2.3 Output Page

Output per package defines the behaviour of the Task Automation command with respect to Output Window reporting. The *Output per package* radio button has the following settings subject to user configuration:

- **Detailed calculation status** The behaviour of this option is dependent on the task execution mode:
  - Sequential Task Execution: All messages of executed commands are shown in the output window.
  - Parallel Task Execution: A message is issued when the calculation of a task starts and one on success or failure at the task end. Details about which command failed in the task are additionally issued.
- **Short (only issue errors)** The behaviour of this option is dependent on the task execution mode:
  - Sequential Task Execution: Only errors issued during the calculation command execution are displayed.
  - Parallel Task Execution: one message is issued when the calculation of a task starts and another one at the calculation end (reporting execution success or failure).

22.3 Task Automation Results

The *Task Automation* executes a series of commands either sequentially or in parallel using different local machine processor cores and parallel processes. Therefore, there will be no single set of results readily available after executing the *Task Automation* command. The results of an individual command (i.e. from the *Selection of commands/additional results* list) are recorded during its execution or right after it finished by means of an additional results file.

The available tools for obtaining results from the Task Automation command are summarised below:

- The calculation status of individual commands is issued in the Output Window during the task processing as described in Section 22.2.3. Moreover, there is an error summary of all failed commands per study case printed to the Output Window at the end of executing the Task Automation command.

- Beside results files created during the execution of individual commands, additional results files can be defined which are created after the individual command execution. Pre-defined variables can be recorded, as shown in Section 22.2.1. Note that all such results files together with references to results files generated during the calculation are added to a results folder per study case (as defined in the Task Automation command).
- Access all these results files in a summarised tree-structure manner, where the icon *Task Automation - Show results* ( ) can be used. The icon is available from the main toolbar, *Additional Tools* toolbox, next to the Task Automation command.
- Result log files are created for each parallel process. The log files are saved under the *PowerFactory* workspace folder, "db-process-1\log" subfolder (e.g. *C:\Users\MyUser\AppData\Local\DIgSILENT* \PowerFactory
textbackslash Workspace.nnnnnnn\db-process-1\log). These files provide further information on the execution details of each parallel process.

22.4 Parallel Computing Manager

When creating a new Task Automation command or any other command which supports parallel computation (e.g. *Contingency Analysis*, *Quasi Dynamic Simulation*, *Reliability Analysis*, etc.), *PowerFactory* links the specific command to a Parallel Computing Manager object (e.g. as seen in the *Parallel Computing* page of the Task Automation command dialog). This object contains the necessary settings for the parallel computation of tasks and by default it is located in within the *PowerFactory* database under "/System/Configuration/Parallel Computation". This object has read-only rights for a non-administrator *PowerFactory* user account: it can be used by the Task Automation command (or any other command which supports parallel computation) but cannot be edited by the normal user. However, it is possible for a user to customise the Parallel Computing Manager settings in order to use fewer cores than the maximum defined by the Administrator. See Section 7.11 for details.

Alternatively, it is possible to create a user defined settings object by following the steps below:

- Log in *PowerFactory* using the administrator account;
- Using the Data Manager, verify under the "/Configuration" folder if there exists a subfolder named "Parallel Computation" whose key (parameter *loc_name*) is "Parallel". If it does not exist then create a new folder, give it a suitable name and assign the "Parallel" key to it;
- Create a new *Parallel Computing Manager* object ("*.SetParalman") under the newly created system folder (e.g. "Parallel Computation" folder) and give it a suitable name;
- Edit and modify corresponding settings by opening the Parallel Computing Manager Dialog;
- Log out of the *Administrator* account and log in with the normal *PowerFactory* user account;
- Go back to the "Parallel Computing" page of the specific command and notice the changed reference to the newly created *Parallel Computing Manager* object.

The Parallel Computing Manager has the configuration options as summarised below:

- Basic Options page
  - Master host name or IP
  - Parallel computing method
  - Max. number of processes on local machine
- Communication page
  - Communication method

Note: The Parallel Computing Manager settings can be changed only by the Administrator account.

22.4.1 Basic Options Page

Master host name or IP: The machine name or IP address of the master host. If only the local multi-core machine is used, the name can be "localhost".

Parallel computing method:

- Local machine with multiple cores: all the parallel processes will be started in the local machine.

Max. number of processes on local machine:

- Number of cores: all cores available in the machine will be used for parallel computing.
- Number of cores minus 1: use N-1 cores (N is the number of cores available in this machine).
- User defined: the number of parallel processes as specified by the given table will be started in the local machine. The first column of the table is the number of cores available in the local machine and the second column is the number of parallel processes to be started. For a specific machine, the corresponding row in this table is found according to the number of available cores and then the number of parallel processes in the second column is used. If the row is not found (not specified in this table), all cores are used by default.

22.4.2 Communication page

Communication method: The network data can be transferred to parallel processes either via file or TCP/IP protocol.

Chapter 23

Scripting

This chapter describes the options available for scripting in *PowerFactory*, which are based around two programming languages: the *DIgSILENT* Programming Language ***DPL*** and Python.

Section 23.1 looks at the in-built programming language DPL and Section 23.2 shows how this can be used to build tabular reports. Section 23.3 introduces the open source programming language ***Python***.

The remaining two sections of the chapter provide information about the text editor used for scripting and the concept of Add On Modules, which can be used with either DPL or Python.

23.1 The DIgSILENT Programming Language - DPL

The *DIgSILENT* Programming Language ***DPL*** serves the purpose of offering an interface for automating tasks in the *PowerFactory* program. The DPL method distinguishes itself from the command batch method in several aspects:

- DPL offers decision and flow commands
- DPL offers the definition and use of user-defined variables
- DPL has a flexible interface for input-output and for accessing objects
- DPL offers mathematical expressions

The DPL adds a new dimension to the DIgSILENT *PowerFactory* program by allowing the creation of new calculation functions. Such user-defined calculation commands can be used in all areas of power system analysis, such as

- Network optimising
- Cable-sizing
- Protection coordination
- Stability analysis
- Parametric sweep analysis
- Contingency analysis
- etc.

Such new calculation functions are written as program scripts which may use

- Flow commands like “if-then-else” and “do-while”

- *PowerFactory* commands (i.e. load-flow or short-circuit commands)
- Input and output routines
- Mathematical expressions
- *PowerFactory* object procedure calls
- Subroutine calls

23.1.1 The Principle Structure of a DPL Command

The principle structure of a DPL script is shown in Figure 23.1.1.

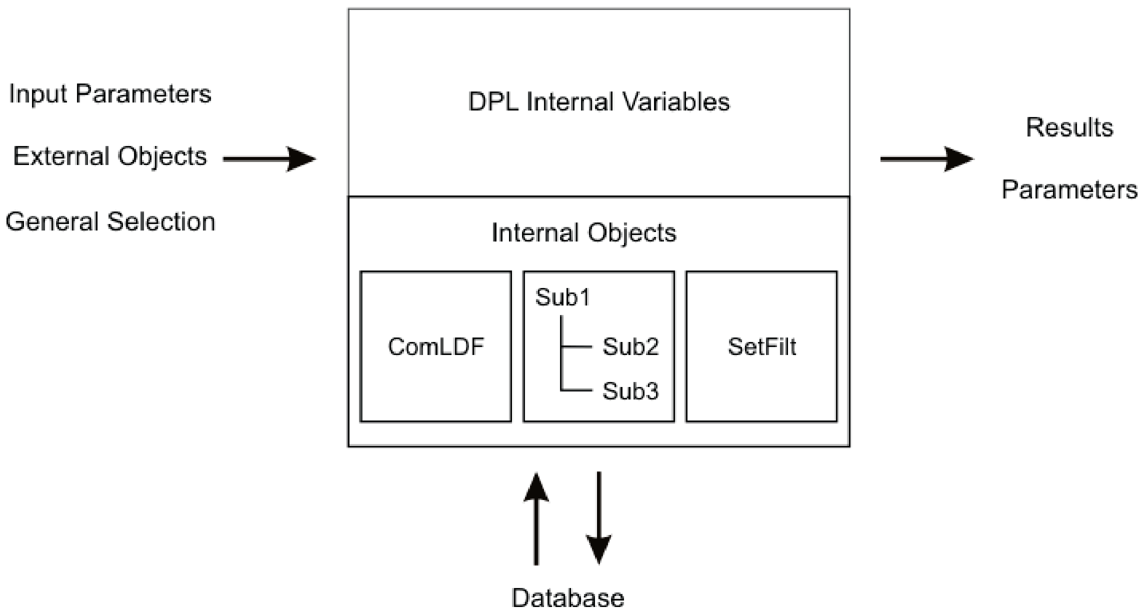


Figure 23.1.1: Principle structure of a DPL command

The DPL command *ComDpl* is the central element, which connects different parameters, variables or objects to various functions or internal elements and then outputs results or changes parameters.

The input to the script can be predefined input parameters, single objects from the single line diagram or the database or a set of objects/elements, which are then stored inside a so called "General Selection".

This input information can then be evaluated using functions and internal variables inside the script. Also internal objects can be used and executed, like

- a calculation command, i.e. *ComLdf*, *ComSim*, etc., especially defined with certain calculation options
- subscripts also released in DPL
- filter sets, which can be executed during the operation of the script

Thus the DPL script will run a series of operations and start calculations or other functions inside the script. It will always communicate with the database and will store changed settings, parameters or results directly in the database objects. Almost every object inside the active project can be accessed and altered.

During or at the end of the execution of the DPL script, the results can be output or parameters of elements may be changed. There is the possibility to execute a predefined output command *ComSh* or to define one's own outputs with the DPL commands available.

23.1.2 The DPL Command

The DPL command element *ComDpl* contains the script code (or a reference to a so called remote script), the definition of input and output parameters, a description and information about versions. DPL command objects can therefore be divided into:

- Root commands, which have their own scripts on the *Script* page of the dialog.
- Referring commands, which use the scripts of remote DPL commands by only adapting input and output parameters and external objects.

23.1.2.1 Creating a new DPL Command

A DPL Command *ComDpl* can be created by using the *New Object* ( ) icon in the toolbar of the Data Manager and selecting *DPL Command and more*. Then press **OK** and a new DPL command is created. The dialog is now shown and the parameters, objects and the script can now be specified.

This dialog is also opened by double-clicking a DPL script, by selecting *Edit* from the context sensitive menu or by selecting the script from the list when pressing the icon .

23.1.2.2 Defining a DPL Commands Set

The DPL command holds a reference to a selection of objects (*General Selection*). At first this general selection is empty, but there are several ways to define a special set of object used in the DPL command. This "DPL Commands Set" (*SetSelect*) can be specified through:

- Select one or more elements in the single line diagram. Then right-click the selection (one of the selected elements) and choose the option *Define...* → *DPL Commands Set...* from the context sensitive menu.
- It is also possible to select several elements in the Data Manager. Right-click the selection and choose the option *Define...* → *DPL Commands Set...* from the context sensitive menu.

23.1.2.3 Executing a DPL Command

To execute a DPL command or to access the dialog of a script, the icon can be activated. This will pop up a list of available DPL and Python scripts from the global and local libraries.

The easiest way to start a DPL command AND define a selection for it is to:

- Select one or more elements in the single line diagram or in the Data Manager and then right-click the selection.
- Choose the option *Execute Script* from the context sensitive menu.
- Then select a DPL script from the list. This list will show DPL scripts from the global as well as from the local library.
- Select a DPL script, insert/change the variables and then press the button **Execute**

In this way the selection is combined into a **DPL Commands Set** and the set is automatically selected for the script chosen.

Only one single DPL command set is valid at a time for all DPL scripts. This means that setting the DPL command set in one DPL command dialog, will change the DPL command set for all DPL commands in the database.

Note: To choose different sets for various DPL scripts you can either use different selection object *SetSelect* like the "General Set". Or new DPL command sets can be created and selected inside the active study case. This is done by pressing , selecting "other" and the element Set (*SetSelect*) and then selecting the set type.

The interface section *Input Parameters* is used to define variables that are accessible from outside the DPL command itself. DPL commands that call other DPL commands as subroutines, may use and change the values of the interface variables of these DPL subroutines.

The list of *External Objects* is used to execute the DPL command for specific objects. A DPL command that, for example, searches the set of lines for which a short-circuit causes too deep a voltage dip at a specific busbar, would access that specific busbar as an external object. Performing the same command for another busbar would then only require setting the external object to the other busbar.

23.1.2.4 Results

On this page, the *Result parameters* can be defined. These parameters are results from the script and they are stored inside the results object. Hence it is possible to access them through the variable monitor and display them in a plot. In addition to the value itself, the name, the type (if a string, object or number), the unit and the parameter description can be entered.

23.1.2.5 DPL Script Page

The most important part of a DPL command is of course the DPL script code. That script is written on the *Script* page of the DPL command dialog. As an alternative to writing the code directly, the command can reference an existing script by selecting it as a *Remote script*.

On this page the DPL code of an already defined script is shown and/or new command lines can be inserted for modifying this script or writing a new script. The available commands and the DPL language are described in the following sections.

The edited program code also features highlighting specially suited for handling DPL scripts.

23.1.2.6 DPL Script Encryption

PowerFactory offers the possibility to encrypt the script code of a DPL command. The encryption action can be initiated by pressing the corresponding button in the edit dialog of the DPL command object (does not work for commands with a remote script referenced). The encryption process then asks in a dialog for a password and its confirmation. The password is only needed to decrypt the script at a later stage. The encrypted script can be executed without entering the password. After completing the encryption with **OK**, the code is hidden and only the name of the command itself, the values of the input parameters and external objects can be changed. If there are subscripts stored as contents, they will be encrypted with the same password, too.

Note: The user should be aware that encryption can never guarantee complete security. The chosen technology balances the requirements for security with the usability and performance of encrypted models. Generally, users are advised to share models only with trusted partners.

The encryption is reversible; an encrypted script can be decrypted using the corresponding button in the edit dialog of the encrypted ComDpl-object. After entering the password and confirming with **OK**, the script returns to its original status, where all properties may be changed and the script code is shown.

Note: The encrypt-action affects the script for which it is executed and does not create an encrypted copy of the ComDpl-object.

23.1.3 The DPL Script Editor

The *Script* page of the DPL command includes a built-in editor based on the *Scintilla* editing component (http://www.scintilla.org), which offers the following features:

- **Auto-completion:** when typing a new word, a list of suitable suggestions for keywords, global functions, variable names (defined within the current script or as input/result variables) and sub-scripts will pop up. Using the arrow keys the user may explore all suggestions, and insert the currently selected suggestion. The autocompletion can be deactivated in the editor settings.

Note: The suggestion lists do not contain deprecated names.

- **Bracket match checking:** when the cursor stands before an opening or closing bracket, the editor will check if the brace is matched. If it is, the bracket and its partner are highlighted in blue. If the bracket, however, is not matched, it will be highlighted in red.
- **Automatic bracket insertion:** when typing in an opening bracket, the editor will automatically insert a matching closing bracket and position the caret between the two brackets. Additionally, if the caret stands before a matched closing bracket, typing a closing bracket of the same type will simply result in the caret moving forwards. This helps users who are not familiar with automatic bracket insertion avoid inserting unnecessary additional closing brackets.
- **Automatic quote character insertion:** similar to automatic bracket insertion; when typing in a single quote character ('), the editor will automatically insert an additional single quote character and position the caret between the two quote characters. Additionally, if the caret stands before a quote character, typing a quote character of the same type will simply result in the caret moving forwards.
- **Zoom-in/Zoom-out:** using the key combination **Ctrl + Mousewheel** will increase or decrease the zoom. Note that this only temporarily modifies the used font size and has no effect at all on the font size that the user chose in the editor font settings. The key combination **Ctrl + 0** restores the font to its original size.
- **Selection highlighting:** whenever text is selected (not counting column selections and selections that span more than one line), all occurrences of the selected text in the current document are lightly highlighted using the last known search settings.
- **Instance-independent search terms and search settings:** whenever the user opens the find (or find/replace) dialog, the chosen search term and search settings are used in every open editor component. This enables users to search the same term with the same settings in multiple documents without having to call the find (or find/replace) dialog for each one of them.
- **Advanced syntax styling:** the script will be coloured according to this scheme: keywords are blue, (recognised) global function and method names are light blue, string literals are red, number literals are turquoise, operators are light brown, identifiers are dark blue, comments are green.

To open the editor (23.4) in an additional window press the icon  on the bottom side of the *Script* page of the DPL Command dialog. Note that when the script is opened in an additional window, it cannot be edited via the DPL Command.

23.1.4 The DPL Script Language

The DPL script language uses a syntax quite similar to the C++ programming language. This type of language is intuitive, easy to read, and easy to learn. The basic command set has been kept as small as possible.

The syntax can be divided into the following parts:

- variable definitions
- assignments and expressions
- program flow instructions
- method calls

The statements in a DPL script are separated by semicolons. Statements are grouped together by braces. Example:

```
statement1;
statement2;
if (condition){
    groupstatement1;
    groupstatement2;
}
```

23.1.4.1 Variable Definitions

DPL uses the following internal parameter types

- **double**, a 15 digits real number
- **int**, an integer number
- **string**, a string
- **object**, a reference to a *PowerFactory* object
- **set**, a container of objects

Vectors and Matrices are available as external objects.

The syntax for defining variables is as follows:

```
[VARDEF] = [TYPE] varname, varname, ..., varname;
[TYPE]   = double | int | object | set
```

All parameter declarations must be given together in the top first lines of the DPL script. The semicolon is obligatory.

Examples:

```
double  Losses, Length, Pgen;
int     NrOfBreakers, i, j;
string  txt1, nm1, nm2;
object  O1, O2, BestSwitchToOpen;
set     AllSwitches, AllBars;
```

23.1.4.2 Constant parameters

DPL uses constant parameters which cannot be changed. It is therefore not accepted to assign a value to these variables. Doing so will lead to an error message.

The following constants variables are defined in the DPL syntax:

SEL is the general DPL selection

NULL is the "null" object

this is the DPL command itself

Besides these global constants, all internal and external objects are constant too.

23.1.4.3 Assignments and Expressions

The following syntax is used to assign a value to a variable:

```
variable = expression;
variable += expression;
variable -= expression;
```

The add-assignment "+=" adds the right side value to the variable and the subtract-assignment "-=" subtracts the right-side value.

Examples:

```
double x,y;
x = 0.5*pi();      ! x now equals 1.5708
y = sin(x);        ! y now equals 1.0
x += y;            ! x now equals 2.5708
y -= x;            ! y now equals -1.5708
```

23.1.4.4 Standard Functions

The following operators and functions are available:

- Arithmetic operators: +, -, *, /
- Standard functions (all trigonometric functions based on radians (RAD))

function	description	example
sin(x)	sine	sin(1.2)=0.93203
cos(x)	cosine	cos(1.2)=0.36236
tan(x)	tangent	tan(1.2)=2.57215
asin(x)	arcsine	asin(0.93203)=1.2
acos(x)	arccosine	acos(0.36236)=1.2
atan(x)	arctangent	atan(2.57215)=1.2
sinh(x)	hyperbolic sine	sinh(1.5708)=2.3013
cosh(x)	hyperbolic cosine	cosh(1.5708)=2.5092
tanh(x)	hyperbolic tangent	tanh(0.7616)=1.0000
exp(x)	exponential value	exp(1.0)=2.718281
ln(x)	natural logarithm	ln(2.718281)=1.0
log(x)	log10	log(100)=2
sqrt(x)	square root	sqrt(9.5)=3.0822
sqr(x)	power of 2	sqr(3.0822)=9.5
pow (x,y)	power of y	pow(2.5, 3.4)=22.5422
abs(x)	absolute value	abs(-2.34)=2.34
min(x,y)	smaller value	min(6.4, 1.5)=1.5
max(x,y)	larger value	max(6.4, 1.5)=6.4
modulo(x,y)	remainder of x/y	modulo(15.6,3.4)=2
trunc(x)	integral part	trunc(-4.58823)=-4.0000
frac(x)	fractional part	frac(-4.58823)=-0.58823
round(x)	closest integer	round(1.65)=2.000
ceil(x)	smallest larger integer	ceil(1.15)=2.000
floor(x)	largest smaller integer	floor(1.78)=1.000

Table 23.1.1: DPL Standard Functions

- Constants:

pi()	pi
twopi()	2 pi
e()	e

Table 23.1.2: DPL Internal Constants

23.1.4.5 Program Flow Instructions

The following flow commands are available.

```
if ( [boolexpr] ) [statlist]
if ( [boolexpr] ) [statlist] else [statlist]
do [statlist] while ( [boolexpr] )
while ( [boolexpr] ) [statlist]
for ( statement ; [boolexpr] ; statement ) [statlist]
```

in which

```
[boolexpr] = expression [boolcomp] expression
[boolcomp] = "<" | ">" | "=" | "<=" | ">=" | "<>"
[statlist] = statement; | { statement; [statlist] }
```

- Unary operators: “.not.”
- Binary operators: “.and.” | “.or.” | “.nand.” | “.nor.” | “.eor.”

- Parentheses: {logical expression}

Examples:

```
if (a<3) {
    b = a*2;
}
else {
    b = a/2;
}
while (sin(a)>=b*c) {
    a = O:dline;
    c = c + delta;
}
if ({.not.a}.and.{b<>3}) {
    err = Ldf.Execute();
    if (err) {
        Ldf:iopt_lev = 1;
        err = Ldf.Execute();
        Ldf:iopt_lev = 0;
    }
}
for (i = 0; i < 10; i = i+1){
    x = x + i;
}
for (o=s.First(); o; o=s.Next()) {
    o.ShowFullName();
}
```

Break and Continue

The loop statements “do-while” and “while-do” may contain “break” and “continue” commands. The “break” and “continue” commands may not appear outside a loop statement.

The “break” command terminates the smallest enclosing “do-while” or “while-do” statement. The execution of the DPL script will continue with the first command following the loop statement.

The “continue” command skips the execution of the following statements in the smallest enclosing “do-while” or “while-do” statement. The execution of the DPL script is continued with the evaluation of the boolean expression of the loop statement. The loop statement list will be executed again when the expression evaluates to TRUE. Otherwise the loop statement is ended and the execution will continue with the first command following the loop statement.

Example:

```
O1 = S1.First();
while (O1) {
    O1.Open();
    err = Ldf.Execute();
    if (err) {
        ! skip this one
        O1 = S1.Next;
        continue;
    }
    O2 = S2.First();
    AllOk = 1;
    DoReport(0); !reset
    while (O2) {
        err = Ldf.Execute();
        if (err) {
          ! do not continue
```

```
          AllOk = 0;
          break;
        }
        else {
         DoReport(1); ! add
        }
        O2 = S2.Next();
    }
    if (AllOk) {
        DoReport(2); ! report
    }
    O1 = S1.Next();
}
```

23.1.4.6 Input and Output

The “input” command asks the user to enter a value.

```
input(var, string);
```

The input command will pop up a window with the string and an input line on which the user may enter a value. The value will be assigned to the variable “var”.

The “printf” command can be used to write text to the output window.

```
printf(string);
```

The string may contain “=-” signs, followed by a variable name. The variable name will then be replaced by the variable’s value.

Example:

```
double diameter;
input(diameter, 'enter diameter');
printf('the entered value = %f',diameter);
```

The example results in the pop up of a window as depicted in Figure 23.1.2.

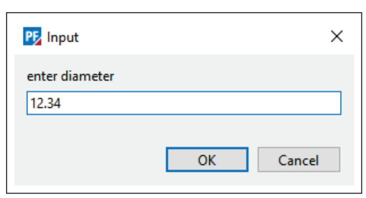


Figure 23.1.2: The input window

The following text will appear in the output window:

```
the entered value = 12.3400
```

Refer to the DPL Reference for more information about the printf command.

23.1.5 Access to Other Objects

With the syntax for the parameter definitions, program flow and the input and printf, it is already possible to create a small program. However, such a script would not be able to use or manipulate variables of “external” objects. It would not be possible, for instance, to write a script that replaces a specific line by possibly better alternatives, in order to select the best line type. Such a script must be able to access specific objects (the specific line) and specific sets of objects (the set of alternative line types).

The DPL language has several methods with which the database objects and their parameters become available in the DPL script:

- The most direct method is to create an object, or a reference to an object, in the DPL command folder itself. Such an object is directly available as “object” variable in the script. The variable name is the name of the object in the database.
- The DPL command set may be used. This method is only useful when the order in which the objects are accessed is not important. The DPL command set is automatically filled when a selection of elements is right-clicked in either the single line graphic or the Data Manager and the option *Execute DPL Script* is selected.
- The list of external objects is mainly used when a script should be executed for specific objects or selections. The list of external objects is nothing more than a list of “aliases”. The external object list is used to select specific objects for each alias, prior to the execution of the script.

23.1.5.1 Object Variables and Methods

If a database object is known to the DPL command, then all its methods may be called, and all its variables are available. For example, if we want to change a load-flow command in order to force an asymmetrical load-flow calculation, we may alter the parameter `iopt_net`. This is done by using an assignment:

```
Ldf:iopt_net = 1; ! force unbalanced
```

In this example, the load-flow objects is known as the objects variable “Ldf”. The general syntax for a parameter of a database object is

```
objectname:parametername
```

In the same way, it is possible to get a value from a database object, for instance a result from the load-flow calculations. One of such a result is the loading of a line object, which is stored in the variable `c:loading`. The following example performs the unbalanced load-flow and reports the line loading. Reported value is always represented in the unit selected in *PowerFactory*. In our case returned value is in % but for example returned value for active power (`m:P:bus1`) can be represented in MW, kW, etc.

Example

```
00. int error;
01. double loading;
02. Ldf:iopt_net = 1; ! force unbalanced
03. error = Ldf.Execute(); ! execute load-flow
04. if (error) {
05.   exit();
06. } else {
07.   loading = Line:c:loading; ! get line loading
08.   printf('loading=%f', loading ); ! report line loading
09. }
```

This examples is very primitive but it shows the basic methods for accessing database objects and their parameters.

23.1.6 Access to Locally Stored Objects

Locally stored objects (also called “internal objects”) can be accessed directly. They are known in the DPL script under their own name, which therefore must be a valid DPL variable name. It will not be possible to access an internal object which name is “My Load-flow\∼{}1*”, for instance.

Internal objects may also be references to objects which are stored elsewhere. The DPL command does not distinguish between internal objects and internal references to objects.

The example DPL script may now access these objects directly, as the objects “Ldf” and “Line”. In the following example, the object “Ldf”, which is a load-flow command, is used in line 01 to perform a load-flow.

```
00. int error;
01. error = Ldf.Execute();
02. if (error) {
03.   printf('Load-flow command returns an error');
04.   exit();
05. }
```

In line 01, a load-flow is calculated by calling the method `Execute()` of the load-flow command. The details of the load-flow command, such as the choice between a balanced single phase or an unbalanced three phase load-flow calculation, is made by editing the object “Ldf” in the database. Many other objects in the database have methods which can be called from a DPL script. The DPL contents are also used to include DPL scripts into other scripts and thus to create DPL “subroutines”.

23.1.7 Accessing the General Selection

Accessing database objects by storing them or a reference to them in the DPL command would create a problem if many objects have to be accessed, for instance if the line with the highest loading is to be found. It would be impractical to create a reference to each and every line.

A more elegant way would be to use the DPL global selection and fill it with all lines. The Data Manager offers several ways in which to fill this object **DPL Command Set** with little effort. The selection may then be used to access each line indirectly by a DPL “object” variable. In this way, a loop is created which is performing the search for the highest loading. This is shown in the following example.

Example

```
00. int error;
01. double maxi;
02. object O, Omax;
03. set S;
04.
05. error = Ldf.Execute();      ! execute a load-flow
06. if (error) exit();          ! exit on error
07.
08. S = SEL.AllLines();         ! get all selected lines
09. Omax = S.First();           ! get first line
10. if (Omax) {
11.   maxi = Omax:c:loading;     ! initialise maximum
```

```
12. } else {
13.   printf('No lines found in selection');
14.   exit();                    ! no lines: exit
15. }
16. O = S.Next();                ! get next line
17. while (O) {                  ! while more lines
18.   if (O:c:loading>maxi) {
19.     maxi = O:c:loading;       ! update maximum
20.     Omax = O;                ! update max loaded line
21.   }
22.   O = S.Next();
23. }
24. printf('max loading=%f', maxi); !print results
25. Omax.ShowFullName();
```

The object **SEL** used in line 08 is the reserved object variable which equals the *General Selection* in the DPL command dialog. The **SEL** object is available in all DPL scripts at all times and only one single “General Selection” object is valid at a time for all DPL scripts. This means that setting the **General Selection** in the one DPL command dialog, will change it for all other DPL commands too.

The method `AllLines()` in line 08 will return a set of all lines found in the general selection. This set is assigned to the variable “S”. The lines are now accessed one by one by using the set methods `First()` and `Next()` in line 09, 16 and 22.

The line with the highest loading is kept in the variable “Omax”. The name and database location of this line is written to the output window at the end of the script by calling “ShowFullName()”.

23.1.8 Accessing External Objects

The DPL contents make it possible to access external objects in the DPL script. The special general selection object (“SEL”) is used to give all DPL functions and their subroutines access to a central selection of objects. i.e. the DPL Command Set.

Although flexible, this method would create problems if more than one specific object should be accessed in the script. By creating references to those objects in the DPL command itself, the DPL command would become specific to the current calculation case. Gathering the objects in the general selection would create the problem of selecting the correct object.

To prevent the creation of calculation-specific DPL commands, it is recommended practice to reserve the DPL contents for all objects that really “belong” to the DPL script and which are thus independent on where and how the script is used. Good examples are load-flow and short-circuit commands, or the vector and matrix objects that the DPL command uses for its computations.

If a DPL script must access a database object dependent on where and how the DPL script is used, an “External Object” must be added to the external object list in the DPL root command. Such an external object is a named reference to an external database object. The external object is referred to by that name. Changing the object is then a matter of selecting another object.

In Figure 23.1.3, an example of an external object is given. This external object may be referred to in the DPL script by the name “Bar1”, as is shown in the example.

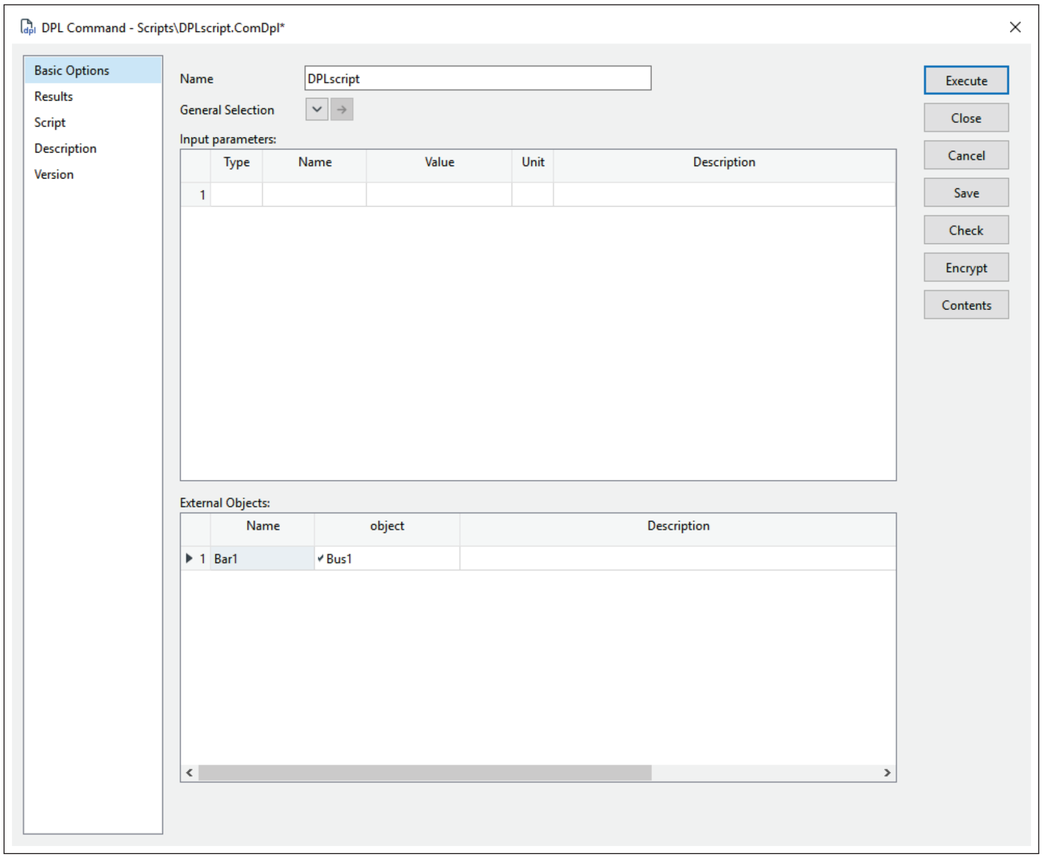


Figure 23.1.3: DPL external object table

Example:

```
sagdepth = Bar1:u;
```

23.1.9 Remote Scripts and DPL Command Libraries

To understand the DPL philosophy and the resulting hierarchical structure of DPL scripts, it is important to understand the following:

- A DPL command either executes its own script or the script of another, remote, DPL command. In the first case, the DPL command is called a "***root command***" and the script is called a "***local script***". In the second case, the DPL command is called a ***"referring" command*** and the script is called a ***"remote script"***.
- A root command may define interface variables that are accessible from outside the script and which are used to define default values.
- Each root command may define one or more ***external objects***. External object are used to make a DPL command run with specific power system objects, selections, commands, etc.
- A referring command may overrule all default interface values and all selected external objects of the remote command.
- Each DPL command can be called as a subroutine by other DPL commands.

The use of remote scripts, external objects and interface variables makes it possible to create generic DPL commands, which may be used with different settings in many different projects and study cases.

The easiest way to develop a new DPL command is to create a new *ComDpl* in the currently active study case and to write the script directly in that DPL object. In such a way, a DPL "root command" is made. If this root command needs DPL subroutines, then one or more DPL command objects may be created in its contents. Each of these subroutines will normally also be written as root functions.

The newly written DPL command with its subroutines may be tested and used in the currently active study case. However, it cannot be executed when another study case is active. In order to use the DPL command in other study cases, or even in other projects, one would have to copy the DPL command and its contents. This, however, would make it impossible to alter the DPL command without having to alter all its copies.

The solution is in the use of "remote scripts". The procedure to create and use remote scripts is described as follows.

Suppose a new DPL command has been created and tested in the currently active study case. This DPL command can now be stored in a safe place making it possible to use it in other study cases and projects.

This is done by the following steps:

- Copy the DPL command to a library folder. This will also copy the contents of the DPL command, i.e. with all its DPL subroutines and other locally stored objects.
- "Generalise" the copied DPL command by resetting all project specific external objects. Set all interface variable values to their default values. To avoid deleting a part of the DPL command, make sure that if any of the DPL (sub)commands refers to a remote script, all those remote scripts are also stored in the library folder.
- Activate another study case.
- Create a new DPL command (*ComDpl*) in the active study case.
- Set the "Remote script" reference to the copied DPL command.
- Select the required external objects.
- Optionally change the default values of the interface variables
- Press the **Check** button to check the DPL script

The **Check** or **Execute** button will copy all parts of the remote script in the library that are needed for execution. This includes all subroutines, which will also refer to remote scripts, all command objects, and all other objects. Some classes objects are copied as reference, other classes are copied completely.

The new DPL command does not contain a script, but executes the remote script. For the execution itself, this does not make a change. However, more than one DPL command may now refer to the same remote script. Changing the remote script, or any of its local objects or sub-commands, will now change the execution of all DPL commands that refer to it.

Note: *PowerFactory* is delivered with several ready-to-use scripts, which are located in the corresponding folder in the global library. They can be used as root commands for remote scripts or adapted as required, enhancing their functionality. The description and version page contain information about their functionalities, parameters and handling.

23.1.9.1 Subroutines and Calling Conventions

A DPL command may be included in the contents of another DPL command. In that case, the included DPL "subroutine" may be called in the script of the enclosing DPL command. In principle, this is not different from calling, for example, a load-flow command from a DPL script.

As with most other commands, the DPL command only has one method:

int Execute(); executes the DPL script.

The difference is that each DPL subroutine has different interface parameters, which may be changed by the calling command. These interface parameters can also be set directly at calling time, by providing one or more calling arguments. These calling arguments are assigned to the interface parameters in order of appearance. The following example illustrates this.

Suppose we have a DPL sub-command “Sub1” with the interface section as depicted in Figure 23.1.4.

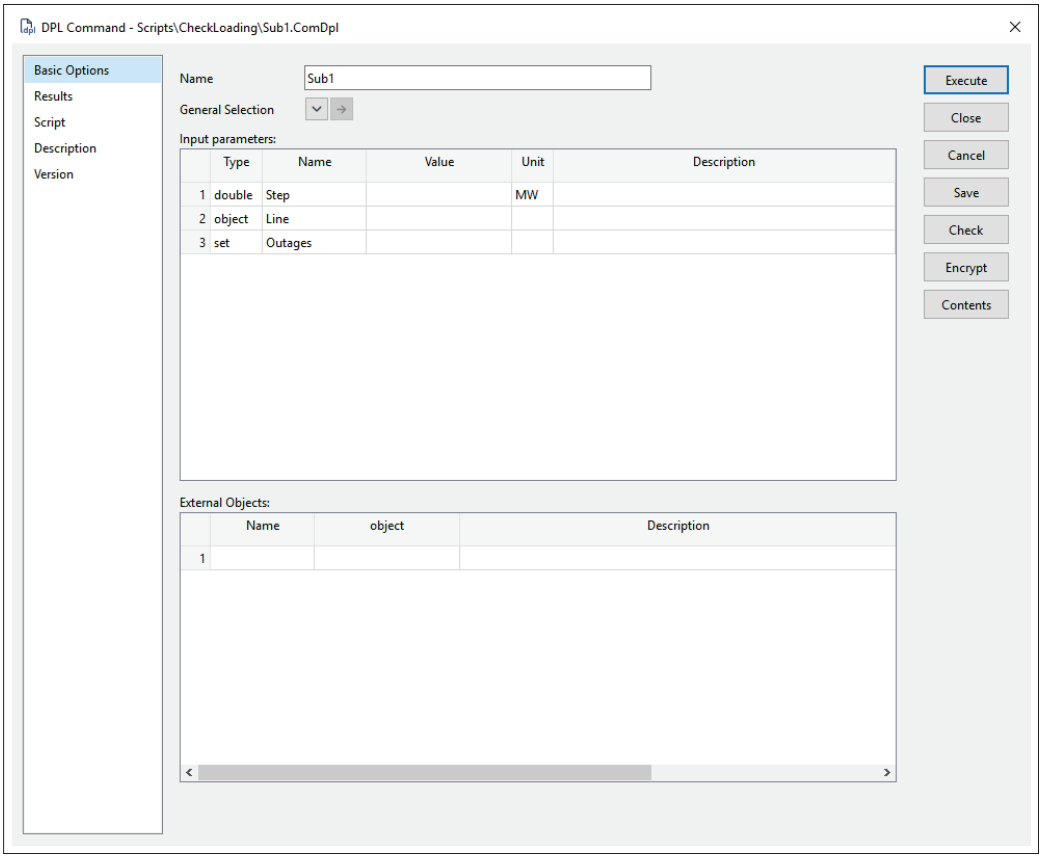


Figure 23.1.4: Interface section of subroutine

The calling command may then use, for example:

```
! set the parameters:
Sub1:step    = 5.0;
Sub1:Line    = MyLine;
Sub1:Outages = MySelection;
! execute the subroutine:
error = Sub1.Execute();
```

However, using calling arguments, we may also write:

```
! execute the subroutine:
error = Sub1.Execute(5.0, MyLine, MySelection);
```

23.1.10 DPL Functions and Subroutines

The DPL syntax is very small because it mainly serves the purpose of basic operations like simple calculations, if-then-else selections, do-while loops, etc..

The strength of the DPL language is the possibility to call functions and to create subroutines. A function which can be called by a DPL command is called a "method". Four types of methods are distinguished:

Internal methods These are the build-in methods of the DPL command. They can always be called.

Set methods These methods are available for the DPL "set" variables.

Object methods These methods are available for the DPL "object" variables.

External methods These are the methods which are available for certain external *PowerFactory* objects, such as the load-flow command, the line object, the asynchronous machine, etc.

Refer to the DPL Reference for a description of these functions including implementation examples. The DPL Reference is also accessible selecting *Help* → *Scripting References*→ *DPL* from the main menu.

23.2 Tabular Reports

Tabular reports are a powerful tool for generating your own personalised data views and showing all the information you want to extract from the *PowerFactory* model in one table. Before starting to create your own tabular reports, you should be familiar with the DPL programming language (Section 23.1). The command *ComTablereport* extends the DPL programming language with a simple framework to create tables. The command *ComTablereport* is not available in Python. In Python the use of a external GUI framework like TKInter or Qt is recommended. It is also possible to create a Tabular Report with DPL and use Python subroutines to calculate the contents.

Tabular Reports provide the following features:

- Display user-defined data in a table:
  - Allows the user to sort the table by each column.
  - Allows the user to use a data filter for each column.
  - Provides a callback function to add cell data editing features.
  - Provides a callback function to add additional entries to cells context menu.
  - Allows copy and paste of the table data.
- Allows the addition of global programmable selection filters and input data.
- Allows the addition of extra buttons to trigger user defined actions.
- Allows a user-defined data plot to be displayed instead of the table.
- Allows the user to export the table content to a HTML or Excel file.

23.2.1 Basic Structure of a Tabular Report

A Tabular Report always consists of a *ComTablereport* object and one or more callback *ComDpl* objects as children. The *ComTablereport* only supports the following predefined set of *ComDpl* objects:

- **Init** *optional* The Init script is called only once, when the report is displayed the first time.
- **Create** *mandatory* The Create script is called every time, when the report is displayed or rebuilt.

- **Edit** *optional* The Edit script is called after the user changes a cell.
- **Action** *optional* The Action script is called after the user selects a user-defined entry from the context menu.
- **ButtonPressed** *optional* The ButtonPressed script is called after the user presses a button.

Only the *Create* object is mandatory. The concept is quite simple. The whole report is defined with the *Create* function. If something changes, the whole report has to be rebuilt from scratch with the *Create* function.

23.2.2 The Table Report Command

The Tabular Report element *ComTablereport* itself contains only a few attributes:

- **Name** *loc_name* The name of the ComTablereport object.
- **Use Selection** *iSelection* A flag indicating whether the report use a Selection as input.
- **Class Filter** *sFilter* A class filter, which is applied to the input selection to extract only the relevant elements for the report.
- **Description attributes** Additional attributes to provide a short and a detailed description.
- **Version attributes** Additional attributes to provide version, author and copyright information.

The *ComTablereport* object provides a wide variety of functions, to define the tabular report, as shown in Figure 23.2.1 below. Please refer to the function descriptions in the DPL Reference.

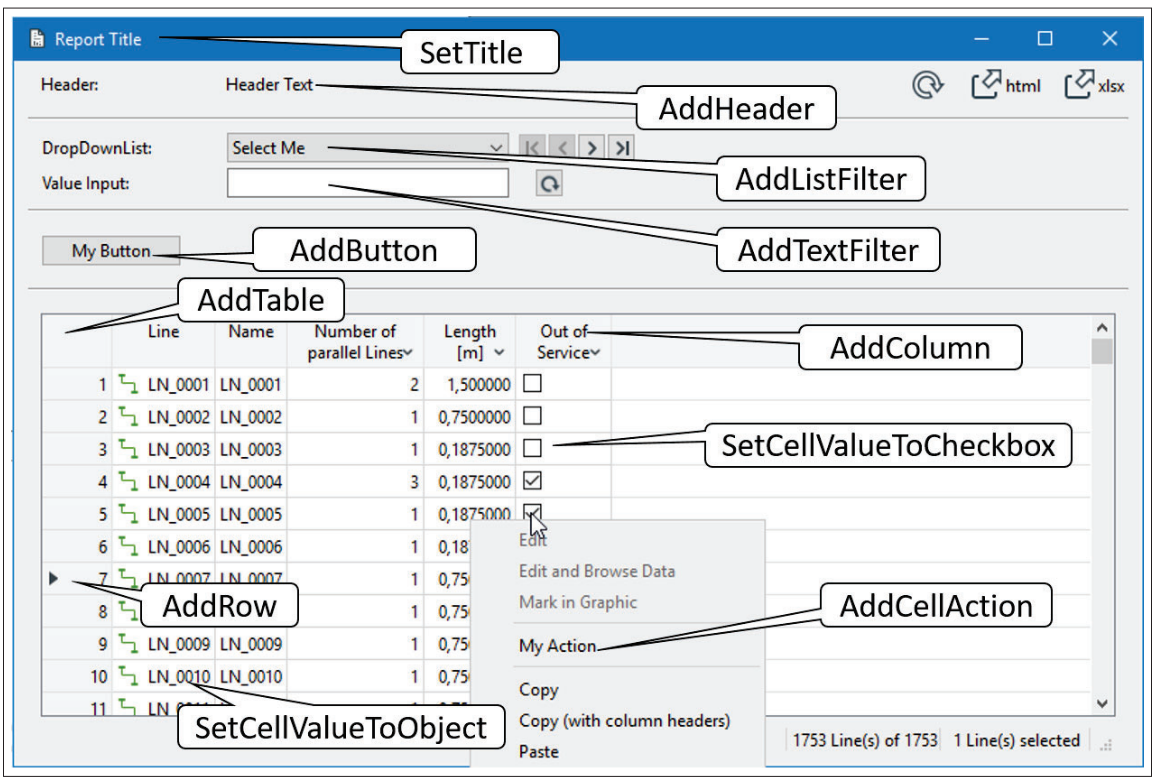


Figure 23.2.1: ComTablereport functions

23.2.2.1 Creating a new Table Report Command

A tabular report command element *ComTablereport* can be created by using the *New Object* () icon in the toolbar of the Data Manager and selecting *Others*. If the *Filter* property is set to *Commands (Com*)*, it is possible to select *Table Report (ComTablereport)* from the *Element* list.

After creating the *ComTablereport* object, a new *ComDpl* object with the name *Create* has to be added as a child of the *ComTablereport* object. See 23.2.3 for an example. Other *ComDpl* functions (see 23.2.6) may be added optionally.

23.2.2.2 Executing a Table Report Command

To execute and show a Tabular Report:

- Right-click at the *ComTablereport* object and choose *Execute* from the context menu.
- or Open or Edit the *ComTablereport* object and press the *Execute* button.

If the Tabular Report uses a Selection:

- Select some elements in a single line diagram or in the data manager.
- Then right-click and choose from the context menu “Execute Table Report”
- Then select a *ComTablereport* object from the list. (This step is omitted if there is only one Report with a Selection.)

23.2.3 A minimal Tabular Report

A minimal tabular report is built from a *ComTablereport* with a DPL script *ComDpl Create* as child. Section 23.2.2.1 describes how to create these. With the following code in the *Create* script, the report will show a table with 2 columns and a row with the words *Hello World*.

Example 1 “Hello world report” *Create* script**:**

```
object oReport;
oReport = this.GetParent();
oReport.AddTable('Table ID');
oReport.AddColumn('Table ID', 'Column 1 ID', 'First Column');
oReport.AddColumn('Table ID', 'Column 2 ID', 'Second Column');
oReport.AddRow('Table ID', 'Row 1 ID');
oReport.SetCellValueToString('Table ID', 'Column 1 ID', 'Row 1 ID', 'Hello');
oReport.SetCellValueToString('Table ID', 'Column 2 ID', 'Row 1 ID', 'World');
```

As the first step, the script has to access the *ComTablereport* object. This command object is always the parent of the *Create* script. So a simple *this.GetParent()* will work. As the second step, a table with an identifier is defined with the function *AddTable('Table ID')*. Use this table identifier in all subsequent functions to define the table. In the next step the table columns will be defined. With each call to the *AddColumn* function a new column is defined with its own column identifier and a column header text. In a similar way the rows of the table will be defined. The function *SetCellValueToString* fills the content of a cell with a string value. To refer a specific cell, the defined table, column and row identifiers must be used.

To test the report, you have to execute the tabular report (see 23.2.2.2), but not the *Create* script.

23.2.4 Handling different kinds of data

In the first example, the function *SetCellValueToString* was used to fill the table cells with content. To view different kinds of data, there are several similar functions. See the DPL Reference for a description of all these functions. The following example demonstrates the use of some of these functions. This example requires an active project with at least some lines (*ElmLne* objects) in the network model. You could use one of the *PowerFactory* example projects to test the report.

Example 2 "Lines report" *Create* script**:**

```
set     aLines;
object  oReport,
        oLine;
string  sRowId;

oReport = this.GetParent();
oReport.AddTable('Table ID');
oReport.AddColumn('Table ID', 'line column', 'Line');
oReport.AddColumn('Table ID', 'name column', 'Name');
oReport.AddColumn('Table ID', 'parallel column', 'Number of\nparallel Lines');
oReport.AddColumn('Table ID', 'length column', 'Length\n[m]');
oReport.AddColumn('Table ID', 'out column', 'Out of\nService');

aLines = GetCalcRelevantObjects('*.ElmLne');
for(oLine = aLines.First(); oLine; oLine = aLines.Next()){
  sRowId = oLine.GetFullName();
  oReport.AddRow('Table ID', sRowId);
  oReport.SetCellValueToObject('Table ID', 'line column', sRowId, oLine);
  oReport.SetCellValueToString('Table ID', 'name column', sRowId, oLine:loc_name);
  oReport.SetCellValueToInt('Table ID', 'parallel column', sRowId, oLine:nlnum);
  oReport.SetCellValueToDouble('Table ID', 'length column', sRowId, oLine:dline);
  oReport.SetCellValueToCheckbox('Table ID', 'out column', sRowId, oLine:outserv);
}
```

Some of these functions not only show plain data, but provide additional features. If a *PowerFactory* object is shown with the function *SetCellValueToObject*, a double click in the cell opens the edit dialog of this object. In addition, there is a new entry in the context menu of the cell, which will mark the object in a graphic. The *SetCellValueToCheckbox* expects a boolean value, which is displayed as a checkbox.

23.2.5 Advanced Features

23.2.5.1 Programmable Filters and Input Values

The *ComTablereport* function *AddListFilter* adds a drop-down list to the report. The drop-down list has an identifier and a visible name. The function *AddListFilterEntries* can be used to add entries to the list. To get the selected entry, the *Create* script has to define a *string* input parameter with the name *s<identifier>*. When the *Create* script is called the first time to generate the report, this parameter contains the default value, or an empty string if no default value is set. Every time the user selects a new entry from the drop-down-list or another event refreshes the report, the *Create* script is called again with the selected entry in the input parameter *s<identifier>* to recreate the report. In the example below, the drop-down list is given the identifier UFilt. So the *Create* script should contain a *string* input parameter *sUFilt*. This parameter defines the currently selected entry. The function SetListFilterSelection can be used to set the selected entry in the drop-down list.

Example 3 "Drop-down list report" *Create* script**:**

```
object oReport;
oReport = this.GetParent();
oReport.AddListFilter('UFilt', 'Usage');
```

```
oReport.AddListFilterEntries('UFilt', 'Busbar');
oReport.AddListFilterEntries('UFilt', 'Junction');
oReport.AddListFilterEntries('UFilt', 'Internal');
oReport.SetListFilterSelection('UFilt', sUFilt);
```

If all the list information is available at once, it is possible to shorten the above code and create the whole drop-down list with one call to *AddListFilter*. To distinguish the list entries, they have to be separated by "\n".

Example 3 "Drop-down list report" *Create* script change**:**

```
oReport.AddListFilter('UFilt','Usage:','Busbar\nJunction\nInternal',aDummy,sUFilt);
```

With this code it is possible to create different views of the report, depending on the input parameter *sUFilt*.

The *ComTablereport* function *AddTextFilter* creates an additional text input field in the report. In contrast to the drop-down list, the report will not trigger an automatic rebuild if the user changes the text. Instead, the user has to press the **Refresh** button at the end of the text field. To obtain the user input, the *Create* script has to define the input parameter *s<identifier>*

Example 4 "Input value report" *Create* script**:**

```
object oReport,
oReport = this.GetParent();
oReport.AddTextFilter('TFilt', 'Text input', '', sTFilt);
```

The *Create* script can also modify the user input. For example, if the user must be able to input numbers, the script code can just convert the input in a number and discard any invalid input data.

Example 4 "Input value report" *Create* script**:**

```
object oReport,
string sCurrentValue,
int    iNumber, iRes;

oReport = this.GetParent();
iNumber = 0;
iRes = sscanf(sNFilt, '%d', iNumber);
sCurrentValue = sprintf('%d', iNumber);
oReport.AddTextFilter('NFilt', 'Nominal Voltage above', 'KV', sCurrentValue);
```

The above code always resets the input to 0 if the user enters invalid data. Sometimes it may be useful to remember the last valid value to restore. In this case a hidden filter can be used with the function *AddInvisibleFilter*. The hidden filter does nothing but preserve the data between 2 consecutive calls of the Create function. Again the *Create* script has to define the *string* input parameter *s<identifier>* to get the value back in the call of the function.

Example 4 "Input value report" *Create* script**:**

```
  object oReport,
  string sCurrentValue,
  int    iNumber, iRes;

  oReport = this.GetParent();
  iNumber = 0;
  iRes = sscanf(sOldValue, '%d', iNumber);
  iRes = sscanf(sNFilt, '%d', iNumber);
  sCurrentValue = sprintf('%d', iNumber);
  oReport.AddInvisibleFilter('OldValue', sCurrentValue, NULL);
  oReport.AddTextFilter('NFilt', 'Nominal Voltage above', 'KV', sCurrentValue);
```

The third parameter of the function *AddInvisibleFilter* takes a *PowerFactory* object as input. To get this object back in the next call, the *Create* script has to define an external object *o<identifier>* as input.

23.2.5.2 Edit Cells

The *ComTablereport* function *SetCellEdit* makes an individual cell editable. To identify the cell the defined table, column and row identifiers must be used. In addition, the callback *ComDpl* script *Edit* must be provided as a child of the *ComTablereport*. This callback report takes five input parameters:

- **sColumnId** *string* The column identifier of the changed cell.
- **sRowId** *string* The row identifier of the changed cell.
- **sNewValue** *string* The new value, if the cell contains a string.
- **iNewValue** *int* The new value, if the cell contains an int.
- **dNewValue** *double* The new value, if the cell contains a double.

It is the responsibility of the programmer to change the underlying value with the *Edit* callback script. With the help of this script the user input can also be changed or rejected, if it is invalid. Every time the user changes a cell, the *Edit* callback script is called. Following the execution of the *Edit* script, the Tabular Report is rebuilt with a call of the *Create* script. To assist the edit process, a set of cell-specific objects can be passed with the call of the function *SetCellEdit*. These objects will be available in the *Edit* callback script as external input objects:

- **oEditObject1** The first object passed from the *Create* script with *SetCellEdit*.
- **oEditObject2** The second object passed from the *Create* script with *SetCellEdit*.
- **oEditObject<X>** The Xth object passed from the *Create* script with *SetCellEdit*.

As an example of a tabular report with editable cells, you can extend the script from chapter 23.2.4. Add a new *set* variable to the **Create** script at line 1.

Example 2 "Lines report" *Create* script extension**:**

```
set aEditSet;
```

Add the following code to the *Create* script before line 23.

Example 2 "Lines report" *Create* script extension**:**

```
  aEditSet.Clear();
  aEditSet.Add(oLine);
  oReport.SetCellEdit('Table ID', 'name column', sRowId, aEditSet);
  oReport.SetCellEdit('Table ID', 'parallel column', sRowId, aEditSet);
  oReport.SetCellEdit('Table ID', 'length column', sRowId, aEditSet);
  oReport.SetCellEdit('Table ID', 'out column', sRowId, aEditSet);
```

Add a new *ComDpl* object *Edit* to the tabular report. Define the input parameter (see above) and an external object *oEditObject1*. Fill the script content of the *Edit* script with the following code:

Example 2 "Lines report" *Edit* script**:**

```
object oLine;
int    iColumnCmp;
oLine = oEditObject1;
iColumnCmp = strcmp(sColumnId, 'name column');
if (iColumnCmp = 0){
```

```
  oLine:loc_name = sNewValue;
}
iColumnCmp = strcmp(sColumnId, 'parallel column');
if (iColumnCmp = 0){
  oLine:nlnum = iNewValue;
}
iColumnCmp = strcmp(sColumnId, 'length column');
if (iColumnCmp = 0){
  oLine:dline = dNewValue;
}
iColumnCmp = strcmp(sColumnId, 'out column');
if (iColumnCmp = 0){
  oLine:outserv = iNewValue;
}
```

With this script all visible attributes of the line could be changed. **Note:** this example modifies the attributes of the lines of your active project!

23.2.5.3 Additional Context Menu Entries

The *ComTablereport* function *AddCellAction* adds a new entry to the context menu of a cell. To identify the cell the defined table, column and row identifiers must be used. The fourth parameter is the *Action Identifier* and the fifth parameter defines the text of the context menu entry. In addition, the callback *ComDpl* script *Action* must be provided as a child of the *ComTablereport*. This callback script takes two input parameters:

- **sActionId** *string* The *Action Identifier*, which was defined with the call of *AddCellAction*.
- **aObjects** *set* The set of objects, which were passed through with the call of *AddCellAction*.

The callback function *Action* is always called if the user selects an additional entry from the cell context menu. The function *AddCellAction* provides an optional parameter *refresh* to define whether the report shall be rebuilt after the execution of the *Action* script or not. The default is set to rebuild the report with a call of the *Create* script.

23.2.5.4 Additional Buttons

The *ComTablereport* function *AddButton* adds a new button with an identifier and a label to the top of the table report. The callback *ComDpl* script *ButtonPressed* must be provided as a child of the *ComTablereport*. This callback script takes one input parameter.

- **sButtonId** *string* The *Button Identifier* which was defined with the call of *AddButton*.

The callback function *ButtonPressed* is always called if the user presses a button in the tabular report. The function *AddButton* provides an optional parameter *refresh*, which defines whether the report shall be rebuilt after the execution of the *ButtonPressed* script or not. If the callback script *ButtonPressed* changes the content of table, the parameter *refresh* will be always set to 1.

23.2.5.5 Using a Selection

If the tabular report must be able to show user defined pre-selected elements, it must have set the **Use Selection** *iSelection* attribute of the *ComTableReport*. See chapter 23.2.2 for a description of the *ComTableReport* attributes. In addition, the report must be stored in the report folder *Library\TableReport* of the project. The exact name of the report folder depends on the *PowerFactory* language settings at the time of project creation. If both conditions are fulfilled, the report can be executed with a user selection (see 23.2.2.2). Inside the *Create* or *Init* callback scripts the *ComTablereport* function

GetSelection provides access to this selection. If the *ComTablereport* attribute **Class Filter** *sFilter* is set, the function *GetSelectedElements* could be used instead. This function returns an already filtered set of selected objects.

23.2.6 Table Report Callback Script Reference

23.2.6.1 Init

- optional
- called only when the report is opened the first time
- no input parameter

23.2.6.2 Create

- mandatory
- called if the report is created or rebuilt
- input parameter passed through from *AddInvisibleFilter*, *AddListFilter*, *AddMultiListFilter*, *AddTabularFilter* or *AddTextFilter*:
  - **s<FilterId>** *string* The selected value of a filter.
- external input objects passed through from *AddInvisibleFilter*, *AddListFilter* or *AddMultiListFilter*:
  - **o<FilterId>** The selected object of a filter.

23.2.6.3 Edit

- optional
- called if:
  - a cell defined with *SetCellEdit* is changed by the user *or*
  - a filter defined with *AddInvisibleFilter*, *AddListFilter*, *AddMultiListFilter*, *AddTabularFilter* or *AddTextFilter* is changed by the user
- input parameters:
  - **sColumnId** *string* The column identifier of the changed cell or the filter identifier of the changed filter.
  - **sRowId** *string* The row identifier of the changed cell.
  - **sNewValue** *string* The new value, if the cell contains a string, or the new value of the changed filter.
  - **iNewValue** *int* The new value, if the cells contains an integer.
  - **dNewValue** *double* The new value, if the cells contains a double.
- external input objects:
  - **oEditObject1** The first object passed from the Create script with *SetCellEdit* or the selected object of a changed filter.
  - **oEditObject2** The second object passed from the Create script with *SetCellEdit*.
  - **oEditObject<X>** The Xth object passed from the Create script with *SetCellEdit*.

23.2.6.4 Action

- optional
- called if the user selects an additional context menu entry defined with *AddCellAction*.
- input parameters:
  - **sActionId** *string* The *Action Identifier* which was defined with the call of *AddCellAction*.
  - **aObjects** *set* The set of objects, which were passed through with the call of *AddCellAction*.

23.2.6.5 ButtonPressed

- optional
- called if the user presses a button defined with *AddButton*.
- input parameter:
  - **sButtonId** *string* The *Button Identifier* which was defined with the call of *AddButton*.

23.3 Python

This section describes the integration of the Python scripting language in *PowerFactory* and explains the procedure for developing Python scripts. The Python scripting language can be used in *PowerFactory* for:

- Automation of tasks
- Creation of user-defined calculation commands
- Integration of *PowerFactory* into other applications

Some of Python's notable features include:

- General-purpose, high-level programming language
- Clear, readable syntax
- Non-proprietary, under liberal open source licence
- Widely used
- Extensive standard libraries and third-party modules
  - Interfaces to external databases and Microsoft Office-like applications
  - Web services, etc.

The integration of Python into *PowerFactory* makes the above-mentioned features accessible to users of *PowerFactory*.

To execute a Python script the following steps have to be considered:

1. Python interpreter has to be installed. (Subsection 23.3.1)
2. Python file .py that contains code of the script has to be created by external editor. After being created .py file can be link to the *ComPython* object inside of *PowerFactory*. (Subsection 23.3.3)
3. In each .py file *PowerFactory*Python module 'powerfactory.pyd' has to be imported. (Subsection 23.3.2)
4. To run the script the User has to execute the (*ComPython*) object. (Subsection 23.3.3.2)

23.3.1 Installation of a Python Interpreter

When *PowerFactory* is installed, the installation does not include a Python interpreter and therefore this must be installed separately. The recommended Python versions are available on `https://www.python.org/downloads/`. All supported Python versions can be checked inside of the *PowerFactory* Configuration dialogue (see Figure 23.3.1) here is also where a preferred Python version can be selected.

The *PowerFactory* architecture (32- or 64-bit) determines the Python interpreter as shown below:

- *PowerFactory* 32-bit requires a Python interpreter for 32-bit
- *PowerFactory* 64-bit requires a Python interpreter for 64-bit

To check which *PowerFactory* architecture is installed, press **Alt-H** to open the Help menu and select *About* PowerFactory. If the name of *PowerFactory* includes "(x86)", then a 32-bit version is installed; if the name of *PowerFactory* instead includes "(x64)", then a 64-bit version is installed. To avoid issues with third-party software, the Python interpreter should be installed with default settings (for all users), into the directory proposed by the installer.

Depending on the functions to be performed by a particular Python script, it may be necessary to install a corresponding Python add-on/package. For example, Microsoft Excel can be used by Python scripts if the "Python for Windows Extensions" *PyWin32* (`http://sourceforge.net/projects/pywin32/`) package is installed, which includes Win32 API, COM support and Pythonwin extensions.

23.3.2 The Python *PowerFactory* Module

The functionality of *PowerFactory* is provided in Python through a dynamic Python module ("powerfactory.pyd") which interfaces with the *PowerFactory* API (Application Programming Interface). This provides Python scripts with access to a comprehensive range of data in *PowerFactory*:

- All objects
- All attributes (element data, type data, results)
- All commands (load flow calculation, etc.)
- Most special built-in functions (DPL functions)

A Python script which imports this dynamic module can be executed from within *PowerFactory* through the new Python command *ComPython* (see Section 23.3.3), or externally (*PowerFactory* is started by the Python module in non-interactive mode) (see Section 23.3.4).

23.3.2.1 Python *PowerFactory* Module Usage

To allow access to the Python *PowerFactory* module it must be imported using the following Python command:

```
import powerfactory
```

To gain access to the *PowerFactory* environment the following command must be added:

```
app = powerfactory.GetApplication()
```

A Python object of class powerfactory.Application is called an application object. Using the application object from the command above("app"), it is possible to access global *PowerFactory* functionality. Several examples are shown below:

```
user = app.GetCurrentUser()
project = app.GetActiveProject()
script = app.GetCurrentScript()
objects = app.GetCalcRelevantObjects()
lines = app.GetCalcRelevantObjects("*.ElmLne")
sel = app.GetDiagramSelection()
sel = app.GetBrowserSelection()
project = app.CreateProject("MyProject", "MyGrid")
ldf = app.GetFromStudyCase("ComLdf")
```

The listed methods return a data object (Python object of class powerfactory.DataObject) or a Python list of data objects. It is possible to access all parameters and methods associated with a data object. Unlike DPL syntax, Python syntax requires use of the dot (.) operator instead of the colon (:) in order to access element parameters of objects (i.e. name, out of service flag, etc.).

Examples:

```
project = app.GetActiveProject()
projectName = project.loc_name
project.Deactivate()
```

All other object parameters (calculated, type, measured, . . .) are to be called by GetAttribute() method and using the colon (:), as is done in DPL.

```
lines = app.GetCalcRelevantObjects("*.ElmLne")
line = lines[0]
currLoading = line.GetAttribute("c:loading")
```

For printing to the *PowerFactory* output window, the following application object (e.g. "app" object) methods are provided:

```
app.PrintPlain("Hello world!")
app.PrintInfo("An info!")
app.PrintWarn("A warning!")
app.PrintError("An error!")
```

Printing the string representation of data objects to the *PowerFactory* output window makes them selectable (i.e. creates a hyperlink string in the output window):

```
project = app.GetActiveProject()
app.PrintPlain("Active Project: " + str(project))
```

A list of all parameters and methods associated with an object can be obtained using the `dir()` function as shown below:

```
project = app.GetActiveProject()
app.PrintPlain(dir(project))
```

23.3.2.2 Python *PowerFactory* Module Reference

A Python Module Reference document is available in the Help menu containing a list of offered functions. Python reference

23.3.3 The Python Command (*ComPython*)

In contrast to DPL, the Python command only links to a Python script file. It stores only the file path of the script and not the file itself.

The script may be executed by clicking on the **Execute** button of the corresponding dialog. Editing the script file is possible by clicking the **Open in External Editor** button.

The preferred editor may be chosen in the *External Applications* page of the PowerFactory *Configuration* dialog by selecting the *Tools* → *Configuration...* menu item from the main menu (see section 5.2.4). Python scripts may be created in any text editor as long as the script file is saved using the UTF-8 character encoding format. The Python version can be selected in the *PowerFactory Configuration* dialog.

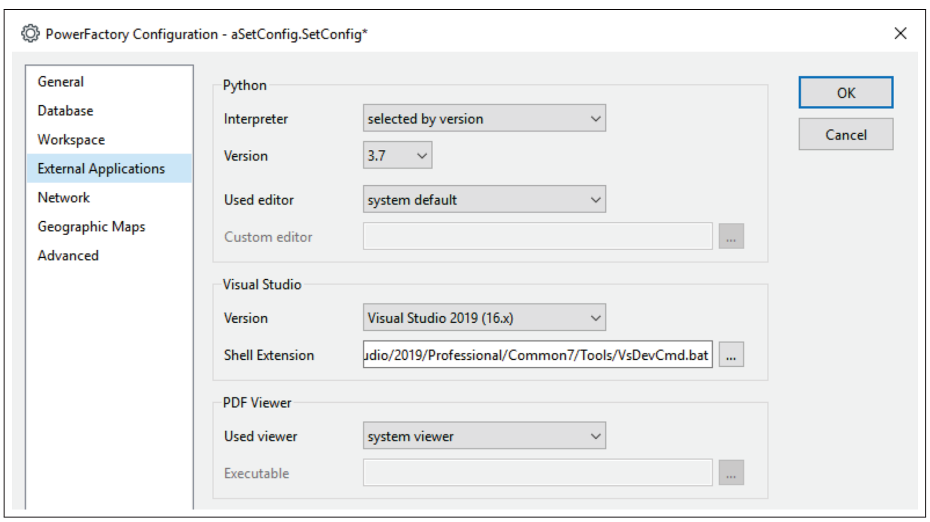


Figure 23.3.1: Selection of a preferred Python editor program

On the *Basic Options* page of the Python command (*ComPython*), the user can define *Input parameters* and *External objects*. The *Input parameter* table in the dialog is used as in DPL to define variables that can be accessed from outside of the Python script. *Input parameters* may be the following data types: double, int and string. All other fields (Name, Value, Unit, Description) are user definable.

The *External object* table allows the direct configuration of objects under investigation. An external object is an object external to the Python command that the user wants to access in the script. By defining the *External object* here, the user avoids accessing it via Python methods inside the script (thereby allowing the script to execute faster).

Important: To access an external object or input parameter in Python, the user has first to get the script and can then access the parameter through ScriptObj.ExternObjName:

```
script=app.GetCurrentScript()  #to call the current script
extObj=script.NameOfExternObj #to call the external object
```

```
inpPar=script.NameOfInpPram    #to call input parameter
```

Result parameter section is to be found on the *Results* page. The *Result parameters* represents results from the script and they are stored inside the specified results object. The *Script* page contains three sections:*Remote script*,*Script file* and *Interface version*. *Remote script* offers a selection of a remote script, which is used instead of the code defined in the *Script file* section. A remote script can be advantageous in cases where the user has multiple Study Cases using the same script code, but different input parameters. If the user wants to use a remote script, then modifying the master script will affect all the Python scripts that refer to it. If the user had locally-defined scripts, then they would need to change the code in each of them.

Python Script: The source of the Python script can be selected. Two options are available:

- *External*: An external *.py Python script file has to be selected under *Script file*.
- *Embedded*: The script is embedded in the ComPython-object and can directly be developed using the internal editor (see Section 23.4). Please note that only single file Python scripts are supported.

Script file will be available if *External* (see above) is selected. It is a field that contains the path of the Python script file (*.py).

Embedded Code: Available if option *Embedded* (see above) is selected. The Python script is embedded in the ComPython-object and can directly be developed in the editor.

Interface Version refers to the returned value of some Python methods. For example if selecting

Version 1-“old interface version”. Data object methods with arguments such as the method GetPage() from the class SetDesktop return list containing [[returned value],argument of the following entries] :

```
list(DataObject, ...) SetDesktop::GetPage(str,[int])
```

Version 2- “new interface version”. Methods like the method GetPage() returns just the result, without input parameters.

```
DataObject SetDesktop::GetPage(str,[int])
```

The Python command may also contain objects or references to other objects available in the *PowerFactory* database. These can be accessed by clicking on the **Contents** button. New objects are defined by first clicking the *New Object* icon in the toolbar of the Python script contents dialog and then selecting the required object from the *New Object* pop-up window. References to other objects are created by defining a reference object “IntRef”. **Note**: Python supports different access to this objects than DPL. See example

```
script=app.GetCurrentScript()
ContainedObejct=script.GetContents('Results.ElmRes')
```

23.3.3.1 Creating a New Python Command

To create a new Python command click on the *New Object* ( ) icon in the toolbar of the Data Manager and select *DPL Command and more*. From the *Element* drop-down list select ’Python Script (*ComPython*)’. Then press **OK** and a new Python command will be created. The Python command dialog is then displayed. The name of the script, its input parameters and the file path to the script, etc, can now be specified. The Python command dialog is also opened by double-clicking a Python script; by selecting *Edit* from the context-sensitive menu or by selecting the script from the list after clicking on the main toolbar icon *Execute Script* ().

23.3.3.2 Executing a Python Command

A Python command may be executed by clicking the **Execute** button in the dialog.

Alternative methods for executing a Python script include:

- From the Data Manager:
  - Right-click on the Python command and select *Execute* from the context-sensitive menu.
  - Right-click in a blank area and select *Execute Script* from the context-sensitive menu. A list of existing DPL and Python scripts contained in the global and local libraries will pop up. Select the required Python script and click **OK**.
- From the single line diagram:
  - Select one or more elements in the single line diagram. Right-click the marked elements and select *Execute Script* from the context-sensitive menu. A list of existing DPL and Python scripts contained in the global and local libraries will pop up. Select the required Python script and click **OK**.
  - A button may be created in the single line diagram to automate the execution of a specific Python script.
- From the main toolbar:
  - Click the icon *Execute Script* . A list of existing DPL and Python scripts from the global and local libraries will appear. Select the specific Python script and click **OK**.

23.3.4 Running *PowerFactory* in Non-interactive Mode

PowerFactory may be run externally by Python. In order to do this, the script must additionally add the file path of the dynamic module (“powerfactory.pyd”) to the system path. Example:

```
# Add powerfactory.pyd path to python path.
import sys
sys.path.append("C:\\Program Files\\DIgSILENT\\PowerFactory 2020\\Python\\3.7")

#import PowerFactory module
import powerfactory

#start PowerFactory in non-interactive mode
app = powerfactory.GetApplication()

#run Python code below
#....................
```

The *PowerFactory* environment can be accessed directly from the Python shell as shown in Figure 23.3.2

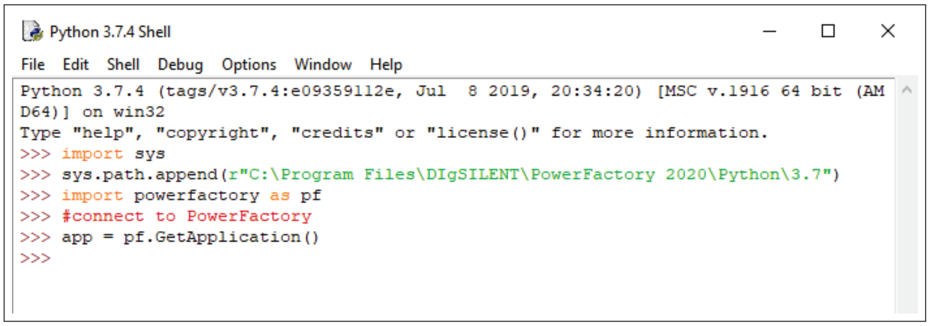


Figure 23.3.2: Python shell

Note: If an error message appears when importing the *powerfactory* module stating " DLL load failed: the specified module could not be found", this often means that the corresponding Microsoft Visual C++ Redistributable are not installed on the computer.

23.3.5 Performance of Python Scripts

The execution time of a Python script can vary strongly depending on the used environment set up. There are dedicated environment functions Python Reference to improve the performance of a script. One example for a possible performance improvement is disabling the *User Break*-button. This button has a huge impact on the execution time of some scripts, because every few simulation steps Python has to check, whether the button has been pressed. The *SetUserBreakEnabled()*-function allows to disable and enable the break button freely during the execution of a script.

23.3.6 Debugging Python Scripts

As with any other Python script, it is possible to remotely debug scripts written for *PowerFactory* by using specialised applications.

23.3.6.1 Prerequisites

The recommended IDE for debugging is Eclipse (`www.eclipse.org`) with the Python add-on PyDev (`www.pydev.org`).

1. Install Eclipse Standard from `www.eclipse.org/downloads/`
2. Open Eclipse
3. Click "Install New Software . . . " in the "Help" menu
4. Add the repository `http://pydev.org/updates` and install PyDev

23.3.6.2 Debugging a Python script for *PowerFactory*

The following is a short description of remote debugging with PyDev. For more information please consult the remote debugger manual of PyDev (`http://pydev.org/manual_adv_remote_debugger.html`).

1. Start Eclipse
2. Open Debug perspective

3. Start the remote debugger server by clicking “Start Debug Server” in the “Pydev” menu
4. Start *PowerFactory*
5. Prepare the Python script for debugging:
   - Add “pydevd.py” path to sys.path
   - Import PyDev debugger module “pydevd”
   - Start debugging calling pydevd.settrace()

   Example:

```
#prepare debug
import sys
sys.path.append ("C:\\Program Files\\eclipse\\plugins\\
              org.python.pydev_2.8.2.2013090511\\pysrc")
import pydevd

#start debug
pydevd.settrace()
```

6. Execute the Python script
7. Change to Eclipse and wait for the remote debugger server

It is not possible to stop and restart the remote debugger server while running *PowerFactory*.

23.3.7 Example of a Python Script

The following example Python script calculates a load flow and prints a selection of results to the output window. The script can be executed from within *PowerFactory*.

```
if __name__ == "__main__":
    #connect to PowerFactory
    import powerfactory as pf
    app = pf.GetApplication()
    if app is None:
        raise Exception("getting PowerFactory application failed")

    #print to PowerFactory output window
    app.PrintInfo("Python Script started..")

    #get active project
    prj = app.GetActiveProject()
    if prj is None:
        raise Exception("No project activated. Python Script stopped.")

    #retrieve load-flow object
    ldf = app.GetFromStudyCase("ComLdf")

    #force balanced load flow
    ldf.iopt_net = 0

    #execute load flow
    ldf.Execute()

    #collect all relevant terminals
    app.PrintInfo("Collecting all calculation relevant terminals..")
    terminals = app.GetCalcRelevantObjects("*.ElmTerm")
    if not terminals:
        raise Exception("No calculation relevant terminals found")
    app.PrintPlain("Number of terminals found: %d" % len(terminals))
```

```

    for terminal in terminals:
        voltage = terminal.GetAttribute("m:u")
        app.PrintPlain("Voltage at terminal %s is %f p.u." % (terminal , voltage))

    #print to PowerFactory output window
    app.PrintInfo("Python Script ended.")
```

23.4 Editor

PowerFactory has an integrated editor which can be used to write and execute scripts or as normal text editor. The editor can be reached by pressing “ctrl e” on the keyboard or from the script page of a ComDpl/ComPython dialog by right clicking somewhere in the code and selecting “Open Text Editor”. When the editor is opened in an additional window, the available tools are shown in the *Editor Toolbar*; note that the same tools are available when using the page *Script* of ComDpl/ComPython command by using the context sensitive menu.

Open external text file

Save file. If the file is a script the changes will be saved to the script object. Otherwise the file will be saved as a .txt file

Print the current file

Check the syntax of the script. Only available for DPL scripts

Execute the current script. The button is interpreted as save and execute.

A browser with the script contents is shown.

With this *Edit* icon the *edit* dialog of the script is opened.

Cut part of the text.

Copy part of the text.

Paste the copied or cut text.

The text of the script will be deleted

Undo the last operation.

Redo the last operation.

With the *Search* icon the user can activate a *Find*, a *Replace* or also a *Go To* function inside the editor.

Find/replace/go to the next matching word.

Find/replace/go to the previous matching word.

With this icon bookmarks can be set in the editor.

Go to the next bookmark.

Go to the previous bookmark.

Clear all the existing bookmarks.

Open the User Settings dialog in the *Editor* page. More information is given in section 7.7.

When editing is complete, press the icon or icon and the script will be synchronised with the main dialog.

In order to close an editor window click on the “close button” (“**X**”) on the top of the right side of the window directly underneath corresponding buttons on the title bar.

23.5 Add On Modules

The purpose of Add On Modules is to allow the user more flexibility in the processing of calculations and the presentation of results. By using Add On Modules the user can, for example, execute a number of different calculations and present the results together on a graphic or in a report. Furthermore, the concept of user-defined variables is introduced, these being variables created by the user in order to store results, parameters, or any other information relevant to the process. After a module has been executed, the user-defined variables may be accessed in the same way as the standard results variables, for example being selected to display in a flexible data page. An Add On Module is created and executed using an Add-On Command (*ComAddon*), which can be stored within a project or stored in a central location and accessed via the user-defined toolbar.

23.5.1 Add On Module framework

The Add On Module is created using an *ComAddon* command, which typically consists of the following components:

- A DPL or Python script, which:
  - uses a *CreateModule* command to create the Add On module;
  - may include a range of calculation commands such as *ComLdf* and *ComShc*;
  - includes commands to transfer results variables or other information to the user-defined variables;
  - uses a *FinaliseModule* command to complete the Add On module and make the results in the user-defined variables available to the user.
- One or more User-defined variable definitions *IntAddonvars*; these are where the user specifies a set of user-defined variables for a particular element class.

As can be seen in figure 23.5.1 below, the Add On command object also specifies two other pieces of information:

Module Name: This is the name that will be given to the module itself. When the command is executed, the module is created and this name will appear, for example, when the user selects variables to display on a flexible data page.

Module Key: This is a key used internally by *PowerFactory*. It is used as a reference so that the appropriate flexible data, graphic colouring and result boxes are used when the command is executed. It is possible, if appropriate, for several Add On commands to use the same module key.

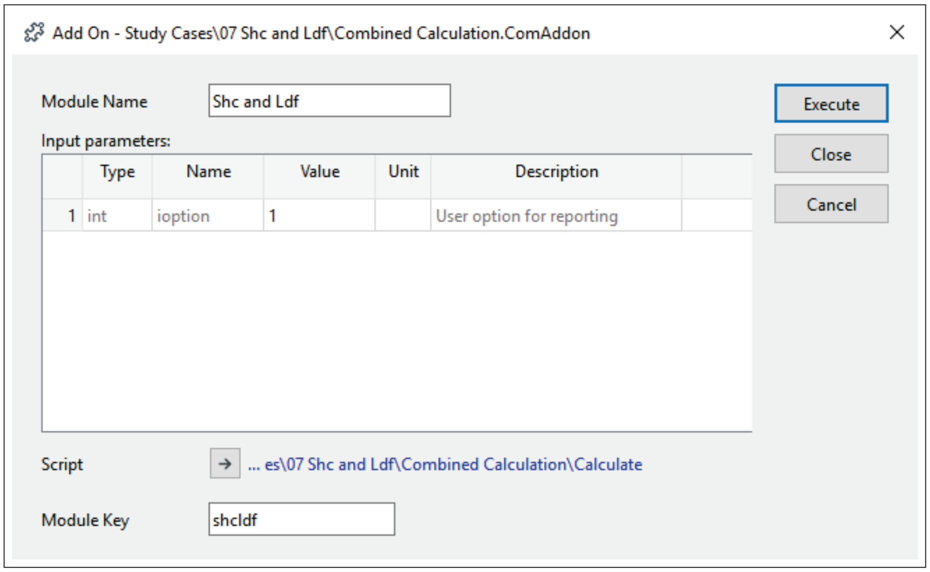


Figure 23.5.1: Add On Command object

So when the Add On command is executed, the Add On module that is created can be regarded as a bespoke *PowerFactory* function, handled in the same way as other functions such as a load flow or RMS simulation, but offering great flexibility for the user.

23.5.2 Creating a new Add-on Module command

To create a new Add On Module in a study case, the following steps can be followed:

- In a Data Manager, select the study case and click on the *New Object* icon .
- Select from the list *Others* and as *Filter* “Commands (Com*)”. Choose from the *Element* drop down list “Add On (ComAddon)” and confirm with OK.
- Give the Add On a name and a key and confirm the edit dialog with Close.
- Select the Add On on the left side in the tree hierarchy of the Data Manager and press the *New Object* icon again.
- Four options are available initially. The DPL Command (*ComDpl*) or Python Script(*ComPython*) can be selected to create the script for the module. The third option, Variable Definition of Add On (IntAddonvars) is used to create the user-defined variable definitions (*IntAddonvars*). The fourth option, Flexible Data (SetFoldflex), offers the possibility to define a new tab (like the flexible data page) and a predefined set of displayed variables.
- Once a script exists, any additional objects created in the module can only be user-defined variable definitions *IntAddonvars* or the definition of an additional flexible data page *SetFoldlfex*.
- The *IntAddonvars* is configured to set up the user-defined variables. In the example in figure 23.5.2 below, short-circuit results variables are defined, one for each phase, and also line loading, line length and an internal counter from the script.

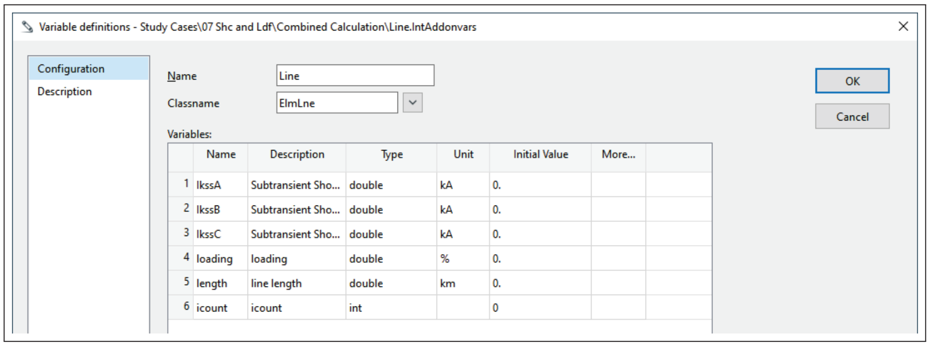


Figure 23.5.2: Example of user-defined variable definitions for line elements

The setup of an additional flexible data page could be done as follows:

- Create a Flexible Data definition *SetFoldflex* inside the Add On.
- Execute the Add On.
- Open the Network Model Manager and look at the object classes for which Add On variable definitions were created.
- Switch to the newly created, user defined flexible data tab at the bottom.
- Add user defined and whatever parameters of interest to the displayed variables via the *Variable Selection* .
- With the variable selection done for all object classes of interest, the corresponding settings can be copied into the Add On. Navigate in the Data Manager to the Settings folder of the project and within it to the newly created flexible data page.
- Copy the *Flexible Page Selector* objects (*IntMonsel*) and paste them inside the Add On Flexible Data definition (see figure 23.5.3).
- When this Add On is then executed in another project, the additional flexible data page will contain the configured sets of variables.

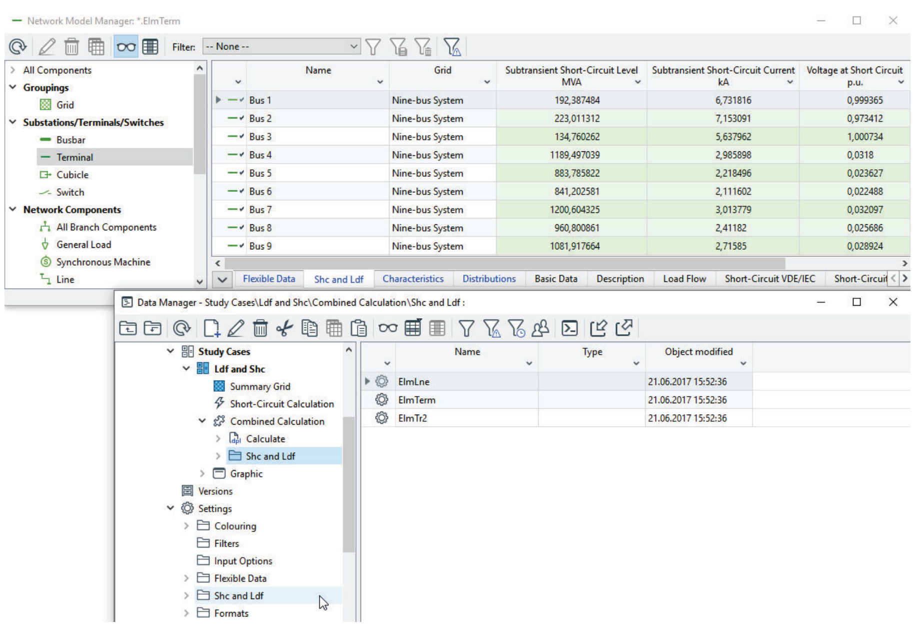


Figure 23.5.3: Definition of additional flexible data page and corresponding variable sets for the object classes.

For the user-defined variables, supported data types include integer, double, string, object (reference), arrays and matrices; for edge elements, variables can be defined as per phase and/or per connection quantities.

Within the script itself, important features are the command *CreateModule();*, which is followed by all the required calculations and data manipulation, new DPL and Python methods used for the handling of the variables, and the *FinaliseModule();* command, which is used once the calculations and data manipulation are complete and which defines the point at which no more changes are made to the user-defined variables and the results are ready to be viewed.

23.5.3 Executing an Add-on Module command

Add-on Modules can be executed directly or from an icon on the Additional Tools toolbar as shown in figure 23.5.4 below. Clicking on this icon will bring up a list of all Add On commands stored in the active study case or in the Add On folder in the Configuration area (see section 23.5.4).

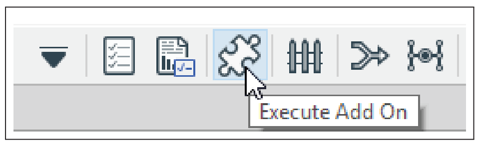


Figure 23.5.4: Running Add On modules from the Additional Tools toolbar

When the script has been executed, it is possible to access the user-defined variables in a flexible data page as shown in figure 23.5.5 below, or add them to a result box.

If an additional flexible data page is defined in the Add On, the corresponding flexible data page can be selected and further adapted.

In figure 23.5.5, it can be seen how the Add On Module name appears on the left-hand side together with the standard *PowerFactory* calculation functions.

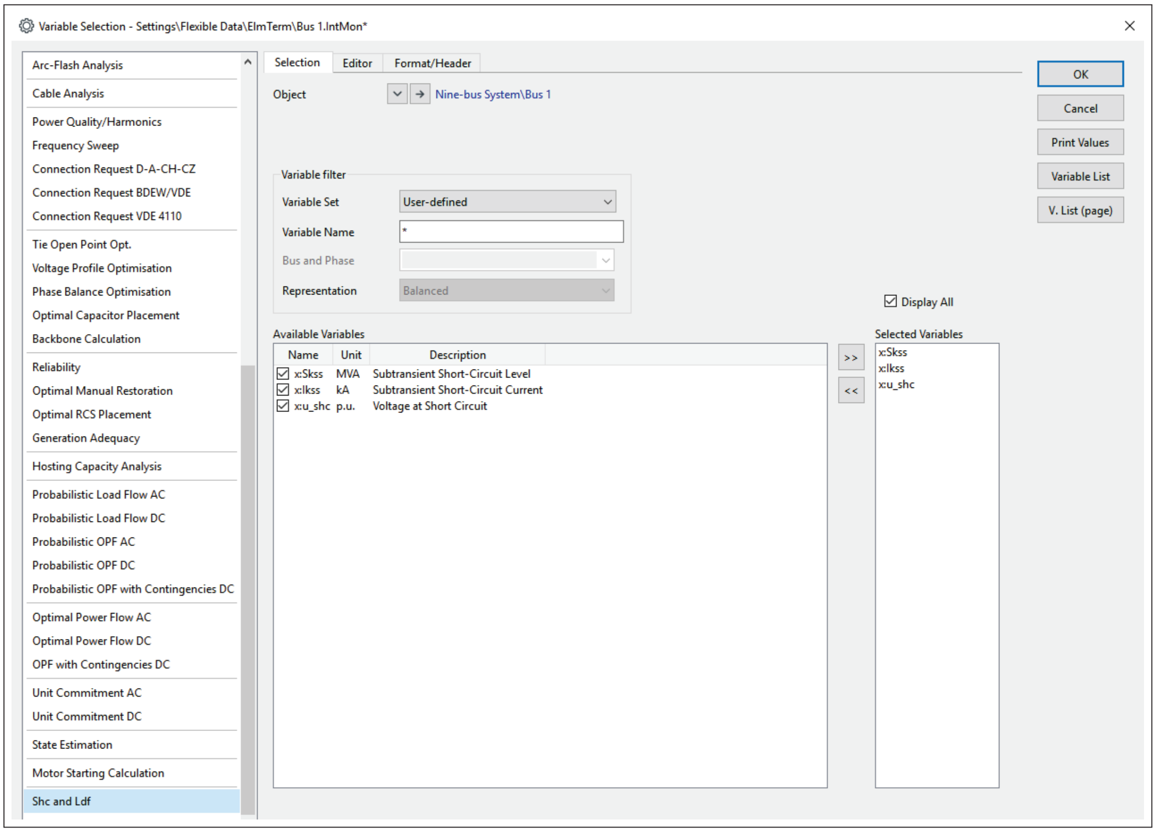


Figure 23.5.5: Selecting user-defined variables for a flexible data page

23.5.4 Adding Add On Modules to the User-Defined Tools toolbar

Add On Modules can be made available for use in different projects and (in the case of a multi-user database) by other users, by adding the Add-On commands to the User-defined Tools toolbar. Section 6.7.1 describes how the User-defined Tools toolbar is configured; the Add On Module command, including contents, is placed in the Add On folder of the Configuration folder.

Chapter 24

Interfaces

24.1 Introduction

PowerFactory supports a wide set of interfaces. Depending on the specific data exchange task the user may select the appropriate interface.

The interfaces are divided as follows:

- Interfaces for the exchange of data according to *DIgSILENT* specific formats:
  - DGS
  - StationWare (*DIgSILENT* GmbH trademark)
- Interfaces for the exchange of data using proprietary formats:
  - PSS/E (Siemens/PTI trademark)
  - NEPLAN (NEPLAN AG trademark)
  - ELEKTRA[1]
  - INTEGRAL
  - PSS/SINCAL (Siemens/PTI trademark)
  - Functional Mock-up Interface (Modelica Association trademark)
- Interfaces for the exchange of data according to standardised formats:
  - UCTE-DEF
  - CIM
  - OPC
- Programming interfaces for integration with external applications
  - C++ API

The above mentioned interfaces are explained in the following sections.

24.2 DGS Interface

DGS (**DIgS**ILENT) is *PowerFactory*'s standard bi-directional interface specifically designed for bulk data exchange with other applications such as GIS and SCADA, and, for example, for exporting calculation results to produce Crystal Reports, or to interchange data with any other software package.

[1] Available on demand

Figure 24.2.1 illustrates the integration of a GIS (Graphical Information System) or SCADA (Supervisory Control And Data Acquisition) with *PowerFactory* via the DGS interface

Here, *PowerFactory* can be configured either in GUI-less or normal mode. When used in GUI-less mode (engine mode), *PowerFactory* imports via DGS the topological and library data (types), as well as operational information. Once a calculation has been carried out (for example a load flow or short circuit), the results are exported back so they are displayed in the original application; which in this example relates to the SCADA or GIS application. The difference with *PowerFactory* running in normal mode (see right section of Figure 24.2.1) is that, besides the importing of data mentioned previously, the graphical information (single line graphics) is additionally imported, meaning therefore that the results can be displayed directly in *PowerFactory*. In this case, the exporting back of the results to the original application would be optional.

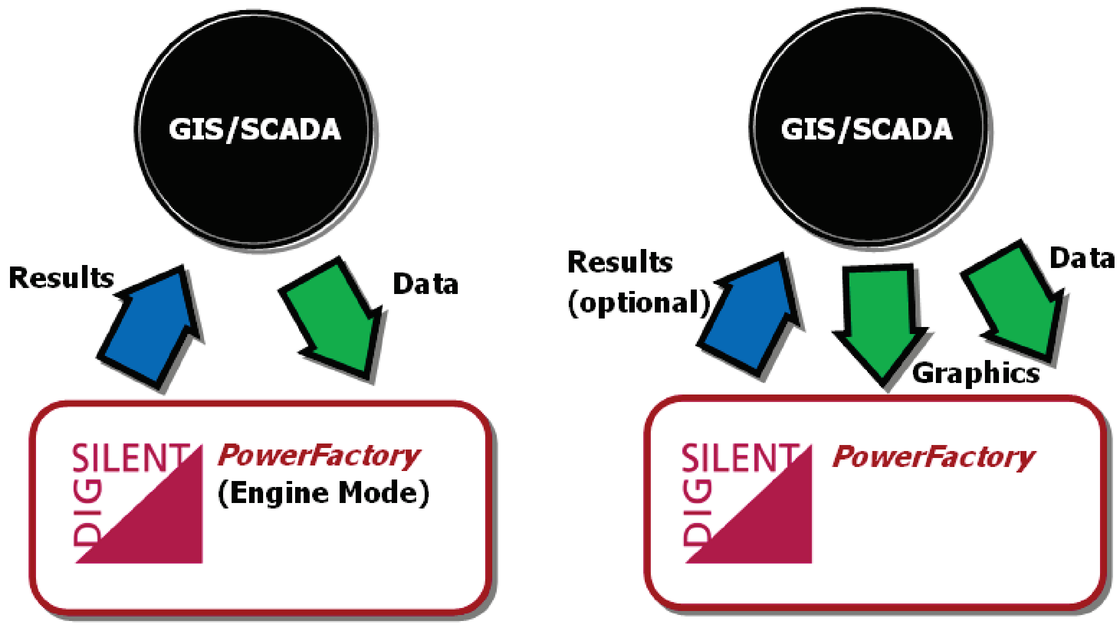


Figure 24.2.1: DGS - GIS/SCADA Integration

Although the complete set of data can be imported in *PowerFactory* every time a modification has been made in the original application, this procedure would be impractical. The typical approach in such situations would be to import the complete set of data only once and afterwards have incremental updates.

24.2.1 DGS Interface Typical Applications

Typical applications of the DGS Interface are the following:

- **Importing to *PowerFactory***
  - Data Import/Update into *PowerFactory* from external data sources such as GIS (Network Equipment), SCADA (Operational Data) and billing/metering systems (Load Data) in order to perform calculations.
- **Exporting from *PowerFactory***
  - Performing calculations in *PowerFactory* and exporting back the results to the original application.
- **Integration**

- Importing data sets to *PowerFactory* from GIS or SCADA, performing calculations, and exporting back results to GIS or SCADA.

24.2.2 DGS Structure (Database Schemas and File Formats)

PowerFactory's DGS interface is strictly based on the *PowerFactory* data model. Data can be imported and exported with DGS using different file formats and database schemas.

The following databases or file formats are supported:

- **Databases**
  - Oracle DB Server (ODBC client 10 or newer)
  - Microsoft SQL Server (ODBC driver 2000 or newer)
  - System DSN (ODBC)
  - Generic ODBC
- **File Formats**
  - DGS File - ASCII
  - XML File
  - Microsoft Excel File (2003 or newer)
  - Microsoft Access File (2003 or newer)

Important to note here is that the content of the files is the same, the only difference being the format.

Note: Due to changes in the format, DGS is available in several versions. It is highly recommended to always use the latest available DGS version.

The core principle of DGS is to organise all data in tables. Each table has a unique name (within the DGS file or database/table space) and consists of one or more table columns, where generally all names are case-sensitive.

More information on DGS and examples can be accessed by selecting from the main menu *Help* → *Additional Packages*→ *DGS Data Exchange Format*

24.2.3 DGS Import

To import data via the DGS interface, the general procedure is as follows:

- From the main menu go to *File* → *Import...*→ *DGS Format...* which opens the DGS-Import dialog.
- Specify the required options in both the *General* and *Options* pages, and click on the **Execute** button.

When importing DGS files, the user has two options:

1. Importing into a new project. With this option selected a newly generated project is left activated upon completion.
2. Importing into an existing project. If an operational scenario and/or a variation is active at the moment the import takes place, the imported data set will be divided correspondingly. For example importing breaker status (opened/closed) while an operational scenario is active will store this information in the operational scenario.

The following sections describe each of these options.

24.2.3.1 General Page

Import into New Project By choosing this option, a project will be created where all the DGS data will be stored. The user will have the option of specifying a specific name and location (other than the default).

Import into Existing Project By choosing this option, the DGS data will be imported into an already existing project. Here, the data can be selective and its not required that the imported data must be complete. In some cases, most of the objects already exist and only an update is required for some of them.

Import from The source of the data to be imported is specified with this option. If a *File Format* source is selected then the location and type of data (DGS, XML, MDB or XLS) must be specified. If a *Database* source is selected, then a service, User and Password information is required (the SQL server option will require an extra Database information).

24.2.3.2 Options Page

The visible options depend on the DGS version being used, and on the users choice of the *Import Format*.

Options for DGS version 4.x

Predefined Library
A predefined library located somewhere else in the database can be selected. The option of copying the library into the project is available.

Create Switch inside Cubicle
In cases where the source data has no switches defined inside the cubicles, the enabling of this option will create the switches automatically during the import. If switches already exist in a certain cubicle, the creation of switches in that particular cubicle is ignored.

Replace non-printable characters
If the source data contains not allowed characters ($\sim$, ?, etc.), they are replaced by an underscore character.

Options for DGS version 5.x

Open single line diagram(s)
If the DGS source contains graphics objects for single line diagrams, these will be opened automatically after import.

Dataset Import (only available if a* Database Schema *is selected as* Import Format*)
For DGS version 5 or higher, a labelled version of the data in the source data base can be selected for import. The labelled versions are mainly used to chose a time-dependent state of the data. The label name is input in the field *Label*.

Options for DGS version 6.x

The options for DGS version 5.x are available for DGS version 6.x as well. In addition, the options described below can be configured.

Global type library
The DGS import in earlier versions only allows for references to existing objects within the active project. With the DGS version 6.x, a global type library can be selected. Elements in the DGS source can refer to the *foreign key* of types in the selected global library.

Partial Import (only available if a Database Schema is selected as Import Format)
For DGS version 6 or higher, labelled regions can be selected for import in the source data base.

The labelled regions can be used to select specific voltage levels or sub-grids from the bulk data set. The label names are input in the field *Labels* separated by commas.

The option pages for all DGS versions contain the field

Additional Parameters
This field is specified for internal use only. No extra information is required by the user.

More detailed information on DGS and examples can be accessed by selecting from the main menu *Help → Additional Packages→ DGS Data Exchange Format*

24.2.4 DGS Export

In contrast to the *DGS Import*, where it is not relevant if a project is active or not; the *DGS Export* is based on what information is active at the moment the export takes place. In other words, only the active project, with the corresponding active *Study Case*, active *Scenario*, and active *Variations* are exported (objects are exported in their current state). Furthermore, the export can be fully configured, meaning that the user has the option of selecting the amount of information to be exported per class object. In general, the following data can be exported:

- Element data
- Type data
- Graphic data
- Result data (e.g. load flow results)

To export data via the DGS interface, the general procedure is as follows:

- Create an Export Definition
- Activate the project to be exported, considering which *Study Case, Scenario* and *Variations* should be active.
- From the main menu go to *File → Export. . . → DGS Format. . .* which opens the DGS-Export dialog.
- Specify the required options in both the *General* and *Options* pages, and click the **Execute** button.

The following sections describe each of these options.

24.2.4.1 General Page

DGS Version
Version of the DGS structure.

Format
Output format. Either as ASCII, XML, MS Excel or MS Access file (for Excel or Access, Microsoft Office must be installed on the computer) or as Oracle, MS SQL Server, ODBC DSN or generic ODBC databases (respective data base drivers must be installed.).

File Name or Data Base Service
Depending on the *Output Format*, a file name for the output file or the data base service and user access information are required.

Insert Description of Variables
If checked, a description of the column headers is included in the output file (only available for ASCII, XML and MS Excel).

Variable Sets
Select the variable set definition for export. The data exported will be according to the variable definition specified (see the explanation at the beginning of the section). It is required to select a folder that contains the monitor variable objects (*IntMon*) related to each class that is to be exported.

24.2.4.2 Options Page

The visible options mostly depend on the DGS version 4.x, 5.x or 6.x and on the users choice of the *Export Format*.

DPL Script
Independent of the DGS version, the user can select a DPL script. This DPL script is automatically executed before DGS export.

Options for DGS version 5.x

Allow hierarchy and references of exported objects to be incomplete
If this option is set, error messages because of references to external objects (e.g. types in the global library) that will not be exported are omitted. Furthermore, the export of data classes is possible although their parent-folder class is not contained in the *variable definition set*.

Allow user-defined table names
With this option, a prefix and suffix can be added to all table names on DGS export. The prefix or suffix is defined in the field *Additional Parameters*.

Export as dataset (available for export to data base formats only)
This option is used to write the exported data to a specifically labelled data set (session state) into the data base. A label identifier must be given. Optional, a description can be added.

Options for DGS version 6.x

All options for DGS version 5.x are available for DGS version 6.x. In addition, the following options exist:

Export as Update
DGS 6.0 supports an explicit marker *OP* to mark a data record to be created (C), updated (U) or deleted (D) on DGS import. On export, the user can chose to mark all data as *Update* by means of this option. Otherwise the exported data are marked for *Create*.

Categorisation of data for partial import (available for export to data base formats only)
This option is used to define for each grid (ElmNet) in the active project a *data part* in the data base. In addition, certain data types and elements are labelled with respect to their meaning in the context of this partial export: global types, local types, boundary set (branches connecting elements that belong to different grids), other elements (e.g. characteristics). The names of these labels are input in the respective fields (all fields should be filled in). In addition, a global library folder can be selected to include referenced types from this library on DGS export.

Options for DGS version 4.x

Export Grid Name
If this option is set, a column “Grid” is added to all tables containing network elements.

Export Cubicles
With this option, cubicles are exported. Cubicles describe the connectivity of nodes and branches. The export of cubicles can be omitted if the grid topology is not needed (e.g. for result export).

Export Graphical Data
The user can select one of three options for the export of graphical data:

No No graphical data are exported.

Yes, with Graphic (*IntGrfnet*) Names Graphical data are exported. All graphic object tables contain a column for the name of the graphic scheme (*IntGrfnet*).

Yes, without Graphic (*IntGrfnet*) Names Graphical data are exported. The graphic scheme is not referenced by the exported graphic objects.

The option page contains independent of the DGS version always the field

Additional Parameters
This field is specified for internal use only. No extra information is required by the user.

More detailed information on Variable Sets definitions (*IntMon*) can be accessed by selecting from the main menu *Help* → *Additional Packages*→ *DGS Data Exchange Format*.

24.3 ANAREDE and ANAFAS Interface

ANAREDE is a network calculation software product from Electrobras Cepel (Brazil), for the analysis of steady state phenomena in electrical power systems. ANAREDE *.pwf files can be imported and exported.

Similarly, importing and exporting of ANAFAS *.ana files is now possible, where ANAFAS is Electrobras Cepel's network calculation software for short-circuit analysis in electrical power systems.

24.4 PSS/E File Interface

Although both import and export functions for PSS/E files are integrated commands of *PowerFactory*, the export function is licensed separately. For more information on prices and licensing contact the sales department at mail@digsilent.de.

PSS/E Import supports versions 23 to 34 and can be carried out by going to the main menu and selecting *File* → *Import...* → *PSS/E*.

In the same manner, and provided the appropriate licensing exists, a project can be exported in PSS/E format by selecting from the main menu *File* → *Export...* → *PSS/E*. For exports, PSS/E is supported up to and including version 33.

24.4.1 PSS/E File Types and Versions

PowerFactory is able to convert the following PSS/E 'Source Format' file types:

RAW Power Flow Data (Import and Export) *.raw extension.

SEQ Sequence Data (Import and Export) *.seq extension.

DYR Dynamics Data (Import and Export) *.dyr extension.

PowerFactory is not able to convert the following PSS/E 'Binary Format' file types:

SAV Power Flow Data "Saved Case": needs to be converted to RAW format in PSS/E.

SNP Dynamics Data "Snapshot": needs to be converted to DYR format in PSS/E.

SLD Single Line Data "Slider Data": not supported by *PowerFactory*.

DRW Single Line Data: not supported by *PowerFactory*. Instead, diagrams can be created in *PowerFactory* using the diagram layout tool.

PSS/E files from versions v35.x cannot be converted/imported to *PowerFactory* at present. They first have to be converted to the PSS/E version v34 in PSS/E.

24.4.2 Importing PSS/E Steady-State Data

PowerFactory is able to convert both steady-state data (for load-flow and short-circuit analysis) and dynamic data files. It is good practise to first import the steady-state data (described in this section), then to add the dynamic models (described in Section 24.4.3: Import of PSS/E file (Dynamic Data).

Before starting the next steps for importing a PSS/E file, make sure that no project is active. Once this has been confirmed, select from the main menu *File* → *Import...* → *PSS/E*. By doing so, the *Convert PSS/E File*s command dialog will be displayed, asking the user to specify various options.

24.4.2.1 General Page

Nominal Frequency Nominal frequency of the file to be Converted/Imported.

PSS/E File Type

PSS/E Raw data Location on the hard disk of the PSS/E raw data file. By default the program searches for *.raw extensions.

Sequence Data Location of the PSS/E sequence data file. By default the program searches for *.seq extensions.

Note: From *PowerFactory* 2020 onwards the drw file is no longer converted/imported. Instead, network graphics can be generated after import into *PowerFactory*, using the *Diagram Layout Tool*.

Save converted data in

Project The project name that will be assigned to the converted/imported file in *PowerFactory*.

in Location in the Data Manager tree where the imported file will be stored.

The following topics:

- ***Dyn. Models Data***,
- ***Composite Frame Path***,
- ***DSL - Model Path***,
- ***Parameter Mapping***

are not used for the import of steady-state data and will be explained in the dynamic import Section 24.4.3.

24.4.2.2 Options Page

- **Convert only sequence data file** - With this option enabled, the converter will only add the sequence data to an existing project.
- **Convert only dynamic models file** - With this option enabled, the converter will only add the dynamic data file to an existing project (only for dynamic data import).
- **Only convert file (no DB action)** - Internal option used for syntax check and error messages during conversion. Normally this box should be left unchecked.

- **Output only used dynamic models** - Displays a list of used dynamic models (only for dynamic data import).
- **Unit of ’LEN’ for lines in miles instead of km** - With this option enabled, all lengths will be interpreted in miles in the PSS/E raw files.
- **Consider transformer phase shift** - With this option enabled, transformer phase shifts will be considered. This option is recommended and activated by default.
- **Convert Induction Machines (Generators: P<0)** - With this option enabled, all generators in the raw data file that have negative active power will be converted to asynchronous machines. For transmission grids the option should be disabled for proper modelling of phase shift generators.
- **Automatic 3-W. Transformer detection/conversion** - In versions <27, PSS/E does not handle 3-winding transformers as a dedicated model. In such cases, the 3-winding transformer is modelled with three 2-winding transformers connected to a busbar. If this option is selected, the converter will try to detect the existence of three 2-Winding Transformers connected to a busbar. If any candidates are available, *PowerFactory* will replace them by a 3-Winding Transformer. The detection algorithm uses the impedances and the voltage control of the transformers as reference. From version 27 onwards PSS/E supports the 3W-transformer model, so that *PowerFactory* does not start an automatic detection of 3W-Trf modelled as 2W-Trfs.
- **Convert capacitive line shunts to line susceptance B’** - If a line has line shunts the converter adds automatically the line shunt capacitance to the C1’ (B1’) in the *PowerFactory* line type.
- **Convert Common Impedance as Transformer** - If this option is selected, the Common Impedance in PSS/E may be converted to a *PowerFactory* common impedance or to a transformer.
- **Convert Series Capacitance as Common Impedance** - Older versions of PSS/E do not handle series capacitances as a dedicated model. These elements therefore are represented by lines with negative reactances. During the conversion, *PowerFactory* detects these branches and converts them to series capacitances (by default) or to common impedances (when this option is active).
- **Convert off-nominal turn ratio to transformer tap** - Transformer ratios different from the rated ratio are automatically converted to a transformer type using taps, including the correct tap position.
- **Busbar naming: ’PSSE_NAME’** - With this option enabled, the busbars are named similar to the PSS/E raw data file (without bus number).
- **Branch naming: ’BUSNAME1_BUSNAME2_ID’** - With this option enabled, the branches are named as the name of the busbars + ID.

Additional Parameters - This field is specified for internal use only. No extra information is required by the user.

24.4.3 Import of PSS/E file (Dynamic Data)

As explained in Section 24.4.2 it is good practice first to import the steady-state data and then to add the dynamic model data.

Some dynamic models used in PSS/E are available in the global *DIgSILENT* Library. User defined dynamic models should be modelled in *PowerFactory* before importing the program. In this case, an important condition for successful file conversion is that all DSL models used during the conversion process should be stored in the same model library folder.

If the original library should use specific folders for the different types of controllers (AVR,PCO,PSS, etc.), the user should copy all of the models into the same library folder, in this case the recommendation is to copy the dynamic models from the *DIgSILENT* library into the library where the rest of the user defined models are located. After the conversion, the user may re-arrange the models.

The procedure to start the import of dynamic network data is very similar to the import of steady-state data. Some parameter adjustments have to be made.

24.4.3.1 General Page - Dynamic Models

On the *General* page of the import dialog the following topics have to be specified:

Dyn. Models Data - Location of the PSS/E Dynamic Models data file. By default the program searches for *.dyn and *dyr extensions.

Use Standard Models from global library - If this option in enabled, *PowerFactory* will automatically point to the *PSS/E compatible* library located in the *DIgSILENT* library. There will be no need of selecting the composite Frame Path and DSL Model Path.

Composite Frame Path - Location in the *PowerFactory* data base where the composite frames are stored (Standard Models/Composite Models Frames. . .).

DSL - Model Path - Location in the *PowerFactory* data base where the DSL models are stored (Standard Models. . .).

Parameter Mapping - Location of the *PowerFactory* mapping file. This is an option that normally will not have to be defined by the user. By default *PowerFactory* will automatically set up its own internal mapping file. This file defines how to translate the PSS/E internal models into *PowerFactory* models, including the mapping of controller parameters. For automated conversion of user-defined PSS/E controllers the mapping file may be customised.

24.4.3.2 Import Options Page - Dynamic Model Import

On the *Options* page of the import dialog the following options should be considered:

Convert only dynamic models file - With this option enabled, the converter will only add the dynamic data file to an existing project.

Output only used dynamic models - Displays a list of used dynamic models.

24.4.4 Exporting a project to a PSS/E file

This function allows the export of the network model in PSS/E format. The export comprises both steady-state and dynamic data sets. The correct conversion of dynamic models is only possible for the standard IEEE models. Models which the user implemented in *PowerFactory*'s DSL can not be automatically translated and must be modelled as user-defined controller types separately in PSS/E.

To export a project in PSS/E format select *File* → *Export. . .* → *PSS/E* from the main menu.

24.4.4.1 Export General Page

RAW Conversion File - Path and file name for the PSS/E RAW file, containing the symmetrical description of the model. By default the program selects the *.raw extension.

SEQ Conversion File - Path and file name for the PSS/E SEQ file, containing the additional description of the model necessary for unbalanced conditions. By default the program selects the *.seq extension.

DYN Conversion File - Path and file name for the PSS/E DYN/DYR file, containing the dynamic models of the project. By default the program selects the *.dyr extension.

PSS/E Version - Version of the exported PSS/E file (25 to 33).

24.4.4.2 Export Options Page

Convert Motors to Generators if P<0 - With this option enabled, all asynchronous machines in generator mode will be converted to synchronous machines.

Export branch as single equivalent line - Selecting this option will convert the branch models to an equivalent line.

Convert SVS to generator - This option defines how the SVS elements will be exported. Three options are available:

- ***No:*** the SVS elements won't be exported.
- ***Only voltage controlled:*** will convert the SVS elements with control mode set to *Voltage Control* (Load Flow page of the element) to generator models.
- ***Always:*** all the SVS elements will be converted to generator models.

Base Apparent Power - Base for the power values given in per-unit system.

Min (Zero) Impedance Branch - Minimum impedance for ideal connections.

PSS/E Bus Number - This option defines the naming convention when exporting terminals *ElmTerm*. Three options are available:

- ***Automatic:*** the number assigned will be according to the name (in ascending/alphabetical order).
- ***Use Serial Number:*** the serial number information stated in the *Description* page of each terminal will be used for assigning the PSS/E bus number.
- ***Use Characteristic Name:*** the characteristic name information stated in the *Description* page of each terminal will be used for assigning the PSS/E bus number.

Export PSS/E-Area index as - The way the Area index is defined in PSS/E is defined here, two options are available:

- ***Grids:*** the exported file will have the areas defined according to the Grids defined in the *PowerFactory* model.
- ***Areas:*** the exported file will have the areas defined according to the Areas defined in the *PowerFactory* model.

Additional Parameters - This field is specified for internal use only. No extra information is required by the user.

24.5 PSS/U Interface

PowerFactory offers the user the option to import PSS/U files. The following files are supported:

- *.dat

In addition, data dictionaries can also be imported.

24.5.1 Importing PSS/U Data

To import data via the PSS/U interface, the general procedure is as follows:

- From the main menu go to *File* → *Import...* → *PSS/U...* which opens the *Convert PSS/U files* dialog.
- Click on the dots (...) to select the *.dat-file and specify the required options and click on the **Execute** button.

Once the import process has been executed, the project (new or existing) is left activated upon completion.

The following section describes each of the dialog options.

24.5.1.1 General Settings

Nominal Frequency Nominal frequency of the file to be converted/imported.

File Type Choose the *.dat file from your system that should be converted.

Save Converted Data in *PowerFactory* offers the possibility to import the project into an existing project or to create new project for the import. If no project is selected *PowerFactory* will create a new project.

Common Conversion Settings Following settings are available:

- Line Length in miles instead of km: The unit of the line length of the input file.
- Import Graphic Information: A graphic will automatically be created based on the information of the input file.
- Create switch within branches: If this option is selected switches are created at the beginning and the end of every branch. It is recommended to choose this option, so that lines can be switched off later on.

Construction Data Dictionary Choose the *.con file with the corresponding construction data dictionary. If no dictionary is selected the types are created automatically based on the *.dat file.

Machine Data Dictionary Choose the *.mot file with the corresponding machine data dictionary. If no dictionary is selected the types are created automatically based on the *.dat file.

Additional Parameters Additional parameters for the import. This field is normally left blank.

24.5.1.2 Options

On the options page additional parameters regarding the graphic are specified.

Automatic Drawing of Network Elements Defines which elements should be drawn automatically.

Offset Factor for semi-orthogonal branches Sets the offset factor for semi-orthogonal line routing in *PowerFactory*. This factor is applied if the nodes in PSS/U were plotted as bars rather than points.

Scaling Factor The scaling factor defines the graphical distance between the nodes.

24.6 PSS/ADEPT Import Converter

PSS/ADEPT is a former network calculation software product provided by Siemens, and for a number of years the import of older PSS/ADEPT formats (PSS/U) into *PowerFactory* has been possible. PSS/ADEPT HUB and Construction Dictionary files can be imported.

HUB files (*.dmp) are ASCII files that contain all the significant data of the network, including the graphics. Construction Dictionary files (*.con) contain line and cable type data.

Construction Dictionary files are not mandatory for the import.

The import dialog can be started in two ways:

- File - Import - PSS/ADEPT...
- Data Manager - Input Window - ed/adept

24.7 ELEKTRA Interface

PowerFactory offers the user the possibility to import different types of ELEKTRA files.

Note: The ELEKTRA Interface is only available on demand.

The files supported for import are as follows:

- **Elektra network models**
  - Element data (*.esd) from Elektra Version 3.60 to 3.98, which contain the topological and electrical data of the elements in the grid.
  - Network diagrams (*.enp) from Elektra Version 3.92 to 3.98, which contain the graphical representation of grids.
- **Elektra equipment type library**
  - Type data (*.dat), which contains equipment types.

24.7.1 Import of Elektra Data

The general way to import data via the Elektra interface is as follows:

- From the main menu, select: *File* → *Import*→ *Elektra. . . .* The Elektra-Import dialog will be displayed.
- Select the desired options and click on the **Execute** button.

Note: The Elektra import cannot be executed if Elektra is open. Close the software before executing the import.

The import will be executed regardless of whether a project is activated or not. At the end of the import, the project will be activated. If there is another project activated while importing the Elektra data, *PowerFactory* will deactivate the active project, and activate the newly-created or selected project (according to the settings).

The options available in the Elektra import dialog are described in the following section.

24.7.2 General Settings

Import into

New project A new project will be created in which all of the Elektra data will be stored. The user can select a name and a storage location. Different versions of the same network model should be stored in new projects.

Existing project Elektra data will be imported into an existing project. Use this option if grids from different regions will be connected and should be calculated together in one project.

Files

Kind of data Within the Elektra import, *Element/graphic data* (data type *.esd and *. enp) or *Type data* (data type *.dat) can be imported, according to the selection.

Element data If *Element/graphic data* is selected, set the storage location of the Elektra element data by clicking the "..." icon.

Graphical data Add graphical data for the element. Select *Delete* to remove the data from the list.

Type data If *Kind of Data*: *Type data* is selected, click on *Add* to select the Elektra type library (*.dat) for import. Repeat this step if more type libraries should be added to the import. Select *Delete* to delete single files from the selection.

24.7.3 Advanced Settings

On the *Advanced* settings page, the following options can be used to simplify the imported network. In addition, there are two options to activate the import of coupling impedances and active/reactive power characteristics (Q(P) curves).

General Options

consider graphical node representation If a node is set to *Internal Node* in the Elektra element data, *PowerFactory* will also set the node to *Internal Node*. That is, the *usage* of the node in *PowerFactory* is set according to the *usage* in Elektra element data.

create detailed busbar systems for single busbars By default, a detailed representation of substations is generated for all Elektra busbars in a *PowerFactory* substation. This is done regardless of whether it is a single or double busbar. This option should be chosen to set locations where only single busbars exist, to single busbars in *PowerFactory*.

create auxiliary graphic objects in annotation layer Objects in the Elektra open graphic (open texts, memos, rectangles, pictures, ...) will be transformed into the *annotation layer* of *PowerFactory* by default. These layers can be scaled and changed in *PowerFactory*. As an alternative, graphical objects can be split into parts in the import process. This leads to limited options in later adaptations of the objects.

create element names with reference to the node name In *PowerFactory*, every element must have a unique name. To ensure this uniqueness for the Elektra import, the names are comprised of the following parts: *Elektra element name - Elektra name of terminal 1 - Elektra name of other terminal*. If this name has more than 40 letters it will be shortened.

coupling impedances Coupling impedances between adjacent overhead lines in Elektra network data are converted into corresponding tower elements (*ElmTow*) and tower types *TypTow* in *PowerFactory*.

convert Q(P) curves The reactive power behaviour of generator units or synchronous machines in Elektra data can be given as an active/reactive power characteristic. These curves are converted into a Q(P) characteristic in *PowerFactory* and assigned to the corresponding static generator/s or synchronous machine/s.

Individual scaling factors at Elektra node elements

Active and reactive power can be modified through scaling factors in Elektra on different layers. These factors are transformed into scalar *PowerFactory* characteristics, upon import of Elektra element data. If there are many individual scaling factors for Elektra node elements, one of the following options can be chosen. These options may assist in reducing the number of characteristics in *PowerFactory*.

Ignore all scaling factors The factors for active and reactive power for Elektra node elements are ignored within the data import. The results of the load flow calculation are influenced by this option.

Calculate resulting power quantities The multiplication of the active and reactive power by the Elektra node element factor is transferred into *PowerFactory*.

Create individual scale factor objects For all factors for Elektra node elements that are set to a value different to '1', corresponding scalar characteristics are created in *PowerFactory*. This is the default option.

Additional Parameter This field is for internal use. No additional information is required from the user.

24.7.4 Importing Elektra Network Data

To import Elektra network data, choose *Kind of data*: *Element/graphic data*. The following combinations of element and graphic data exist:

1. Selection of Elektra element data (*.esd) without graphic data
   The element and topological data from the *.esd file will be imported. Type data for the element data will be created. There is no creation of a network diagram.

2. Selection of Elektra element data (*.esd) and one or more corresponding graphic files (*.enp)
   The included topological and type data from the *.esd file will be imported. Type data for the element data will be created. Additionally, a network diagram for every selected Elektra graphical data will be created and elements are linked to the graphical objects (if present in both files).

3. Selection of Elektra graphical data (*.enp), without element data.
   If only graphical data has been selected, for each graphic file one network diagram will be created. From the topological information in the *.enp file, network data will be created. This network data does not contain technical parameters or type references.

24.7.5 Importing Elektra Type Data

To import Elektra type data, select one or more *.dat files.
In the folder *Library/Equipment Type Library* from the import project, a new Equipment Library will be created for each file and relevant kind of element.
If the successfully imported type data should be used in *PowerFactory*ś global library, continue as follows:

1. Change the user to Administrator by selecting *Tools* → *Switch User...* → *Administrator* via the main menu.

2. Open the *PowerFactory* Data Manager, and create a new folder of type *Library* within the directory Database.

3. Copy the Equipment Library from the import project into this folder.

24.7.6 Output Window

During the import the following information is provided in the output window:

- Network elements which do not coexist in the Elektra element and in the Elektra graphical data (multiple entries while importing multiple graphical files are possible).

- Network elements which are generated from power ratings in Elektra nodes.

- Coupling objects between different locations, which cannot be converted.

- Graphical objects whose names are adapted during import.
- Inconsistent or incomplete element parameters.

24.8 NEPLAN Interface

PowerFactory offers to the user the option of importing different types of NEPLAN files. The files supported for importing are the following:

- **NEPLAN 4**
  - Project File Data (∗.mcb) containing the topological, electrical and graphical data.
  - Line Data Type (∗.ldb) containing the line type information.
- **NEPLAN 5**
  - Node Table (∗.ndt) containing the node data, such as rated voltages and loads.
  - Element table (∗.edt) containing the branch data, such as lines and transformers.
  - GIS/NMS Interface (∗.cde) containing the graphical information of all the networks which are part of the NEPLAN project.

24.8.1 Importing NEPLAN Data

To import data via the NEPLAN interface, the general procedure is as follows:

- From the main menu go to *File* → *Import...* → *Neplan...* which opens the NEPLAN-Import dialog.
- Specify the required options and click on the **Execute** button.

The NEPLAN data import always creates a new *PowerFactory* project. Once the import process has been executed, the newly generated project is left activated upon completion.

Independent of the NEPLAN file version (4 or 5), the user has the option of importing the data with or without graphical information. That is, if the user selects importing the data without graphical information, only the topological and electrical data will get imported, and no single line graphic will be generated. In order to import NEPLAN 5 graphics, the path to the NEPLAN files should not contain spaces.

Importing NEPLAN 4 Files

When importing NEPLAN 4 files, the user has basically two options:

1. Selection of a *.mcb file.
   If the user selects this type of file and if a corresponding *.ldb file is present (should be in the same directory where the *.mcb is stored), then the information of both files gets imported. If only the *.mcb file exists, then only the information regarding this file is imported (which can also contain line data).
2. Selection of a *.ldb.
   If the user selects this type of file only the information regarding this file (line data) is imported.

Importing NEPLAN 5 Files

When importing NEPLAN 5 files, the user is only required to select the *.ndt. By doing so, the corresponding *.edt file is automatically imported also. This basically means that a *.edt file must be present otherwise the import will not be executed. The *.cde file is however optional. Additionally, all three files must have the same name and must be in the same directory! As a recommendation, create a separate folder and place all the files there.

The following section describes each of the NEPLAN import dialog options.

24.8.1.1 General Settings

File Type

Neplan Data Location on the hard disk of the NEPLAN data file. Three types of files are available: *.mcb, *.ldb and *.ndt.

Save converted data in

Project The project name that will be assigned to the converted/imported file in *PowerFactory*.

in Location in the Data Manager tree where the imported file will be stored.

Common Conversion Settings

Automatic busbar system detection

Import Graphic Information If this option is enabled then the graphical information is imported and the single line diagram is generated. In case of NEPLAN 5 import the *.cde file is required.

Graphic Import Options (only for NEPLAN 5 import)

Additional Rotation Angle for 1-port Elements (deg) If a value different than 0 is stated, then the single port elements (loads, generators, motors, etc.) are rotated counter clockwise (degrees) with respect to the original position.

Automatically scale to A0 If this option is selected, then the graphic is rescaled according to the A0 page format.

User defined scaling factor

Additional Parameters
This field is specified for internal use only. No extra information is required by the user.

24.9 INTEGRAL Interface

PowerFactory offers the user the option to import Integral files for Load Flow and Short Circuit analysis. The following files are supported:

- *.dvg
- *.dtf
- *.xml

Furthermore Integral files can be export as *.xml files.

24.9.1 Importing Integral Data

To import Integral data, the procedure is as follows:

- From the main menu go to *File* → *Import...* → *Integral...* (this will open the Integral import dialog).

In the **'Save converted data in'** field the user can enter a project name, and the *PowerFactory* user for this project can be selected. The Integral data import always creates a new *PowerFactory* project.
The *.xml Integral files contain graphical information. However, for older Integral files with the ending *.dvg and *.dtf it is necessary to select graphical data with the ending *.bild.

More information about the Integral Import is available in the German version of the User Manual.

24.9.2 Export Integral Data

The Integral export converts the *PowerFactory* project into an *.xml file in Integral format. Therefore **'XML Data'** must be defined as the path where to store the xml file. If the ending .xml is not given, it will automatically added.

More information about the Integral Export is available in the German version of the User Manual.

24.10 PSS SINCAL Interface

PowerFactory offers the user the option to import MS Access database files from PSS SINCAL for Load Flow and Short Circuit analysis. The following files are supported:

- *.mdb

24.10.1 Importing PSS SINCAL Data

The procedure to import PSS SINCAL data is as follows:

- From the main menu go to *File* → *Import...* → *Sincal...* (this will open the PSS SINCAL import dialog).
- Select the file location of the MS Access database file of the SINCAL project (usually named *database.mdb*) in the field *Database name*.
- In the *Save converted data in* field, the user can enter a project name, and the *PowerFactory* user for this project can be selected.

The PSS SINCAL data import will always create a new *PowerFactory* project.

The SINCAL *.mdb database files contain graphical information. This information is converted into a *PowerFactory* network diagram.

Note: A SINCAL import error message appears when the *PowerFactory* installation type (32bit or 64bit) is different from the Microsoft Office installation type (32bit or 64bit). To open the SINCAL database (*.mdb) during the import to *PowerFactory*, MS Access is required.

During the installation of Office / Access 32bit, Microsoft only installs the 32bit ODBC driver by default. *PowerFactory* (64bit) needs the 64bit ODBC driver. The same goes for the other way around: Office / Access 64bit is installed by Microsoft by default only with the 64bit ODBC driver. *PowerFactory* (32bit) needs the 32bit ODBC driver.

The user has to install the missing driver from here:

https://www.microsoft.com/en-US/download/details.aspx?id=13255

We recommend “Microsoft Access Database Engine 2010 Redistributable” instead of the newer 2016 version, which requires more rights and a work-around for a successful installation.

24.11 UCTE-DEF Interface

In *PowerFactory*, both export and import of **UCTE-DEF** (**U**nion for the **C**o-ordination of **T**ransmission of **E**lectricity - **D**ata **E**xchange **F**ormat) is supported. The UCTE interface is currently intended for importing/exporting grid data of a country belonging to the former UCTE community.

The data contained in these files correspond basically to load flow and short circuit (3 phase) type data. Furthermore, it only considers specific UCTE voltage levels according to voltage level codes, as well as UCTE specific country codes, such as DK for Denmark, P for Portugal, etc.

Important to note here is that from 1^{st} of July 2009, **ENTSO-E** (European Network of Transmission System Operators for Electricity) took over all operational tasks of the 6 existing TSO associations in Europe, including the Union for the Coordination of Transmission of Electricity (UCTE).

For more information related to the UCTE format, refer to the ENTSOE website: https://www.entsoe.eu

24.11.1 Importing UCTE-DEF Data

To import data via the UCTE interface, the general procedure is as follows:

- From the main menu go to *File* → *Import...* → *UCTE...* which opens the UCTE-Import dialog.
- Specify the required options and click on the **Execute** button.

Once the import process has been executed, the project (new or existing) is left activated upon completion.

The following section describes each of the UCTE import dialog options.

24.11.1.1 General Settings

Import into

New Project By choosing this option, a project will be created where all the UCTE data will be stored. The user will have the option of specifying a specific name and location (other than the default).

Existing Project By choosing this option, the UCTE data will be imported into an already existing project.

File Type

Add UCTE Files Location on the hard disk of the UCTE files. Two types of files are available: *.uct and *.ucte.

Options

Import for DACF process With this setting the user has the option to import the Day Ahead Forecast.

Convert negative loads to generators With this option enabled, negative loads defined in the UCTE file will be converted to generators in the *PowerFactory* model.

Convert transformer equivalent to common impedance With this option enabled, transformer equivalents defined in the UCTE file will be converted to common impedances in the *PowerFactory* model.

Ignore reactive power limits for generators With this option enabled, the reactive power limits of the generators defined in the UCTE file will be ignored .

Additional Parameters This field is specified for internal use only. No extra information is required by the user.

24.11.2 Exporting UCTE-DEF Data

As in the other export interfaces, the *UCTE Export* is based on the **active** project at the moment the export takes place. To export data via the UCTE interface, the general procedure is as follows:

- Activate the project to be exported, considering the which *Study Case, Scenario* and *Variations* should be active.
- From the main menu go to *File* → *Export...* → *UCTE...* which opens the UCTE-Export dialog.
- Specify the required options, and click on the **Execute** button.

The following section describe each of these options.

24.11.2.1 General Settings

File Type

UCTE Data Location on the hard disk where the UCTE files will be stored. Two types of files are available: *.uct and *.ucte.

Grids Selection of which grids to export.

Options

Export UCTE voltage >= Only the elements having a voltage greater than the UCTE voltage specified are exported.

Export branch as single equivalent line By enabling this option the export will convert the *PowerFactory* branch definitions into single equivalent lines.

Use first character of characteristic name as branch order code If checked, the characteristic name (first character) is used in the branch order code of the exported UCTE file.

Additional Parameters This field is specified for internal use only. No extra information is required by the user.

24.12 CIM Interface

In *PowerFactory*, both export and import of **CIM** (**C**ommon **I**nformation **M**odel) is supported. The CIM interface is currently intended for importing/exporting the following profile:

- ENTSO-E 2009

(Options "ENTSO-E 2010" and "ENTSO-E 2009 Dynamic Models" in the drop-down menu relate to profiles which were never formally released. Therefore, although these options are available to the user, they are not supported.)

CIM is defined in IEC-61970, and its purpose is to allow the exchange of information related to the configuration and status of an electrical system.

For information relating to CGMES, please see section 24.13 below.

24.12.1 Importing CIM Data

To import data via the CIM interface, the general procedure is as follows:

- From the main menu go to *File* → *Import...* → *CIM...* which opens the CIM-Import dialog.
- Specify the required options and click on the **Execute** button.

Once the import process has been executed, the project (new or existing) is left activated upon completion.

The following section describes each of the CIM import dialog options.

24.12.2 General Page

Import into

New Project By choosing this option, a project will be created where all the CIM data will be stored. The user will have the option of specifying a specific name and location (other than the default).

Active Project By choosing this option, the CIM data will be imported into the active project.

Import from

Profile Currently the profile ENTSO-E 2009 is supported.

separated Files With this setting the user has the option to import the equipment, topology and solved state files separately.

CIM File Location on the hard disk of the CIM files. Two types of files are supported: *.zip and *.xml.

Additional Parameters This field is specified for internal use only. No extra information is required by the user.

24.12.3 Exporting CIM Data

As in the other export interfaces, the *CIM Export* is based on the **active** project at the moment the export takes place. To export data via the CIM interface, the general procedure is as follows:

- Activate the project to be exported, considering which *Study Case, Scenario* and *Variations* should be active.
- From the main menu go to *File* → *Export...* → *CIM...* which opens the CIM-Export dialog.
- Specify the required options, and click on the **Execute** button.

The following sections describe each of these options.

24.12.3.1 General Page

Export to

Profile Currently the profile ENTSO-E 2009 is supported.

separated Files With this setting the user has the option to export the equipment, topology, and solved state files separately.

CIM File Location on the hard disk where the CIM files will be stored. Two types of files are supported: *.zip and *.xml.

Export Selection

Grids Selection of which grids to export.

Border Nodes Grid Selection of the grid which contains the X-nodes.

24.13 CGMES Tools

The CGMES Tools provide an additional interface to CIM. These tools are accessible via the main menu in *PowerFactory* under *Tools* → *CGMES Tools*. This section describes these tools and uses the following naming conventions:

- **Model folder** stores all CIM data.
- **Archive** contains all corresponding CIM Models.
- **CIM Model** stores all data contained in a single instance file (mainly CIM Objects and namespaces).
- **CIM Object** stores all data contained in a single object.

The CGMES Tools separates each action (i.e. the import and export of CIM data) into two steps. To import data from a CIM file (XML format) into *PowerFactory*, the first step is to import the CIM data into CIM objects (*CIM Data Import*). This offers the user the possibility to directly interact with the CIM data from within *PowerFactory*. The second step is to convert the CIM objects into a grid model (*CIM to Grid Conversion*). To export grids from *PowerFactory*, they must first be converted to CIM objects (*Grid to CIM Conversion*). These may still be modified if required, and also directly reimported. The newly-created CIM objects can then be exported as a ZIP or XML file in conformity with the CIM data structure (*CIM Data Export*).

24.13.1 CIM Data Import

The CIM Data Import is accessible in *PowerFactory* under *Tools* → *CGMES Tools*→ *CIM Data Import*. The following options for the import location are available:

- **New archive in new project** creates a new project with the given name and imports the data into an archive with the same name.
- **New archive in active project** imports the data into a new archive with the given name, within the currently active project.
- **Existing archive in active project** imports the data into the given archive; data models that are already contained in the archive will not be modified.

Import of instance data belonging to a profile (XML file)

By selecting "**New archive in new project**", the name of the project and the file to import can be specified. Upon clicking the **Execute** button, the content of the XML file will be imported. This step results in an active project containing the "CIM Model" folder, which itself has a single archive. This archive contains the model representations from the XML file.

Import of multiple profiles instance data (ZIP file)

By selecting "**New archive in new project**", the name of the project and the file to import can be specified. Upon clicking the **Execute** button, the content of the ZIP file will be temporarily extracted and imported. This step results in an active project containing the "CIM Model" folder, which has a single archive. This archive contains the model representations from the ZIP file.

Import of multiple files

To import several files in a row, the destination **Existing archive in active project** must be used. The archive created by the first step must be selected as the "Path" for all subsequent files. This will import the data into the archive provided.

24.13.2 CIM Data Export

The CIM Data Export is accessible via *Tools* → *CGMES Tools*→ *CIM Data Export* or *right-click* → *CIM Data Export*. The archive selected as "Source data" will be exported. This can be fine-tuned by an option for each available profile instance.

The option "Create archive for each CIM model" can be used to match the DACF and D2CF requirements by ENTSO-E to upload each individual profile within a separated zip file.

In both cases the naming rules can be defined per profile. As these rules might be different for various processes, the naming rules can be configured on the "Advanced Options" page according to the individual needs for each profile.

24.13.3 CIM to Grid Conversion

The CIM to Grid Conversion is accessible under *Tools* → *CGMES Tools*→ *CIM to Grid Conversion* or *right-click* → *CIM to Grid Conversion*

To convert all data contained in an imported archive, select "Source Archives" followed by **Execute**. This will additionally consider any valid difference models contained in the archive. To import a base model only, the difference models must be deleted before conversion. If only specific profile information should be exported, select "Convert selected profiles" and specify those that should be considered. It should be noted that the resulting subset of profiles still needs to be complete in terms of dependencies.

Additional information (e.g. SSH data) can be added to already converted models (i.e. EQ/TP); these must be selected as "Additional Archives".

To convert an archive including models that have dependencies on other models not included in the archive (e.g. the boundary grid), the other models must be specified as "Additional Archives".

For both selections "Additional Archives" as well as "Source Archives" the user may either select a single or multiple CIM archives in order to convert multiple archives at once or reference to multiple ones (e.g. the Boundary and the used EQ file) accordingly.

All available profiles in the selected archive will be shown on per MAS basis in the "Modelling Authority Sets" table. These MAS will be linked to existing grid elements if a matching rdfID is found, but can be adjusted manually if needed. Additionally the user can define how to deal with the models per MAS. Therefore, the options convert, link and ignore are available, where link means that the model is not converted, but only linked (e.g. an old version of a boundary is used). In case of conversion of an already existing (referenced) model, this will be updated in terms of new elements are added.

Note: A regular task in CGMES is to update the currently used boundary file. This can be achieved by referencing the old boundary grid in the CIM to Grid conversion and converting it. This way a merged boundary will be created (nodes that are used in the original Boundary that are not existing anymore in the new one will remain).

24.13.4 Grid to CIM Conversion

The Grid to CIM Conversion is accessible under *Tools* → *CGMES Tools*→ *Grid to CIM Conversion*.

The following options for the destination are available:

- **New archive** converts the networks into a new archive with the given name.
- **Existing archive** imports the data into the archive specified in "Target Archive"; if the archive already contains models for the selected networks, the original models will be preserved.

- **Additional Archives** is used when references to other archives are required.

For “Additional Archives” the user may either select a single or multiple CIM archives for example to manage dependencies on the correct Boundary grid.

Profile version: A drop-down menu allows the user to select which of the supported CGMES versions should be used.

A model for each profile selected will be created per network in the target archive. By default all profiles are selected. Boundary grids will only be exported with EQ and TP profiles (the respective boundary versions).

The option **Create difference models** creates a difference model, when selected. Note that difference models can only be created if *Existing archive* is selected as the destination. This archive should contain the base models for the difference models.

In order to create a bus-branch model, the option **Create bus-branch model** can be selected. When ticked, an internal reduction from a node-breaker model to a bus-branch model is done automatically. The resulting CIM archive will be a bus-branch model according to CGMES.

Convert network selection

The *PowerFactory* networks to be converted can be selected by ticking the checkbox. Only activated networks can be converted. If one of the networks is to be treated as boundary grid, the corresponding checkbox must be ticked as well. Each network to be converted must have a Modelling Authority Set URI.

- Two or more networks can be associated to a common Modelling Authority Set. In such a case, all networks and data associated to a MAS will be exported into the same model set.
- If the selection contains two or more Modelling Authority Sets (apart from the Boundary network), and SV and/or DL profiles are selected for export, these instance data will be exported into an “Assembled” model set. Otherwise, if only one MAS is converted (apart from the Boundary network) SV and DL data will be exported into the same model set as EQ and TP models.

Advanced Options

Further information for the model to be converted can be altered under the *Advanced Options* page:

- **Version** is an optional parameter to define the model version
- **Description** is an optional parameter to describe the model
- **Additional Parameters** are inputs causing specific behaviour and do not need to be used.

24.13.5 CIM Data Validation

The CIM Data Validation is accessible under *Tools* → *CGMES Tools*→ *CIM Data Validation* or *right-click* → *CIM Data Validation*.

This data validation is based on the UML profile information and can be used on CIM archives to validate their CGMES profile compliance. The archives used for this validation can be selected via “CIM Archives or Models”. Therefore a multiselection is possible.

Note: This build-in validator can be used to validate archives of third parties or in case there are issues in the conversion (e.g. missing dependencies). The validator is not officially supported by the ENTSO-E.

24.13.6 Import and Export of the EIC as additional parameter

Grid to CIM Conversion

The “IdentifiedObject.energyIdentCodeEic” attribute is not applicable to PowerFactory data-model. However, it is possible to assign an “EIC” to a *PowerFactory* element, by adding the following “User Attribute” entry at the end of the description field of *PowerFactory* elements:

<Attribute Name=“EIC” Type=“string”>the code</>

For *PowerFactory* elements where no “EIC” is provided, no “IdentifiedObject.energyIdentCodeEic” is set in the corresponding CIM object.

CIM to Grid Conversion

The “IdentifiedObject.energyIdentCodeEic” attribute is not applicable to *PowerFactory* data-model, thus is converted as a “User Attribute” entry at the end of the description field in PowerFactory elements as follows:

<Attribute Name=“EIC” Type=“string”>the code</>

If no “IdentifiedObject.energyIdentCodeEic” is set in a CIM object, the entry will not be created in the corresponding PowerFactory element. For CIM object classes which have no representation in *PowerFactory*, the “EIC” code is not converted, thus not visible in *PowerFactory* data-model (e.g. RegulatingControl, GeneratingUnit etc.).

24.14 Functional Mock-Up Interface

For a detailed description on the FMI refer to Section 30.12.1: Functional Mock-Up Interface.

24.15 OPC Interface

PowerFactory’s **OPC** interface is an asynchronous communication and data exchange mechanism used in process interaction and is widely applied in SCADA and control systems. This OPC implementation assumes that the *PowerFactory* software is executed as an OPC Client while the OPC Server is controlled via the external source. OPC server libraries are available from various manufacturers. An example of a freeware OPC Server is that available from Matrikon (“MatrikonOPC Simulation Server”).

PowerFactory supports both OPC DA (data access) and OPC UA (unified architecture) standards.

Figure 24.15.1 illustrates the integration of a SCADA system with *PowerFactory* via the OPC interface. In this OPC implementation, *PowerFactory* can be used either in GUI-less or normal mode. Some further characteristics of this integration include:

- OPC Client/Server exchange of any *PowerFactory* object parameter as well as any signal (bi-directional Data Exchange).
- *PowerFactory* listening mode to receive any data or signal from a registered OPC Server.
- *PowerFactory* sending mode to write back any data or signal to a registered OPC Server.

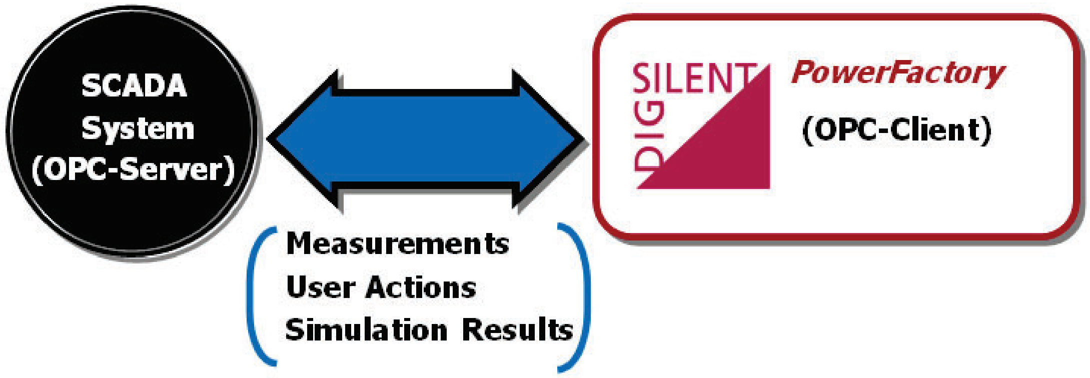


Figure 24.15.1: SCADA -*PowerFactory* integration via the OPC interface.

The OPC interface can be configured in two different modes:

- **Offline**
  - The bi-directional data exchange is carried out through an explicit command given by the user in *PowerFactory*. For example, by pressing a button predefined by the user in *PowerFactory*.
- **Online**
  - The bi-directional data exchange is automatically carried out at a certain frequency rate; where the frequency rate is determined by the user.

24.15.1 OPC Interface Typical Applications

Some typical applications of the OPC Interface are the following:

- **SCADA Online State Estimation**
- **SCADA Simulation Mode**, for example dispatcher load flow, switching validation.
- **SCADA Training Simulator**
- **Importing to *PowerFactory***
  - in order to update the operational data.
  - in order to reflect the Operator actions, such as breaker status and tap positions.
  - in order to perform state estimation based on the measured network data.
- **Exporting from *PowerFactory***
  - in order to update the SCADA interface with the calculated results.

24.16 StationWare Interface

This chapter describes the *StationWare* interface. An introduction into *StationWare* is provided in Section 24.16.1.

The following two sections describe the overall *StationWare* architecture (Section 24.16.2) and the conceptual differences between *PowerFactory* and *StationWare* (Section 24.16.3).

Both *PowerFactory* and *StationWare* have to be configured before they can be used together (Section 24.16.4).

The *Getting Started* section (Section 24.16.5) provides an introduction to the most important features. The complete documentation can be found in the section 'Description of the Menu and Dialogs' (Section 24.16.6).

The terms *StationWare* and ***PSMS*** are used synonymously throughout this chapter. ***PSMS*** stands for Protection Settings Management System, and stresses the more internal and technical part of *StationWare*.

24.16.1 About StationWare

DIgSILENT StationWare is a centralised asset management system for primary and secondary equipment. It provides a reliable central protection settings database and management system for the complete power system data, both to manage the various control parameters and to centrally store power system related information and data, based on the latest .NET technology.

StationWare stores and records all settings in a central database, allows modelling of all relevant work flow sequences, provides quick access to device manuals, interfaces with manufacturer-specific relay settings software, and integrates with *PowerFactory* software, allowing powerful and easy-to-use settings co-ordination studies.

Modern numerical relays have a large number of settings that are determined, stored and communicated by proprietary software solutions (these may be suitable for only one particular manufacturer or only one series or type of relay). This results in a fragmented and distributed settings "database". *DIgSILENT StationWare* provides a single system that incorporates all different device protocols, thereby providing one manageable software data storage system, based on modern IT techniques, facilitating data interfacing and exchange in a transparent and straightforward manner.

PowerFactory's data exchange facility allows it to access the settings stored in *StationWare*, such that these may be used as input to the powerful *PowerFactory* system simulation and protection settings tools. Settings that are calculated by using these tools may then be transferred back to *StationWare*.

24.16.2 Component Architecture

DIgSILENT StationWare is a so-called *Client-Server Application*: the functionality is distributed over at least two computers: client and server. Figure 24.16.1 gives an overview of the components.

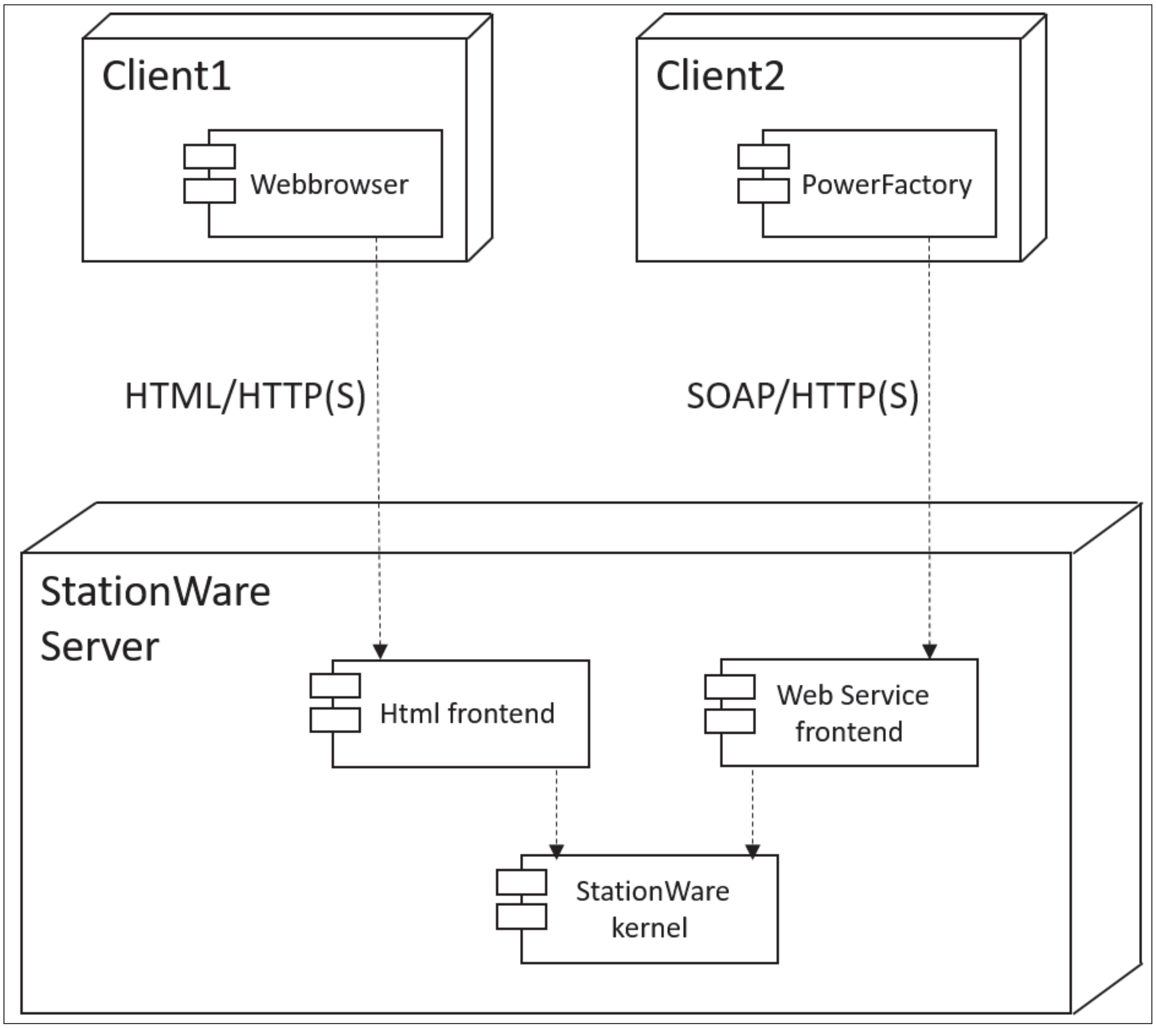


Figure 24.16.1: Architecture overview

There are usually several clients. One main advantage of this architecture is that the data is stored in one central database on the server. One client connects to the server and fetches the data from there, modifies it, and then stores it back to the server. These changes are visible on other clients.

DIgSILENT StationWare server provides two interfaces to access from client machines:

- Visualisation by means of a standard web browser. The HTML interface can be used with an usual web browser (e.g. Microsoft Internet Explorer or Mozilla Firefox).
  The browser displays HTML pages which are created by *StationWare*'s HTML front end. The HTML pages are transferred using the HTTP(S) protocol on top of the TCP/IP internet protocol.
  HTML allows to present all kind of data e.g. plain text, tables or images.
  Additionally HTML provides concepts to achieve interactivity: by submitting HTML forms or pressing on hyperlinks data is sent to the server. The server interprets such requests and creates new HTML pages which are displayed by the browser again.
- The web service interface, similar to the HTML interface uses the HTTP(S) protocol to communicate with the web service frontend, though no HTML pages are transferred but lower-level data (SOAP/XML encoded). The web service client application is responsible to present this data conveniently.
  PowerFactory is able to play the role of a web service client. It integrates parts of *StationWare*ś data and concepts smoothly into its own world.

Note: The default *StationWare* configuration requires SSL for the *StationWare* applications (web GUI and web services). Please use HTTP instead of HTTPS, if SSL is not enabled for your *StationWare* applications. In the following, the expression HTTP(S) is used.

The functionality of the HTML interface is covered in the *StationWare* manual. The remainder of this chapter focuses on *PowerFactory* as client.

24.16.3 Fundamental Concepts

Although *StationWare* and *PowerFactory* store data and settings associated with primary devices such as lines, transformers, . . . and secondary devices, i.e. relays, CTs, VTs and circuit breakers, the two systems utilise different concepts to deal with this data.

In *StationWare* it is possible to model a location hierarchy and associate the devices to nodes in this hierarchy (e.g. substations). This has no equivalent in *PowerFactory*, where the devices are stored inside the parent grid (*ElmNet*) object.

Conversely, *PowerFactory* allows to the creation of a topological representation of networks which is not supported in *StationWare*.

This section describes the concept mismatch between *PowerFactory* and *StationWare*. In order to use the *StationWare* interface, it is important to understand the differences between both applications.

Location

In *StationWare* each device belongs to exactly one location. There are different location types e.g. *Region, Area, Site, Substation*, or *Bay*. The locations are organised in a hierarchy tree as shown in Figure 24.16.2.

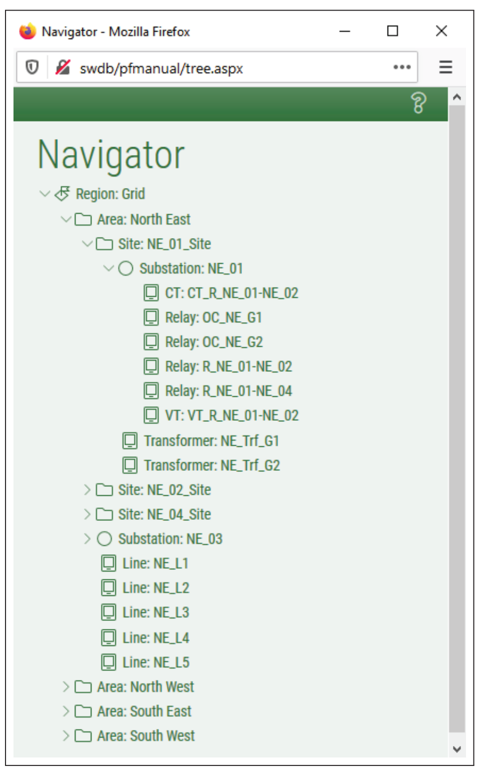


Figure 24.16.2: *StationWare* locations

In *PowerFactory* the data is organised in projects (*IntPrj*). A project may have one or more grids (*ElmNet*) which in turn contain net elements e.g. terminals, cubicles, and relays (*ElmRelay*). See Figure 24.16.3 for a typical *PowerFactory* project.

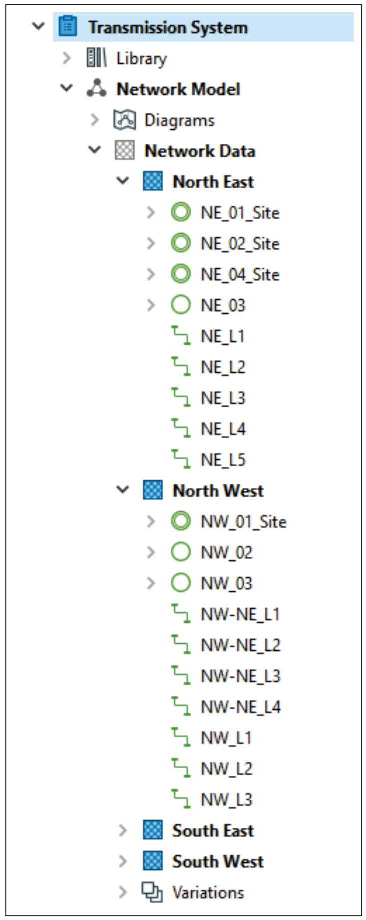


Figure 24.16.3: *PowerFactory* project

StationWare's location concept and *PowerFactory*'s project/grid concept hardly fit together. That's the reason why the data mapping between *PowerFactory* and *StationWare* begins at the device level which is the subject of the next sections.

Device

StationWare manages a set of devices e.g. relays, CTs, VTs, circuit-breakers, ... Each device is associated with a device type e.g. *ABB DPU2000R* or *SEL421 003*. In addition, each device has an unique ID: the *device ID*.

In *PowerFactory* a relay is represented by an *ElmRelay* object which references exactly one *TypRelay* object. The *ElmRelay* object contains several sub-components e.g. the I> component (a *RelToc* object), the Logic component (*RelLogic*), or the Ios component (*RelMeasure*). See Figure 24.16.4 for an example. The device ID is used to link one *StationWare* device to one *PowerFactory* device. The *PowerFactory* device e.g. an *ElmRelay* object stores the *StationWare* device ID as foreign key.

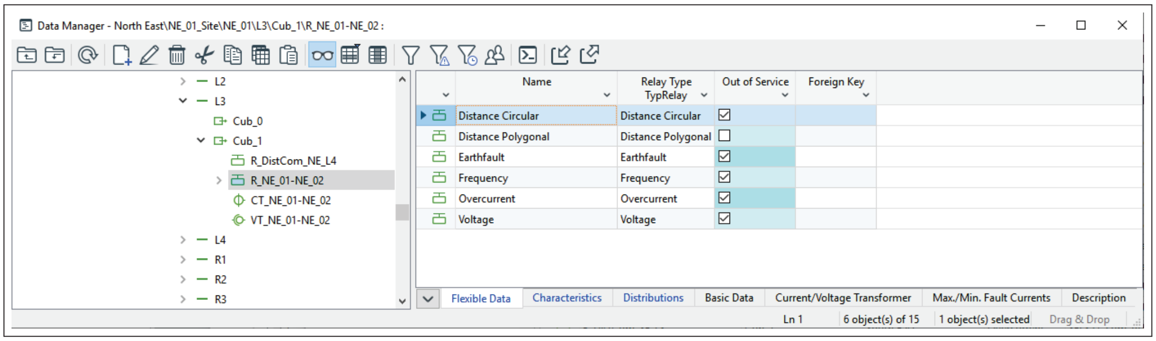


Figure 24.16.4: *PowerFactory* relay

Device State

A device's state is in *StationWare* called setting. A setting is a list of parameters, and describes the state of one device completely. An parameter is a tuple of

- parameter *name*,
- parameter *type* which can be an arbitrary integer or floating point number, optionally with a range restriction, or a string, or a enumeration type.,
- a *default* value,
- an optional *unit*.

A complex relay may have thousands of parameters. In *StationWare* the setting parameters are organised in so-called setting groups. A setting group groups the parameters together which belong somehow together. It's often defined by the device manufacturer. Each parameter belongs to exactly one setting group. Inside a group the parameter name is unique.

The device type defines which parameters and groups characterise a device. Table 24.16.1 shows an example of a possible device type. There are two setting groups G and H. Group G has the parameters a, b, and c, group H has the parameters d and e.

Group	Name	Type	Default	Unit
G	a	integer in [0,10]	0	A
	b	float	-0.32	l/s
	c	float in [0.03, 4.65]	1.0	
H	d	string	'DEFAULT'	
	e	enum 'yes', 'no', 'maybe'	'yes'	

Table 24.16.1: Settings Definition

According to this parameter definition a device can have settings as shown in tables 24.16.2 or 24.16.3.

Group, Name	Value
G,a	7
G,b	23.43
G,c	1.1
H,d	'abc'
H,e	'maybe'

Table 24.16.2: Settings Example 1

Group, Name	Value
G,a	8
G,b	0
G,c	1.1
H,d	'abcdef'
H,e	'yes'

Table 24.16.3: Settings Example 2

On the *PowerFactory* side there are neither settings nor groups. There is the *ElmRelay* object and its sub-objects. These objects can have parameters. See Table 24.16.4 for a definition and Table 24.16.5 for an example. The *TypRelay* type defines components and parameters.

StationWare parameters are mapped to *PowerFactory* parameters and vice versa. The mapping is non-trivial since only a small subset of the parameters (the calculation-relevant data) is modelled in *PowerFactory* and vice versa. Additionally there is no one-to-one relationship between the *StationWare* and *PowerFactory* parameters; i.e. a *PowerFactory* parameter may be calculated from several *StationWare* parameters.

Component	Parameter	Type
i>	o	integer
Logic	p	string
	q	enum 'enabled','disabled'
Ios	r	float
	s	float

Table 24.16.4: Parameter Definition

Some relays support *multiple setting groups* (MSG) also called *parameter sets*. Such relays have the same group many times (c.f. table 24.16.5). The groups H1, H2, and H3 have the same set of parameters (c and d). The relay models in *PowerFactory* do not support this concept. Instead of modelling all MSGs, only one instance of the H groups is provided.

In this case a group index parameter defines which of the MSGs actually is transferred from *StationWare* to *PowerFactory*.

Lifecycle Phase

In *StationWare* each setting has one lifecycle phase e.g. *Planning* or *Applied*. At each point in time a device can have a set of settings e.g. three *Planning* settings, one *Applied* setting and 12 *Historic* settings.

Component Parameter	Value
i>:o	8
Logic:p	'HIGH'
Logic:q	'enabled'
Ios:r	18,5
Ios:s	19,5

Table 24.16.5: Parameter Example

Group	Name	Type	Default	Unit
G	a	integer in [0,10]	0	A
	b	float	-0.32	l/s
H1	c	string	'DEFAULT'	
	d	float in [0.03,1.65]	1.0	
H2	c	string	'DEFAULT'	
	d	float in [0.03,1.65]	1.0	
H3	c	string	'DEFAULT'	
	d	float in [0.03,1.65]	1.0	

Table 24.16.6: Multiple Setting Group Definition

In *PowerFactory* a device has exactly one state (or setting). Therefore when data is transferred between *PowerFactory* and *StationWare* always a concrete device setting in *StationWare* must be specified.

For *PowerFactory* purposes a special *PowerFactory* planning phase is introduced. The transfer directions are specified as follows:

- Imports from *StationWare* into *PowerFactory* are restricted to `Applied` and `PowerFactory` settings. Applied denotes the current applied setting (`Applied`) or a previous applied (`Historic`) setting.
- Exports from *PowerFactory* to *StationWare* are restricted to the `PowerFactory` setting. (`Applied` and `Historic` settings are read-only and can never be changed).

(Actually *PowerFactory*ś sophisticated variant management is similar to the phase concept, but there is no obvious way how to bring them together.)

24.16.4 Configuration

In order to transfer data between *PowerFactory* and *StationWare* both systems must be configured.

***StationWare* Server**

An arbitrary *StationWare* user account can be used for the *StationWare* interface in *PowerFactory*. The user must have enough access rights to perform operations e.g. for the export from *PowerFactory* to *StationWare* write-rights must be granted.

The bi-directional transfer of settings is restricted to lifecycle phases

1. of the phase type PLANNING or REVIEW and
2. with a cardinality constraint of 1 i.e. there may exist one or no such setting for one device.

Ensure that at least one phase fulfils these requirements, and there exists a setting of this phase.

***PowerFactory* Client**

The client operating system must allow connections to the server (network and firewall settings etc.).

Nothing has to be done in the *PowerFactory* configuration itself. The *TypRelay* in the Library must of course support *StationWare* - *PowerFactory* mapping.

24.16.5 Getting Started

The mapping between *PowerFactory* object attributes and calculation results with *StationWare* device settings or process attributes, or additional attributes of devices, is done via flexible DPL scripts. These scripts have access not only to data in *PowerFactory* objects themselves, but also to other related objects e.g a relay type object or relay sub-blocks.

To be able to transfer data from *PowerFactory* to *StationWare* and vice versa, suitable DPL scripts have to be created and placed in an appropriate location in the project library folder.

```
Project.IntPrj
+- Library.IntPrjfolder
   +- Equipment Type Library.IntPrjfolder
      +- Relay Type X.IntFolder                        //TypRelay-specific scripts
      |   +- Relay Type X.TypRelay
      |       +- PsmsExport.ComDpl
      |       +- PsmsImport.ComDpl
      |
   +- Operational Library.IntPrjfolder
   +- Scripts.IntPrjfolder
   +- StationWare.IntPrjfolder                         //StationWare stuff
      +- Attributes.IntFolder                          //Folder for additional attributes
      |   |
      |   +- ElmLne.IntFolder
      |   |   +- PsmsExport.ComDpl
      |   |   +- PsmsImport.ComDpl
      |   +- ElmTr2.IntFolder
      |       +- PsmsExport.ComDpl
      |       +- PsmsImport.ComDpl
      |
      +- Results.IntFolder                             //Folder for results export
      |   |
      |   +- arcflash.IntFolder
      |   |   +- ElmTerm.IntFolder
      |   |   |   +- PsmsExport.ComDpl
      |   |   +- ElmLne.IntFolder
      |   |       +- PsmsExport.ComDpl
      |   |
      |   +- shc.IntFolder
      |       +- ElmTerm.IntFolder
      |       |   +- PsmsExport.ComDpl
      |       +- ElmLne.IntFolder
      |           +- PsmsExport.ComDpl
      |
      +- Settings.IntFolder                            //Device settings export/import
      |   |
      |   +- ElmLne.IntFolder
      |   |   +- PsmsExport.ComDpl
      |   |   +- PsmsImport.ComDpl
      |   +- ElmTr2.IntFolder
      |       +- PsmsExport.ComDpl
      |       +- PsmsImport.ComDpl
```

Figure 24.16.5: Structure of the project library folder

The scripts for importing/exporting device settings should be located in the sub-folder “Settings” of the *StationWare* folder inside the Project in *PowerFactory*.
`Project\Library\StationWare\Settings`.

For importing/exporting additional attributes from *StationWare* DPL scripts should be located in the sub-folder “Attributes”. To be able to export results from *PowerFactory* to *StationWare* the corresponding script should be located in the sub-folder “Results”. None of these folders are by default in project library folder and must therefore be created.

Important: DPL scripts for import/export relay settings must be saved in the same folder as the relay model (as contents of a *TypRelay* object).

In difference to the data exchange of device settings, additional attributes and results, the DPL import/-export scripts for relay settings can refer to mapping tables which simplify the mapping of individual parameters and the implementation of dependencies between parameters. Therefore, there are two different ways of exchanging relay settings. Either the mapping of the parameters in the DPL script code or the parameter mapping in mapping tables. Depending on which variant is selected, the basic DPL scripts differ. More information about the different mapping possibilities can be found in the documentation for Protection Devices.

24.16.5.1 Import/Export of Relay Settings

This section is a simple walk-through and covers the most essential *StationWare* interface functionality.

By using a basic *PowerFactory* project and basic *StationWare* substation, it describes

1. how relays in *StationWare* and *PowerFactory* are created,
2. how these relays are linked,
3. how settings can be exported from *PowerFactory* to *StationWare*
4. how settings can be imported again into *PowerFactory*.

All (especially the more advanced) options and features are described in the section 'Description of the Menu and Dialogs' (see Section 24.16.6).

Prepare substation in StationWare

We begin with the *StationWare* side. We create a substation and two relays within:

- Start the web browser,
- log on to the *StationWare* system,
- create a new substation titled *Getting Started*,
- create two relays named *Getting Started Relay 1* and *Getting Started Relay 2* in the *Getting Started* substation.

In the HTML interface the station detail page should look as shown in Figure 24.16.6.

- Go to the detail page of the *Getting Started Relay 1* (Figure 24.16.7).

Since we have just created the device it has no settings, yet. Later it will contain a *PowerFactory* setting which reflects the relay state on the *PowerFactory* side.

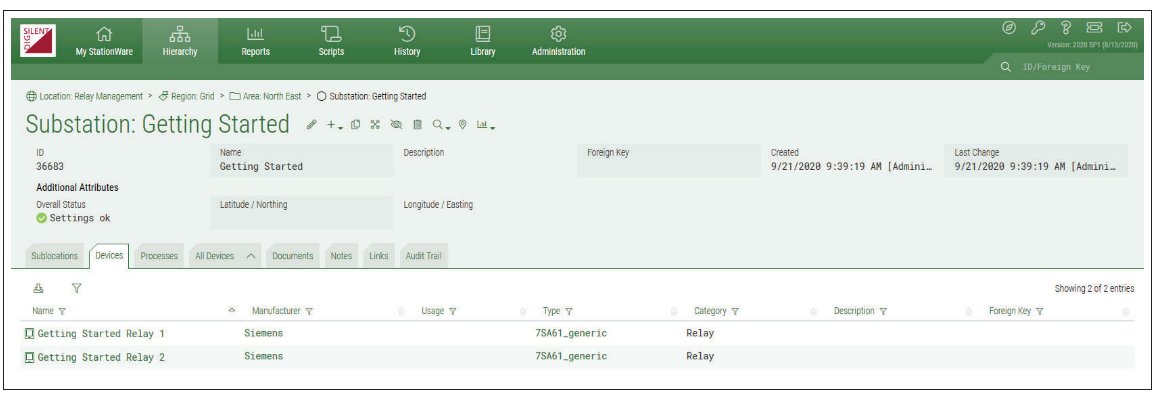


Figure 24.16.6: Substation

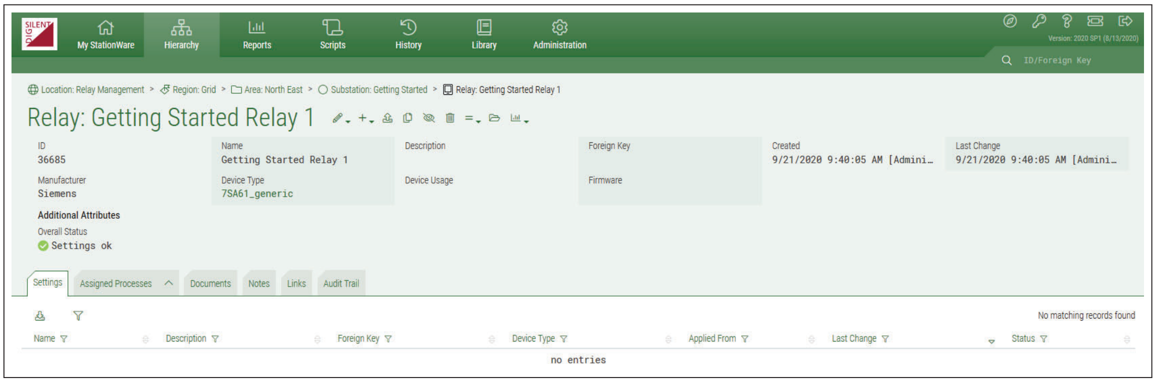


Figure 24.16.7: Device

Prepare project in *PowerFactory*

Create a new *PowerFactory* project and create a simple grid within:

- Start *PowerFactory*,
- create a new project titled *GettingStarted*,
- draw a simple grid with two terminals (*ElmTerm*) connected by a line (*ElmLne*) as shown in Figure 24.16.8.

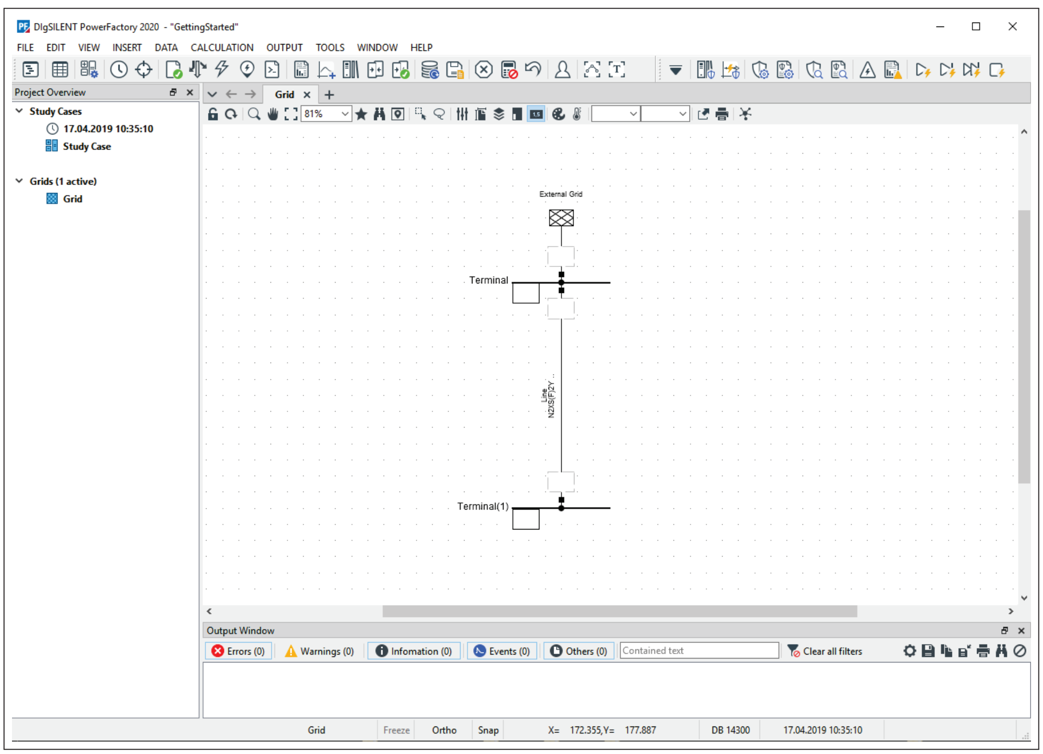


Figure 24.16.8: Grid

Now add a relay to the upper terminal:

- Right-click the cubicle quadrangle with the mouse. A context menu pops up.
- Select *New Devices.../Relay Model...* as shown in Figure 24.16.9.

A dialog pops up that allows you to specify the settings of the new relay (*ElmRelay*).

- Insert *Getting Started Relay 1* as Name,
- select an appropriate *Relay Type* which supports *StationWare* import/export (see Figure 24.16.10),
- press **OK**,
- in the same way add a relay *Getting Started Relay 2* to the second terminal.

PowerFactory's object filter mechanism gives an overview over all devices inside the current project.

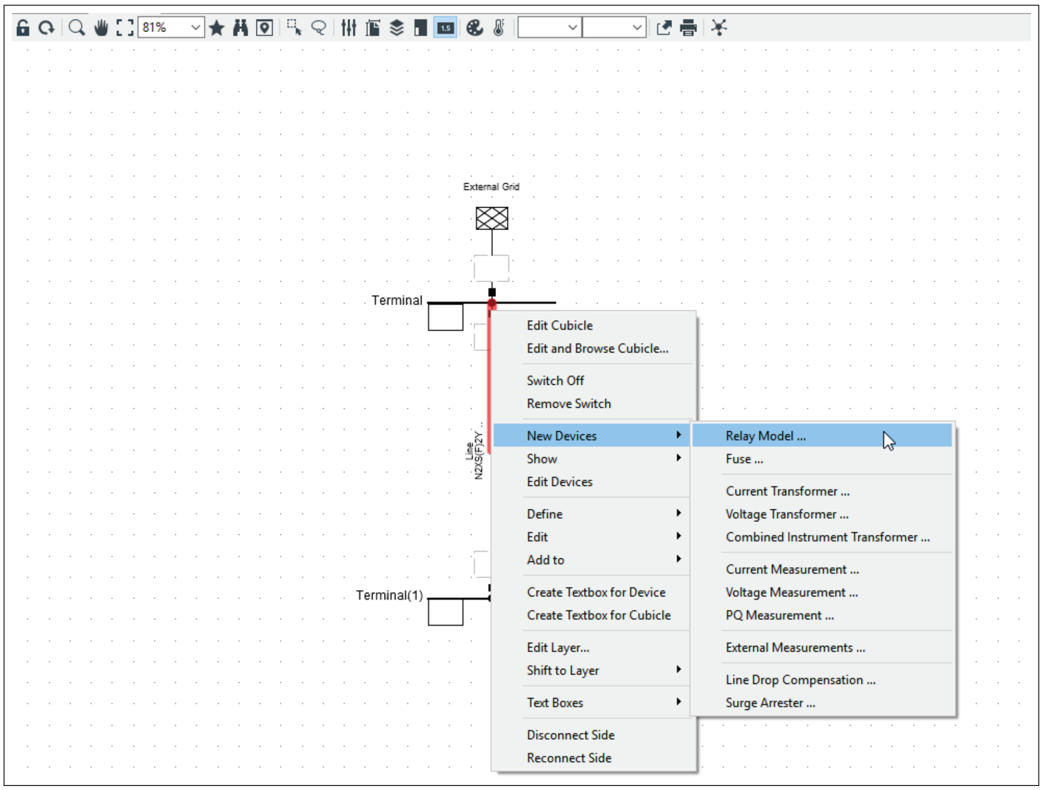


Figure 24.16.9: Cubicle context menu

- Press the icon (*Open Network Model Manager...*) in the toolbar and select the class (*ElmRelay*) to filter out all non-relay objects.

All calculation relevant relays (actually there only the two we created above) are displayed in a table (see Figure 24.16.11).

Link Relays and establish a Connection

Now the *PowerFactory* relays must get linked to the *StationWare* relays. To be able to make a connection:

- Ensure that the DPL Import/Export scripts are saved in the same folder as the relay model. If mapping tables are used, ensure that the path and the name of the mapping tables is set in the DPL Import/Export scripts.
- Mark both relay icons with the mouse.
- Press the right mouse button.

A context menu pops up as shown in Figure 24.16.12.

- Select the *StationWare* menu item,
- select the *Select Device ID* item.

A Log on to *StationWare* server dialog pops up. Since this is the first time *PowerFactory* connects to the *StationWare* server some connection settings must be entered.

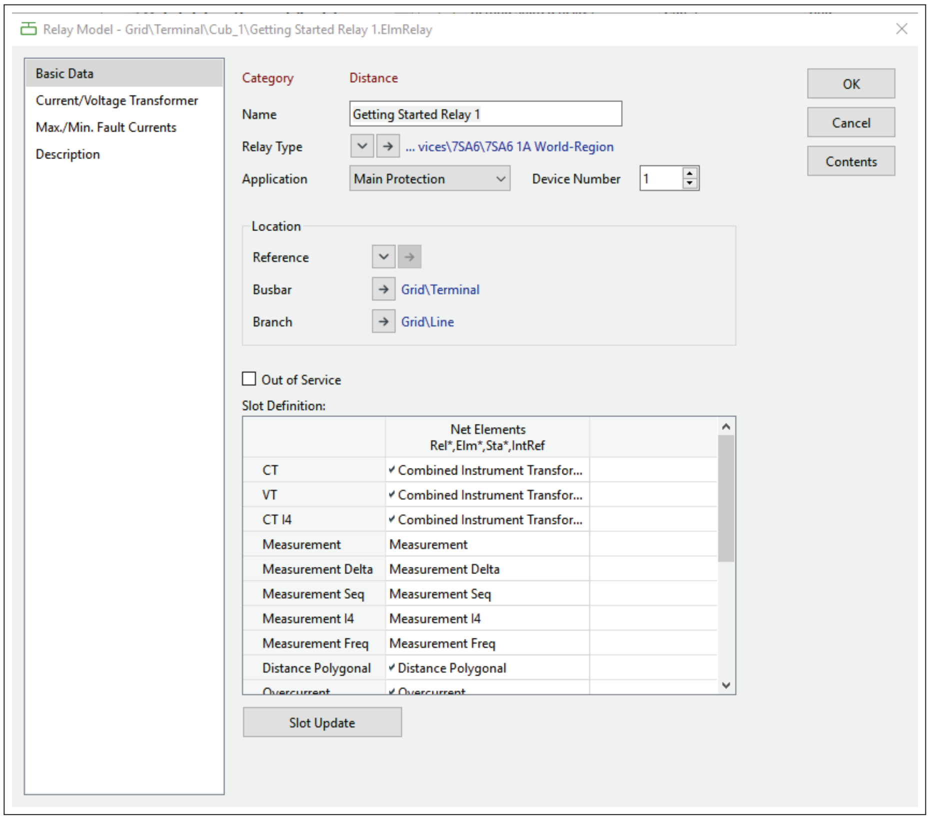


Figure 24.16.10: Relay dialog

- Enter the Server Endpoint URL of the *StationWare* server. The URL should have a format similar to
`http(s)://192.168.1.53/psmsws/PSMSService.asmx`.
- Enter *Username* and *Password* of a valid *StationWare* user account.

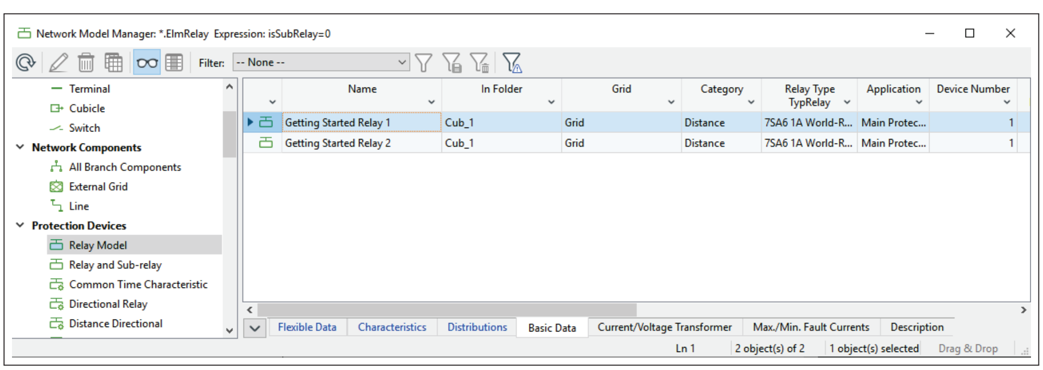


Figure 24.16.11: Relay display

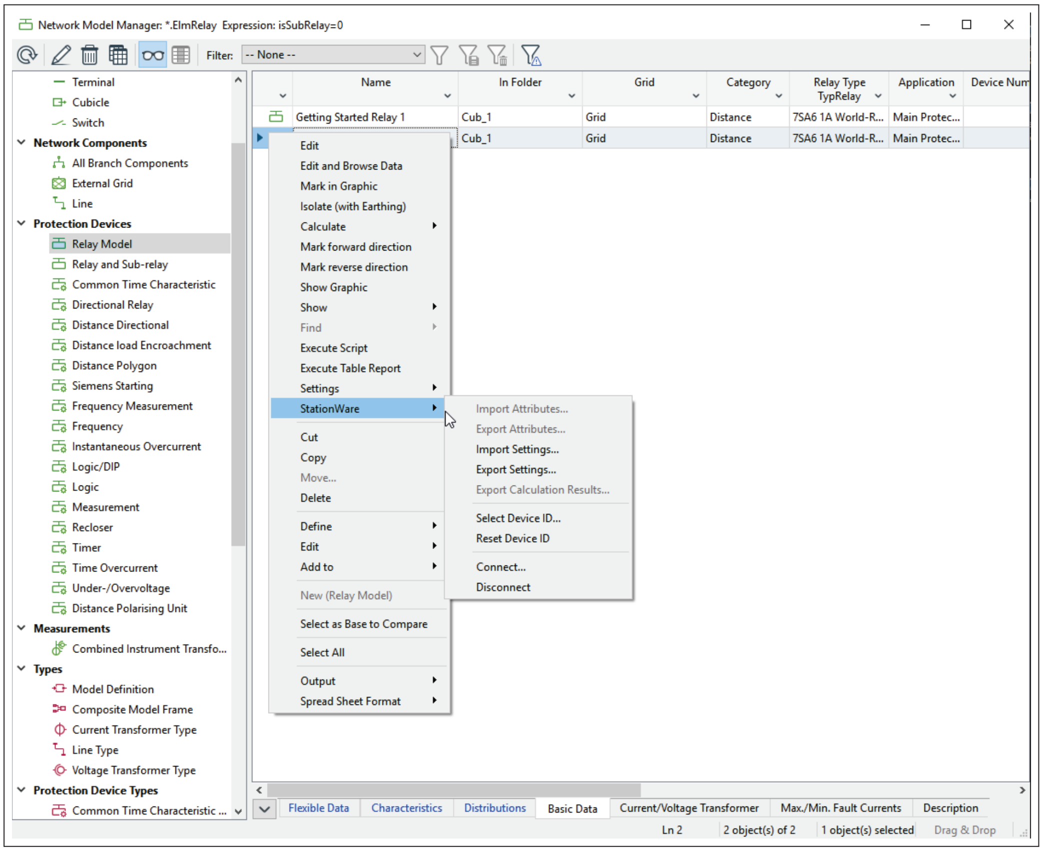


Figure 24.16.12: Device context menu

Figure 24.16.21 shows the dialog settings.

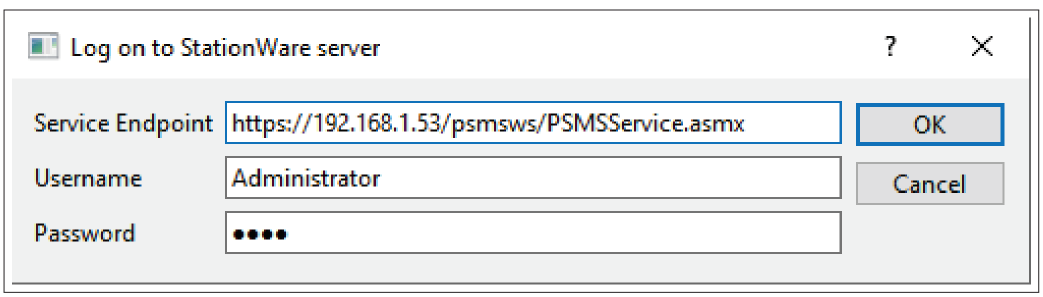


Figure 24.16.13: Log on dialog

- Press **OK**.

The connection procedure may take some seconds. If the server could be accessed and the user could be authenticated a success message is printed into the output window

```
Established connection to StationWare server
'https://192.168.1.53/psmsws/PSMSService.asmx'(version 18.2.6981) as user
'Administrator'
```

Otherwise an error dialog pops up. Correct the connection settings until the connection is successfully created. The section 'Description of the Menu and Dialogs' (Section 24.16.6) explains the connection options in detail.

Having established a connection to the server, a browser dialog pops up which displays the location hierarchy as known from the *StationWare* HTML interface. The dialog is shown in Figure 24.16.14.

- Navigate to the *Getting Started* substation,
- select the *Getting Started Relay 1* device,
- press **OK**.

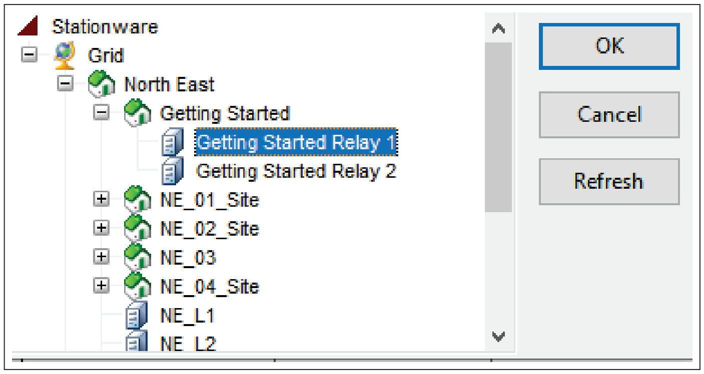


Figure 24.16.14: Browser dialog

Now the *PowerFactory* relay is “connected” to the *StationWare* device.

- In the same way select *Getting Started Relay 2* for the second *PowerFactory* relay.

Export and Import Settings

Having linked *PowerFactory* to *StationWare* devices, the transfer between both systems can be started.

- Mark the relays with the mouse and right-click to get the relay context menu as shown in Figure 24.16.12.
- Select the *Export Settings. . .* in the *StationWare* menu entry.

A *ComStationware* dialog is shown which allows to specify the export options. See section ’Export and Import Settings’ in the chapter 24.16.6 for all export options.

- Select *PowerFactory* as lifecycle phase,
- press **Execute**.

After a few seconds the relay settings are transferred to the server, and the output window contains the message

```
Exported 2 of 2 device settings successfully
```

The result can now be observed in the *StationWare* HTML interface.

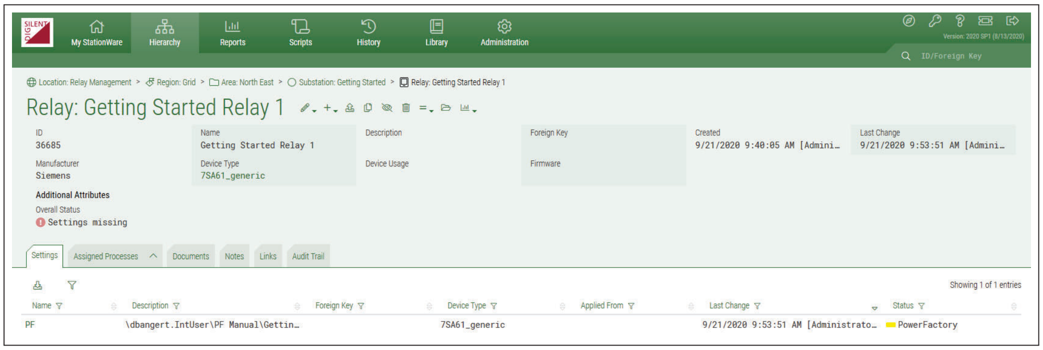


Figure 24.16.15: Device detail page

- Navigate to the relay detail view of the *Getting Started Relay 1* relay (see Fig. 24.16.15)

Observe the new created PF setting. The phase of this setting is *PowerFactory*.

- Switch to the settings detail page of the new *PowerFactory* setting (see Fig. 24.16.16).

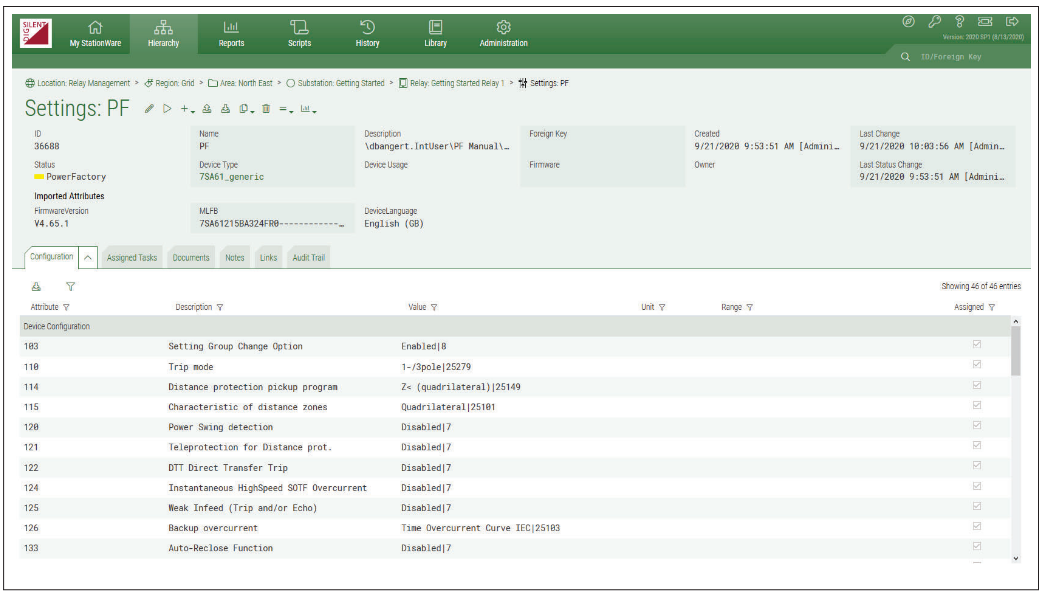


Figure 24.16.16: Setting detail page

The setting values should correspond to the relay state in *PowerFactory*. In the same way the *Getting Started Relay 2* relay has a new PF setting.

Now try the opposite direction and import a setting from *StationWare* into *PowerFactory*.

- Modify the PF settings in *StationWare* by entering some other values.
- In *PowerFactory* mark the relays with the mouse and right-click to get the relay context menu as shown in Figure 24.16.12.
- Select the *Import Settings. . .* in the *StationWare* menu entry.

Again the *ComStationware* dialog pops up as known from the export.

- Leave the default settings,
- press **Execute**.

Again the result of the settings transfer is reflected in the output window:

```
Imported 2 of 2 device settings successfully
```

- find *ElmRelay* object parameters changed according to the changes on the *StationWare* side

All import options are described in detail in the section 24.16.6: Import/Export Options.

24.16.5.2 Import/Export of the Additional Attributes

Additional attributes represent additional information which users may find useful for a location, device or settings within a device. These are not directly part of a settings record but are user-defined. For example, a common additional attribute that is useful for a feeder or substation location is the nominal voltage level in kV. Primary elements such as lines do not posses settings but instead parameters. Parameters such as length or impedance are then presented by the use of additional attributes.

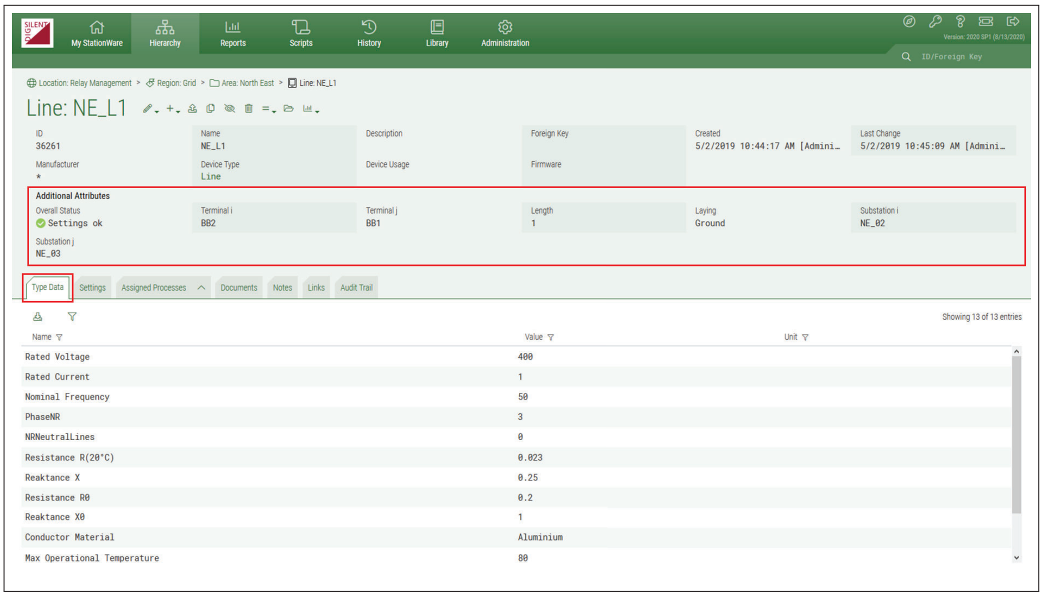


Figure 24.16.17: Additional attributes on the 'Device' page

The following information for additional attributes can be imported/exported:

- Name of the attribute
- Description
- Unit
- Value (Bool, String, Integer, Real, Enumeration, Data Time)
- Type (Attribute, Propagate, Overall Status, Revision Number)

Import/export of additional attributes also requires that the DPL script be saved in the appropriate place (see Section 24.16.5). All actions are similar to those described for settings (see Section 24.16.5.1) .

24.16.5.3 Export of the calculation results

Calculation results data exchange is only possible in one direction: from *PowerFactory* to *StationWare*. It is important to know that *PowerFactory* stores calculation results in attributes of temporary so-called "calculation objects".

This data will be exchanged between “calculation objects” and *StationWare* process objects.

Preparation of *StationWare* for importing result data

Inside a *StationWare* project, define the process lifecycle, category and type. This process object should be configured to be capable of result data storage and presentation (e.g. “ArcFlashLabel Type” see 24.16.18).

Important: Process lifecycle must posses a phase named “*PowerFactory*” of type “Planning”.

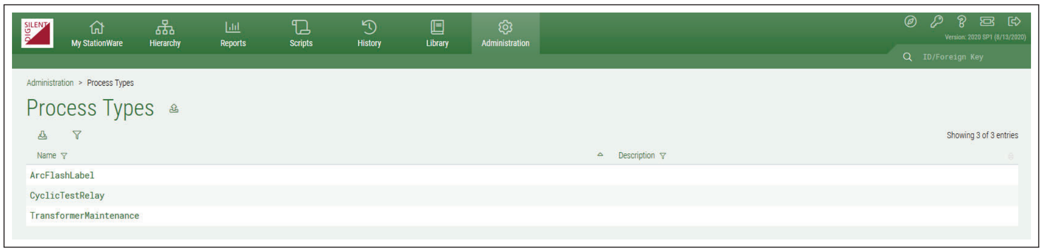


Figure 24.16.18: Process types page in *StationWare*

After being defined, the process should be created and have a device assigned to it.

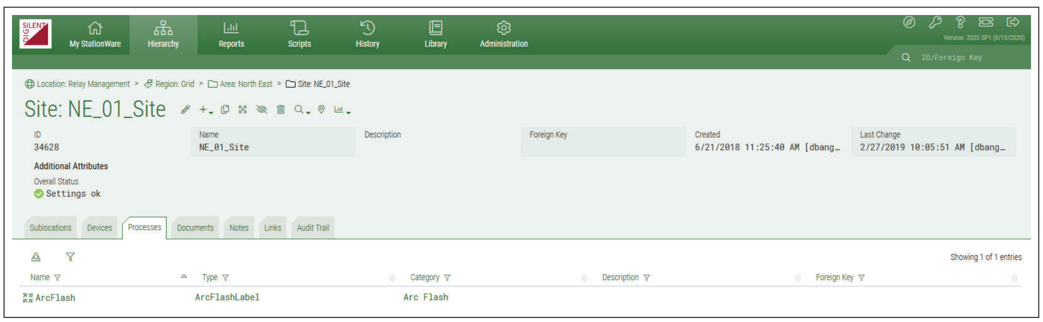


Figure 24.16.19: Location page where process is created

Preparing *PowerFactory* for export of result data

In *PowerFactory* it is important to have the DPL transfer script created and saved in a proper place inside the project library folder (see Section 24.16.5). It is necessary to use separate scripts for each calculation type and for each *PowerFactory* object class.

Connection of *PowerFactory* and *StationWare*

Refer to Section 24.16.5.1.

Export of results

Refer to similar section 24.16.5.1.

24.16.6 Description of the Menu and Dialogs

This section describes all options and features concerning the *StationWare* interface.

The Device Context Menu

Almost all functionality can be accessed by the device context menu. Mark one or more objects which supports the *StationWare* transfer e.g. *ElmRelay*

- in the object filter (Figure 24.16.12)
- in the Data Manager as shown in Figure 24.16.20.

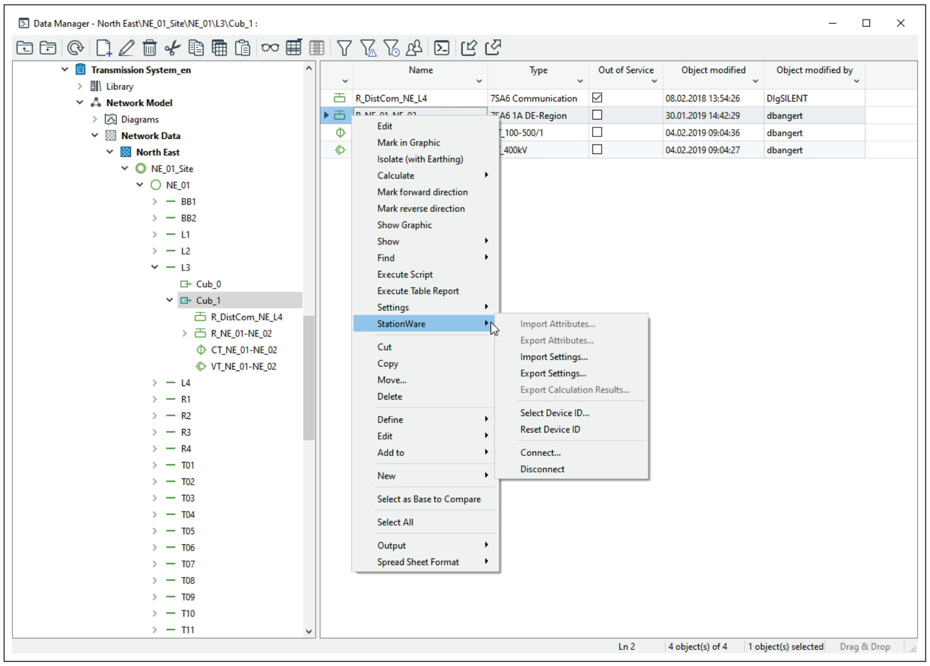


Figure 24.16.20: Device context menu

The *StationWare* submenu contains the entries as follows:

Import X... opens the *ComStationware* dialog and sets the device selection according to the above selected device objects. The *ComStationware* dialog settings are explained in detail in Section 24.16.6: The *ComStationware* Object.

Export X... does the same for the export direction.

Select Device ID... starts the Browser dialog (Figure 24.16.24) to link this device to a *StationWare* device. The dialog is subject of Section 24.16.6 : The Browser dialog.

Reset Device ID resets the device ID.

Connect... terminates the current *StationWare* session if it's already existing. Shows a Log On dialog. The connection settings are covered by Section 24.16.6. This may be useful when you are using several *StationWare* accounts and want to switch between them.

Disconnect terminates the *StationWare* session

Connection

Similar to the HTML interface the *StationWare* interface in *PowerFactory* is session - oriented: when a user logs on to the system by specifying a valid *StationWare* account (username and password) a new session is created. Only inside such a session *StationWare* can be used. The account privileges restrict the application functionality e.g. an administrator account is more powerful than a usual user account.

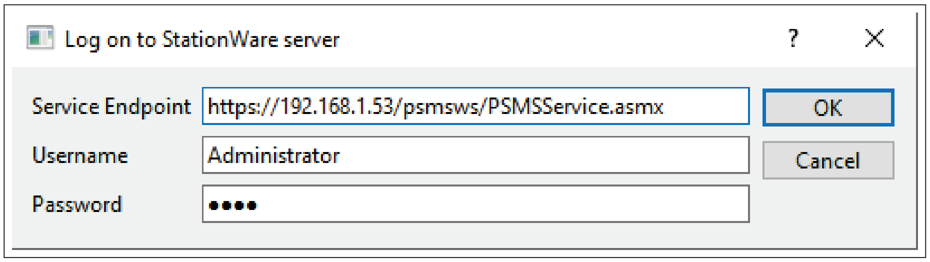


Figure 24.16.21: Log on dialog

Working with *PowerFactory* for the first time, the *StationWare* server is required, and the Logon dialog is as shown in Figure 24.16.21.

The *StationWare* connection options are stored in the user settings (Figure 24.16.22). After each successful logon the user settings are updated.

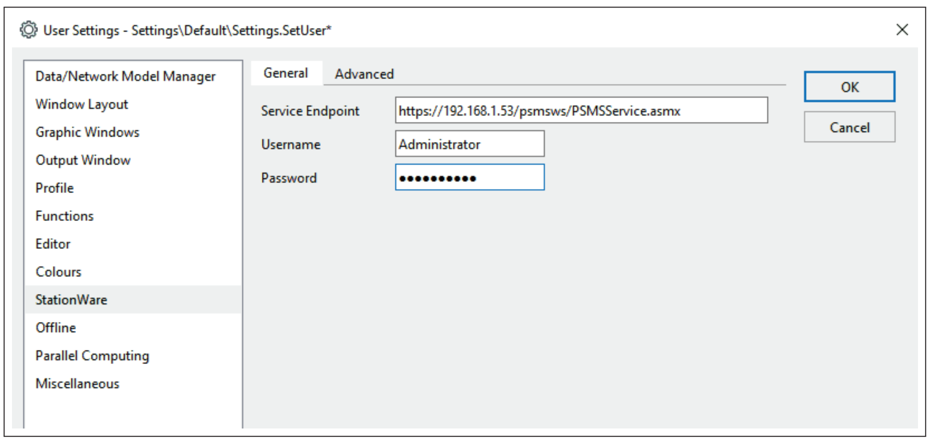


Figure 24.16.22: Log on dialog

As mentioned in the Architecture section (Section 24.16.2) *StationWare* is a client-server application. The *StationWare* server component is located on a server machine in the internet. The client component is the *PowerFactory* application which is running on a client machine.

The technology *PowerFactory* and *StationWare* use to communicate is called web services and is standardised like many other internet technologies (HTML, HTTP(S)). The server computer (or more exactly the *StationWare* service application on the server computer) has a 'name' by which it can be accessed. This 'name' is called service endpoint and resembles a web page URL:

```
https://the.server.name/psmsws/PSMSService.asmx
```

or

```
https://192.168.1.53/psmsws/PSMSService.asmx
```

http(s) denotes the protocol, the.server.name is the computer name (or DNS) of the server computer and psmsws/PSMSService.asmx is the name of the *StationWare* application.

The connection options are as follows:

Service Endpoint The Service Endpoint denotes the *StationWare* server 'name' as described above

Username/Password Username and Password have to be valid user account in *StationWare*. A

StationWare user account has nothing to do with the *PowerFactory* user account.

The very same *StationWare* account can be used by two different *PowerFactory* users. The privileges of the *StationWare* account actually restrict the functionality. For device import the user requires read-access rights. For exporting additionally write-access rights are required.

The Browser Dialog

As mentioned in the Concept description (see Section 24.16.3: Device) the *StationWare* device ID is stored as Foreign Key in the e.g. *ElmRelay* object dialog (Description page) as shown in Figure 24.16.23.

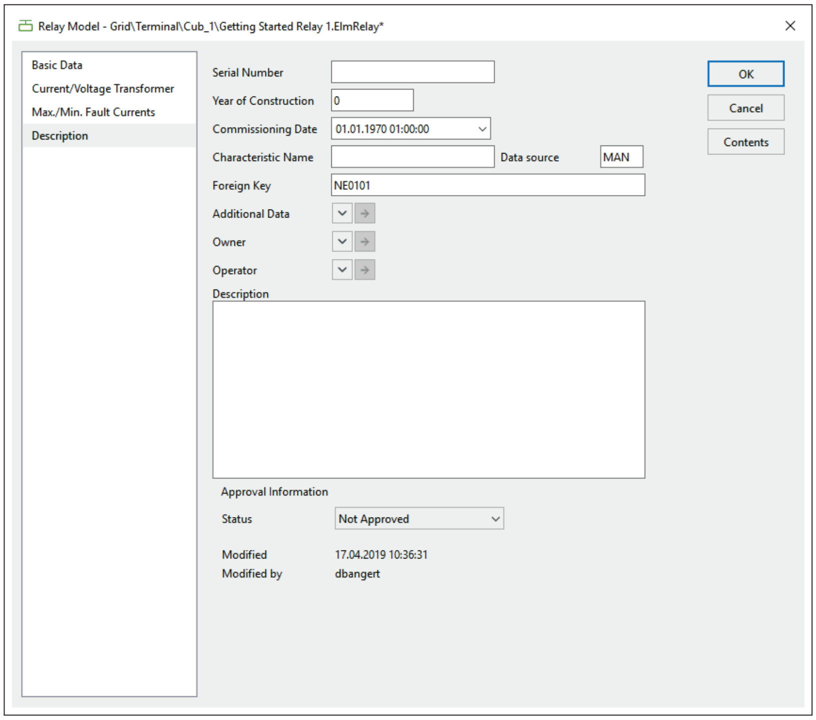


Figure 24.16.23: *ElmRelay* dialog

A more convenient way is to use the Browser dialog shown in Figure 24.16.24. The dialog allows to browse through the *StationWare* location hierarchy and select a device. The hierarchy data is cached to minimise network accesses. Due this caching it's possible that there may exist newly created locations or devices which are not displayed in the browser dialog. The **Refresh** button empties the cache and enforces *PowerFactory* to re-fetch the correct data from the server.

The ComStationware Object

In *PowerFactory* almost everything is an object: relays are *ElmRelay* objects, users are *IntUser* objects, and grids are *ElmNet* objects, . . .

What may be on the first sight confusing is the fact that actions are objects as well: for a short-circuit calculation a *ComShc* object is created. The calculation can be performed with several options e.g. 3-Phase, single phase, or 3 Phase to Neutral.

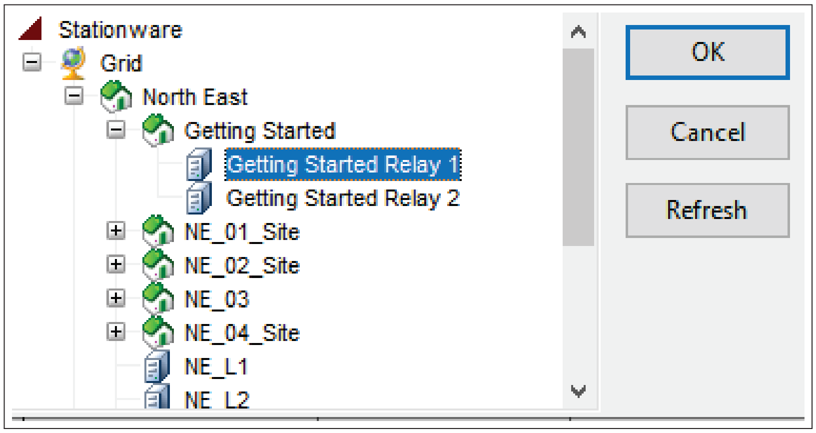


Figure 24.16.24: Browser dialog

You can even specify the fault location. All these calculation options are stored in the *ComShc* object. Every action object has an **Execute** button which starts the action. In fact there is a large number of parameterised actions like load flow calculation (*ComLdf*), simulation (*ComSim*), there is even a *ComExit* object that shuts down *PowerFactory*. All objects which can 'do' something have the Com prefix.

Since the *StationWare* interface is actually 'doing' something (it does import data, it does export data) it is implemented as a *ComStationware* object.

The *ComStationware* object is used both for the import and the export. It is located in the project's study case according to *PowerFactory* convention.

By default the study case of a new project contains no *ComStationware* object. It is automatically created when it is first needed, as well as the *ComShc* object is instantiated at the time when the first short-circuit calculation is performed.

Import/Export Options

The *ComStationware* dialog provides import/export options as follows:

Transfer Mode select Import/Export from *StationWare* as Transfer Mode

Transfer Data select Import/Export Data from *StationWare* (Attributes, Settings, Results of last calculation)

Check only Plausibility if the Check only Plausibility flag is enabled the import is only simulated but not really executed.

Lifecycle Phase/Time stamp A list of available lifecycle phases is shown.

- *PowerFactory* selects the current setting with *PowerFactory* phase as source setting.
- If Applied is selected the current Applied setting is transferred. If additionally a Timestamp value is entered the setting that was applied at this time is transferred which may either be Applied or

Historic.

The Timestamp format is in ISO format: e.g. 2005-02-28 22:27:16

The time part may be omitted. Then 00:00:00 AM is assumed.

All Devices If All Devices is enabled, all calculation-relevant devices are imported/exported.

Device Selection Unless All Devices is enabled, the Device Selection provides a more subtle way to specify which devices are to be transferred.

The Device Selection is automatically set if the Device Context Menu mechanism (Section 24.16.6: The Device Context Menu) is used.

All Settings Groups/Group Index This parameter specifies how multiple settings groups (MSG) are handled.

The import/export transfer is started by pressing **Execute**.

24.17 API (Application Programming Interface)

For a detailed description on the API, a reference document is available via the main menu *Help* → *Additional Packages*→ *Programming Interface (API)*