What's new in 4.18?

www.and-solution.com

Table of contents

Splitter in MUC

1 Abstract

So far a multiconnector (MUC) in AND can hold only 1:1-connections.
From now on it is possible to create 1:n-connections for fibre wires.
An optical splitter (or attenuator) from the library can be assigned to fibre connections.
The attenuation of that splitter is considered by the power calculation.
In BOM, power calculation and dynamic labels the two connecting splices are counted.

2 Details

For a splitter in the multiconnector, there is a terminated IN connection, and multiple OUT connections.
In the editing GUI (see below), the IN connection is shown as parent element of the OUT connections.

Only the IN connection may be assigned a library object (see below), the OUT connections inherit that assignment.

Both IN and OUT connections have a new *name* attribute, which serves as splitter name (IN connection) and may serve as port name (OUT connection, prefixed implicitly by the splitter IN connection name).

The splitter shows up as a location hierarchy element, as does the port name:
the name of the splitter IN connection is used as a splitter object name if given, otherwise the splitter type (if library object given) shows "Splitter".
The input name of the splitter is taken from the splitter library object port identifier if given, otherwise "In".
The output names of the splitter are taken from the name (OUT connection) if given, otherwise "Out".
There is no relationship from the connection to the splitter library output port identifier, only the name in the multiconnector, so the identifier of the output ports in the library cannot be shown if no explicit name is given to the multiconnector connection.

The splitter is located implicitly in the container/tray of the connection (container of the IN terminated/root connection).
The OUT connections/splices may reside in different containers/trays (container of the OUT connection).
The location hierarchy displays the splitter location (IN connection container) as mounting/rack position in the dynamic label. The splitter type is displayed if given, otherwise "Splitter"

3 Editing

The editing GUI allows connecting one single wire with multiple output wires.
See chapter "Splitter Connection" in the documentation for the editing GUI.

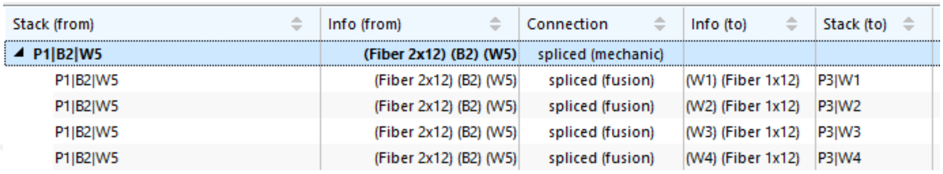
4 Validity of library object

The editing GUI allows to assign any library object to a connection.
That library object is counted in BOM.
But the attenuation and the two splices are counted only if

- it is a single fibre connection
- the library object is a symbol of the type "Optical Splitter", "Optical Attenuator" or "Optical Adapter"
- the symbol has exactly one input
- the attenuation of all outputs is equal. There is only one datapak in the library describing the unique attenuation of all outputs.

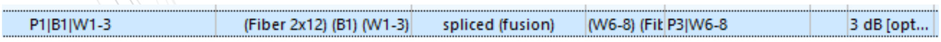
5 BOM

The splicing work assigned to splitter-connections count in BOM: e.g. the following splitter-output-connections count in BOM as one mechanic splice (for the input) and 4 fusion-splices for the 4 outputs:



Stack (from)	Info (from)	Connection	Info (to)	Stack (to)
◢ **P1|B2|W5**	**(Fiber 2x12) (B2) (W5)**	spliced (mechanic)		
P1|B2|W5	(Fiber 2x12) (B2) (W5)	spliced (fusion)	(W1) (Fiber 1x12)	P3|W1
P1|B2|W5	(Fiber 2x12) (B2) (W5)	spliced (fusion)	(W2) (Fiber 1x12)	P3|W2
P1|B2|W5	(Fiber 2x12) (B2) (W5)	spliced (fusion)	(W3) (Fiber 1x12)	P3|W3
P1|B2|W5	(Fiber 2x12) (B2) (W5)	spliced (fusion)	(W4) (Fiber 1x12)	P3|W4

If an optical attenuator/adapter is assigned to a fibre connection, two splices per wire are counted: e.g the following connections for Wire 1-3 count as 2x3 = 6 splices:



P1|B1|W1-3	(Fiber 2x12) (B1) (W1-3)	spliced (fusion)	(W6-8) (Fit	P3|W6-8		3 dB [opt...

6 Dynamic labels and reports

The dynamic labels in the multiconnector splitter behave roughly the way they do when drawing a splitter explicitly.

The splice report and interconnect plan (stations of one route) and interconnect plan multi (multiple routes, stations grouped by location) have two entries for each splitter connection: one for the IN side, one for the OUT side.
The splitter in the multiconnector is shown as next destination, like with drawn splitters.

Convert MUC to Sheet

1 Abstract

If a multiconnector (MUC) is converted into a sheet, the sheet is automatically filled with a network of components which are equivalent to the connections entered in MUC on data level.
For each side of each connection the necessary expanders and auto-splice-boxes are created.
For split fibre connections optical splitters are created additionally.
The leaves of the expanders are connected by standard lines, e.g. fibre-bundles, cables, and microducts.

2 Calling "Convert to Sheet"

Move the mouse over the MUC, click right to open the context menu and select the command "Convert to Sheet".
Shortcut for this command is 'a'.

3 Sheet-connectors

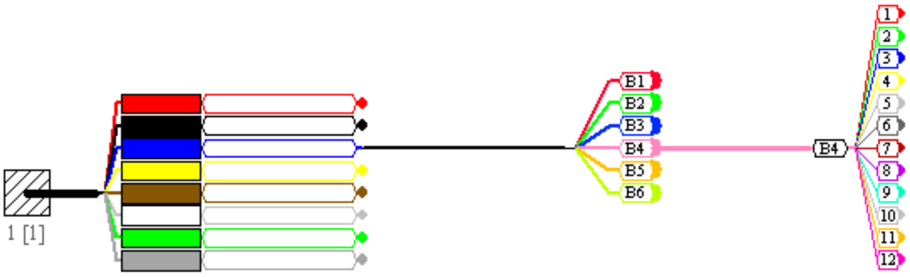
For each MUC-pin, which is connected to a line, a sheet-pin and a corresponding sheet-connector inside the sheet are created.
If the connected line is a trenchline, the sheet-connector is a trench-expander. In all other cases it is a normal sheet-connector.

The initial position of the sheet-connector inside the sheet is calculated from the corresponding position of the MUC-pin in the area of MUC-symbol.

A special area is the sheet-legend. If a sheet-connector would be placed inside the sheet-legend, it will be automatically moved horizontally aside.

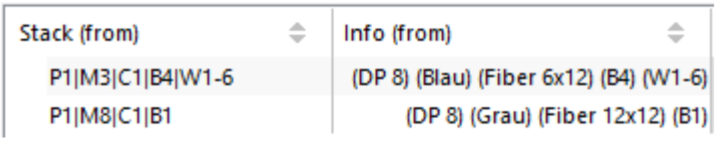
4 Expander-tree

For all connections inside the MUC the necessary expanders are created: e.g. if the stack of one side of the connection is (M3|C1|B4|W...), the created expanders could look like this:



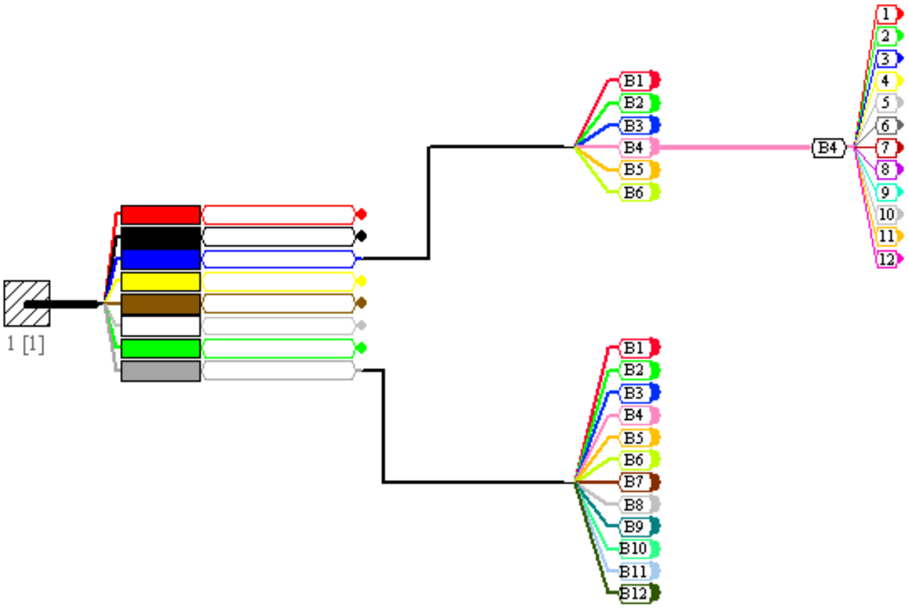
Picture 1: Expanders for connection-stack (M3|C1|B4|W...)

If there are several connections to the same MUC-pin, this will result in an expander-tree, e.g. for the following 2 connections:



Stack (from)	Info (from)
P1|M3|C1|B4|W1-6	(DP 8) (Blau) (Fiber 6x12) (B4) (W1-6)
P1|M8|C1|B1	(DP 8) (Grau) (Fiber 12x12) (B1)

The result will be as shown:



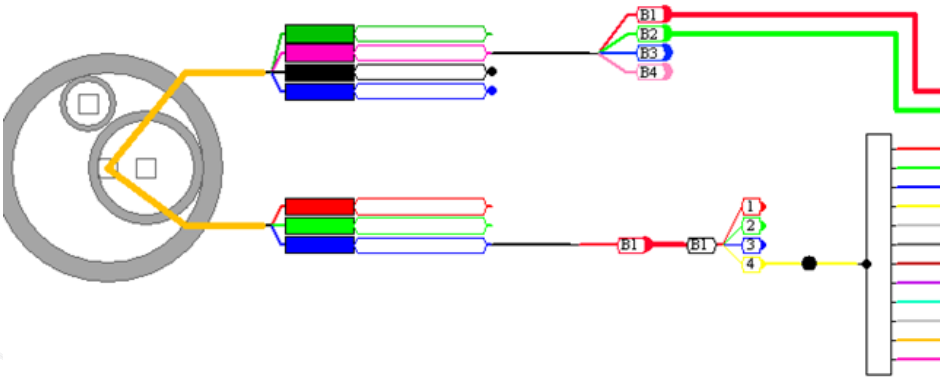
Picture 2: Expander-tree for the 2 connections shown above

The expanders can be only created if the corresponding line objects are really connected to the initial MUC-pin.
E.g. if the cable "Fiber 12x12" inside microduct 8 wouldn't exist, the lower autosplicebox in picture 2 would be missing.

- The tree is arranged in a balanced way so that the maximum coordinate extent upward is nearly equal to the extent downward.
- If an expander-tree would be partially out of sheet bounds, the tree is moved inside. Such a moved tree isn't coordinated balanced any more.
- As first try the initial positions of the sheet-connectors are used. If the trees of neighboring sheet-connectors would overlap, the sheet-connectors of each side are rearranged. If rearrangement doesn't solve the overlap, because the joint expander-trees are too large, the whole sheet is enlarged.

5 Trench-expanders

If a trenchline is connected to the MUC, a trench-expander is created. Trench-expanders are special, because the cables must be pulled out and the sheet is formally not schematic any more.



Picture 3: Example of an expander-tree with a trench-expander as root.

6 Optical splitters as "expanders"

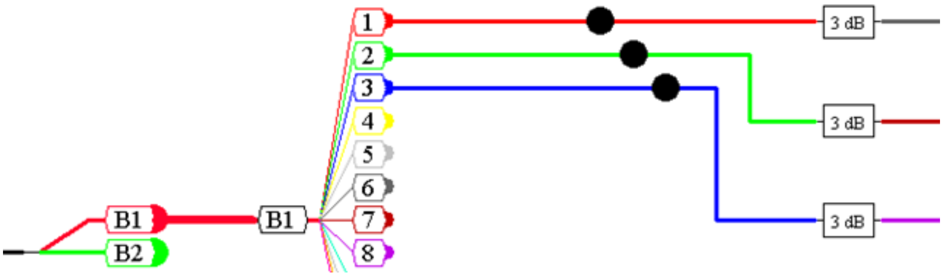
6.1 Splitters

If several output wires are connected to the same input wire, a splitter is created as expander. Picture 3 above shows such a splitter in the bottom right corner.

The library object entered in the MucConnection is taken over. If the MucConnection doesn't hold a valid library object, default splitters are used. Default splitters exist with 2, 4, 8, 12, 16, 24, 32 or 48 outputs. The splitter creation would fail, if in the MUC a splitter connection would be defined with more than 48 outputs and no library object is entered. In this case a warning is shown, which informs the user and allows to cancel the sheet conversion.

6.2 Optical attenuators and adapters

If an optical attenuator or an optical adapter from the library is assigned to a single wire connection, the symbol for it will be created.



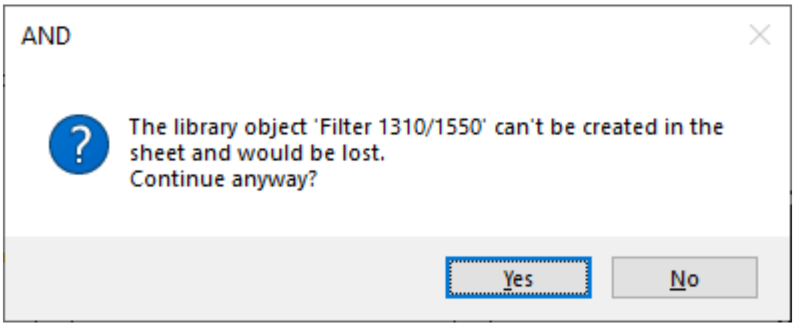
Picture 4: Created MucConnections with 3 dB attenuator

6.3 Restriction of library objects

For the conversion into a sheet the library objects assigned to MucConnections must fulfill the following requirements:
Every object

- is a symbol of type
  - optical splitter
  - optical adapter
  - optical attenuator
- has only one data-pak (same attenuation for all outputs)
- has exactly one optical input
- in case of attenuator/adapter: has exactly one optical output.

If library objects are used, which don't fulfill these requirements, the following message box appears:



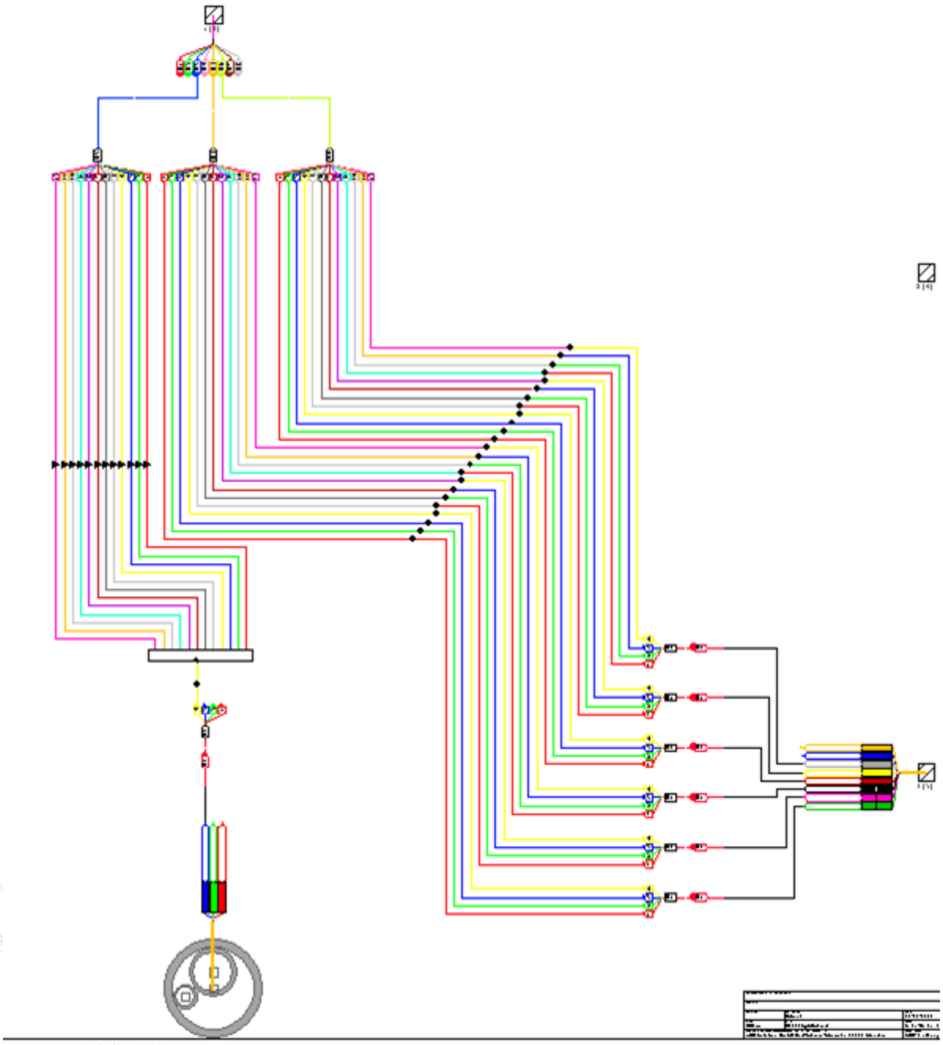
AND

The library object 'Filter 1310/1550' can't be created in the sheet and would be lost.
Continue anyway?

Picture 5:
Pressing 'Yes' causes the sheet loosing that/these library object(s).
Pressing 'No' cancels the sheet conversion and allows user to check which library objects would be lost.

7 Connecting lines

After the expanders are positioned, the connecting lines defined in the MucConnections are created. If possible, all lines between a given pair of pins are connected within a congregated ribbon consisting of all individual lines. E.g:



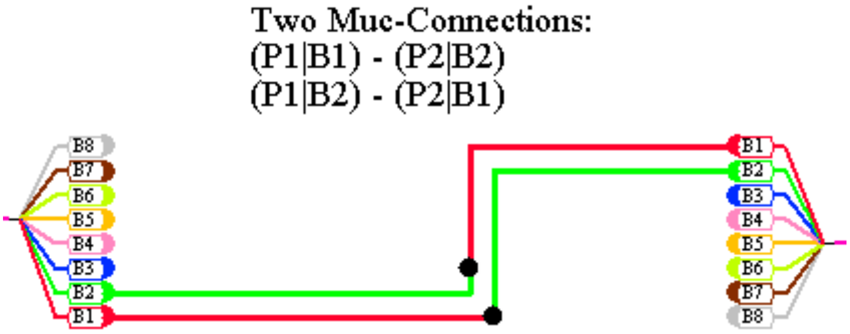
Picture 6: Example of a result of the function "Convert to sheet".
In this example two ribbons are shown:
One for the connection from the bottom sheet-connector to the top sheet-connector.
A second one for the 6x4 connections from the right sheet-connector to the top.

8 Crossing reduction

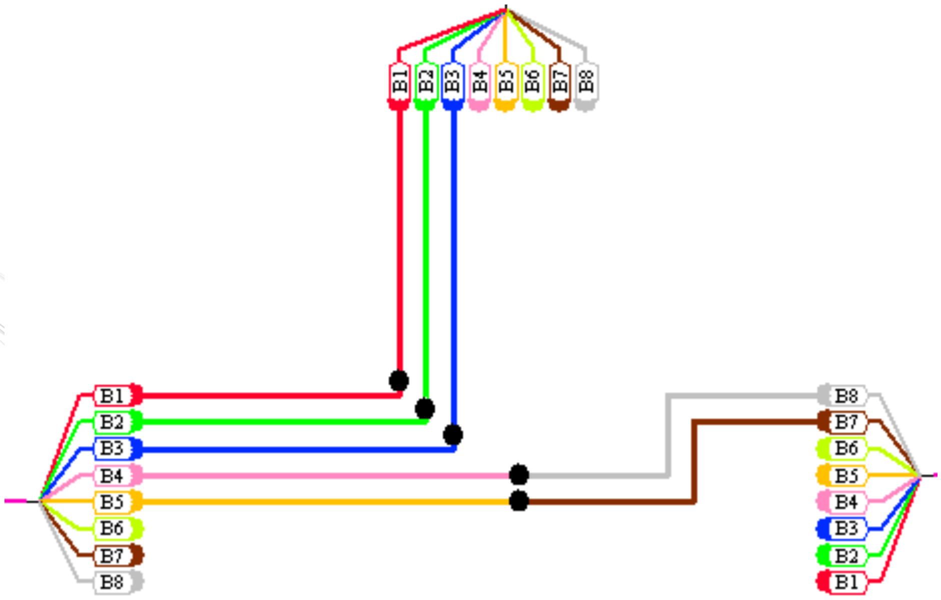
In general the connecting lines will cross each other.
The following methods are used to keep the number of crossings as small as possible:

8.1 Automatic mirroring (swapping) of expanders

Expanders and autospliceboxes have two representations: a normal one and a mirrored one with inverted sequence of the leaf pins. Two expanders oriented face-to-face have usually opposite mirror states. An extra mirroring is applied, if that results in less crossings.

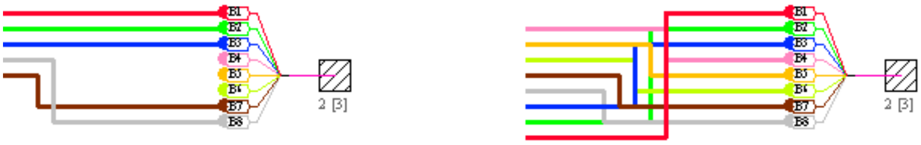


Picture 7: The left expander is created mirrored to avoid crossings of the bundles



Picture 8: The right side expander is created with mirror-state inverse to normal.

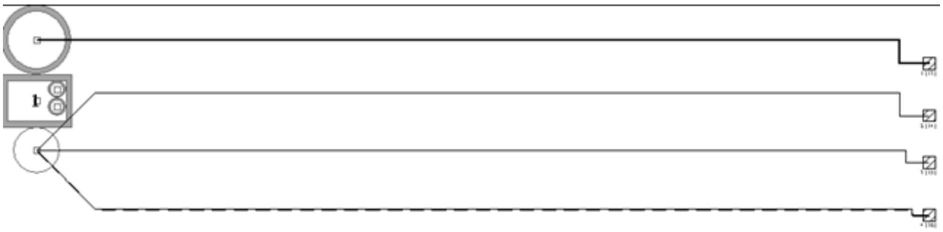
In general crossings can’t be avoided. In this case the mirroring is chosen to reduce the number of crossings.
An optimization of the line path in case of crossings isn’t implemented. In some cases the result looks clearly arranged, in others not, especially in complicated cases.



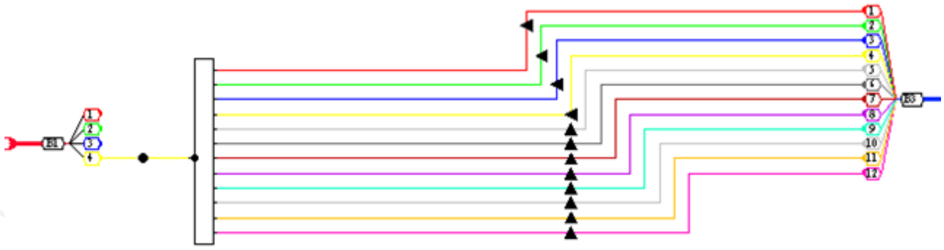
Picture 9: Clearly arranged (left) and not clearly arranged (right).

8.2 Trench-expanders and splitters

If there are several cables in the same trench-duct, the cables are pulled out in a sequence which reduces crossings. The same applies for splitters.



Picture 10: The 3 cables in the lower duct are pulled out so that there are no crossings



Picture 11: For splitters it is arbitrary/undefined which wire is connected to which output. So the wires are connected in a sequence which avoids crossings.

8.3 Limitations

Finding the optimum path of connecting lines in any situation would be very complex. In some cases there will be unnecessary crossings and overlaps.
The same applies to automatic mirroring. In complex cases, the mirror states can be suboptimal.

9 Data inheriting

9.1 Cables, microduct-packages

The small line pieces, which connect sheet-connectors and expanders (Cables, duct-packages), automatically inherit the data from the counter parts outside.
If the source for the line is a cable instantiated in TrenchDuct (CIiD), the data (e.g. installation number) are copied from there and the CIiD is deleted automatically.
The same applies to A-function on sheet-connector-pin or MUC-pin.

Additionally the cable inherits the reserve length from the MucConnection (in case of cable in duct-package, the cable gets the reserve length, not the duct-package)

9.2 Bundles

The bundles connecting the leaves of 2 expander trees inherit the following data of the MucConnection:

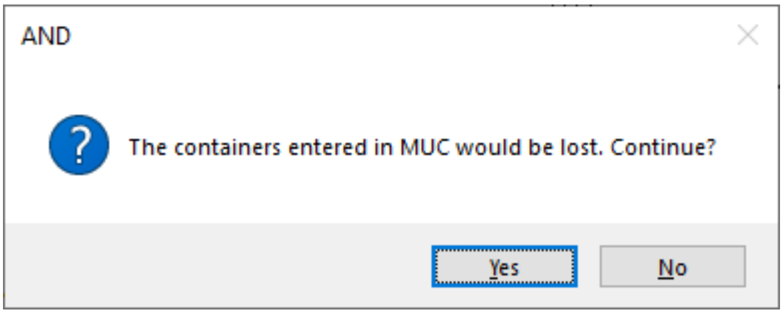
- Connection type (e.g. “spliced”)
- Task
- State
- Deadline
- Installation number
- Comment (will be migrated into the “description” of the bundle)

9.3 Splitters and attenuators

If a Splitter or attenuator is created for a MucConnection, the created symbol inherits the installation number and the task from the MucConnection.

9.4 Container information would be lost

The containers of the MUC aren't created in the sheet. If the MUC contains containers, e.g. splice-cassettes, these data would be lost. If there are any containers entered in the MUC, the following message box appears:



AND

The containers entered in MUC would be lost. Continue?

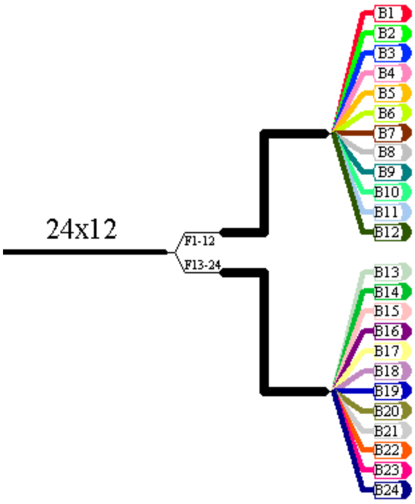
Picture 12:
Pressing 'Yes' creates the sheet loosing containers.
Pressing 'No' is cancelling sheet conversion.

9.5 Location rectangle

A location rectangle is created automatically inside the sheet.
That rectangle is enclosing all except the sheet legend.
The MUC-data shown in the dialog "Location/InstNo" are automatically transferred to the rectangle. e.g. address, installation number etc.
If the current crosstext rule for location rectangles leads to a non empty text, a crosstext for the rectangle is created and positioned at the center below the rectangle.

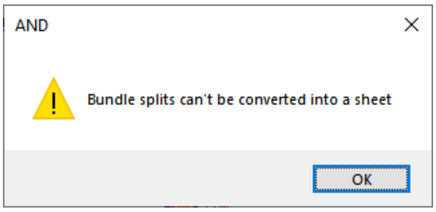
10 No bundle splits

A MUC can be used as a logical bundle split. E.g.



Picture 13: The MUC splits the 24 bundles in 2 groups F1-12 and F13-24.

If the MUC contains a bundle split, the conversion is refused showing the following messagebox:



Bundle splits can't be converted into a sheet

Picture 14: Messagebox

The reason for this is that a MUC is the only object to realize a bundle split.
The conversion would create a sheet and inside the sheet once more the same MUC.

Single microduct continuation object

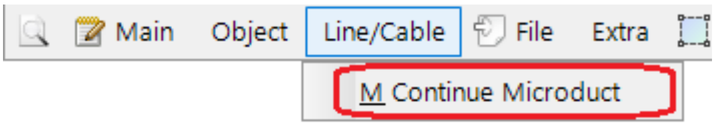
1 Abstract

Microduct continuation objects (MCOs) can be used to connect two microduct expanders on microduct level.
Additionally they can be used to continue a microduct inside a subsheet or to create a microduct connection between MUCs or label objects.

MCOs inherit color and linestyle from the connected (micro-)duct-package.
A direct creation as standalone object from symbol selection is not implemented.
MCOs are schematic objects, which can't be created in a GIS-sheet.

2 Creation

MCOs are created from the context menu of a selected pin.
The command is "Continue Microduct", shortcut ‘M’.



The command appears only, if the selected pin is open and a microduct of a duct-package is linked to that open pin, in the sense that a microduct of a parent duct-package can be reached by routing on 1:1 connections.

On that command AND is switching in the line-creation mode. The MCO is hanging on the mouse, the first line point is already placed at the start pin and every click is adding a line point.

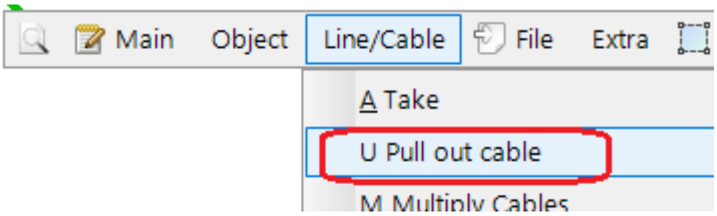
"M Continue Microduct" can be called on open pins of the following objects:

- Sheet-connectors, with a microduct leg connected outside (also cascaded: leg – sheet-connector - MCO – sheet-connector etc.)
- MUC-pins to which the MUC-connections lead a microduct
- Leaf-pins of microduct expanders
- Label-objects, if on the opposite side is a microduct leg or MCO.

Note the difference to command "A Take" for symbols: For a sheet-connector, MUC or expander the command 'A' pulls the cable out of the microduct, while 'M' continues the microduct.

For label-objects 'A' always creates a continuation, but in order to be consistent, 'M' is the command for continuation, if on the opposite side is a microduct leg.
"M Continue Microduct" isn't available for pins of a GIS-sheet or a sheet with meter-scale.

For MCOs the command for pulling out a cable is "U Pull out cable", because "A Take" is already occupied for switching into the edit mode for line points.
This is the same for leg-pins of duct-packages.



3 MCO features

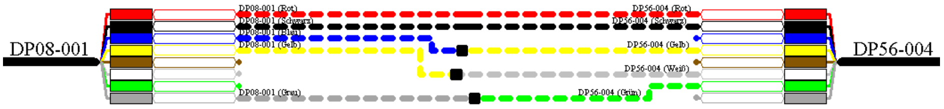
MCOs inherit the color from the microduct of the connected (remote) duct-package.
If the MCO is a continuation of a deployed leg, the line style is inherited too, otherwise it is dot-dot.

If color and line style for both sides are different, the line is split into 2 parts, each part showing what microduct is on the corresponding side.
For patched MCOs a point label is drawn at the transition point of both sides. The label can be moved similar to bundles.
For a MCO the only possible connection types are uncut and patched.
MCOs hanging on the mouse are automatically set to patched, if both pins are connected to different sides.

The automatism is only in this direction: A MCO can be set to patched even if both sides are equal.
Only the microducts are patched.
A cable inside is continued uncut.

If the MCO has lost the connection to the ductpackage, line style and color remain in the state of the last moment of connection.

MCOs display the name of microduct and duct-package as small text at the end of each side. These texts ("Cable numbers") have a fixed size and can't be moved. Display can be switched on/off for the whole sheet with the checkbox "Cable numbers" in the dialog "Sheet View settings"



4 Calculations, BOM

A MCO can hold a length.
MCOs are supported by all calculations.
If the MCO has a length > 0, the cable inside inherits the length.

If a MCO is patched only the microducts are patched.
A cable inside is continued uncut.

A MCO counts in BOM in the following way:
- The patch (working) costs are counted according to splice/patch costs for fibre bundles
- The connector component of the patch is the connector entered in the libobject of the duct-package.
- If length > 0, the microduct component is the subduct entered in the libobject of the duct-package.
- If length > 0, the cable inside the microduct is counted in BOM too.

5 Restrictions, warnings, compatibility

A MCO can't be in a trenchline. Connecting a MCO to a trench-expander is regarded as error.

If different microducts are connected by MCO or by a chain of MCOs, netcheck shows a warning if none of the MCOs is patched.

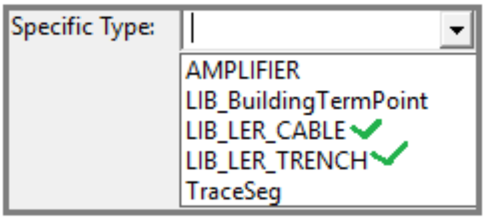
In previous AND versions formal fibre bundles are used for connecting microduct-expanders.
If both pins of the formal bundle are connected to microduct-expanders AND recognizes that bundle as connection on microduct level.
This former documentation style is still supported for compatibility reasons.
An automatic conversion isn't implemented.

LER2: Replacing library

AND LibEdit

When replacing a library, the new dots are created, deleted or modified, according to the specific type set.

The LER specific type can be set directly in LibEdit.exe for cables (trenches, duct-package, hybrid, cooper, fiber, coax):



Specific Type:

- AMPLIFIER
- LIB_BuildingTermPoint
- LIB_LER_CABLE ✓
- LIB_LER_TRENCH ✓
- TraceSeg

Supposedly the user has already a project with cables, with/without specific type = LER.

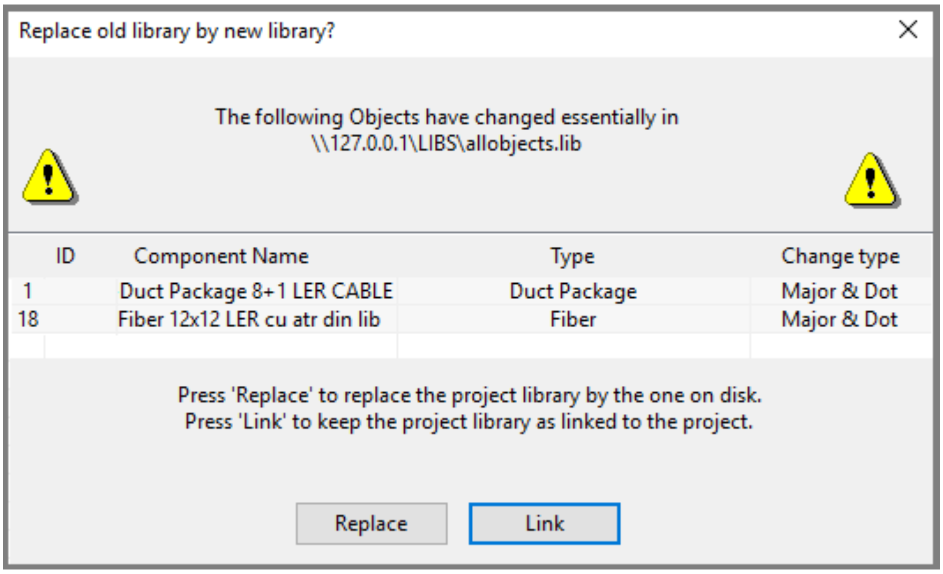
Next the user might decide to make some changes inside the library like:

1. add SpecificType=LER for a cable that did not had specific type = LER before
2. delete SpecificType=LER for a cable that had specific type = LER before
3. modify SpecificType=LER from one type to another

The library needs to be saved.

AND Client

When opening a project in the AND Client where one or more objects of a library are used where the specific type was changed the following dialog is displayed:



Replace old library by new library?

The following Objects have changed essentially in
\\127.0.0.1\LIBS\allobjects.lib

ID	Component Name	Type	Change type
1	Duct Package 8+1 LER CABLE	Duct Package	Major & Dot
18	Fiber 12x12 LER cu atr din lib	Fiber	Major & Dot

Press 'Replace' to replace the project library by the one on disk.
Press 'Link' to keep the project library as linked to the project.

The user now can decide which objects of the list shall be adjusted.
To get them propagated to the LER2 Webservice the user needs to press "Replace".

This dialog will be displayed in the following cases:

- project closed + change lib + project reopen
- project stays open while the lib is changed
- project closed + change lib + project check-out + project check-in (without opening the project).

Every replace of an object processed generates an entry in the output window of the AND Client with the possibility to jump to the changed object:

Info: Obj 205 - New dot of type LER_Trench_NCW created
Warning: Obj 198 - Dot of type LER_Cable deleted

Info: Obj 6 - new dot of type LER_Trench_NCW created, replacing LER_Cable

The object was modified after loading. Please save the project or check-in before you start a new search.
Drawing ok.

To get any changes propagated into the LER tables in database, the user must check-in the project.

Fixes and improvements for optical EEPs

Fixed bugs:

Bug 1:
If you set in AND LibEdit a wire to output or bidirectional, but enter no signal for it, in AND a signal with default wavelength = 1310nm is created for that wire.
From now on no signal is created.

Bug 2:
If you do Group-Replace (press g when mouse on ONT), the signals were doubled.
This is fixed from now on.

Improvements:

Improvement 1:
In AND LibEdit the list of possible directions of a wire of an optical EEP is extended by "Reserve".

With this setting, the wire is currently unused, but reserved for future use.
A wire of type "Reserve" isn't checked for input level and doesn't create an output signal by default.

Improvement 2:
For wires of optical EEPs from lib it is from now on possible to add a signal, except for wires with direction "Input".

WMTS in AND WMS Client

In short:
The WMTS is implemented in a way it can be used (nearly) exact the same way as WMS.
It is even possible to use WMS- and WMTS-datasources side-by-side within one *MAPDOC*.

There are only very little differences:

- In the layer-tree layers within a WMTS datasource are used exclusiveley, so only one layer can be turned visible at a time.
- In the datasource-options there's a configurable **Render quality**, which affects reprojection-performance of maps by using different resampling algorithms.

Q-Function running through splicebox without internal connection

Remarks:
The implementation works on pinId, not on pin position.

If in the replacing libobject pin 2 and pin 3 are exchanged, the connections to the cable at former pin 2 are now the one which were initially assigned to pin 3.

It is recommended to not exchange the pins in AND LibEdit.