What's new in 4.19?

www.and-solution.com

Table of contents

AND Universal GIS Interface	3
Vector tiles support (AND WMSClient plugin)	15
Message queue service	18
Mark trench path	19
Consolidate layer duplicates	23
Handling patch cables and pigtails (Warning)	25
Symbols as cassette	26
Project maintenance	28
Microduct color for jetter plan	30
Deploy microducts in reverse order	31
Cable Pull Report & Cable Length Report	32
Unlink libraries via AND DataService	40

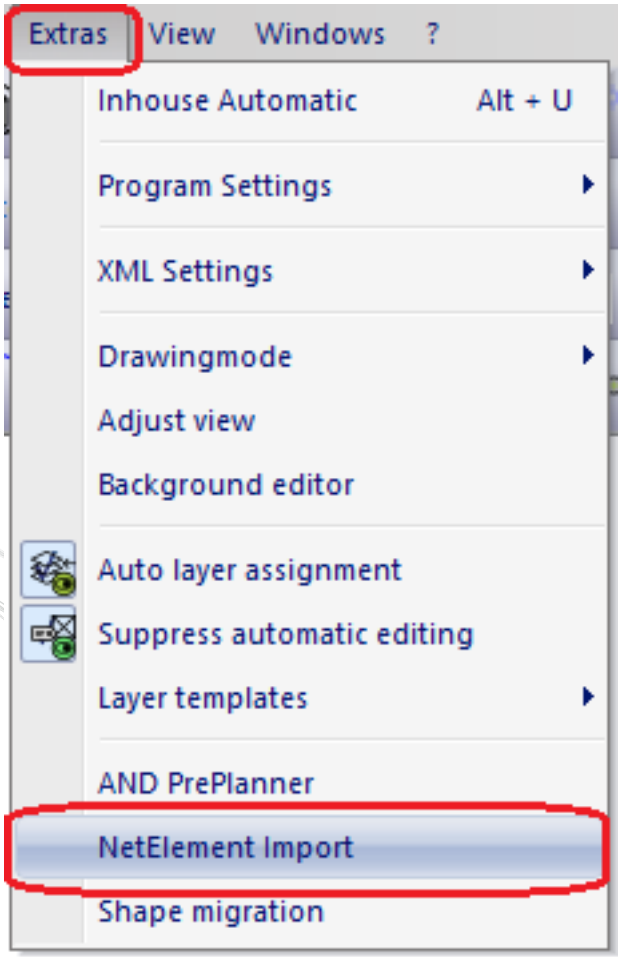
AND Universal GIS Interface

1 Abstract

In AND-Client 4.19 an import of a trench network from GeoJSON format data is possible.
Supported elements of the network are trenches and sites.
The import can be configured so that different attributes of the input objects result in different library objects or different sheet templates.

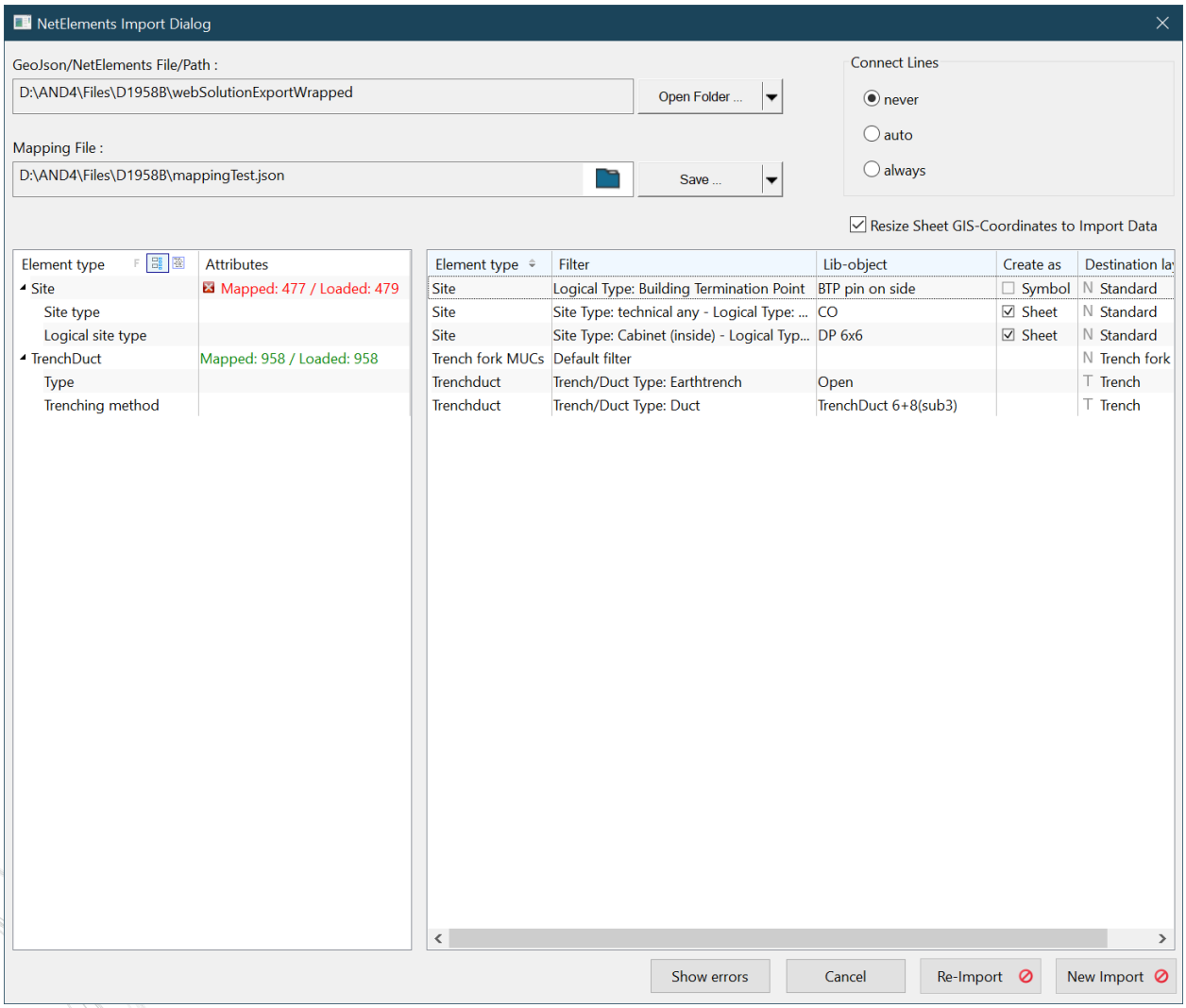
2 Starting the import

The import is started from the menu *Extras*, *NetElement Import*



On that command the *NetElements Import Dialogue* appears.

3 NetElements Import Dialogue



4 Input directory

The current input directory is shown top left. It can be changed using the *Open Folder* button.

The input directory should contain valid GeoJSON files (see the specification of GeoJSON at https://geojson.org/). GeoJSON files with a filename containing "site" are considered to be sites.

If the filename contains "trench" or "duct" or "trench_duct" the content is regarded as trenchlines.

By clicking on the arrow on the right side of the button, it is possible to switch the mode for importing single GeoJSON files.

Please note that only if sites and trenches are imported together the sites are correctly connected to the trench lines.
This means only in this case sheet connectors are created automatically. Importing complete folders is the recommended way.

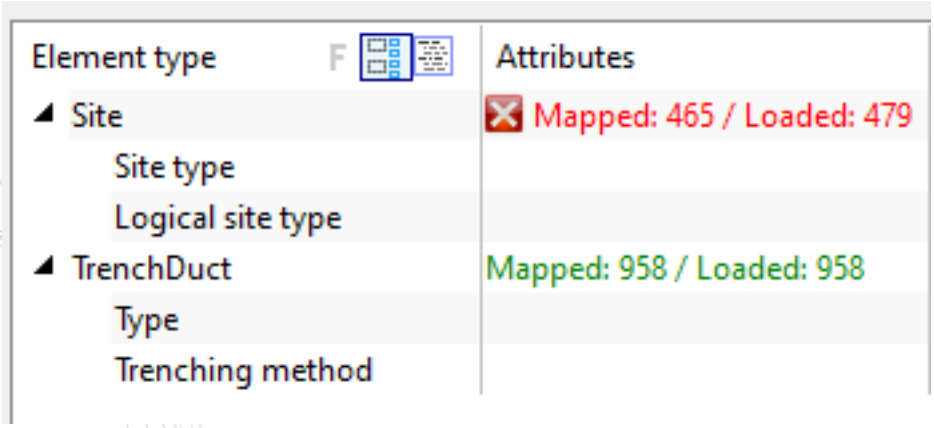
Full specification including future extensions are available on request.

4 Mapping

The main activity in this dialogue is the mapping of the imported sites and trench ducts to AND objects.
The mapping can be defined dependent on the attributes of the input objects. Import is only possible if all input objects are mapped.

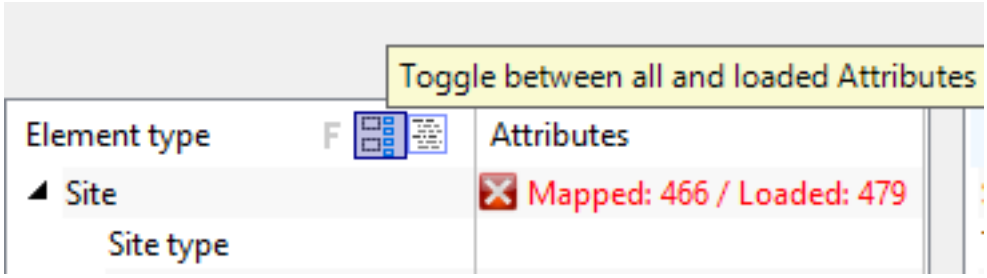
4a Input side

The list on the left side of the dialogue shows the situation on the input side:

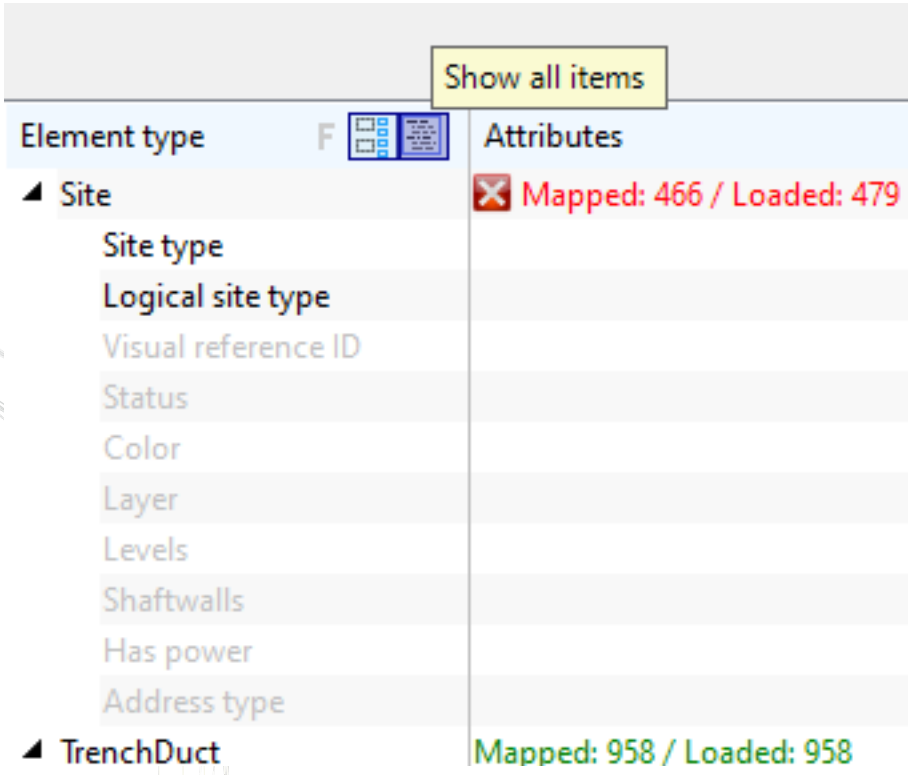


In the screenshot the situation is the following:
The input contains a total of 479 sites, 465 of which are already mapped. The text displaying that is red, because there are still 14 unmapped sites remaining.
The text for the trench ducts is green, because all 958 trench ducts are mapped.

The child entries of "Site" are the attributes which exist at least once in the site input.
If the toggle button is clicked, all possible attributes are displayed – including those not contained in the input.

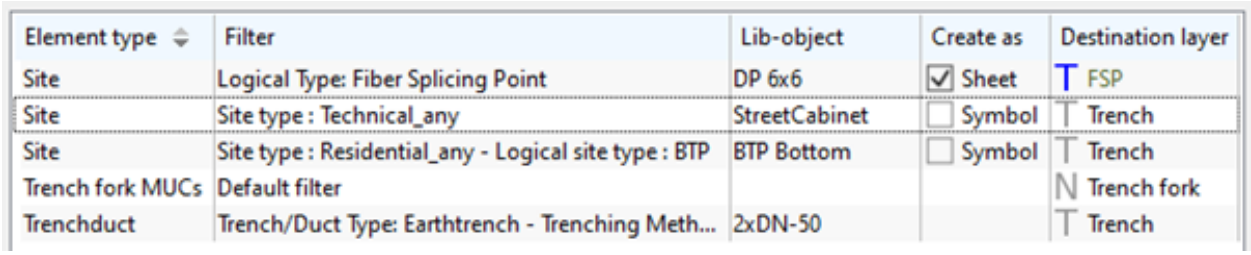


The toggle button "Show all items" also shows all possible attributes, with the unused ones displayed in grey:



4b Mapping rules

The list on the right side of the dialogue shows the current mapping rules:



Element type	Filter	Lib-object	Create as	Destination layer
Site	Logical Type: Fiber Splicing Point	DP 6x6	☑ Sheet	T FSP
Site	Site type : Technical_any	StreetCabinet	☐ Symbol	T Trench
Site	Site type : Residential_any - Logical site type : BTP	BTP Bottom	☐ Symbol	T Trench
Trench fork MUCs	Default filter			N Trench fork
Trenchduct	Trench/Duct Type: Earthtrench - Trenching Meth...	2xDN-50		T Trench

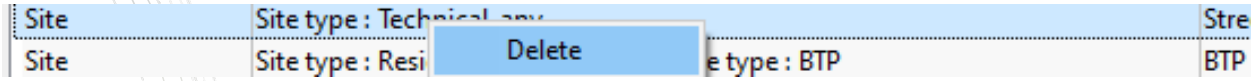
Here 5 mapping rules are defined, one per line.

The first rule says that for each site with Logical Type = "Fiber Splicing Point" a library object "DP 6x6" will be created as a worksheet on the FSP layer.
T means layer type "trench", N means layer type "netlist".

The 3rd rule says that for each site with Site Type = "Residential_any" and logical site type = "BTP" a library object "BTP Bottom" will be created as a symbol on layer "Trench".
Conditions inside a rule are always combined by a logical "and". Combining with "or" is not possible – for this case 2 rules must be made.

The 4th rule is a built-in rule that can't be deleted. This rule describes the creation of the "Trench-MUC which will be created automatically at all trench forks where trench ducts meet without a site.
For the trench fork rule only the layer can be edited.

A rule can be deleted by right clicking it and choosing the "Delete" command:

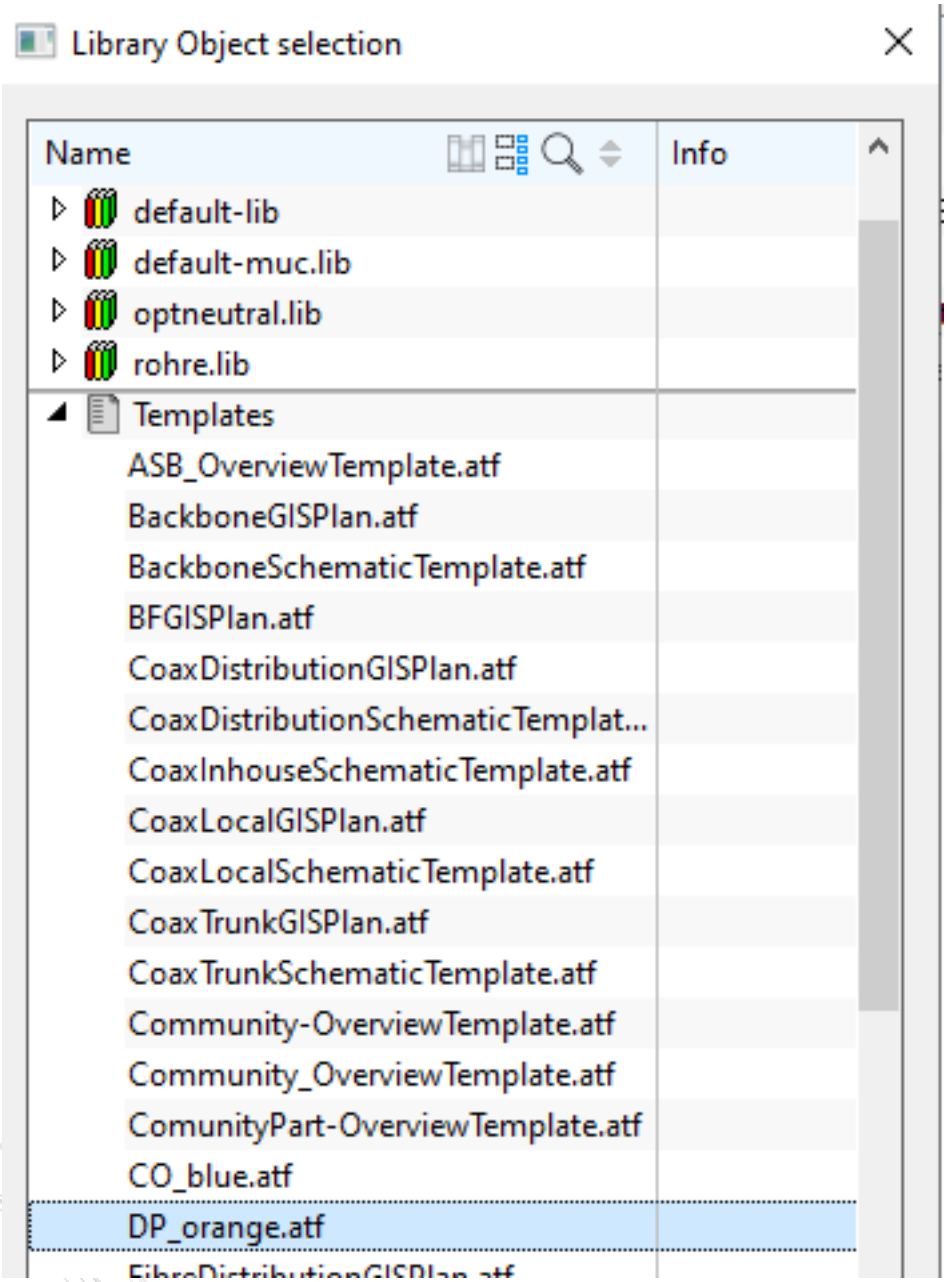


5 Library object or template

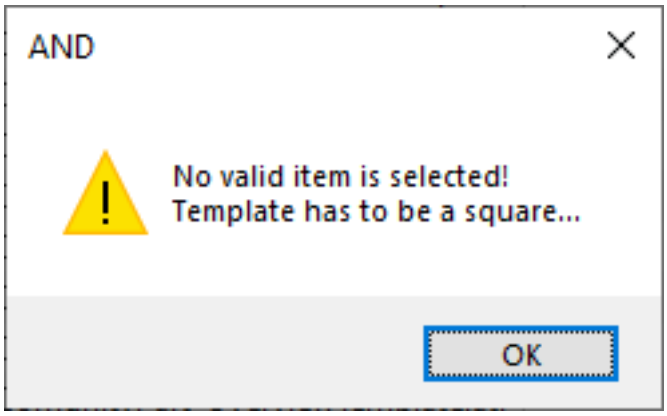
For existing rules the library object can be changed by clicking on the column "Lib-object".

For sites only symbols with quadratic shape and even size (squares) are allowed. This means symbols with size 2x2, 4x4, 6x6, 8x8 etc. in AND LibEdit. Symbols which don't match these criteria are not offered.

It is possible to use a sheet template instead of a symbol. Sheet templates can be found at the bottom of the library list:



Note: the restriction to quadratic sheet symbols exists for the templates too.
If you select a non-quadratic template you get a refusal with following message:



For trench ducts allowed library objects are:
- Trench ducts from TrenchTemplate.lib and UserTrenchTemplate.lib
- Single ducts from library
- Microduct packages from library.

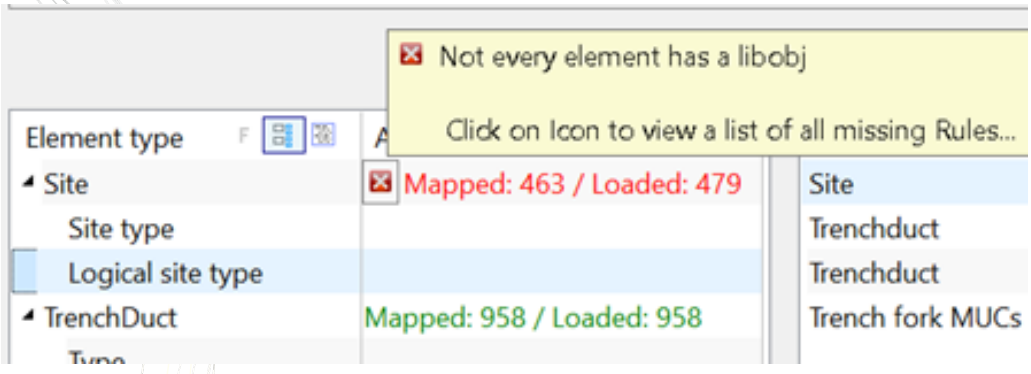
5 Layer

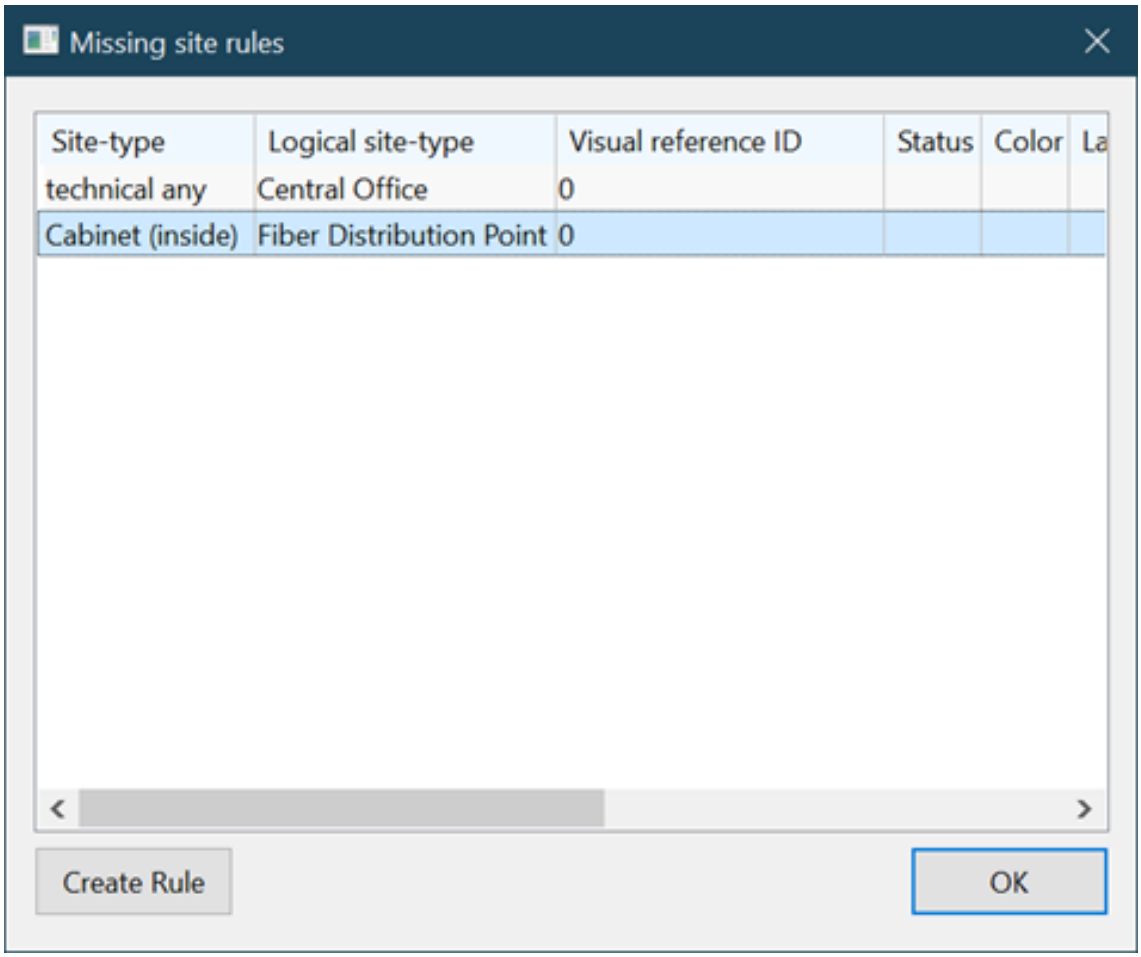
If the mouse is moved over the right side of the layer, an arrow appears. Clicking on that arrow allows selecting an existing layer.

It is also possible to create a new layer for a rule. This can be done by double clicking on the layer in the 5th column and entering the new layer's name.
For these new layers it is possible to toggle the layer type by clicking on the T/N-Icon. If the layer type is changeable, the icon is blue.

6 Adding a new rule

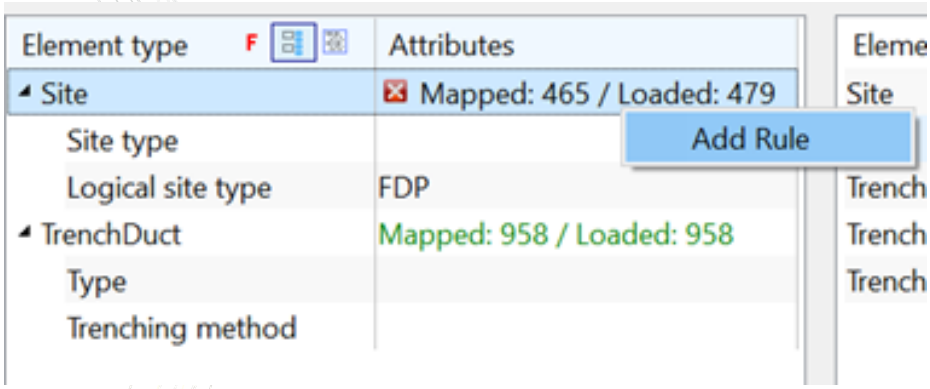
When clicking on the red x icon shown below a dialogue appears and lists the not yet mapped combinations.





If one or more lines are selected and the button „Create Rule" is clicked, a dialogue appears to select the library object or template for the rule(s).
After that selection the new rule is added to the list on the right side and the number of mapped objects is updated.
The destination layer and the flag „Create as sheet" must be edited in the right side list.

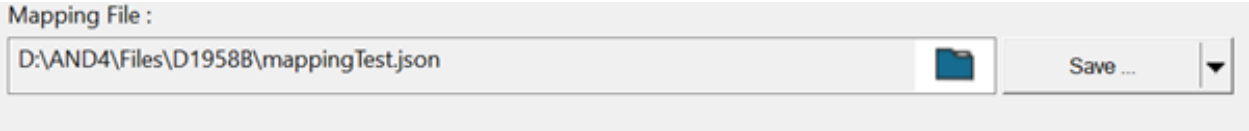
As alternative way it is also possible to directly right click on the red or green text displaying how many objects are mapped and select the command *Add Rule.*



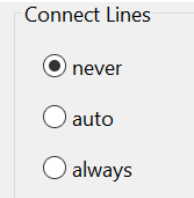
This creates a new rule using the current attribute situation.
In the situation shown in the screen shot above this is Logical site type = „FDP“ and Site type = any.
If no special value of an attribute is set, this means any value.

7 Saving and loading mappings

The current mapping can be saved using the „Save“ button.
A mapping file can be loaded using the folder-button left of the „Save“ button



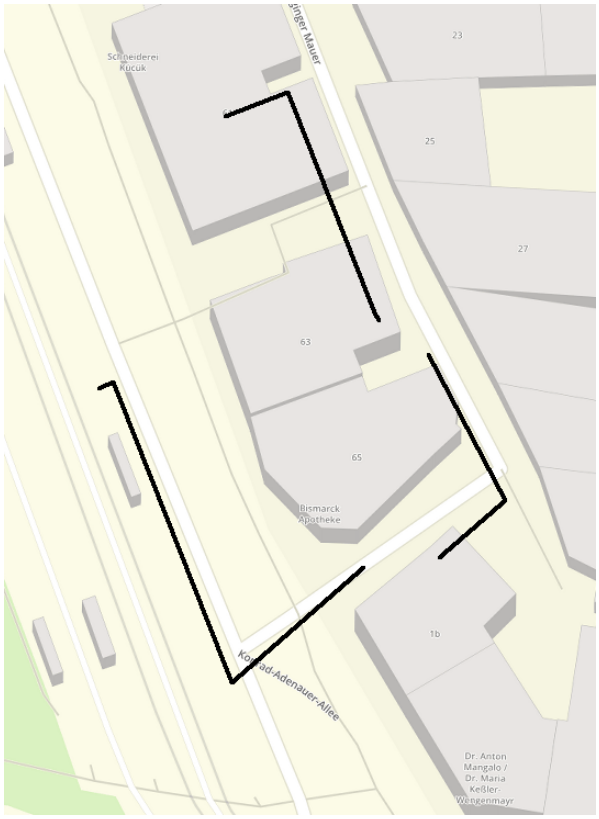
8 Option „Connect Lines“

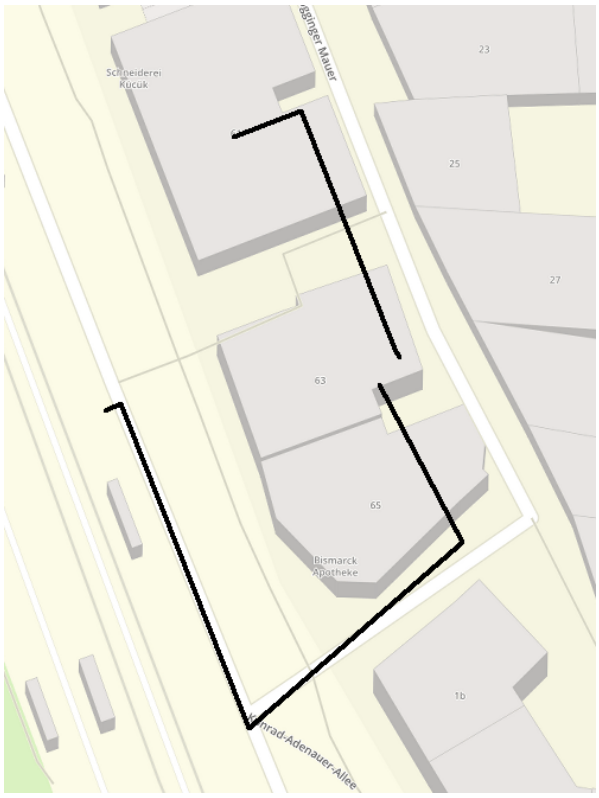


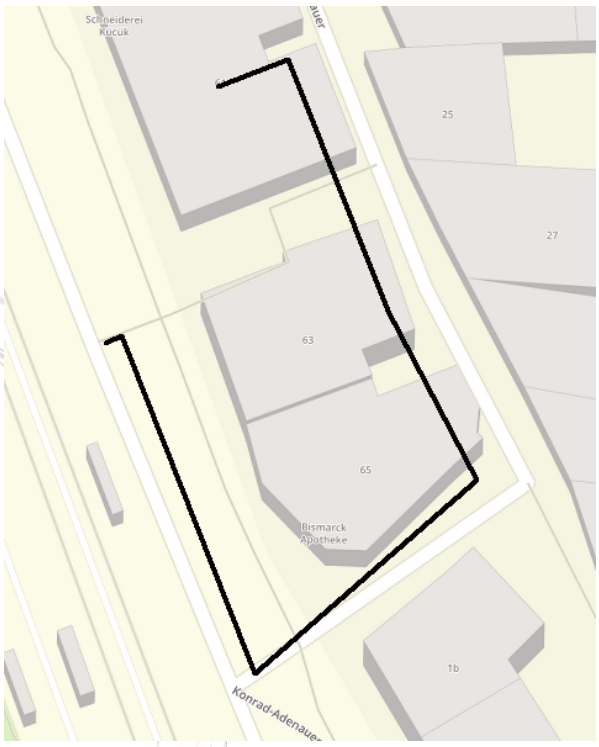
This option determines how a single pair of connected trenches is imported.

- never → never connect lines no matter what (see first example where each line is not connected)
- auto → connect lines but only if they are the same and the attribute "identification" is the same as the connecting line (see second example)
- always → connect lines if they are the same, ignoring attributes etc.

(example input file: trench_duct(connect-lines).geojson)


Schneiderei
Kücük
23
25
27
63
65
Bismarck
Apotheke
1b
Konrad-Adenauer-Allee
Dr. Anton
Mangalo /
Dr. Maria
Keller-
Weingenmayr


Schneiderei
Kücük
23
25
27
63
65
Bismarck
Apotheke
1b


Schneiderei
Kücük
25
27
63
65
Bismarck
Apotheke
1b

Please note that this option only handles pairs of trenches.If three or more trenches meet at one point, this is a path fork for which a Default-MUC is created. In case of path forks all affected trenches remain separate objects.

9 Microducts

Microduct packages are trenches with type = 105 in the geojson.

The legs of the microducts are trenches with that microduct set as parent.

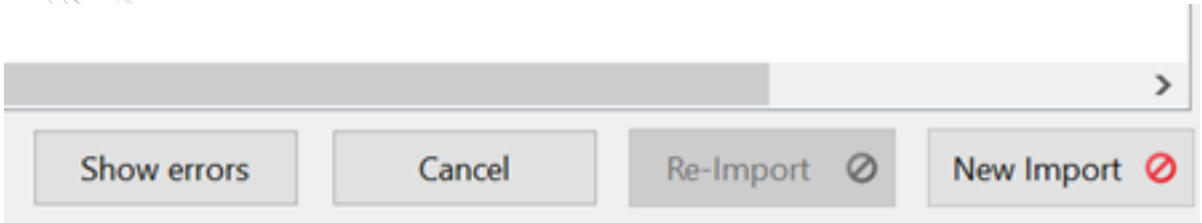
Example in GeoJSON form:

"id": "1054",
"properties": {
..
"type": 105,// that trench is a microduct package
"trenching_method": 1201
}

"id": "1029",
"properties": {
....
"type": 105,//can be any value.the type of microduct legs is ignored
"trenching_method": 1201,
"parent_id": "1054" // is a microduct-leg of the parent microduct package with id = 1054
}

If the microduct package in the GeoJSON file is mapped to a microduct package from a library, the import will create one microduct object for the package and all its legs.

10 Finalize the import



If all objects are mapped, the Button *New Import* is enabled. When clicking it, trenches and sites are imported and added to the project.
If the resize option is enabled, the sheet bounds are reduced to imported area plus 10%.

The purpose of the Button *Re-Import* is for repeating the same import with other settings without being asked whether to delete all previous imported objects.
The difference to a normal import is that on re-import all previously imported GIS objects are deleted automatically.
Only objects originating from a previous import and only those inside the polygon of the data to be imported will be deleted.
The Button *Re-Import* is only enabled if all data is mapped and if at least one object which would be deleted exists.

The ID of the GeoJSON source object is written into the AND object created from it as reference (as triple with name „GisImportSourceId“).

The MUC created for the trench forks is a built-in object which exists since AND 4.19.
As consequence the projects created by the AND Universal GIS Interface can be used only in AND 4.19 and higher.

Vector tiles support (AND WMSclient plugin)

AND now is capable of using MapTiles-servers (MVT) to request maps as vector-tiles (*.pbf) and to render maps to use/show them the same way as with WMS/WMTS-data sources.

1 General info

Due to the requirements the internal workflow is complex:

1. Request tile from the server and store in tiles-cache on disk
2. Reproject to each requested SRS on-the-fly when needed in AND, projected tiles (one per original-tile's layers) are saved on disk too.
3. Load projected files into a memory cache
4. Render maps using these memory-cached files via Mapnik (see below).

This means depending on the current situation the time needed for displaying the map after a zoom-/scroll-action can differ a lot depending on the fact which of these steps are currently needed.

2 Styles

Rendering the maps requires a set of styles. Styles define the rendering options like colors, line widths, icons and fonts depending on feature data and scale.

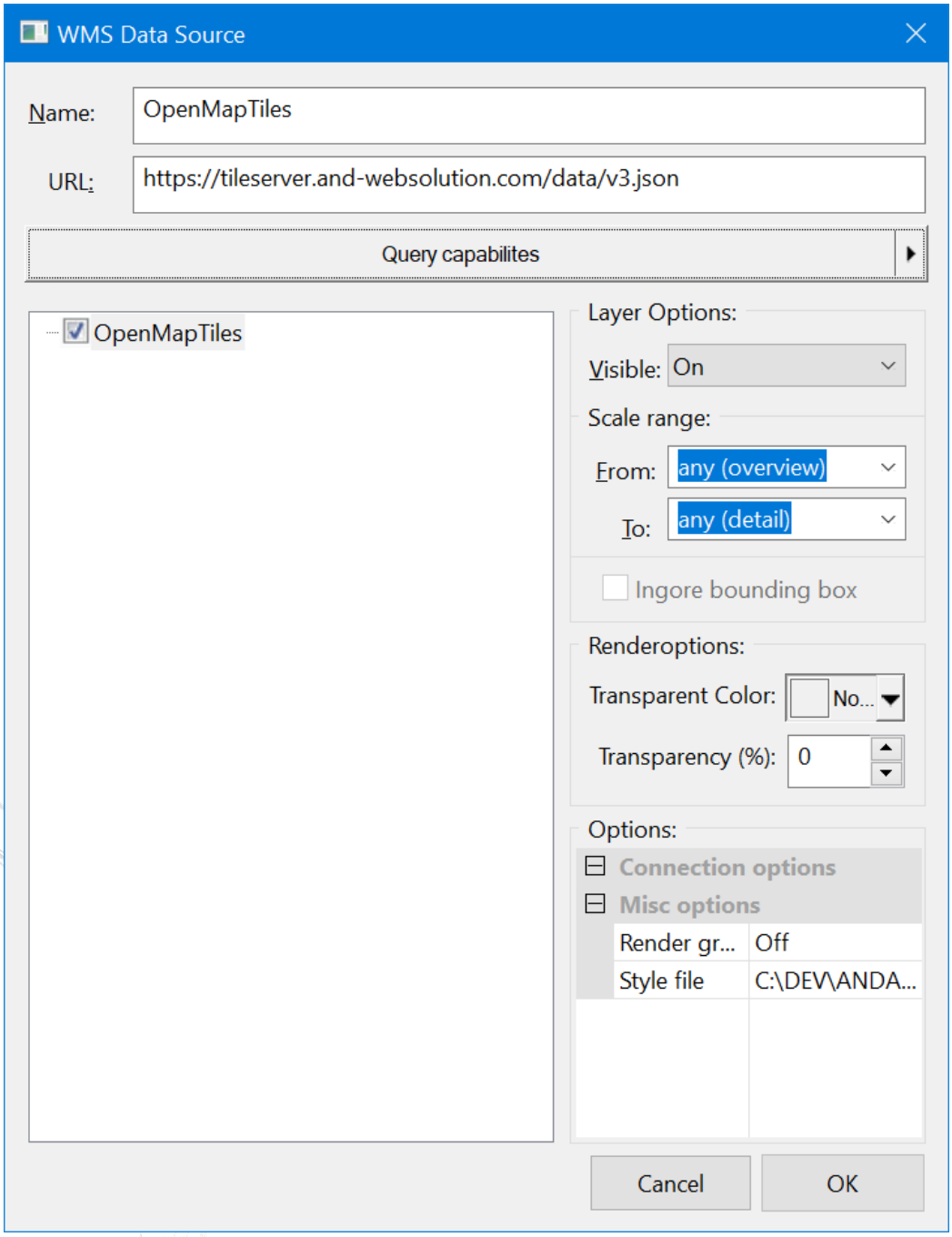
AND uses the open source library Mapnik (https://github.com/mapnik/mapnik) which uses Mapnik styles (a so-called Mapnik style sheet plus resources in form of a set of icons, PNGs and/or fonts) for rendering.

Up to now there is no support for MapTiles-styles loaded from the AND WebSolution-Server. It is planned to add the functionality needed to download the styles/fonts/icons used by the OpenMapTiles-server on https://tileserver.and-websolution.com/. Before this is available it is necessary to configure MapTiles datasources to use locally stored Mapnik styles.

AND supports its own kind of style files (*.styles) which are simply zipped folders with a Mapnik style sheet and all needed resources, and we deliver a default style-set (*<exe-path>\data\WebSolution\WebSolution (OSM).styles*) with a Mapnik style sheet and icons and patterns which makes rendered maps look quite similar to the *InvNet style based on OSM Bright* section at https://tileserver.and-websolution.com/

3 GUI

The configuration is nearly identical to that for WMS/WMTS, it is executed via the *WMS Data Source*-dialog from the WMS-Plugin's toolbar, when using the URL to a valid MVT-json file.



The link in this screenshot uses our official WebSolution https://tileserver.and-websolution.com/ in the section „Data".

4 AND WebSolution

The preliminary target at the moment is to add AND WebSolution maps to AND. For this we deliver a default (read-only) WMS document *<exe-path>\data\WebSolution\WebSolution (OSM).mapdoc*, which is always loaded in the WMS-Plugin.

Message queue service

A new service module for the AND SmartServer makes it possible to automatically send event-dependent messages to other systems.

Rules for triggers, filters and message content can be defined for this purpose.

When an AND planning document is committed/checked in, the system first checks whether there is a possible trigger, such as a status change. The filter is then used to determine whether the type is for example an ONT. The other associated rules then allow the topic path and the content of the message to be sent to be defined, for example

Type: ONT-A99

Number: 4826846

StatusNew: in operation

Address: Street, postal code and city

POP: 928

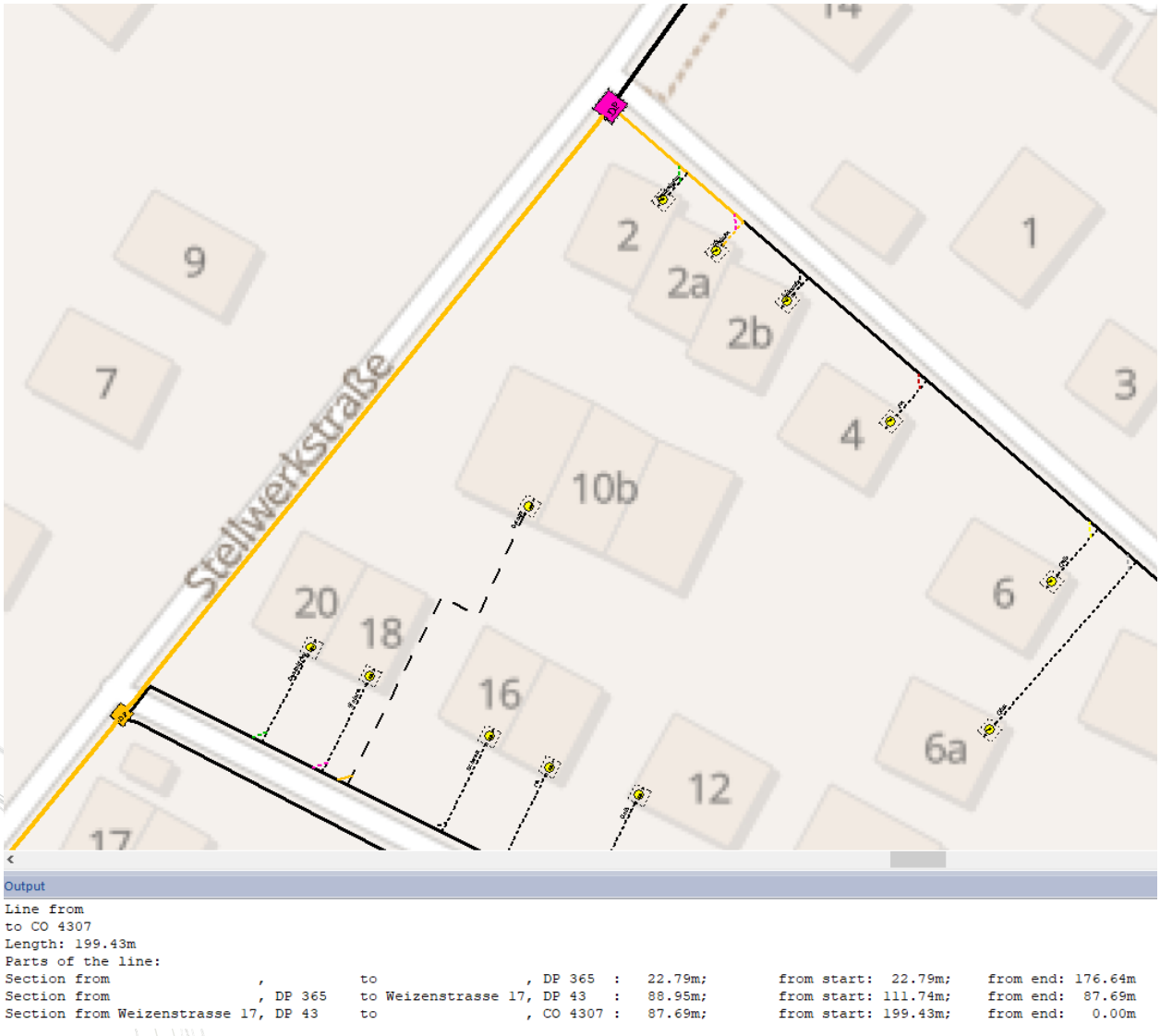
ODF port: HE4/S3/P2

For each change in the AND document to which the triggers and filters apply, an individual message is generated and sent to a message broker (e.g. Kafka). Another tool like a CRM can subscribe such a message and will then immediately be informed for example to switch a subscriber contract to active.

Mark trench path

Since AND 4.19 a new function makes it possible to mark the trench path from a given start pin to the highest hierarchy object (e.g. POP).

The marking color is the same as for the Q-function and the stations are listed in the output window too. The shortcut is SHIFT + Q.



The command is available for pins of trench lines, (micro-)duct packages, microduct legs and the symbols connected to it (SheetSymbol, MUCs, TrenchDivider).

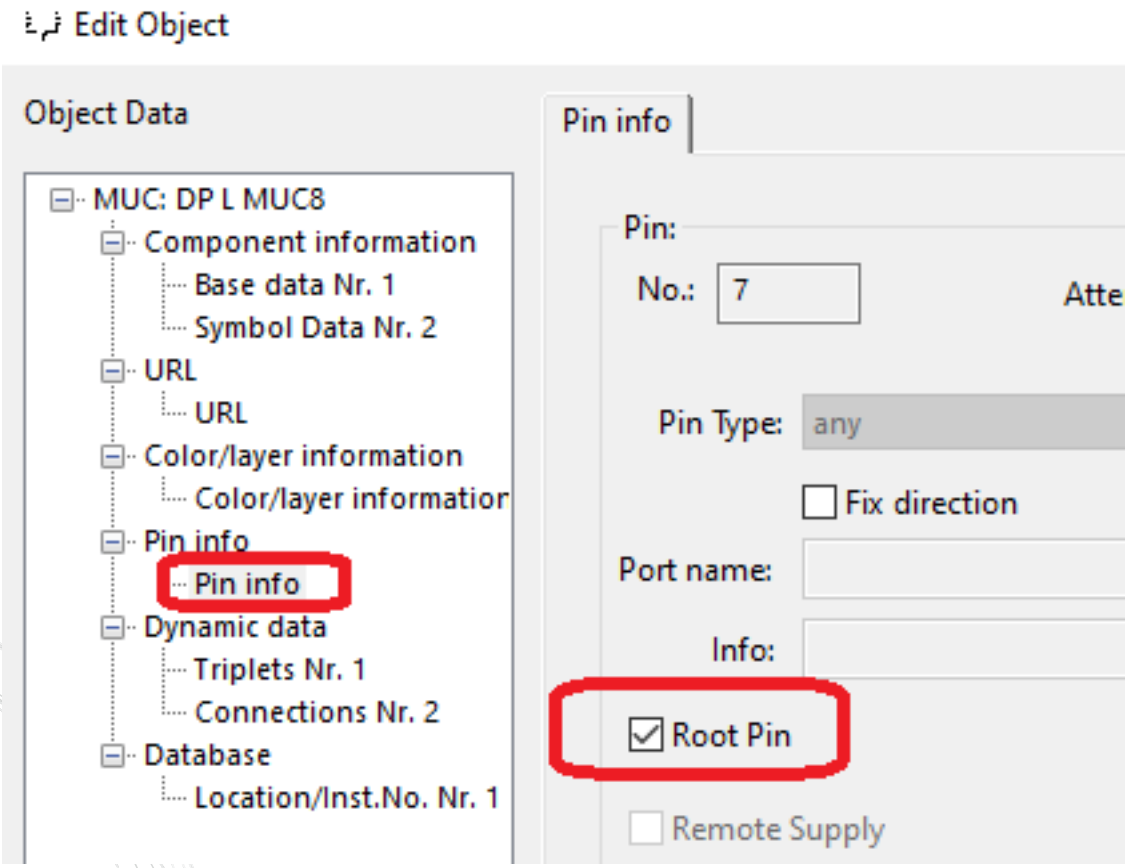
The routing walks along the trench path in the same (GIS-)sheet. If it meets a sheet symbol, it doesn't enter the sheet. Instead it directly walks to the opposite pins of the sheet symbol.

Only pins of type "Trench", "DuctPackage" and "Microduct" are accepted as opposite pins of trench path objects. Pins of other types (optical, coax etc.) are ignored.

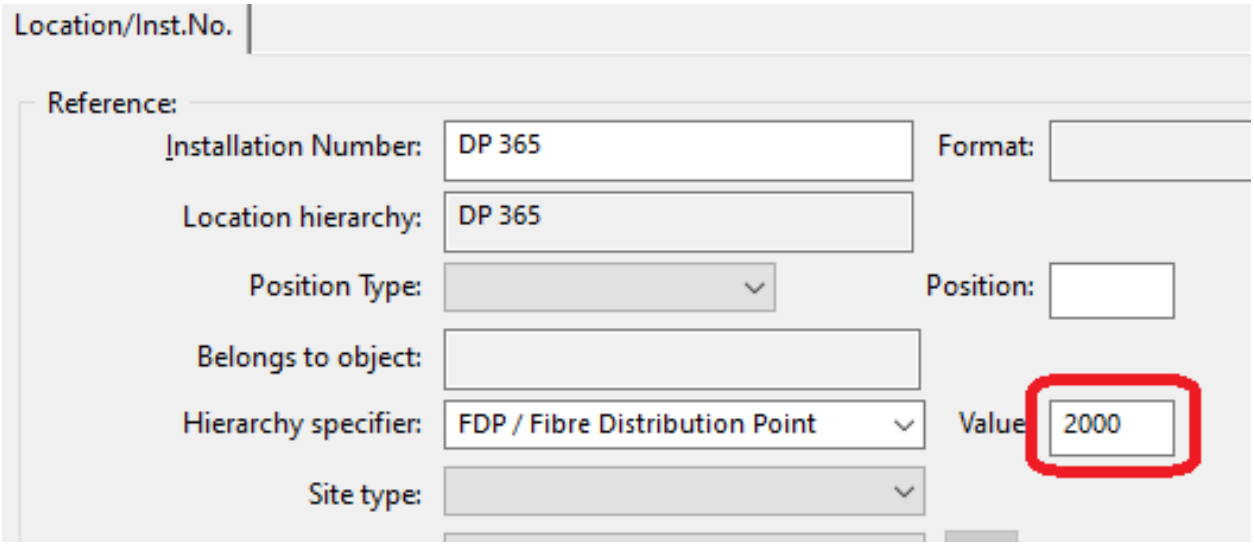
Symbols with three or more trench-like pins are path forks.

If one of the opposite pins is marked as "Root Pin", the routing is continued with that root pin, otherwise the routing is stopped.

For 1:1-connectors the routing is always continued without using any root pin.

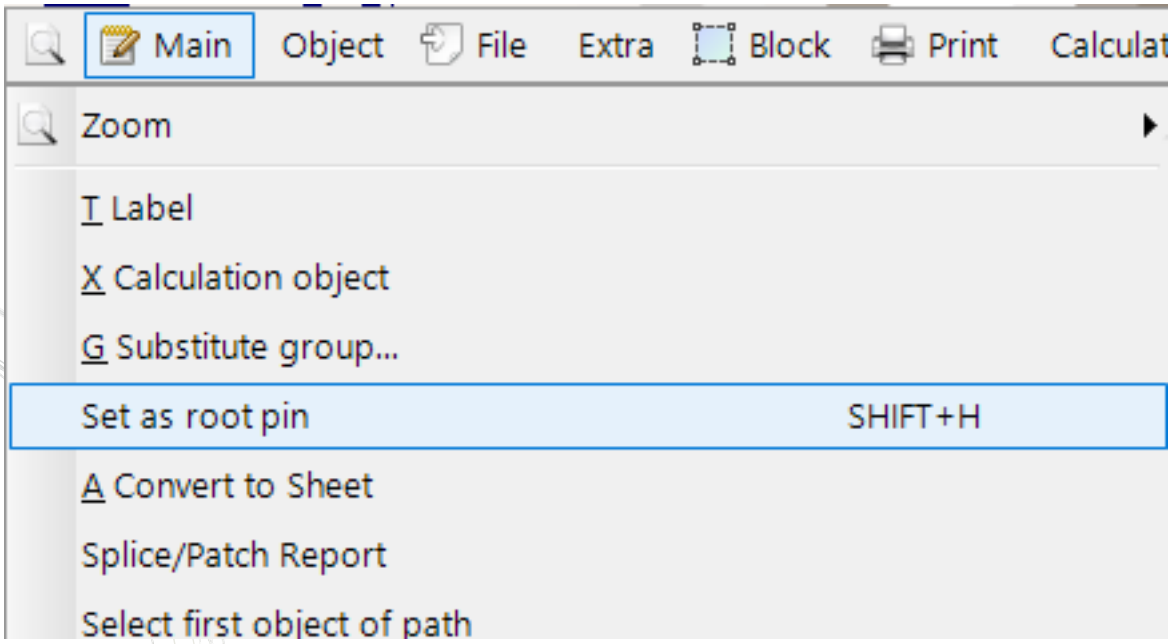


The marked result path is the one to the symbol with highest hierarchy value along the passed objects.
This can be shorter than the path from start pin to the pin where routing stopped, e.g. if routing stopped at an open line pin.
If several symbols exist with equal maximum hierarchy value in a path, the longest path is displayed.



If the hierarchy value for a sheet isn't set, the highest value of all objects inside the sheet is taken.

The flag "Root Pin" can be edited in the "Pin info" dialogue shown on the previous page. In order to accelerate editing, an extra command for toggling that flag was implemented. The shortcut is SHIFT+H.



If the flag is off at the moment, the text in context menu is "Set as root pin", otherwise it is "Switch off root pin".

Different from the normal Q-function the output window shows the object data for start and end point, not the pin data.

If the start pin is a line pin and that pin is connected to a symbol, the data of that symbol is displayed as start data.

AND Universal GIS Interface automatically sets the root-pin-flag for the "input" of all path forks connected to CO.
As a result marking a trench path immediately works in projects produced by AND Universal GIS Interface.

Direction determination

For a given start pin two directions are possible: "away from object of start pin" or "through object". Internally both directions are tried and then both paths where compared.

Symbols with root pin passed (from back side) are used as direction confirmators, which increase the value of confirmed directions.

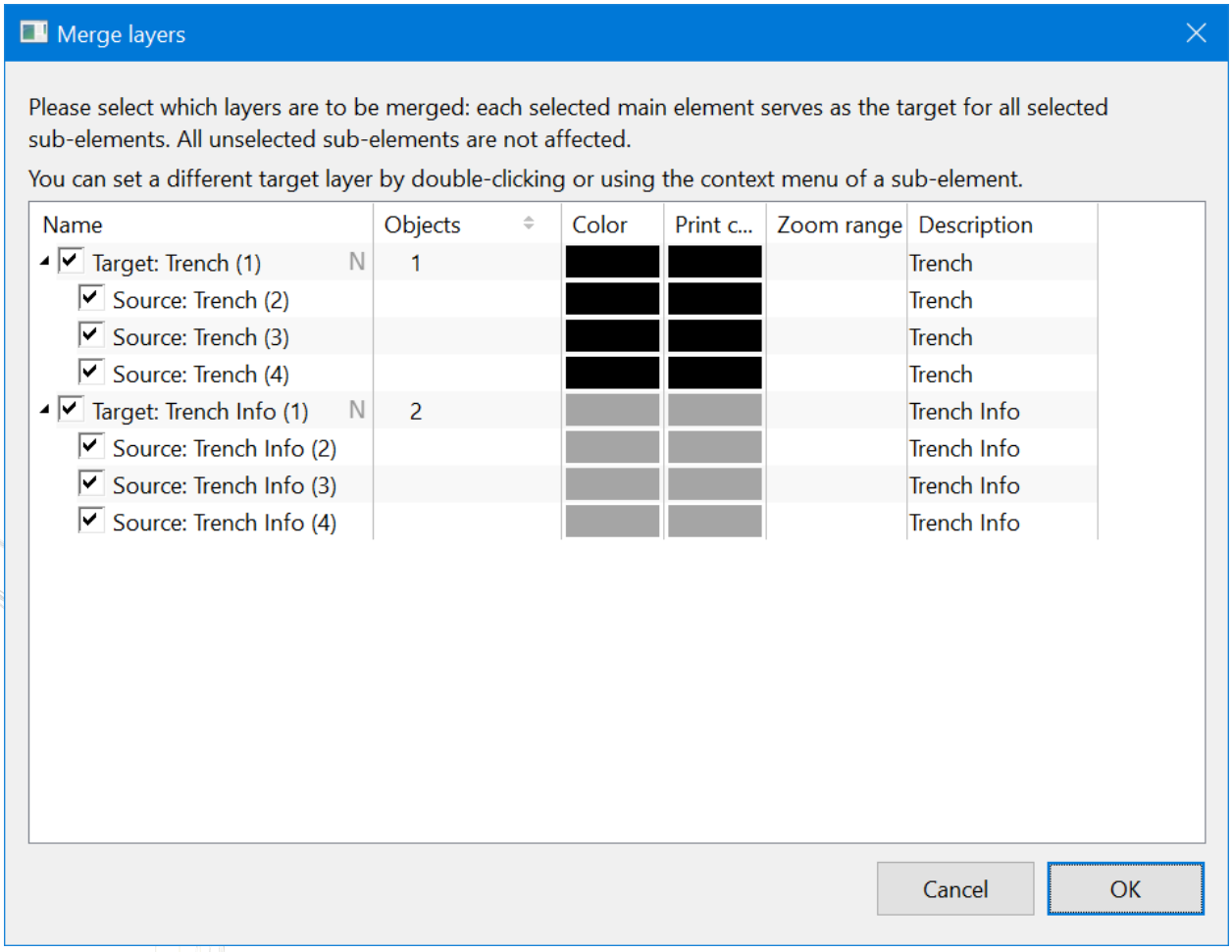
The path with more confirmed directions is regarded as the right one to be marked.
If the number of confirmed directions is equal, the path containing the higher hierarchy object is taken.
If that hierarchy value is equal too, the path containing more objects is taken.

Consolidate layer duplicates

Layer duplicates are layers which have a name followed by one or more numbers in braces: *Trench (1)* or *Trench (1) (2)* are duplicates of a layer *Trench* if more than one of layers with such names exist within one of the layer groups *Netlist* or *Civilworks*.

Those duplicates most probably are relicts of an error which caused the creation of more and more duplicates if the real layer (i.e. *Trench*) existed in another group than expected in AutoLayerAssignment.xml.
If a project contains such layer duplicates a dialogue is available to consolidate them by merging them into one.

To do this click on *Consolidate layer duplicates* in *Project maintenance* (in menu *Extras*) to open the *Merge layers* dialogue:



The layers on the root level (prefixed with *Target*) are the layers where all objects from all selected sub-items' layers (prefixed with *Source*) will be moved to.

It is possible to skip merging either for the complete set of duplicates found for one layer name or to just choose some possible source layers for merging.

The *Target* layer for each group can be replaced by any of its listed *Source* layers by double clicking the *Source* layer or via the context menu.

After clicking ***OK*** (and confirming the following warning) all objects of all selected *Source* layers are moved to their selected *Target* layers and the selected *Source* layers are deleted automatically.

Handling patch cables and pigtails (Warning)

New warning, if patch cables or pigtails are connected by sheetconnector or link.

Due to misunderstandings in the handling of patch cables and pigtails in AND, from AND 4.19 on there is a new warning when loading a project and during netcheck if the check of net structure is activated.

Text in English:
"Patch cables and pigtails shouldn't be connected via sheet connector or link
(Counts 2x in BOM, calculation, but is in reality one object)".

The recommended way to continue a patchcable in another sheet is to use a linked object and to have the cable in only in one sheet.

The warning can be switched off for the current user here:

[HKEY_CURRENT_USER\SOFTWARE\CDS\AND\4.0\OEM]
"WarnPatchCableToPatchCable"=dword:00000000

and for all users here:

[HKEY_LOCAL_MACHINE\SOFTWARE\CDS\AND\4.0\OEM]
"WarnPatchCableToPatchCable"=dword:00000000

Symbols as cassette

1 Motivation

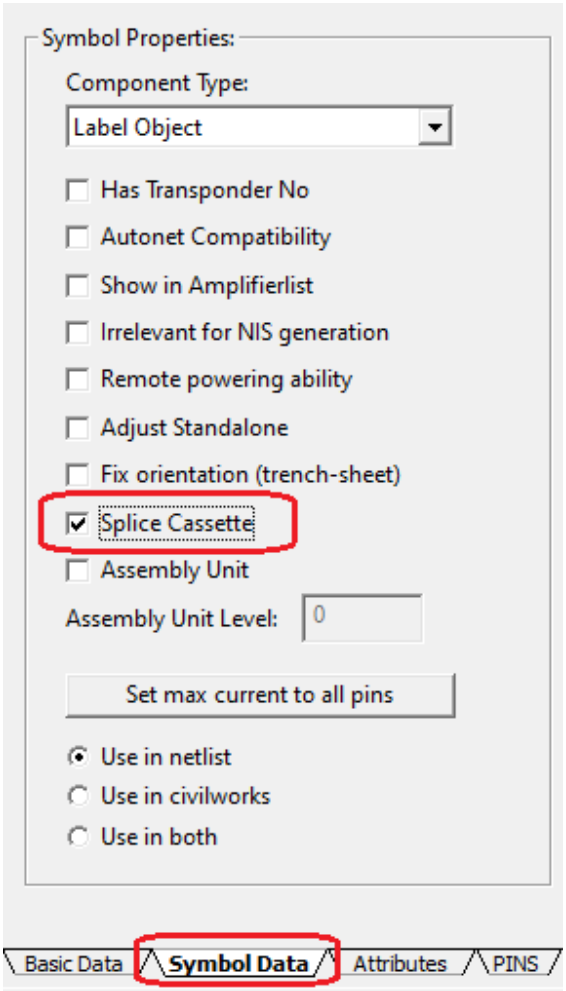
Until now splice cassettes are represented in AND by special tray rectangles („Kassettenrahmen“ in German).

All splices inside such a rectangle are regarded as inside a cassette with the name of the rectangle.

Since AND 4.19 it is possible to mark symbols in the library as "Splice Cassette". Drawing an extra tray rectangle is often not necessary anymore.

2 AND LibEdit

Marking a symbol as cassette in AND LibEdit:



4 AND client

If a symbol is marked as "Splice Cassette" in the library, the installation number of that symbol is regarded as surrounding cassette in splice reports and dynamic labels.

A conventional tray frame is more relevant and overwrites the cassette name inherited from symbols.
This allows manual user corrections and guarantees downward compatibility.

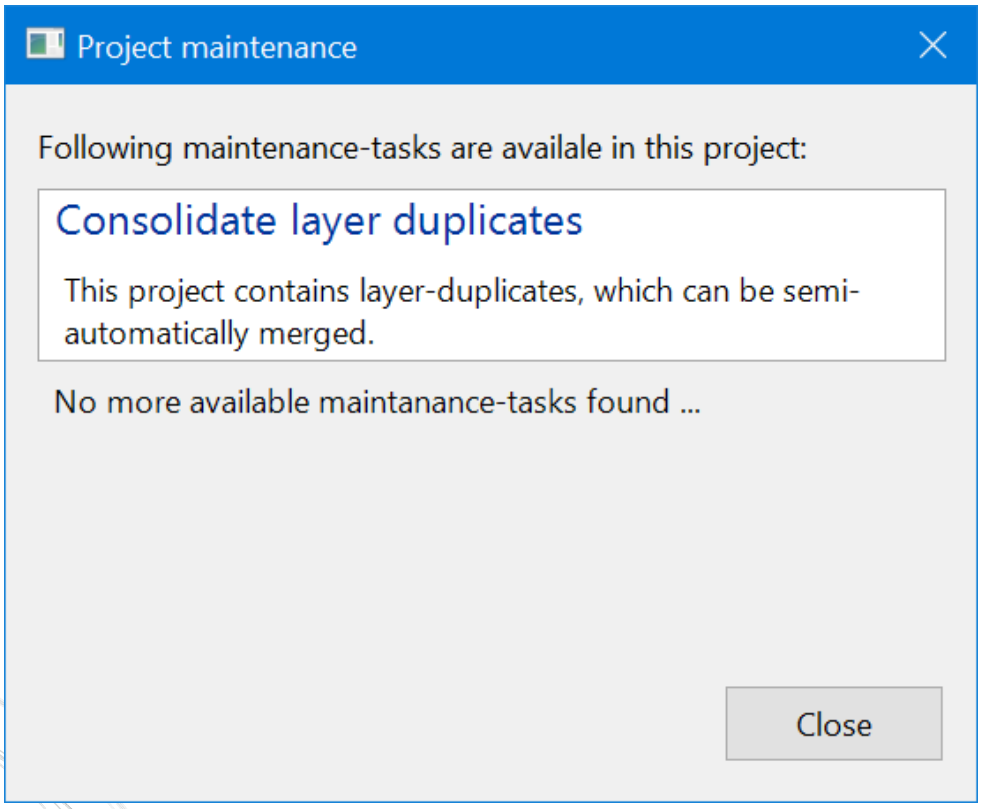
If there is more than one cassette symbol around a splice, the innermost symbol is the relevant one.
Cassette symbols will be ignored if they aren't drawn completely inside each other (This is not recommended, same as for location hierarchy of rectangles).

Project maintenance

1 Abstract

In the *Extras* menu a new item *Project maintenance* has been added. This offers a list of tasks a user can perform to do any kind of work which helps to keep the project clean/small.

The dialogue is very simple: it just shows a list of buttons (currently only one is implemented) like this:



Pressing the buttons then execute tasks, which usually means some further dialogue is shown.

2 Implemented maintenance tasks

The following task is currently implemented:

Consolidate layer duplicates

3 Planned maintenance tasks

The following tasks are planned to be implemented in future releases:

Cleanup unnecessary data

A central place to clean up data like e.g. unused tasks, locations, layers, a.s.o.

Remove unreferenced EEP-links

A functionality to remove link-info for unreferenced EEP-links (aka. EEPs where the partner-drawing doesn't exist anymore).

Microduct color for jetter plan

The report variables **JetSegmentMicroductColorFrom** and **JetSegmentMicroductColorTo** have been added.

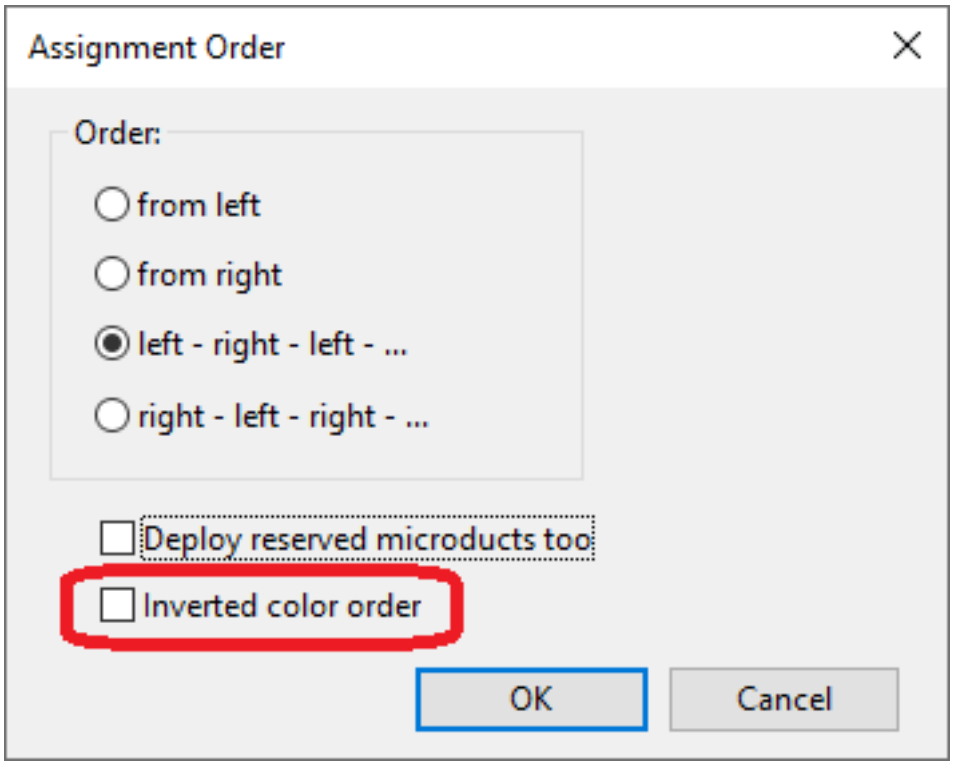
If the flag "Bitmap data field" for the data field containing the variable is enabled, the color text is converted to a color bitmap.

The standard sample jetter plan report templates have been changed to contain the color bitmap fields.

Deploy microducts in reverse order

Use case:
A microduct package has 24 microducts. The first 12 will be continued in the next station with a 12-microduct package. Block deploy in AND should allow to deploy the microducts 13-24.

In the dialogue for the options of assignment there is a new option "Inverted color order".



If the option is off, the microducts are deployed in the order given by the library 1, 2, 3, 4 etc.

If the option is check on, the last microducts in the library are deployed first, e.g. if the microduct package has 10 microducts in the order 10, 9, 8, 7 etc.

Cable Pull Report, Cable Length Report

1 Abstract

For AND client AND 4.19 two new reports are implemented: The **Cable Pull Report** and the **Cable Length Report**.

Both reports list the path of a given cable in the site plan.

Stations for that report are sheets or MUCs in the GIS-MainSheet.

Segments of the report are the cables between those stations. Cable pieces inside the stations are reported as reserve length.

Difference between both reports:

The **Cable Pull Report** shows only segments where a cable is inside a trench duct. Main purpose is a working plan for the field crew pulling the cable into the duct.
In the pull report cables without duct or within microducts are irrelevant and are not displayed.

The **Cable Length Report** shows all segments of a cable in the site plan. For the length report it doesn't matter if the cable is inside a trench, trench duct, microduct etc.
The cable length report replaces the old report Line Length Plan, which was removed from AND.

2 Content of the reports

As headline the name (CableNo) of the cable and the library component name are displayed.

Below the headline the list of segments is shown.

For both sides of the segment the data at the entry point are:

- Data of the surrounding duct or microduct. For trench ducts it can be configured which data is displayed.
  By default are shown:
  - Name of the duct
  - Component name of the duct (name in the library)
  - Marker line (text attribute of the duct)
- Data of the station
  - Name of the station (hierarchy path)
  - Address of the station

- Data of the cable
  - Total cable length counted from the start point
  - Cable metrization value at that point (optional, entered by user)
  - Reserve length inside station. The reserve length is determined by AND. E.g. if inside the station a cable flowing out of the trench expander has a length of 2m, the reserve length in the report is 2m. If inside a sheet-station a cable is passing uncut, the reserve length assigned to sheet-entry and sheet-exit is halved.
- Name and component name of all other cables in the same trench duct. Knowing the parallel cables is important, because normally cables are pulled in together.

The cable path is oriented from higher to lower location hierarchy value of the two end objects. This is the hierarchy value entered in symbol of the station (MUC or Sheet-Symbol).

If a location rectangle is drawn around the sheet connector inside a sheet, the hierarchy value of the location rectangle is taken.

3 How to start the reports in GUI

The report can be created:

1. For a single cable from context menu.

   Normally this is called for the cable piece connected to the trench connector or sheet connector.

2. For a trench line in the main GIS sheet from context menu.

   In this case all cables inside the trench line are part of the report. This includes cables instantiated in the trench line.
   For cables assigned to a trench duct but drawn in parallel, no report is possible. Those cables are listed as parallel cables.

3. For a selected block from context menu.

   In this case all cables within the block are part of the report.

4. For the whole project from project data menu.

   This can be called via menu „Project Data“, item „Project Lists...“ selecting the report template „CablePullReport.rep“ or "CableLengthReport.rep".

4 How the path of a cable is built

For a given cable AND determines the path by walking along 1:1-connections and trench lines and micro ducts (length report only). AND only walks along the whole cable, not of parts (wires etc.).

If a cable is terminating inside a station or passing it only partially, AND is searching for a matching continuation cable connected to same SheetSymbol or MUC.

An opposite cable is regarded as matching if it has the same name and the same library component. If a matching cable is found, the path walk is continued. This way it is possible that the length of a cable in the pull report is longer than the total length shown in AND (e.g. in Q-function or crosstext). A pull report over project borders (linked EEPs) is not implemented.

If a duct is passing a station uncut or plugged, the station is ignored in both reports. In case of plugged ducts, it is possible that the ducts on both ends of a segment are different.

A station without internal connections appears in the report. If it is wanted that such a station is ignored, a connection on duct level must be established by user (In MUC enter a connection from duct to duct, in trench sheet connect the trench expanders by a trench connector: stock object from toolbar).

5 Report template

5a Overview

The data can be displayed in two new types of report, the **cable pull report** and the **cable length report**.

As usual the report **header variables ~_Report*** provide information about the selection, filter, sort order and for the cable pull/length report also a summary of the data in the report (summary of lengths per cable type).

Sorting of the cable pull/length reports is done by location hierarchy/address of the cable start. Filtering allows to select for tasks (the end points of the segments in the stations are used to determine the tasks).
Filters are provided for the cable type, length and cable data (total wires, bundles, fibers) and for segment numbers and lengths. For lengths, counts and cable data, ranges may be entered as a filter with no value given meaning no boundary.

The output is grouped per cable, followed by the segments of the cable.
As for the splice/patch report, the **group variables ~_Group*** are used for the group header data (cable data) and for the cable pull/length reports the **variables ~Segment*** are used to provide data of the various segments of the cable.
Also as usual, _Group and Segment variables are filled alternatingly (only one of _Group / Segment empty, fields in report can be on the same place in the data section).

Default cable pull/length report templates are provided.

The existing jetter plan report may now use the same variables as the cable pull/length reports, but for the jetter plan report the old variables Jet* are still provided. This enables to have only one template design and save it with different report types (jetter / cable pull / cable length). Some report variables like metrization marker and parallel cables are empty for the jetter plan report.

5b Report variables

As usual, report variable names have to be prefixed by a "~" to be recognized as report variables in report header/data fields. In addition to the specific variables for the cable pull/length reports, the "AND: PROJECT DATA" variables can be used (~ProjectNumber etc.) – refer to the relevant documentation for details.

Variable	Content	Note
_ReportCable Filter	Description of the filter	Nothing is listed if a filter is not provided
_ReportCableCount	Total number of cables in the report	
_ReportCableInstNos	List of the cable numbers	e.g. CB01, CB07 etc.
_ReportCableTypes	List of the cable types and lengths	e.g. Fiber 1x24 (2: 200.3 + 298.3 = 498.6m), Fiber 2x12 (4: 308.6 + 95.9 + 95.9 + 346.8 = ...
_GroupCable Name	Name of the cable	
_GroupCableType	Type of the cable	
_GroupCableLength	Total length of the cable with reserve	
_GroupCableLengthSegOnly	Total length of the cable segments (only)	
_GroupCableSegmentCount	Number of segments of the cable	
_GroupCableTasks	Tasks of the segments of the cable	Segment tasks from station pins
_GroupCableSelectedTasks	Tasks for which cable was selected	Intersection with filter
_GroupCableAddressFrom	Address at start of first segment	
_GroupCableAddressTo	Address at end of last segment	
_GroupCableLocFrom	Location hierarchy at start of first segment	
_GroupCableLocTo	Location hierarchy at end of last segment	
_GroupCableSheetFrom	Sheet at start of first segment	Empty if top sheet, e.g. 1: sheet
_GroupCableSheetTo	Sheet at end of last segment	Empty if top sheet
_GroupCableSheetBrFrom	Sheet at start of first segment with brackets	Makes it more easy to use in one field, e.g. (1: sheet)
_GroupCableSheetBrTo	Sheet at end of last segment with brackets	

Variable	Content	Note
SegmentCount	Number of segments of cable	e.g. 6
SegmentPos	Segment number inside cable	e.g. 2
SegmentPosCount	Pos/Count	e.g. 2/6
SegmentLength	Length of segment (only)	
SegmentLengthReserveFrom	Reserve length at From side	
SegmentLengthReserveTo	Reserve length at To side	
SegmentLengthWithRes	Length of segment with reserve	
SegmentLengthFrom	Cumulative lengths of segments to From side	
SegmentLengthTo	Cumulative lengths of segments to To side	
SegmentLengthWithResFrom	Cumulative lengths with reserve to From side	
SegmentLengthWithResTo	Cumulative lengths with reserve to To side	
SegmentLengthRevFrom	Cumulative lengths of segments from end to From side (reverse)	
SegmentLengthRevTo	Cumulative lengths of segments from end to To side (reverse)	
SegmentLengthRevWithResFrom	Cumulative lengths with reserve from end to From side (reverse)	
SegmentLengthRevWithResTo	Cumulative lengths with reserve from end to To side (reverse)	
SegmentParallelCables	Parallel cables in (direct parent) duct	
SegmentTasks	Tasks of the segment	
SegmentSelectedTasks	Tasks for which segment was selected	
SegmentDuct**Parent**sNameFrom	Duct path names at From side	**Duct Parent**s in which cable is; e.g. d1.d2; empty names replaced by duct type
SegmentDuctParentsNameTo	Duct path names at To side	
SegmentDuctParentsTypeFrom	Duct path types at From side	e.g. t1.t2
SegmentDuctParentsTypeTo	Duct path types at To side	
SegmentDuctParentsNameTypeFrom	Duct path names and types at From side	e.g. d1/t1.d2/t2
SegmentDuctParentsNameTypeTo	Duct path names and types at To side	
SegmentDuctParentsNameTypeMarkerFrom	Duct path names and types and markers at From side	e.g. d1/t1|m1.d2/t2|m2

Variable	Content	Note
SegmentDuctParentsNameTypeMarkerTo	Duct path names and types and markers at To side	
SegmentDuctParent1NameFrom	Direct duct parent name at From side	Empty if none
SegmentDuctParent1NameTo	Direct duct parent name at To side	
SegmentDuctParent1TypeFrom	Direct duct parent type at From side	
SegmentDuctParent1TypeTo	Direct duct parent type at To side	
SegmentDuctParent1MarkerFrom	Direct duct parent marker at From side	
SegmentDuctParent1MarkerTo	Direct duct parent marker at To side	
SegmentDuctParent2NameFrom	Upper duct parent name at From side	Empty if none
SegmentDuctParent2NameTo	Upper duct parent name at To side	
SegmentDuctParent2TypeFrom	Upper duct parent type at From side	
SegmentDuctParent2TypeTo	Upper duct parent type at To side	
SegmentDuctParent2MarkerFrom	Upper duct parent marker at From side	
SegmentDuctParent2MarkerTo	Upper duct parent marker at To side	
SegmentMetrizationFrom	Metric marker at From side	
SegmentMetrizationTo	Metric marker at To side	
SegmentAddressFrom	Adress at From side	
SegmentAddressTo	Adress at To side	
SegmentLocFrom	Location at From side	
SegmentLocTo	Location at To side	
SegmentSheetFrom	Sheet at From side	Empty if top sheet, e.g. 1: sheet
SegmentSheetTo	Sheet at To side	
SegmentSheetBrFrom	Sheet at From side	With brackets, e.g. (1: sheet)
SegmentSheetBrTo	Sheet at To side	

5c Report options

Same as for jetter plan report.

5d Report output

If in the options, clickable report rows in the output window are chosen in addition to the segment sides (pin is displayed), the cable may be clicked: this will center and zoom to the cable segment collection and color it in bright green.

Unlink libraries via AND DataService

1 Overview

As an extension of our AND SOAPinterface service it is now possible to unlink libraries per project via a new interface function.

For this a new SOAP operation was implemented and exposed within the already existing Da-taService service.

2 Conditions

The library must be present and contain the lib-object with no essential changes. If these conditions are not met, an appropriate error message is logged and returned to the caller.

3 Methods

UnlinkLibraries

A SOAP operation implemented as part of the already existing SOAP service AND DataService. The new operation works on the FILE_ID or FILE_NAME of the project. It can be executed syn-chronously or in an async mode. When in sync mode, the "unlink-libraries" response is directly sent to the caller. In async mode and in case of success an info message is returned containing the opera-tion's TASK_ID (specifying that the current operation was successfully queued and is ready to be processed by a free AndApplicationService). This ID can be used within the command "Get-TaskStatus" to ask for more/detailed info of the unlink process.

Example Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cds="http://data.service.and-
cds.com/2008/04/CDS.NDS.Data.Service.BasicDataService">
  <soapenv:Header/>
  <soapenv:Body>
    <cds:UnlinkLibraries>
      <cds:InputData>
      <cds:SessionID>?</cds:SessionID>
        <!--Optional:-->
        <cds:OperationSyncMode>0</cds:OperationSyncMode>
        <!--Optional:-->
        <cds:FileName></cds:FileName>
        <cds: FileID>1976</cds:FileID>
        <!--Optional:-->
        <cds:ModifiedBy>Martel</cds:ModifiedBy >
      </cds:InputData>
    </cds:UnlinkLibraries>
  </soapenv:Body>
</soapenv:Envelope>
```

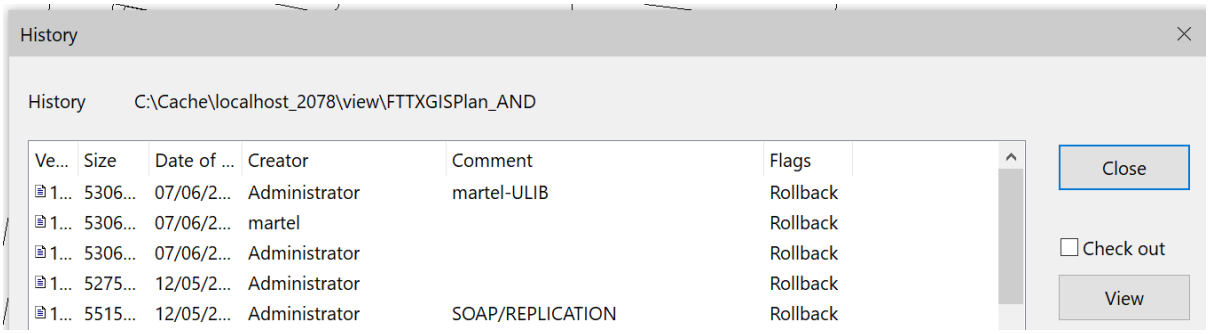
Example Response (OK):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <UnlinkLibrariesResponse xmlns="http://data.service.and-
cds.com/2008/04/CDS.NDS.Data.Service.BasicDataService">
      <UnlinkLibrariesResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Status>0:stOk</Status>
        <Message>Task successfully queued. Use GetTaskStatus for more info</Message>
        <TaskID>2022</TaskID >
      </UnlinkLibrariesResult>
    </UnlinkLibrariesResponse>
  </s:Body>
</s:Envelope>
```

Unlink Process

During the unlink process a verification of parameters takes place. On any problem with e.g. the source project or an invalid library the unlink will be skipped and a warning will be launched. The task in the NAT tasklist will be set to an error code and a log entry generated like described in the chapter "Logging".

On server side the NAT checks out the related project and unlinks all linked libraries. If there are no errors, a new project version will be created with the comment "[username]-ULIB". Otherwise, the operation is canceled.



If the targeted project is already checked out the execution is postponed and a NAT task will be cre-ated for the operation. The operation will then be executed asynchronously as soon as the project is no longer checked out.

On any call of the SOAP method "UnlinkLibraries" a new version of the related AND project is instantiated. The username for the check-in is determined by the SOAP login of the session calling "UnlinkLibraries". This login user must be a user existing within the AND SystemSolution user management. Additionally, a new attribute username gets handed into "UnlinkLibraries" and will be put to the comment field of the AND version history.

GetTaskStatus

Sample Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cds="http://data.service.and-
cds.com/2008/04/CDS.NDS.Data.Service.BasicDataService">
  <soapenv:Header/>
  <soapenv:Body>
    <cds:GetTaskStatus>
      <cds:SessionID>54e37ef2-8b9d-4362-9727-acb1536dff91</cds:SessionID>
      <cds:TASK_ID>37712</cds:TASK_ID>
    </cds:GetTaskStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response (OK):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetTaskStatusResponse xmlns="http://data.service.and-cds.com/2008/04/CDS.NDS.
Data.Service.BasicDataService">
      <GetTaskStatusResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <OUTPUT_DATA xmlns="">
          <STATUS>(0) No Error.</STATUS>
          <MESSAGE>(UnlinkLibraries)-->Info: All libraries were successfully un-
linked</MESSAGE>
        </OUTPUT_DATA>
      </GetTaskStatusResult>
    </GetTaskStatusResponse>
  </s:Body>
</s:Envelope>
```

Sample Response (NOT OK, already checked out):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetTaskStatusResponse xmlns="http://data.service.and-cds.com/2008/04/CDS.NDS.
Data.Service.BasicDataService">
      <GetTaskStatusResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <OUTPUT_DATA xmlns="">
          <STATUS>(167) Task blocked/postponed... project is already checkedout</STATUS>
          <MESSAGE>( UnlinkLibraries)-->Task blocked/postponed... project is already
checked-out</MESSAGE>
        </OUTPUT_DATA>
      </GetTaskStatusResult>
    </GetTaskStatusResponse>
  </s:Body>
</s:Envelope>
```

Sample Response (NOT OK, wrong/missing library):

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetTaskStatusResponse xmlns="http://data.service.and-cds.com/2008/04/CDS.NDS.
Data.Service.BasicDataService">
      <GetTaskStatusResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <OUTPUT_DATA xmlns="">
          <STATUS>(166) UnlinkLibraries failed, check log/db for more info</STATUS>
          <MESSAGE>( UnlinkLibraries)-->Error: Missing library 'libName' or object library
'libobject' or libobject was essentially changed
        </MESSAGE>
        </OUTPUT_DATA>
      </GetTaskStatusResult>
    </GetTaskStatusResponse>
  </s:Body>
</s:Envelope>
```

4 Logging

The replacement request and the result of the operation get logged. The trigger of the operation and the result might be two distinct entries in case the operation gets blocked by a checked out file.

Any operation will be logged and marked with the sessionID of the login, the new attribute username and the unlink information.

Sample in log file:

```
(role: 1) [NAT][pid: 31476, tid: 41200] Getting file from server (4243_1_VzK_59.top). User
'martel - SOAP(ExecuteXML)'
(role: 1) [NAT][pid: 31476, tid: 41200] Warning: (UnlinkLibraries) --> Error: Missing
library 'libName' or object library 'libobject' or libobject was essentially changed
Modified by: xxx
```

The same info is stored within the database (if logger is configured to log also there).

Please note that the error codes/messages may change during the final implementation.