Release Information – PSS®SINCAL Platform 23.0

This document describes the most important enhancements and changes in the new program version. See the product manuals for more detailed descriptions of the functionalities.

PSS®SINCAL **2**
Example Networks 2
Graphical User Interface 2
General Extensions 2
Network Graphic 5
Documentation 6
Electrical Networks 6
Harmonic Stability (OB) 6
Load Allocation (LA) 11
Dynamics (ST, EMT) 20
Protection Coordination (OC, SZ, DI, PSA) 22
Protection Data Manager (PDM) 31
Contingency Analysis (CA) 32
CIM Import and Export 32
Automation 35
Changes to the Automation Functions (API) 35
Automation of the Calculation Methods 35
Automation of the Database Manager 39
Documentation 39
Network Model (Database) 40
Changes to the Network Model Schema (Data Model) 40
Information Technology **49**
Licensing 49
Installation 49
System Requirements 49

PSS®SINCAL

Example Networks

This product version delivers new and revised example networks. These example networks demonstrate the basic use of the product features and can be used to test the various calculation modules.

Network model	Description
Example Ele2	In this version, the harmonic contingency function has been rebranded and revised. The example has been updated to illustrate the changes.
Example IBSA	This new example illustrates the newly introduced feature impedance-based stability analysis for evaluating the harmonic stability of inverter-based systems. By defining a frequency-dependent inverter impedance characteristic curve, it is possible to use the Nyquist criterion to determine which of the defined operating configurations are or will become unstable. Since impedance-based stability analysis is a screening method, the result should be verified by a time-domain simulation (EMT).
Example Imp Excel	Updated example for running the load allocation in Module LA and importing MRIDs.
Example LV	Updated example for load allocation (Module LA) in a low-voltage network with a focus on measured current values.
Example MV	Updated example for performing load allocation (Module LA) in a medium-voltage network, focusing on measured power values and the automatic calculation of power values for the loads (load allocation) for all active operating points, as well as the automatic application of the results to new (or existing) operating points.
Example MV 2	Updated example demonstrating the ability to import growth scenarios for loads and generation (via Excel import) for long-term network planning with network development (Module LD).
Example NMM	GIS data has been adjusted such that the entire workflow can be carried out without using intermediate results. Moreover, the example has been extended and revised to showcase all functions for automatically and manually drawing network graphics in additional geographical and schematic views. Finally, the CGMES export has been updated to reflect the enhancements in the current version.
Example TSDI	Updated example for performing load allocation (Module LA) in time-series analysis (for each time step) and integrating measurement data using the time-series data interface (TSDI).

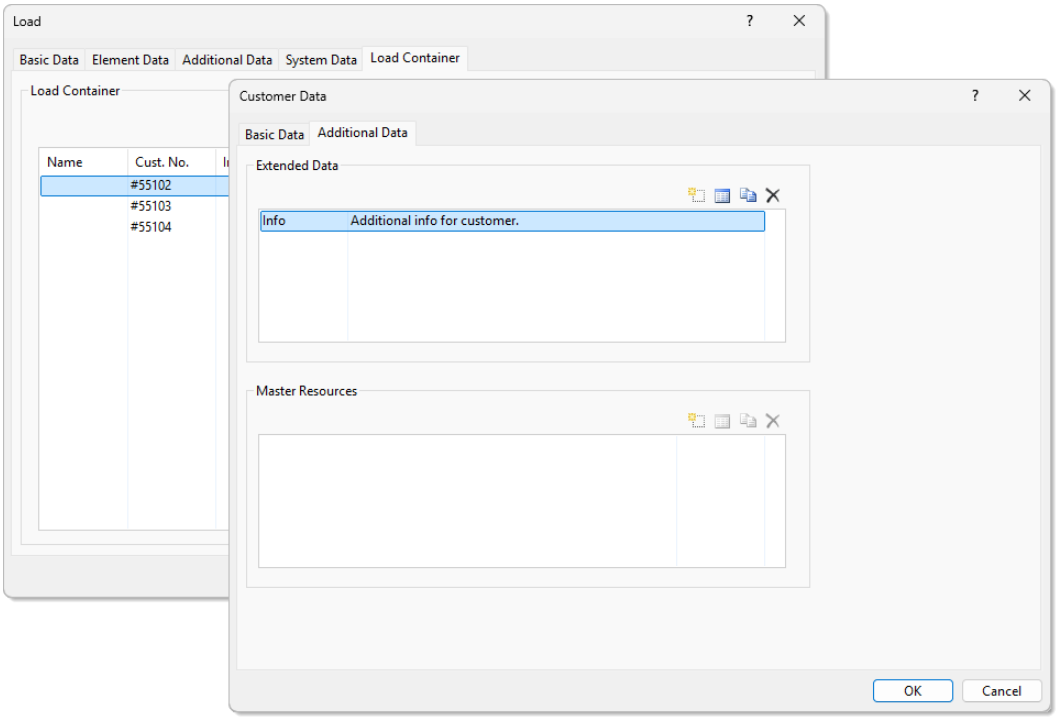
Graphical User Interface

General Extensions

Extended Data

The handling of extended data in PSS SINCAL network models has been unified and expanded. User-defined text and numerical attributes assigned to network elements are now stored centrally in the network database. These attributes serve as supplementary information and do not influence calculation results.

In the user interface, **Extended Data** can still be viewed and edited for nodes, network elements, and substations. With this version, this functionality is also available for customer data within load containers.



The storage of extended data has been fundamentally redesigned. Instead of separate tables for nodes (NodeExt), network elements (ElementExt), and substations (EcoStationExt), extended data is now stored centrally in the new **ExtendedData** table. The structure is based on the **MasterResource** table and allows any additional information to be assigned to each record in the network model. Records are uniquely identified using the **RowType** and **Row_ID** fields.

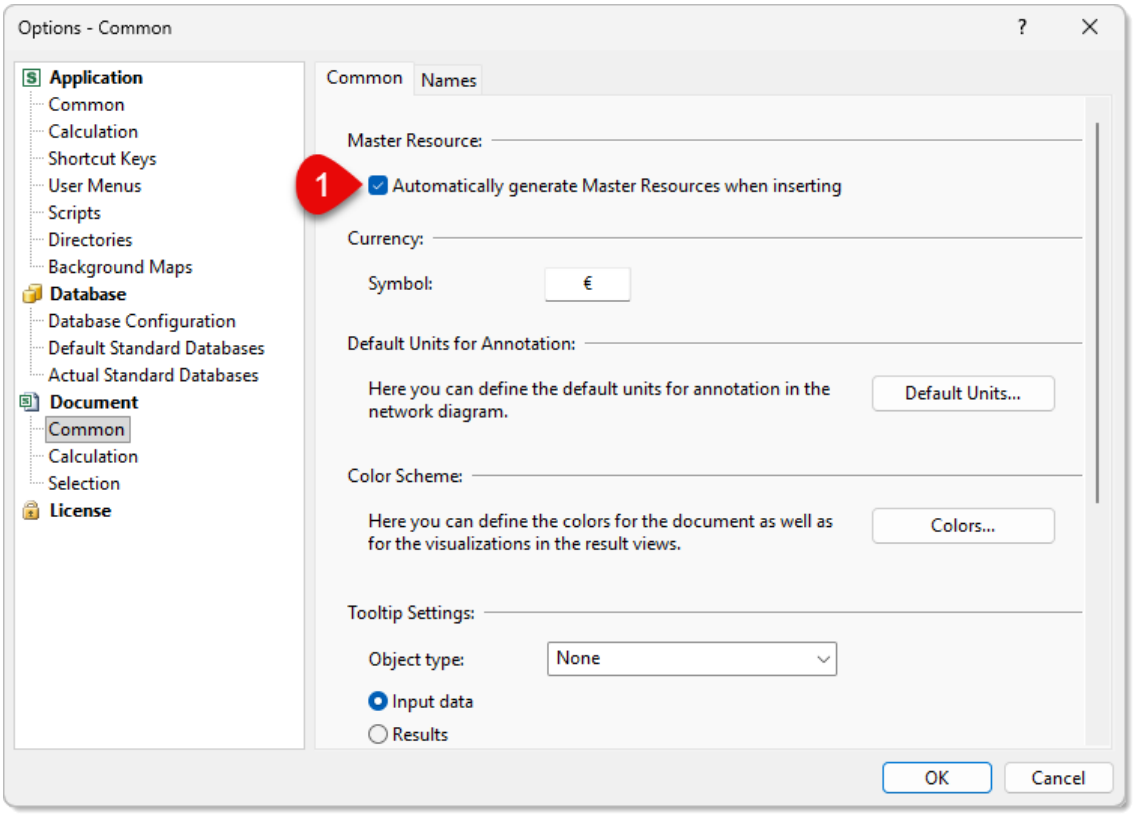
Attribute name	Data type	Short name	Unit	Std.	Description
ExtendedData_ID	Long Integer			0	Primary Key – Extended Data
Variant_ID	Long Integer			1	Secondary Key – Variant
Flag_Variant	Integer			1	Element of Current Variant 0: No 1: Yes
RowType	Integer			0	Row Type
Row_ID	Long Integer			0	Secondary Key – Data ID
Name	Text (8)			0	Name of Attribute
Flag_Type	Integer	Type		1	Type of Value 1: Number 2: Text
Value1	Double	Nval		0	Numeric Value
Value2	VarString	Tval		0	Text Value
Unit	Text (8)			0	Unit

When the network model of an older version is updated, all existing data is automatically transferred to **ExtendedData** without any data loss.

Automatic Generation of Master Resources

When nodes, network elements, substations, protection devices, and measuring devices are created in the user interface, unique master resources without field assignments can now be generated automatically. This allows objects to be uniquely identified via their master resource without relying on internal database IDs.

Automatic generation is by default inactive and can be activated in the **Options** dialog box under **Common Document Settings** (#1):



In calculation automation, the automatic generation of master resources can be activated using the "MasterResourceGeneration" parameter:

```
try:
    # Get database
    SimulationDB = Simulation.DB_EL()
    if SimulationDB is None:
        print("Error: Getting database object failed!")
        return

    # Open database for processing
    _iState = SimulationDB.OpenEx()

    # Get existing node and network level from it
    objNodeSS1C = SimulationDB.GetCommonObject("Node", "SS1-C")
    if not objNodeSS1C:
        print("Error: Unable to find node!")
        return

    lNetworkLevelID = objNodeSS1C.GetValue("VoltLevel_ID")
    lNetworkAreaID = objNodeSS1C.GetValue("Group_ID")

    # Preselect network level and network area for new created elements
    SimulationDB.SetParameter("NetworkLevel", lNetworkLevelID)
    SimulationDB.SetParameter("NetworkArea", lNetworkAreaID)
    SimulationDB.SetParameter("MasterResourceGeneration", True)

    print("Adding new node SS1-D")

    # Create a new node
    objNodeSS1D = SimulationDB.CreateNode("SS1-D")
    if not SUCCEEDED(objNodeSS1D):
        return
```

```
finally:
        SimulationDB = None
```

In addition, the user interface now includes a function for checking master resources (**Tools – Check – Check Master Resources**), which can be used to verify whether a master resource has been assigned to all nodes, network elements, substations, protection devices, and measuring devices.

Update of Global Protection Databases & Equipment Databases

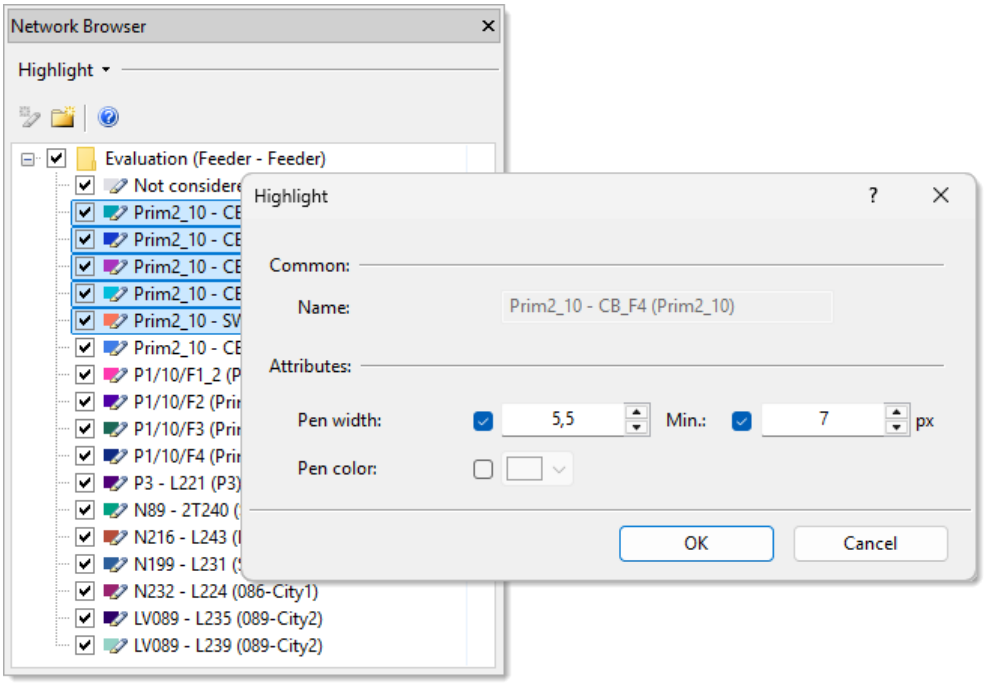
PSS SINCAL contains extensive protection and equipment data, which is automatically installed with each new product version. For version 23.0, these global databases are stored in the "%APPDATA%\Siemens\SINCAL\23.0\Database" directory.

Until now, these databases had to be updated manually following changes caused by product updates. In the future, installing a product update will automatically replace the global protection and equipment databases with the latest versions.

Network Graphic

Highlight

The network browser includes a feature for highlighting network graphic elements by color, either in groups or individually. If you select multiple highlights from the displayed list and open the **Edit** dialog box via the pop-up menu, you can modify all selected entries at once.



Another new feature is that the pen width of highlights can now be specified with decimal places. This is particularly helpful for large network models and high zoom levels.

Documentation

Excel Import

The documentation for the Excel import has been thoroughly revised and made easier to follow. It can be found in the **System Manual** under the chapter **Importing from Excel**. The "Example Imp Excel" application example demonstrates the step-by-step import of tabular data into a network model, as well as how to update the model using the Excel import function with the appropriate configuration.

Network Graphic in Additional Views

The documentation on auto-drawing or manual drawing of the network graphic in additional schematic or geographical network graphic views has also been redesigned. The System Manual explains the functionality and creation of the network graphic in the chapter **Network Graphic Views**. The "Example NMM" application example demonstrates the functions in practice.

Electrical Networks

Harmonic Stability (OB)

High- and, more often, medium-voltage network operators face the challenge of ensuring grid stability across a wide frequency spectrum when integrating converter-based systems, such as PV systems, battery energy storage systems (BESS), and STATCOMs. In addition to detailed and complex time-domain simulations using manufacturer models, impedance-based methods are easier to apply and to evaluate harmonic stability, as they can be performed in the frequency domain without the need for detailed system models. In fact, transmission and distribution system operators are increasingly requesting these screening methods when processing connection applications as proof of compliance with their requirements regarding harmonic stability.

To meet these requirements, PSS SINCAL now offers the new **Harmonic Stability** module.

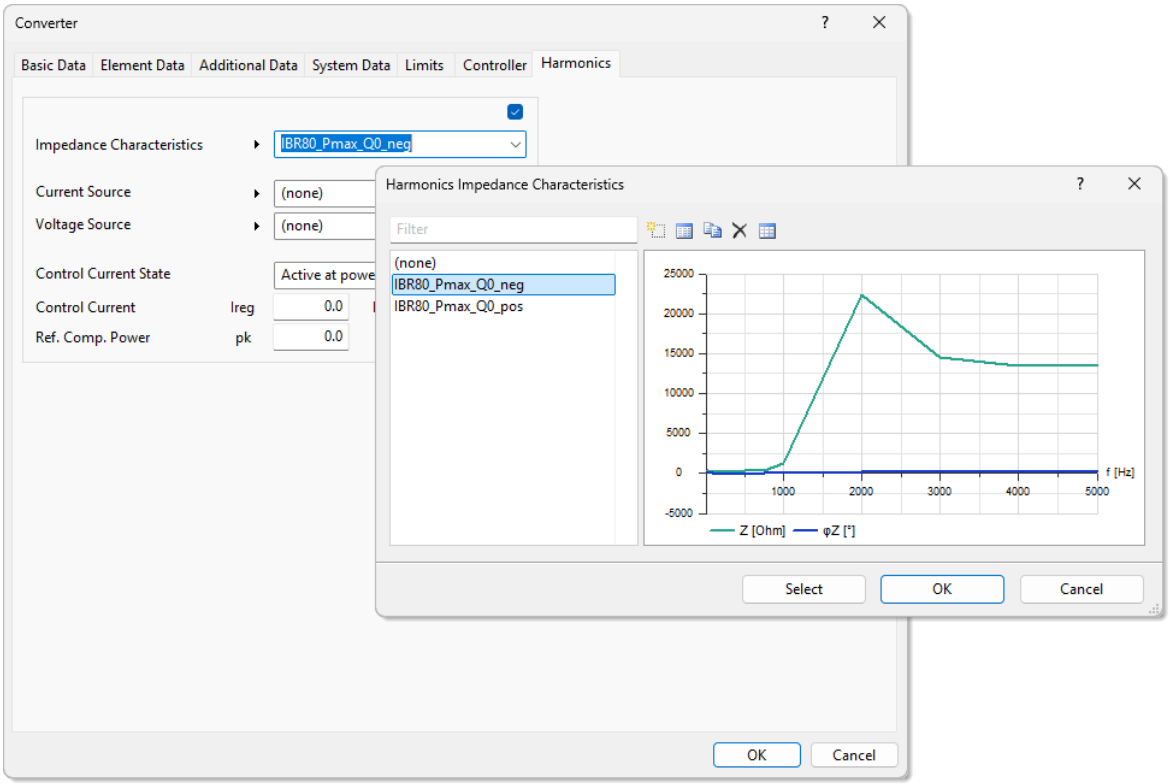
The basic concept of harmonic stability involves comparing the input impedance of the network to which a converter is connected to with the impedance of the converter itself. It is known from the literature that an intersection of the amplitudes of both impedances indicates a potential instability if the difference between the two phases approaches 180°. This condition can be verified using Bode or Nyquist diagrams.

To do this, the frequency-dependent converter impedance is compared with the network impedance measured at the connection point across the same frequency spectrum, typically for a predetermined number of switch states. This procedure requires that both subsystems – the converter and the network – be stable on their own.

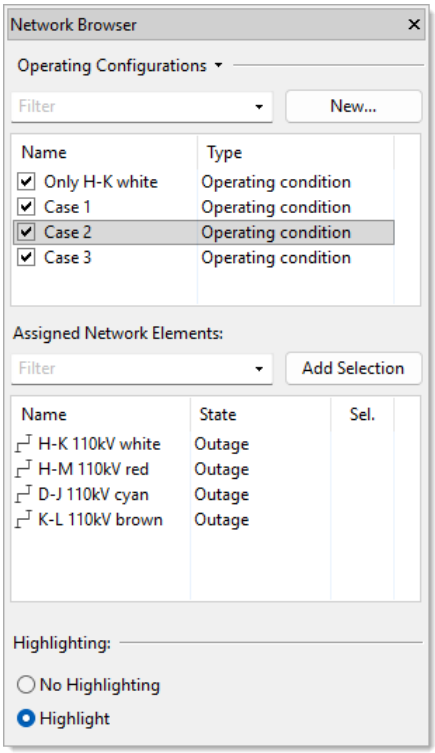
It is important to note that impedance-based methods are effective, but they are merely screening methods suitable for identifying potentially unstable network configurations. The results should always be verified using detailed time-domain simulations.

Using the New Calculation Module

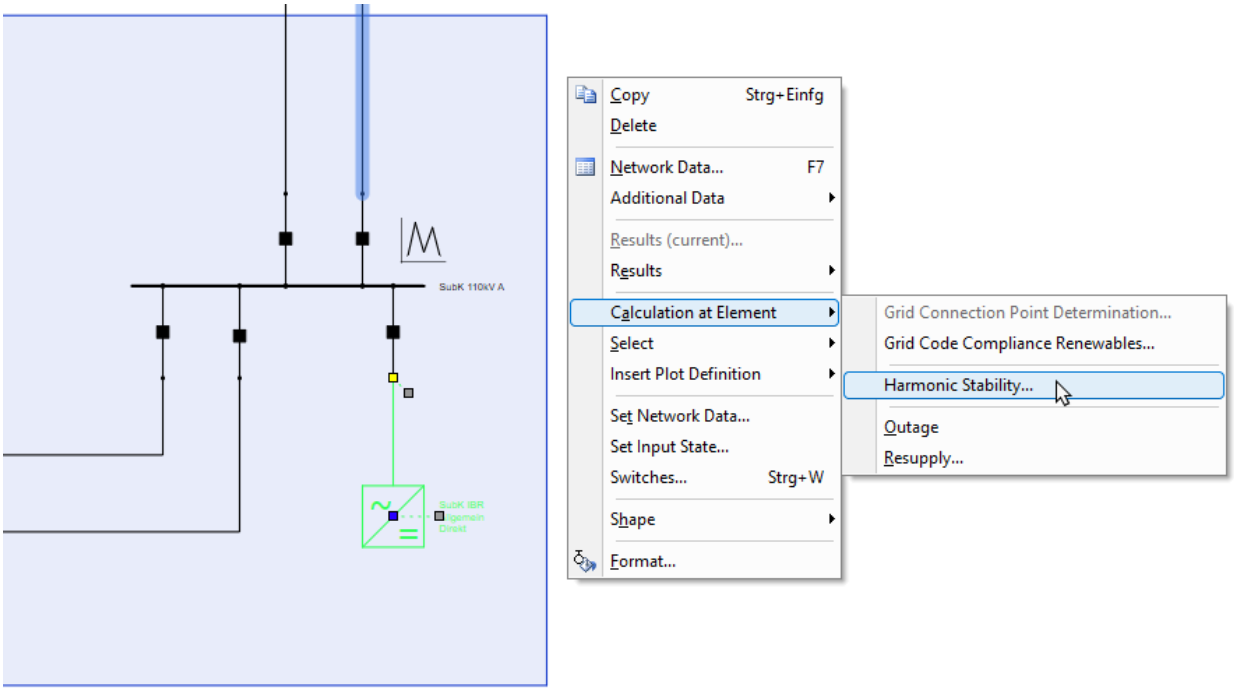
For the calculation, an impedance characteristic curve must be assigned to the converter, and the network model must be appropriately configured for harmonic analysis.



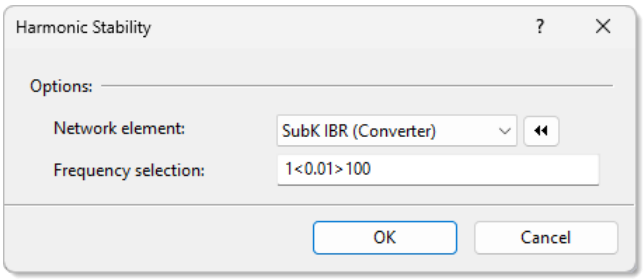
The network configurations (switch states) to be tested must also be defined. For this purpose, the **Operating Configurations** are now available in the network browser. These can be used to open and close switches and to simulate network contingency scenarios.



The new calculation module can then be launched directly from the pop-up menu on the converter by selecting the menu item **Calculation at Element – Harmonic Stability**.

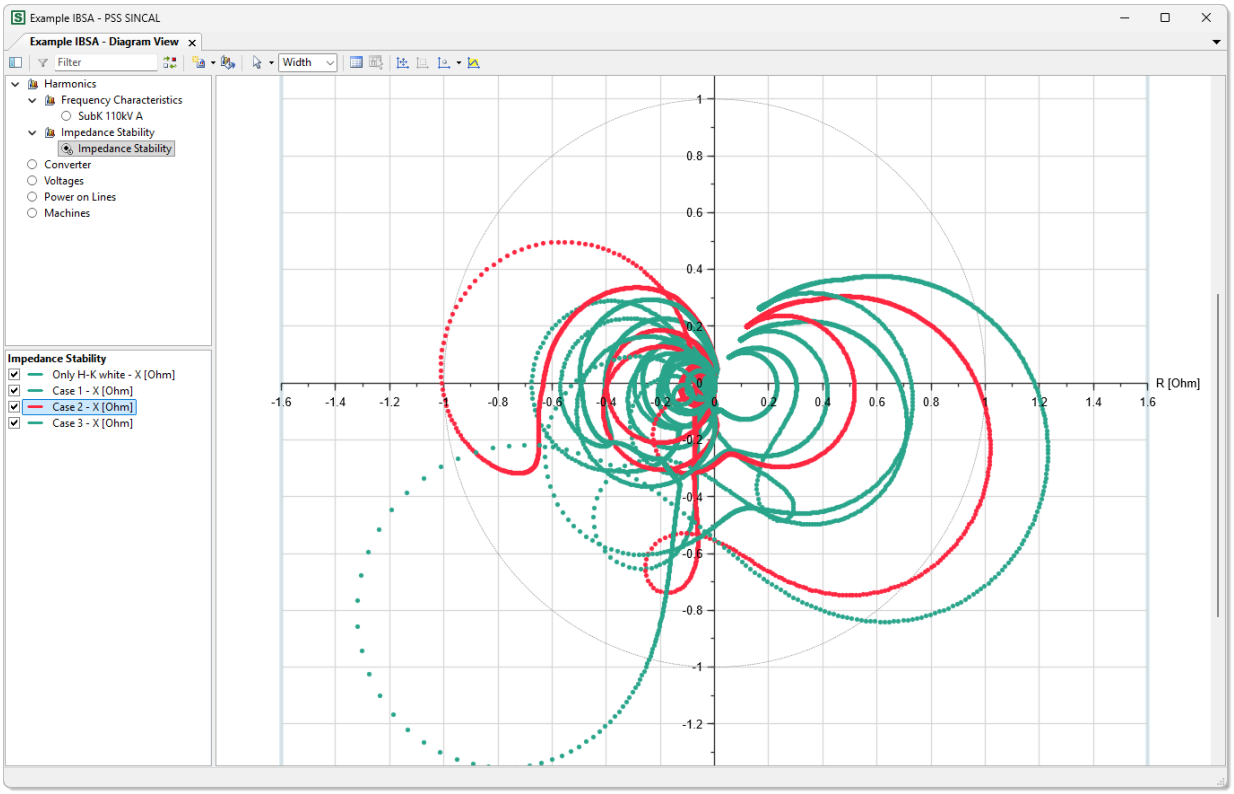


This opens a dialog box in which the frequency range to be analyzed is specified using harmonic numbers with a freely definable step size.

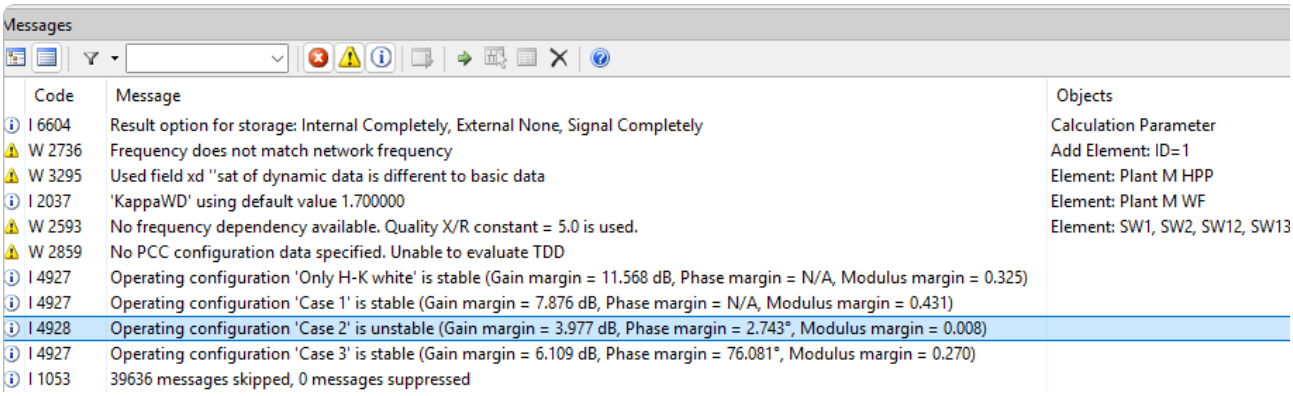


Harmonic calculations are performed for all active operating configurations within the defined frequency range. Network stability is then evaluated using the Nyquist criterion. To do this, the frequency response of the network without the converter is first determined and then compared with the frequency-dependent impedance of the converter at the same frequencies. An intersection of the amplitude responses indicates a potential instability.

The results are presented in diagrams and messages.



In the example shown, the "Case 2" operating configuration exhibits an unstable state. Therefore, the corresponding locus curve is displayed in red in the analyzed frequency range. In addition, the information message "I 4928" appears, indicating the unstable operating configuration. The messages "I 4927", on the other hand, indicate stable operating configurations.


Messages

Code	Message	Objects
I 6604	Result option for storage: Internal Completely, External None, Signal Completely	Calculation Parameter
W 2736	Frequency does not match network frequency	Add Element: ID=1
W 3295	Used field xd ''sat of dynamic data is different to basic data	Element: Plant M HPP
I 2037	'KappaWD' using default value 1.700000	Element: Plant M WF
W 2593	No frequency dependency available. Quality X/R constant = 5.0 is used.	Element: SW1, SW2, SW12, SW13
W 2859	No PCC configuration data specified. Unable to evaluate TDD	
I 4927	Operating configuration 'Only H-K white' is stable (Gain margin = 11.568 dB, Phase margin = N/A, Modulus margin = 0.325)	
I 4927	Operating configuration 'Case 1' is stable (Gain margin = 7.876 dB, Phase margin = N/A, Modulus margin = 0.431)	
I 4928	Operating configuration 'Case 2' is unstable (Gain margin = 3.977 dB, Phase margin = 2.743°, Modulus margin = 0.008)	
I 4927	Operating configuration 'Case 3' is stable (Gain margin = 6.109 dB, Phase margin = 76.081°, Modulus margin = 0.270)	
I 1053	39636 messages skipped, 0 messages suppressed	

The messages list the following safety distances:

- Modulus margin:
  The smallest distance that the curve reaches from the critical point (-1+j0) across all frequencies.
- Gain margin:
  Indicates, at the point where the curve crosses the -180° phase, how much additional gain the system can still handle before it clips. N/A is displayed if the curve never reaches this crossing point.

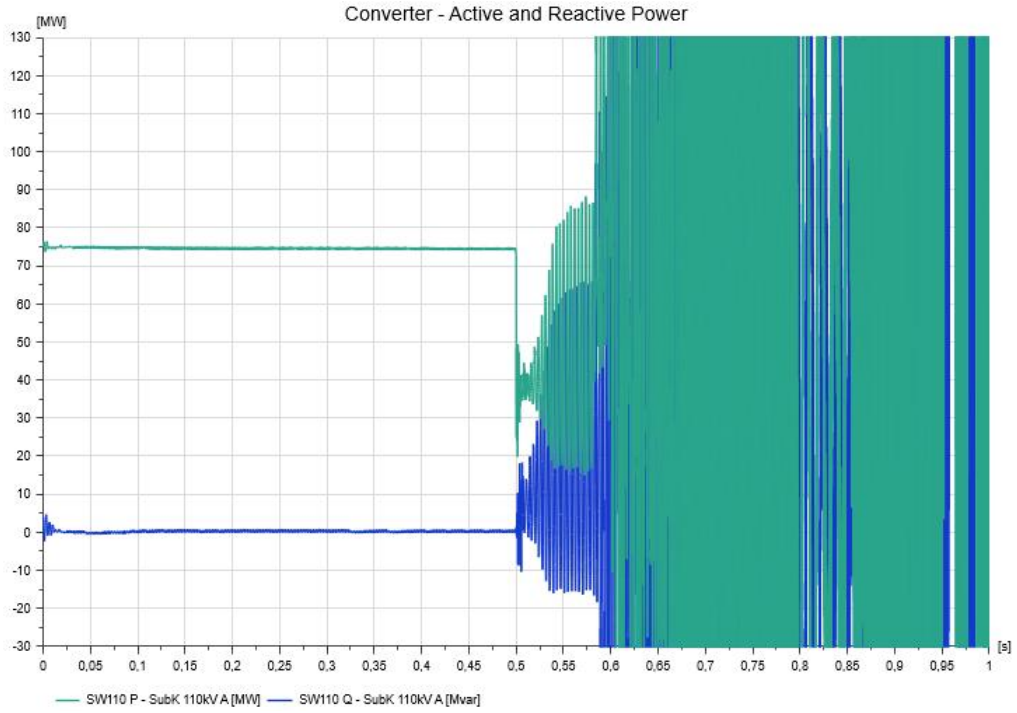
- Phase margin:
  Indicates, at the point where the curve intersects the unit circle, how much additional phase delay the system can still tolerate.

Comparison with Dynamic Simulation Results

To validate the results obtained and to evaluate measures for preventing unstable behavior in the systems under investigation, dynamic simulations are performed at the same operating point and with the same configurations as those used for harmonic stability analysis. In other words, the operating configurations are modeled for the dynamic simulation by **opening switches**, and the contingency for the "H-K 110kV white" line is simulated by a fault.

Based on the time-domain analysis using dynamic simulation, it can be seen that "Case 1" and "Case 3" are stable operating configurations, while "Case 2" leads to instability in the converter following the fault on the 110 kV white H-K line.

The following diagram shows the active and reactive power curves at the converter's installation location (PCC) for the "Case 2" operating configuration:



Application Example

The new application example "Example IBSA" is available for harmonic stability. Using a 400/110-kV network as an example, it demonstrates the parameterization and application of the new calculation module, as well as the workflow for further analysis of unstable operating configurations using dynamic simulation.

Rebranding of Harmonic Variations as Harmonics Scenario Analysis

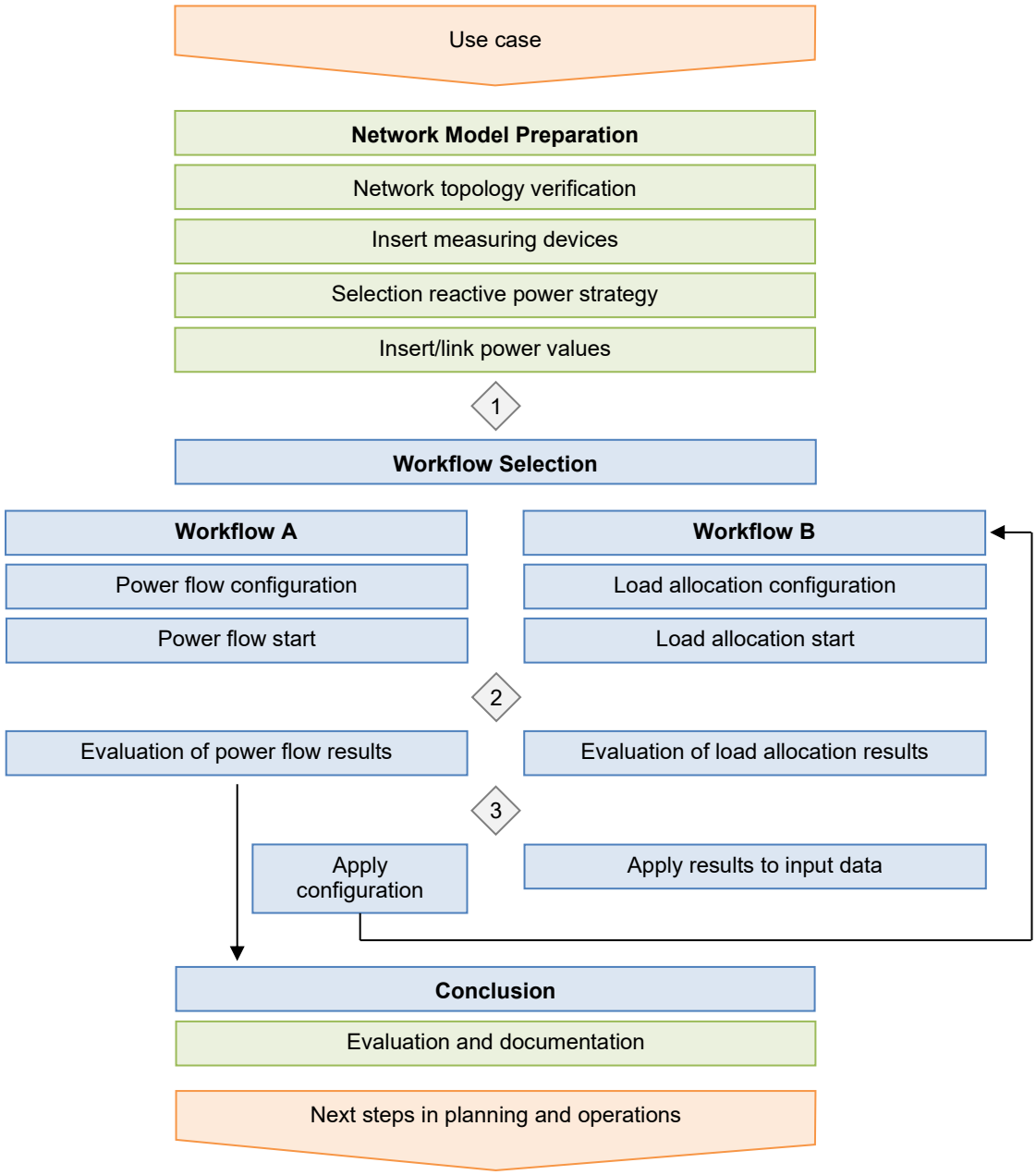
In this version, the previous **Harmonic Contingency** feature has been revised and renamed to **Harmonic Scenario Analysis**. The underlying calculation logic remains unchanged, and the GUI and API have been adapted to the new concept.

The definition of scenarios (formerly known as variations) has been standardized. Similar to impedance-based stability analysis, scenarios are now defined based on **Operating Configurations** that describe the respective network states. This enables the modeling of different switching states as well as network element outage scenarios.

Load Allocation (LA)

The **Load Allocation** module has been revised. The new version supports two workflows for load allocation based on measured values: automatic execution at each time step of a time-series simulation or for individual "snapshots" (**Workflow A**), as well as execution as a module (**Workflow B**), whose results can be applied to new or existing operating points.

The following figure provides a comprehensive overview of the workflow. After the network model has been prepared in the same way, the workflow then splits into two paths for A and B. Workflow A can be executed to prepare for workflow B.



Workflow A – Power Flow in Load Allocation Mode

The power flow can be configured so that a load allocation is also performed automatically (in every power flow calculation).

In this mode, load allocation is available in every module that uses power flow calculation.

The following modules can be used in combination:

- Direct adjustment of load power values in the feeder-based evaluation (Module Feeder)
- Direct adjustment of load power values in each individual power flow of the operational and time-series calculation (Module LP)

Workflow B – Load Allocation as a Module for Adjusting Load Modeling

The load allocation module uses the power flow described above internally and provides additional functionality for applying power values to the input data of the loads. (See **Workflow B**).

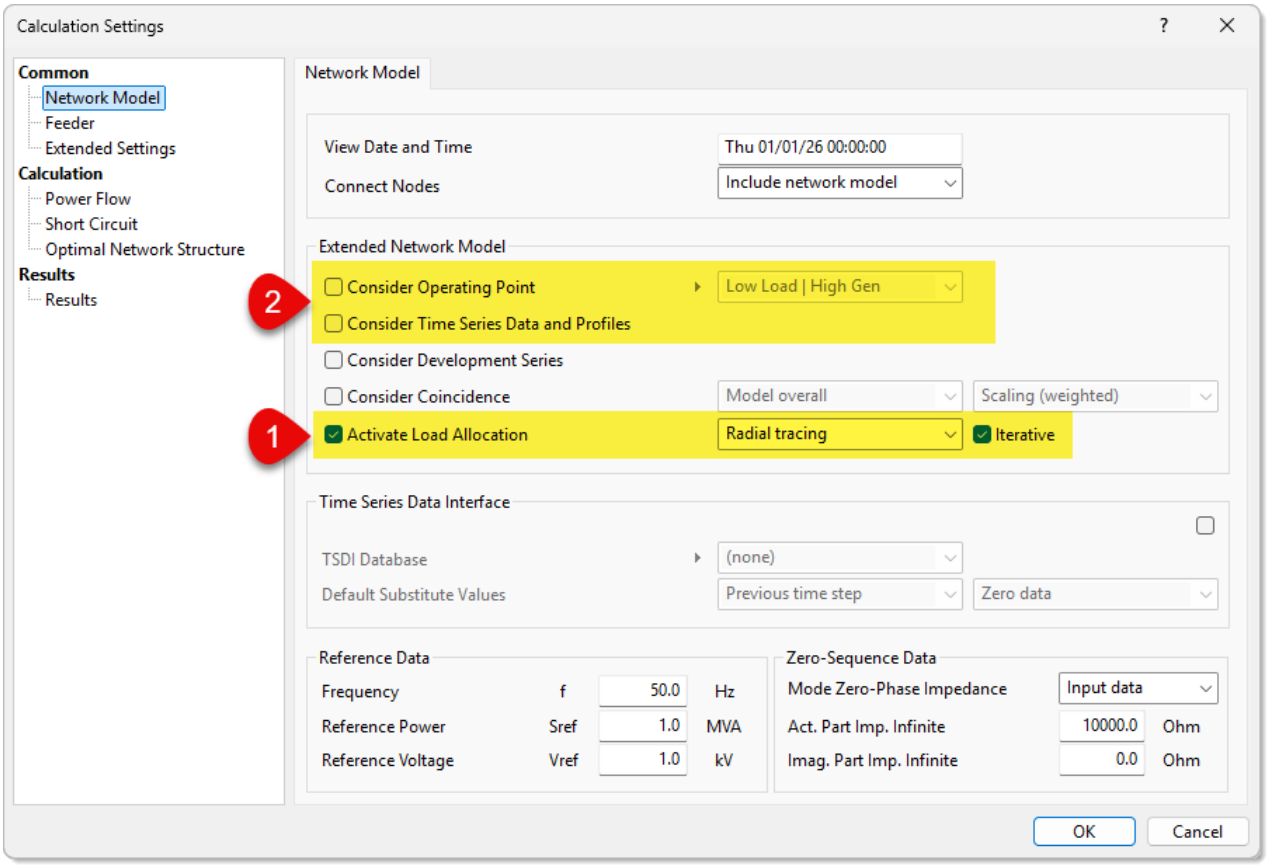
The following modules can be used in combination:

- The measured values can be set up directly as operating points and then used as reference when performing load allocation.
- The measured values can be integrated as time-series data (via Module TSDI) or as relative time-series data (profiles) using the Time-Series Calculation (Module LP), with the goal of deriving operating points for selected time points. This allows, following an initial time-series calculation and evaluation of the critical time steps, the computational effort for further analyses to be specifically reduced to selected states without a temporal sequence by converting the data into operating points.
- If the measured values are not up to date, factors can be applied to the measured values by assigning development series from the Network Development (Module ND) to the measuring devices in order to model measurement inaccuracies or worst-case scenarios.
- Creating operating points for the loads based on the determined power values prepares them for further use in the Operating Point Calculation (Module LP) and for combined use with the Network Development (Module LD).
- The creation of respective underlying base cases for the modules for Contingency Analysis (Module CA), Hosting Capacity (Module ICA) calculation, and Optimal Branching (Module OT), in which the network topology or configuration is changed during the procedure so that the power values are applied from the initial state and are not to be changed thereafter.

Using Workflow A – Power Flow in Load Allocation Mode

In Workflow A, the power flow is performed with active load allocation.

To do this, select the **Activate Load Allocation** option (#1) in the Calculate – Settings – Network Model dialog box. Optionally, you can also activate the consideration of an operating point or a time series here (#2).

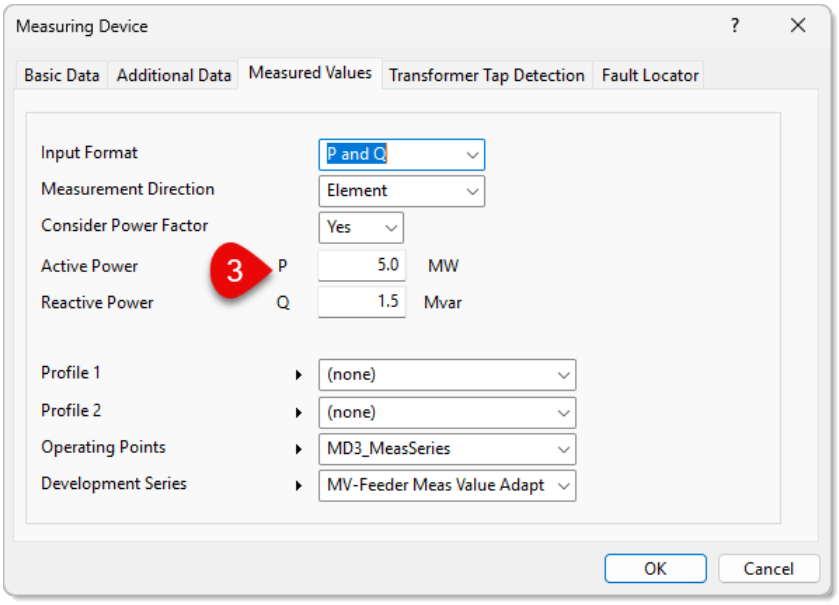


In the selection field for the **Activate Load Allocation** option, you can configure the target structure or the mode for assigning loads to the measuring devices for the subsequent load allocation:

- **Radial tracing**:
  A radial network topology is expected in the measured areas. Each participating load can be topologically and uniquely assigned to a measuring device.
- **Mesh tracing**:
  For meshed network models in which participating loads can be assigned to multiple measuring devices, the loads are assigned based on the direction of active power flow.

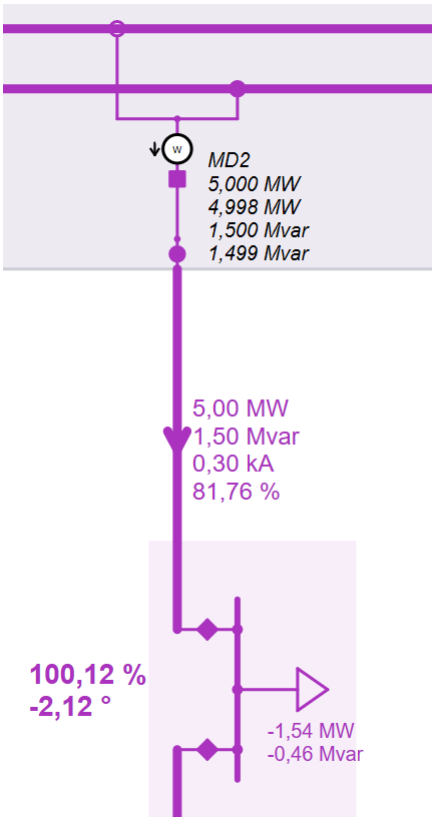
The **Iterative** option determines whether the load allocation is performed iteratively or by scaling the loads once. The iterative calculation provides more accurate results because it considers nonlinear transmission losses and power losses.

The functionality of workflow A can be illustrated using the "Example MV" application example. For the "MD2" measuring device, the **measured values** P = 5.0 MW and Q = 1.5 Mvar are entered (#3):

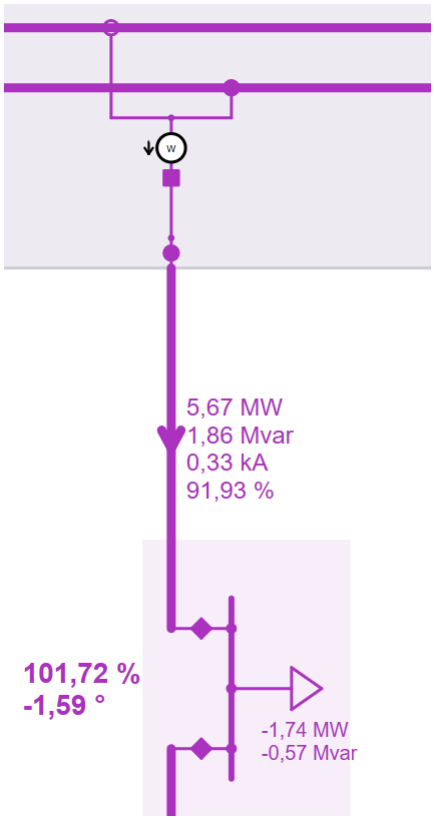


When the power flow is started via **Calculate – Power Flow**, the load allocation is performed automatically. If an operating point is selected, its data is used for measured values and network elements instead of the base values.

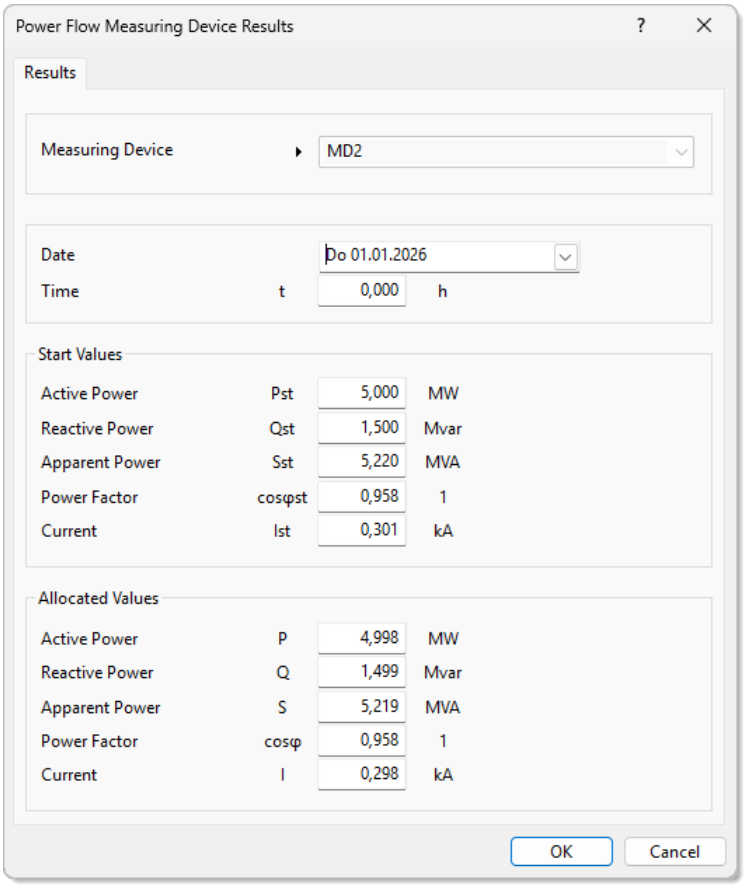
Power flow with active load allocation:



Power flow without load allocation (for comparison):



The image on the left shows the power flow results with load allocation. The load values were adjusted iteratively until the specified measured values (start values) were achieved within the permitted deviation for power flow accuracy. The measuring device also provides additional results that document the specified measured value and the power flow result. These can be displayed in the network graphic and shown in the result dialog box.

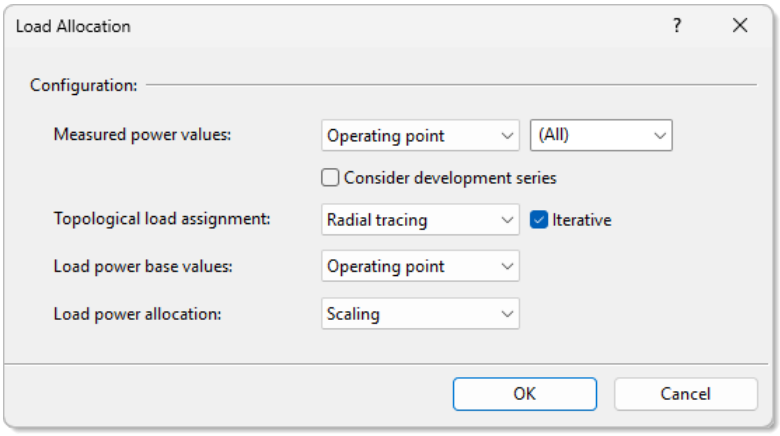


Using Workflow B – Load Allocation as a Module for Adjusting Load Modeling

In Workflow B, the calculated power values are first made available as results, with the goal of subsequently updating the input data for the loads with these power values.

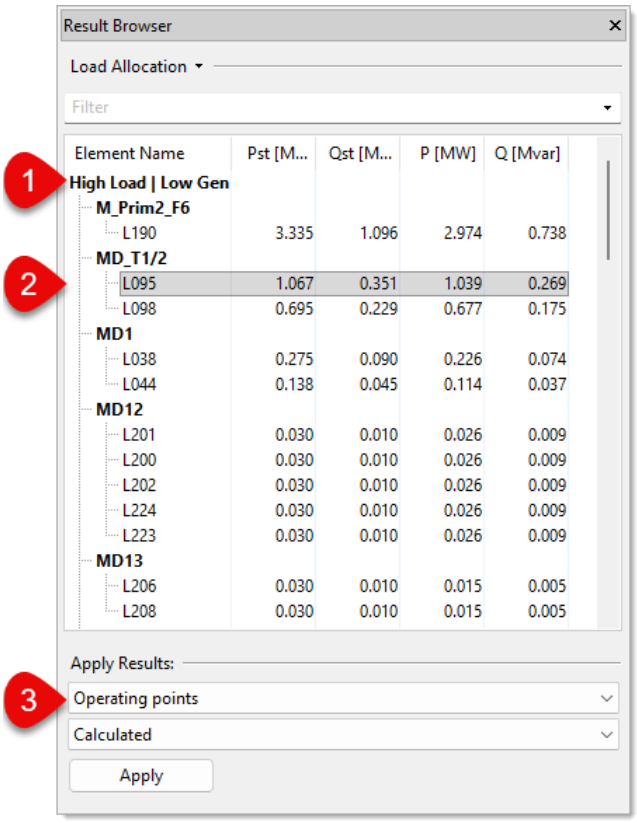
In the following example, the load allocation is performed for all operating points defined in the network model. The function can be tested using the "Example MV" application example.

The load allocation module (Workflow B) is started via the menu item **Calculate – Power Flow – Load Allocation**, which opens the configuration dialog box.



This example demonstrates how to perform a load allocation for all operating points in a single calculation run (afterward, the participating loads are to be automatically updated with the calculated power values). To do this, select the **Operating point** option in the **Measured power values** and **Load power base values** selection fields, and then start the calculation by clicking **OK**.

Once the calculation is complete, the results are available in the **Result Browser** under **Load Allocation**:



The results are grouped and displayed according to the calculated operating points (#1). For each operating point the active measuring devices are listed. For each measuring device, all loads that have been assigned during the module execution to the respective measured area and whose power values have been adjusted by the load allocation are displayed. In addition to the start values **Pst** and **Qst**, the resulting power values **P** and **Q** are displayed for each load.

The calculated power values were not automatically applied as new load data from the previous calculation. They can be applied explicitly using the **Apply** function, so that the results can be reviewed before and the application of the results can be configured.

The following options are available for applying the results (#3):

- **Input Data:**
  The calculated load values are automatically applied to the loads and replace the previous input data. Supported load input formats are retained. For unsupported formats, the values are converted automatically in the **P and Q** format, or – in the case of unbalanced results – in the **Pi and Qi – star** format.
- **Input Data – Factors:**
  The calculated powers are converted into scaling factors for the loads, to the extent possible. For input formats that do not support factor-based conversion, processing is performed according to the **Input Data** option.
- **Operating Points:**
  The calculation results are saved as operating points in the operating point series for the respective loads without changing the existing basic data of the loads. Users can specify how the operating points for the individual elements are either created or updated.

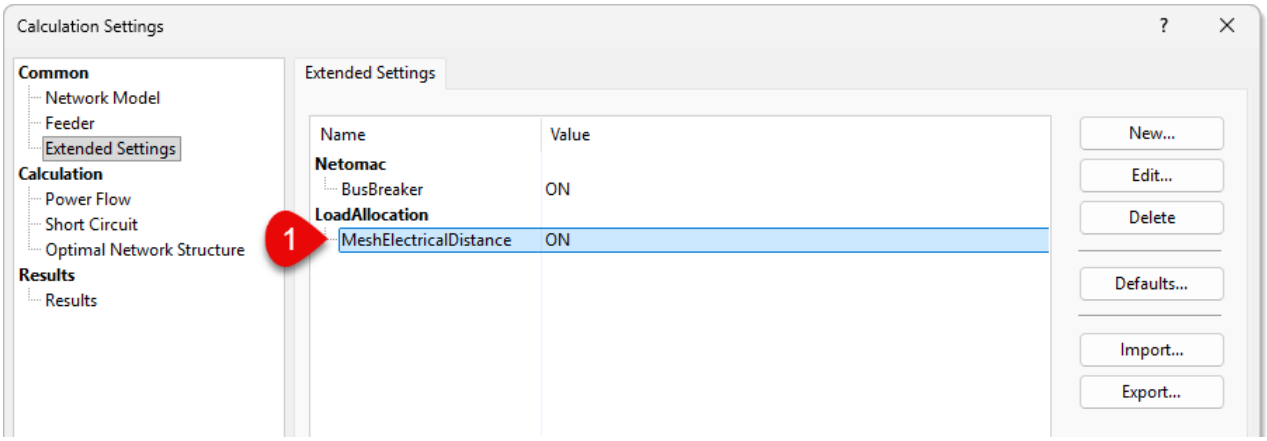
Enhancements to the Calculation Module

Load allocation has been extended to include an additional assignment mode to enable load allocation even in **meshed networks with "reverse feed" at the measuring devices** (e.g., a closed medium-voltage ring with feed-in power > consumption and, therefore, reverse feed in both fields at the substations).

Previously, the allocation based on power flow directions could not be performed unambiguously in certain network topologies and power flow scenarios, causing the calculation to terminate with an error message. The new feature now provides an alternative fallback mechanism.

In this fallback mode, loads are assigned to measuring devices based on **electrical distance**. For each network node, the electrical distance to the respective measuring device is determined based on the network impedances. The loads are then assigned to the "closest" measured area.

The new feature is not enabled by default. To enable it, use the extended calculation settings LoadAllocation.MeshElectricalDistance:= ON (#1):



This enhancement increases the applicability of load allocation in complex meshed network models and offers an alternative for network models with reverse feed at the measuring devices.

Extended Messages

The messages generated during load allocation have been comprehensively revised and extended. The goal is to improve transparency in model validation and to identify data and modeling issues more quickly.

Among the most important improvements are:

- Extended plausibility checks for measurement and load data before and during load allocation.
- More precise error and warning messages in the event of inconsistent input data or implausible load assignment.
- Improved diagnostic capabilities for convergence issues and insufficient initial data for the load scaling.
- New information regarding modeling approaches that are still supported but no longer follow the recommended approach.

In particular, users are now notified when separate measuring devices have been modeled for individual phases. This type of modeling was required in older versions but is no longer necessary. The new warning messages help users adapt existing models to the current recommendations.

In addition, the new plausibility checks detect situations in which the sum of the available loads is significantly smaller than the measurement power to be assigned. This allows unrealistic scaling factors to be identified early on and alerts users to potential modeling errors.

The revised reporting structure improves the traceability of calculation results and facilitates the analysis and quality assurance of network models.

Documentation

The documentation for the load allocation module has been comprehensively revised.

The following application examples provide a detailed illustration of how to configure and use the module for various use cases:

- Example MV:
  Load allocation for Workflow A and B in a medium-voltage network.
- Example LV:
  Load allocation in Workflow A in a low-voltage network.
- Example TSDI:
  Load allocation in Workflow A, when measured values are mapped directly via the time-series data interface (Module TSDI), thereby enabling direct integration of the measurement data into the network model.
- Example Imp Excel:
  Load allocation when measuring devices and measured values are imported into the network model via Excel import.

Dynamics (ST, EMT)

Models

The expansion of the model library for the excitation of synchronous machines from the previous version of PSS SINCAL continues. The following excitation and power system stabilizer models have been added in this version. Detailed information on their parameterization and use can be found in the **Models** manual.

Model	Change	Description
AC1A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC1C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC2A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC2C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC3A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC3C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC4A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC4C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC5A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC5C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC6A.xmac	New	AC Excitation Model according to IEEE 421.5-1992
AC6C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
AC7B.xmac	New	AC Excitation Model according to IEEE 421.5-2005
AC8B.xmac	New	AC Excitation Model according to IEEE 421.5-2005
AC8C.xmac	New	AC Excitation Model according to IEEE 421.5-2016
PSS1A.xmac	New	Power System Stabilizer according to IEEE 421.5-1992
PSS2A.xmac	New	Power System Stabilizer according to IEEE 421.5-1992
PSS2B.xmac	New	Power System Stabilizer according to IEEE 421.5-2005
PSS2C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
PSS3B.xmac	New	Power System Stabilizer according to IEEE 421.5-2005
PSS3C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
PSS4B.xmac	New	Power System Stabilizer according to IEEE 421.5-2005
PSS4C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
PSS5C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
PSS6C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
PSS7C.xmac	New	Power System Stabilizer according to IEEE 421.5-2016
ST1A.xmac	New	Static Excitation Model according to IEEE 421.5-1992
ST1C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST2A.xmac	New	Static Excitation Model according to IEEE 421.5-1992
ST2C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST3A.xmac	New	Static Excitation Model according to IEEE 421.5-1992
ST3C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST4B.xmac	New	Static Excitation Model according to IEEE 421.5-1992
ST4C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST5B.xmac	New	Static Excitation Model according to IEEE 421.5-2005
ST5C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST6B.xmac	New	Static Excitation Model according to IEEE 421.5-2005
ST6C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST7B.xmac	New	Static Excitation Model according to IEEE 421.5-2005
ST7C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST8C.xmac	New	Static Excitation Model according to IEEE 421.5-2016
ST9C.xmac	New	Static Excitation Model according to IEEE 421.5-2016

ST10C.xmac	New	Static Excitation Model according to IEEE 421.5-2016

The following models are obsolete and have been replaced by new models. They are no longer being developed, but remain available in the “Models\Deprecated” folder to ensure compatibility with existing network models. The listed models correspond to older implementations based on IEEE 421.5 and have been replaced by the models that have now been added. We plan to remove these models in the upcoming SINCAL version 25.0.

Model	Change	Description
DC4B.mac	Replacement	
ESAC1A.MAC	Replacement	
ESAC1A.xmac	Replacement	
ESAC2A_ascii.mac	Replacement	
ESAC2A.mac	Replacement	
ESAC2A.xmac	Replacement	
ESAC3A.mac	Replacement	
ESAC4A.xmac	Replacement	
ESAC5A.mac	Replacement	
ESAC5A.xmac	Replacement	
ESAC6A.mac	Replacement	
ESAC6A.xmac	Replacement	
ESAC8B.mac	Replacement	
ESAC8B.xmac	Replacement	
ESDC1A.mac	Replacement	
ESDC1A.xmac	Replacement	
ESDC2A.mac	Replacement	
ESDC2A.xmac	Replacement	
ESST1A.mac	Replacement	
ESST1A.xmac	Replacement	
ESST2A.mac	Replacement	
ESST2A.xmac	Replacement	
ESST3A.MAC	Replacement	
ESST3A.xmac	Replacement	
ESST4B.mac	Replacement	
ESST4B.xmac	Replacement	
EXAC1A.xmac	Replacement	
EXAC3.mac	Replacement	
ExcAC1A.mac	Replacement	
ExcAC2A.mac	Replacement	
ExcAC3A.mac	Replacement	
ExcAC4A.mac	Replacement	
ExcAC5A.xmac	Replacement	
ExcAC6A.xmac	Replacement	
ExcAC8B.xmac	Replacement	
ExcDC1A.xmac	Replacement	
ExcDC2A.xmac	Replacement	
ExcDC3A.xmac	Replacement	
ExcIEEEAC1A.mac	Replacement	
ExcIEEEAC2A.mac	Replacement	
ExcIEEEAC3A.mac	Replacement	
ExcIEEEAC4A.mac	Replacement	
ExcIEEEAC5A.mac	Replacement	

ExcIEEEAC6A.mac	Replacement	
ExcIEEEAC7B.mac	Replacement	
ExcIEEEAC8B.mac	Replacement	
ExcIEEEDC1A.mac	Replacement	
ExcIEEEDC2A.mac	Replacement	
ExcIEEEDC3A.mac	Replacement	
ExcIEEEDC4B.mac	Replacement	
ExcIEEEST1A.mac	Replacement	
ExcIEEEST2A.mac	Replacement	
ExcIEEEST3A.mac	Replacement	
ExcIEEEST4B.mac	Replacement	
ExcIEEEST5B.mac	Replacement	
ExcIEEEST6B.mac	Replacement	
ExcIEEEST7B.mac	Replacement	
ExcST3A.xmac	Replacement	
EXDC2.mac	Replacement	
EXDC2.xmac	Replacement	
EXST2A.xmac	Replacement	
PSS1A.mac	Replacement	
pss2a.mac	Replacement	
PSS2B.mac	Replacement	
PSS3B.mac	Replacement	
PSS4B.mac	Replacement	
PssIEEE1A.mac	Replacement	
PssIEEE2B.mac	Replacement	
PssIEEE3B.mac	Replacement	
PssIEEE4B.mac	Replacement	
ST5B.mac	Replacement	
ST6B.mac	Replacement	
ST7B.mac	Replacement	

Protection Coordination (OC, SZ, DI, PSA)

New and Modified Protection Devices

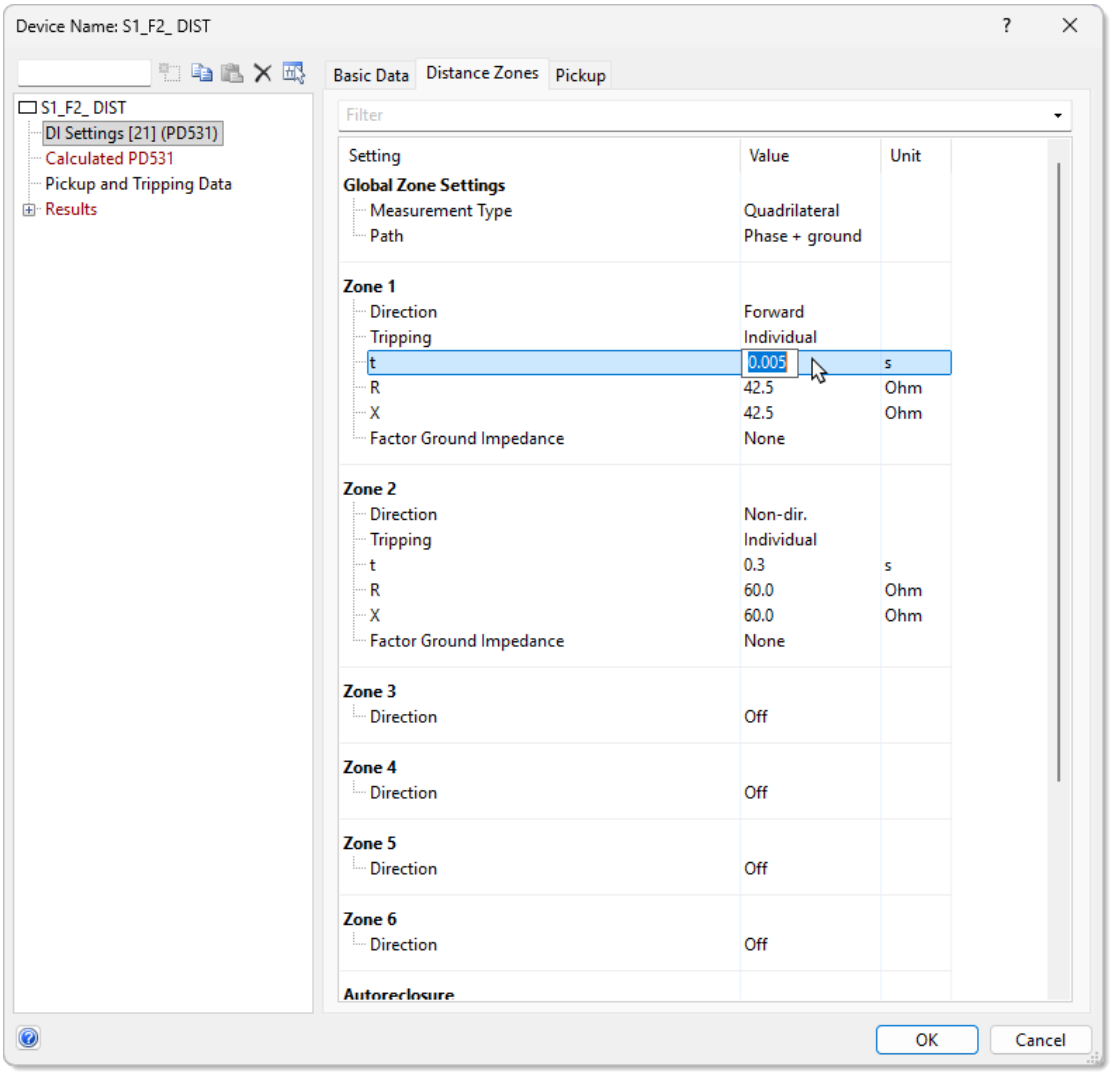
The library of protection devices has been expanded in this product version. The added devices are listed below. Detailed descriptions of the protection devices are available in the **Protection Coordination** and **Input Data** manuals.

Protection device	Description
7SX85	New protection device

Distance Protection: New Database and New Design for Tripping Functions

As part of the ongoing development of the distance protection models, the tripping functions were migrated to the new distance protection database. This enables the provision of additional device-specific information and lays an important foundation for the future.

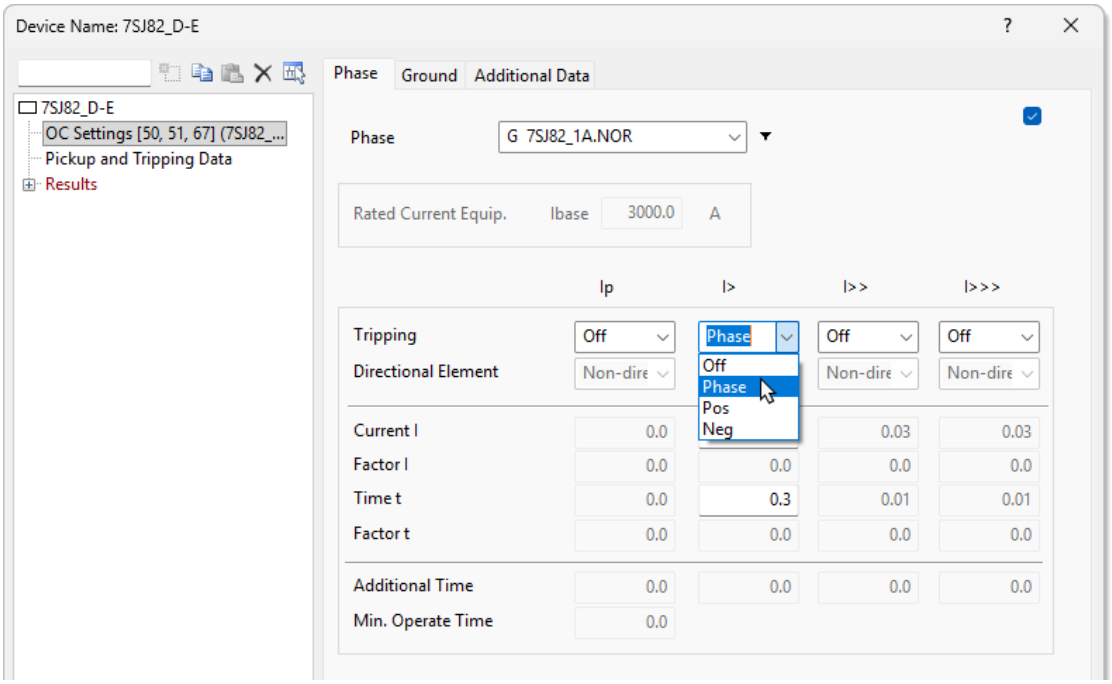
In addition, the dialog box for the **Distance Zones** has been redesigned to match the new design of the **Pickup** dialog box. This not only makes it easier to manage protection settings – since only the parameters that are actually in use are displayed – but also allows for a clearer presentation of device-specific functions.



The dialog boxes displaying the results of the distance protection setting calculation have also been revised. They now feature the new, clear design.

Overcurrent Time Protection: Device-Specific Tripping

For overcurrent time protection, the selection values under **Tripping** have been implemented on a device-specific basis. This means that, for all device series, only those options are now available that the real devices support.



The Lim functions for It characteristics are also no longer available, since the limit cannot be configured on real devices. Whether a characteristic is limited, is now implemented on a device- specific basis.

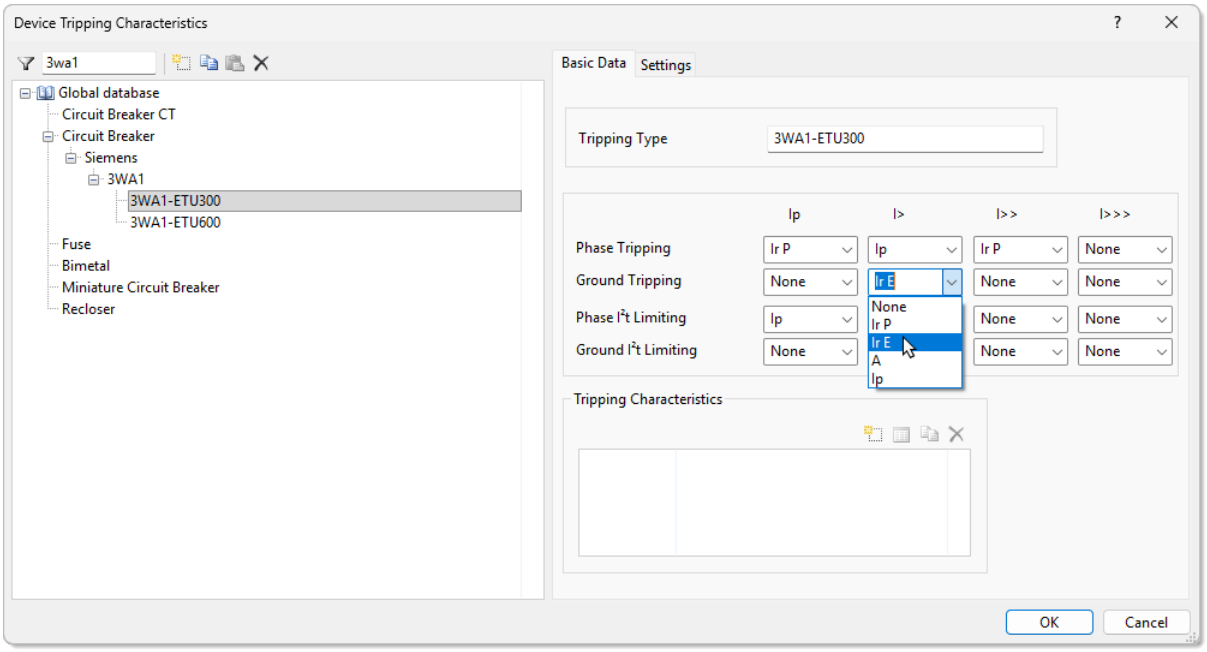
Machine Protection only within the Protection Zone

Voltage and frequency protection are used almost exclusively for decoupling protection functions, which must be treated fundamentally differently from conventional line protection. Previously, this differentiated treatment could be achieved using the option **Only consider machine protection within the protection area**. To ensure that decoupling protection is treated differently automatically in the future, this logic is now always active by default.

Rated Current per Phase, even with Ground as a Reference Value

The modeling options for OC protection devices have been extended. The rated phase current (In) can now also be used as a reference value in the event of an earth fault.

For example, the Siemens 3WA1 low-voltage circuit breaker has only one specification for the rated current. The ground settings must therefore also be referenced to this rated current.



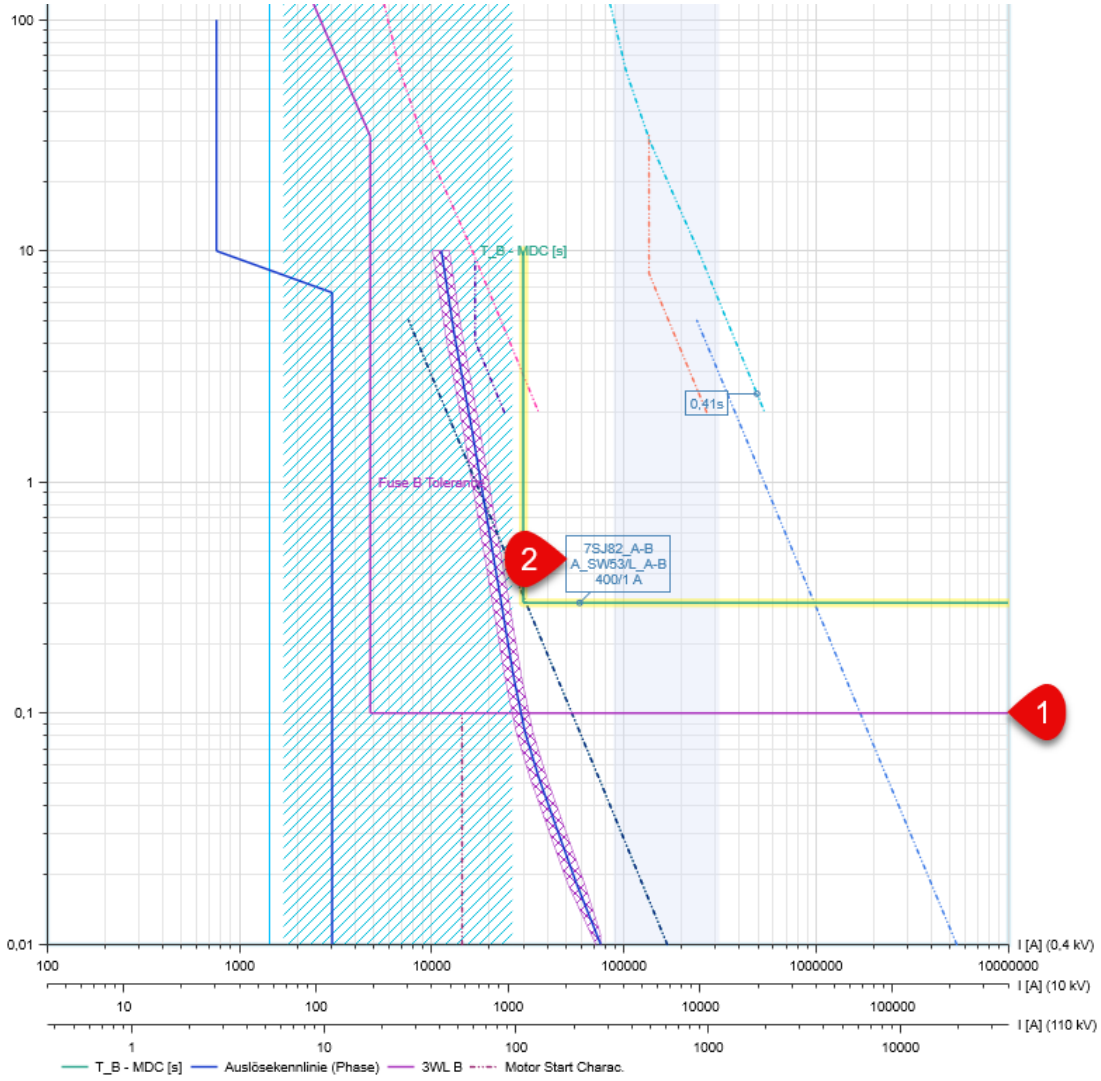
To this end, the selection of reference values in the tripping characteristics was adjusted:

- The previous entry **In** was relabeled: **In P** stands for the rated phase current, and **In E** for the rated ground current.
- For **Ground Tripping** and **Ground I²t Tripping**, both options are now available. The ground settings can thus refer either to the rated ground current (In E) or to the rated phase current (In P).
- For **Phase Tripping** and **Phase I²t Tripping**, only the phase rated current (In P) can be selected.

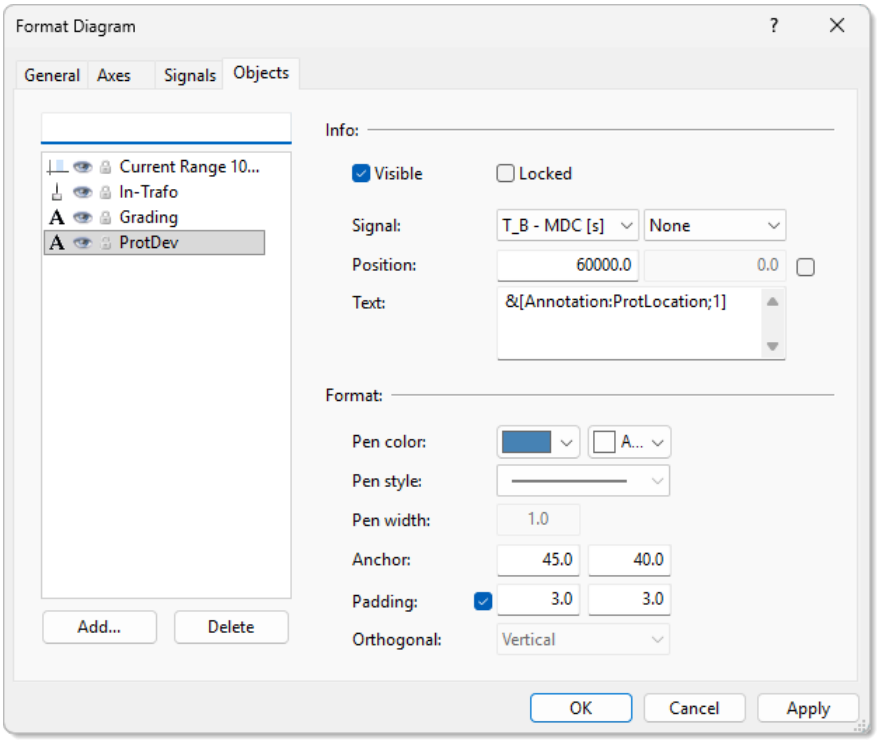
For existing data, the setting is adjusted automatically. Where **In** was previously selected, the setting is now **In P** for phase-steps and **In E** for ground-steps. The tripping behavior of existing protection devices remains unchanged.

Diagrams

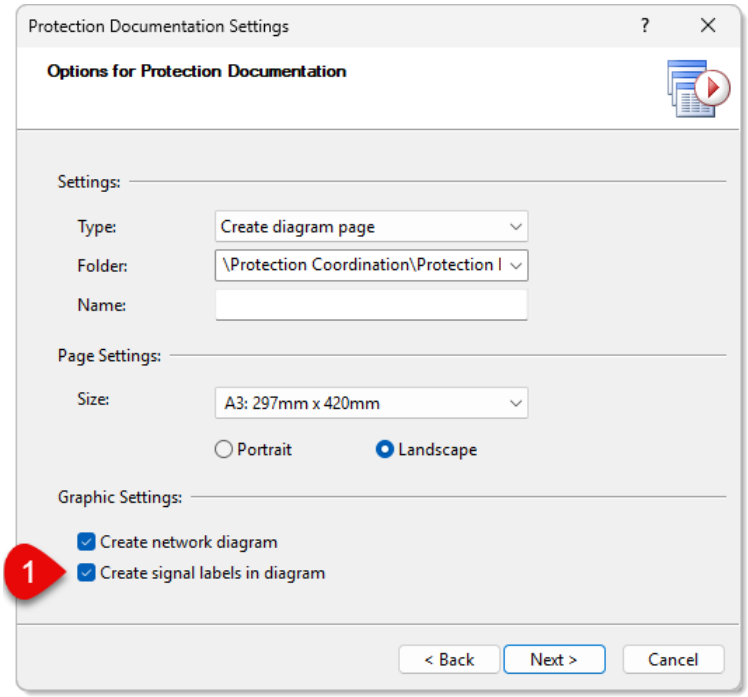
The display functions in the protection diagrams have been expanded. In the **Result Diagrams**, the display of the It characteristic curve is now extended with a horizontal line to the right margin of the diagram (#1). In the **Input Data Diagrams**, this display can also be activated, using an extended parameter.



Advanced labeling features are available for the labels in the diagrams (#2). A new token can now be used in the label to display the annotation from the network graphic. The label text is edited using the **Format Diagram** dialog box. Here, in the **Text** input field, you can use the new token **&[Annotation:ID]** to take advantage of the advanced annotation feature.



When creating protection documentation, you can use the **Create signal labels in diagram** option (#1) in the start dialog box to control whether the labels for all tripping characteristics should be created automatically.



Device-Specific Protection Report

The **Protection Settings** report has been extensively extended for distance protection devices. Like the editing dialogue boxes, it is now dynamically generated based on device-specific settings from the protection device database. This means that the settings displayed in the report are dynamically compiled using the new protection database. Device-specific labels are used, and only the available settings are displayed.

Creating new Protection Devices

The placement of protection devices on switches is no longer supported from a functional standpoint. This change is also consistent with the existing modeling concept of PSS SINCAL, according to which protection devices must not be positioned directly on switches.

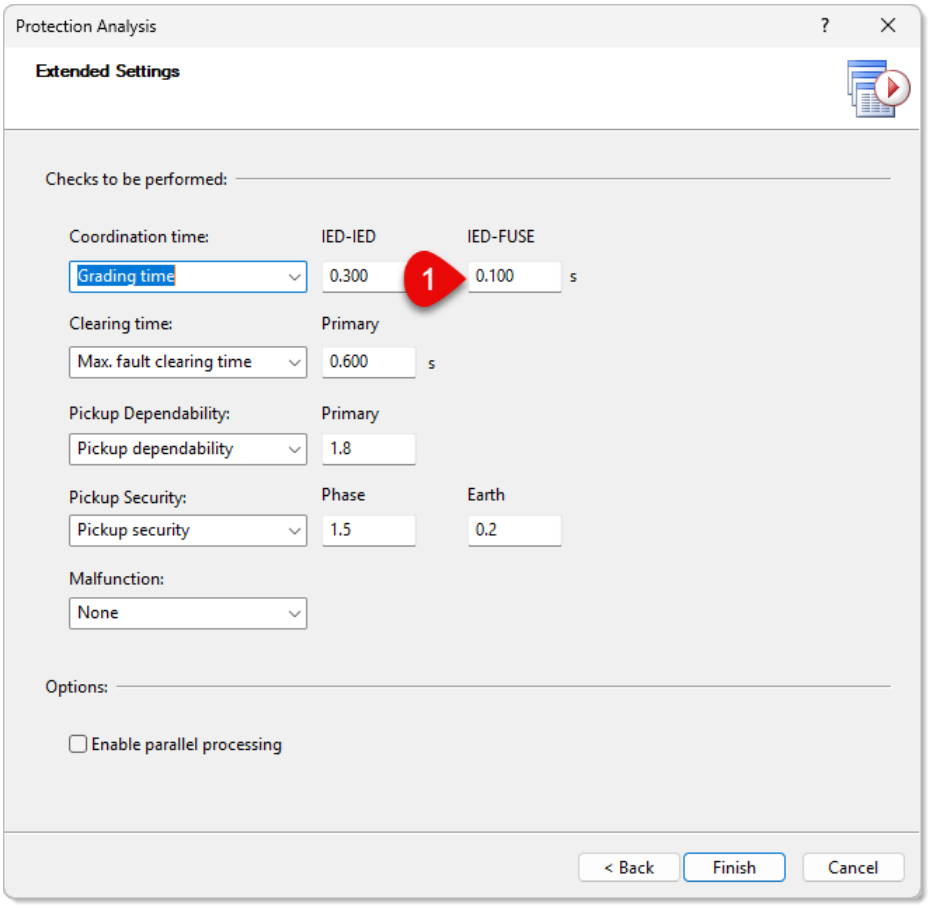
Placing devices on the line side of switches could result in a reversal of the protection direction. As a result, the affected protection devices were no longer correctly considered in protection analysis and protection route calculations.

Furthermore, from a physical standpoint, placement on the busbar side is incorrect and generally does not correspond to the actual system configuration. This is because faults in the circuit breakers are intended to be quickly cleared by the busbar protection. A current transformer on the busbar side would exclude the switch from the protection zone of the busbar protection and prevent rapid fault clearance.

To prevent incorrect modeling and the resulting invalid analysis results, it is no longer possible to place protection devices on switches. This applies to both manual modeling and the **Duplicate Protection Device** function.

A Dedicated Grading Time for OC/Fuse in the Protection Analysis

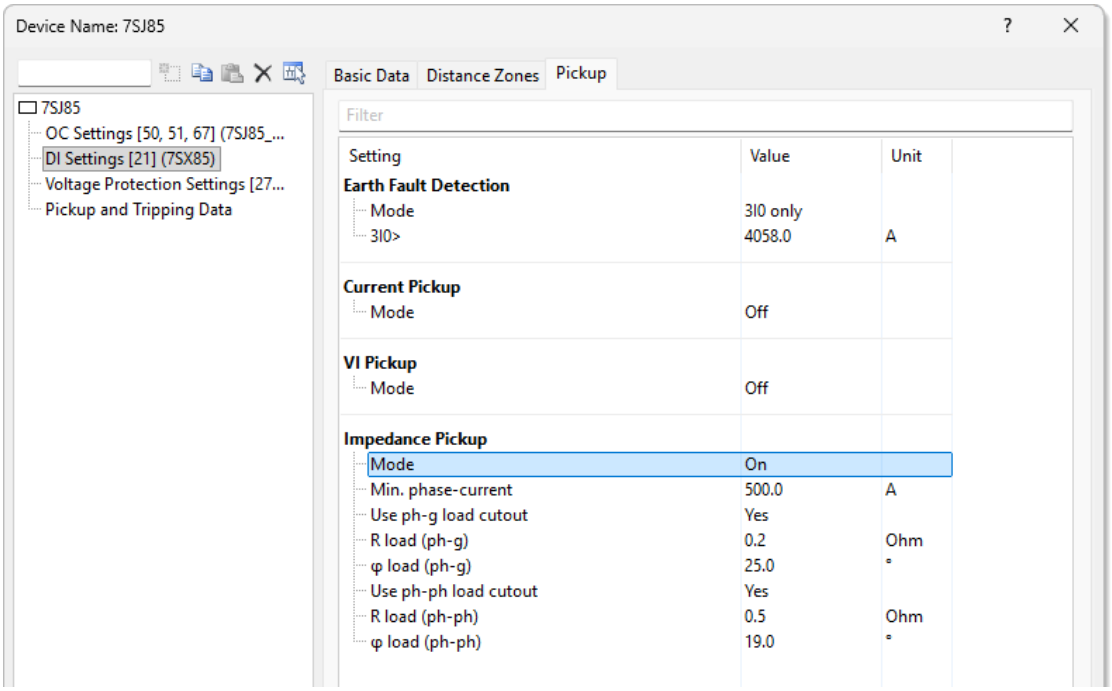
For the selectivity check, an additional grading time between digital protection devices (overcurrent protection device/dependent overcurrent time protection) and fuses has been introduced. For fuses, the new grading time **IED-FUSE** (#1) will be used in the future instead of the previous standard grading time.



This provides a more precise representation of the coordination between protection devices and fuses, particularly in the interaction between APSD and PSA.

7SX85

The new **7SX85** protection device is now available in PSS SINCAL. The device is part of the SIPROTEC 5 series and is based on the 7SA86 distance protection device.



In addition, the impedance pickup for all SIPROTEC 5 devices has been revised and aligned with the DIGSI 5 configuration tool:

- The previously separate sections **Impedance Pickup Phase** and **Impedance Pickup Ground** have been combined into a single section, **Impedance Pickup**, with a common activation option. The parameters are listed in the same order as in the parameterization tool.
- Instead of two separate minimum currents **Iph>**, there is now only one common minimum current **I>** for phase and ground.
- The load cut out can now be activated separately for phase-ground and phase-phase (**P-E load cut out** and **P-P load cut out**).
- The load parameters are now named the same as in the configuration tool: **R Load (P-E)**, **φ Load (P-E)**, **R Load (P-P)**, and **φ Load (P-P)**.

1-Phase Pickup for SIPROTEC 5 Protection Devices

This version supports 1-phase pickup for SIPROTEC 5 protection devices. This applies to all SIPROTEC 5 devices, including those already available. For other protection devices, such as SIPROTEC 4, 1-phase pickup is not currently available.

For SIPROTEC 5 devices of the **3-phase, grounded** type, 1-phase pickup is automatically activated. No additional configuration in the user interface is necessary. As a result, the phase-ground loop is enabled for the evaluation of the tripping zones even if the earth fault detection has not been triggered. These cases are marked with **1Ph** in the results.

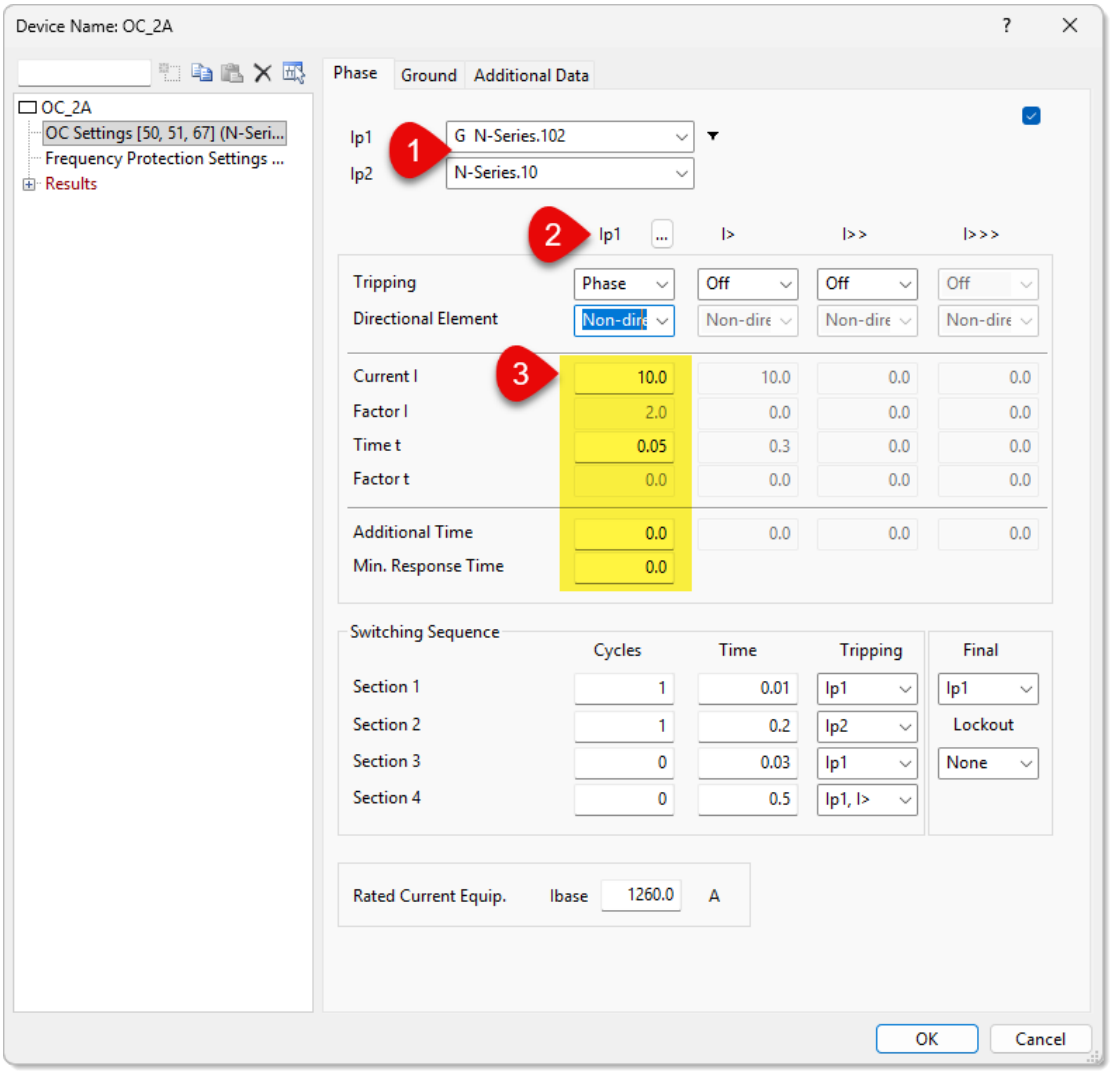
In existing networks with SIPROTEC 5 devices of this type, this may cause the results to differ from those of earlier versions. A detailed description can be found in the documentation under the section **Measurement with 1-phase pickup**.

Extended Modeling of IP Zones in Reclosers

Until now, while the two IP zones in reclosers could be configured independently, the associated parameters could only be defined together for both zones.

With this version, the parameters for the two IP zones can now be **defined and managed separately**. This aligns the functionality with the general concept of zone-based modeling in the protection database and supports device-specific mapping.

In addition, the configuration dialog box has been redesigned. The **Ip1** and **Ip2** characteristic curves are selected in (#1). The selection button (#2) allows you to switch between **Ip1** and **Ip2** (#2). The corresponding parameters for the selected IP zone are then displayed (#3) and can be configured individually.



Protection Data Manager (PDM)

The protection data exchange has been extended to support not only the transfer of settings but also the automatic creation of protection models.

Until now, the focus of data exchange has been on updating existing protection functions. In modern integration scenarios – such as the interaction between PDM, GIS interfaces, and PSS SINCAL – there is, however, a growing need to set up protection systems in a fully automated manner.

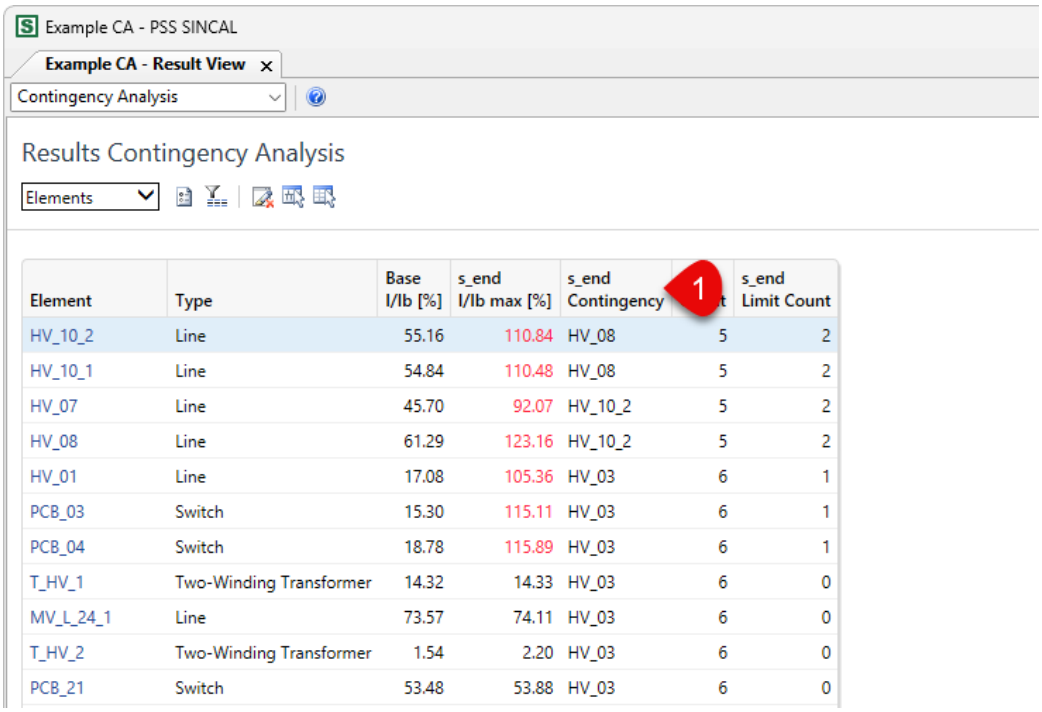
With this version, protection functions can now be created automatically during data exchange from PDM to PSS SINCAL. This occurs when no protection functions have yet been assigned to the target protection device in PSS SINCAL. This significantly simplifies the initial creation of protection models and reduces the effort required for manual modeling.

To avoid unintentionally modifying existing models, the previous behavior is retained as long as at least one protective function is already present on the protection device. In this case, no additional protection functions are automatically added; instead, only the existing data is updated.

Contingency Analysis (CA)

Result View

The result view for the contingency analysis has been extended. In **Elements** display mode, contingencies (#1) that cause the maximum load **I/Ib (max)** are now also displayed.



Element	Type	Base I/Ib [%]	s_end I/Ib max [%]	s_end Contingency	t	s_end Limit Count
HV_10_2	Line	55.16	110.84	HV_08	5	2
HV_10_1	Line	54.84	110.48	HV_08	5	2
HV_07	Line	45.70	92.07	HV_10_2	5	2
HV_08	Line	61.29	123.16	HV_10_2	5	2
HV_01	Line	17.08	105.36	HV_03	6	1
PCB_03	Switch	15.30	115.11	HV_03	6	1
PCB_04	Switch	18.78	115.89	HV_03	6	1
T_HV_1	Two-Winding Transformer	14.32	14.33	HV_03	6	0
MV_L_24_1	Line	73.57	74.11	HV_03	6	0
T_HV_2	Two-Winding Transformer	1.54	2.20	HV_03	6	0
PCB_21	Switch	53.48	53.88	HV_03	6	0

CIM Import and Export

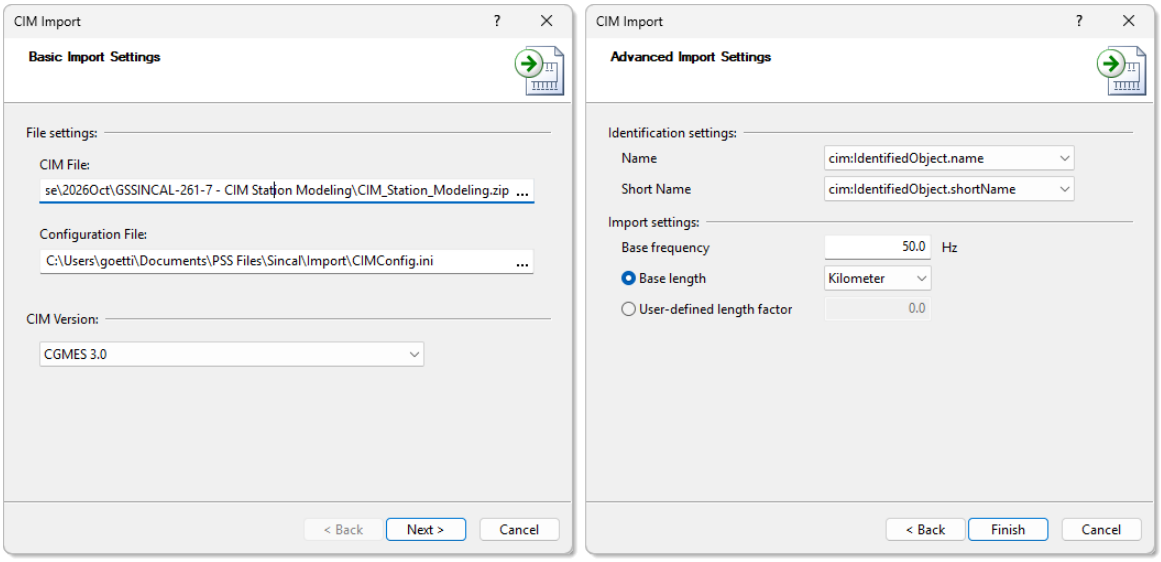
This release completes the development of the CGMES 3.0 import and export interface for PSS SINCAL. New rule-based procedures ensure a more consistent assignment between the CGMES and PSS SINCAL data models, allowing imported network models to be used directly in other PSS SINCAL modules. The enhancements include improved substation modeling, the definition of slack nodes, and the uniform treatment of nodes, busbars, and joints.

In addition, the **ENTSO-E Certification Process for CGMES 3.0** was initiated to formally verify the compliance of PSS SINCAL's import and export functions.

Simplified Import and Export Functions in the User Interface

The CIM import and export dialog boxes have been extensively redesigned to simplify use and improve the user experience. The workflows are now more organized and provide clearer guidance through each step.

To this end, the number of parameters was reduced to only those that are actually necessary and import and export processes can now be configured more quickly and carried out with less effort.



Automatic Determination of Substation Types During CIM Import and Export

The handling of **substation types** in CIM import and export has been extended to ensure consistent and realistic network modeling. Since the CIM standard does not include information on substation types, substations were previously imported as **secondary substations**, regardless of their actual function. This led to the incorrect classification of power stations, primary substations, and switching substations and required manual corrections by the user.

In this version, the appropriate PSS SINCAL station type of **power station**, **primary substation**, **secondary substation**, or **switching substation** is determined automatically during the CIM import based on the existing network data. The assignment of the station type is rule-based and considers characteristics such as connected generating units and transformers as well as their voltage level. .

Automatic substation type detection offers the following advantages:

- Correct modeling of substations immediately after import
- Time-consuming manual adjustments to substation types are no longer needed
- More reliable topology checks
- Immediate application of the feeder tracing without further processing

During a CIM export, the determined substation type is saved using the **SIN extension**, ensuring that the information is retained during a subsequent import and can be exchanged consistently between systems.

This feature is supported in **CGMES 2.4.15** and **CGMES 3.0**.

Improved Representation of Nodes, Busbars and Joints in Cim Import and Export

The import and export functionality has been extended to accurately represent the network topology and reliably preserve the model structure during import and export cycles (round trips).

During import, CIM ConnectivityNodes are now automatically imported into PSS SINCAL as busbars, joints, or nodes, depending on their intended use.

The export has also been adjusted such that the PSS SINCAL elements are correctly translated to the corresponding CIM objects. Busbars are now properly exported according to the required CIM objects and links:Nodes and joints are represented as ConnectivityNodes.

This feature is supported in **CGMES 2.4.15** and **CGMES 3.0**.

Advanced Graphics Exchange

The **exchange of network graphics has been improved**. As part of this enhancement, position information for network elements is now consistently assigned to the respective PowerSystemResources in both the GL and DL profiles. This functionality enables the export to comply with the requirements of the CIM/CGMES specification, improves interoperability with other applications and simplifies the export logic by reducing the number of references required.

Another new feature is the **import and export of graphical substation information** in both geographical and schematic views. This allows substations to be consistently displayed and visualized when exchanging data between PSS SINCAL and other applications. During import, graphical information from the GL (Geographical Layout) and DL (Diagram Layout) profiles is evaluated and applied where available. If no graphical information is provided, the substation graphics areegenerated automatically based on the assigned network elements.

Other Enhancements to CIM Import and Export

The handling of **multiple slack buses during CIM import** has been improved to correctly model distribution systems with multiple feed-in points. Previously, when multiple EnergySource objects with voltage information were imported, a selection rule was applied so that only one network feeder was modeled as a slack bus. With this version, all EnergySource objects containing voltage information are now imported as slack buses. The parameters defined in the respective CIM model are applied and mapped in PSS SINCAL. This enhancement enables more realistic modeling of distribution systems with multiple network feeders and ensures more consistent import of network data from CIM models.

The CIM interface now supports **PSS SINCAL-specific load types**. Load type definitions are stored and retrieved via the **SIN extension** in the CGMES model, enabling seamless preservation of load information during import and export. If no load type is specified, the standard load type **Common** is applied. This enhancement improves interoperability and allows to include PSS SINCAL-specific load modeling in CGMES-based workflows.

Improved Documentation

The documentation for **CIM Import and Export** has been revised and extended. It now contains a complete and up-to-date description of the supported parameters, the processing logic, and the import and export behavior.

The new documentation (available in the **System Manual** under the chapter **Importing and Exporting**) provides users with more precise guidance when using the CIM interfaces and makes it easier for them to understand the supported features and limitations.

Automation

The new version extends the PSS SINCAL API with a focus on simpler and more intuitive automation. A new, streamlined workflow for connecting network models and starting calculations replaces the previous multi-step processes. Moreover, the new workflow allows users to explicitly control whether to work with virtual or on-disk databases.

Additional enhancements include asynchronously initiated calculations, the ability to check calculation progress, and control over the calculation engine. Variants can be now created and activated directly via the API, and SINX archives can be generated automatically.

Changes to the Automation Functions (API)

The changes described below were made to the automation functions of the PSS SINCAL Platform. A detailed description is available in the **Automation Manual**.

Calculation Methods

Automation function	Change	Description
BatchMode	deprecated	Activate virtual database
SetInputState	deprecated	Set the input state
LoadDB	deprecated	Load the input data from database
ConnectDB	new	Set the database
InitCalc	new	Initialize calculation
SetVirtual	new	Activate virtual database
Start	changed	Start calculation
AsyncStart, AsyncEnd, AsyncAbort	new	Start/stop asynchronous calculation
ProgressPercent, ProgressText	new	Check state and progress
GetLastError, GetLastErrorText	new	Returns the last error
CreateVariant	new	Create new variant
SetVariant	new	Change variant
Quit	new	Quit PSS SINCAL application

Database Manager

Automation function	Change	Description
ArchiveSINX	new	Create a PSS SINCAL network archive (SINX)

Automation of the Calculation Methods

Simplified Selection of the Network Model

Previously, simulation initialization required the use of several API functions and the definition of batch modes, making the implementation complex and requiring detailed knowledge of internal data structures and processing workflows. In particular, handling virtual databases and preparing calculations involved considerable development effort.

This version introduces a new, simplified workflow that abstracts the previously required batch-mode configurations and initialization steps. New API functions provide a unified and intuitive interface for working with virtual databases and initializing calculations, significantly reducing implementation complexity.

Loading a network model and performing a simulation now requires only the following three steps:

- **ConnectDB():**
  Define the network database to be used.
- **SetVirtual():**
  Activate the virtual database. This is only necessary if you plan to work with the virtual database.
- **InitCalc():**
  Initialize the simulation and load the network model into main memory.
- **Start():**
  Start the calculation.

The following Python script demonstrates how to use the new APIs. The file "hello_sincal_nonvirtual.py"" is also included in the automation samples (**View – Project Explorer – Script – Samples Auto**).

```
import contextlib
import os
import sqlite3

import pandas as pd
from win32com import client

# Path to an existing SINCAL model (.sin file)
SAMPLES_PATH = os.path.expanduser(R"~\Documents\PSS Files\Sincal\Samples")
MODEL_PATH = os.path.join(SAMPLES_PATH, "Example MV.sin")

# StatusID returned by SINCAL when the load flow finished successfully
LF_SUCCESS = 1101

# Show all DataFrame columns when printing instead of truncating them with "..."
pd.set_option("display.max_columns", None)

# --- 1. Start the simulation server -----------------------------------------
sim_server = client.Dispatch("Sincal.SimulationSrv")
sim = sim_server.GetSimulation()

# Everything after this point runs inside try/finally so the simulation server
# is always shut down, even if an error occurs
try:
    # --- 2. Connect to the model --------------------------------------------
    sim.ConnectDB(MODEL_PATH)

    # --- 3. Run the load flow -----------------------------------------------
    sim.InitCalc("LF")
    sim.Start("LF")

    # --- 4. Check the status and read the results ---------------------------
    if sim.StatusID != LF_SUCCESS:
        print(f"Calculation failed: {sim.StatusText}")
    else:
        print("Load flow calculation completed successfully.")

        # In non-virtual mode, SINCAL writes the results directly into the model's
        # SQLite database (database.db in the model's "_files" data directory)
        result_db = os.path.join(sim.GetDataDirectory("NETWORK"), "database.db")

        # sqlite3.connect would silently create an empty file for a wrong path,
        # so check that the result database really exists first
        if not os.path.exists(result_db):
            raise FileNotFoundError(f"Result database not found: {result_db}")

        # Read the result tables into pandas DataFrames
        # contextlib.closing makes sure the connection is closed afterwards
        # (sqlite3's own "with" block only commits, it does not close)
        with contextlib.closing(sqlite3.connect(result_db)) as conn:
            lf_node_result = pd.read_sql_query("SELECT * FROM LfNodeResult", conn)
```

```
            lf_element_result = pd.read_sql_query("SELECT * FROM LfBranchResult", conn)

        print("Node results:")
        print(lf_node_result)
        print("\nElement results:")
        print(lf_element_result)

finally:
    # --- 5. Clean up ----------------------------------------------------------
    # Release the simulation object and shut down the simulation server
    sim = None
    sim_server.Quit()
```

Asynchronous Start of the Calculation & Progress State

The PSS SINCAL calculation modules can now also be executed asynchronously. In this mode, calculations run in the background without blocking the rest of the application. This mode also enables the execution status can be queried at any time, e.g., to monitor progress or display status information.

The following API functions are provided for asynchronous execution of PSS SINCAL calculation modules:

- **AsyncStart():**
  Start of an asynchronous calculation
- **AsyncEnd():**
  Check if asynchronous calculation is still active
- **AsyncAbort():**
  Abort an asynchronous calculation
- **ProgressPercent():**
  Return calculation progress of the calculation as a percentage
- **ProgressText():**
  Return calculation progress as a text stringThe following Python code example demonstrates how asynchronous execution works.

The complete implementation can be found in the DoLFAsync() function of the sample script "SimAutoSincal.py" available in the automation samples.

```
from sinpy import sincal

# Load flow calculation using async mode
def DoLFAsync():

    # Get Sincal.Simulation object
    Simulation, SimulationSrv = sincal.GetSimulation(bUseExternServer)
    if Simulation is None:
        print("Error: Sincal.Simulation not available!")
        CleanupAndQuit()

    # Setting databases & data to be loaded
    Simulation.ConnectDB(strNetwDB)

    # Set virtual mode -> all modifications and results will be stored in virtual database
    Simulation.SetVirtual(True)

    # Specify data to be considered in simulation
    Simulation.InitCalc("LF")

    # Perform specified calculation in async mode and loop until finished
    Simulation.AsyncStart("LF")
    while True:
        bFinished = Simulation.AsyncEnd(False)
        if bFinished:
            break
```

```
        iStatusID = Simulation.StatusID
        strStatusText = Simulation.StatusText
        iProgressPercent = Simulation.ProgressPercent

        print(f"Status: {iStatusID}: {strStatusText} ({iProgressPercent}%)")
        time.sleep(0.1)  # Sleep for 100 milliseconds

    if Simulation.StatusID != sincal.SimState.OK:
        sincal.WriteMessages(Simulation)
        CleanupAndQuit()

    # Save results to physical database
    Simulation.SaveDB("LF")

    # Free all stuff
    CleanupAndQuit()
```

Ending the Calculation Process

A PSS SINCAL calculation process started using the automation functions continues to run in the background as long as objects associated with the process are still referenced. The calculation process terminates automatically when no calculations are active and all associated automation objects have been released.

Depending on the programming language used, automation objects may not be released immediately but only after garbage collector has run. In such cases, the calculation process may continue to run longer than expected. To ensure immediate shutdown, the **Quit()** function can be called at the end of an automation sequence to completely shut down the calculation process.

```
from sinpy import sincal

# Get Simulation Server
Simulation, SimulationSrv = sincal.GetSimulation(True)

# Perform automation stuff
...

# Free COM objects
Simulation = None

# Quit simulation server
SimulationSrv.Quit()
```

Create and Select Variants

The new functions **SetVariant()** and **CreateVariant()** provide automation support for managing variants. These functions can be used to activate existing variants and create new variants based on a selected parent variant.Newly created variants are automatically activated in the Simulation Engine. Consistent error handling has been implemented to ensure reliable in automation and integration scenarios.

SetVariant() and **CreateVariant()** offer the following benefits:

- Direct management of variants through the simulation engine
- Simplified automation of scenario analyses and simulation workflows
- Efficient creation and evaluation of alternative network and result scenarios
- Improved integration of PSS SINCAL into automated calculation and optimization processes

```
# Change to variant "Base"
lVariantID = Simulation.SetVariant("Base", True)
if lVariantID == 0:
    iErrorCode = Simulation.GetLastError()
```

```
    strErrorText = Simulation.GetLastErrorText()

# Create a new variant with name Sub1
lVariantID = Simulation.CreateVariant("Sub1")
if lVariantID == 0:
    iErrorCode = Simulation.GetLastError()
    strErrorText = Simulation.GetLastErrorText()
```

Automation of the Database Manager

Creating a Network Archive

The Database Manager API has been extended to support the **creation of SINX archives** directly from automation and integration workflows.

Previously, exporting the PSS SINCAL exchange format **SINX** was only available through the user interface. With this version, SINX files can now be created programmatically via the Database Manager. This makes it possible, for example, to implement automated backup processes, scheduled backups, and standardized exchange workflows without manual user interaction.

The new API function **ArchiveSINX()** allows you to save the current network model as a SINX archive to a user-defined target path. The API function **ExtractSINX()** can be used to convert a SINX archive into a network model.

Result and error handling is handled through the Database Manager's existing mechanisms and integrates seamlessly into existing automation solutions.

```
# Create a PSS SINCAL network archive
DBMgr.ArchiveSINX(R"C:\Temp\Example Ele1.sin")
DBMgr.ArchiveSINX(R"C:\Temp\Example Ele1.sin", R"D:\Backup\Example Ele1.sinx")

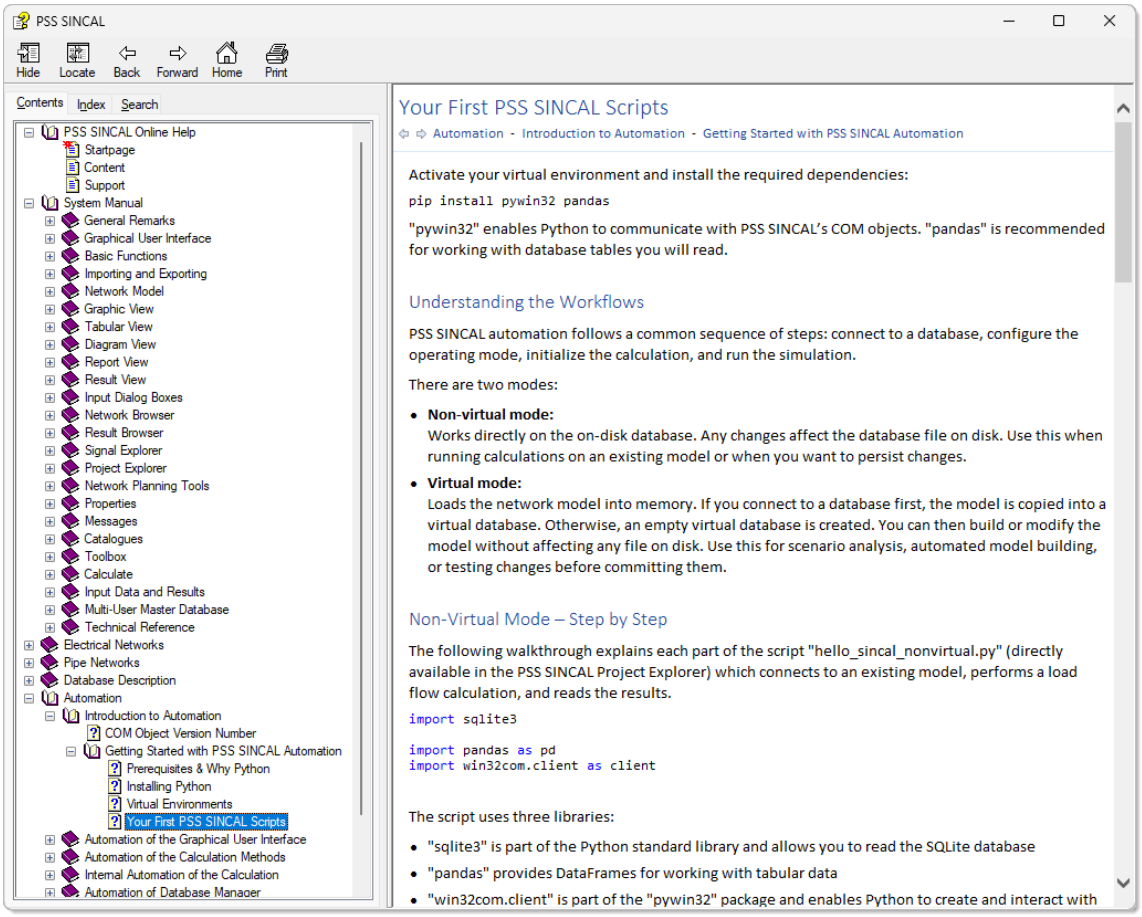
# Extract the PSS SINCAL network archive
DBMgr.ExtractSINX(R"C:\Temp\Example Ele1.sinx")
```

Documentation

The new chapter, **Getting Started with PSS SINCAL Automation**, is now available in the **Automation** manual. This guide provides a practical introduction to developing Python scripts with the PSS SINCAL Automation API. It explains how to set up the development environment and guides users through their first automation tasks.

Topics covered include:

- Setting up a python development environment with virtual environments
- Installing the required dependencies
- Supported Python versions and compatibility requirements
- Creation and execution of an initial script for power flow calculation



Network Model (Database)

Changes to the Network Model Schema (Data Model)

The changes described below were made to the data model of the PSS SINCAL network database. A detailed description of all tables and fields can be found in the **Database Description** and **Input Data** manuals.

Electrical Networks

SQL Name	Data type	Unit	Description	Comment
AsynchronousMachine				
Flag_LA	Integer		Join Load Assignment 0: No 1: Yes	unused
CalcParameter				
Flag_UseTimeSer	Integer		Consider Time Series Data and Profiles 0: No 1: Yes	rename
Flag_UseIncSer	Integer		Consider Development Series 0: No 1: Yes	rename

Flag_UseLA	Integer		Activate Load Allocation 0: No 1: Yes	rename
DCInfeeder				
HarImp_ID	Long Integer		Secondary Key – Impedance Characteristics	new
Flag_LA	Integer		Join Load Assignment 0: No 1: Yes 2: Yes – Q ind./cap.	unused
ExtendedData				
ExtendedData_ID	Long Integer		Primary Key – Extended Data	new
RowType	Integer		Row Type	new
Row_ID	Long Integer		Secondary Key – Data ID	new
Name	Text		Name of Attribute	new
Flag_Type	Integer		Type of Value 1: Number 2: Text	new
Value1	Double		Numeric Value	new
Value2	VarString		Text Value	new
Variant_ID	Long Integer		Secondary Key – Variant	new
Flag_Variant	Integer		Element of Current Variant 0: No 1: Yes	new
Unit	Text		Unit	new
GraphicHighlight				
PenWidth	Double	0.25mm	Pen Width	new
GraphicHighlightCommand				
PenWidth	Double	0.25mm	Pen Width	new
LFMeasureDataResult				
Result_ID	Long Integer		Primary Key – Result	new
MeasData_ID	Long Integer		Secondary Key – Measure Device	new
Variant_ID	Long Integer		Secondary Key – Variant	new
P	Double	MW	Active Power – Allocated Value	new
Q	Double	Mvar	Reactive Power – Allocated Value	new
S	Double	MVA	Apparent Power – Allocated Value	new
cos_phi	Double	1	Power Factor – Allocated Value	new
I	Double	kA	Current – Allocated Value	new
P_start	Double	MW	Active Power – Start Value	new
Q_start	Double	Mvar	Reactive Power – Start Value	new
S_start	Double	MVA	Apparent Power – Start Value	new
cos_phi_start	Double	1	Power Factor – Start Value	new
I_start	Double	kA	Current – Start Value	new

Flag_Result	Integer		Result Type 0: Power Flow 1: Time Series 2: Network Development 3: Motor Start-Up 4: Motor Start-Up (Simple) 5: Power Flow (NETOMAC) 6: Malfunction 7: Load Balancing 8: Resupply 9: Optimal Branching 10: Power Flow Optimization 11: Compensation Power 12: Load Allocation 13: Grid Connection Point Determination 14: Transformer Tap Detection 15: Capacitor Placement 16: PV Curve 19: Operating Points 20: Time Series Min/Max 22: Scenarios 23: Hosting Capacity 24: Contingency Analysis 25: Planning Variations	new
ResDate	TimeStamp		Date	new
ResTime	Double	h	Time	new
Flag_DataState	Integer		Data Source for Calculation 0: Not considered 1: Input data 2: Profile value 3: Operating point 4: Development value 5: Profile and development value 6: Operating point and development value 10: TSDI value 11: TSDI previous value 12: TSDI zero data 13: TSDI subst. input data 14: TSDI subst. profile value 15: TSDI quality violation 16: TSDI next valid data 17: TSDI and development value 18: Load allocation 19: Load allocation and profile value 20: Load allocation and operating point 21: Load allocation and development value 22: Load allocation, profile and development value 23: Load allocation, operating point and development value 24: Load allocation and TSDI value 25: Load allocation, TSDI and development value	new
LoadTrimRes				
Op_ID	Long Integer		Secondary Key – Operating Point	new
ResDate	TimeStamp		Date and Time	new
Op				
OpDate	TimeStamp		Date and Time	new
OperatingConf				
OperatingConf_ID	Long Integer		Primary Key – Operating Configuration	new
Variant_ID	Long Integer		Secondary Key – Variant	new

Flag_Variant	Integer		Element of Current Variant 0: No 1: Yes	new
Name	Text		Name – Operating Configuration	new
ShortName	Text		Short Name – Operating Configuration	new
Flag_State	Integer		State 0: Off 1: On	new
Flag_Type	Integer		Type of Data 1: Operating condition 2: Scenario	new
OperatingConfRel				
OperatingConfRel_ID	Long Integer		Primary Key – Operating Configuration Relationship	new
Variant_ID	Long Integer		Secondary Key – Variant	new
Flag_Variant	Integer		Element of Current Variant 0: No 1: Yes	new
OperatingConf_ID	Long Integer		Secondary Key – Operating Configuration	new
Flag_Typ	Integer		Type of Configuration 1: Outage 2: Switch Off 3: Switch On	new
Element_ID	Long Integer		Secondary Key – Network Element	new
Pos	Long Integer	1	Position	new
ProtDIRelais				
Flag_GFData	Integer		Factor Ground Impedance 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
GF_rezrl	Double		Ratio Ground to Phase Resistance	new
GF_xezrl	Double		Ratio Ground to Phase Reactance	new
GF_zezrl	Double		Ratio Ground to Phase Impedance	new
GF_phi_zezzl	Double	°	Angle Impedance Ratio	new
GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence	new
GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence	new
s2_Flag_GFData	Integer		Factor Ground Impedance – Zone 2 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
s2_GF_rezrl	Double		Ratio Ground to Phase Resistance – Zone 2	new
s2_GF_xezrl	Double		Ratio Ground to Phase Reactance – Zone 2	new
s2_GF_zezrl	Double		Ratio Ground to Phase Impedance – Zone 2	new
s2_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Zone 2	new
s2_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Zone 2	new
s2_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Zone 2	new
s3_Flag_GFData	Integer		Factor Ground Impedance – Zone 3 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
s3_GF_rezrl	Double		Ratio Ground to Phase Resistance – Zone 3	new
s3_GF_xezrl	Double		Ratio Ground to Phase Reactance – Zone 3	new
s3_GF_zezrl	Double		Ratio Ground to Phase Impedance – Zone 3	new
s3_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Zone 3	new

s3_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Zone 3	new
s3_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Zone 3	new
s4_Flag_GFData	Integer		Factor Ground Impedance – Zone 4 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
s4_GF_rezrl	Double		Ratio Ground to Phase Resistance – Zone 4	new
s4_GF_xezrl	Double		Ratio Ground to Phase Reactance – Zone 4	new
s4_GF_zezrl	Double		Ratio Ground to Phase Impedance – Zone 4	new
s4_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Zone 4	new
s4_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Zone 4	new
s4_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Zone 4	new
s5_Flag_GFData	Integer		Factor Ground Impedance – Zone 5 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
s5_GF_rezrl	Double		Ratio Ground to Phase Resistance – Zone 5	new
s5_GF_xezrl	Double		Ratio Ground to Phase Reactance – Zone 5	new
s5_GF_zezrl	Double		Ratio Ground to Phase Impedance – Zone 5	new
s5_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Zone 5	new
s5_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Zone 5	new
s5_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Zone 5	new
s6_Flag_GFData	Integer		Factor Ground Impedance – Zone 6 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
s6_GF_rezrl	Double		Ratio Ground to Phase Resistance – Zone 6	new
s6_GF_xezrl	Double		Ratio Ground to Phase Reactance – Zone 6	new
s6_GF_zezrl	Double		Ratio Ground to Phase Impedance – Zone 6	new
s6_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Zone 6	new
s6_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Zone 6	new
s6_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Zone 6	new
ip_Flag_GFData	Integer		Factor Ground Impedance – Auto-Reclosure 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
ip_GF_rezrl	Double		Ratio Ground to Phase Resistance – Auto-Reclosure	new
ip_GF_xezrl	Double		Ratio Ground to Phase Reactance – Auto-Reclosure	new
ip_GF_zezrl	Double		Ratio Ground to Phase Impedance – Auto-Reclosure	new
ip_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Auto-Reclosure	new
ip_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Auto-Reclosure	new
ip_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Auto-Reclosure	new
pp_Flag_GFData	Integer		Factor Ground Impedance – Tele Protection 0: None 1: Ratios Re and Xe 2: Ratio Ze and angle 3: R0 and X0	new
pp_GF_rezrl	Double		Ratio Ground to Phase Resistance – Tele Protection	new
pp_GF_xezrl	Double		Ratio Ground to Phase Reactance – Tele Protection	new

pp_GF_zezrl	Double		Ratio Ground to Phase Impedance – Tele Protection	new
pp_GF_phi_zezzl	Double	°	Angle Impedance Ratio – Tele Protection	new
pp_GF_X0	Double	Ohm	Reactance – Zero-Phase Sequence – Tele Protection	new
pp_GF_R0	Double	Ohm	Resistance – Zero-Phase Sequence – Tele Protection	new
s1_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 1	unused
s2_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 2	unused
s3_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 3	unused
s4_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 4	unused
s5_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 5	unused
s6_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Zone 6	unused
ip_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Auto-Reclosure	unused
pp_GrndFct_ID	Long Integer		Secondary Key – Factor Ground Impedance – Tele Protection	unused
ProtOCSetting				
Flag_Typ_ID	Integer		Secondary Key – Standard Type State 0: None 1: Global standard type 2: Local standard type	new
ProtOCSettingValue				
Name	Text		Name	new length
Unit	Text		Unit	new length
ProtPickup				
Flag_p_imin	Integer		Use ph-ph Load Cutout 0: No 1: Yes	reuse
Flag_e_imin	Integer		Use ph-g Load Cutout 0: No 1: Yes	reuse
ProtSet				
Unit	Text		Unit	new length
UnitSec	Text		Secondary Unit	new length
SynchronousMachine				
Flag_LA	Integer		Join Load Assignment 0: No 1: Yes 2: Yes – Q ind./cap.	unused
ULFMeasureDataResult				
Result_ID	Long Integer		Primary Key – Result	new
MeasData_ID	Long Integer		Secondary Key – Measure Device	new
Variant_ID	Long Integer		Secondary Key – Variant	new
P	Double	MW	Total Active Power – Allocated Value	new

Q	Double	Mvar	Total Reactive Power – Allocated Value	new
S	Double	MVA	Total Apparent Power – Allocated Value	new
P1	Double	MW	Active Power L1 – Allocated Value	new
Q1	Double	Mvar	Reactive Power L1 – Allocated Value	new
S1	Double	MVA	Apparent Power L1 – Allocated Value	new
cos_phi1	Double	1	Power Factor L1 – Allocated Value	new
I1	Double	kA	Current L1 – Allocated Value	new
P2	Double	MW	Active Power L2 – Allocated Value	new
Q2	Double	Mvar	Reactive Power L2 – Allocated Value	new
S2	Double	MVA	Apparent Power L2 – Allocated Value	new
cos_phi2	Double	1	Power Factor L2 – Allocated Value	new
I2	Double	kA	Current L2 – Allocated Value	new
P3	Double	MW	Active Power L3 – Allocated Value	new
Q3	Double	Mvar	Reactive Power L3 – Allocated Value	new
S3	Double	MVA	Apparent Power L3 – Allocated Value	new
cos_phi3	Double	1	Power Factor L3 – Allocated Value	new
I3	Double	kA	Current L3 – Allocated Value	new
P_start	Double	MW	Total Active Power – Start Value	new
Q_start	Double	Mvar	Total Reactive Power – Start Value	new
S_start	Double	MVA	Total Apparent Power – Start Value	new
P1_start	Double	MW	Active Power L1 – Start Value	new
Q1_start	Double	Mvar	Reactive Power L1 – Start Value	new
S1_start	Double	MVA	Apparent Power L1 – Start Value	new
cos_phi1_start	Double	1	Power Factor L1 – Start Value	new
I1_start	Double	kA	Current L1 – Start Value	new
P2_start	Double	MW	Active Power L2 – Start Value	new
Q2_start	Double	Mvar	Reactive Power L2 – Start Value	new
S2_start	Double	MVA	Apparent Power L2 – Start Value	new
cos_phi2_start	Double	1	Power Factor L2 – Start Value	new
I2_start	Double	kA	Current L2 – Start Value	new
P3_start	Double	MW	Active Power L3 – Start Value	new
Q3_start	Double	Mvar	Reactive Power L3 – Start Value	new
S3_start	Double	MVA	Apparent Power L3 – Start Value	new
cos_phi3_start	Double	1	Power Factor L3 – Start Value	new
I3_start	Double	kA	Current L3 – Start Value	new
Flag_Result	Integer		Result Type 0: Power Flow 1: Time Series 2: Network Development 3: Motor Start-Up 4: Motor Start-Up (Simple) 5: Power Flow (NETOMAC) 6: Malfunction 7: Load Balancing 8: Resupply 9: Optimal Branching 10: Power Flow Optimization 11: Compensation Power 12: Load Allocation 13: Grid Connection Point Determination 14: Transformer Tap Detection 15: Capacitor Placement 16: PV Curve 19: Operating Points 20: Time Series Min/Max 22: Scenarios 23: Hosting Capacity 24: Contingency Analysis 25: Planning Variations	new

ResDate	TimeStamp		Date	new
ResTime	Double	h	Time	new
Flag_DataState	Integer		Data Source for Calculation 0: Not considered 1: Input data 2: Profile value 3: Operating point 4: Development value 5: Profile and development value 6: Operating point and development value 10: TSDI value 11: TSDI previous value 12: TSDI zero data 13: TSDI subst. input data 14: TSDI subst. profile value 15: TSDI quality violation 16: TSDI next valid data 17: TSDI and development value 18: Load allocation 19: Load allocation and profile value 20: Load allocation and operating point 21: Load allocation and development value 22: Load allocation, profile and development value 23: Load allocation, operating point and development value 24: Load allocation and TSDI value 25: Load allocation, TSDI and development value	new
Flag_Phase	Integer		Phase 1: L1 2: L2 3: L3 4: L12 5: L23 6: L31 7: L123 8: N	new
VarShuntElement				
Flag_LA	Integer		Join Load Assignment 0: No 1: Yes 2: Yes – Q ind./cap.	unused

Deleted Tables
DataGroup
DataGroupDef
DataInterface
DataVal
DataValue
DataValueDef
EcoStationExt
ElementExt
LFBreakerResult
Manipulation
MeasureDataResult
NodeExt
ProtDISettingMod
SC1BreakerResult
SC2BreakerResult
SC3BreakerResult

ULFBreakerResult

Pipe Networks

SQL Name	Data type	Unit	Description	Comment
FlowExtendedData				
ExtendedData_ID	Long Integer		Primary Key – Extended Data	new
RowType	Integer		Row Type	new
Row_ID	Long Integer		Secondary Key – Data ID	new
Name	Text		Name of Attribute	new
Flag_Type	Integer		Type of Value^P1: Number^P2: Text	new
Value1	Double		Numeric Value	new
Value2	VarString		Text Value	new
Variant_ID	Long Integer		Secondary Key – Variant	new
Flag_Variant	Integer		Element of Current Variant^P0: No^P1: Yes	new
Unit	Text		Unit	new
FlowGraphicHighlight				
PenWidth	Double	0.25mm	Pen Width	new
FlowGraphicHighlightCommand				
PenWidth	Double	0.25mm	Pen Width	new

Deleted Tables
FlowDataGroup
FlowDataGroupDef
FlowDataInterface
FlowDataVal
FlowDataValue
FlowDataValueDef
FlowElementExt
FlowManipulation
FlowNodeExt

Information Technology

Licensing

New license files are required for PSS SINCAL Platform 23.0. They can be requested at the **PSS SINCAL Platform Support** (sincal.support.it@siemens.com).

Installation

A new user interface has been introduced for installing PSS SINCAL and the PSS SINCAL license server. The underlying installation technology remains unchanged continues to be based on the Wix Toolkit.

Silent installation and uninstallation are also still possible. To this end, run the installer with the following parameters:

- **Silent Installation**: SincalSetup.exe /install /quiet
- **Silent Deinstallation**: SincalSetup.exe /uninstall /quiet

System Requirements

PSS SINCAL User Interface – Client Computer

The following hardware and software requirements specify the minimum requirements for running applications on the PSS SINCAL Platform with a graphical user interface.

Hardware Requirements	
PC or Notebook	
CPU: x64, MultiCore	
RAM: >= 16 GB	
Free hard disk space: >= 50 GB	
Graphics card: >= 1920 x 1200, True Color	

64-bit Server Operating Systems Supported	
Windows 10	No longer supported as of V23.5
Windows 11	
Windows Server 2016	No longer supported as of V23.5
Windows Server 2019	No longer supported as of V23.5
Windows Server 2022	
Windows Server 2025	

Database Systems Supported	
SQLite 3.x	
Oracle 18c	No longer supported as of V23.5
Oracle 19c	
Oracle 21c	
SQL Server 2016	No longer supported as of V23.5
SQL Server 2017	
SQL Server 2019	
SQL Server 2022	

PSS SINCAL Engine – Server Computer

The following hardware and software requirements specify the minimum requirements for running the calculation module of the PSS SINCAL Platform on a server computer without a graphical user interface.

Hardware Requirements	
CPU: x64, MultiCore	
RAM: >= 16 GB	
Free hard disk space: >= 50 GB	

64-bit Server Operating Systems Supported	
Windows Server 2016	No longer supported as of V23.5
Windows Server 2019	No longer supported as of V23.5
Windows Server 2022	
Windows Server 2025	

Database Systems Supported	
SQLite 3.x	
Oracle 18c	No longer supported as of V23.5
Oracle 19c	
Oracle 21c	
SQL Server 2016	No longer supported as of V23.5
SQL Server 2017	
SQL Server 2019	
SQL Server 2022	

PSS SINCAL License Server – Server Computer

The following software requirements must be met to run the PSS SINCAL Platform License Server.

64-bit Server Operating Systems Supported	
Windows Server 2016	No longer supported as of V23.5
Windows Server 2019	No longer supported as of V23.5
Windows Server 2022	
Windows Server 2025	

There are no specific hardware requirements. Any x64 server computer that meets the requirements of the supported server operating systems is suitable.