SIEMENS

COMOS

Process P&ID

Operating Manual

08/2011

A5E03640196-01

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the relevant information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03640196-01
Ⓟ 08/2011

Table of contents

Trademarks

1

Trademarks

Registered trademark: COMOS®

Introduction

2

Overview

The "P&ID" module is a tool for quickly and effectively creating pipe and instrumentation flow diagrams.

Creating drawings

Drawings are easy to create using drag-and-drop. The module "P&ID" toolbars and tools are provided for drawing. Predefined objects for filters, mixers, cooling equipment, and so forth are available in a comprehensive library.

Alternatively, you can create your own objects.

All objects can process and transport data. Objects are also linked to other objects.

Process data

No process data is calculated within COMOS. Usually, this task is handled by simulators.

Objects

The "P&ID" module provides its own object class for the use and management of pipes. There are many auxiliary objects for pipes, including page references, pipe breaks, and labeling symbols. All auxiliary objects react automatically to changes in data or changes in flow diagrams.

Action lines are also supported and are offered as connections without objects.

In addition, the functional scope includes automatic equipment and material flow bars and specialized measurement functions.

See also

Controlling the apparatus bar/substance stream bar of the P&ID (Page 130)

Preparations

3

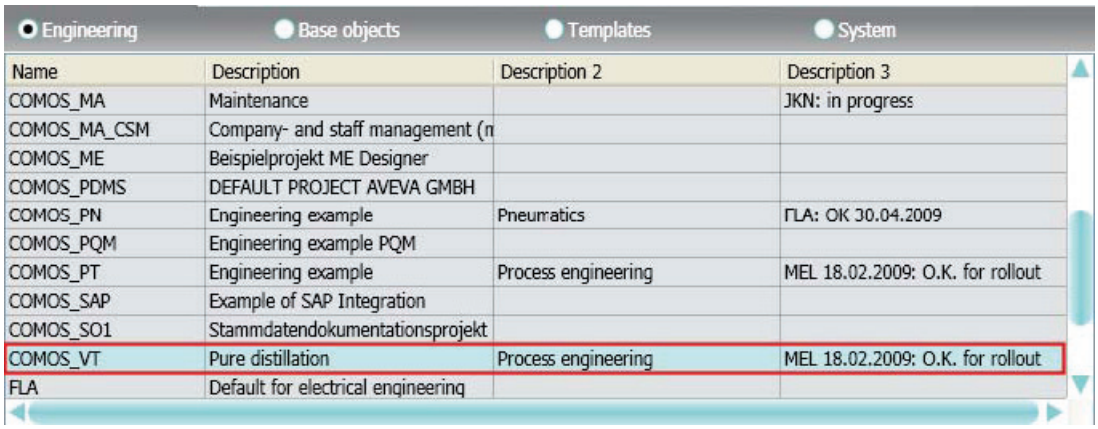
3.1 The example project

Requirement

A COMOS DB is open.

Preparation

Open the "COMOS_VT Pure distillation" project in COMOS.



Name	Description	Description 2	Description 3
COMOS_MA	Maintenance		JKN: in progress
COMOS_MA_CSM	Company- and staff management (n		
COMOS_ME	Beispielprojekt ME Designer		
COMOS_PDMS	DEFAULT PROJECT AVEVA GMBH		
COMOS_PN	Engineering example	Pneumatics	FLA: OK 30.04.2009
COMOS_PQM	Engineering example PQM		
COMOS_PT	Engineering example	Process engineering	MEL 18.02.2009: O.K. for rollout
COMOS_SAP	Example of SAP Integration		
COMOS_SO1	Stammdatendokumentationsprojekt		
COMOS_VT	Pure distillation	Process engineering	MEL 18.02.2009: O.K. for rollout
FLA	Default for electrical engineering		

All the settings that are useful for a P&ID scheme have already been set in this project. To change the defaults, contact the responsible administrator.

A unit has already been created in the example project.

3.2 Creating a unit structure

Introduction

Usually, the unit structure already exists or is created by employees in other departments.

Procedure

To create a new unit, proceed as follows:

1. Select the "Units" tab in the Navigator.
2. Right-click on the project.

3. Click on "New" in the context menu and select the required object.

   All objects underneath the separating line in the context menu have been prepared so that you can work with them immediately without any need for further preparations.

4. Select the menu entry "U01 Factory/Building/Production EN Standard" to work with the same unit structure that is used in this manual.

   The unit is created in the Navigator, underneath the project root.

5. Right-click on the unit in the Navigator.
6. From the context menu, select the "New > Unit" command.

   A main unit is created.

7. Right-click on the main unit in the Navigator.
8. From the context menu, select the "New > Subunit" command.

Result

The unit structure is created.

3.3 Creating a document

Introduction

P&ID drawings, flowcharts, and other diagrams are also called reports in COMOS.

Procedure

To create a report, proceed as follows:

1. Right-click on the subunit.
2. Select the "New" command and the required report from the context menu.

Result

The report is created in the Navigator, below the subunit.

Predefined reports

Several predefined reports are available in the context menu. Even a blank document features a number of automatic functions to assist you in your work. You find additional information on this topic in the "Reports - Basic Operation" manual and in the "Basic" Quickstart, key word "Reports".

3.4 Overview of P&ID objects

When you work with the P&ID module, you use the following types of object:

P&ID base objects

COMOS is supplied with a comprehensive library of preconfigured P&ID objects. Each P&ID object that you create in the engineering project is based on a base object.

See also section P&ID objects (Page 21).

Pipes

Pipes are P&ID objects. See also section Pipes (Page 29).

Functions

Functions are objects that constitute an interface between the P&ID area and the I&C area. They have numerous automatisms and are also handled separately. See also section Functions (Page 61).

P&ID text symbols

P&ID text symbols are tags that are added to components. Objects of this object type exist only on the report but not in the Navigator. They only read the data of the component they are connected to and output it on the report. For that reason they are also called flags. They can contain both pure text and other graphical elements.

A number of different labeling symbols have been predefined in base object node "@01 > PID > 99 > 01 P&ID labeling symbols" in the COMOS DB. Most of the predefined labeling symbols are for pipes. See also section Pipe flags (Page 43).

P&ID graphic symbols

Similar to P&ID labeling symbols, P&ID graphic symbols are objects that are purely graphical additions to a drawing, cannot be ordered, and are not evaluated in lists of objects.

However, differing from P&ID text symbols, an object is created in the Navigator for P&ID graphical symbols. See also section P&ID graphic symbols (Page 50).

Attribute texts

You can use drag-and-drop to move attributes to the report. The saved value is output. See also section Placing and editing editable texts on the report (Page 125).

You can customize the appearance of the attributes in the report. See also section Changing the appearance of the editable text in the report (Page 126).

P&ID objects

4

4.1 Start configurations

Project properties

Comos starts by searching the project properties of the open engineering project. The settings on the "Options > Process engineering" tab are relevant in this context. See also Control elements on the "Process engineering" tab (Page 262).

If you have not made any settings on this tab, Comos uses the settings from the base object.

4.2 Placing objects on the report

Introduction

When you place a P&ID object on an interactive P&ID report, the symbol of the P&ID object is output on the report. The symbol is the graphical representation of the object.

Placing objects

There are four ways of placing an object on a report:

- Via the base object symbol bar defined for the report
- Via the "Base objects" tab in the Navigator
- Via the "Units" tab in the Navigator
- By placing another report or a report template on the P&ID report

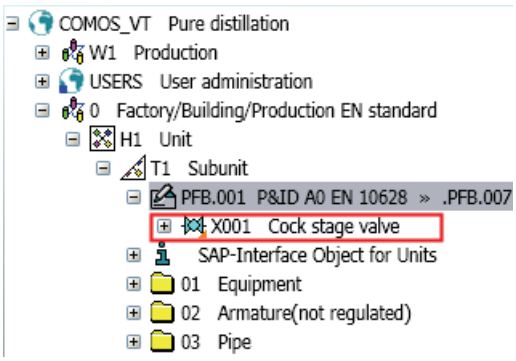
Objects in the Navigator

When you place an object from the base object symbol bar, from the "Base objects" tab in the Navigator, or from a report template on the report, by default, this object is created as an engineering object in the Navigator underneath the report.

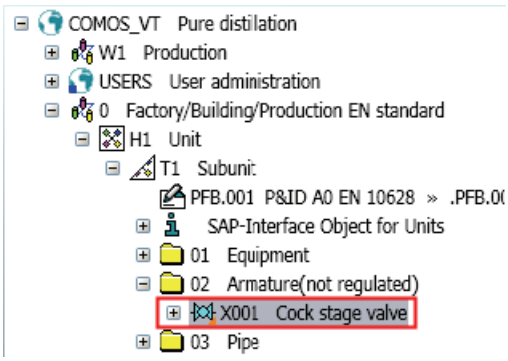
The administrator in charge can configure the report so that the objects in the Navigator are sorted into the folders underneath the subunit by category.

Example

Under document



In category



See also

Automatically sorting components (Page 94)

4.3 Connecting P&ID objects

4.3.1 Connectors

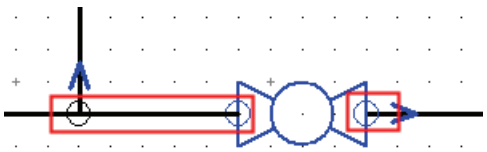
Overview

Connectors are COMOS objects that represent the connections of the P&ID object.

COMOS knows whether the connector is an input or an output and which connector type it has. This information is displayed in the connector properties.

Navigator and report

Connectors exist both in the Navigator and on interactive reports. In the Navigator they are always located below the P&ID object. In the interactive report they are symbolized by a small circle that is appended to the symbol of the P&ID object. This circle is also called a connector point.



Creating connectors automatically

Most connectors are created automatically with their component, regardless of whether or not the component is connected. For components which can have a variable number of connectors, the connectors are not created until you connect the component. These connectors are called dynamic connectors.

4.3.2 Connecting objects

4.3.2.1 Introduction

When you place a P&ID object on a report, you must decide whether you wish to connect the object to another P&ID object directly (via the connectors) or via pipes and measuring and action lines.

Furthermore, it is possible to connect the object immediately when it is placed, or to place it first and connect it subsequently.

4.3.2.2 Using connectors to connect objects

Procedure

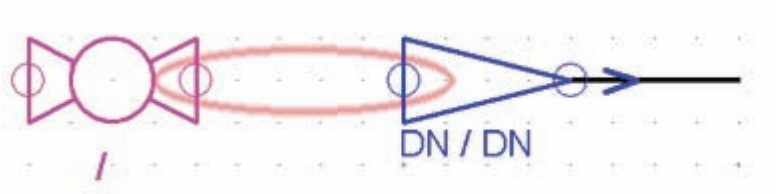
To use object connectors to connect P&ID objects directly, proceed as follows:

1. Select the symbol of the object on the report.
2. Use drag-and-drop to position the symbol so that a connector of the symbol is placed on the connector of another symbol.

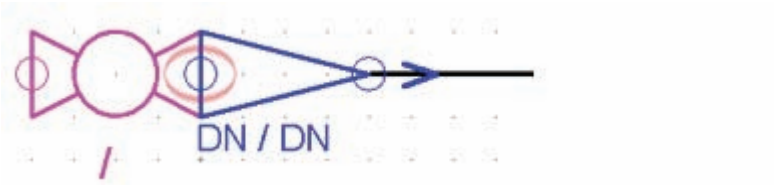
   The objects are interconnected by means of their connectors.

Example

Before drag-and-drop



After drag-and-drop



4.3.2.3 Using a pipe to connect objects

If you are working with functions, use measuring lines and action lines to connect objects. See also section Automatic separation of the pipe with fittings and intermediate components (Page 86).

Usually, objects on a P&ID are interconnected via a pipe.

To use a pipe to connect objects, select one of the following methods:

- If a pipe has already been drawn in on the report, drag the P&ID object onto the report and place it on the pipe.
- If there is no pipe in the report, place the P&ID object on a blank area of the report. Use the "Connection" tool to connect it. See also section Connecting pipes (Page 33).

See also

Functions (Page 61)

4.3.2.4 Rotating a P&ID symbol when inserting it in the report

COMOS automatically rotates P&ID symbols that you draw on existing connections in a report to ensure the correct flow direction. Alternatively, you can manually rotate the P&ID symbol in 90° increments when you insert it. To do this, press the space bar when inserting. Once you have manually rotated a P&ID symbol with the space bar, the automatic symbol rotation of COMOS for this action is overridden.

You can also rotate the P&ID symbols at a later point in time. See also section Miscellaneous (Page 127).

4.3.2.5 Using automatic assignment of line types

Requirement

The automatic assignment of line types is configured by your administrator. See also Configuring automatic assignment of line types (Page 257).

Procedure

To use automatic line types, follow these steps:

1. Place the P&ID object on the report.
2. Open the properties of the xx object.
3. Select the following tabs:
   – For components: "Attributes > System information"
   – For functions: "Attributes > System"
4. Select the required device from the "Device type" list.

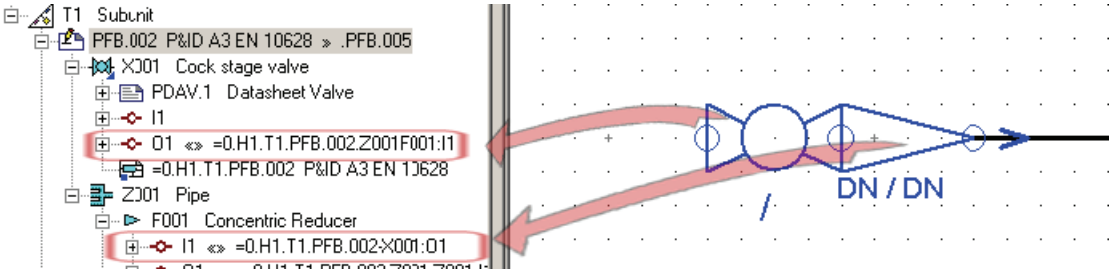
Result

If you connect two P&ID objects for which you have set the "Device type" attribute and defined an automatic line type, COMOS automatically generates a connection to the specified graphical properties.

4.3.3 Connectors in the Navigator

Overview

When you interconnect two objects on a report, this connection information is automatically written to the Navigator and to the connectors of the objects:



However, if two objects are interconnected in the Navigator, this connection information is not automatically shown graphically in the report. Interconnect the objects manually.

See also

Connecting pipes (Page 33)

Automatic separation of the pipe with fittings and intermediate components (Page 86)

Automated data flow (Page 95)

4.3.4 Data transfer via connectors

The connectors of two interconnected objects can exchange data.

Example

The technical data and the substance data of a pipe branch are automatically forwarded to all P&ID components that are connected to this pipe branch via their connectors.

See also

Automated data flow (Page 95)

4.3.5 Connection inconsistencies

Representation of connection inconsistencies

If inconsistencies occur in a P&ID connection report, COMOS shows these inconsistencies in red on the P&ID report and in the Navigator. This allows you to see connection inconsistencies at a glance.

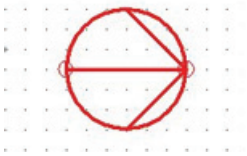
Note

For COMOS to display the connection inconsistencies in color, you need to set the report option `DrawScriptConnectorColor = true`.

The following inconsistencies are displayed:

- Component inconsistency

  A component is inconsistent when the associated engineering object no longer exists or the connection information to the engineering object no longer exists.



- Flow direction inconsistency

  A flow directional inconsistency occurs when you combine two P&ID objects that have opposite flow directions.

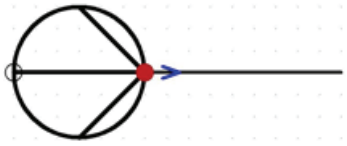
  If you attempt to connect a P&ID object against its flow direction, COMOS highlights this P&ID object in pale red.

- Multiple assignment inconsistency

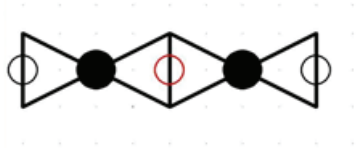
  A connection is inconsistent when you connect it in the Navigator to a pipe other than the one on the P&ID report.

  COMOS distinguishes the following in this case:

  – The connection is shown as a solid red circle when the other P&ID object is a pipe.



  – The connection is shown as an empty red circle when the other P&ID object is a device.



Pipes

5

5.1 Overview

Structure

COMOS stipulates a three-level pipe structure:

- Pipe (container)
- Pipe branch
- P&ID: Pipeline pipe segment, isometrics: Pipe

Creating objects

Objects are usually created in the engineering project. If you use the "Connection" tool for drawing on an interactive report, the corresponding objects are created automatically in the Navigator.

Disconnecting pipes

Placing a fitting on a pipe on the report separates the pipe, dependent upon the settings for the component. See also section Cutting a pipe (Page 87).

5.2 Drawing a pipe

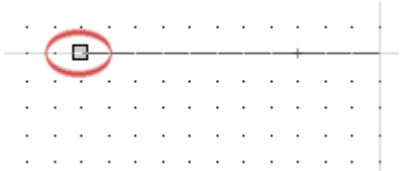
Requirement

A "P&ID" type interactive report is open.

Procedure

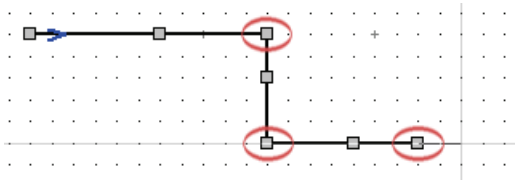
To draw a pipe, proceed as follows:

1. Select the "Connection" tool from the report toolbar.
2. To define the connector for the pipe, left-click on the report workspace.

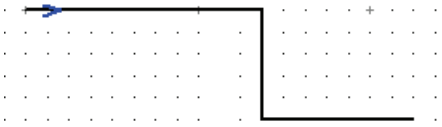


3. If you want the pipe to run diagonally as well as at right angles, select the "Oblique" option from the toolbar.
4. If you are working with pipe specs, select the required pipe spec and the nominal diameter from the toolbar.
5. To specify the pipe route and define intermediate points, move the cursor over the report and left-click.

   The flow direction corresponds to the direction in which you draw the pipe. The flow direction is shown at the start of the pipe as an arrowhead on the pipe.



6. To specify the end of the pipe, right-click on the required place on the report.



Result

A continuous line (polyline) is displayed on the report. The corresponding pipe objects are generated in the background and created in the Navigator.

You can also change the pipe layout subsequently via the grab points of the pipe. See also section Displaying pipes that cross one another (Page 32).

5.3 Pipe structures

Three-level pipe structure

When you draw a new pipe on the report, a three-level pipe structure is created in the engineering data by default:

Pipe

A pipe is created at the top level. This object collects and administers pipe branches and segments. Usually, one pipe object corresponds to an entire pipe run.

Pipe branch

Pipe branches are created underneath the pipe object. Pipe branches are the actual data holders. They encapsulate data changes within a pipe.

The pipe branch automatically takes over some data from the pipe (the "Technical data" tab, for example). Enter all other data in the pipe branch properties.

If you place a pipe branch on the report, COMOS automatically associates this pipe branch with a standard pipe segment, and inserts this pipe segment to the report.

If you use fittings to connect the pipe on the report, the pipe is further subdivided according to the pipe cut mode of the component. Either a new pipe branch is created with a standard pipe segment or the existing pipe branch is split up into pipe segments.

Pipe segment

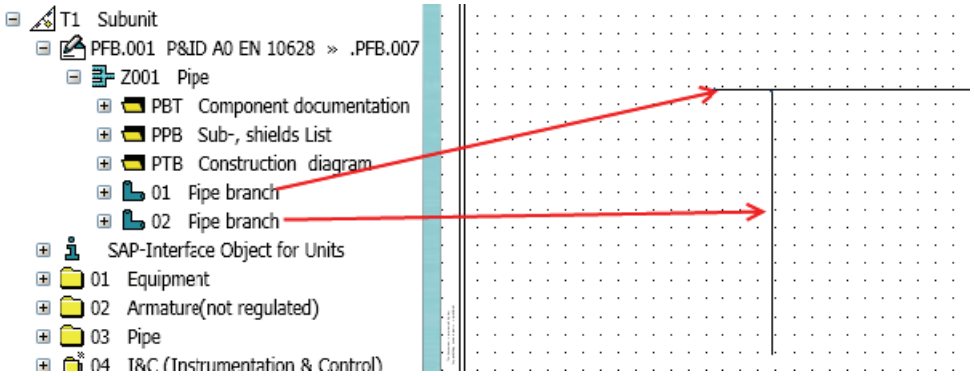
Pipe segments form the lowest level of the pipe structure. They do not have their own tabs.

Pipe pipe segments are placed on the report. When you select a pipe drawn in on a report and select the "Navigate > Object" command from the context menu, the pipe segment is marked in the Navigator.

Continuing to work with the connector tool

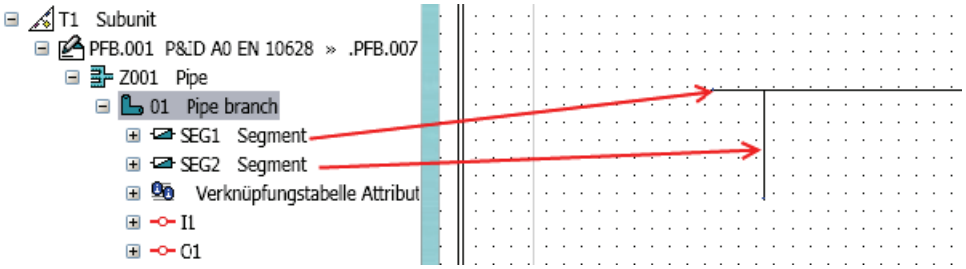
In the engineering project, if you drag a pipe object from the Navigator onto the report and then continue to work with the "Connection" tool, the new pipe branches and pipe segments are created directly underneath the pipe object you dragged onto the report.

Result:



If you drag-and-drop a pipe branch from the Navigator onto the report and then continue to work with the "Connection" tool, the pipe branch you dragged onto the report is pipe segmented and the connection just drawn in is created as a pipe segment.

Result:



The new pipe segments are all created underneath the same pipe branch.

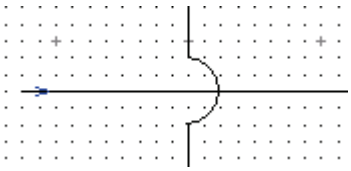
See also

Displaying pipes that cross one another (Page 32)

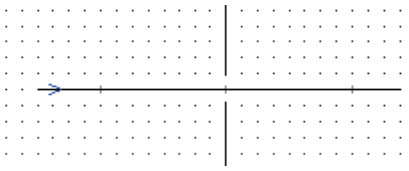
5.4 Displaying pipes that cross one another

Two pipes that cross one another can be displayed in various ways:

The pipes intersect.



The vertical pipe bridges the horizontal one with a semicircle.



The vertical pipe is broken visually.

These variants are displayed automatically. The display is determined by the settings in the options script of the report template.

See also

Predefined connector points (Page 33)

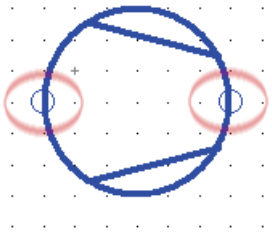
Standard pipe flags (Page 47)

Pipe overlaps on the report (Page 201)

5.5 Connecting pipes

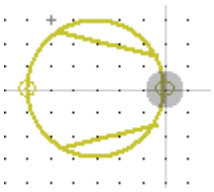
5.5.1 Predefined connector points

Predefined connectors of a component are shown even at an unconnected symbol; they are represented by a dot or a small circle.



"Connection" tool

If the "Connection" tool is active and you touch a predefined connector point, a gray dot appears above it and the associated object turns yellow.



Note

If you attempt to connect an object against its flow direction, COMOS shows this object in pale red.

See also

Displaying pipes that cross one another (Page 32)

Connecting objects from connector point to connector point (Page 34)

Drawing in bending points manually (Page 35)

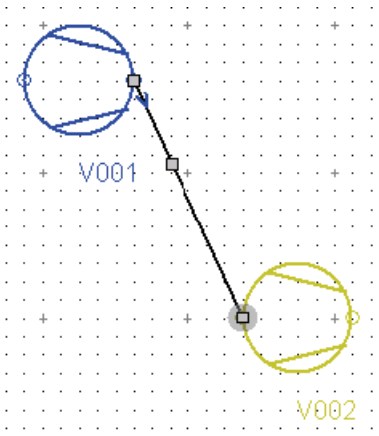
Connecting multiple objects at the same time (Page 35)

5.5.2 Connecting objects from connector point to connector point

Procedure

To connect objects to a pipe via their connector points, proceed as follows:

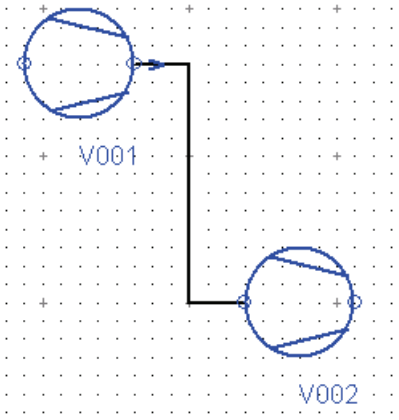
1. Select the "Connection" tool from the report toolbar.
2. Touch a predefined connector point with the mouse.
3. To define the start point of the pipe, click on the connector point.
4. Left-click on the connector point of the object you wish to connect to the object from Step 3.



5. To define the end point of the pipe, right-click on the connector point.

Result

A right-angled pipe is created.



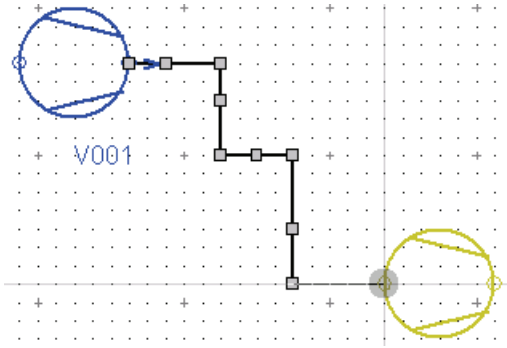
If you have activated the "Oblique" option, an oblique pipe is displayed.

5.5.3 Drawing in bending points manually

Procedure

To draw bending points into the pipe manually, proceed as follows:

1. Select the "Connection" tool from the report toolbar.
2. Touch a predefined connector point with the mouse.
3. To define the start point of the pipe, click on the connector point.
4. Click on the places in the report where you wish to draw in a right angle:



5. Left-click on the connector point of the object you wish to connect to the object from Step 3.
6. To define the end point of the pipe, right-click on the connector point.

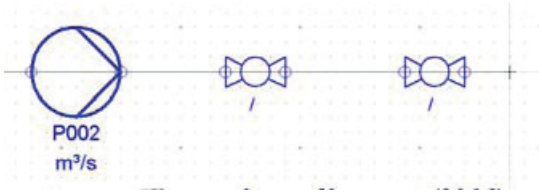
Result

The pipe is displayed with the bending points you defined.

5.5.4 Connecting multiple objects at the same time

Requirement

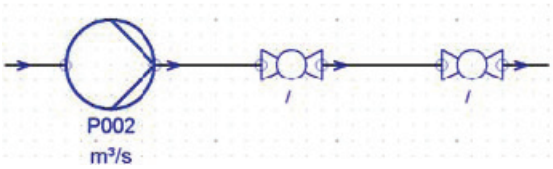
You have created multiple objects in a row, so that the connector points are all located on a single line.



Procedure

To connect multiple objects at the same time, proceed as follows:

1. Touch a predefined connector point with the mouse.
2. To define the start point of the pipe, select the start point of the pipe so that it is located on the same line as the connectors of the object and upstream of the connector of the first object.
3. Drag a connection over the connector points of all objects.



4. To define the end point of the pipe, right-click on the required point.

Result

The objects are interconnected along the pipe.

5.5.5 Connecting a pipe retrospectively

Requirement

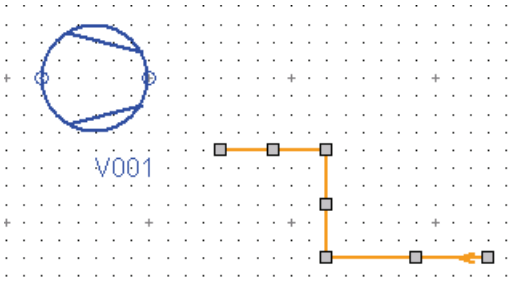
A pipe has been created on the report but not connected with any objects. The report contains an object to which you wish to connect the existing pipe.

Procedure

To connect the pipe to another object subsequently, proceed as follows:

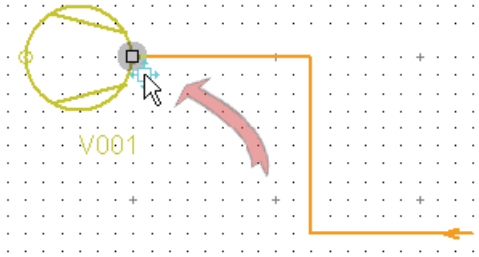
1. Click on the pipe twice in succession.

   The grab points appear.



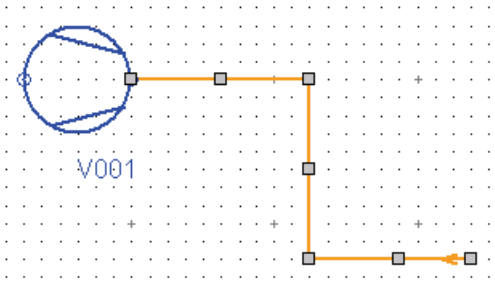
2. Drag the start and/or end point of the pipe onto the predefined connector points of the required object.

   When the pipe touches a predefined connector point, a gray dot appears and the associated object turns yellow.



Result

The pipe is connected with the object. The pipe route does not change.



See also

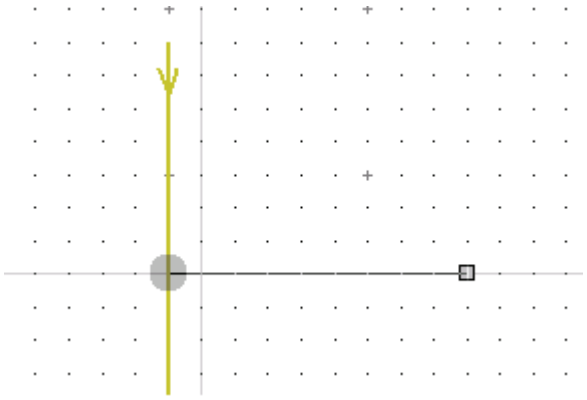
Displaying pipes that cross one another (Page 32)

5.5.6 Pipe branches

Dynamic connector points are only created when required.

Dynamic connector points

A dynamic connector point is generated when you draw in a branch on a pipe. If you touch another pipe with the "Connection" tool or with the end point of a pipe, the pipe is displayed in yellow. A gray circle is displayed at the contact point, indicating that you can connect the pipe here.



Note

Using the two report options `CopyPipeConnectionAutoOff` and `CopyMainBranch`, COMOS allows you to apply the substance data of the pipe to the pipe branch. You can find additional information on this topic in the "Reports - Basic Operation" manual, keywords "`CopyPipeConnectionAutoOff`" and "`CopyMainBranch`".)

See also

Drawing a pipe (Page 29)

5.5.7 Using dynamic connector points to create a connection

Introduction

Many P&ID objects allow you to select connector points freely.

Requirement

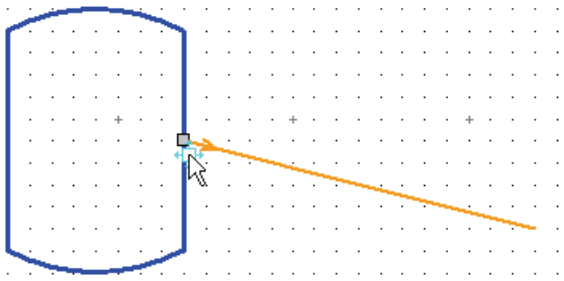
A "P&ID" type report is open. Objects have been placed on the report.

Procedure

To create a connection between an object and a pipe using dynamic connectors, proceed as follows:

1. Place the start or end point of the pipe on a P&ID symbol.

   You must precisely touch the line of the symbol. The P&ID symbol does not turn yellow and no gray circle appears.



2. To define the end point of the pipe, right-click on the required point.

   The "Create connection" window opens.

3. Select the required options.
4. Click "OK" to confirm your selection and close the "Create connection" window.

Result

The connection is created.

Dismantling a connection

If you break the connection, connector points that had been created automatically are retained initially. If you disconnect the connection and wish to delete the connector points, select the "Options > Restore the original symbol" command from the context menu.

See also

Options in the "Create connection" window (Page 261)

5.6 Editing pipes

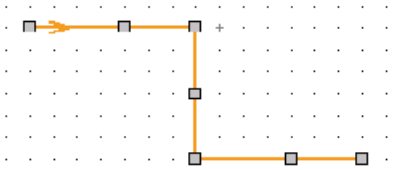
5.6.1 Changing the pipe route

Procedure

To change the pipe route, proceed as follows:

1. Select the "Identify" tool from the report toolbar.
2. Click on the pipe twice in succession.

   The pipe grab points appear. The pipe turns yellow.



3. Select one of the following options:
   - To insert more bending points, click on one of the center points, hold down the mouse button, and drag the pipe in the required direction.
   - To change the length of the individual branches, click on one of the end points, hold down the mouse button, and drag the pipe in the required direction.
   - To connect the pipe to another object, click on the corresponding end point, hold down the mouse button, and drag the pipe to the object's connector point.
   - To disconnect the pipe from another object, click on the corresponding end point, hold down the mouse button, and drag the pipe away from the object.

Result

The pipe route changes accordingly.

See also

Connecting a pipe retrospectively (Page 36)

5.6.2 Changing the flow direction

Introduction

If you connect a measuring or actuating function to other P&ID objects in a P&ID report , COMOS automatically determines the direction of flow. The flow direction is determined regardless of whether you are connecting two P&ID objects by docking or by using the "Connection" tool.

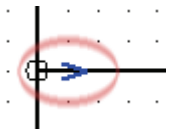
COMOS determines flow direction as follows:

- With measuring functions, the flow direction always points to the measuring function.
- With actuating functions, the flow direction always points always from the measuring function.

You can change the flow direction.

Requirement

A pipe has been created on the "P&ID" type report. The flow direction of the pipe is displayed.



Procedure

To change the flow direction of a pipe, proceed as follows:

1. Select the required pipe on the report.
2. Right-click in the report workspace.
3. Select one of the following options:
   - To change the flow direction of all pipe segments of a pipe branch, select the "Options > Change flow direction" command from the context menu.
   - To change the flow direction of the selected pipe segment, select the "Options > Change flow direction (only own)" command from the context menu.

Result

The flow direction of the pipe or of the pipe segment changes.

If you use standard pipe flags, these change their direction along with their pipe. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "StdPipeNoReflect".

If fittings that are dependent upon flow direction have been built into the pipe, they will not change flow direction automatically. Adapt the flow direction of these fittings manually.

See also

Flow direction of action and measuring lines (Page 68)

5.6.3 Showing and hiding additional pipe symbols

Procedure

To hide or show additional pipe symbols, proceed as follows:

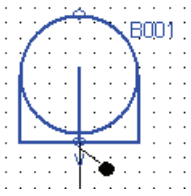
1. Select a pipe on the report.
2. Right-click in the report workspace.
3. Select one of the following options:
   - To show the start symbol, select the "Options > Show only start symbol" command from the context menu.
   - To show the end symbol, select the "Options > Show only end symbol" command from the context menu.
   - To show the start symbol and the end symbol, select the "Options > Show start and end symbol" command from the context menu.
   - To hide the start symbol and the end symbol, select the "Options > Hide start and end symbol" command from the context menu.

Result

The corresponding pipe symbol is shown or hidden.

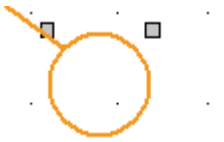
The start and end symbols differ according to drawing type. As a rule, no start and end symbols are displayed in DIN 10628, so these commands have no effect there.

Circles are displayed in DIN 2481:



The fact that the circle at the bottom right in the example is solid means that the device that has been connected on the reverse side is in the same unit or part unit. If that is not the case, then an empty circle is drawn.

Nonetheless, the circles can also be set manually. If you single-click twice to switch the pipe to edit mode, then two grab points appear at the circle. The outer grab point can take on one of three possible states:



Blank

Automatic

Filled in

See also

Symbols: End symbols (Page 253)

5.7 Pipe flags

5.7.1 Text symbols

Labeling symbols

Labeling symbols are not scalable and cannot be resized; they are always displayed exactly as they were drawn.

Text symbols largely react automatically to changes in their connected pipes. The extent to which they are mirrored, automatically rotated or selected together with the pipe is set in the object properties of their base object.

Some labeling symbols only serve as placeholders for texts that are to be output; in other words, they do not have any associated graphics with lines or text frames. If the attribute the symbol evaluates is empty, you will find it difficult to locate the symbol once it has been placed.

The placement points of these symbols are, therefore, drawn in as a blue, non-printable cross.

5.7.2 Placing a pipe flag manually

Introduction

A number of different pipe flags have been predefined in base object node "@01 > PID > 99 > 01 P&ID labeling symbols" in the COMOS DB.

Requirement

A pipe has been placed on the report.

Procedure

To place a pipe flag manually, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Mark the required base object of the pipe flag.
3. Use drag&drop to move the required base object from the Navigator onto the pipe on the report.

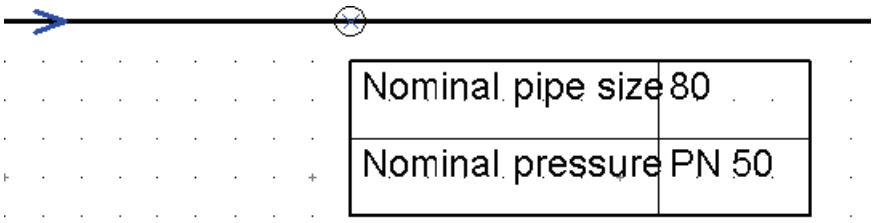
Result

The pipe turns yellow briefly. The pipe flag is connected to the pipe automatically. The pipe flag reads out the pipe data.

When you select or move the pipe, the pipe flag is automatically selected or moved with the pipe.

Example

A pipe flag that reads the nominal diameter and the nominal pressure of its pipe:



Placing the base object of a flag

If you use drag&drop to move the base object of a pipe flag from the Navigator ("Base objects" tab) to a pipe, the pipe flag is connected to the pipe automatically. The flag reads out the pipe data and reacts automatically to graphical changes affecting the connected pipe.

5.7.3 Connecting pipe flags subsequently

Requirement

A labeling symbol (a pipe flag, for example) has been placed on the report but not connected to the pipe.

Procedure

To connect the pipe flag to a pipe subsequently, select the pipe flag and use drag&drop to move it to the pipe.

Result

The pipe flag is connected to the pipe.

Alternative

If you do not wish to place the pipe flag in the direct vicinity of the pipe, COMOS can still read and display the pipe information automatically. To do this, proceed as follows:

1. Place the flag in the required place on the report.
2. Select the corresponding pipe.
3. Select the "Copy" command from the context menu.
4. Select the flag.
5. Select the "Reference > Connect with" command from the context menu.

The flag is not selected or moved automatically together with the pipe, but it always outputs the current pipe data.

5.7.4 Changing the position and alignment of the flag

Introduction

You can use the grab points of the flag to change its position and alignment.

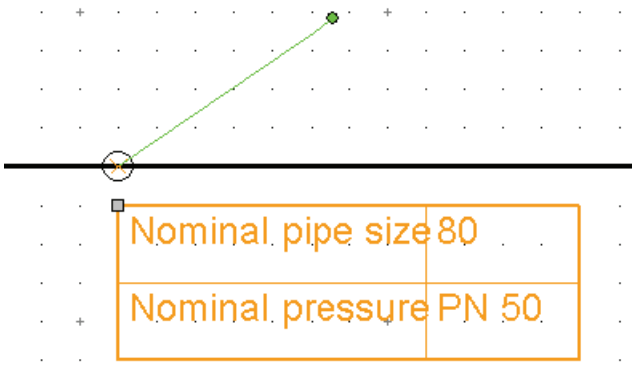
Procedure

To change the position and alignment of the flag, proceed as follows:

1. Click on the flag twice.

   The symbol turns orange and the grab points are shown.

2. To change the alignment of the flag, move the green grab point. To change the position of the flag, move the gray grab point.



Result

The position and the alignment of the pipe flag are changed accordingly.

5.7.5 Retaining pipe flag on deletion

Overview

Two pipe segments or pipe branches are interconnected by means of an insert, for example. You have placed pipe flags at one or both pipe segments or pipe branches.

You delete the insert, taking the one pipe segment or pipe branch with it.

The pipe flag is retained in any case. Even if the pipe segment or pipe branch to which the flag originally belonged has been deleted.

5.7.6 Showing pipe flags

Procedure

To show the pipe flags , follow these steps:

1. Select a pipe on the report.
2. Select "Options > Show all pipe flags" from the context menu.

Result

The pipe flags of all pipes in the report are displayed.

5.7.7 Hiding pipe flags

Procedure

To hide the pipe flags , follow these steps:

1. Select a pipe on the report.
2. Select "Options > Hide all pipe flags" from the context menu.

Result

The pipe flags of all pipes in the report are no longer displayed.

5.7.8 Standard pipe flags

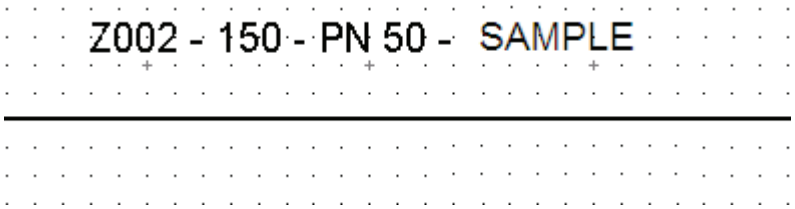
Using the standard pipe flag

Instead of placing a labeling symbol, you can have the standard pipe flag output. The standard pipe flag is a pure text flag.

It is a textual component of the pipe symbol. COMOS neither creates an object for it in the engineering data nor bases it on a base object.

The standard pipe flag is either created automatically when the pipe is drawn in on the report or you show it later by selecting the corresponding command from the pipe context menu.

In the COMOS DB, the standard pipe flag reads the name of the pipe and the nominal pressure, the nominal diameter, and the pipe spec of the pipe branch:



See also

Overview of P&ID objects (Page 19)

Pipe branches (Page 38)

Showing a standard pipe flag (Page 48)

5.7.9 Showing a standard pipe flag

Procedure

To show a standard pipe flag, proceed as follows:

1. Select a pipe on the report.

   To select multiple pipes, hold down <Ctrl> and select the desired pipe.
2. Select "Options > Show standard pipe flag" from the context menu.

Result

The standard pipe flags for this pipe are displayed.

See also

Standard pipe flags (Page 47)

5.7.10 Hiding standard pipe flags

Requirement

The standard pipe flag is showing.

Procedure

To hide the standard pipe flag, proceed as follows:

1. Select a pipe on the report.

   To select multiple pipes, hold down <Ctrl> and select the desired pipe.
2. Select "Options > Hide standard pipe flag" from the context menu.

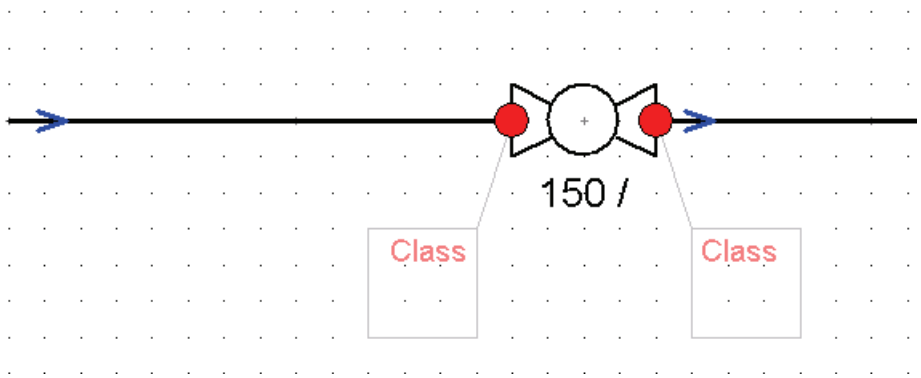
Result

The standard pipe flag is hidden.

5.7.11 Flag for PipeSpec relevant information

Purpose

You can work with pipe specs on "P&ID" type interactive reports. A flag shows if objects whose pipe-spec-relevant attributes (e.g. pipe spec, nominal diameter, and nominal pressure) do not have the same values are interconnected.



The flag contains text indicating the attributes where the connected objects differ.

If you update the values, the entry in the flag is also updated.

5.8 P&ID graphic symbols

Graphical symbols are objects that are purely graphical additions to a drawing; they cannot be ordered and do not appear in lists of objects.

Graphical symbols also transport information via their graphic.

Creating a graphical symbol

When you place a P&ID graphical symbol on a report, an engineering object is created in the Navigator. Some engineering objects have connectors and attributes about which the graphical symbols provide more detailed information. The P&ID graphical symbols are always managed in the Navigator underneath the report and do not get a position number.

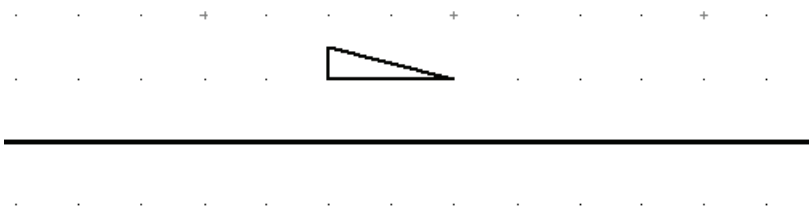
Base object

Most of the graphical symbols that have been predefined in the base data under "@01 > PID > 99 > 02 P&ID graphical symbols" are symbols that relate to pipes.

Example

Base object. "@01 > PID > 99 > 02 > 06 Slope"

This graphical symbol indicates that there is a slope in the pipe and shows in which direction it runs:



5.9 Inserting a pipe break

Requirement

A pipe has been created on a "P&ID" type report.

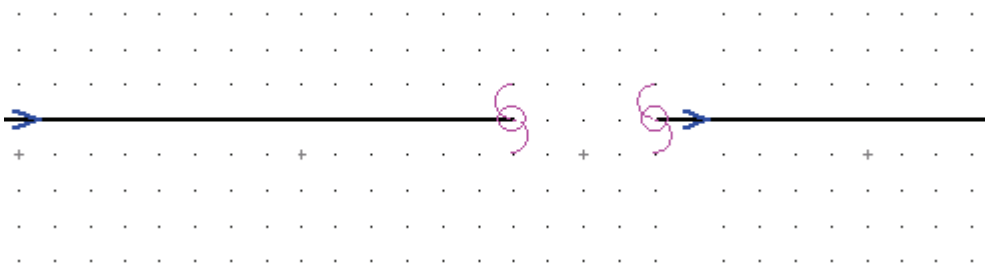
Procedure

To insert a pipe break, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Select the base object "@01 > PID > 99 > 02 > 01 Pipe break".
3. Use drag-and-drop to move the base object to the required pipe.

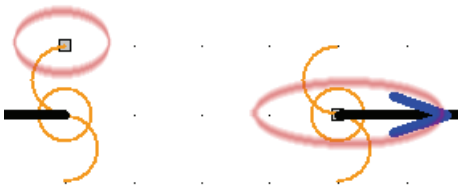
   The pipe turns yellow.
4. Place the pipe break in the required place.

   You can see the pipe break on the report.



5. To configure a pipe break, click the symbol.

   The grab points are displayed.



6. Select one of the following options:
   - To make the pipe break symbol larger or smaller, click on the grab point on the pipe arm in the image on the left and drag the grab point in the required direction.
   - To increase the distance between the two pipe segments, click on the grab point on the pipe arm in the image on the right and drag the grab point in the required direction.

Result

The pipe branch is retained in the database and is segmented. The generated pipe segments are connected to the connectors of the pipe break.

5.10 Initiating the data flow on the P&ID report

Procedure

To update connected components and pipe branches, proceed as follows:

1. On the report, select a component which is connected to other components, for example.
2. Open the properties of the selected device.
3. Make the changed to the properties.
4. Click "OK" to save the changes.
   - The "Update objects?" window opens. The "Detail" button expands the "Update objects?" window so that you can see which object attributes have been updated, and the corresponding values.
   - All components affected by the update are displayed in yellow.
5. Activate one of the following options:
   - To always update the connected components with the changes in the current session, select the "Always" option.
   - To never update the connected components with the changes in the current session, select the "Never" option.
   - To update only changes to specific attributes, select the "User defined" option.

     In the "Always accept the difference" column, select the options of the attributes that you never want to update.

     If the administrator makes the "Accept the difference" column available, select the options of the attributes that you do not want to update in the current update process.

     See also section Global flag (Page 216).
6. Click "OK".

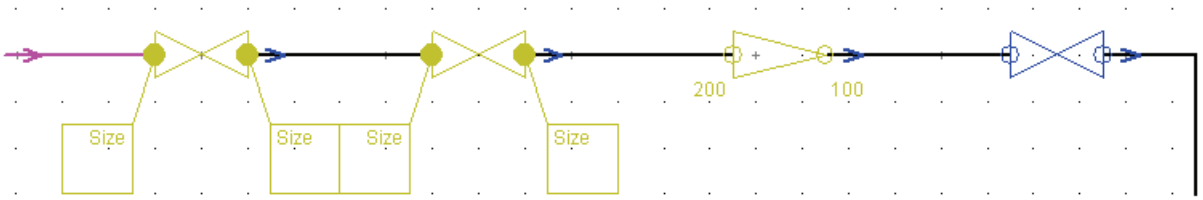
Result

The changes to the properties are saved. The connected objects are updated if applicable.

When you update a component, if appropriate, you can assign a new matching 3D object to it from the component catalog. See also section Pipe spec mapping on a P&ID report (Page 53).

Example

You update the value of the "Nominal diameter" attribute. The passing of the nominal width attribute is stopped at the reducer since taking over the same value for multiple nominal widths does not make a lot of sense. For more information see section Data flow in detail (Page 209).



See also

Data flow for pipes: UpdateConnected (Page 209)

5.11 Start pipe spec mapping

5.11.1 Selection sets of a pipe spec mapping

You can perform pipe spec mapping for two available selection sets:

- For P&ID objects on a P&ID report
- For P&ID objects below a node in the Navigator, you have to place in the "Pipe spec mapping" area per drag-and-drop.

5.11.2 Pipe spec mapping on a P&ID report

Requirement

You have assigned to all pipe branches a pipe spec and a nominal diameter, for which you want to perform pipe spec mapping.

Procedure

1. Open a report.
2. Drag-and-Drop a component onto the pipe branch.

   The "Update objects" window opens.
3. Enable the option "Always" or "User defined".

4. Click "OK".

   The "Pipe spec mapping" window opens. See also section "Pipe spec mapping" window (Page 261).

5. Select the P&ID objects you want to display from the "Display" list.
6. In the "Pipe spec mapping" window, click the "Execute now" button.

   Matching components are determined for all P&ID objects in the device catalog.

7. Select one or several P&ID objects using the <Ctrl> key in the "Pipe spec mapping" area.
   - If you select one P&ID object: All available components for the selected request object appear in the "Existing components" area.
   - If you select multiple P&ID objects: Only the components that meet the requirements of all selected objects are shown as available components in the "Existing components" area.
8. Select one of the available components.
9. Click the "Assign" button.
10. Repeat steps 4 to 6 for all the P&ID objects as required.
11. Click "OK".

Result

Pipe spec mapping is performed for the selected P&ID objects.

5.11.3 Pipe spec mapping via the Navigator

Requirement

You have assigned to all pipe branches a pipe spec and a nominal diameter, for which you want to perform pipe spec mapping.

Procedure

1. Click the "Plugins > Basic> Pipe spec mapping" command in the COMOS menu.

   The "Pipe spec mapping" tab opens. See also section "Pipe spec mapping" window (Page 261).

2. Drag-and-drop a node from the Navigator into the "Pipe spec mapping" area.
3. Select the P&ID objects you want to display from the "Display" list.
4. In the "Pipe spec mapping" window, click the "Execute now" button.

   Matching components are determined for all P&ID objects in the device catalog.

5. Select one or several P&ID objects using the <Ctrl> key in the "Pipe spec mapping" area.
   - If you select one P&ID object: All available components for the selected P&ID object appear in the "Existing components" area.
   - If you select multiple P&ID objects: Only the components that meet the requirements of all selected P&ID objects are shown as available components in the "Existing components" area.
6. Select one of the available components.
7. Click the "Assign" button.
8. Repeat steps 5 to 7 for all the P&ID objects as required.
9. Click "OK".

Result

Pipe spec mapping is performed for the selected P&ID objects.

5.12 Drawing pipes across pages

Requirement

A pipe has already been created on the report.

Procedure

To draw a pipe across pages, proceed as follows:

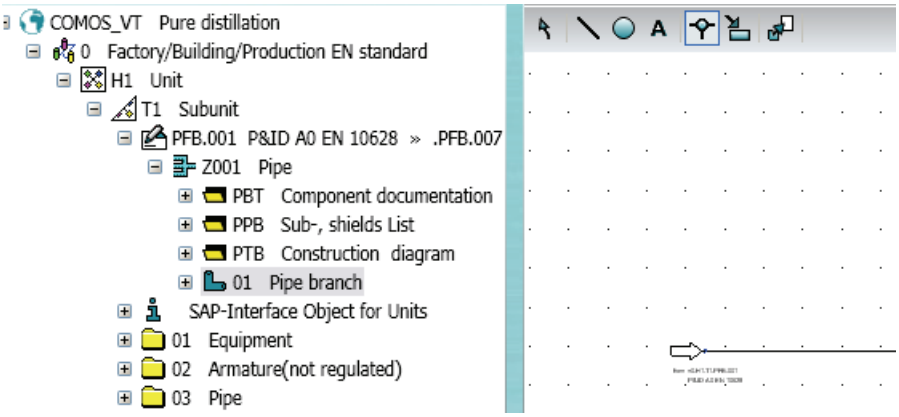
1. Select the required pipe.
2. Navigate to the object of the pipe branch in the Navigator.

   The pipe branch is now selected in the Navigator.
3. Open a second "P&ID" type report.
4. Use drag-and-drop to move the selected pipe branch to the second report.

   COMOS activates the "Connection" tool automatically.
5. Draw the required pipe on the second report.

Result

The pipe branch you dragged onto the second report is separated and now consists of two pipe segments which are placed on different reports. A reference symbol is inserted at the start and end points of the pipe branch.



If P&ID reports are connected via cross-references, you can navigate to the individual documents by selecting the "Navigate > Documents > <Name of the referenced document>" command from the context menu.

Positions 6

6.1 General

Overview

Positions are created during P&ID planning. They designate process-related tasks or control tasks. The position object serves as a folder, below which functions and reports are located.

Class

A P&ID position has:

- Class: "Point"
- Subclass: "(None)"

In addition, all vessels are created with class "Position", subclass "Equipment" in the COMOS DB. However, in this section, the term position always refers to positions in the context of process control, i.e. objects with class "Position", subclass "(None)".

Structure of base objects

In the COMOS DB:

- "@02 > 020 Positions"
  - "> 00 Position acc. to DIN/IEC"
  - "> 01 Position acc. to ANSI"

In the engineering view, the secondary objects are made available via the "New" context menu of the position. However, many of the objects are not required until the I&C planning stage. You find additional information on this topic in the "E&IC" manual, keyword "Basic Engineering" and "Base objects of positions".

See also

Functions (Page 61)

6.2 Creating positions

6.2.1 Automatically

Requirement

Sometimes positions are created automatically by COMOS:

1. Under certain conditions when a function is created by placing it on the report
2. Under certain conditions when inputting a function code in a measurement function
3. Under certain conditions when copying functions
4. Under certain conditions when assigning an object in "Define owner" mode

The base object to which the position is assigned is defined using the `PositionCDeviceFullName` script variable, along with the function's function code.

For 1: By placing on the report

- A function (measurement or actuating function) is placed directly on the P&ID and `SortNewObjectsInCategory` is activated in the report template:

  (Directly = Your base object is moved from the Navigator, "Base objects" tab using drag&drop; the function is not created first in the engineering view in the Navigator.)

  Result:

  A position is created below the I&C category object, and the function below. If the function already had a function code when it was placed, a position of the same type is created. If a function without a function code is placed, an undefined position is created.

  Changing the function code subsequently does not automatically change the type of position.

- A function is created by placing it on the report and `SortNewObjectsInCategory` is deactivated:

  Result:

  The function is created below the report:

  – Measurement function with function code: A position of the appropriate type is created below the report, and the function below it.

  – If a measurement function creates no function code or an actuating function: The function is created directly below the report, but no position is created.

For 2: By entering a function code

If the measurement function has not been assigned to a position in the engineering data when the function code is input, as well as a new base object being assigned to the function (change base object), a position with a matching type is created automatically and the function is moved to this position.

If a new function code is input at an actuating function, this has no effect.

For 3: When copying functions

- If all functions below a position are copied
- If only a subset of the functions are copied: depending on where the original function is located and which paste mode is selected. See also section Copying and pasting on P&IDs (Page 102).

For 4: Object assignment with the "Set owner" mode

If you select a function on the P&ID drawing and use the "Assign object" button to assign it a new owner (mode: "Define owner"), the following items are moved under the new owner:

- the position below which the function is located
- all objects located below the position

If the function was located directly below a P&ID diagram and did not yet have a position, a position is first created below the assigned owner and only then is the function moved below the position.

Determining the base object of the position

The base object on which the position is based is determined in the following way:

The script variable `PositionCDeviceFullName` of the function is evaluated. It is set in `UserScriptBlock1` of the function. This variable determines the base object node under which the search for the base object of the positions is to be carried out.

Afterwards the function code of the function is evaluated. Underneath the base object node input in `PositionCDeviceFullName` a search is carried out for the base object whose name is the same as the first letter of the function code.

If no function code is input, the first base object under the node input in `PositionCDeviceFullName` is used. In the COMOS DB, this is an "Undefined" type of position.

6.2.2 Manually

You can create positions manually. To do this, right-click on the "04 I&C" folder underneath the subunit and select the "New > @P Positions" command from the context menu. See also section Reports and administration objects (Page 174).

The objects that have been prepared underneath the individual positions differ slightly.

Functions 7

7.1 General

Overview

The function describes the measuring or actuating task and the processing function(s) of a position. It corresponds to a "measurement function" in the P&ID scheme.

- "Class": "Function"
- "Subclass":
  - "Instrumentation": describes a P&ID function
  - "Actuator": An object with this actuator subclass separates the pipe during drag&drop.
  - "Actuator (insertion)": An object with this subclass segments the pipe.
- "Creation mode": "Free"
- "Creation option": "Normal"

The "Request" option on the "System settings" tab is not activated for functions. This means that the functions are not implemented. They may contain elements created with the "Request" option.

Methods

A function is primarily defined with the following four methods in COMOS:

- The display of the symbol
- Measurement function: The function code and the type of the measurement function which leads to a change of base object
- The number of process connectors
- The attributes of the function, where the attributes appear dependent upon the change of base object In particular, the component of the subclass is swapped. See also section Process Coupling and Process Connector of a Function (Page 75).

Moving

When functions are cut or pasted, the owner position also moves if all functions of this position have been input.

7.2 Creating measurement functions

Using the symbol bar to create a measurement function

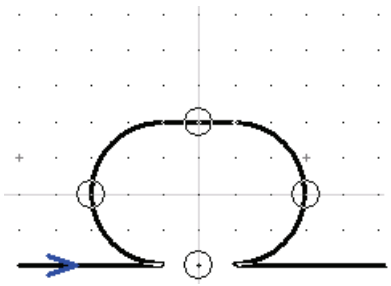
You can use the report symbol bar to create measurement functions. You can connect the measurement function with the process directly when placing it on the P&ID, or in a subsequent work step.

Procedure

To use the symbol bar to create a measurement function, proceed as follows:

1. On the symbol bar, click the symbol for the required measurement function.
2. Move the cursor to the position on the report where you wish to place the measurement function.

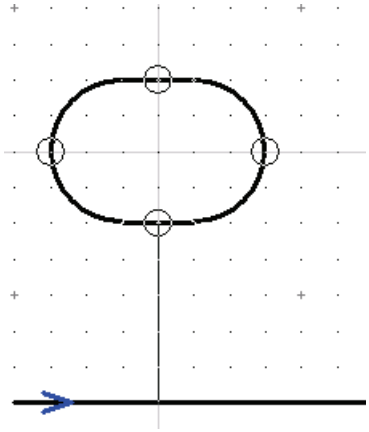
   The symbol for the function moves over the report together with the cursor.
3. Position the cursor so that one of the connector points that was prepared in the symbol touches the pipe.



4. Pause briefly and then move the cursor away from the pipe.

   The function docks on the pipe and a measuring line is drawn in between the pipe and the function.

   If a function was docked in the wrong position during placement, move the cursor out of the drawing area and then back in again. The function moves over the report again, together with the cursor.



5. Left-click once and the function is created in the engineering data:
   - If your administrator has configured the project so that all P&ID objects are sorted by category into folders parallel to your P&ID report, the functions are created under an undefined position in the "I&C (Instrumentation & Control)" folder.
   - If your administrator has configured the project so that all objects are initially grouped under the "P&ID" type report, the function is created directly underneath the report.

Result

The function is connected with the process. As a result it gets several tabs that store the data that is relevant for the process connector.

If you are working with process couplings, the process coupling is also created automatically. A nozzle is used as the default process coupling.

The measuring line gets a direction when the function is placed. This direction is determined by the system. You can also connect the function with any other P&ID object by using the predefined connectors.

If the object does not have any predefined connector points, place the function and connect it subsequently with the "Connection" tool.

The measurement functions that are offered on the symbol bar of the sample project do not have a function code yet. The next section describes how you can input a function code.

You can create the function first in the Navigator and then place it on the report.

Alternative 1

You can also create the measurement function in the Navigator first, on the "Units" tab:

1. Select a function using the "New > <required function>" command from the context menu for a position.
2. Use drag&drop to move the function to the report.

   The function has the same type as its position right from the very beginning.

You should proceed in this way if two functions are to be located underneath one position and the first function has already been created and assigned the required function code.

Alternative 2

You can select a base object with the required function code directly for the function on the "Base objects" tab in the Navigator and use drag&drop to move it to the report.

In this case, a position of the correct type is also automatically created in the engineering data.

See also

Connecting functions (Page 67)

Structure of the base objects of functions (Page 223)

Display on reports (Page 72)

7.3 Inputting a function code

Introduction

Once you have placed and connected a measurement function via the symbol bar, you can input the function code of the measurement function. The function code determines the processing of the function and controls a number of background functions that facilitate your work.

Input the function code in the function label.

Procedure

To input the function code, proceed as follows:

1. Select a function.

   Three text fields are shown in the toolbar.
2. Input the required function code in the text field in the center.

   The first letter of the function code determines the type of function, the following characters the type of processing.
3. Click the  button.

Result

The function code is written to the function label and is output on the drawing. A position of a matching type is automatically created below the report and the measurement function is moved below it.

A change of base object takes place in the background. The COMOS DB contains predefined base objects for the various types of measurement function, such as temperature measurements or pressure measurements. The measurement function automatically gets a pointer to the matching base object when the function code is input.

Function code and process couplings

Although the base object changes when the function code is input, the process coupling is not updated automatically. Functions for flow measurements, for example, have an inline device as the default process coupling. Although the function now has the base object for a flow measurement, the nozzle is not automatically replaced by an inline device.

See also

Changing the process coupling (Page 80)

Function code and base object change (Page 65)

7.4 Function code and base object change

Effect of the function code

- Measuring functions

  When you input a function code for a measurement function, a change of base object takes place automatically: A function is searched in the base data whose name is the same as the first letter of the function code.

  During the base object change it is checked if the function is already located underneath another position. If not, a position of a suitable type is created automatically. See also section Automatically (Page 58).

  Example: A function that has the base object "@03 > BAS > IEC > @F > 01 Measurement function" and does not have a process connector receives the function code "PI". The function gets a new base object automatically: "@03 > BAS > IEC > @F > 01 > P Pressure".

- Actuating functions

  The function code is purely for information purposes and does not initiate any automated mechanisms.

Notation for alarms

You can tell from the "E/I&C Options" tab for the position or subunit whether or not alarms with +/- or H/L have been input. See also section Data flow of positions (Page 222).

7.5 Creating an actuating function

Requirement

A measurement function with function code has been created. The corresponding position is created automatically.

Procedure

If the actuating function is to be located underneath the same position as the corresponding measurement function, proceed as follows:

1. Use the context menu to create an actuating function in the Navigator underneath the position.

   A 2-, 3- and 4-way valve (actuating function) and a responses valve have been predefined in the "New > 02 Actuating functions" context menu of the position. The tabs for the process connector are created at the same time, even if the function has not yet been connected to the process.
2. Create the required valve in the Navigator, using the "New" command from the context menu for the actuating function.
3. Place the valve on the report.
4. Connect the valve to the process.
5. Use the "Graphical settings" command from the context menu to predefine the valve further on the report.

   Example: Create a drive; define the safety settings and the actuating behavior.
6. Next, place the actuating function on the report.
7. Use the "Connection" tool to connect the actuating function to the drive for the valve and the measurement function. See also section Connecting functions (Page 67).

Alternative

If the actuating function is to be located separately from its measurement function in the Navigator, proceed as follows:

1. First of all create the position in the Navigator.
2. The remaining steps are the same as Steps 1-7.

Actuating functions and function code

With actuating functions the function code is purely for information and does not bring about a change of base object.

7.6 Connecting functions

You can connect a function to the process either directly during placement or subsequently using the "Connection" tool.

Connecting a function during placement

See also section Creating a measurement function via the base object icon bar (Page 62).

Connecting functions with the "Connection" tool

You can even place a function on the report first and then connect subsequently it with the "Connection" tool.

Select this method if you want to connect an actuating function or if you want to connect a function to a vessel.

To connect a placed function with another object, proceed as follows:

1. Place the function on a blank point in the report.

   Four connector points have been predefined at the symbol of the function.

2. Select the "Connection" tool and connect one of the connector points to the process.

   Depending on which object you want to connect the function to, connector points are offered by default in the symbol of the object. See also section Predefined connector points (Page 33). Alternatively, you are free to select where to connect the function to the object. See also section Using dynamic connector points to create a connection (Page 38).

Result

An objectless connection line that has a flow direction is created. If you use the "Connection" function to connect the function, make sure that you are drawing in the right direction:

- When connecting a measurement function with the process: From the process to the function. A measuring line is created.
- When connecting an actuating function with its actuator: From the function to the actuator. An action line is created.
- When connecting a measurement function with an actuating function: From the measurement function to the actuating function: An action line is created.

However, you can also change the flow direction subsequently. See also section Flow direction of action and measuring lines (Page 68).

When you connect a function to the process, the function gets a new process connector. In addition, your project can be configured so that a process coupling is automatically built in.

Data is exchanged between function, process coupling, and process; this operation is largely automatic. See also section Automated data flow (Page 95).

See also

Pipe branches (Page 38)

7.7 Flow direction of action and measuring lines

Overview

Action lines and measuring lines always have a flow direction.

If action and measuring lines are created automatically when placing the function on the report, the flow direction is set by COMOS. If you use the "Connection" tool to create a connection, the flow direction is the same as the drawing direction.

See also

Changing the flow direction (Page 41)

7.8 Generating cross-references for action lines

Introduction

Functions can have a relation to one another. For example, a temperature measurement and a density measurement can belong together in a process diagram. Use action line cross-references to see on the report which temperature measurement belongs to the density measurement.

Procedure

To create a cross-reference between two action lines, proceed as follows:

1. Select the "Connection" tool.
2. Draw a line at both functions and leave it open at the end.
3. Select one of the open connections.
4. Select the "Connection > Memorize..." command from the context menu.
5. Select the second open connection.
6. Select the "Connection > Connect with..." command from the context menu.

Result

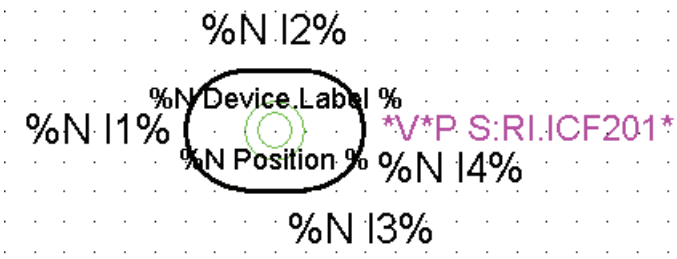
The two open action lines are connected and each one is assigned a cross-reference text.

7.9 Symbols: Function symbols

Either symbols from the symbol library or symbols created by the user with the assistance of the symbol editor must be assigned on the "Symbols tab" as appropriate for the drawing type. When working in the P&ID module, symbols from P&ID drawing types are usually available, depending on which standard you are working with.

%N texts for function symbols

Use %N texts to create variable texts for symbols. In the base objects of the COMOS DB, the function symbols have the following %N texts:



- %N I1% through %N I4%:

  These texts determine the possible connector points at the symbol for connectors that have been allocated automatically. Automated I and W connectors are not implemented by means of text variables and hence are not visible here.

- %N Device.Label % and %N Position %

  These texts display variables that are replaced by the designation of the actual engineering object on the system side when the symbol is placed on a document. Therefore, exact spelling is binding for all text functions and blanks, for example, must not be omitted.

Additional text variables

The above illustration displays only one of the possible variables (shown in color). In the COMOS DB, these text variables have only been predefined in drawing type "P&ID2".

Example

- *V*P S:RI.ICF201*

  This text variable evaluates the attribute "ICF201 Optional: Alarm display"; this is stored with base standard table "IC > Y > 0 > 02 > ICF201 Alarm", which only contains the values "TRUE" and "FALSE". If "TRUE" then the following script is evaluated:

```
Function Geometry (PARAM)
  ResetToDefaults
  Header.Class = "eS2"
  Set p1 = Coord(0,8.75)
  Font.Height   = 7
  DrawText p1, "%N ComosDevSpec('RI', 'ICF202a2',
    'DisplayValue')%", 0, 1
```

```
  Set p2 = Coord(0,6.25)
  Font.Height   = 7
  DrawText p2, "%N ComosDevSpec('RI', 'ICF202a',
    'DisplayValue')%", 0, 1
  Set p3 = Coord(0,3.75)
  Font.Height   = 7
  DrawText p3, "%N ComosDevSpec('RI', 'ICF202',
    'DisplayValue')%", 0, 1
  Set p4 = Coord(0,-3.75)
  Font.Height   = 7
  DrawText p4, "%N ComosDevSpec('RI', 'ICF202b',
    'DisplayValue')%", 0, 1
  Set p5 = Coord(0,-6.25)
  Font.Height   = 7
  DrawText p5, "%N ComosDevSpec('RI', 'ICF202b1',
    'DisplayValue')%", 0, 1
  Set p6 = Coord(0,-8.75)
  Font.Height   = 7
  DrawText p6, "%N ComosDevSpec('RI', 'ICF202b2',
    'DisplayValue')%", 0, 1
End Function
```

- *V*P S:RI.ICF200*

  This variable evaluates the attribute "ICF200 Optional: Graphic for relevant adjustment"; this is stored with base standard table "IC > Y > 0 > 02 > ICF200 Graphic", which only contains the values "TRUE" and "FALSE". If "TRUE" then the following script is executed:

```
Function Geometry (PARAM)
  ResetToDefaults
  Header.Class = "eS2"
  Set p1 = Coord(0,0)
  Font.Height   = 6
  DrawText p1, "*V*P S:RI.ICA110*", 0, 1

  Set p2 = Coord(3.75,0)
  Font.Height   = 6
  DrawText p2, "*V*P S:IC010.ICA103*", 0, 1

  Set p3 = Coord(7.5,0)
  Font.Height   = 6
  DrawText p3, "*V*P S:IC010.ICA101*", 0, 1
End Function
```

Subsymbols

If the attributes are set to "TRUE" (meaning that they are selected), the subsymbols for alarm displays and relevance displays can be displayed in the diagram through scripts that are stored in the standard tables:

- Alarm displays

  "RI Display mode" tab of the function: "ICF201 Alarm display:" is selected and values are input in the attributes for the alarms (ICF202, ICF202a, ICF202a2, ICF202b, ICF202b1, ICF202b2).

- Relevance displays

  "RI Display mode": "ICF200 Graphic for relevant adjustment" tab is selected. In addition: The attribute "ICA110 Safety-relevant (Graphic)" is set to "Yes" or

  "BAS010 General data" tab: The attributes "BAS1111 Quality-/Environment-relevant" and "BAS1113 Calibration-proofed measurement" are set to "Yes".

You can also set the attributes in the report via the context menu: "Graphical settings". See also section "RI Presentation" tab (Page 264).

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Symbol Editor".

New *V*P texts for implementations

New script texts: "*V*P SI:", "*V*P EI:", "*V*P EI:SYMBOL"

These *V*P texts are functionally identical to their counterparts without "I", i.e.: "*V*P S:", "*V*P E:", "*V*P E:SYMBOL".

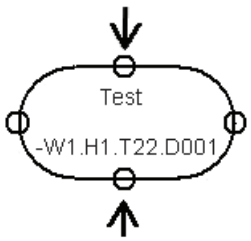
The new *V*P texts serve the following purpose: Instead of the P&ID object being used, its implementation (if there is one) is displayed and used in the graphical settings.

7.10 Display on reports

7.10.1 Neutral function

Display

Functions directly under the "@F Functions" level, known as neutral functions, are displayed as follows:



Function code

Is taken from the label (Label). Since this is empty for neutral functions, the Name is used.

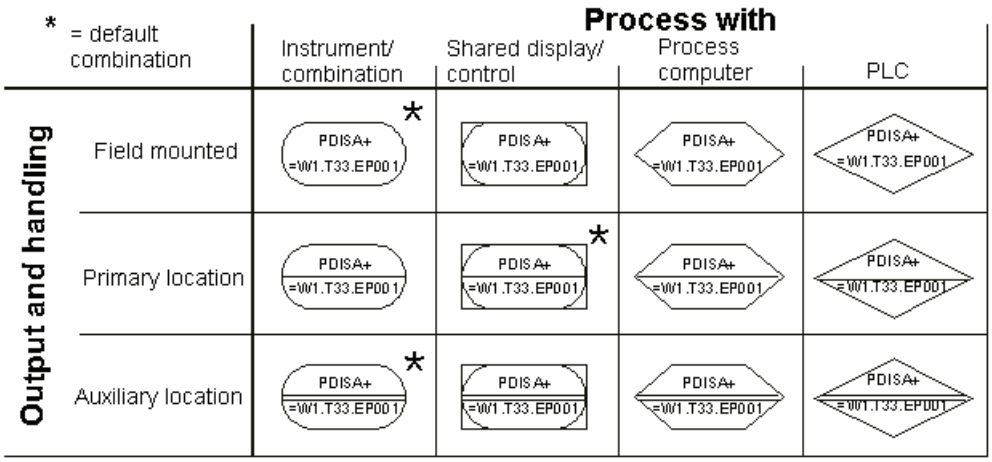
Device label

As a rule, only the position name and the function name are displayed. The FullName of the position with unit or subunit is only displayed if the flow diagram has been allocated to a different plant or unit/subunit than the function.

7.10.2 Basic symbol

The display of a function on the diagram is controlled using the settings on the "IC110 Function data measuring point" or "Function data actuating point" tab and the attributes "ICF101 Process with" and "ICF102 Output and operation".

The symbol on the diagram is adapted automatically, depending on which attributes have been set:



- Shared display/control

  This is "Type 4" according to DIN

- PLC

  This is "Type 5" according to ANSI

- Primary location

  This is "Type 4" according to ANSI

Attributes can be set in two ways:

1. Properties

   The display of the function on the diagram is controlled by the function properties on the "Function data" attribute tab.

2. Context menu

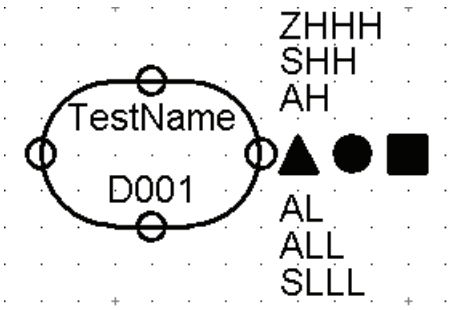
   The attributes responsible for the display are set via report's context menu:

   Select the function on the diagram, "Graphical settings > Output and operation" and "... > Process with" context menu commands.

The basic symbol also outputs the position under which the function is located and its label (measurement functions: label = function code; if label is blank: name).

7.10.3 Optional graphics

If required, you can also show optional additional symbols for alarm and relevance display at the measurement function:



Relevance display

To activate relevance display, select the "Graphical settings > Graphic for relevant adjustment > Yes" command from the function's context menu. The "Graphical settings" context menu expands and you can show or hide the following graphic symbols:

- "Safety-relevant (graphic)"
- "Calibration-proofed measurement"
- "Quality-/Environment-relevant"

The settings you make here are saved in the properties of the function on the "RI Presentation" and "BAS010 General data" tabs.

Alarm display

To show the alarm display, select the function and call the "Graphical settings > Alarm display > Yes" command from the context menu.

If an alarm display was input in the properties of the function on the "RI Presentation" tab, the corresponding values are output on the report.

To hide the alarm display, select the "Graphical settings > Alarm display > No" command from the context menu.

See also

"RI Presentation" tab (Page 264)

Symbols: Function symbols (Page 69)

Process Coupling and Process Connector of a Function

8

Definition

A function can be connected to the process either directly or via a process coupling. In both cases, a process connector is required at the function.

The process connector consists of the connectors via which the function is connected. The process connector data is saved on several tabs.

In I&C planning, you use connectors to transport signals and data. You find additional information on this topic in the "E&IC" manual.

The process coupling is either a valve (actuating function), an inline device (flow measurement), or a nozzle (other measurement functions).

Your administrator can configure the "P&ID" type report so that a function is created automatically together with your process coupling during placement on the P&ID.

8.1 Creating a process connector

In order to be able to connect a function to the process, the function needs a process connector. The process connector is implemented using several tabs that are created with the function:

- BAS02x Substance data
- BAS030 Process connector data
- BAS03x Process connector data

The tabs store information about the process connector, some of which they obtain automatically by means of controlled data transfer. See also section Data flow (Page 76).

How many of these tabs actually have a function depends on how many times they are connected with the process.

Actuating function:

The tabs are created on connection to a process connector. The number of tabs depends on the concrete actuating function and is strictly defined for this.

Measuring function:

The measuring function must have a function code in order to be able to create a process connector. In contrast to the creation process for actuating functions, the user must "initiate" the creation of the tabs by connecting the function to the process on the diagram or by incrementing the number of process connectors via its properties. See also section Number of process connectors (Page 81). The number of tabs is fixed in the case of measuring functions with an inline device, but is variable for measuring functions with a nozzle.

See also

"BAS021 Substance data" and "BAS031 Process connector" tabs (Page 266)

"Process connection data" tab (Page 267)

8.2 Data flow

Some of the data input at the device on the "Substance data" and "Technical data" tabs during P&ID planning (a pipe branch, for example) is also significant for the function. A controlled data transfer is used in COMOS to pass this data to the function.

The following applies to enable the substance data and process data to flow:

- Actuating function: Must be connected to the process.
- Measurement function: Must have a function code. Must have a process connector and must be connected to the process for the same reasons as the actuating function.

"Update connected devices?"

Depending on the version of the database, the information is passed across several stages. Examples:

Pipe branch -> Nozzle -> Measurement function

Pipe branch -> Valve

Pipe branch -> Actuating function

In these cases it is important to click the "Yes" button in response to the "Refresh connected components?" dialog. Reason: The discrepancy warning ("orange switching") only displays the discrepancies at the first level. None of the following objects detect that there is a discrepancy in the data flow.

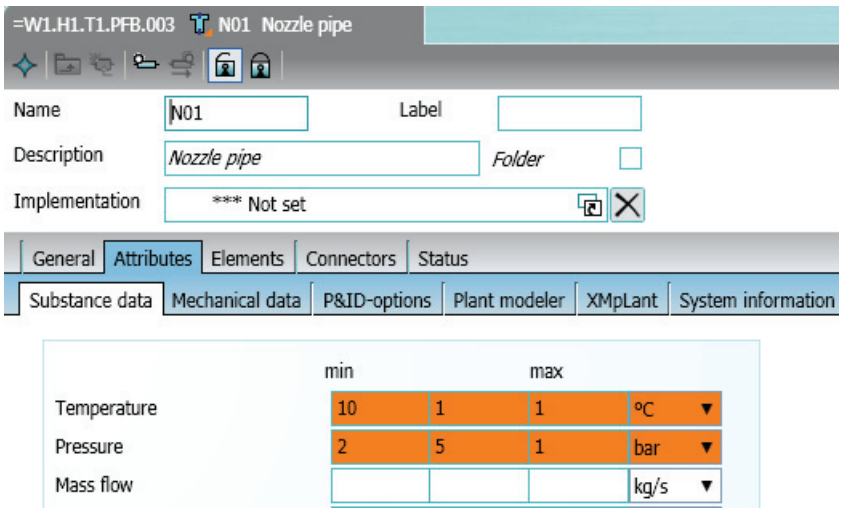
Subsequently updating connected components

If the connected components have not been updated, this can be done later on.

COMOS automatically detects discrepancies between the linked attributes of the first level and of the data source (the pipe segment, for example).

No attribute values are applied at this point in time. The relevant existing values at the pipe branch are simply compared with those of the process coupling so that any differences can be identified.

The corresponding attribute fields with discrepancies turn orange:



Right-click in a free area of the tab and select the "Refresh static links on tab" command from the context menu.

The attributes are aligned, and the fields are no longer orange.

It is, of course, not absolutely necessary that the data is transferred one to one. You can also enter your own values; discrepancy with respect to the data in the process coupling will then be indicated by means of the orange switching.

The data flow takes place at attribute level. This involves two attributes in which only the contents are matched/compared.

The units of the linked attributes do not need to match. The values are converted automatically.

Once the nozzle has been updated, the orange switching also appears in the measurement function.

8.3 Process coupling for measuring functions

Measurement functions use either a nozzle or an inline device as a process coupling.

In contrast to the tabs for actuating functions, these tabs are not generated when the measurement function is created, but only when you connect the function to the process.

Usually, measurement functions only have one process connector. Flow measurements, which can also have two process connectors, are an exception to this rule.

The basic principle is as follows:

For measurement functions that have a function code and work with substance data, a base object change takes place in the background each time the number of process connectors increases. The COMOS DB provides base objects for measurement functions with no process connectors, one process connector, or several process connectors. The number of process connectors determines the number of tabs.

8.3.1 Connectors

The COMOS DB has been prepared so that functions are first generated without connectors. The connectors are not created until you connect the function to another object on a P&ID report. See also section Working in the P&ID diagram (Page 99). COMOS decides automatically what type of connector is to be created.

Connector types

The function always takes the connector type of the object with which it is connected. When you interconnect two functions, "Single line. Simple action line" connectors are created at both functions.

The connector type determines what kind of connecting line is to be drawn.

Example

A measurement function is connected to a pipe on a P&ID report. A nozzle is created automatically as a process coupling. The nozzle has a "P&ID" type connector. The function also gets a P&ID connector and both objects are joined by a data line running through the connectors.

Special case

Objects are connected on a "PFD" or "P&ID" type interactive report. One object has a connector with the subtype "Simple action line" and the other has a connector with the subtype "Medium": The two objects can still be connected. An action line is created as a result.

The COMOS DB does not instantly offer users who work exclusively in the Navigator an option to connect the functions in the engineering view. Their administrator has to modify the database in such a way that functions already have the required connector when they are created in the Navigator.

"I" connector

Type "P&ID" connectors get the name "I" and hence are often known synonymously as "I" connectors. Two with each other connected connectors of this type are connected with a solid line. See also section Data lines / action lines (Page 97).

"W" connector

"Single line. Simple action line" connectors get the name "W" and hence are often known synonymously as "W" connectors. They are used to manage action line connections. These are connections between functions and devices that influence the process. They are displayed with a dotted line.

If there are several connectors of the same type, their internal numbering is incremented: "W1", "W2" or "I1", "I2".

See also section Data lines / action lines (Page 97).

Finding connected objects

In contrast to other objects of the "Device" object class, connected objects for functions ("Device" object class, "Function" subclass) are not searched for via the `OwnConnectors` collection but via the "`AllConnectors`" collection.

Connectors in I&C

In the I&C area the connectors are also used to transport signals and data. To this end, the "Signal of owner" option must have been activated in the connector properties.

You find additional information on this topic in the "E&IC" manual, keyword "Signal".

8.3.2 Connection via nozzle

"@01 > PID > 98 > 05 Nozzles + connections"

Used with:

All measurement functions except flow measurements ("F")

Nozzles are created in the unit structure underneath the pipe branch, not underneath the function.

Data flow of substance data

The nozzle has a "Substance data" tab. The pipe branch passes the substance data to the nozzle, from where it is taken over by the measurement function.

Data flow of process data

The "Process connector" tab of a measurement function inherits information from the nozzle, "Technical data" tab, which in turn inherits it from the pipe branch, "Technical data" tab.

8.3.3 Connection through inline devices

"@01 > EIC > 01 > 170 > F > F5 General flow sensor"

Used with:

Function code "F" - flow measurement

The inline device is created underneath the measuring function.

Data flow of substance data

The measuring function always takes the substance data directly from the pipe branch.

Data flow of process data

The "Process connector P1", "Process connector P2", and "Process connector data" tabs of a measurement function inherit information from the pipe branch, "Technical data" tab.

8.3.4 Changing the process coupling

Introduction

Your administrator can input the process coupling and an alternative process coupling at the base object of a function. This enables you to change the process coupling of a function subsequently.

Procedure

1. Select the required process coupling on the report.
2. Select the "Change process connector to" command from the context menu.

Result

The process connector is changed accordingly.

Example

You have created a function on the report, connected it with the process, and subsequently set the function code to "F" (flow measurement). The function automatically switches its base object, but its process coupling does not change. You can use the "Change process connector to" command to provide the function with an inline device instead of a nozzle.

8.4 Process coupling actuating function

Connection through a valve

"@03 > PID > 310 > EN > D > 04 > 60 Valves" or "@03 > PID > 310 > ANSI > D > 04 > 60 Valves"

Used with: All actuating functions

The valve is created underneath the actuating function or is moved there as soon as it is connected with the actuating function.

Data flow of substance data

Both the actuating function and the valve take the substance data directly from the pipe branch.

Data flow of process data

The "Process connector P1", "Process connector P2", and "Process connector data" tabs of an actuating function inherit information from the pipe branch, "Technical data" tab.

Similarly, the "Technical data" tab of a valve inherits information from the pipe branch, "Technical data" tab.

8.5 Automatically creating process coupling

When a function is placed on a diagram and connected to the process, a process coupling can be automatically created and connected to the function. Which process coupling is created is determined in the base object of the function on the "Attributes > System" tab, in the "PCC process coupling component" attribute.

The following parameter must be set in the options script of the report template to ensure that the process coupling is automatically created:
`EnableProcessConnection = TRUE.`

The automatic creation of the process coupling can also be set separately for measuring functions and actuating functions:

- Only for measuring functions:

  ```
  EnableProcessConnectionSensor = True
  ```

- Only generate for actuating functions:

  ```
  EnableProcessConnectionActor = True
  ```

In the case of measurement functions, the process coupling is automatically connected not only visually on the report but also to the function (via its connectors). That is not the case for actuating function. Reason: The valve still needs to be prepared further and gets a drive. The function then needs to be connected with the drive.

8.6 Number of process connectors

The number of process connectors is fixed in the case of flow measurements. They are connected to the process by means of an inline device. The number of process connectors is also fixed for actuating functions.

However, measurement functions with a nozzle as a process coupling can change the number of process connectors they have. The number of process connectors determines how many "BAS02x Substance data" and "BAS03x Process connector" tabs the measurement function has. Furthermore, the measuring function changes its base object along with the number of process connectors.

Example: Base objects for measuring functions for density according to EN/DIN:

"@03 > BAS > IEC > @F > 01 > D Density" (no process connectors)

"@03 > BAS > IEC > @F > 01 > D > 1 Density" (one process connector)

"@03 > BAS > IEC > @F > 01 > D > 2 Density" (two process connectors)

Creating the measurement function in the Navigator by selecting the "New" command from the context menu produces a measurement function with the base object "@03 > BAS > IEC > @F > 01 > D Density" at which no connectors have yet been predefined. In other words, the function will have no "BAS02x Substance data" or "BAS03x Process connector" tabs.

If one process connector is then assigned to the function, its base object changes to "@03 > BAS > IEC > @F > 01 > D > 1 Density" and it will have the "Substance data" and "Process connector" tabs.

There are two ways to increase the number of process connectors:

- Automatically (when connecting)

  If a measuring function is connected on the P&ID report it is checked whether an additional process connector is required. If so, then it is created automatically. The base object changes automatically (in the case of one connector, to the base object "@03 > BAS > IEC > @F > 01 > D > 1 Density).

- Manually through the properties of the function

Tab: "Function data measuring point", attribute: "Number of process connectors"

The base object changes automatically after saving.

The following applies in both cases: The measurement function must have a function code.

Using P&ID

9

9.1 Importing AutoCAD data

AutoCAD drawings can be imported into a P&ID report using drag&drop.

Alternatively, you can right-click on the report and select the "Options > Import" command from the context menu. Then select the "Autocad (*.dwg; *.dxf)" entry from the "Please select drawing type:" list.

9.2 Process units

Process units are organizational units that are used to resolve a user-defined process technology task.

Drag&drop the process units onto the diagram. The process units are symbolized with captioned boxes.

To change the automatically generated name, proceed as follows:

1. Open the properties.
2. Right-click on the object.
3. Select the "Properties" context menu.
4. Enter a meaningful designation under "Name".

The size of the symbols for the process units can be changed.

9.3 Defining streams

Substance streams are managed as COMOS objects and as such possess all the usual properties (attributes, connectors, etc.). Substance streams are thus aligned objects; they begin at the first mouse-click point, end at the second mouse-click point and are given a direction arrow (direction of flow).

9.4 Working with pipes

9.4.1 Using the "Connection" tool to create pipes

Creating new pipes and extending existing ones

Proceed as follows to establish a connection with the "Connection" tool, thus creating a pipe:

Activate the "Connection" tool to draw in the pipe route: Left-click to specify the start point and left-click again where you wish to insert pipe bends. Right-click to complete the pipe run.

Effect:

- The pipe is created in the P&ID report. In the Navigator, a pipe object and below that a pipe branch is created, along with a pipe segment another level below that is placed on the P&ID report.
- If a pipe is selected in the Navigator while you are drawing with the "Connection" tool, a prompt is displayed allowing you to decide whether the new pipe branch is to be created with the associated pipe segment below this pipe or a new pipe is to be created.
  - To add the new pipe branch with associated pipe segment the selected pipe, click "Yes".

    The new pipe branch takes on the properties of the pipe.
  - To create a new pipe branch and below it the new pipe branch with the associated pipe segment, click "No".

    You may need to equip the pipe branch with attributes in more detail.
- If a pipe branch is selected in the Navigator while you are drawing with the "Connection" tool, a prompt is displayed allowing you to decide whether the connection drawn is to be assigned to the selected pipe branch:
  - To insert an additional pipe segment to the selected pipe branch, click "Yes".
  - To insert a new pipe branch with an associated pipe segment, click "No".

This makes it possible to create a new pipe or to extend existing pipes.

Using the "Connection" tool to interconnect two pipes

Initial situation:

Two pipes that are not interconnected are placed on the report. A pipe and below it a pipe branch always with associated pipe segments is created for each pipe in the Navigator. None of these objects is selected in the Navigator.

If you use the "Connection" tool to draw a connection between the two pipes, the following happens:

- At least one main connector is involved (output or input of that pipe):

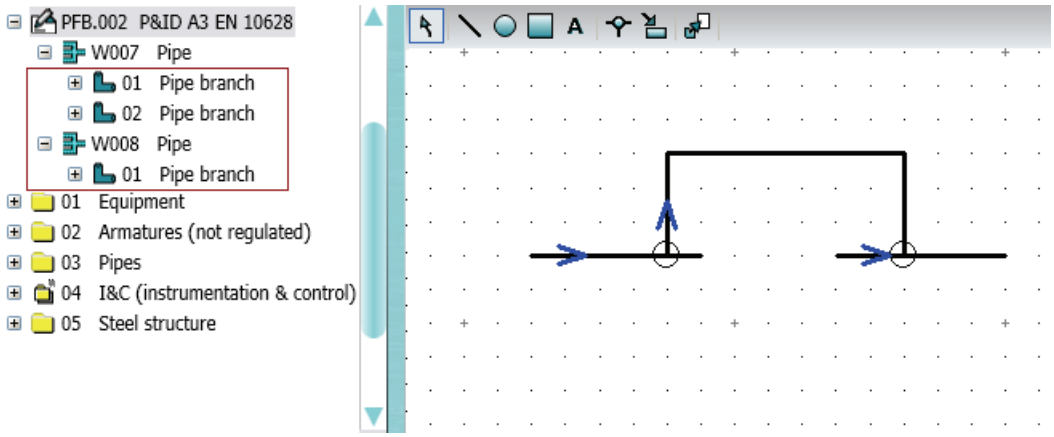
  A dialog opens, prompting the user to select a pipe.

  Result: A new pipe branch with an associated pipe segment is created. The new pipe branch and the pipe branches that previously existed are collected below the selected pipe. The second pipe is deleted.

  "Both pipes ..." option activated: A third pipe is created, and below it the new pipe branch with the associated pipe segment.

- The connection is made exclusively using dynamic connectors:

  The direction of the connection is evaluated. The new pipe branch is created parallel to the pipe branch with which the input of the new pipe branch is connected:



Oblique pipes

Activate "Connection" tool, select "Oblique" option:

Now you can also create oblique piping instead of only rectangular piping.

You find additional information on this topic in the "Applications" manual, keyword "Oblique connections".

9.4.2 Selecting a standard pipe for connection

Requirement

The corresponding program code has been commented in by your administrator. Various pipe and channel structures have been defined.

A report has been created and opened. No objects are selected.

Procedure

To select a standard pipe for connection, proceed as follows:

1. Right-click on the report.
2. Select the "Standard pipe for connecting > <required pipe>" command from the context menu.

Result

If you select the "Connection" tool from the report toolbar and draw a connection, the object defined as the standard object is used to create the connection.

9.4.3 Placing pipes from the Navigator

- From the base data:

  The base object (pipe branch or pipe segment) can be dragged directly onto the diagram. The "Connection" tool is activated and constructed with the dragged pipe until you exit the "Connection" tool.

- From the engineering data:

  The pipe branch or the pipe segment is already in the engineering data. They receive a `DocObj`.

  In addition to pipe segments and pipe branches, you can drag-and-drop to place pipes on the report from the Navigator. When you select a pipe, a new pipe branch with an associated pipe segment is created when you start drawing. When you select a pipe branch, a new pipe segment is created when you start drawing.

After placing the pipe, it is initially displayed on the diagram with a straight symbol. Use the grab points (grabs) of the pipe to bring it into the desired shape or to connect it to other components.

9.4.4 Pipe cut mode

9.4.4.1 Automatic separation of the pipe with fittings and intermediate components

If you place a fitting or an intermediate component on a pipe, the pipe is split in the background:

In the report the pipe is visually interrupted by the component. At the same time, COMOS automatically creates new pipe objects in the Navigator that mirror this break. Which pipe objects are created depends on the pipe cut mode that was specified for the component by your base data administrator.

The following pipe cut modes are possible:

Segmentation, e.g. for valves

- Case 1: The insert is placed on a pipe branch; there are no pipe segments yet.

  Effect: Two pipe segments are created below the pipe branch. The component is joined to the pipe segments via the connectors.

  The pipe branch is no longer placed on the report; instead its two pipe segments are placed on the report.

- Case 2: The component is placed on a pipe segment.

  Effect: A second pipe segment is created. The component is joined to the pipe segments via the connectors.

  In the COMOS DB: Valves and inline devices

Creating a new pipe branch, e.g. for reducers

A second pipe branch is created parallel to the pipe branch on which the component was placed or on the pipe segment on which the component was placed. This second pipe branch and the original pipe branch or its pipe segment are joined to the component via their connectors.

In the COMOS DB: Pipe parts (reducers, tees)

Creating a new pipe container and a new pipe branch, e.g. for pumps

A second pipe is created and below it a pipe branch. The pipe is located parallel to the owner pipe of the pipe branch or the pipe segment on which the component was placed. This second pipe branch and the original pipe branch or its pipe segment are joined to the component via their connectors.

In the COMOS DB: Pumps, compressors

"Attributes > System information" tab

To check the pipe cut mode of a component, open its properties window and check the "Pipe cut mode" attribute on the "Attributes > System information" tab.

The pipe is separated automatically.

9.4.4.2 Cutting a pipe

Components can have different pipe cut modes (attribute "SYS.PIA602 Pipe cut mode"). Depending on the pipe cut mode, new pipe segments, new pipe branches or new pipes and pipe branches are created. See also section Automatic separation of the pipe with fittings and intermediate components (Page 86).

When separating the pipe into pipe branches, then the existing pipe branch is copied. Any existing elements are deleted. Only elements that cannot be placed on a P&ID (Property `RIClass = ""`) are copied as well.

Creating nozzles in the case of pipe-cutting actions

When a pipe is dragged over a pipe-cutting component to a vessel, the connection to the vessel is checked and a nozzle is generated before the connection is broken.

Cut mode

The "PIA602/Cut mode" attribute is now evaluated for components with the "fitting (IsFitting)" property. The cut mode now always sorts the components parallel to the component that it is to be cut:

- Segment-cutting: Parallel to pipe segments
- Branch-cutting: Parallel to pipe branches
- Pipe-cutting: This is a special case, since it would collide with the whole system of categories. There should be no components located parallel to the pipes, as the pipes should be the highest objects in the Pipes categories folder.

Therefore, components that have been labeled as "IsFitting" but are located parallel to the pipe due to the cut mode that has been set are moved to the categories folder when object assignment is carried out. Cut mode can be found in, for example:

"@01 > PID > 01 > 02 > 03.SYS.PIA602 Pipe cut mode"

Pipes across sheets

When a pipe is cut, the recursive movement of the connected objects using reference pipe segments continues.

Background: Pipe branches can be moved across multiple sheets using page references. When the pipe branch is cut on one of the sheets, the pipe segments are assigned to the new pipe branches.

The successor segment is recognized on the basis of the following definition:

The current pipe segment has a free O1 connector: A search is made for the first pipe segment with the free I1 connector that occurs in the OwnerCollection after the current one.

Pipe cut mode

When you place P&ID components on a pipe, the pipe is cut. New pipe segments, new pipe branches, or new pipes and pipe branches are created depending on the pipe cut mode of the component.

You specify the pipe cut mode in the component properties. The following standard table is stored in the base project for the "Pipe cut mode" attribute:

"@System > PipeCutMode"

9.4.4.3 Automatic merging of the pipe when removing components

If the installation of a component in a pipe results in a new pipe branch or a new pipe segment being created, the removal of the component must have the same effect in reverse, i.e. this pipe branch or pipe segment must be deleted again.

Deleting a component

- When you delete a component which cuts the pipe into pipe segments, one of the pipe segments to which it is connected is also automatically deleted in the Navigator.
- When you delete a component which cuts the pipe into pipe branches, the "Select pipe branch" window opens. Select the required option to be executed when the pipe branches are merged.

  If you select the "Keep both pipes (with break)" option, both pipe branches are retained.

  The pipe branches are no longer interconnected, either in the database or on the report. A gap remains on the report where the component had been placed previously.
- When you delete a component which cuts the pipe into several pipes, the "Select pipe" window opens.

  The selected pipe is retained.

9.4.5 Extending a pipe

Requirement

A pipe segment has been placed on a report.

`EnableContinueDrawingConnection` has been activated by your administrator.

Procedure

To extend a pipe, proceed as follows:

1. Select the "Connection" tool from the report toolbar.
2. Select one of the following options:
   - Click on the end of the pipe segment that you want to extend you. Extend the pipe segment as desired.
   - Click before the start of the pipe segment and extend the pipe segment.

Result

If you click exactly at the end of the pipe segment, the pipe segment is extended. The flow direction must be correct. If you extend the pipe segment at an angle to the existing pipe segment, a new pipe segment is created.

You cannot extend the pipe segment beyond a branch with a dynamic connector.

9.4.6 Deleting pipes

Deleting pipe segments

If all pipe segments of a pipe branch are deleted, the pipe branch is also deleted automatically if no checked-in elements can be found below it.

Deleting pipe branches

- Owner: Document:

  If all branches of a pipe are deleted, the pipe is also deleted automatically if no checked-in elements can be found underneath it.

- Owner unlike document:

  The pipe object is retained, even if all its pipe branches are deleted.

Deleting a component with the cut mode:

- "Segment separative":

  One pipe segment is deleted, one pipe segment is retained.

- "Pipe branch separative":

  The "Select pipe branch" dialog opens.

  The user determines which pipe branch is to be retained.

  If the "Keep both pipes (with break)" option is activated, the component is deleted. Both pipe branches are retained. They are not connected with one another, neither in the database nor on the report. A gap remains on the report where the component had been placed previously.

- "Pipe separative":

  The "Select pipe" dialog opens. The pipe whose pipe had been selected by the user is retained.

Closing gaps in pipes

When deleting objects from pipes the pipes may also be connected at a corner. This also functions for diagonal pipes and pipes that are parallel to one another.

Rule for pipe route:

- Extend both ends of open pipes and check if there is a cut point. If there is a cut point, it is taken as the bend point of the new common pipe.
- If there is not a cut point, an automatic pipe route is generated between the two open ends.

Special case, three-way valve: The gap remains.

9.4.7 Other points concerning pipes

New routing

Interactive report, context menu for a pipe "Options > Reroute"

Application: If the pipe route is no longer ideal after long periods of work on the P&ID and repeated manual moving of pipes. The selected pipes are recalculated and redrawn. Oblique pipes are converted to rectangular connections.

Assigned components

Pipes can have secondary objects ("Subdevice") that are not connected with the main pipe. Such objects are primarily supports, clips, and so forth.

- Subdevices of this type are now also selected when the main element is selected.
- Subdevices are also moved when the main element is moved.

9.5 Interactions between P&ID objects

9.5.1 Connecting pipes with components

There are two ways to connect components with a pipe on the P&ID diagram:

1. First draw the pipe, then place the components on the pipe.

   Result:

   The pipe is automatically connected with the components and separated according to the pipe cut mode of its components. See also section Cutting a pipe (Page 87).

   This only works with components for which suitable connectors have been predefined in the symbol, e.g. valves and pumps. It does not work with vessels and columns.

2. First place the components, then draw a pipe:
   - Components with predefined connectors:

     If the connectors of the components to be connected are all in a line: Using the "Connection" tool, you can easily draw over the connectors of the placed device. COMOS generates the required pipe objects, depending on the pipe cut mode of the components, and connects the pipe with the components.
   - Component with dynamic connectors:

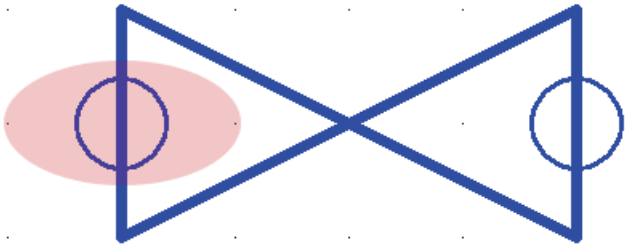
     It is not possible to draw one line that connects all components. Instead, you must draw the incoming pipe and then the outgoing pipe.

Connecting to a fixed connector

Background:

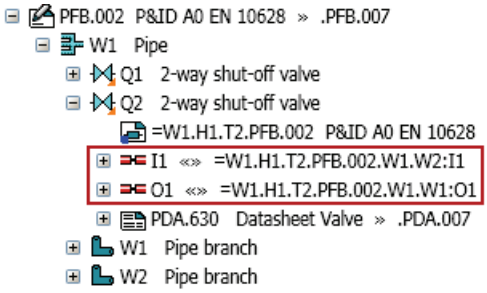
In the symbol script of the base object you can determine which connectors a component should have from the beginning. These connectors, which are predefined on the base object, are also called fixed connectors.

They are displayed on the P&ID right from the start, regardless of whether or not they are already connected with another object (= set):



Using a fixed connector to connect a component with a pipe:

- On the report: The pipe snaps to the connector. If you subsequently move the pipe, the component is automatically moved as well.
- In the Navigator: The two objects are automatically interconnected by means of their connectors ("Connector" system type):



- Use the "Navigate > Connections" command to navigate to the connected connector.

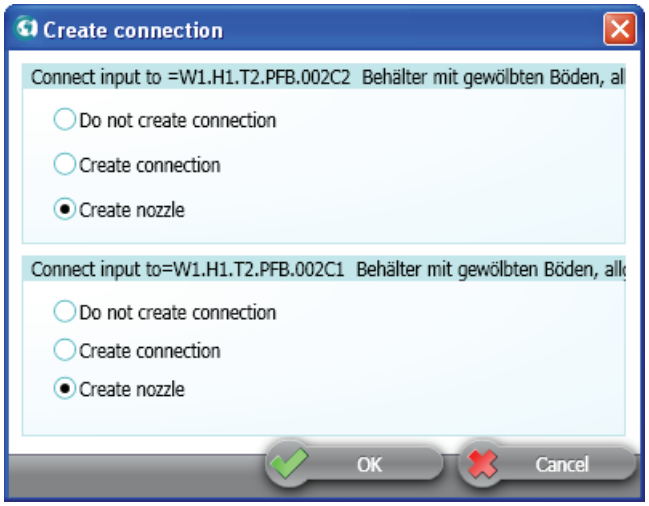
Connection to dynamic connectors

Background:

All P&ID components, pipes, and objects for creating a connection (e.g. nozzles) have dynamic connectors. These connectors do not have to be predefined in the symbol script of the base object; they are created dynamically when required. Some components (e.g. vessels) only have dynamic connectors.

Use the "Connection" tool to draw in a pipe and connect it to the device by right-clicking anywhere on the device to define the end point of the pipe branch (if necessary, disable the snap to grid function by pressing Shift).

The "Create connection" dialog opens:



Here the user decides whether and how the pipe and the component are to be connected:

- "Do not create connection":

  Self-explanatory. It can look as if the objects are interconnected on the diagram.

- "Create connection":

  The pipe and the component are joined directly by means of connectors, both on the diagram and in the database. If necessary, a new connector is generated dynamically at the component.

- "Create nozzle":

  A nozzle is created on the report and in the database. The pipe and the component are joined to the connectors of the nozzle and thus indirectly also to one another.

  This option can only be activated if the base object of the component has a nozzle with the name "NOZZLE".

  If available: Activated by default

Dynamic connectors are initially retained after disconnecting the connector. To delete the connector point after disconnection, select the "Options > Restore the original symbol" command from the context menu.

Dynamic connectors are used in particular with vessels.

See also

Options in the "Create connection" window (Page 261)

9.5.2 Sorting components underneath the pipe

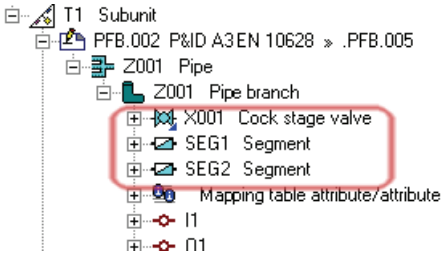
You can configure pipe fittings so that they are automatically sorted underneath the pipe that they are connected to.

To do this, activate the "Sort parallel to cut pipe element" option on the "Attributes > System information" tab in the component properties.

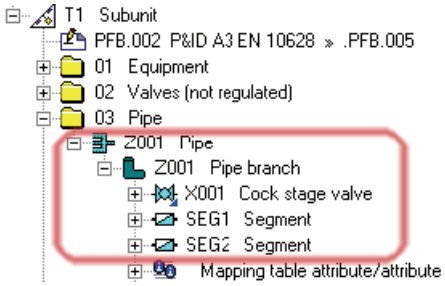
Example

When you place a valve on a pipe, the valve is not created parallel to the other components.

Instead, the valve is created parallel to the pipe object to which it is connected, i.e. parallel to the pipe segments and thus always below the pipe or the associated pipe branch.



This also applies if you are working with categories.



9.5.3 Automatically sorting components

Introduction

When you are working on a "P&ID" type report, the components you create there are normally created in the Navigator underneath the report. You assign the components to a unit at a later stage. See also section The "Assign object" tool (Page 119).

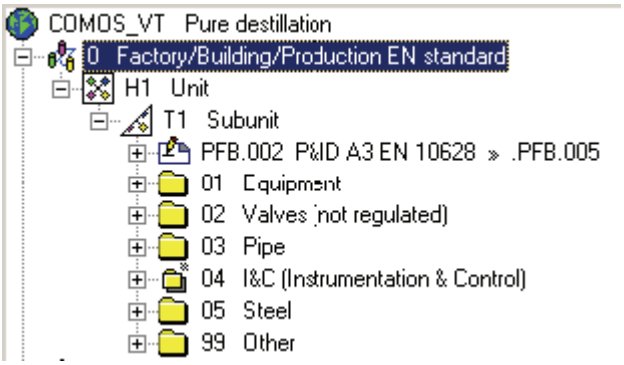
Project settings

Your administrator can configure the project so that the P&ID components are automatically sorted into the unit structure by component type when they are placed on the report. See also section Report template script options (Page 219).

Categories

Categories are special folders. Each category only incorporates objects of a particular component type.

In our example project, categories have been predefined under the subunit:



When creating the subunit, the categories are automatically created as well.

Whenever you place a component on the report, it is sorted into the corresponding category. If you assign the subunit to a component as its owner, rather than being moved directly underneath the subunit, the component is sorted into a suitable category underneath the subunit.

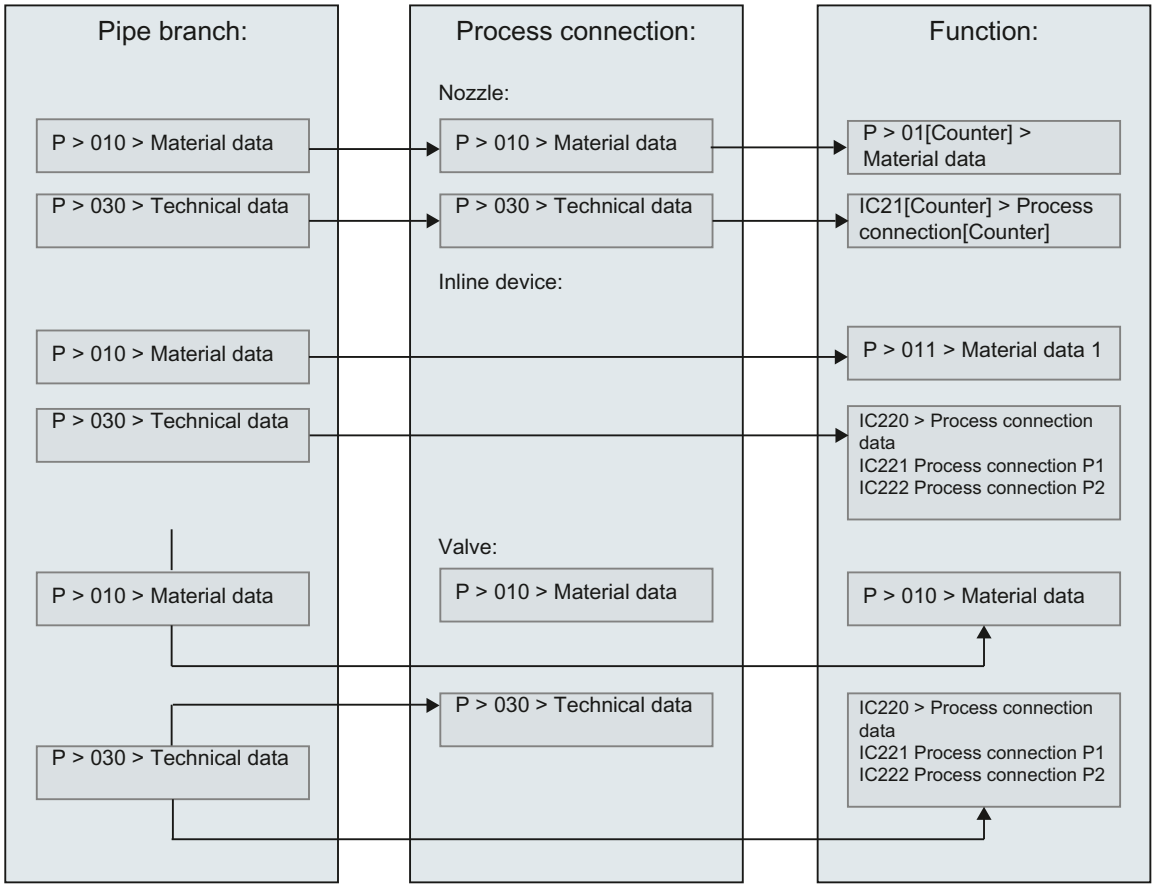
9.5.4 Automated data flow

There is an automatic data flow between the process, process coupling and function.

Overview of data transfer

Some of the data input for the process devices on the "Substance data" and "Technical data" tabs during P&ID planning is also significant for the functions.

In order to pass this data on to the functions, a controlled form of data transfer takes place within COMOS:



Technical background

This data transfer is done by means of the static link COMOS technique. A data source forwards attribute values to the linked attributes of another object. This other object can be connected to the data source via its connectors, but does not have to be.

All these processes run in the background.

See also

Initiating the data flow on the P&ID report (Page 52)

9.5.5 Updating connected components subsequently

Requirement

You have changed the properties of a pipe branch. You have not updated the components that are directly connected to the pipe branch via their connectors.

Procedure

To update the components connected to the pipe branch subsequently, proceed as follows:

1. Open the properties of the component.

   Attribute values which differ from the linked attributes of the pipe branch turn orange.
2. Right-click in the free area of tab.
3. Select one of the following options:
   - To align the orange attribute values on all tabs with the values of the linked attributes, select the "Refresh static links on all tabs" command from the context menu.
   - To align the orange attribute values on this tab with the values of the linked attributes, select the "Refresh static links on tab" command from the context menu.

Result

The new values are written to the attributes and the orange highlighting disappears.

9.6 Data lines / action lines

Data lines and action lines are purely graphical connections; no objects are created for them. They are generated if:

- Two P&ID objects are joined via their W/I connectors (action line)
- A P&ID object with a W/I connector is joined with a pipe (action line)
- a function is joined with another object. A data line or an action line can be created depending on the connector type of the object with which the function is joined. The following applies:
  - Effect line: Created between two connectors of the type "Single line. Simple effect line".
  - Data line: Created between two connectors of type P&ID or of type "Single line. Medium".

See also section Connectors (Page 78).

When a measuring function is connected to a process device, the resulting data line initially runss from the process to the measuring function. When an actuating function is connected, then the user must check for himself or herself whether the action line points in the correct direction.

The direction can be changed at any time by the user via the context menu: "Options > Change flow direction". The design direction is modified in the case of action lines, as it is for pipes.

If objects are connected on a "PFD" or "P&ID" type interactive report and one object has a connector with the subtype "Simple action line" and the other has a connector with the subtype "Medium", both objects can still be connected. An action line is created as a result.

When creating a graphical connection, the graphical properties for the line are read from the standard table "@ConnectionTypeI" of the system project. You can find additional information on this topic in the "Comos Administration" manual, keyword "@ConnectionTypeI (single line)".

Segmenting a data line / effect line

If you segment a data line / effect line, COMOS converts the part that is not connected to the measuring function / actuating function automatically into a pipe branch.

Inheriting graphical properties

Graphical connections transfer their graphical attributes (line type, line thickness, line color, and layer) to new connected graphical connections.

In other words: If the user has already formatted an action line and then draws a new action line that is connected to it, the new action line takes over the user formatting.

Forcing action lines

Attributes tab "SYS", attribute "Behavior".

This attribute controls the behavior of components in P&ID.

SYS.Behavior = "I"

Sets the RIClass to "I". If you evaluate the component in the Object Debugger, you get Device.RIClass = "I".

As a result the component behaves like a measurement function: When it is connected with other components or a pipe, an action line is drawn instead of a standard line.

9.7 Working with functions in the engineeringf view

9.7.1 Overview

First of all the function needs to be created. This can be done in a number of ways:

- Via the bar containing the report's base objects The available functions do not have a function code yet. See also section Creating measurement functions (Page 62).
- Select the base object node with the function base objects, select and create the required function with matching function code.
- There is already a position in the engineering data: "New" context menu for a position according to DIN/IEC or the corresponding menus for a position according to ANSI:

  Measuring functions and actuating functions are available in these menus. The function is created in the Navigator underneath the position.

  The function inherits the label of its position (assuming that the label of the position is not blank) and through this has a function code from the very beginning.

Next, the function must be connected with the process. This is done by connecting the function by means of its process connectors with a process coupling. For this, you place the function on P&ID report or you alternatively work in the Navigator.

When working with actuating functions, you should also regard that the associated actuator still needs to be created.

See also section Creating positions (Page 58).

9.7.2 Working in the P&ID diagram

Measurement function

Open the P&ID diagram that is located below the part unit and drag the function from the Navigator onto the diagram by using drag&drop. Use the "Connection" tool to connect the function to the process.

If the administrator has input a process coupling in the base object of the function as part of the preparation for the project, the process coupling is automatically created when the function is connected (nozzle, or inline device in the case of flow measurements). See also section Automatically creating process coupling (Page 81). The function and the process coupling are automatically interconnected via their connectors. See also section Connectors (Page 78).

The result of the connection: New tabs are created for the function, on which the data for the process connector is saved. See also section Creating a process connector (Page 75) and section Data flow (Page 76).

A solid, objectless line is created – a so-called "data line". The connection always runs from the process coupling to the measurement function. If they were connected in the opposite direction, the line is turned around automatically. The direction can be changed by calling the context menu "Options > Change flow direction".

Actuating function

The same applies to the actuating function: The process coupling (= valve) can be created automatically when connecting the function with the process. But since the valve still needs to be prepared further, the following procedure should be used:

1. First, create the actuating function.

   In the COMOS DB, no icons for actuating functions have been predefined on the symbol bar in the P&ID report.

2. Therefore, we recommend that you simply click the required position in the Navigator and use the position's "New" context menu to select an actuating function.

   The actuating functions listed in the menu are sorted by the valves that are to be created below them.

   Alternatively, you can switch to the "Base objects" tab in the Navigator and select a suitable base object under the base object node "@03 > BAS > IEC > @F > 02 Actuating functions".

3. Next, create the required actuator in the Navigator below the actuating function and place it on the diagram.

   The actuator can also be created via the diagram's symbol bar or in the base data. In the base data of the COMOS DB, valves are located underneath the following base object nodes:

   – "@03 > PID > 310 > EN > D > 04 Valves"

   – "@03 > PID > 310 > ANSI > D > 04 Valves"

4. Select the valve on the diagram and continue the preparation by selecting "Graphical settings > Drive > <required entry>" from the context menu.

5. Now place the actuating function on the diagram and connect it to the valve drive using the "Connection" tool.

   A dotted, objectless line known as an "action line" is drawn.

You need to check yourself whether this line points in the right direction. The direction can be changed by calling the context menu "Options > Change flow direction".

The objects are automatically connected in the database as well through their connectors. See also section Connectors (Page 78).

The result of the connection: The function now has a process connector. See also section Creating a process connector (Page 75) and section Data flow (Page 76).

See also

Display on reports (Page 72)

9.7.3 Working only in the Navigator

Connecting a function with the process only through the Navigator is somewhat more complicated than the method using the P&ID diagram.

Creating a process coupling

First of all you need to create the process coupling below the function. The process couplings of measurement functions are located in the base data under the following base object node:

- Nozzle: "@01 > PID > 98 > 05 > 01 Nozzle"
- Inline device: "@03 > BAS > IEC > @F > 01 > F > 1 Flow, flow rate"

In the case of actuating functions the process couplings are below:

- "@03 > BAS > IEC > @F > 02 Actuating function" or "@03 > BAS > ANSI > @F > 02 ANSI actuating function"

  The valves are prepared in such a way that they can also be used as actuators.

Open an additional Navigator and create the process coupling by dragging the required base object onto the "Units" tab, below the function.

The COMOS DB is set up so that the valves can also be created using the "New" context menu for actuating functions.

Actuating function: Prepare valve

Set the required attributes on the "Accessory" tab in the valve properties. In addition, a drive has to be created below the valve. In the COMOS DB there are already predefined drives in the "New" context menu for the valve.

Their base objects are located under:

"@01 > PID > 98 > 01 Actuators for valves"

Connecting and creating process connectors

Next, the function and the process coupling or, in the case of the actuating function, the drive, have to be interconnected. This is done through the connectors of these objects. The connectors must already have been created beforehand by the administrator in the base data. See also section Connectors (Page 78).

The result of the connection: New tabs are created for the function, on which the data for the process connector is saved. Some of the data written to these tabs comes from the process. See also section Creating a process connector (Page 75) and section Data flow (Page 76).

The number of process connectors increases with each connection that is made. See also section Number of process connectors (Page 81).

9.8 Copying and pasting on P&IDs

9.8.1 Overview

Apply

Select one or more objects on the P&ID, call "Copy" and "Paste" from the context menu, and select a paste mode. Specify with a right-click where the symbol of the copy is to be placed on the diagram.

Paste location and paste mode

Exactly where the copied objects are pasted in the Navigator and which objects are copied depends on a number of factors. See also section Basic algorithm (Page 104).

The examples in the following sections are unlikely to occur in normal work with COMOS. They were chosen to clearly illustrate the effect of copying and pasting.

9.8.2 The paste modes

- "Below document":

  The copy is pasted into the Navigator below the document.

- "Parallel to document":

  The copy is pasted parallel to the document in the Navigator.

- "At same position":

  The copy is pasted parallel to the original in the Navigator.

- "Keep object":

  Is only activated if the "Paste" command is called on a different diagram than the one on which the original is placed. Only the symbol is copied, not the engineering object. The in the report generated copy of the symbol points to the same engineering object as the original symbol which now has two `DocObjs`.

- "Automatically":

  Drawing type specific behavior:

  – "PID" drawing type: The copy is pasted into the Navigator below the document.

  – All other drawing types (for the EE, FLUID modules, for example): The copy is created parallel to the original.

    Exception: If the original was located parallel to the document and is copied across all documents, then the copy is not located parallel to the original but instead parallel to the new document.

  – "FUP_IEC" drawing type:

    The attribute "SYS.MultipleUse Object can be placed multiple times" is evaluated. Activated: The new object is created in the Navigator parallel to the original. Deactivated: Same behavior as in Keep object paste mode.

The objects that were cut are always pasted in according to the paste mode if:

- They are not connected to a pipe when pasted
- They are connected to a pipe when pasted, but the pipe does not call the `SetOwnerbyDeviceConnector` function in the Connect script

  See also section Basic algorithm (Page 104).

9.8.3 Basic algorithm

Interaction of the factors relevant for pasting

As stated before, the entire paste behavior and the paste location are determined by a number of factors:

- Is the object connected with a pipe when pasted, i.e. is the Connect event of the pipe triggered and is the `SetDeviceOwnerbyConnector` function called there?

  Although this case does not occur in the COMOS DB, it is technically possible to trigger the `SetDeviceOwnerbyConnector` function on other objects as pipes too.

  Function call:

  ```
  Workset.lib.ri.SetDeviceOwnerbyConnector connector,true
  ```

- Which objects were selected?

  A number of object types have a special form of paste behavior:

  – Functions

  – Components that are labeled as fittings (`SYS.IsPipeFitting = 1`), for example, valves and reducers.

  In addition to the above, a number of objects presuppose a particular owner structure (e.g. functions and pipes). In this case, in addition to a copy of the original being pasted, a copy of the owner structure is sometimes created as well.

- Which paste mode was selected?

  The diagram below gives you an overview of the interaction between these factors.

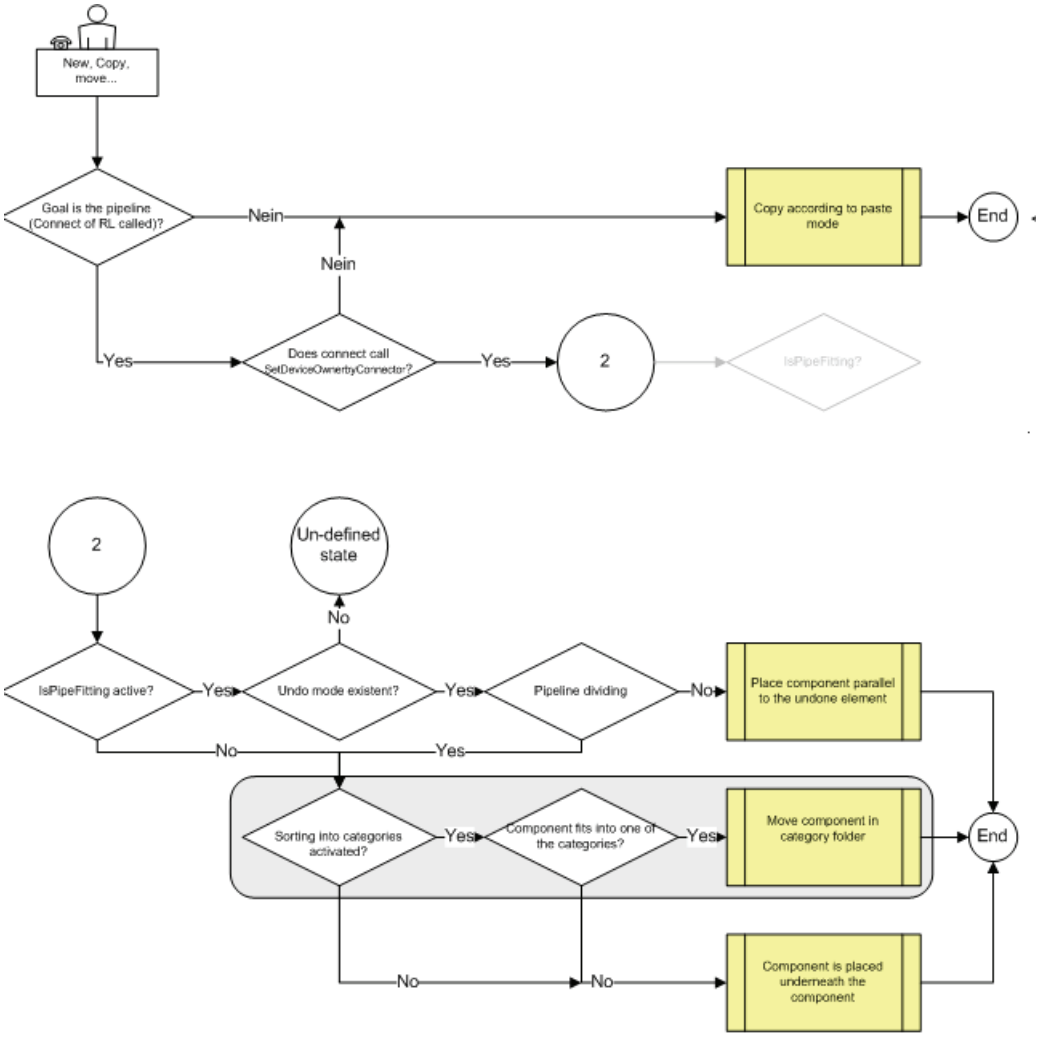
  The exact behavior of pipes and functions during copying and pasting is not part of this overview. Information regarding this can be found in separate sections:

  – In regard to pipes: see also section Pipe segments and pipe branches (Page 115).

    Note: If an object is placed on a pipe when pasted, the pipe is separated according to the pipe cut mode of the object.

  – In regard to functions, see also section Functions / positions (Page 109).

Overview: Basic algorithm for copy and paste



The base algorithm can also come into action when moving. Example: A fitting component is placed on the report but is not fixed initially. The fitting component is later incorporated into a pipeline.

Despite no object being copied or newly created at this time, the base algorithm is processed from the according point . In this case, `SetDeviceOwnerByConnector`.

See also section Basic algorithm (Page 112).

Attributes

All attribute values are copied.

Name, label, and description

Name and label:

Whether or not the object retains its name and label when pasted depends on how the settings are made for the behavior for unique names and labels.

You find additional information on this topic in the following manuals:

- "COMOS Administration", keyword "Unique name across folders"
- "System Type Properties", keyword "Unique label" and "Object behavior"

Name:

- The name is retained if there is no other object with the same name below the owner below which the object was pasted.
- A new name is generated if there already is an object of the same name below the owner:
  - A name mask is prepared at the base object: The mask is always used.
  - No name mask is prepared at the base object: The name of the object is incremented (BSP: Object1: "TEST1", Object2 "TEST1_1").

Label:

Same as name.

Description:

Taken over from the original.

Connection information

Original A is connected with original B. A and B can be copied and pasted: The connection information is copied: Copy A is connected to copy B.

If an object is copied without its connected objects, the connection information is not copied as well.

If the copy is connected with another object when it is placed, the connection information is updated for all involved.

Child objects

If an object is copied below which there are further objects (= children), then these objects are also copied and pasted when pasting the originally copied object.

It makes no difference at all whether the child objects were created as elements or as free objects. It also makes no difference whether the children are reports, queries or other components.

Copying the owner

A number of objects require a particular owner structure, for example pipe objects (Pipe -> Pipe branch -> Pipe segment) and functions with function code (Position -> Function).

The following occurs upon copying such an object:

- The owner structure is copied.
- The object itself is copied.

During pasting:

- the copies of the owners are pasted first.
- Then the copy of the object itself is pasted below the required owner.

The remaining child objects that are located below the owner but were not selected for copying are not copied.

SortNewObjectsInCategories

If `SortNewObjectsInCategories` is activated in the options script of the report template, Comos tries to sort the objects below the owner of the diagram during pasting. If the diagram is located below one of the subunits preconfigured in the COMOS DB, this subunit will only allow categories as direct child objects. The copied objects are sorted into the appropriate category, regardless of where the original was located.

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "SortNewObjectsInCategories (Boolean)".

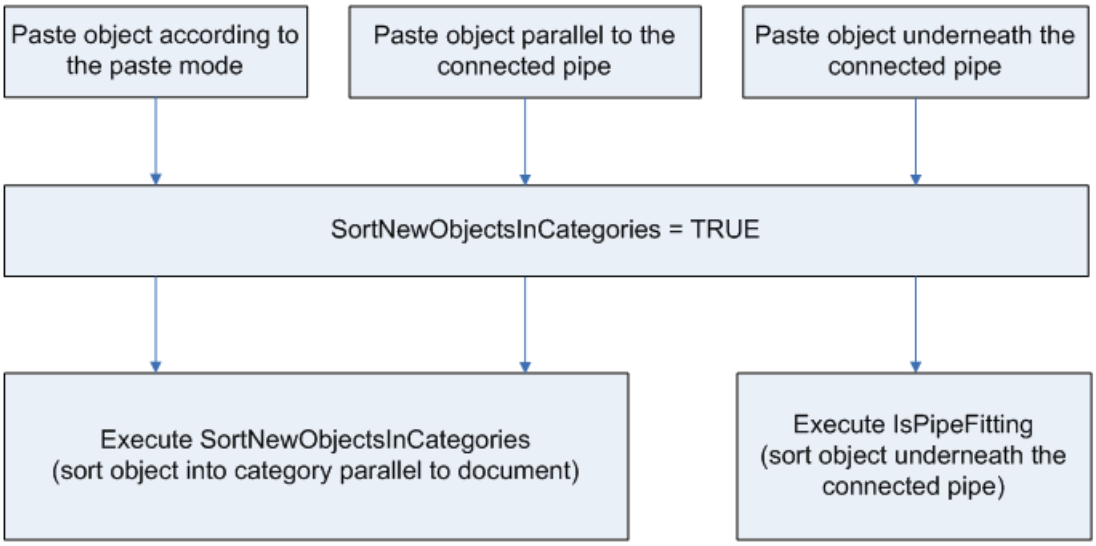
The algorithm mapped in the overview diagram is modified as follows:

- If the paste mode were to take effect:

  `SortNewObjectsInCategories` is executed and overwrites the paste mode.
- If the copy was to be pasted parallel to the connected pipe:

  `SortNewObjectsInCategories` is executed.
- If the copy was to be pasted below the connected pipe:

  `IsPipeFitting` overwrites `SortNewObjectsInCategories`. The copy is pasted below the connected pipe:



Exceptions

Paste mode: "Parallel to document" and "At same position":

If the copied component is placed on an existing pipe when it is pasted and if `IsPipeFitting` is not equal to 1 or is missing:

The copy is created according to the paste mode. The pipe is separated according to the pipe cut mode. New pipe segments and pipe branches are created below the pipe, and new pipes are created parallel to the existing pipe.

9.8.4 Pipe segments and pipe branches

Copying with owner

When you copy a pipe segment or a pipe branch on the P&ID and then paste it using the corresponding command from the context menu, the pipe or the pipe and the pipe branch respectively are copied as well. See also section Basic algorithm (Page 104).

Using paste to cut an existing pipe

If a device is placed on an existing pipe when it is pasted, the component separates the pipe according to the pipe cut mode of the component – either into new pipe segments, into new pipe branches or into a new pipe with a pipe branch:

- Pipe segments: The new pipe segment is created parallel to the existing pipe segment. If no pipe segment exists yet, it is created below the pipe branch.
- Pipe pipe branches: The new pipe branch is created parallel to the existing pipe branch.
- Pipe: The new pipe is created parallel to the existing pipe.

In object queries, the option "From the clipboard" can be selected for the start object. In this case the pipe would also be included in the start quantity.

9.8.5 Functions / positions

The exact paste behavior of functions depends on whether all or only a subset of the functions that are located underneath a position (class "Position", subclass "None") are copied.

Copy all

All functions on a P&ID diagram that are located underneath a position are copied and pasted to the diagram again.

Result: The functions that have just been created are created in the Navigator under a new position.

Position-based copying: CopyThroughPosition

When copying, the complete hierarchy structure of the component to be copied is created up to the position, and only if the attribute of the component contains "Sys.CopyThroughPosition" and it has been set to "True". If there are multiple components in the copy set, they are sorted into the new position hierarchy based on the same order as in the old structure.

Without "CopyThroughPosition" or "CopyThroughPosition" = False: The hierarchy is flattened and all components are copied with the standard procedure.

Copy subset

Result: In the Navigator, the copies are pasted underneath the same position underneath which the originals are located.

Exception:

A paste mode is selected which is to paste the subset at another location of the tree structure than the one at which the position is located that owns the original. A new position is created.

Example:

`SortNewObjectsInCategories` is deactivated. Underneath a P&ID diagram there is a position for pressure measurements, underneath which there are three functions. One of these functions is copied and pasted using "Paste > Parallel to document".

Result: A new position for pressure measurements is created parallel to the document and the function is pasted below it.

9.8.6 Examples for copying and pasting

Example 1

Initial situation:

A part unit with P&ID diagram and categories. A pump and its connected pipe are located underneath the main unit. They have been placed on the P&ID:

Action:

The pump and one of its connected pipe branches are copied and pasted using "Paste > Below document". The copies are not connected to another pipe when they are pasted.

Result 1: `SortNewObjectsInCategories = FALSE`

Since the copied objects were not connected to a pipe when they were pasted, they are pasted in the Navigator according to the paste mode – i.e. in the Navigator under the P&ID.

Result 1: `SortNewObjectsInCategories = TRUE`

`SortNewObjectsInCategories` overwrites the paste mode – the pasted objects are sorted in the Navigator into the categories folder located parallel to the P&ID.

Example 2

Initial situation:

A part unit with P&ID diagram and categories. A valve that does not have a fitting and has not been connected is located underneath the main unit. A pipe with a pipe branch is located underneath the main unit. All objects have been placed on the P&ID.

Action:

The valve is copied and pasted using "Paste > Below document". It is placed on the pipe branch when it is pasted.

Result 1: `SortNewObjectsInCategories = FALSE`

Since the `Connect` script of the pipe branch is called and the valve is not a fitting, it is pasted in the Navigator parallel to the connected pipe.

Result 2: `SortNewObjectsInCategories = TRUE`

The valve is sorted into a category underneath the subunit. The pipe, which was not part of the paste set, remains underneath the main unit.

Example 3

Initial situation:

A part unit with P&ID diagram and categories. A valve that is a fitting and has not been connected is located underneath the main unit. A pipe with a pipe branch is located underneath the main unit. The objects have been placed on the P&ID:

Action:

The valve is copied and pasted using "Paste > Below document". It is placed on the pipe branch when it is pasted.

Result 1: `SortNewObjectsInCategories = TRUE` and

Result 1: `SortNewObjectsInCategories = FALSE:`

Since the Connect script is called and the valve is a fitting, it is always sorted in the Navigator underneath the connected pipe, regardless of whether or not `SortNewObjectsInCategories` has been activated.

Copying valves

When copying a component (a valve object, for example) on a P&ID and then selecting "Paste underneath document", a pipe object (that has not been placed) is not created underneath the document under which the new valve is created.

Fitting: Assignment to pipe branches

If the component has been declared as RIFitting and has already been connected with a pipe branch and has also been placed correctly in accordance with the RIFitting declaration, no moves are made to the new pipe when a new pipe is docked to this component.

9.9 Cut and paste on P&IDs

9.9.1 Overview

Usage

Select one or more objects on the P&ID, call "Cut" and "Paste" from the context menu, and select a paste mode. Right-click once to determine where the symbol is to be placed on the diagram.

Paste location and paste behavior

Exactly where the objects are pasted in the Navigator and which objects are pasted depends on a number of factors. See also section Basic algorithm (Page 112).

9.9.2 The paste modes

The same modes are available as for copying. They function in the same way.

- "Below document":

  The cut objects are pasted into the Navigator underneath the document.

- "Parallel to document":

  The cut objects are pasted into the Navigator parallel to the document.

- "At same position":

  The cut objects are pasted into the Navigator at the same place. In other words, nothing actually changes in the Navigator. Symbols can be moved in this way within the P&ID.

- "Keep object":

  Is only activated if the "Paste" command is called on a different diagram than the one on which the original is placed. Only the symbol is copied, not the engineering object. The engineering object is then placed on two diagrams afterwards (this is the same effect as if you called "Copy" and "Paste > Keep object").

- "Automatically":

  Drawing type-specific behavior that functions analogous to that of copying, only that no copy is produced, and instead the object is moved within the Navigator. See also section The paste modes (Page 102).

The objects that were cut are always pasted according to the paste mode, if:

- They are not connected to a pipe when pasted
- They are connected to a pipe when pasted, but the pipe does not call the `SetOwnerbyDeviceConnector` function in the `Connect` script

See also

Basic algorithm (Page 112)

9.9.3 Basic algorithm

Interaction of the factors relevant for pasting

The exact paste behavior and paste location is determined by the same factors as when copying and pasting. See also section Basic algorithm (Page 104).

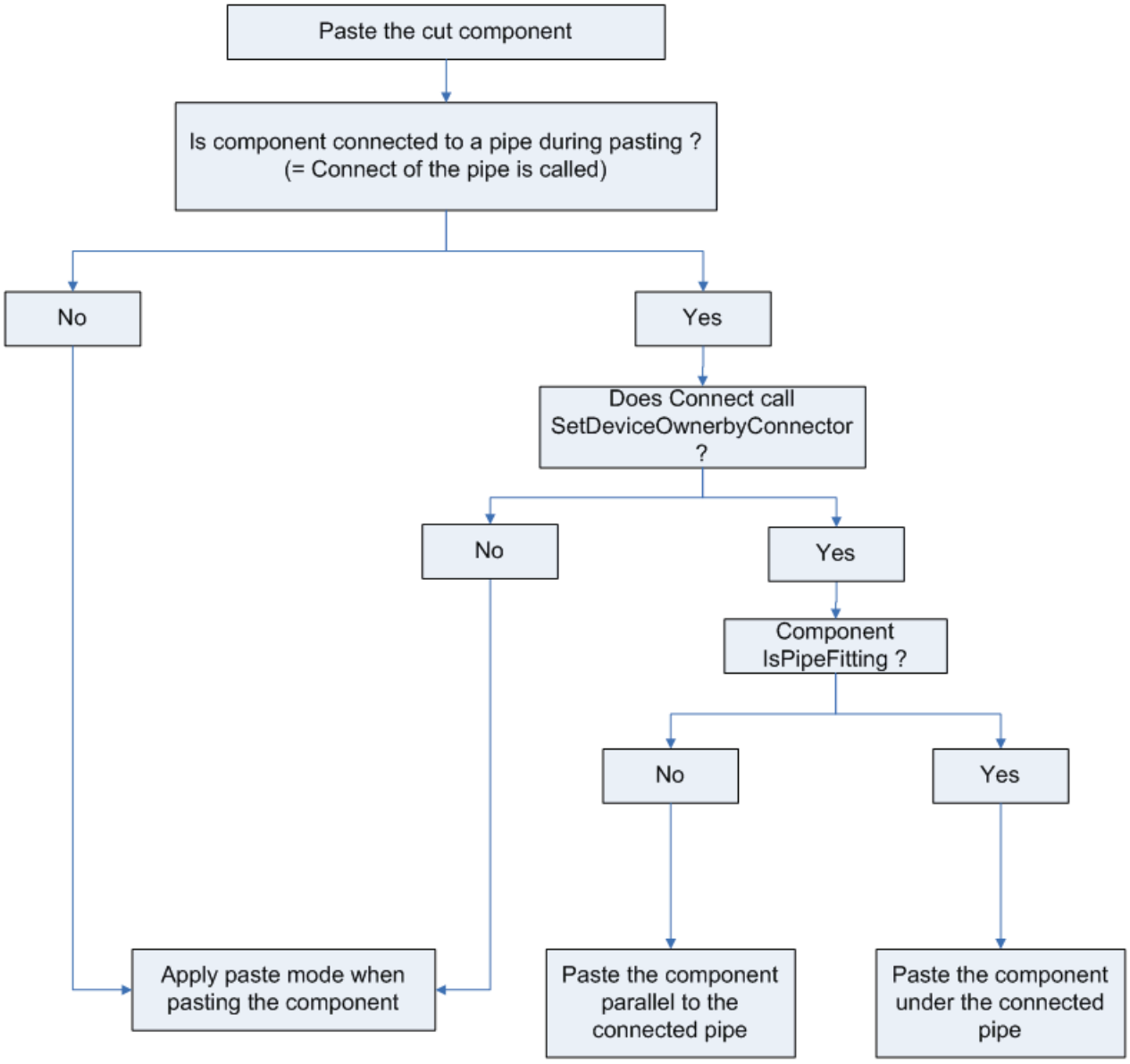
The diagram below gives you an overview of the interaction between these factors. The exact behavior of pipes and functions during cutting and pasting is not part of this overview. Information regarding this can be found in separate sections:

- In regard to pipes: see also section Pipe segments and pipe branches (Page 115).

  If an object is placed on a pipe when pasted, the pipe is separated according to the pipe cut mode of the object.

- In regard to functions, see also section Functions and positions (Page 116).

Overview: Algorithm for cutting and pasting



If the component is to be pasted parallel to the connected pipe and the pipe is located in a category, a search is made for a category located in parallel into which the component may be sorted.

Attributes

The attribute values are retained.

Name, label, and description

Analogous to copying and pasting. See also section Basic algorithm (Page 104).

Connection information

If an existing connection is deleted or if a new connection is made as a result of cutting or pasting, the connection information is updated on all involved connectors.

Example:

A valve is cut from a pipe and pasted onto another pipe. The following are updated: The connectors of the pipe segments that were connected to the valve before cutting, the connectors of the valve, and the connectors of the pipe segments that are connected to the valve after pasting.

Child objects

Analogous to copying and pasting. See also section Basic algorithm (Page 104).

Copying the owner

A number of objects require a particular owner structure, for example pipe objects (Pipe -> Pipe branch -> Pipe segment) and functions with function code (Position -> Function).

The following applies if such an object is cut and pasted with a paste mode that will paste the object at a location in the Navigator other than its original one:

When cutting:

- The owner structure is copied.
- The object itself is cut.

During pasting:

- the copies of the owners are pasted first.
- Then the object itself is pasted underneath this owner.

The remaining child objects that are located underneath the owner but that were not selected when cutting, and also the original owner structure, are not affected by cutting and pasting.

SortNewObjectsInCategories

Analogous to copying and pasting. See also section Basic algorithm (Page 104).

Sole deviation:

If "At same position" is selected, the objects are pasted at the same position in the Navigator where they had been before. In other words, they are not moved in the Navigator. They are not sorted into the categories.

Exceptions

Currently, there are the following exceptions to the algorithm described above:

- A pump (pipe cut mode = pipe separative) is cut and pasted onto a pipe using "Paste > Parallel to document". The Connect script of the pipe calls the `SetDeviceOwnerbyConnector` function.

  The pump is pasted according to the paste mode, not parallel to the connected pipe.

9.9.4 Pipe segments and pipe branches

An existing pipe is cut by pasting

The following applies, analogous to copying and pasting:

If a device is placed on an existing pipe when it is pasted, the component separates the pipe according to the pipe cut mode of the component – either into new pipe segments, into new pipe branches or into a new pipe with a pipe branch.

The new pipe objects generated as a result are always created parallel to the existing pipe object, regardless of whether or not the `SetDeviceOwnerbyConnector` function is called in the `Connect` script block.

Copying the owner structure

The following applies, if:

- A pipe segment or a pipe branch is cut on the P&ID and the paste mode takes effect when pasting
- The paste mode has, in addition, been selected in such a way that the pipe is pasted in the Navigator somewhere other than its original position

Result: The pipe/pipe branch below which the pipe branch/pipe segment was located is copied as well and then pasted at the location determined by the paste mode. See also section Basic algorithm (Page 104).

Example:

A pipe that has two pipe branches is located underneath a subunit. One pipe branch is cut and then pasted back in using "Paste > Below document".

Result: The pipe is copied and the copy is pasted below the document. The selected pipe branch is cut and then pasted underneath the pipe that has just been created.

Deleting superfluous pipe branches or pipe segments

A consequence of cutting a device can be that a pipe segment, a pipe branch or even an entire pipe becomes superfluous:

- Pipe segment:

  Two pipe segments are merged when the one pipe segment is automatically deleted by the system.

  Example: A valve is positioned on a pipe and is cut.

- Pipe branch:

  Two pipe branches are merged. The user is prompted in a dialog to specify which pipe branch is to be retained.

  It is not possible to combine both pipe branches. They remain interconnected in the database, but are shown on the diagram as visually interrupted.

  Example: A reducer is positioned on a pipe and is cut.

- Pipes:

  Two pipes are merged with one another. In the same way as with the pipe branch, the user decides which pipe is to be retained.

  Example: A pump is connected with two pipes and is cut.

9.9.5 Functions and positions

The cutting and pasting of functions works in the same way as the copying and pasting of functions: Again, it is necessary to distinguish whether all functions that are located underneath a position (class "Position", subclass None) or only a subset of them are affected by the action.

- Cut all:

Result: A new position is created in the Navigator in exactly the position determined by the paste mode. For example: Below document, Parallel to document. The functions are moved underneath this new position. There are no longer any functions underneath the old position, but it is not deleted automatically.

- Cut subset:
  - The paste location that is determined by the paste mode is identical to the position at which the functions are currently located in the Navigator:

    Result: The functions are not moved in the Navigator. No new position is created.
  - The paste location that is determined by the paste mode is not identical to the position at which the functions are currently located in the Navigator:

    Result: A copy of the position is created in the Navigator in the location determined by the paste mode. The functions that are cut are moved underneath this new position.

9.9.6 Examples: cutting and pasting

Example 1

Initial situation:

A part unit with P&ID diagram and categories. A pump and its connected pipes are located underneath the main unit. There is an additional pipe underneath the main unit. The objects have been placed on the P&ID:

Action:

The pump is cut and pasted using "Paste > Below document". It is placed on the second pipe when it is pasted.

Result 1: `SortNewObjectsInCategories = TRUE`

The user is prompted to decide which pipe is to be retained. The superfluous pipe is deleted, together with its pipe branch. The pump is sorted into a category that is located parallel to the P&ID. When it is placed on the second pipe, the pipe is separated into two pipes in accordance with the pipe cut mode (= "Pipe separative"). The new pipe and its pipe branch are created parallel to the old pipe, i.e. underneath the main unit.

Result 2: `SortNewObjectsInCategories = FALSE`

As result 1, except that the pump is not sorted into a category but is pasted parallel to its connected pipe underneath the main unit. Hence its position in the Navigator does not change.

Example 2

Initial situation:

A part unit with P&ID diagram and categories. Two pipes are located underneath the main unit. A valve is located underneath one pipe, pipe A. The valve is a fitting and connected to the pipe segments of pipe A. All objects have been placed on the P&ID.

Action:

The valve is cut from pipe A and pasted using "Paste > Below document". It is placed on the second pipe, pipe B, when it is pasted.

Result 1: `SortNewObjectsInCategories = TRUE` and

Result 2: `= FALSE:`

The second pipe segment below pipe A is deleted as a result of cutting the valve. Pipe B is segmented when the valve is pasted. Since the valve is connected with the pipe and is a fitting, it is always sorted parallel to the pipe segment that it is connected to, i.e. below the pipe, regardless of whether or not `SortNewObjectsInCategories` is activated.

Example 3

Initial situation:

A part unit with P&ID diagram and categories. A pipe, pipe A, is located underneath the main unit. A valve is located underneath the pipe. The valve is a fitting and is connected with the pipe segments of the pipe. All objects have been placed on the P&ID.

Action:

The valve and one of its connected pipe segments, pipe segment B, are cut and pasted using "Paste > Below document". The objects are not connected with another pipe when they are pasted.

Result 1: `SortNewObjectsInCategories = TRUE:`

The valve and pipe segment B are deleted from the structure below pipe A. Then they are sorted into the categories underneath the subunit. Since the pipe segment must be located below a correct pipe structure, the pipes and the pipe branch must have been copied beforehand and pasted as the owner of pipe segment B into the category for pipes. Although the valve is a fitting, it is not pasted underneath the pipe but instead into the category for valves. Reason: The valve is already connected, and the `Connect` script is not executed when pasting.

Result 2: `SortNewObjectsInCategories = FALSE:`

The valve and pipe segment B are deleted from the structure below pipe A. In the same way as above, the pipe and the pipe branch are also copied; they are pasted as the owner of pipe segment B according to the paste mode, i.e. below the document. Although the valve is a fitting, it is not pasted underneath the pipe but instead likewise directly underneath the document. Reason: The valve and the pipe segment are already connected. No `Connect` is performed during pasting.

9.10 Opening process units / further level

The individual processes can be run once the basic flow sequences have been described in the form of process units and substance streams.

Open a process unit by double-clicking on it.

The result is that you are get a new diagram on which only the selected process unit can be seen. The incoming and outgoing parts are displayed by bold arrows:

The incoming and outgoing parts are displayed. For that reason the superordinate substance stream should already have been created.

All elements can be positioned within the process unit, these being process units, substance streams, instrumentations.

Once you have completed all inputs, close the window by clicking on the "Close" button and confirm the prompt to save with "Yes".

Whether or not the objects on the inside of a process unit are displayed at the higher level depends on the object used or the setting. However, the objects on the inside of a process unit can only be changed when the process unit is opened by double-clicking on it.

9.11 Hierarchical structures

If you drag base objects onto a P&ID diagram, engineering objects are created underneath the diagram as a result.

Exception: The `SortNewObjectsInCategories` function is activated in the script options of the report template.This function tries to directly sort the objects into categories upon being created. See also section Report template script options (Page 219).

However, the engineering objects cannot be left underneath the report permanently and have to be sorted into a meaningful structure.

9.11.1 The "Assign object" tool

9.11.1.1 Object assignment

Overview

You can use the "Assign object" tool to change objects placed on the report in the database.

- "Automatic" mode

  The mode assigns a new owner to an object in the engineering data and the object is moved in the Navigator so that it is located underneath a different object.

  When you place a base object onto a type "P&ID" interactive report, the corresponding engineering object is created in the Navigator underneath the report. You can use the "Assign object" tool to move these objects to the required unit subsequently.

- "Change object" mode

  The mode replaces an object placed on the report with another object that has not yet been placed on the report.

- "Set owner" mode

  The mode assigns a new base object to an object.

Changing object properties

Using the "Assign object" tool significantly changes some of the properties of the object being edited.

9.11.1.2 Operation

Procedure

To work with the "Assign object" tool, proceed as follows:

1. Select one or more objects on the report.
2. Select the "Assign object" tool.

3. Select the required mode from the report toolbar.
4. Drag&drop an object from the Navigator onto the selected objects.

   You can only ever drag one object from the Navigator. If you select multiple objects in the Navigator, the tool is blocked.

Result

The assignment is made corresponding to the selected mode. COMOS automatically checks whether the action you have selected is permitted for the selected objects. If the action is not permitted, it is aborted and you are told why.

Deactivating the "Assign object" tool

If you no longer wish to work with the "Assign object" tool, right-click on the report.

9.11.1.3 Assignment mode "Automatically"

Overview

If you have activated "Automatic" mode, COMOS automatically analyzes which objects you have selected on the report and which object you have dragged from the Navigator onto the report. COMOS then adapts the assignment function accordingly.

The result of this mode depends on the following factors:

- Single or multiple selection in the report
- Selection of an engineering object or a base object in the Navigator
- Whether a selected engineering object can be displayed on the report A P&ID symbol must have been predefined at the base object of the object on the "Symbols" tab.

Multiple selection

If you have selected multiple objects on the report, the "Define owner" assignment function is always activated.

All objects you have selected on the report and for which an assignment is possible are moved in the database so that they are located below the engineering object which you moved to the report from the Navigator using drag&drop. In the report, a text is displayed on the symbol identifying the owner.

Objects to which a new owner cannot be assigned include the base objects from additional graphics. These objects are not moved.

Single selection

- "Change object" mode

  If you drag&drop an engineering object from the Navigator onto the report which can be displayed on the report, the "Change object" assignment function is activated. The base object of the object selected in the Navigator must have the same class as the engineering object already placed on the report.

  The object placed on the report is replaced by the object dragged from the Navigator. The original object is no longer placed on the report. If the original object was located directly below the report, it is deleted automatically.

- "Set owner" mode

  If you drag&drop an engineering object from the Navigator onto the report that does not have a symbol that can be displayed on the P&ID report, or one whose base object does not have the same class as the original object, the "Define owner" assignment function is activated.

  The object placed on the report is replaced by the object dragged from the Navigator. The original object is no longer placed on the report. If the original object was located directly below the report, it is deleted automatically.

  You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Define owner".

  See also section "Set owner" mode (Page 121).

- "Change base object" mode

  If you drag&drop a base object from the Navigator onto the report which can be displayed on the report, the "Change base object" assignment function is activated.

  A new base object is input in the properties of the engineering object on the "General" tab. The engineering object inherits the attributes from its new base object. An exception to this rule are all attributes that had been input at the engineering object and had not been transferred from the base object. If the new base object has the same attributes, these values that had been set at the engineering object are retained. The symbol of the new base object is displayed in the report.

  This assignment function is only available in "Automatic" mode.

Example

A 2-way valve can be replaced by a 3-way valve since both have the class "Device". Similarly, a vessel can be replaced by a pump, since both have the class "Position".

A function cannot be replaced by a valve, since they have different classes.

9.11.1.4 "Set owner" mode

The "Define owner" mode assigns a new owner to the objects that have been selected on the diagram. The objects are moved below the new owner in the Navigator.

Procedure for "Define owner"

1. Mark the objects on the P&ID that you want to assign to a new owner.
2. Select the "Assign object" tool, "Define owner" mode.
3. Use drag&drop to move the new owner from the engineering data to the selected objects. This will usually be a unit.

   If you select the subunit predefined in the COMOS DB, the objects are automatically sorted into the categories located underneath the subunit.

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Define owner".

Changing owners

The "Change owner" function (ChangeOwner) is available on P&ID reports when using the "Assign object" tool. This function has been extended:

The Device.Label is now also checked on ChangeOwner. A new label is generated if the label at the target owner is already in use for another child object.

Fitting

In the case of components with the "fitting (IsFitting)" property, object assignment does not trigger sorting into categories. Instead, these objects remain below the pipes, pipe segments, or pipe branches determined by the cut mode.

Technical background: In current databases, a component is recognized as a fitting by the "IsPipeFitting" attribute being set to "True".

9.11.1.5 Special modes

Object assignment also supports a number of special modes in addition to "Automatic" mode:

- You can also activate "Define owner" and "Change object" modes directly without using "Automatic" mode. The function of the modes remains the same.
- The special modes "Assign unit", "Assign location", or "Process stream" assign the objects selected on the report with a reference to the object you moved from the Navigator using drag&drop. Select the object in the Navigator according to the activated mode.

  To use these modes, the corresponding report template script must be enabled. See also section Document base objects and their properties (Page 217).

9.11.2 Categories

With the help of the category method components are not assigned globally to a unit but instead can be sorted below the unit separated by their class.

See also section Basic algorithm (Page 104).

Categories have been prepared underneath subunit in the COMOS DB.

9.12 Search strings and cross-references

Search strings can be created and executed for pipes and pipes segments that display engineering objects. Select "Settings > Search text" from the context menu to go to the appropriate mask.

Measuring functions and data lines (multiple connections without objects) can be connected via cross-references: Create an open connection on both sides, then select the first connection and call "Connection > Memorize" in the context menu; the other connection is assigned using "Connection > Connect with...". These connections can be removed in the usual way using "Disconnect ...".

9.13 Extending the graphical settings

The "Graphical settings" context menu can be extended by the user for a component placed on a P&ID diagram if one of the following script functions has been set at the base object on the "Script" tab, `UserScriptBlock1`:

- `AddToGraficalParameterRI1`:

  Extends the context menu for drawing type "PID1"

- `AddToGraficalParameterRI2`:

  Extends the context menu for drawing type "PID2"

- `AddToGraficalParameterRI`:

  Extends the context menu for drawing types whose name starts with "PID".

Is only evaluated if `AddToGraficalParameterRI1` and `AddToGraficalParameterRI2` are not found.

These variables define attributes in which standard tables have been assigned. These standard tables store additional symbols. This means that you can use the context menu to show or hide additional symbols on the report that define the properties of the object more precisely.

Example: For a valve:

```
Dim AddToGraficalParameterRI (4)
AddToGraficalParameterRI (0) = "PI040.PIA045"  'Drive
AddToGraficalParameterRI (1) = PI040.PIA043"   'Continuos
                                               'control
```

This method can also be used with other drawing types and modules. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "AddToGraficalParamater<drawing type>(Num)".

9.14 Changing color settings for symbols globally

Requirement

You have placed objects on the report. No objects are selected.

The defaults for using the color settings have been made by the administrator.

Procedure

To change the color settings for symbols of placed objects globally, proceed as follows:

1. Right-click on the report.

   The context menu opens.

2. Select the "Select color set" command from the context menu, along with the required color.

Result

The symbols for all placed valves, for example, are displayed in the selected color, dependent upon the setting made by the administrator. This color setting for the symbols is temporary.

The selected command is highlighted in the context menu.

"Base objects" tab

The color settings are stored as a local base object on the "Base objects" tab of the current project in the Navigator under the "@System > @Profiles > <name of user currently logged in>" node.

You can thus save custom color settings for symbols in each project. The stored color settings are called when you open a report.

See also

Updating color settings for symbols (Page 125)

Changing color settings globally (Page 228)

9.15 Updating color settings for symbols

Procedure

To update the color settings for the symbols of placed objects on the report, select one of the following options:

- Select the "Refresh using color set" command from the report context menu.
- If you have changed the color settings for the symbols in another report, select the "Refresh using color set" command in the context menu of the report currently open.
- Close and re-open the report.

Result

The color settings for the symbols are updated.

9.16 Editable text on the report

9.16.1 Placing and editing editable texts on the report

Requirement

- Attributes are displayed in the Navigator.
- The attribute is a text field and belongs to a pipe.
- The editing of texts has not been switched off.

  See also section Locking editable text against editing (Page 126).

Procedure

To place and edit editable texts on a report, proceed as follows:

1. Expand the structure in the Navigator until you can see the individual attributes of the pipe's tabs.
2. Use drag&drop to move the required attribute from the Navigator to the report.

   The text field is placed on the report. If you have already made an entry in this field, the content is displayed on the report.
3. To edit the content of the text field, select the placed text field.

   You cannot edit merged texts.
4. Click the selected text field.

   The text field enlarges and is surrounded by a green border.

5. Enter the required text.
6. To finish editing the text, click next to the text field.

Result

The entry from the text field is written to the attribute on the corresponding tab. If a unit belongs to the text field, it can be placed on the report. The unit cannot be edited.

9.16.2 Locking editable text against editing

You can lock editable text using a script option in the report template of a report so that it cannot be edited. You can find additional information on this topic in the "Reports - Basic Operation" manual, keyword "EnableInteractiveEditableTexts".

9.17 Changing the appearance of the editable text in the report

Requirement

- An interactive report has been created and opened.
- An attribute has been inserted in a report.

Procedure

To change the appearance of editable text in the report, follow these steps:

1. Select the desired attribute in the report.
2. Select the command "Options > Graphical settings" from the context menu of the attribute.

   The "Attribute settings" window opens.
3. In "Configuration" tab, specify the description to be displayed in addition to the attribute value.
4. Specify how the text should appear in the "Display mode" tab.
5. Save the settings with "OK".

9.18 Pipe monitor

Overview

This component improves the overview of pipes. In practical applications, the pipe monitor is located between the structural object view of a pipe in the Navigator and the graphical display on a P&ID.

Opening

Report, right-click on a line: "Options > Pipe monitor"

The selected pipe structure and all components are displayed in a separate customized additional Navigator.

Interaction between pipe monitor and report

- If you use <Ctrl> and left-click to select a component in the pipe monitor, this subset, the selected object, and the components located below it in the hierarchy are highlighted on the report.
- Navigation between pipe monitor and report.

9.19 Miscellaneous

Multiple placement of components

Whether or not a component can be dragged multiple times from the Navigator onto the same diagram or a diagram of the same type is determined in the properties of the component:

"SYS System" tab, "MultipleUse Permit multiplacement" option.

Activated:

The object can be placed multiple times on the same diagram or on a diagram of the same drawing type. The user no longer has to confirm the placement explicitly. The engineering object can either be dragged from the Navigator onto the diagram (drag-and-drop) or its symbol is selected on the diagram and copied using the "Copy" and "Paste > Keep object" commands from the context menu.

Deactivated:

The object may only be placed once on the same diagram or on a diagram of the same drawing type. The user must confirm the multiple placement explicitly. Default.

Grouping on the report

If an element is included in the selection, then the selection is automatically extended to cover the group that the element belongs to.

Example:

A pump with a motor as element is placed on the P&ID. After that, a text is set next to the motor and grouped only with the motor.

If the user now selects the pump, as well as the element of the pump (i.e. the motor), the text is also selected, since it is included with this motor in a group.

The motor must have a `DocObj` of its own in order to be able to select the motor without the pump. First of all, the motor is created on the diagram as a subsymbol, i.e. it shares the `DocObj` with the pump. If you once again drag the motor manually from the Navigator onto the diagram, the motor is gets its own `DocObj`.

Mirroring symbols on the report

The symbol of a component placed on a P&ID diagram can be mirrored using the commands from the context menu: Select the symbol, call "Options > ..." in the context menu:

- Symbols with orthogonal symbol alignment:

  "...> Mirror symbol vertically" and "...> Mirror symbol horizontally"

- Oblique symbols:

  "...> Mirror symbol at main axis" and "...> Mirror symbol at minor axis"

If a symbol is composed of other symbols, COMOS automatically mirrors the individual symbols.

Rotating symbols

P&ID basic symbols can only be rotated in 90° increments. You can also set any other angle by using the "Transform" tool, however. The texts for this symbol can be rotated in steps of 45°.

If you rotate a P&ID symbol with the "Transform" tool, the automatic rotation of COMOS symbol is overridden. See also section Rotating a P&ID symbol when inserting it in the report (Page 24).

If a P&ID basic symbol has left the 90° axes, a movable text can be rotated freely with the grab points. In particular, the text can be rotated back to the horizontal.

Example:

- Container symbol is rotated through 35°
- Text with name is rotated through 35°
- Text can now be rotated back to the horizontal with the grab point

Once the P&ID basic symbol has completed a full 90° step again, it automatically returns to the basic method.

Lists and data sheets 10

10.1 Creating data sheets

Overview

Data sheets contain technical data about an engineering object. COMOS reads this data from the properties of the engineering object and displays it in the data sheet.

By default, data sheets are created automatically with the object.

Dependent upon the administrator setting in the base object properties, in some cases data sheets are not created automatically. In these cases you use the context menu of the required object to create the data sheet. The data sheet is created in the Navigator below the P&ID object.

You can only create each data sheet once. If the data sheet already exists, it is no longer offered for selection in the context menu.

Check the object properties to see if the data sheet is created automatically ("Mode" list on the "General" tab).

10.2 Editing data sheets

Introduction

Green fields can often be seen in the data sheets. They are interactive fields in which you can change the properties of the P&ID object.

Procedure

To change the properties of an object via the data sheet, proceed as follows:

1. Left-click in a green field.
2. Depending on whether the attribute in question is a text field, a dropdown list or a checkbox, you can input any desired value or select a specified value.
3. When you save the data sheet, the new value is saved in the database in the properties of the object.

   You cannot edit white fields.

Result

The properties of the object change accordingly.

10.3 Creating and using lists

Function of lists

Lists provide an overview of several engineering objects, for example of all P&ID objects of a particular type that are located below a unit. They contain important technical data on an engineering object. COMOS reads this data from the properties of the engineering objects and displays it in the list, where you can change it as required.

Creating a list

To create a list, proceed as follows:

1. Right-click on the plant you created under the project root. See also section The example project (Page 17).

   Various predefined lists are displayed in the context menu.

2. Select the required list.

Result

The list is created below the factory.

You can only create each list once. If the list already exists, it is no longer offered for selection in the context menu.

Editing lists

Just like data sheets, lists can also be edited via the green, interactive text fields. See also section Creating data sheets (Page 129).

Lists work in a context-related way. If you have created a pipe list for a unit, all pipes that are located below this unit are searched for and output in the list. It makes no difference where exactly the pipes are located in the object structure below the unit.

10.4 Controlling the apparatus bar/substance stream bar of the P&ID

Introduction

In "P&ID" type interactive reports, you can show an apparatus bar and a substance stream bar.

These two bars are nothing more than small lists that output the data of the apparatus and the substances being transported.

Example

Apparatus list

Name	W001
Description	Shell - Tube Exchanger,general
Technical data	m²
	m
Permitted operating pressure	bar bar
Permitted operating temperature:	100 °C °C
Material	

Showing a bar

Your administrator specifies which data is included in the list by means of the report template.

In the properties of the P&ID report, you can specify whether or not the lists are to be shown, and whether a comprehensive list or a short version is to be displayed.

To show the bars, proceed as follows:

1. Open the properties of the report on which you wish to show the bars.
2. Select the "Attributes > Substance stream bar - Apparatus" tab.
3. Make the required settings.
4. Click the "OK" button to save your settings and close the report properties.

Result

The next time you open the report whose properties you have edited, the bars are displayed.

Editing a list

These lists only read data. To edit the lists, proceed as follows:

1. Open the properties of the required object.
2. Enter the required values.
3. Click the "OK" button to save your changes and close the object properties.
4. Click the "Reevaluate document" button on the report's function bar.

Result

The list is updated.

Pipe Easy

11

Introduction

The Pipe Easy tool is used to process imported DGN, DWG, or DXF files.

You can use the tool to carry out the following actions:

- Convert lines into pipe branches and pipe segments using graphical properties
- Process texts in accordance with a script
- Convert graphical symbol drawings into COMOS symbols

11.1 Starting Pipe Easy

Requirement

The "Display PipeEasy conversion dialog" button is displayed on the report toolbar.

Procedure

To start Pipe Easy, click on the "Display PipeEasy conversion dialog" button on the report toolbar.

Result

The "Conversion settings" window opens.

See also

Adding the "Display PipeEasy conversion dialog" button to the toolbar (Page 247)

11.2 "Create pipes" tab

Overview

You can create pipes on the basis of symbols. When you do this, please take care with the grid settings. If the grid is set too large, the maximum distances will never be obtained and the conversion will not appear to function. In the German report templates, one grid point corresponds to one millimeter.

Use the "Execute" button to select and convert different lines or objects one after the other.

Context menu cannot be called

When you are working on the "Create pipes" tab, you cannot call the context menu or execute actions. To call the context menu, switch to a different tab or close the "Conversion settings" window. The settings in the "Conversion settings" window are retained.

Alternative options

If alternative options exist during the conversion process, the first object in the collection of the report objects is taken into account. Normally, this is the object that was created first.

Example

The end of a line lies on the start of two other lines. As no T-connections are created, COMOS must choose one of the lines.

See also

"Create pipes" tab (Page 282)

11.3 "Assign texts" tab

This tab evaluates the following information:

- A script (text rule)
- A selected text
- A selected symbol

How this information is evaluated and converted is relatively open.

Example

The fixed texts describe a unit label or a device label of the symbol, for example. A script is used to move the COMOS object of the symbol. The actual device label then corresponds to the fixed text. Next, delete the fixed text and replace it with a device label.

See also

Creating base objects for text rules (Page 248)

"Assign texts" tab (Page 285)

11.4 Creating rules

Introduction

You can use symbol and text or attributes to create rules.

Requirement

The "Assign texts" tab in the "Conversion settings" window is open. The database contains base objects for text rules.

Creating rules using symbol and text

To create a rule using symbol and text, proceed as follows:

1. Use drag&drop to move the action base object under which the base objects for text rules are located from the Navigator to the "Base object for text rules" field.
2. Enter the nested name of the attribute which will form the basis of the object search in the "Attribute for text rules" field.
3. Enter the tolerance within which a text will still be interpreted as belonging to a symbol in the "Maximal distance inaccuracy in mm" field.
4. Enter a layer which has not yet been used in the "Layer for converted texts" field.

   The layer entered automatically applies for the next action also.
5. Select "Create rule" from the "Action" list.
6. Select a symbol and a text on the report.
7. Select the required base object from the "Rule" list.
8. Click on the "Execute" button.

Result

This action measures the relations between the symbol and the text:

- Symbol

  SystemFullname of the base object, angle, mirroring
- Text

  Angle, relative distance to the symbol

The relation is saved in the base project on the "Base objects" tab under the "@System > @Profiles > @AllUsers" node.

Then, the corresponding script function is executed on the two marked objects.

Using attributes to create rules

To use an attribute to create a rule, proceed as follows:

1. Drag&drop the required attribute from the Navigator into the "Attribute for text rules" field.

   The nested name of the attribute is entered.
2. Enter the tolerance within which a text will still be interpreted as belonging to a symbol in the "Maximal distance inaccuracy in mm" field.
3. Enter a layer which has not yet been used in the "Layer for converted texts" field.

   The layer entered automatically applies for the next action also.
4. Select "Create rule" from the "Action" list.
5. Select a symbol on the report.
6. Select the required base object from the "Rule" list.
7. Click on the "Execute" button.

Result

This action measures the relation between the symbol and the attribute:

- Symbol: SystemFullname of the base object
- Attribute: NestedName

The relation is saved in the base project on the "Base objects" tab under the "@System > @Profiles > @AllUsers" node.

If the COMOS object has the specified attribute, it transfers the value of the attribute to the corresponding script function as text.

See also

"Assign texts" tab (Page 285)

11.5 "Create symbol" tab

Due to the symbols created previously by the DGN or DWG/DXF import, you do not need this tab.

If symbols in DGN or DWG/DXF files are not created as blocks, they are not detected as symbols when the file is imported and are created as lines on the report. To convert these graphical symbols into COMOS symbols, use the "Create symbol" tab.

See also

DGN import (Page 160)

Control elements on the "Create symbol" tab (Page 287)

Importing AutoCAD data (Page 83)

11.6 Creating symbols

Requirement

You have completed a DGN import. The "Create symbol" tab in the "Conversion settings" window is open.

Procedure

To create a symbol, proceed as follows:

1. Drag&drop the required base object from the Navigator into the "Object for symbol" field.
2. Select the required option.
3. On the report, select the graphical objects from which a symbol is to be created.
4. Click on the "Execute" button.

Result

A report object is created. The point of origin of the object is located in the top left-hand corner of the selected graphical objects.

An engineering object with the selected base object is created in the Navigator under the report, underneath the "__IMPORT" object. The report object is assigned to this engineering object.

The selected graphical elements are deleted.

See also

Control elements on the "Create symbol" tab (Page 287)

11.7 "Assignment" tab

You use the "Assignment" tab to link imported base objects to existing COMOS base objects. You can scale placed objects.

If you assign an imported engineering object to the COMOS base object, COMOS automatically uses the base object of the engineering object for the assignment.

See also

Control elements on the "Assignment" tab (Page 288)

11.8 Linking imported base objects to Comos base objects

Requirement

You have imported base objects. The "Assignment" tab in the "Conversion settings" window is open.

Procedure

To link an imported base object to a COMOS base object, proceed as follows:

1. Drag&drop the required imported base object from the Navigator into the "Imported base object" field.
2. Drag&drop the required COMOS base object from the Navigator into the "COMOS base object" field.
3. Select the required options.
4. Enter the required scaling.
5. Click on the "Execute" button.

Result

The imported base object is linked to the COMOS base object.

See also

Control elements on the "Assignment" tab (Page 288)

Mirroring flow-direction-relevant fittings 12

12.1 Objectives and areas of application

General

As well as the existing option of changing the flow direction of pipes, you can change the flow direction of pipe fittings.

To do this, connect the components for which the flow direction is relevant directly or indirectly to the pipe through a device connect to the pipe. If you change the flow direction of a pipe segment, the flow direction for all other pipe segments of the associated pipe is also changed. See also section Special features (Page 142).

The action is carried out in the flow direction of the pipe until a corresponding break condition is found or a component that has no further flow-direction-relevant connectors. By default, connectors "I1" and "O1" apply as flow-direction-relevant, for both pipes and fittings. Dynamic connector points are not taken into consideration.

You can also start the action from the fittings. In this case all connectors of the component are regarded as flow direction relevant.

12.2 Components involved

COMOS only mirrors components which have an input or output. Whether a symbol is mirrored depends on whether the connectors of the symbol are defined accordingly.

See also

Break conditions (Page 139)

12.3 Break conditions

Conditions

The function tracks the pipe route until one of the following conditions is found:

- The Start connector is reached again
- The component has only one connector O
- The component has the attribute "SYS.BehaviorFlowDir = 1".
- The component has neither I1+O1 connectors nor a "SYS.BehaviorMultiFlowDir" attribute in which further flow-direction-relevant connectors are defined.

"SYS.BehaviorFlowDir" attribute

The "SYS.BehaviorFlowDir" attribute changes the behavior of the flow direction. The attribute has the standard table "@SYSTEM > @BEHAVIORFLOWDIR" with the following values:

"Name" column	"Description" column	"Value 1" column	Result
1	Break	1	Start or destination, break, symbol is not mirrored
2	Multi-way object	2	Symbol is not mirrored

COMOS also uses the "`SYS.BehaviorFlowDir = 0`" attribute to check if there are several flows for a components that have been defined with the "`Sys.BehaviorMultiFlowDir`" attribute. Of the flow direction changes, COMOS only mirrors the connectors affected by the flow direction changes.

12.4 Procedure

12.4.1 Call and result

Introduction

Call the "Change flow direction" function on the interactive report for a pipe, a pipe segment, or a pipe branch, or another connected element.

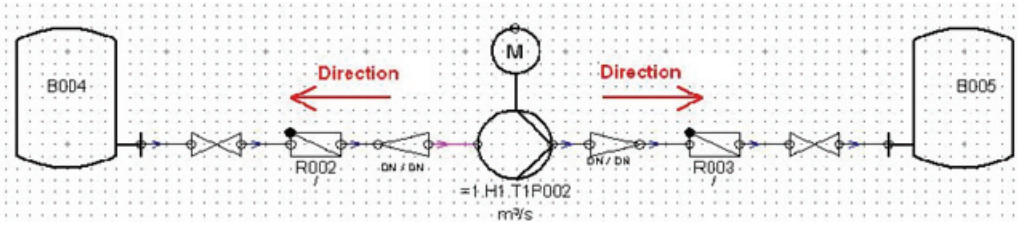
Procedure

To change the flow direction, proceed as follows:

1. Select the required object.
2. Right-click on the report.
3. Select "Options > Change flow direction" from the context menu.

   All report elements concerned are selected. You are prompted to confirm that you wish to change the flow direction of the selected objects.
4. Confirm the change of flow direction.

Example



In this example, the "Change flow direction" function is called up by the selected pipe. COMOS searches the pipe route for a break condition in both directions, starting from the input and output connectors of the selected pipe.

Behavior left side

In the example shown above, the left-hand side behaves as follows:

- Reducer: The reducer has two neutral connector points and is not mirrored.
- Pipe segment: The flow direction is changed for the pipe segment.
- Non-return valve: The non-return valve has a fixed input and output connector (I1&O1). It is mirrored with the connectors.
- Pipe segment: The flow direction is changed for the pipe segment.
- Valve: The valve has two neutral connectors and is not mirrored.
- Pipe segment: The flow direction is changed for the pipe segment.
- Nozzle: The nozzle only has one connector point and is not mirrored.

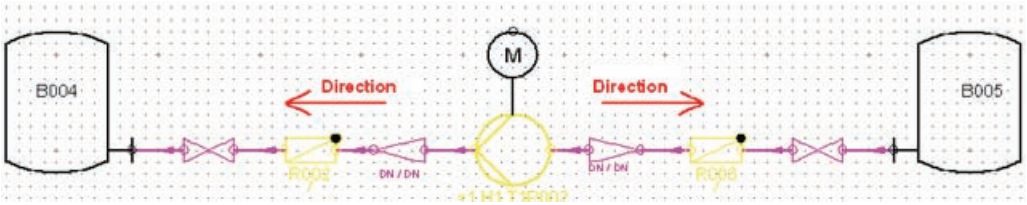
Behavior right side

In the example shown above, the right-hand side behaves as follows:

- Pump: The pump has fixed connector points and is mirrored.
- Pipe segment: The flow direction is changed for the pipe segment.
- Reducer: The reducer has two neutral connector points and is not mirrored.
- Pipe segment: The flow direction is changed for the pipe segment.
- Non-return valve: The non-return valve has a fixed input and output connector. It is mirrored with the connectors.
- Pipe segment: The flow direction is changed for the pipe segment.
- Valve: The valve has two neutral connectors and is not mirrored.
- Pipe segment: The flow direction is changed for the pipe segment.
- Nozzle: The nozzle only has one connector point and is not mirrored.

Result

The result of the action looks like this:



The pipe route identified is displayed in magenta from start to end. This enables you to see the total scope of the action. All components which have been mirrored are displayed in yellow. When you confirm the change of flow direction, the object color-coding disappears.

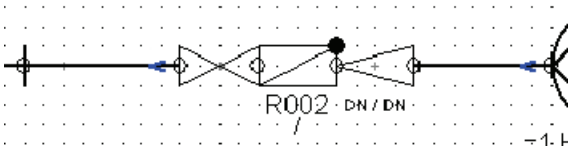
See also

Special features (Page 142)

12.4.2 Special features

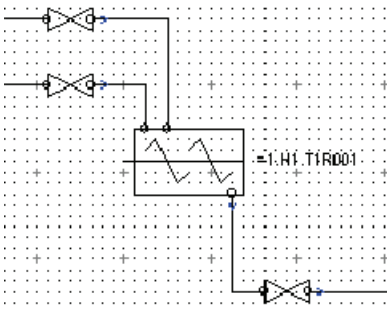
The following special features may be encountered:

Multiple to one another connected components



If the flow direction changes in this example, both pipe segments and the non-return valve are modified to match the new flow direction. The action is not canceled, although the valve is no longer connected to a pipe.

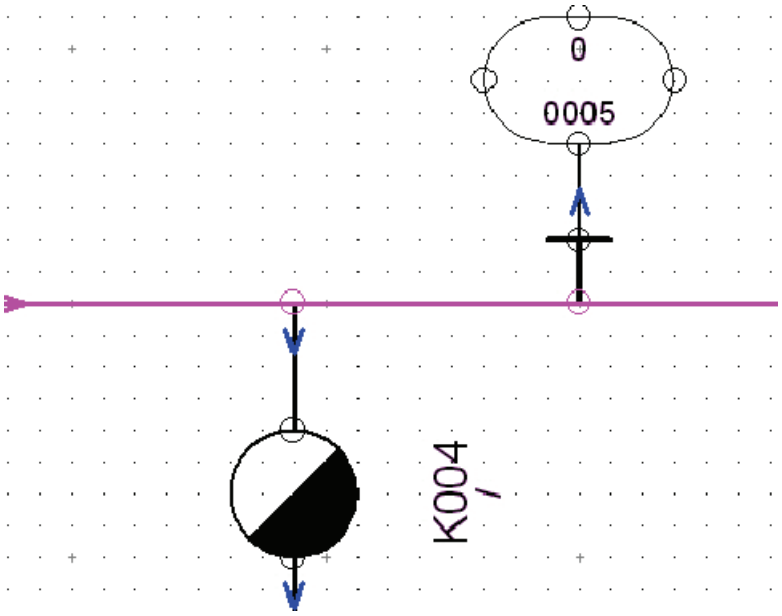
Components with multiple connection points



The object is usually defined as the break (SYS.BehaviorFlowDir=1). If this is not the case and there are no flow-direction-relevant connectors (I1/O1) or "SYS.BehaviorMultiFlowDir" is defined, a break occurs.

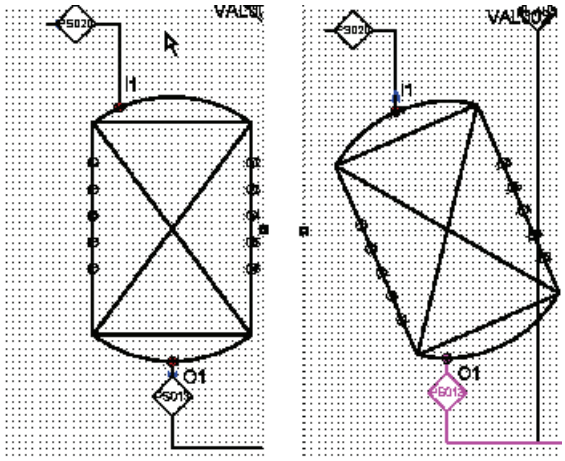
Dynamic connectors

Dynamic connectors of a pipe or a fitting are generally not tracked.



Vessel with output and input connectors, e.g. FEED

The image on the left shows the vessel before the change of flow direction. The image on right left shows the vessel after the change of flow direction.



If you assign connectors "I1" and "O1" to non-symmetrical points on the vessel, the change of flow direction results in asymmetric mirroring. In this case, the vessel is displayed diagonally on the drawing.

If you connect a stream to connector "I1", the stream of connector "O1" is copied.

If you use attribute "SYS.BehaviorFlowDir = 1" for the vessel, there is no mirroring and no copying in this status.

Same IO type

If you change all flow directions, both the input connector and the output connector have the same IO type.

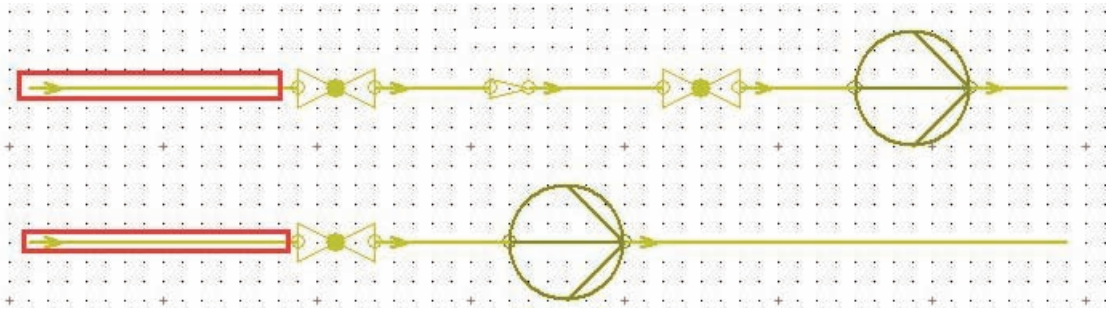
Flow direction for half-open connections

Connecting two half-open action lines with "CONNECTION > MEMORIZE" and "CONNECTION > SET" sets the flow direction on both sides. Any contradictory design directions that arise are synchronized and reset if necessary.

Changing the flow direction for an entire component string

A string of connected components and pipes is known as a component string. On account of the rule that all pipe segments of a pipe have to have the same flow direction, the following applies with regard to component strings:

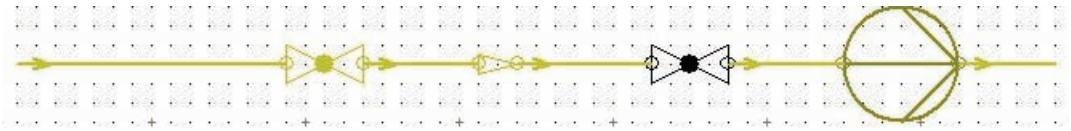
If you change the flow direction of a pipe segment in a component string, other pipe segments of the associated pipe which are in different components strings also change. Changing the flow direction for the pipe segment triggers the action to change the flow direction for the other component string. In the figure, the two pipe segments outlined in red belong to the same pipe branch.



Component with SYS.BehaviorFlowDir= 1

The pipe route stops on reaching a component with SYS.BehaviorFlowDir= 1. In the example, this is the valve:

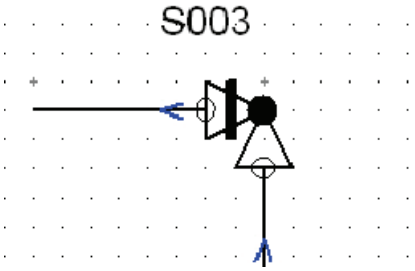
The rule does not apply if there is a pipe segment downstream of the component with SYS.BehaviorFlowDir= 1 whose pipe has at least one other pipe segment whose flow direction is changing. In other words, the rule that all pipe segments of a pipe must have the same flow direction takes priority over the attribute "SYS.BehaviorFlowDir". In this case, the break condition only applies for the relevant component. The action continues with the component downstream of it:



12.5 Handling right-angle valves

Overview

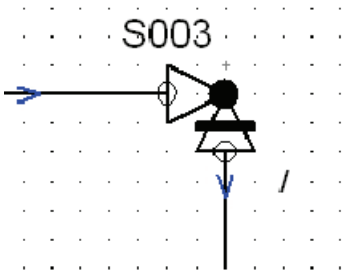
Some right-angle valves are not relevant for changes in flow direction. They include the safety right-angle valve, for example, whose bar always points in the output direction.



The mirroring axis is calculated as half the angle between the connectors "I1" and "O1" for components such as right-angle valves.

Result

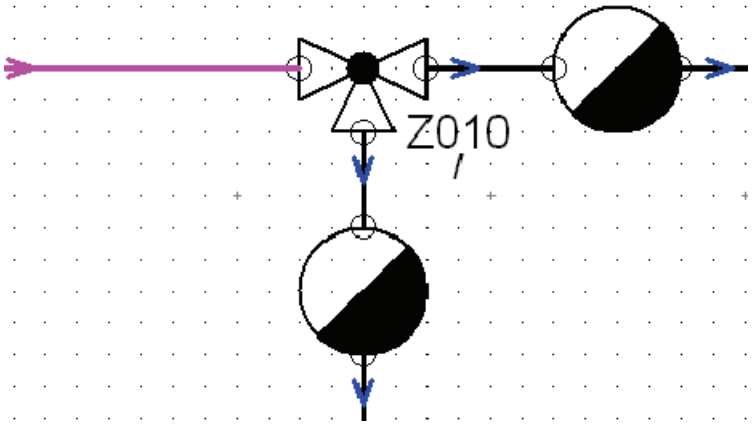
The figure below illustrates the result of the action:



12.6 Handling multi-way objects

Multi-way objects

Multi-way objects are objects such as three-way valves or tees, for example, which have more than two connectors but are usually viewed as standard pipe fittings.



"SYS.BehaviorMultiFlowDir" attribute

The "SYS.BehaviorMultiFlowDir" attribute can be set at the device so that multiple flow direction dependent connectors can be run through.

If, for example, the main direction (the one shown horizontally in the figure) is represented by the connectors "I1" and "O1", the first flow direction is defined. If the definition "I1>O2" is specified at the component for the "BehaviorMultiFlowDir" attribute, the second flow direction is defined. To define more connectors, add them with ";" (e.g. BehaviorMultiFlowDir = "I1>O2;I3>O4;I5>O5").

In this case, the flow direction of both connected pipes and their components changes if the component is relevant to the flow direction. The component is not mirrored because it possesses neutral connector points.

Interfaces 13

13.1 Conval

The interfaces are addressed using the corresponding base objects:

"@03 > PID > 600 > 1 Conval"

The objects, documents, and scripts that are required for the Conval interface are prepared here.

See also

Base object "@03 > PID > 600 Interfaces" (Page 190)

13.2 PDS 2D import

13.2.1 Basic principles

13.2.1.1 Aim

Importing the PDS 2D data of a DGN drawing

The COMOS P&ID module implements an import interface to PDS 2D from Intergraph.

Importing allows you to import DGN drawings into COMOS, which were created with Bentley MicroStation and using PDS 2D. This gives you the chance to transfer projects from PDS 2D to COMOS and continue your engineering in the COMOS P&ID module without any interruption.

The PDS 2D import is performed once the import of a PDS 2D based DGN drawing is completed. It complements the graphical DGN import with a database import: COMOS imports the PDS 2D data from the DGN drawing and converts as many elements of the drawing as possible into P&ID objects.

13.2.1.2 Principle

Two-level architecture

The import is available for P&IDs, in which the user has imported a PDS 2D based DGN drawing. It is performed after the DGN import.

DGN import

The following happens with the previous DGN import:

- The cells in a DGN drawing are imported.
- The purely graphical elements of the DGN drawing are imported.

Importing a cell:

- DGN import objects are created for each cell: An import base object and an import engineering object
- The DGN import objects store the following cell data:
  - Name
  - Graphical information
  - If available: The raw data
  - References to PDS 2D datasets
- The import engineering objects are placed on the P&ID.
- If the import is configured accordingly, empty cells are also imported.
- The import engineering objects were imported to the same hierarchy levels. This means that the owner structure of the DGN cells is not retained.

Importing purely graphical elements:

- The purely graphical elements of the DGN drawing are imported as purely graphical elements and placed on the P&ID.

  Example: Pipes

PDS 2D import

The following happens during a PDS 2D import:

- The import engineering objects are converted into P&ID engineering objects through a base object change.
- The pipes imported as line are replaced by objects.
- Data from the PDS 2D database that are referenced on the DGN drawing are imported into COMOS. They are written to the attributes of the P&ID engineering objects.
- The owner structure of the DGN cells is restored.

  Example: If a vessel has nozzles on the DGN, the nozzles are created under the vessel in COMOS.
- Additional graphics, such as pipe flags and tag numbers become intelligent.

13.2.1.3 Requirements

The following requirements must be met before a user can perform the PDS import for a P&ID:

Installed software

- COMOS version: 9.0 or higher
- MicroStation version: J and XM
- PDS 2D version: 7

Database

- COMOS DB: The base objects and tabs required for the import were imported into the database.
- A complete DGN import has been performed for the P&ID. This means:
  - The DGN cell library was fully imported.
  - The DGN drawing was imported.

DGN import settings

The following import settings are required to import the DGN cell library and the DGN drawings, which precede the PDS 2D import:

- "Create XML log" option: Enabled

The following import settings are recommended:

- "Always import as local script" option: Enabled

  See also section Configuring the symbol display (Page 233).
- "Import empty cells" option: Enabled

  See also section Importing text cells as objects (Page 238).

13.2.1.4 Workflow

Introduction

You have to differentiate between the following scenarios during the workflow:

- The DGN import already used a DB choice object.
- The DGN import did not use a DB choice object.

With DB choice object

If the DGN import did already use a DB choice object, the workflow looks like the following:

User	Step	Details
Administrator	Step 1 Already completed	By the DGN import: Import of the DGN cell library: • Create a P&ID • Create the DB choice object • Link the P&ID and DB choice object with each other • Open the P&ID • Call the DGN import command • Specify the DGN import settings • Start the DGN import
User	Step 2 Already completed	By the DGN import: Import of the DGN drawing: • Create a P&ID • Link the P&ID to the DB choice object from step 1 • Import a PDS 2D based DGN drawing that uses the cell library from step 1 into the P&ID
Administrator	Step 3	Assign P&ID base objects to the import base objects
Administrator	Step 4	Add import scripts to the P&ID base objects
Administrator	Step 5	Configuring the DB choice object: • Unit mapping • Other import settings
User	Step 6	Perform the PDS 2D import for the P&ID from step 2: • Open the properties of the P&ID • Call the DGN import
User	Step 7	Optional: Performing the concluding work • Precision work on the P&ID • Sorting P&ID engineering objects under new owners

Without DB choice object

If the DGN import did not use a DB choice object, the workflow looks like the following:

User	Step	Details
Administrator	Step 1 Already completed	By the DGN import: Import of the DGN cell library • Create a P&ID • Open the P&ID • Call the DGN import command • Specify the DGN import settings • Start the DGN import
User	Step 2 Already completed	By the DGN import: Import of the DGN drawing: • Create a P&ID • Import a PDS 2D based DGN drawing that uses the cell library from step 1 into the P&ID
Administrator	Step 3	Create a DB choice object
Administrator	Step 4	Move the import base objects that were created in step 1 by the import of the cell library under the DB choice object
User	Step 4	Link the P&ID from step 2 to the DB choice object from step 1
Continue as described in "With DB choice object", step 3		

13.2.2 Functional scope

13.2.2.1 Overview

The following sections provide information regarding the functions of the PDS 2D import:

- Single import and multiple import (Page 152)
- Import of the cell data from the PDS 2D database (Page 152)
- Symbol display (Page 152)
- Handling of already existing P&ID engineering objects (Page 152)
- Owner structure (Page 153)
- Pipes (Page 154)
- Measuring lines and effect lines (Page 154)
- Functions and control valves (Page 154)
- Importing text cells (Page 155)
- Database-specific import settings (Page 155)
- Import logs (Page 156)

13.2.2.2 Single import and multiple import

Single import and multiple import

You select from the following options:

- You perform the import for a single P&ID.
- You perform the import for all P&IDs that use the same DB choice object.

13.2.2.3 Import of the cell data from the PDS 2D database

Assignments between PDS 2D and COMOS

The data that was saved to the cells of the imported DGN drawing in the PDS 2D database is imported into the attributes of the COMOS objects.

13.2.2.4 Symbol display

DGN symbol or P&ID symbol

Whether the symbol of the cell generated by the import of P&ID engineering objects or the P&ID base object are used, depends on the DGN import settings and the configuration of the base data.

13.2.2.5 Handling of already existing P&ID engineering objects

Introduction

Import engineering objects are converted to P&ID objects during the PDS 2D import.

One result of the conversion is that the name of the former import engineering object is replaced with a user-defined name.

If an engineering object with the same name already exists, the administrator's settings determine the import behavior.

User-defined import behavior

The administrator defines in the DGN base objects how the PDS 2D import behaves when an object with the same name is found during the conversion. This gives the administrator the chance to select a suitable setting for each DGN base object.

He selects from the following options:

1. A new object is created.

   Consequence: The name of the new object is incremented by a number.

2. The existing object is used.

   Consequence: The PDS 2D data linked to the cell is imported into the COMOS attributes of the existing object.

If the existing object is used, the administrator also defines the following points:

- Is the graphic of the existing object or the graphic of the imported cell used?
- Will there be a base object change?

  When there is a base object change, the mapping between the PDS 2D data and COMOS attributes is based on the new base object.

See also

Implementing the import script (Page 238)

13.2.2.6 Owner structure

Importing the owner structure

If a DGN cell has subcells, this owner structure is restored by the PDS 2D import.

Example:

- In the DGN drawing: A vessel has an agitator and several nozzles.
- DGN import: The DGN import objects for vessels, agitators and nozzles are located on the same hierarchical level.
- PDS 2D import: The agitator and the nozzles are moved under the vessel.

The administrator has the option to implement an import script in which the due to this reason moved objects are processed according to their new owner structure.

Functions and control valves under positions

The user has the option to save, in the PDS 2D database, to which position a function or control valve belongs.

COMOS uses this information and builds an appropriate owner structure during the import.

See also

Functions and control valves (Page 154)

Implementing the import script (Page 238)

13.2.2.7 Pipes

Creating a three-level pipe structure

The pipe route is only defined by lines in the DGN drawing.

A three-level pipe structure is created during the PDS 2D import, consisting of:

- Pipe
- Pipe branch
- Pipe segment

The lines of the DGN drawing are converted into pipe branches and pipe segments and are created under P&ID pipe objects.

13.2.2.8 Measuring lines and effect lines

Importing measuring lines and effect lines

Measuring lines and effect lines are available after the PDS 2D import as objectless connections, as it is customary for measuring lines and effect lines in COMOS:

- The graphical connections are drawn in on the P&ID.
- The connection information is stored at the connectors of the connected P&ID objects.
- No objects are created in the Navigator.

13.2.2.9 Functions and control valves

Generating positions

During the PDS 2D import of a function / control valve, the following default import behavior applies:

- COMOS reads from the PDS 2D database to which position ("loop") a function or control valve belongs.
- If there is no position with the same name under the P&ID, it is created. The function / control valve is imported under the position.
- If there is a position with the same name under the P&ID, the function / control valve is imported under this postion.

Special case: Several functions or control valves under a position in PDS 2D

There is the chance that several functions and control valves were assigned to a position in the PDS 2D database. In this case, it is not possible to determine which control valve belongs to which function.

To clarify this missing assignment, the administrator has the option to define a deviation from the standard import behavior and to import functions and control valves under separate positions.

13.2.2.10 Importing text cells

Background

If you use the default import settings for the DGN import, cells that only consist of text and have no graphical information are imported as pure graphics. This means that no objects are created. There is no connection in the database between the text and the component to which the text refers.

Example:

- Pipe flags
- Tag numbers

Importing text cells as objects

The user has the option to configure the DGN import and the PDS 2D import in such a way that pure text cells are imported as objects during the PDS 2D import. P&ID label symbols are created instead of a pure graphic.

This has the advantage that there is a connection in the database between the text and the component to which the text refers. If the component data changes, the content of the labeling symbol is updated automatically.

13.2.2.11 Database-specific import settings

Databases with different data models

The user has the option to import data from different PDS 2D databases into his COMOS project. The PDS 2D databases can use different data models.

The administrator can write a dedicated import script for each database.

Import settings

The administrator determines database-specific import settings for each PDS 2D database.

13.2.2.12 Import logs

Each PDS 2D import is logged at several positions:

- In short form:
  - In the properties of the P&ID
  - In the properties of the in the import involved DB choice object
- In long form: In an XML file

13.2.3 Elements and basic settings

13.2.3.1 DB choice object

Definition

DB choice objects are base objects. They fulfill the following purpose:

- Saving database-specific import settings:

  Your administrator creates a DB choice object for each PDS 2D database from which you import.

  In the DB choice object, the administrator defines the import settings that apply for an import from the PDS 2D database.
- Linking the P&ID to the DB choice object:

  COMOS can only perform a PDS 2D import for a P&ID if it knows what import settings are used for this import. For this reason, each P&ID for which you call a PDS 2D import must have a link to a DB choice object.
- Collecting DGN base objects:

  For each cell that is imported with a DGN import managed by a DB choice object, COMOS creates an import base object under the DB choice object. When you import DGN drawings that use different PDS 2D databases, the import base objects are therefore managed separated by databases.
- Starting the PDS import for all P&IDs linked to the DB choice object

Basic settings

Your administrator configures the DB choice objects. You usually will not change his default settings.

See also

PDS 2D (Page 229)

13.2.3.2 DGN import objects

Definition

When you import a DGN drawing, the following DGN import objects are automatically created:

- Import base objects
- Import engineering objects derived from the import base objects

COMOS converts the import engineering objects into P&ID engineering objects during the PDS import.

How COMOS proceeds when doing so is also defined in the import base objects.

Basic settings

- Administrator:

  The administrator configures the import base objects.

- User:

  Depending on the rights you have for base objects or the editing mode your administrator set for the attributes of the import base objects, you can change the default settings prior to the import.

  Change the default settings only if you have been adequately trained to do so.

See also

PDS 2D (Page 229)

13.2.3.3 P&ID base objects

Definition

The aim of the PDS 2D import is to convert as many elements of the DGN drawing imported in the P&ID as possible into P&ID engineering objects, and to write the PDS 2D data of the DGN cells into the P&ID objects.

How COMOS proceeds when doing so can also be defined in the base objects of the P&ID engineering objects.

Basic settings

The administrator configures the P&ID base objects.

Depending on the rights you have for editing P&ID base objects, you have the option to change the administrator's default settings prior to the import.

Change the default settings only if you have been adequately trained to do so.

See also

Configuring P&ID base objects (Page 234)

13.2.3.4 Import log

Definition

You can check the course of the import by using an import log.

The following applies for the import log:

- The log is created automatically.
- Several logs are created for each import and are stored in different locations.
- There is a short and a long version of the log.

Short version

- Properties of the P&ID, "Attributes > PDS import settings" tab, "Log text" field:

  Short log of the PDS 2D import, which was performed for this P&ID
- Properties of the DB choice object, "Attributes > PDS import settings" tab, "Log text" field:

  Short log of the PDS 2D imports, which were performed for all P&IDs linked to this DB choice object

Long version

You find detailed logs of all PDS 2D imports in your local temp directory, "ComosImport" subdirectory. The imports are logged in XML files.

The following logs are generated:

- One log for each P&ID, for which a PDS 2D import was called

  File name:
  - User-defined: Properties of the P&ID, "Attributes > PDS import settings" tab, "Log file name" field:
  - Default setting: The name of the P&ID
- A log for each DB choice object that was involved in an import

  File name:
  - User-defined: Properties of the DB choice object, "Attributes > PDS import settings" tab, "Log file name" field:
  - Default setting: The name of the DB choice object

13.2.4 Executing a PDS 2D import

13.2.4.1 Linking the P&ID to the DB choice object

Procedure

Ensure that the P&ID is linked to a DB choice object before performing the PDS 2D import. To do this, proceed as follows:

1. Open the properties of the P&ID.
2. Switch to the "Attributes > PDS import settings" tab.
3. "Selected DB choice object" field:
   - Check the link you administrator entered as the default setting in the field.
   - If necessary, link to another DB choice object.

     In the COMOS DB, you find the DB choice objects under the following node: "Import > @DGN base objects for external data import"
   - The DB choice object has to be linked to the same PDS 2D database as the cells of the DGN drawing imported on the P&ID.

13.2.4.2 Starting a PDS import for a P&ID

Procedure

To start the PDS 2D import for a single P&ID, proceed as follows:

1. Open the properties of the P&ID.
2. Switch to the "Attributes > PDS import settings" tab.
3. Click the "Start import" button.

Result

The PDS 2D import is performed for the P&ID.

13.2.4.3 Starting a PDS import for multiple P&IDs

Introduction

Each P&ID for which you run a PDS 2D import is linked to a DB choice object. The DB choice object saves the import settings.

COMOS uses the backpointer procedure for each DB choice object to determine which P&IDs are linked to it. In this way, you can at once start the PDS import for all P&IDs, which are linked to the same DB choice object.

Procedure

To start the import for all P&IDs that are linked to the same DB choice object, proceed as follows:

1. Open the properties of the DB choice object.
2. Switch to the "Attributes > PDS import settings" tab.
3. Click the "Start import" button.

Result

The import is performed for all P&IDs that are linked with this DB choice object.

13.3 DGN import

13.3.1 Overview

You can use the commands in the context menu for an interactive report to import DGN drawings from Bentley MicroStation to COMOS and to make general import settings.

Dependent upon the import settings, select one of the following import scenarios:

- Import graphics
- Create objects during import

See also

"Options" window (Page 279)

Importing a DGN file (Page 161)

Import graphics (Page 160)

Create objects during import (Page 161)

13.3.2 Import graphics

Overview

- The imported graphic is available on the interactive report. There are no graphical symbols next to the COMOS objects.
- You can continue to build pipes in the report. You can edit an imported drawing in COMOS.
- You can import DGN drawings and then use the document management functions (revisioning, for example).

13.3.3 Create objects during import

Creating COMOS objects

- COMOS engineering objects and base objects are created for the cells available on the DGN drawing. The engineering objects as well as all graphical elements that have not been dissolved into objects are displayed on the report.
- The engineering objects created are placeholders. Replace these placeholders with COMOS engineering objects.
- Not all graphical elements can be imported in COMOS objects. For example, pipes are not created by an import.
- After the import and the mapping of graphical elements to COMOS objects, which is largely an automated process, edit the objects.
- You can also make the settings so that only base objects are created.

13.3.4 Importing a DGN file

Requirement

An interactive report has been created.

Procedure

To import a DGN file, proceed as follows:

1. Open the interactive report.
2. Right-click in the report workspace.

   Make sure that no objects are selected.
3. Select "Options > Import" from the context menu.

   The "Import and dissolve drawing" window opens.
4. Make the required settings.
5. To select the import file, click the "..." button.

   The "Open import file" window opens.
6. Select the required file.
7. To close the window and accept your selection, click "OK".

   This takes you back to the "Import and dissolve drawing" window.
8. Click the "OK" button to confirm the settings.

   The "Convert DGN drawing to engineering objects" window opens.
9. Make the required settings.
10. To close the window and start the import, click "OK".

Result

COMOS searches the DGN file for cells. An object is created for each cell found. The remaining graphic elements are imported as pure graphics.

The imported drawing is aligned with the point of origin of the interactive report.

See also

Saving import data (Page 162)

"Import and dissolve drawing" window (Page 275)

"Convert DGN drawing to engineering objects" window (Page 276)

13.3.5 Saving import data

Requirement

You have imported a DGN file.

Procedure

To save the import file, save the interactive report from which you started the import.

If you have only imported base data and the import has not resulted in any changes to the interactive report itself, you must still save the report.

13.3.6 Undoing the import

Overview

If you have imported a DGN drawing and are not satisfied with the result (because an import option was not set correctly, for example), you can undo the import. To do this, close the interactive report without saving. The imported data (base objects, engineering objects, or graphics, for example) is deleted automatically.

Graphical import

If you have selected the "Graphical import only" option in the "Convert DGN drawing to engineering objects" window, the "Undo" button is grayed out in the report toolbar for the duration of the import. Once the data import is complete and you have saved the data, the button becomes available for selection again.

13.3.7 Algorithm for generating the DGN base objects

Introduction

COMOS evaluates the DGN drawing during the import. COMOS creates a base object for each Bentley MicroStation cell found.

If COMOS finds multiple cells with the same name, a base object is only created for the first cell. If two cells have the same name but their properties are different, the information which does not match is lost.

Recommendation

If the DGN drawing contains modified cells, select the "Import always as local script" option. If the option is not selected, modified graphics are lost.

Alternatively, you can also activate the "Search for changed cells" option. See also section "Convert DGN drawing to engineering objects" window (Page 276).

"Import > @DGN"

The base objects generated by the import are located in the Navigator on the "Base data" tab, under the "Import > @DGN" node.

When you import data into an engineering object, only local base objects are generated. These are quite simple objects that contain the following data:

Tab	Attribute	Description
	"Name"	This attribute shows the name of the cell in Bentley MicroStation. If the string is blank, COMOS assigns a default name.
"Symbols"		The graphical symbol for the cell is stored in the properties of the base object, on the "Symbols" tab. The symbol is saved for the drawing type for which you called the import.
	"List attribute"	The texts that were assigned to the cell in Bentley MicroStation are stored as attributes below the list attribute "DGN". If this is a text type that has a name and value, the attribute name is the same as the name of the text in Bentley MicroStation. If it is a type of text without name, the attribute name corresponds to the "Value[n]". The "[n]" here stands for a counter that starts with 0. The attribute value is first set on the engineering side and is identical with the Bentley MicroStation text string.

13.4 XMpLant export

13.4.1 Fundamentals

13.4.1.1 XMpLant standard

Definition

XMpLant is an XML standard developed by Noumenon.

Aim of the XMpLant standard

XMpLant aims to establish a consistent, non-proprietary XML scheme for the storage and exchange of data for use by plant engineers, builders, and operators:

Data created in an engineering application is exported to an XMpLant-compliant XML file. This data can then be imported and processed in all other engineering applications in which an XMpLant interface is implemented, independent from the application and format originally used to create the data.

ISO standard 15926

XMpLant is based on ISO standard 15926 "Industrial automation systems and integration— Integration of life-cycle data for process plants including oil and gas production facilities".

ISO 15926 defines a standard for data exchange and integration of data between different computer systems.

XMpLant allows you to map data you have created in your proprietary software to ISO 15926 and then transfer it to any other desirable proprietary system.

13.4.1.2 Functional scope of the XMpLant export in COMOS

P&ID export

The COMOS P&ID module implements an XMpLant interface. Using this interface, you can export a P&ID to an XMpLant-compliant XML file (referred to as a XMpLant file in the remainder of this document).

You call the export from within the P&ID.

The following data is exported:

- The on the P&ID placed objects:
  - Equipment
  - Instrumentation
  - Functions and their positions
  - Pipes, pipe branches, and pipe segments
  - Pipe parts
  - All other graphics placed on the P&ID for which an object exists in the Navigator
- Data lines and effect lines
- Purely graphical elements of the P&ID:
  - Texts
  - Lines
  - Circles
  - Elbows
- Data from the P&ID itself

Data from the on the P&ID placed objects

The on the P&ID placed objects are exported as PlantItems. The owner structure of the objects is retained.

The following data is exported:

- "Label" property
- The "SystemFullName" property
- Export relevant metadata

  The software uses hard-coded default values or user-defined values for metadata.
- The attributes which are located on the subtabs of the "Attributes" tab and whose "Value" is set.

  There are two options:
  - Export all attributes
  - Exporting user-defined attributes
- P&ID coordinates of the object
- Symbol graphic

Data of the data lines and effect lines

Data lines and effect lines are exported as PlantItems of the "SignalLine" type.

The following data is exported:

- P&ID coordinates

P&ID data

The P&ID is exported as a Drawing .

The following P&ID data is exported:

- "SystemFullName" property
- "Description" property
- Drawing type
- Report frame
- Report header

Graphical elements of the P&ID

The purely graphical elements are exported as ShapeItems.

The following data is of the purely graphical elements is exported.

- P&ID coordinates
- Graphics and/or text

13.4.1.3 Exported owner structure

Owner structure retained

If an on the P&ID placed object is located in the Navigator below another object placed on the P&ID, this owner structure is retained in the XMLpLant file.

Pipe structure

The pipe structure is retained if:

- The pipe segment is saved below the PlantItem node of the pipe branch.
- The pipe branch is saved below the PlantItem node of the pipe.

Functions and positions

When a function is placed on the P&ID, the following data is exported:

- The function
- The position below which the function is located

In the XMpLant file, the PlantItem node of the function is created below the PlantItem node of the position.

Other objects

Assemblies and accessories are saved below the same owner as in the COMOS Navigator.

Examples:

- In COMOS, a valve is located below a pipe branch. In the XMpLant file, the PlantItem node of the valve is located below the PlantItem node of the corresponding pipe branch.
- In COMOS, a nozzle is located below a vessel. In the XMpLant file, the PlantItem node of the nozzle is located below the PlantItem node of the vessel.

13.4.1.4 Prerequisites

The following prerequisites must be met in order to export a P&ID to an XMpLant file.

COMOS version

Installed COMOS version: 9.0 or higher

Target application

The XMpLant export interface in COMOS 9.0 or higher is based on XMpLant version 3.2.0.

Likewise, the XMpLant interface of the applications into which you want to import COMOS data must also be based on XMpLant version 3.2.0.

Customer database

Whenever you export user-defined metadata, the following applies:

The base data administrator has added the "XMpLant" tab and the standard tables it uses to your base data.

13.4.1.5 Workflow

The steps presented below are based on the assumption that you are using user-defined metadata.

Workflow

To export a P&ID to an XMpLant file, proceed as follows:

1. The base data administrator provides the base objects of the objects placed on the P&ID with the "XMpLant" tab, on which he creates the required attributes, as well as any desired default settings.
2. The planning engineer draws the P&ID.
3. The planning engineer specifies the engineering objects:
   - The attributes on the "XMpLant" tab
   - The attributes on the other tabs
4. The planning engineer calls the export from within the P&ID.

Result

The P&ID is written to an XMpLant file.

13.4.2 Elements and basic settings

13.4.2.1 Administration of the "XMpLant" tab

This section is aimed at base data administrators.

Introduction

In the XMpLant scheme, P&ID objects are saved as PlantItem nodes. In order to export objects in compliance with the standard, the interface requires certain metadata types from the P&ID objects. You have the possibility to choose from the following values for the metadata:

- Hard-coded default values in the software
- User-defined values

When using user-defined values, you must define them on the following tab:

- Name: "XMpLant"
- Description: "XMpLant"

Tab and its attributes in the catalog

In the COMOS DB, the tab attributes are defined below the following node:

- "@10 > PID > 0 > 99 > XMpLant"

In the COMOS DB, the tab is defined below the following node:

- "@10 > PID > 3 > 01 > XMpLant"
- The tab has all the attributes of the "@10 > PID > 0 > 99 > XMpLant" node.

Base data level for inserting the "XMpLant" tab

Insert the tab as high up as possible in the P&ID base data.

In the COMOS DB, the tab is inherited to the following nodes:

- "@01 > PID > 01 Selection catalog PI":

  Result: All objects placed on a P&ID have the tab.

  No attribute values are set at this level. The configuration was not adapted to the component types until the below located levels.

- "@02 > 020 Positions":

  The configuration corresponds to the Position type.

- "@02 > 030 Functions":

  The configuration corresponds to the Function type.

Base data level for creating attributes

Rather than using the tab that has been preconfigured in the "@10 > PID > 3 > 01 > XMpLant" node, you have the option of determining yourself which attributes are used on the "XMpLant" tab. The following applies:

- Default values are hard-coded in the software for all attributes of the "XMpLant" tab. When exporting an object, the default value of an attribute is only overwritten if the object has the corresponding attribute.
- You should configure the "PlantItemNodeName" attribute as high up in the base data as possible.

  In the COMOS DB, a value is assigned to the attribute at the following levels:

  – At the levels below "@01 > PID > 01 Selection catalog PI"

  – "@02 > 020 Positions"

  – "@02 > 030 Functions":

- Create all other attributes at the below located levels, depending on the respective project requirements.

Configuring attribute values

Following applies when configuring attributes:

- Either you specify the attribute values in the base data or the users set the attribute values on the engineering side.
- During an export, values set on the engineering object are always written to the XMpLant file.
- If no value was assigned to an attribute, an empty string is written to the export file.
- If a value has been assigned to an attribute, a user-defined value is written to the export file.

Export without the "XMpLant" tab

If an object placed on the P&ID does not have an "XMpLant" tab: COMOS determines which PlantItem type corresponds to this object based on the "Class" and "Subclass" properties of the associated base object.

You will find these properties on the "General" tab of the base object.

Result

- A PlantItem node of the corresponding type is created for the object in the export file.
- Objects whose class and subclass have no corresponding "PlantItem type" (revision clouds, for example) are exported as equipment.
- The hard-coded default values are assigned to the PlantItem node attributes.

See also

Attributes of the "XMpLant" tab (Page 272)

13.4.2.2 Hard-coded PlantItem types

For certain objects, hard-coded PlantItem types are used regardless of the value the "PlantItemNodeName" attribute has on the "XMpLant" tab.

Pipes

The following PlantItem types are used for pipes:

	Three-level pipe structure	Two-level pipe structure (pipe branch and pipe segment)
Pipe	"PipingNetworkSystem"	"PipingNetworkSystem" dummy node
Pipe segment	"PipingNetworkSegment"	See three-level structure
Pipe segment	Not an own PlantItem node. A Centerline node is created underneath the corresponding PipingSegment node. The Centerline can have any number of coordinate nodes.	See three-level structure

Functions and positions

- Function: "ProcessInstrument"
- Position: "InstrumentLoop"

13.4.3 Exporting P&ID to an XMpLant file

Procedure

To export a P&ID to an XMpLant-compliant XML file, proceed as follows:

1. Open the P&ID.
2. Open the context menu in the P&ID.
3. Select the "Options > Export > XMPlant" command.
   The "Save As" Windows dialog opens.
4. Specify the path and the name of the export file.

Result

The P&ID is exported and saved as an XMpLant file.

13.5 XMpLant import

13.5.1 Basics of XMpLant import

Overview

In COMOS, you can import XMpLant drawings in an interactive P&ID report.

In the import settings, you specify whether the objects of the XMpLant drawing are to be imported as base objects only or as engineering objects.

COMOS supports XMpLant Version 3.3.0.

13.5.2 Importing XMpLant drawings

Requirement

An interactive report has been created and opened.

Procedure

To import a XMpLant drawing, follow these steps:

1. Click on the working area of the report.

   Make sure that no objects are selected.
2. From the context menu of the report, select the command "Options > Import".

   The "Import and dissolve drawing" window opens.
3. Enter the required import settings.
4. Select the required import file.
5. Click "OK" to start the import.

Result

The XMpLant drawing is imported into the interactive report. Depending on the selected import settings, you specify whether the objects of the XMpLant drawing are to be imported into the project as base objects only or as engineering objects.

See also

"Import and dissolve drawing" window (Page 275)

Administration 14

14.1 Structure of base objects

Introduction

The base objects are stored in particular categories. These categories structure the base objects and simplify the search for required objects.

"@01 Material"

All objects which can be ordered are located underneath the "@01 Material" node.

"@02 General objects"

The following objects (examples) may be located underneath the "@02 General objects" node:

- Signals
- Queries and scripts
- Search objects
- Project structures

"@03 Structures"

The structures for the "New" context menu of the objects are located underneath the "@03 Structures" node.

"@10 @Y Attributes catalog"

The tabs and attributes which can be used on multiple occasions are located underneath the "@10 @Y Attributes catalog" node.

14.2 Unit structure

14.2.1 General

Context menu

Various unit structures have been predefined in the COMOS DB. Depending on which project structure was selected in the properties of an engineering project on the "Links" tab, "Project structure" field, other unit structures can be accessed by selecting the "New" command from the context menu.

The unit structures displayed when you select the "New" command are located in the base data under the base object node "@03 > PID > 310 Device preconfiguration New button".

The structures according to EN and ANSI follow the usual planning structures of the chemical industry.

14.2.2 Reports and administration objects

Reports that are required for and simplify planning are provided within the unit structure. This applies above all to structures conforming to EN and ANSI. With the exception of a very small number, these reports are all structured in the same way. The AKZ structure has the same administration objects as the structures according to EN and ANSI from a certain hierarchy level onwards.

Structures according to EN and ANSI

- EN: "@03 > PID > 310 > EN > U > 0 Plant" and ANSI: "@03 > PID > 310 > ANSI > U > A! > 01 Plant/Building/Production":

  A number of evaluating reports are located here. The signal list is an exception. This is a list whose report templates are located in the base project on the "Documents" tab under the following base object node: "CRp > @0 > EF > EFP Signal descriptions".

- "@03 > PID > 310 > EN > U > 0! > H1 Unit" and "@03 > PID > 310 > ANSI > U > A! > 01! > 01 Main unit":

  P&ID reports can be created in the structures below the main unit.

- "PI > EN > U > A > 01 > 01 > 01 Subunit" and "@03 > PID > 310 > ANSI > U > A!> 01! > 01! > 01 Subunit":

  – P&ID reports are created below the subunit.

  – A number of category folders are created automatically below the subunit when a subunit is created.

    The functions category creates a clearer structure for the engineering objects, for example, by automatically collecting all pipes placed on a P&ID in one folder. You find more information on this topic in the "COMOS Administration" manual, keyword "Categories".

    You do not usually need the "I&C" folder until I&C planning. The positions are created below it, and the functions are created below. However, for P&ID engineers, it is sufficient to simply place functions on a P&ID report. If, in this case, you enter a function code, a position is automatically created at the function and both of them are moved to the "I&C" folder by means of the "function" category. See also section Creating positions (Page 58). You find additional information on this topic in the "E&IC" manual, keyword "I&C".

  – Placement overview "QDev010 objects with DocObj"

    This query in the context menu of the subunit is an example of how to collect information.

    Class:

    The default is "pipe". However, you can set "All", for example.

    The start object is automatically the subunit.

    There is a "Placing filter" button on the symbol bar of the query. Drawing type "RI2 (RI 10628)" must be entered here. You now have the option to select the following entries for the placing filter:

    "All", "Placed", and "Not placed"

    Application example: Search through all objects that have not been placed yet and distribute them on the reports.

AKZ structure

Folders that inherit from the same base objects as those of the structures according to EN and ANSI are likewise created automatically below level 6 under the base object nodes "@02 > 010 > 1 > AKZ > 06 Level 6: Sequential number).

14.3 Base objects

Overview

The "@01 > PID pipes and instrumentation" base object node contains devices and administration objects that are required exclusively for pipes and instrumentation.

14.3.1 Base object node "@01 > PID > 01 Catalog PI"

Overview

Base object node "@01 > PID > 01 Catalog PI" is a general device catalog.

All of the devices contained in this base object node have certain basic properties. Furthermore, there are objects that bring along additional properties and abilities.

14.3.1.1 Labeling

Catalog "PI"

The sorting does not directly comply with the standard, but instead attempts to offer the most important sections of 26004 and 2401. The catalog in "PI" is not a closed labeling system that can be used immediately.

Most of the base objects also have ID letters conforming to 26004 and ISO 10628 (formerly DIN 28004/DIN2481).

14.3.1.2 Symbols

Drawing types

All simple devices have one or more symbols on the "Symbols" tab. As a rule, the drawing types "RI", "RI1", and "RI2" are equipped with symbols:

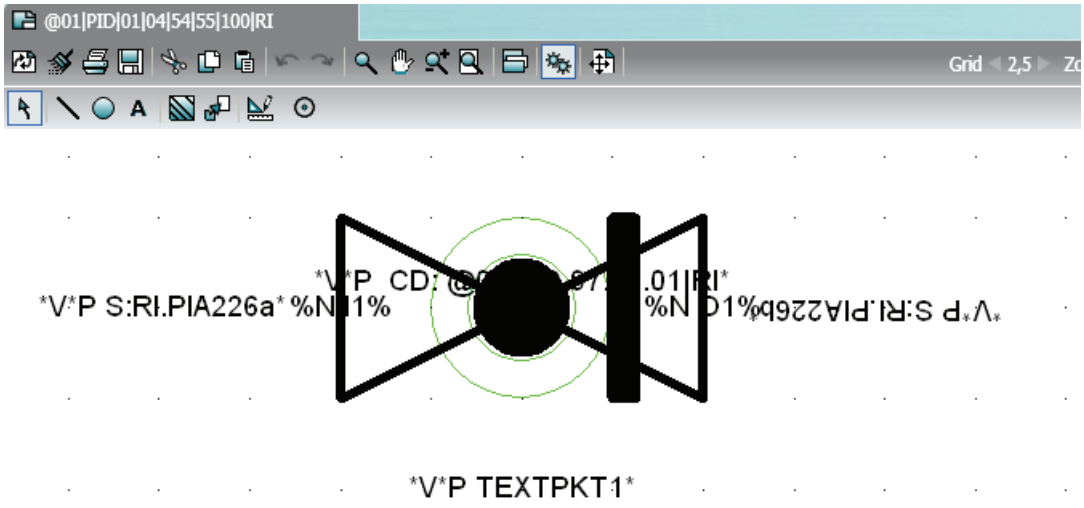
Type	Drawing type	Symbol	Text
RI	PFD	SCRIPT	<SCRIPT>
RI1	P&I (DIN 2481)	SCRIPT	<SCRIPT>
RI2	P&ID (ISO 10628)	SCRIPT	<SCRIPT>
RI_ANSI	ANSI-PIP for P&ID	SCRIPT	<SCRIPT>
RI_ANSI_M	ANSI metric	SCRIPT	<SCRIPT>

- "PID" drawing type: Main area of application in PFDs.
- "PID1" drawing type: Main area of application in power stations/DIN 2401.
- "PID2" drawing type: Main area of application in chemical plants/DIN 28004 or new: ISO 10628.

You can modify the symbols to suit your requirements. Symbols are hierarchically inherited downwards.

Graphical symbol

You can change the graphical symbols in the properties of the base object, on the "Symbols" tab. To open the Symbol Editor, double-click in the symbol column.



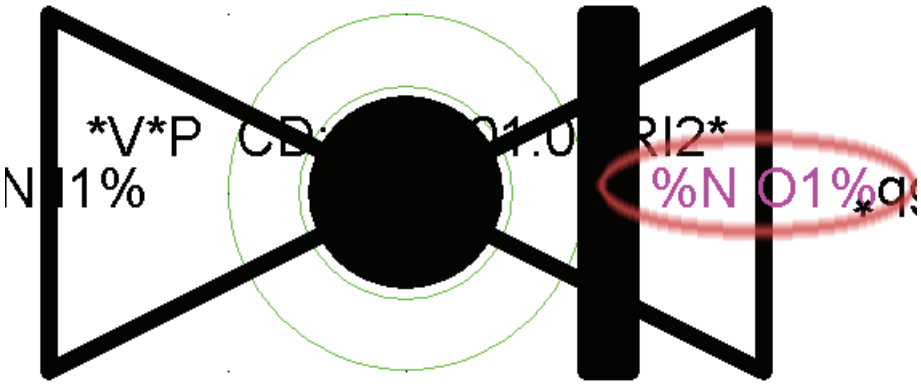
In the Symbol Editor, you can modify the graphical elements of the symbol using various drawing tools. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Symbol Editor".

The drawing type symbols have a fixed size and are optimized for a grid and a scale. Various options enable the symbols to also be used on reports with a different grid and scale.

Each symbol has a placing point which is used for positioning on the grid. The green circles in the figure above identify the point of origin. The placement point only seldom coincides with the top left-hand corner, instead it is designed in such a way that the connectors can lie on the grid.

To extend a graphical symbol, insert functions. The following function types are available:

- Functions that determine which connector points a symbol has and where they are attached to the symbol:



  See also section Connectors and auxiliary connectors (Page 180). You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Connector points".

- Functions that attach subsymbols, i.e. additional symbols, to the graphical main symbol.

  Example:

  A valve can receive a drive. If you have specified a drive, it should be added to the main symbol of the valve as a subsymbol.

  Whether or not the subsymbol has its own DocObj depends on which function is used to create the subsymbol.

  You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Subsymbol/Additional symbol".

Text symbols

You call text symbols in the properties of a base object, on the "Symbols" tab. To do this, double-click in the "Text" column. Define a text symbol that is attached to the graphical symbol of the object by means of `*V* P Textpkt1*`.

Many objects from the base object node "@01 > PID > 01 Catalog PI" already have a text symbol on the top level.

Type	Drawing type	Symbol	Text
RI	PFD	SCRIPT	<SCRIPT>
RI1	P&I (DIN 2481)	SCRIPT	<SCRIPT>
RI2	P&ID (ISO 10628)	SCRIPT	<SCRIPT>

This text symbol is inherited to all base objects below, but not evaluated for all base objects. It is only evaluated at those base objects that call the text symbol by means of `*V* P Textpkt1*`. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Text symbol".

The P&ID text symbol usually outputs the position of the device.

The header of the script for the text symbol contains blind entries:

```
Header.Layer = "10"
Header.Class = "e72"
```

Layer "10" is a random number; it is used to bundle specific information. You can also enter another random layer. Depending on the individual data structure, it is necessary to ensure that you do not inadvertently use a layer that is already being used for some other purpose.

Header "e72" is a random description that is only used to bundle specific items of information. You can also enter another random header. Please note that a number of headers have been pre-assigned by the system. Thus, for example, `Header.Class = "eZ"` means that the text cannot be moved.

Duplicate connectors in the symbol

When you evaluate a P&ID symbol on a diagram, a search is run for duplicate connectors. If COMOS finds a duplicate connector, the symbol on the diagram is displayed as inconsistent (red) and a corresponding error entry is made in the error list.

A duplicate connector exists if there are at least two connectors in the symbol at the same position.

You can declare connectors with duplicate graphic links to be valid (so that no red connector point is drawn) by implementing a script function in "UserScriptBlock1" at the base object. For example, an inconsistency of this type caused by duplicate connector assignment by a process stream and a reference stream can be removed using the script.

Function `IsPartnerConnectionValid`(cc1, cc2, ValidationCode)

cc1 (input): COMOS connector of the dedicated engineering object.

cc2 (input): COMOS connector of the graphically linked partner engineering object.

ValidationCode (input): The status of the two COMOS connectors in relation to one another, where 1 means that the COMOS connectors are not interconnected because at least one of the two connectors might be linked to another connector.

IsPartnerConnectionValid (Output): Bool

- TRUE :

  The graphical connection is valid (no red connector).

  The Con2 (a piece of equipment) connected to the stream Con1 is not displayed in red.

- FALSE (default):

  The graphical connection is invalid (red connector).

14.3.1.3 Connectors and auxiliary connectors

P&ID objects

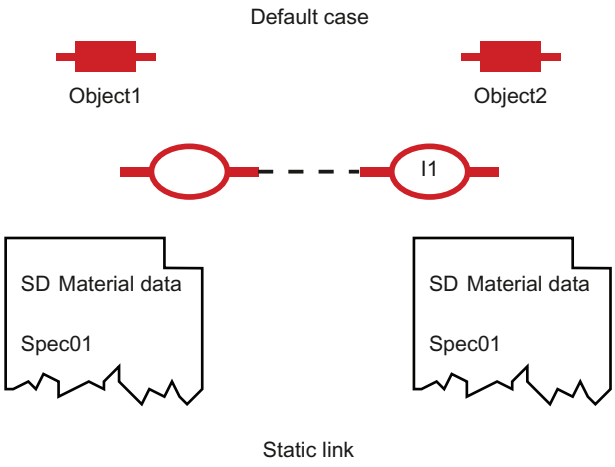
Whether or not a P&ID object has connectors depends on the purpose of the application. Auxiliary objects, such as stirrers, for example, do not have connectors.

In P&ID there are two types of connecting objects:

- The connectors are not joined directly but are linked by means of a pipe (connection by means of an object).
- The connectors are joined directly by means of data lines or action lines (connection without an object).

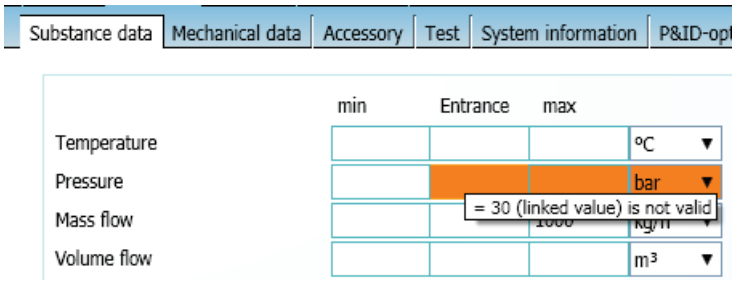
Auxiliary connectors are possible, but are hardly ever used in practice.

In the P&ID module, connectors are also used to transport data. For this purpose the attributes use a "By connector" type link. For example, substance data is transported this way:



The "target attributes" of the substance data search for an object via connector "I1", and from the "SD" tab of this object they take the value of the corresponding attribute (operator is "=").

In the COMOS DB, attributes that are linked via their connectors are identified. A corresponding tooltip appears when you hover the cursor over the field:



Measurement functions work somewhat differently in this respect. See also section Functions (Page 61).

Assessment of report connectors and COMOS connectors

In the P&ID module, COMOS stores whether you joined the connectors in the report first or in the Navigator first.

If you joined the connectors in the report first, the settings of the report connectors take priority over the settings of the COMOS connectors.

If you joined the connectors in the database first, the settings of the COMOS connectors take priority over the settings of the report connectors.

You find additional information on this topic in the "Comos.dll" class documentation, keyword "Document".

14.3.1.4 Attributes, general

Device attributes

Each device is provided with its own special attributes on the "Attributes" tab. The attributes are inherited hierarchically; if necessary, they are supplemented or modified by additional attributes at each level.

Attributes that can be used centrally are also managed centrally. The base object node "@10 @Y Attributes catalog" is used for this purpose.

14.3.1.5 "SYS system" tab

Permitted multiplacement

Use the "Permit multiplacement" option to control multiplacement.

See also section Miscellaneous (Page 127).

Attribute "IsPipeFitting"

Pipe fittings can have the "Sort SYS.IsPipeFitting parallel to cut pipe element" attribute:

- This attribute has the display type "Checkbox".
- The default value is "0".
- Inheritance source: "@10 > PID > 0 > 99 > Sort SYS.IsPipeFitting parallel to cut pipe element"

If you select this option, in the engineering view the insert is always sorted parallel to the connected pipe branch or pipe segment. This makes it part of the pipe structure.

If you deselect this option, the fitting is sorted either directly below the P&ID report or into its designated category. How the fitting is sorted depends upon the options script of the report template or the unit configuration respectively.

Background: Older P&ID structures featured a "GD" tab. This tab no longer exists in the new COMOS DB. This explains why the old functions do not work properly. The "GD" tab has been replaced by the new variable "SYS.IsPipeFitting". If this is set to "1", a "True" is now also returned. Therefore: If you create an attribute with the name "IsPipeFitting" on the "SYS" tab on the base object and set the corresponding value to "1", the component is declared as a fitting.

This attribute is already used in the COMOS DB, e.g. for reducers.

Attributes on the report bar

COMOS provides a general method which simplifies the setting of attribute values for engineering objects.

The attribute values are set via the report bar of an interactive report or 3D space. Only attributes that were input in the "VSUI" attribute are offered for selection on the report bar.

In the P&ID, the "VSUI" attribute must appear on the "System" tab.

You can find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Editing attributes on the report bar".

Attribute "Behavior"

If the "SYS.Behavior" attribute exists, the first character in this field sets the RIClass. If this attribute does not exist, the previous behavior applies (backward compatibility). Currently only the letter "I" for Instrumentation is supported. Therefore, you can set "SYS.Behavior = I" to force a component to behave in such a way that would otherwise only be typical of measurement functions.

Two valves that have the above setting are assigned an objectless connection instead of a pipe connection when they are connected with the "Connection" tool.

Application example: Simplified display of ISA functions

14.3.1.6 Elements

Element "@3D"

You can find the "@3D" element at various places in the P&ID base data.

Example: "@03 > PID > 310 > EN > D > 01 > 03 > 03 Vessel with lagged floor, gen."

These elements have no meaning within P&ID. They enable P&ID base objects to be used in the 3D applications of COMOS as well so that you can work with the same objects in the P&ID and in 3D. In other words, you can work in 3D with a P&ID object below which the 3D element is created as an implementation.

The "@3D" element has a "GD Geometry" tab that stores all data of the P&ID object that is required for working in 3D.

To set 3D mode, call the properties of the attribute and select the "3D mode" option on the "Link" tab.

"GD" tab for the P&ID object and project setting

Formerly, P&ID objects could be used in the 3D applications of COMOS if they had a "GD Geometry" tab. This tab stored all information required to work with the object in 3D. The checkbox "Only apply values linked through the GD tab" had to be activated in the project properties, "Attributes > Process engineering" tab, so that, of all the attributes linked via connectors, only those that had been defined via the "GD" tab were evaluated.

In the meantime this methoud is outdated. If you work with the "@3D" element, the "GD Geometry" tab of the element is used. The project option "Only apply values linked through the GD tab" is not relevant any longer.

Exception: In the case of vessels, the project option "Only apply values linked through the GD tab" takes priority and "wins", so that a "GD Geometry" tab is required at the P&ID vessel. Hence, if the method involving "@3D" elements is used, the vessel is then not allowed to have a "GD" tab and the project option must be deactivated.

14.3.2 Base object "@01 > PID > 01 > 01 > 03 Vessel"

Elements of the vessel

PFD vessels can have hierarchically subordinate objects such as nozzles. These objects are created as elements of the vessel.

The base object of the vessel must have a nozzle with the name "NOZZLE" as an element so that the nozzle can be created automatically on connection. See also section Connecting pipes with components (Page 91) and section Base object "@01 > PID > 98 > 05 Nozzles + connections" (Page 184).

Connectors

Vessels are joined via dynamic connectors. The following steps must be carried out when connecting a pipe to a vessel:

1. A dynamic connector is created at the vessel.
2. A dialog prompts the user to decide whether a nozzle is to be generated.
3. If the user confirms this dialog, a nozzle connected to the pipe is created as before.
4. In this case the dynamic connector at the vessel is not required. This connector at the vessel is not connected and is deleted at the next opportunity.

Special case:

If a nozzle with two connectors is used, the dynamic connector at the vessel is retained. See also section Base object "@01 > PID > 98 > 05 Nozzles + connections" (Page 184).

Miscellaneous

In the COMOS DB, vessels have the "Position" class and the "Equipment" subclass. See also section Positions (Page 57).

14.3.3 Base object "@01 > PID > 01 > 04 Valves"

Valves

COMOS does not distinguish between general valves and special actuators. The valves are prepared so that they can also be used as actuators. The actuators belong to the actuating functions. See also section Functions (Page 61).

In the COMOS DB, you can use the context menu to supplement the symbol for valves in the report. You can display flange connectors and map safety positions, for example. To do this, set the attributes provided in the "UserScriptBlock1" script block ("PI040.PIA045 Drive" and "PI040.PIA043 Continuous control", for example).

14.3.4 Base object node "@01 > PID > 98 Elements"

General

Under this base object node there are objects that, as a rule, are used as constituent parts of another object (the drives for valves, nozzles, and column fittings, for example).

These objects are then entered on the "Elements" tab for the main objects, making them available when you select the "New" command from the context menu.

14.3.4.1 Base object "@01 > PID > 98 > 05 Nozzles + connections"

General

Nozzles are used to connect pipes to objects or objects to pipes.

Nozzles are only used in the P&ID module, because in COMOS, objects without nozzles are managed in the PFD module.

Using nozzles

For a component to be connected automatically via a nozzle, the nozzle must have been input in the base data as an element of the component and must have the name "NOZZLE". Please note that the name is case-sensitive (all letters upper case)!

You can use both of the following methods to create nozzles on the report:

- Recommendation: Make a connection to the object on the report. The window contains the "Create nozzle" option. See also section Connecting pipes with components (Page 91).

  The option is deactivated if the object does not have a "NOZZLE" element.
- Alternative: Use the "New" command from the context menu for the component to create the nozzle in the Navigator. Then drag the nozzle onto the report. When you create a nozzle in the Navigator, it is always aligned to the right. If the nozzle is to point in another direction, rotate it first. Only then can you place the nozzle on the object.

Nozzles with two connectors

A nozzle can also have two connectors instead of one.

If a nozzle with two connectors is used, the dynamic connector that is generated when creating the vessel is not deleted.

The one connector of the nozzle is joined with the pipe in the usual way. The second connector has to be created in such a way that it lies on the edge of the vessel in graphical terms and moreover, precisely at the point at which the dynamic connector of the vessel is created.

In this case the vessel connector is joined with the second connector of the nozzle.

Standard graphical P&ID mapping is used in this case. The connector from the vessel and the connector from the nozzle can find one another because they are located on top of one another.

The connection is broken if you move the nozzle. However, this cannot be seen at once on the report, because the "connector ring" is located on the nozzle and continues to be displayed.

"Technical data" tab

Information on the nozzle itself is collected here.

"Substance data" tab

Information on the medium to be transported is collected here. Many of these attributes are linked and pass on their information via their connected components.

"P&ID Options" tab

Information that is evaluated on the P&ID report is stored here (description, position on the Apparatus toolbar, for example).

14.3.5 Base object node "@01 > PID > 99 Additional graphics and symbol construction"

14.3.5.1 Base object "@01 > PID > 99 > 01 P&ID labeling symbols"

Objects for pipe flags have been predefined under base object "@01 > PID > 99 > 01 P&ID labeling symbols" in the COMOS DB.

Class: "Data set"

These objects create a text flag when they are dragged onto a P&ID report. If the text flag is dragged onto a pipe, it then docks to the pipe and is automatically moved or deleted along with it.

These objects do not create objects in the engineering view but are only found on the report. The objects of this base object node belong to pipes and must be placed in such a way that a connector to a pipe is created.

When you select a flag on the report, the menu for the pipe is offered for selection from the context menu.

Attributes

"P&ID Options" tab:

- Inheritance source: "@10 > PID > 3 > 01 > RI > RI P&ID options"
- "RI.REFLECT_ALLOWED| Graphical mirroring permitted"

  Activated: The text flag is turned around if the text flag is connected to a pipe and the flow direction is changed.

  Default = FALSE,

  Graphical flags (without text): TRUE

  Text flags: FALSE, otherwise there could be errors in the text assignment.
- "RI.ROTATION_ALLOWED| Automatic rotation permitted"

  Activated: The symbol is aligned to the symbol of the pipe. If the pipe is located vertically, the symbol is also placed vertically, and so forth. The texts within the symbol are rotated automatically in the appropriate direction.

  In this case the manual rotation is disabled. Although the rotation grab is still displayed, rotation does not have any effect and the flag always jumps back to the direction specified by the pipe.

  Default = FALSE,

  Graphical flags (without text): TRUE

  Text flags: FALSE, otherwise there could be errors in the text assignment.
- "RI.SELECTEDWITH_ALLOWED| Automatic selection permitted"

  Activated: Only takes effect if the text flag is connected to a pipe. When the pipe is selected, the text flag is automatically selected as well, even if the text flag no longer has any direct contact with the pipe.

  Default = FALSE

  Objects that can be selected along with their main object (i.e. their device is selected): TRUE

  Text flags (e.g. legends, etc.): FALSE

Text flags without direct contact to the pipe

The text flag can also be torn off after docking onto the pipe. The logical connection is retained and the text flag continues to output the pipe information.

Alternatively, the text flag can be freely positioned right from the beginning. The flag is then connected to the pipe as follows:

- Select pipe, "Copy" command from context menu
- Select text flag, "Reference > Connect with" command from context menu

14.3.5.2 Base object node "@01 > PID > 99 > 02 P&ID graphical symbols"

- Class: "Element"
- Subclass: "Graphic"

Objects that are based on these base objects are always created below the P&ID diagram.

In contrast to the P&ID components that are located below the P&ID diagram, they are by default colored black instead of blue.

Reason: Later on these objects are not be sorted into one of the categories that are available under the unit. Visual labeling on the diagram is not necessary.

The color set on the report can be changed by selecting "Options > Graphical properties" from the context menu.

Most of the objects from this base object node are placed on pipes. If the symbols touch a pipe, the pipe turns yellow. Their purpose is to provide additional information on a pipe in purely graphic terms. Most of the objects objects that are created in the engineering view do not even have any attributes.

However, some of the objects have connectors and are connected to the pipe with these connectors when placed on a pipe.

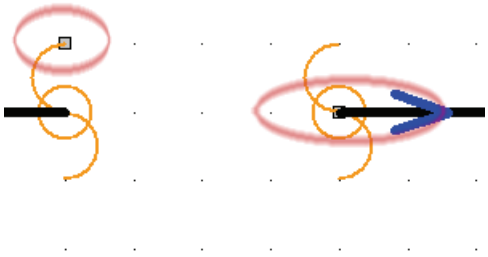
The function and use of most of the graphic symbols is self-explanatory. Therefore the following only introduces some of the graphic symbols.

Base object "@01 > PID > 99 > 02 > 01 Pipe break"

A symbol for a graphic break.

Drag the base object onto the pipe. The pipe is interrupted visually and two break symbols are displayed.

The symbols have two grab points:



The grab point that is located on the arm of the break (figure on the left) scales the break symbols.

The grab point that is located on the pipe (figure on the right) increases the distance between the break.

The pipe is retained as a whole in the database, but it is segmented and joined with the connectors of the pipe break via its connectors.

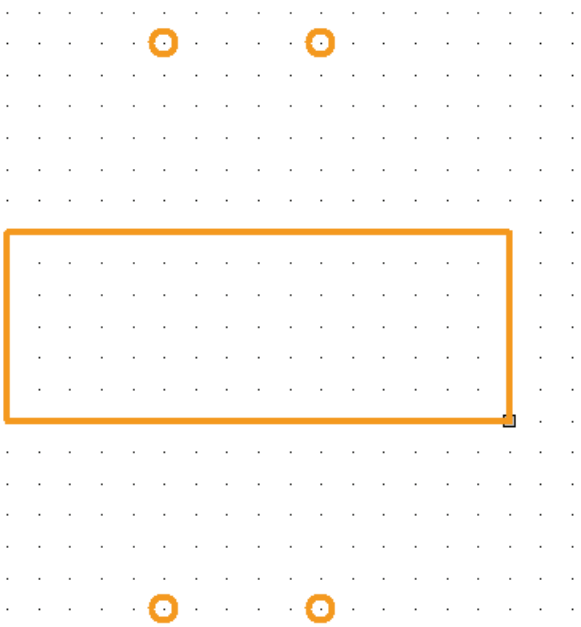
Base object "@01 > PID > 99 > 02 > 02 Page reference"

A page reference is a symbol that groups connections across pages visually in one place on the report. To use page references, proceed as follows:

1. Create a page reference on the first report.
2. Create a page reference on the second report.

   You are not using an object on two page reference reports (as is the case with pipes), but instead each report has its own page reference object.
3. Connect the pipes to the page reference object on each of the reports.
4. If not enough connector points are visible, drag the symbol to enlarge it. To do this, single-click on the page reference symbol twice.

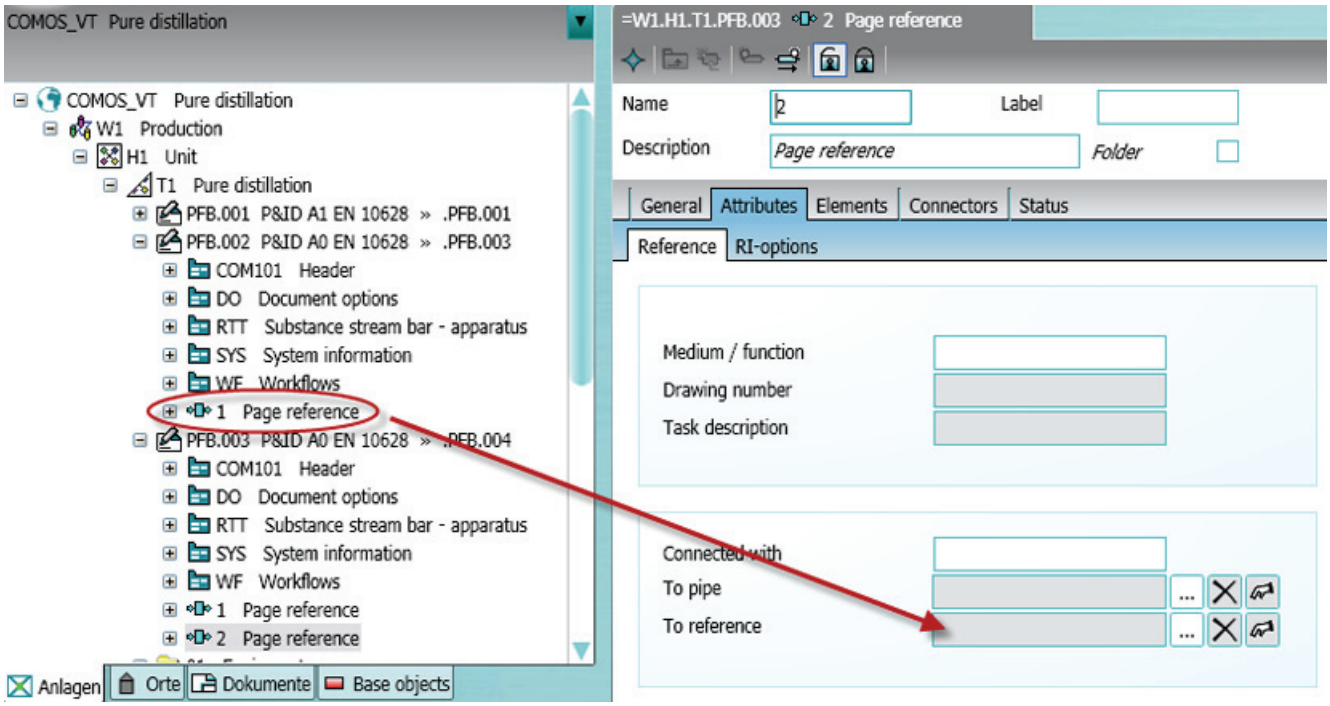
   A grab point appears in the lower right-hand corner:



   When you enlarge the symbol by dragging this grab points, additional grabs points appear at each of the sides.
5. Open the properties of one of the page references.
6. Select the "Attributes > Reference" tab.

7. Drag&drop the other page reference object into the "To drawing reference" field.

   The page references are assigned in pairs.



8. Update the reports.

Result

Information about the corresponding counterpart is displayed in the page reference symbols.

Inheritance source: Tab "@10 > PID > 3 > 01 > PI100 Reference"

Two-way page references are not displayed for duplicate placements of pipe segments (segments placed on two different report documents).

"@01 > PID > 99 > 02 > 11 Revision cloud "

A revision cloud is also a purely graphic piece of information. However, unlike the other base objects below "@01 > PID > 99 > 02", the revision cloud does not belong to a pipe.

This object has a large number of freely movable grab points that you can use to shape the "cloud" in such a way that all required objects are included in it visually.

The revision cloud has no effect on the COMOS revision.

14.3.6 Base object node "@02 > 200 Queries, scripts, imports, decision tables, eBlocks"

14.3.6.1 Base object "@02 > 200 > PID > Q Queries"

Copy the queries to the required location and open them. The subsequent work steps are self-explanatory.

14.3.6.2 Base object "@02 > 200 > PID > S Script library"

Base objects whose only purpose is to archive a script are managed in this branch. Therefore, these base objects have a script block on the "Script" tab but no other settings.

14.3.6.3 Base object "@02 > 200 > PID > X Imports"

Standard imports without further settings. The imports could also be opened from the "@System" base object node. The only reason that they are located here again is to speed up access.

14.3.7 Base object "@02 > 200 > PID > Q > D01 Description of analysis objects"

Word document

You will find the Word document "Description of analysis objects" under "@02 > 200 > PID > Q > D01 Description of analysis objects". The document contains information about the analysis objects.

14.3.8 Base object node "@03 > PID > 120 Documents"

Base objects of the report templates for P&ID. See also section Configuring P&ID Reports (Page 217).

14.3.9 Base object "@03 > PID > 600 Interfaces"

The Conval calculation program has a bidirectional link with COMOS. It is used to calculate the following objects:

- Control valves and security valves
- Covers
- Breaker plates
- Pipes

Supported versions

COMOS supports Conval V5 and Conval V6, and the following functions:

- Security valve dimensioning and pressure loss calculations
- Calculation of valve/pipe/vessel dimensions
- Wall thickness/substance data calculations

Request via document

Dedicated document types are available for Conval. You find additional information on this topic in the "Document Management" manual, keyword "Document types".

To call Conval from a document, first create a "Document" type base object in the base data with the required Conval type. Use the "UserScriptBlock" script block for the base object and insert `Sub` twice:

```
Sub ConvalExport( Calcer)
...
End Sub
Sub ConvalImport( Calcer)
...
End Sub
```

Control of Calcer

To transfer texts, use the following script:

```
calcer.cvCalculation.CalculationData.paramByname("Material").text =
calcer.bobj.spec("TD.L08").value + " " +
calcer.bobj.spec("TD.L07").value
```

To transfer attribute values and units, use the following script:

```
calcer.copyval "","TD.L13","L", true
True = Comos-Conval;
False = Conval - Comos
```

To check the "Volumeflow" option in Conval, use the following script:

```
calcer.cvCalculation.CalculationData.paramByname("QmorQv").SwitchSta
te = 1 'masflow - Radiobutton in Conval wird gesetzt
calcer.cvCalculation.CalculationData.paramByname("QmorQv").SwitchSta
te = 2
```

Identifying field names

To display a field name, click in a Conval field and press <Ctrl + Shift> or <Ctrl>.

Application

When you open document with a Conval document type, Conval itself starts up with the default settings. You can then input data in the usual way. When you close Conval, the data is written to COMOS and Conval automatically.

14.3.10 Object classes of P&ID objects

Classes

- Action
- Unit
- Connector
- Data record
- Document
- Document group
- Element
- Function
- Device
- Device request
- Location
- Position
- Revision

Subclasses

- Equipment
- Blackbox
- Graphic
- Instrumentation
- Category
- Object query
- Pipe
- Nozzles
- Symbol

You can find more information on this topic in the "Basic Operation" manual, keyword "Object status for system types CDevice/Device".

14.4 Standard Tables for P&ID

14.4.1 System project

In the system project, you will find a "ConnectionType<TYPE>" standard table for each connector type.

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Connector-specific line types".

See also

Data lines / action lines (Page 97)

14.4.2 Base objects

The following table describes the relevant standard tables:

Standard table	Description
"@SYSTEM > @PipeCutMode Pipe cut mode"	You find additional information on this topic in the "COMOS Administration" manual, keyword "PipeCutMode".
"@SYSTEM > @PIPEENDSYMBOL/Pipe: End symbols"	You find additional information on this topic in the "COMOS Administration" manual, keyword "Pipe end symbol".
"@SYSTEM > @USERLNTYPE User-defined line types"	For the management of user-defined line types. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Line types".
"@SYSTEM > @LINETYPES > <Type>	For the assignment of user-defined line types to drawing types. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Line types".
"@SYSTEM > @CONSYMBOL Connector symbol (spec break)"	Flag in a diagram that signals whether connected objects differ from one another with regard to PipeSpec-relevant attributes. See also section Flag for PipeSpec relevant information (Page 49).
"@IRF_RI P&ID text selection"	Standard table for the management of text functions that are made available in the symbol editor.
"PI > Y > 0 > 0 > 3D > ANSI > H Line thicknesses"	Defines the automatic line thickness of PFD/P&ID objects. Pipes also use this entry.
"PI > Y > 0 > 1 > I Colors"	Defines the automatic colors of PFD/P&ID objects. Pipes also use this entry. All objects placed on PFD/P&ID reports whose counterparts in the COMOS DB are located below the report are displayed in blue. Example: Base objects you have moved to a PFD/P&ID report using drag&drop.

Standard table	Description
"PI > Y > 0 > 0 > IC > SY02 Failure mode"	Stores symbols for the failure mode. For example, the standard table is assigned to attribute "PI040.PIA046 Safety position of valves" so that it can be shown on the report as an additional symbol for the valve.
"PI > Y > 0 > 1 > L > 02 Valves, pumps"	Stores symbols for the flanges of valves and pumps. Is assigned to attributes "RI| PIA226a, PIA226b", etc., which can be used to show and hide the flanges of the valve/pump on the report.
"PI > Y > 0 > 1 > J Logocad line types"	A list of line types that are available in P&ID diagrams. The system accesses this standard table internally. You may extend but not modify or delete this table. You find additional information on this topic in the "COMOS Administration" manual, keyword "Logocad line types".
"PI > Y > 0 > 1 > K Line types for action lines"	A list of action line types that are made available in P&ID diagrams. The system accesses this standard table internally. You may extend but not modify or delete this table. You find additional information on this topic in the "COMOS Administration" manual, keyword "Line types for action lines".

14.5 Pipe structure

14.5.1 References to the base objects of the pipe structure

Project properties

You define which base objects can be used for the pipe structure in the project properties on the "Options > Process engineering" tab.

Field	Description	Predefined object in COMOS DB
"Base object for pipe"	The pipe object for level 1.	@03 > PID > 310 > EN > D > 03 > 01
"Base object for pipe branch"	The pipe object for level 2. This object has an element that is used as the object for level 3.	@03 > PID > 310 > EN > D > 03 > 02 > 01

If no base objects have been specified in the engineering project, the links from the base project are used automatically.

Compatibility with old databases: If the references have not been set in the base project either, a search is made in the base project for base object "@1RI|@PP". The base object no longer exists in the current database.

Pipe specs

If you are working with pipe specs, you might overwrite the references on the "Options > Process engineering" tab with the base objects entered on the "GD Function" tab.

Because pipes are usually created within P&ID basic planning, the base objects used in the COMOS DB for pipes come from node "@01 > PID > 01 > 03".

Exception

When constructing pipes on an isometric drawing, objects that are created on the third pipe level originate from the components catalog ("@ISO Isometry").

14.5.2 First level: Pipe

"System" tab of the base object

The following table describes the settings you select on the "System" tab of the base object:

Base object	Class	Subclass	Creation option	Creation mode
"@01 > PID > 01 > 03 > 01 Pipe"	Position	None	Normal	Free

Folder for pipe objects

On the top (first) level there is an object for a pipe. This is a folder which manages the actual pipe objects, pipe branches, and pipe segments. This folder only exists in the Navigator and you cannot place it on a report.

In the engineering project, the pipe is the owner of the pipe branch. It can have any desired number of pipe branches.

"Elements" tab

On the "Elements" tab, you prepare objects created by default under a pipe, e.g. a pipe branch. This is also how reports, e.g. an isometric drawing, are made available to you in the engineering project.

See also

Second level: Pipe branch (Page 196)

14.5.3 Second level: Pipe branch

"System" tab of the base object

The following table describes the settings you select on the "System" tab of the base object:

Base object	Class	Subclass	Creation option	Creation mode
"@01 > PID > 01 > 03 > 02 > 01 Z Pipe branch" or "@01 > PID > 01 > 03 > 02 > 02 Z Pipe branch (KKS)"	Position	Pipe	Normal	Free

"Connectors" tab

An input "I1" and an output "O1" of type P&ID. The connectors must be named "I1" and "O1".

"Elements" tab

To work in the P&ID, a pipe segment must be predefined here. See also section Third level: Pipe segments (P&ID) (Page 197).

Pipe branches are not placed on the report. If you nevertheless place a pipe branch on the report, COMOS automatically associates this pipe branch with a standard pipe segment, and inserts this pipe segment to the report.

The pipe branch must be placed underneath a pipe in the engineering view. If a pipe does not yet exist, it is created automatically also.

Pipe branches are treated as elements underneath the pipes. This especially applies to copying: The pipe branches are grouped together and hence are always placed under a common owner.

Alternative class for pipe branches

If the pipe branches are not to be sorted automatically underneath objects with class "Position" (in other words, under pipes), an alternative base object can be created and used for pipe branches:

Class: Device

Subclass: Pipe

14.5.4 Third level: Pipe segments (P&ID)

"System" tab of the base object

The following table describes the settings you select on the "System" tab of the base object:

Base object	Name	Class	Subclass	Creation option	Creation mode	Virtual	Inheritance mode
"@01 > PID > 01 > 03 > 02! SEG pipe segment"	SEG (must not be changed)	Element	Pipe	Normal	Free	N times	Active

"Connectors" tab

See also section Second level: Pipe branch (Page 196).

"Elements" tab

The base object of the pipe segment is entered as an element in the "Elements" tab of the pipe branch.

The pipe segment is created automatically as soon as you work on the P&ID with the "Connection" tool. The pipe segment must be placed below a pipe branch in the engineering view. If a pipe branch does not yet exist, it is created automatically. If a pipe container is selected in the Navigator before you start to use the "Connection" tool, the pipe branch with the associated pipe segment is created below it.

Pipe segments also occur when you place an insert with the attribute value "SYS.PIA602 Pipe cut mode: Cut in segments" on a pipe segment in a P&ID report. The branches are created underneath the pipe branch and joined to the fitting by means of their connectors. See also section Cutting a pipe (Page 87).

Pipe segments are abstract objects that provide a logical view of the pipe. When constructing an isometric drawing, however, no pipe segments are created on the third level of the pipe structure. Instead, actual pipes are created (welded, flanged, or bolted, etc.) and other base objects are selected accordingly. You find additional information on this topic in the "Isometrics" manual.

Placing pipe branches multiple times

A new pipe segment is created immediately if drag-and-drop is used to place a pipe segment that has already been placed.

14.6 Attributes of pipes

Certain graphic options such as line thickness, color, line type, or the position on the material stream toolbar can be controlled using attributes. The "Attributes > P&ID options" tab, which contains certain attributes, must have been created.

The following tables describe the attributes which must be available on the "P&ID options" tab:

"BREADTH Line thickness"

Property	Default
"Display type"	Edit field
"Description"	Line thickness
"Name"	BREADTH
"Value"	Value from the assigned standard table
"Format", "Length"	-
"Unit"	Unit system specified via "SYS.NSYS"
"Type"	Numeric
"Standard table"	"PI > Y > 0 > 0 > 3D > DIN > H Line thicknesses"
"Edit mode"	Editable - normal
"Base attribute"	"@10 > PID > 3 > 01 > RI > 03 > RI > BREADTH Line thickness"

"COLOR Line color"

Property	Default
"Display type"	Edit field
"Description"	Line color
"Name"	COLOR
"Value"	Value from the assigned standard table or, if a color was set manually, "-1" (object selected in the report, "Options > Graphical properties" context menu command). The Windows color code is stored internally as LValue. In this case, the color that was assigned to the object on the interactive report applies. The color value can be reassigned at any time via the attribute. This value corresponds to the Logocad color code and was taken from the assigned standard table. If the attribute accesses the Windows palette (RI.COLOR.Value = -1) but no color is currently being read from the Windows palette (XValue(0) without a number, e.g. a blank string), the script color is used.
"Format", "Length", "Unit"	-
"Type"	Alphanumeric
"Standard table"	"PI > Y > 0 > 1 > I Colors"
"Edit mode"	Editable - normal
"Base attribute"	"@10 > PID > 3 > 01 > RI > 03 > RI > COLOR Line color"

"LNTYPE Line type"

Property	Default
"Display type"	Edit field
"Description"	Line type
"Name"	LNTYPE
"Value"	Value from the assigned standard table list If a line type was set manually (object selected in the report, "Options > Graphical properties" context menu command), then only a number is displayed. In the software, this number is used as a key for the corresponding line type. The line type that was assigned on the report applies. The line type can be reassigned at any time via the attribute. The value derives from the assigned standard table.
"Format", "Length", "Unit"	-
"Type"	Alphanumeric
"Standard table"	"PI > Y > 0 > 1 > J Logocad line types"
"Edit mode"	Editable - normal
"Base attribute"	"@10 > PID > 3 > 01 > RI > 03 > RI > LNTYPE Line type"

"PS position substance stream bar"

Property	Default
"Display type"	Edit field
"Description"	Position material flow bar
"Name"	PS
"Value"	Value from the assigned standard table list
"Format", "Length", "Unit"	-
"Type"	Text
"Standard table"	-
"Edit mode"	Editable - normal
"Base attribute"	"@10 > PID > 3 > 01 > RI > 03 > RI > PS position substance stream bar"

"LAYER graphical properties"

Property	Default
"Display type"	Edit field
"Description"	Graphic properties
"Name"	LAYER

Property	Default
"Value"	Sets the level on which the P&ID object is located. Once you have set the level in the "Attributes > P&ID options" tab, the name of the level is displayed. If you have set the level directly in the report using the context menu command "Options > Graphics properties", the key of the level is displayed instead of the name.
"Format", "Length", "Unit"	-
"Type"	Alphanumeric
"Standard table"	"@System > @D > @GRAPHICS > @LAYERS > <R/x>"
"Edit mode"	Editable - normal
"Base attribute"	"@10 > PID > 3 > 01 > RI > 03 > RI > Layer"

Undefined graphical attributes

For "LNTYPE", "COLOR", and "BREADTH", the following applies:

Old label for undefined graphical attributes: -1.

New label for undefined graphical attributes: -999

The label has been changed due to a new line type being set to -1.

Backward compatibility: If required, there is a technique available to convert the label -999 back to -1.

Continuation of graphical attributes

Graphical attributes are normally assigned to the pipe via the context menu. The following definitions exist as to when these graphical characteristics are to be applied to the following work steps:

1. Connection to dynamic connectors

   The pipe is connected to another cable at any desired point. In this case a dynamic connector is created. The following applies:

   – Default:

     CopyPipeConnectionAutoOff is not available or is False:

     The graphical attributes are automatically taken over.

   – CopyPipeConnectionAutoOff is True:

     The user is asked whether a copy is required (graphical attributes are continued) or not (a default pipe is generated).

2. Connector at the component

   The pipe is connected to a component that is connected to an existing pipe via a connector. The following applies:

   – The cut mode of the component is determined.

     Cut mode "Create segment": A copy is generated (graphical attributes are continued).

     All other cut modes: No copy is created (default pipe is created).

Color of the pipe

The color of the pipe takes priority over the color of the 3D object. Example: The color of the pipe in MicroStation shows which medium is flowing through it.

14.7 Pipe overlaps on the report

Exactly how pipes that intersect on the P&ID diagram are displayed is controlled in the options script of the report template.

Pipes intersect

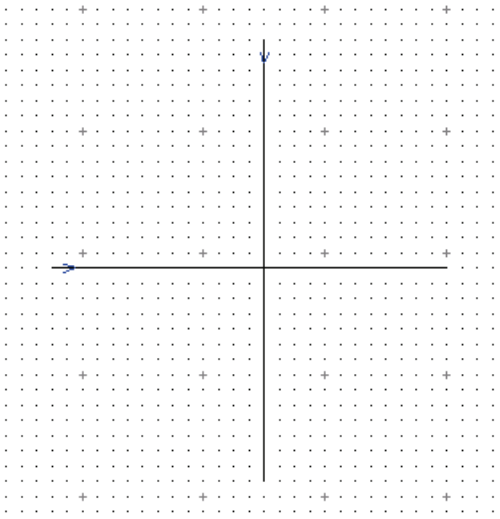
Prerequisite:

Options script of the report template:

`DrawIntersectionArc` is deactivated and `IntersectionRadius` <= 0.

The pipes intersect.

Example:



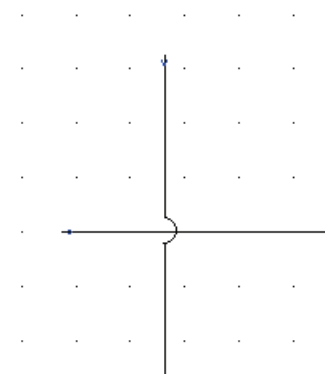
Bridging the pipe with an arc

Prerequisite:

`DrawIntersectionArc` is activated and `IntersectionRadius` > 0.

The pipes do not intersect directly. Instead, a semicircle is drawn at the vertical pipe (the radius is defined by IntersectionRadius).

Example:



The semicircle is always drawn at the vertical connection. The "Options > To foreground" context menu has nothing to do with this setting; neither should it be used in conjunction with this configuration.

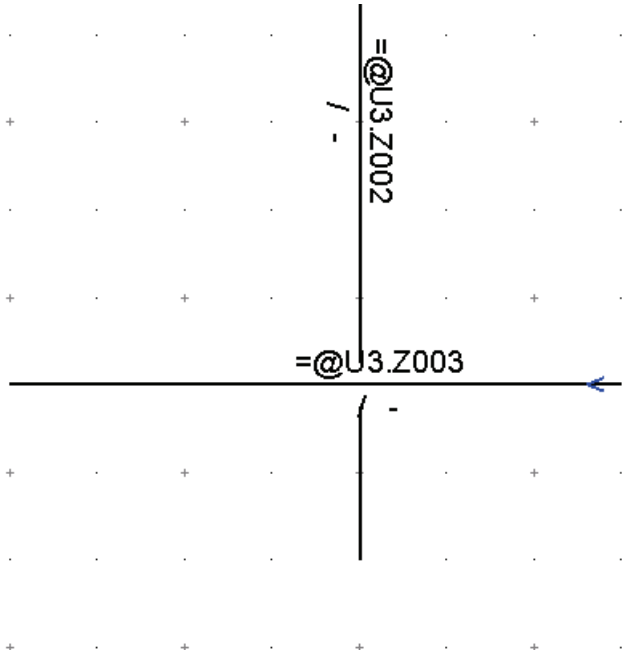
Visually separating a pipe: I

Prerequisite:

`DrawIntersectionArc` is deactivated and `IntersectionRadius` > 0.

The vertical pipe is cut visually by generating an invisible circle with the specified radius. The result is that the connecting line that crosses vertically at the connecting point is interrupted.

Example:



The semicircle is always drawn at the vertical connection. The "Options > To foreground" context menu has nothing to do with this setting; neither should it be used in conjunction with this configuration.

Break pipe visually: II

Should only be used if `DrawIntersectionArc` and `IntersectionRadius` are both deactivated.

Select a pipe and call "Options > To foreground" from the context menu.

The pipes are visually separated by a white line that is drawn in next to the pipe in the foreground. In the options script, the line thickness is specified with `CutLineWidth` and `CutLineConnectorOffset`.

This is an older method to visually interrupt the pipe.

Process streams on P&IDs

If process streams that have been placed on a P&ID intersect, they behave in the same way as a pipe.

Intersecting connecting lines

As soon as the report template option `IntersectionRadius` is activated, the "To foreground" and "To background" context menus are hidden and a deletion area cannot be set up around the connection. In other words: The new intersection method excludes the use of the old method.

Refresh intersections for connection breaks

To refresh intersections for connection breaks, select the "Options > Refresh intersections for connection breaks" command from the report's context menu.

See also

Displaying pipes that cross one another (Page 32)

14.8 Jacket pipes

Standard table of line types

The line type to display the jacket pipe is taken from the normal standard table: A new type, "jacket pipe", is created in the "@SYSTEM > @USERLNTYPE" list.

This line type is declared in the following standard tables: "PID > Y > 0 > 1 > J Logocad line types" and "@SYSTEM > @LINETYPES"

Base object of the jacket pipes

Jacket pipes are available for PFD documents. Create the following attribute as a checkbox at the base object of a pipe branch:

"RI.JACKET"

Base object of the valves

You can control whether or not a valve is to be displayed as surrounded by the heating medium. A "JACKET" attribute is also used for this purpose; it must be created at the base object of the valve as a checkbox:

"RI.JACKET"

Providing a jacket pipe as the default pipe

`CObjectFullNameForPipes` can be used to make the jacket pipe available in the report's context menu as the default pipe:

```
Dim CObjectFullNameForPipes(2)
CObjectFullNameForPipes(0) =
Document.Spec("SYS.PIA601").LinkObject.SystemFullName
CObjectFullNameForPipes(1) = "PI|EN|D|03|02|03"    'JACKET
```

The dialog for the selection of default pipes must be activated to make this work.

You can find additional information on this in the "Reports - Basic Operation" manual, keyword "CopyPipeConnectionAutoOff".

Dynamic connectors

A dynamic connector is created and drawn inside as normal. If you press the <Shift> key, a connection point is still created inside, but the connecting line is only drawn as far as the jacket. This connector is made as a branch. In practice, you can use this method to display the infeed of a heating medium, for example.

Generated engineering objects

Jacket pipes are always created as pipe branches. This also applies if you draw a new connection to a jacket pipe on the report.

Color display when drawing

<Shift> key when moving end grabs: The snap point is displayed in cyan (turquoise) when moving over another connection.

14.9 Implementing a flag for pipe-spec-relevant information

Overview

A script is evaluated to display the flag set in the base project in the "@SYSTEM > @CONSYMBOL" standard table for the corresponding report type.

In the COMOS DB, the script is configured so that it compares the values of the following attributes:

- Nominal diameter (PI030.PIA008), flag text: "Size"
- Nominal pressure (PI030.PIA009), flag text: "Rating"
- Pipe spec (PI030.PIA012), flag text: "Class"

Only attributes linked via their connectors are evaluated: (connection type: "By connector", "Value": "Static").

The way the COMOS DB is configured means that a change of pipe spec or nominal diameter width between two pipe segments or pipe branches will not be indicated by a flag of this type, because the preconfigured attributes have the link type "By owner".

Calling the script

The script is evaluated automatically whenever an object placed on the diagram is evaluated. The script is called just once for each connector (in other words, once per pair of connectors).

14.10 Miscellaneous

Pipe segment: Reverse page order

Initial situation:

A pipe branch that has been divided into multiple pipe segments is placed on precisely two "P&ID" type reports. The "Options > Reverse page order" command is available in the context menu for the pipe segments.

Example

Before the call:

Sheet 1: Seg1 + Seg2 + Seg3

Sheet 2: Seg4 + Seg5

After the call:

Sheet 1: Seg3 + Seg4 + Seg5

Sheet 2: Seg1 + Seg2

The sequence of the pipe segments remains unchanged within a page.

Increasing the processing speed (deactivating the script)

If the cutting of pipes runs relatively slowly, you should check if the script is really necessary.

In the base data, navigate to the base object of the pipe and select the pipe segment element (name: SEG) in the "Elements" tab. Open the properties of the pipe segment element and select the "Script" tab. There you can find a "Connect" script that is not necessarily always required. You might be able to create a pipe base object for which this script is not always called.

Placing attribute flags on pipe segments

Pipe segments do not have an "Attributes" tab or attributes. These data are managed at the pipe branch that owns the pipe segment.

Normally, the owner of an attribute must be placed on a diagram so that you can place the attribute on the diagram using drag-and-drop.

However, where pipe segments are concerned, you can also drag an attribute of the pipe branch below which the pipe segment is located in the Navigator and place the attribute on the diagram on the pipe segment. The attribute must be placed directly on the pipe segment.

Once the pipe branch has been placed, the attribute flag does not need to be placed directly on the pipe branch.

14.11 Pipe structures and channel structures

14.11.1 Using pipe structures and channel structures

Introduction

If the corresponding program code has been commented in in the report template options, the user has the option, by default, to use both pipe structures and channel structures on the report. To do this, the user selects the "Standard pipe for connecting > <required pipe>" command from the context menu.

Procedure

To make the "Standard pipe for connecting" command available in the report context menu, proceed as follows:

1. Open the report template for the report for which you wish to make the command available in the context menu.
2. Right-click on the working area of the report template.

   The context menu opens.
3. Select the "Options" command from the context menu.

   The "Options" window opens.
4. Find the `CObjectFullNameForPipes` entry.
5. Comment in the following lines:

```
Dim CObjectFullNameForPipes(2)CObjectFullNameForPipes(0) = "@01|PID|01|03|01!Z1"
'pipe branchCObjectFullNameForPipes(1) = "@01|PID|01|03|04!Z1" 'duct branch
```

6. Click "OK" to save your input and close the "Options" window.
7. Click "Save" in the report template.

Result

The "Standard pipe for connecting" command is available in the context menu for all reports based on the selected report template. The next time the user opens a corresponding report, the mode is available.

14.11.2 Creating additional pipe structures

Introduction

In addition to the existing pipe structures and channel structures, you can make other three-level pipe structures available to the user in the report context menu.

Procedure

To create other three-level pipe structures and make them available in the report context menu, proceed as follows:

1. In the base project, create a new base object on the "Base objects" tab under the "@01 > PID > 01 > 03 Pipes" node (by copying the existing "01 Pipe" object, for example).
2. Give the new object a unique name.
3. Open the report template for the report for which you wish to make the command available in the context menu.
4. Right-click on the working area of the report template.

   The context menu opens.
5. Select the "Options" command from the context menu.

   The "Options" window opens.
6. Find the entry:

```
Dim CObjectFullNameForPipes(2)CObjectFullNameForPipes(0) = "@01|PID|01|03|01!Z1"
'pipe branchCObjectFullNameForPipes(1) = "@01|PID|01|03|04!Z1" 'duct branch
```

7. Taking your lead from the existing entries, insert a new line for the new base object you have just created.
8. Click "OK" to save your input and close the "Options" window.
9. Click "Save" in the report template.

Result

The "Standard pipe for connecting" command is available in the context menu for all reports which access the selected report template. The next time the user opens a corresponding report, the mode is available.

14.12 Data flow for pipes: UpdateConnected

14.12.1 Introduction

Attribute changes through dynamic connectors offer the advantage that they run automatically. However, overwriting these dynamically referenced values on a single object is not possible and single scripts such as "OnChange" are not compatible with dynamic links.

However, attribute changes based on static links have to be made one by one and manually, but offer full control for updates.

The "UpdateConnected" function automates attribute changes based on static links and also offers the option to influence updating. The passing of attributes to connected components is conducted in one direction until it is ended by stop criteria.

14.12.2 Project properties

"Options > Process engineering" tab

In order for the `UpdateConnected` function to work, as described below, you have to configure the project accordingly.

The "Only apply values linked through the GD tab" option in the project properties on the "Options > Process engineering" tab must not be activated. The settings are only necessary if the tab already exists in the project. It does not have to be especially created.

14.12.3 Data flow in detail

The "UpdateConnected" function is executed recursively for all connectors of the start object; in other words, data flows in all directions and via multiple objects.

The following applies per step:

- Connected components are found through connections.
- Each connected component is searched for attributes that are connected statically via a connector: Dependent upon how you called "UpdateConnected", the search looks for only the attributes appearing on the VSUI bar or all attributes connected statically via a connector.
- For each target attribute, the search looks for the counter connection of the connection entered in the "Connection name" field.
- If the object of the counter connection has the "GetConnectorSpecification" script block, this script block is executed. All objects that have attributes from an attribute family have to implement this script block.

- Attributes from an attribute family further serve as a stop criteria. Their value is only forwarded if the object implements the "GetConnectorSpecification" script block.

  See also section Attribute families (Page 213).
- If the object of the counter connection does not have this script block, the search looks for the source attribute specified for the object in the "Attribute" field, and its value is written to the target attribute.
- If it is not possible to forward a value (because the attribute on the connected object does not exist, for example, or because the attribute is not linked), the data flow for this attribute is stopped at this level.
- If key attributes for pipe spec mapping were updated using the "UpdateConnected" function, the user has the option to start pipe spec mapping once the data has been forwarded.

See also

Initiating the data flow on the P&ID report (Page 52)

14.12.4 Special consideration when calling the "UpdateConnected" function

Introduction

The execution of the "UpdateConnected" function depends on how you have edited your attributes.

Editing attributes

You have the following options for editing the attributes of a P&ID object in COMOS:

- In the tabs of the properties of a P&ID object
- In the properties tree

  You can specify that the properties tree should be displayed automatically when you open a report. To do this, set the `EnableVSUI2 = True` report option in the report template.

  You can find more information on the properties tree in the "Basic Operation" manual, keyword "Properties tree".
- Attributes that are displayed in the toolbar of the report.

  The attributes in the toolbar are only available if the properties tree is hidden.

Updating objects

When you edit the attributes in the tabs of the properties of a P&ID object or in the properties tree, COMOS updates all connected objects.

When you change the attributes in the toolbar of the report, COMOS updates the connections of valves only if you assign the script "GetConnectorSpecification" to the valve. You can find more information about this script under Attribute families (Page 213).

14.12.5 Configuration

14.12.5.1 Prerequisites

The following prerequisites have to be fulfilled in order to be able to execute the data transfer:

1. The "UpdateConnected" function is called.

   The function is only called under certain circumstances.

   See also section Calling UpdateConnected (Page 212).

2. All attributes of components connected with the start object which are to be updated (in other words, the target attributes) must have a "By connector" static link.

   See also section Configuring the "By connector" attribute link (Page 212).

3. If an attribute on a component exists in different forms and the attribute is, therefore, part of an attribute family, the component must implement the "GetConnectorSpecification" script block.

   Example: The two nominal width attributes of a reducer.

   See also section Attribute families (Page 213).

4. In the dialog, which can be opened through "UpdateConnected" if needed, the user must confirm that data flow is to take place.

   See also section Initiating the data flow on the P&ID report (Page 52).

14.12.5.2 Calling UpdateConnected

The "UpdateConnected" function starts and controls data transfer. "UpdateConnected" is called in the following cases:

1. If you enter a new value on the toolbar and then click the "Apply" button.

   All attributes of the start object that appear on the toolbar are forwarded. These attributes are automatically passed on as parameters when the function is called.

   The call of the "UpdateConnected" function cannot be modified to meet customer requirements.

   When you call "UpdateConnected" via the toolbar, the call to "UpdateConnected" via the "OnEditOk" script block is ignored.

2. If a called script block calls the "UpdateConnected" function.

   In the COMOS DB, many P&ID components are configured so that their "OnEditOk" script block explicitly calls "UpdateConnected".

   Consequence:

   Data transfer starts when you click "OK" or "Apply" in the properties of a component.

   All component attributes that have a "By connector" static link are updated, not just those that were edited at the start object.

3. If a base object is placed on a P&ID and is therefore connected with other components.

   All component attributes that have a "By connector" static link are updated.

"UpdateConnected" is not called when you place and connect a component from the engineering data on a P&ID for the first time.

14.12.5.3 Configuring the "By connector" attribute link

Requirement

The link is set up in the base data.

Procedure

Data transfer is implemented using the COMOS attribute link method. To implement data transfer, as the base data administrator, you must proceed as follows to configure all attributes on P&ID components that are updated through data flow:

1. Open the properties of the attribute.
2. Select the "Link" tab.
3. Select "By connector" from the "Link type" list.
4. From the "Connector name" list, enter the name of the connector which links the attribute value to the component.

5. Enter the nested name of the required attribute in the "Attribute" field.

   If the attribute value can be linked to the component by a number of connectors, use commas to separate each of them.

6. Select "Static" from the "Value" control group.
7. Click "OK" to save your inputs and close the dialog window.

Example

Valve: Attribute PI030.PIA008 Nominal diameter DN,

link type = "By connection", value = "Static",

connector name = "I1,O1", attribute = "PI030.PIA008"

The valve is added to a the pipe branch. The nominal diameter of the pipe changes on the toolbar, thereby calling the "UpdateConnected" function. Regardless of whether the update is made from the input or output of the valve, the following applies: The value entered in "PI030.PIA008" at the pipe segment is applied to the nominal diameter attribute of the valve.

You find additional information on this topic in the "System Type Properties" manual, keyword "Linking/Using attributes".

14.12.5.4 Attribute families

There are objects that own attributes with different characteristics. These attributes belong to an attribute family, e.g. the two nominal diameter attributes of a reducer or the two nominal pressure attributes of a pump.

Special features for "UpdateConnected"

The attributes of an attribute family are of particular significance with regard to data exchange:

- A component which is connected to a component, which owns an attribute with several characteristics, needs to know from which attribute it has to take over the value.

  If attribute families come into play, the names of the target attribute and the source attribute will not always be identical. The name of the source attribute may differ dependent upon which connector forwards the attribute value to a component. However, you can only enter one attribute name in the "Attribute" field on the "Link" tab.

  Therefore, components with attribute families have to implement the "GetConnectorSpecification" script. It explicitly maps the connection and attributes to each other.

  Example:

  A reducer is built into a pipe of DN 100.

  Input of the reducer (PIA008): DN 90

  Output of the reducer (PIA008): DN 125

  Result of the installation: The pipe is divided into two pipe branches. Pipe segment A (connected to the input of the reducer) must adopt the "PIA008" value; pipe segment B (connected to the output of the reducer), on the other hand, must adopt the value from "PIA008a".

- They serve as stop criteria in the context of data transfer.

  Example:

  A reducer connects pipe segment A with pipe segment B. The nominal diameter of pipe branch A changes. Up to the reducer, all components connected to A adopt the new nominal diameter. The following applies to reducers: The input adopts the new nominal diameter ("PIA008"), the output ("PIA008a") does not.

  Result: All objects connected directly or indirectly to the output of the reducer retain their old nominal diameter value.

"GetConnectorSpecification"

Components with attributes from attribute families must have the "GetConnectorSpecification" script block.

If a component has this script block, the "Connector name" field is no longer evaluated. Therefore, the script should be implemented so that normal attributes which do not belong to an attribute family are also considered.

Base object: Script block "GetConnectorSpecification:

Name	GetConnectorSpecification
Parameter:	`Device`: Owner of the connector via which the attribute arrives at the component. `ConnectorName`: The name of the connector via which the attribute value arrives at the component (= counter connection of the component which is to be updated). `SpecName`: Attribute name in the link.

Example:

`GetConnectorSpecification` for "@01 > PID > 01 > 03 > 03 > 01 Reducer"

```
Function GetConnectorSpecification (Device, ConnectorName, SpecName)
'Returns specification object of a connector-dependent
'specification link
'     Input: Device - Owner of connected connector
'     ConnectorName - Name of connected connector
'     SpecName - Specification name of link

     Set GetConnectorSpecification = Nothing

    If ConnectorName  = "O1" Then
       If SpecName = "PI030.PIA008" or SpecName = "PI030.PIA009"
Then
          Set GetConnectorSpecification = Device.spec(SpecName+"a")
       Else
            Set GetConnectorSpecification = device.spec(Specname)
       End If
    Else
       Set GetConnectorSpecification = device.spec(Specname)
    End If

End Function
```

You find additional information on this topic in the "System Type Properties" manual, keyword "GetConnectorSpecification".

14.12.5.5 Global flag

"Update objects?" window

By default, the user can make settings in the "Update objects?" window to define if components should be updated and if pipe spec mapping should be performed. The setting in the relevant window can also be applied for the entire duration of the current COMOS session so that the window does not appear again during that session.

See also section Initiating the data flow on the P&ID report (Page 52).

To enable the user to accept one of value differences of an attribute, also always provide the "Always accept the differences" column.

See also Configuring columns of the "Update objects" window (Page 216).

Project settings

The administrator can set up the project so that the corresponding window is not displayed at all and a preconfigured setting is executed.

The settings for the window for updating the components can be influenced using the following variable, e.g. via the Object Debugger:

```
Workset.Globals.UpdateObject
```

1 = Show and hide the dialog window

3 = No dialog window, always update components

4 = No dialog window, never update components

The settings for the dialog window for performing pipe spec mapping can be influenced using the following variable, e.g. via the Object Debugger:

```
Workset.Globals.ExecutePipeClass
```

1 = Show and hide the dialog window

3 = No dialog window, always execute pipe spec mapping

4 = No dialog window, never execute pipe spec mapping

14.12.5.6 Configuring columns of the "Update objects" window

Introduction

You can provide the user with additional columns in "Update objects?" window. Using these columns, the user can accept differences once or permanently when updating the attributes.

Requirement

The "Update objects?" is window open.

Procedure

To configure the columns of the "Update objects" window , follow these steps:

1. Click the "Detail" button.
   The attributes that can be updated are displayed.
2. Select the "Options" command from the context menu of the table header.
   The "Options" window opens.
3. Select the "Edit columns" tab.
4. Enable the desired columns in the "Visible" column.
5. Close the dialog box by clicking "OK".

See also

Global flag (Page 216)

14.12.5.7 Script call

All attributes are checked:

```
Set CInst = CreateObject("ComosPIDUpdate.Lib")
CInst.PIDUpdate. UpdateConnected StartDev, ""
```

Only stated attributes are checked:

```
Set CInst = CreateObject("ComosPIDUpdate.Lib")
CInst.PIDUpdate. UpdateConnected StartDev, "PI030.PIA008;PI010.PIA047"
```

Alternatively to check all attributes:

```
Workset.Lib.RI.UpdateConnected StartDev
```

14.13 Configuring P&ID Reports

14.13.1 Document base objects and their properties

The base objects for P&ID documents are located below "@03 > PID > 120 Documents":

- "> "> PDA data sheet"
- ">PFB flow diagram"
- "> PFB ANSI flow diagram ANSI"
- "> PPB lists"
- "> ZZ SubReport"

The base objects "PFB" and "PFB ANSI" have the tabs that are listed below. The attributes on these tabs control the exact behavior of the flow diagram:

"SYS System"

Attribute "PIA601 Base object for pipe branches":

Type: Link

Determines which base object is used to create the connections that are drawn with the "Connection" tool. In the COMOS DB, the options script of the report template is set up so that the variable `CObjectFullNameForPipe` reads the base object that is input here. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "CObjectFullNameForPipe (String)".

This allows normal users to determine the base object for pipe branches easily via the properties window.

The value that is set here overwrites the settings from the project properties.

Attributes "PIA906 Mode 1", "PIA906a Mode 2", "PIA906b Mode 3": To customize the "Assign object" tool.

The `**Assign Object Mode**` script must be commented in in the report template options. The "Sys.PIA906X" attributes are used here. They determine the display name in the menu of the "Assign object" tool. In this case, the DisplayValue of the relevant attribute is used as the display name.

"RTT Substance stream bar - Apparatus"

In the COMOS DB, the report template is configured so that detailed information about substance streams and apparatus can be output on the P&ID diagram in two row reports.

"RTT201| With apparatus head" attribute: Checkbox activated: Apparatus bar is displayed.

"RTT202 With substance stream bar" attribute: Checkbox activated: Substance stream bar is displayed.

Attributes "RTT211 Display mode for apparatus bar" and "RTT212 Display mode for substance stream bar": Determine in how much detail the bars are displayed.

Stored with standard table "PI > Y > 0 > 1 > G Display mode" from the base project.

Attributes "RTT205 A01" through "RTT210 A06": The values input here are output as the line caption of the apparatus bar.

Attributes "RTT222 S01" through "RTT229 S08": The values input here are output as the line caption of the substance stream bar.

"DO Document options"

Flow diagrams according to EN standard also have a "DO Document options" tab.

Attribute "ND0593 Show equipment list for PID": Self-explanatory.

14.13.2 Report template script options

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Script options in reports".

14.13.3 Quadrants/page areas

You can spilt reports of the "PFD" application into quadrants.

You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Quadrant".

The evaluation is carried out by means of `DocObj.XQuadrant` and `DocObj.YQuadrant`; in reference arrows by means of the `%NToDOcObj.XQuadrant%` functions, etc.

`ToDocObj.X` and `ToDocObj.Y` are likewise evaluated.

14.13.4 Properties of the P&ID report

Deleting free connectors

Context menu of a component, "Options > Delete free connectors"

Deletes connectors that were created dynamically and are no longer needed.

Example: A vessel is connected with a pipe by means of a connector; then the pipe is deleted. The dynamic connector of the vessel is also deleted.

Open

If blank P&ID reports open relatively slowly (five seconds and longer), then you should check the following setting:

Properties of the report, "Substance stream bar - Apparatus" tab, deactivate "With apparatus header" and "With substance stream bar" options.

Automatic connection when changing symbols

If a symbol is replaced by another symbol on the P&ID, a check is made automatically to ascertain whether the new symbol has report connectors that can be connected as specified by the COMOS connectors.

In other words: If an object was connected both in the database and on the report, once the symbols have been swapped, the system tries to restore the report connections on the basis of the joined connectors in the database.

This also happens if the report was closed when the change of symbols took place.

Such a symbol change can take place as the result of a change in an attribute. When the attribute changes its value, a symbol adjusts itself accordingly on the report.

Report context menu

When the context menu for the diagram is called (i.e. no object has been selected), the following commands are available:

- "Options > Set point of origin":

  Determines the point of origin of the report (default point of origin: top left corner). This is evaluated, for example, if a report is placed on another report: Report A will be placed with its point of origin at the location of Report B, which the user has designated with the mouse.

- "Options > Scale":

  Changes the scaling of the report. The symbols are scaled accordingly.

- "Options > Construction":

  Two new switches have been added to the toolbar:

  – "Dimension":

    A purely graphical dimensioning line that can be controlled in the report script using the following options: `DimensionSymbol`, `DimensionTextHeight`, `DimensionUnit`. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Script options in reports".

  – "Hatching": Functions in the same way as the hatching tool of the Report Designer. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Hatching".

- "Options > Autogrouping mode":

  In this mode, P&ID objects that are connected with each other are automatically assembled into a group.

  If you connect two groups together, the two groups merge into one group.

  If you connect an "Element" type component to another P&ID object, the owner of the device is automatically included in the group.

  If you cut a pipe, which was created when autogrouping mode was disabled, the P&ID objects at the end of the pipe are not added to the new group.

- "Options > Import":

  To import DGN drawings. Opens the "Import and dissolve drawing" dialog:

  Select a drawing type from the dropdown list and determine the import file. Confirm with "OK".

  In the Convert DGN drawing to engineering objects dialog that follows, you can specify the unit conversion and stipulate whether the drawing is to be imported into the engineering data and placed on the P&I diagram, or whether only matching base objects are to be generated (checkbox "Create base objects only" has been activated):

  You find additional information on this topic in the "Interfaces" manual, keyword "Import DGN drawings".

- "Connect automatically":

  If two engineering objects are connected in the Navigator by means of a pipe but the connection does not exist any longer in the report, then this command automatically draws in the connection (the pipe) on the report as well.

- "Place template":

  You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Place template".

- "Check":

  Compares the report objects and `DocObjs` of a report in order to identify any inconsistent references. (This refers to cases in which the `DocObj` still exists but the report object has been deleted.).

- "Inconsistency > ...":

  "...> Show previous", "... > Show next", "... > Analyze", "... > Delete all inconsistent objects"

  Analyzes the diagram and verifies whether there are any inconsistencies (missing report objects, wrong direction of flow, etc.). You find additional information on this topic in the "Inheriting, Copying, Deleting" manual, keyword "Inconsistent objects".

Context menu of a selected object

Varies according to the number and type of the selected objects.

- Multiple objects selected: "Grouping > ...":
  - "... > Create", "... > Cancel", "... > Restore", "... > Remove from group": self-explanatory
- Pipe and device selected: "Connection > ..."

  "... Delete": Deletes the selected pipe in the report and in the Navigator, and deletes the connection made by the COMOS connectors.

  "... Options" and "... Settings": Access to the pipe context menu commands of the same name.

- Pipe and a device selected: "Device > ..."

  "... Delete": Deletes the selected device in the report and in the Navigator, and deletes the connection made by the COMOS connectors.

  "... Options", "...> Settings", and "...> Graphical settings": Access to the device context menu commands of the same name.

- "Options > ...":
  - "... > Delete free connectors": self-explanatory
  - Various menu items for the editing and mirroring of the symbol and to restore the original symbol.
  - "... > Lock":

    The symbol cannot be changed via the report until it has been released again via "Options > Release". (No move or delete, many commands are deactivated.)
  - "... > Graphical properties":

    Graphical properties of the symbol such as color, line type, and line thickness can be set here.
- "Settings > ...":
  - "... Fix" and "... Search text": You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Settings".
  - "... Connector labels visible"
- "Graphical settings":

  This context menu is used to set attributes of the selected object, with their form being displayed by means of the symbol. Example: The outer wall of a vessel.

  Exactly which attributes are offered here has already been determined in the system to some extent, but can also be customized by the administrator (using the `AddToGraficalParameter` script option). See also section Extending the graphical settings (Page 123).

14.14 Data flow of positions

All attributes on the following tabs of a position are statically linked to the corresponding attributes of the subunit:

- "BAS22 Security requirements"
- IC030 E/I&C options
- "IC020 Environmental conditions"

Their data is required later for selecting a signal. When a position is created, all static links are automatically updated once by COMOS PT.

The attributes on these tabs are relevant in I&C planning. They are almost entirely for informative purposes and do not initiate any automated mechanisms.

You find additional information on this topic in the "E&IC" manual, keyword "Tab attributes/Position attributes".

"E/I&C options" tab: Attributes "EMS102 H,L" and "EMS103 +,-"

You can only select one of the two attributes. The attribute shows the user whether the maximum and minimum level are to be input in the label with "H, L" or "+, -".

14.15 Structure of the base objects of functions

In the COMOS DB, you find functions below the base object "@02 > 030 Functions".

Measurement and actuating functions

Objects for measurement functions and actuating functions have been predefined under this node.

- Objects for measurement functions are located under "@02 > 030 > 00 > 01 Measurement functions".
- Objects for actuating functions are located under "@02 > 030 > 00 > 02 Actuating functions".

You find additional information on this topic in the "E&IC" manual, keyword "Base objects of functions".

14.15.1 "@02 > 030 > 00 Functions, general"

Placeholders

When the user creates functions in the engineering project on level A1, these are "placeholders" of the "measurement function" or "actuating function" function type. A detailed specification of the function has not yet been drawn up or will not be drawn up. These functions are described as neutral or general functions. They are replaced by more concrete functions in I&C planning.

Function code

The functions of this level have no function code. The base object does not change when you enter function codes.

Process connector

Connecting a wildcard function on a report to the process does not create a process connector.

Creating reports

Various evaluating or interactive reports can be created underneath functions on this and the following level. Reports are not needed until I&C planning. You find additional information on this topic in the "E&IC" manual, keyword "Report".

14.15.2 "@02 > 030 > 00 > 01 Measurement functions" and "@02 > 030 > 00 > 02 Actuating functions"

Functions already have a function type from this level onwards. See also section "SYS System" tab (Page 263).

"@02 > 030 > 00 > 01 Measurement functions"

If you are working with the unit structure according to ANSI, the "Local measuring", "Control and monitor centrally", and "Controlling and monitoring at local or other point" measurement functions will be available for selection from the "New" context menu for a position in the Navigator. The are located in the base data, below the following node:

"@03 > BAS > ANSI > @F > 03 > 01 Structure elements measurement functions ANSI"

A measurement function with this base object does not initially have a function code. If it is placed on the diagram and connected to the process, it does not get a process connector or any new tabs, as is the case with the measurement functions of the levels below it.

If you input a function code, a base object change takes place from this level onwards. If the function had been connected with the process before the function code was input, it furthermore gets a process connector afterwards. See also section Creating a process connector (Page 75).

The objects in branch "@03 > BAS > IEC > @F Functions" correspond to DIN 19227.

On this level of the unit structure according to KKS you can find a structure that improves the bundling of the underlying measurement functions.

"02 Actuating function"

This base object is a structure below which the actually creatable actuating functions are collected.

In the COMOS DB, there are numerous structures under the functions from this level onwards. These structures are then made available in the engineering view via the "New" context menu. However, they are not required until the I&C planning stage and hence are covered in the I&C section. You find additional information on this topic in the "E&IC" manual, keyword "@F Functions".

14.15.3 Levels underneath "@02 > 030 > 00 > 01 Measurement functions"

The concrete functions, which are a prerequisite for the actual I&C planning, start at this level.

Measurement function

Measurement functions have a function code. Base objects for measurement functions have been predefined without a process coupling, with one process coupling or with two process couplings. See also section Number of process connectors (Page 81).

14.15.4 "@02 > 030 > 00 > SE Structure elements for functions"

"@02 > 030 > 00 > SE > 97 (Sub)functions"

The functions created here inherit from the functions created under "@02 > 030 > 00 > 01 Measurement functions". They are elements of the created positions (via their "Elements" tab). Consequence: They are offered for selection in the engineering view via the "New" context menu of the position.

See also section "@02 > 030 > 00 > 01 Measurement functions" and "@02 > 030 > 00 > 02 Actuating functions" (Page 224).

"@02 > 030 > 00 > SE > 98 Functional elements"

You find ANSI, EN, and KKS functional elements underneath this node.

"@02 > 030 > 00 > SE > 99 Structure elements template"

You find ANSI, EN, and KKS template structure elements underneath this node. The actuating functions created here can be used as the base object for the modules for functions that are created in the base project.

14.15.5 "@10 @Y Attributes catalog > BAS @Y General attributes"

Most of the attribute tabs and most of the attributes of functions are contained within this node. However, some tabs and attributes of functions are created in the base object nodes EIC and PID.

In addition: The E/I&C options tab and its attributes, which are found in units and positions, originally derive from this base object node.

More details in the sections on the relevant tabs.

14.15.6 Scripts

The following scripts have been input at the base object of a function on the "Script" tab:

"`Connect`"

Call: when joining the connectors, thus, for example, when connecting the function on the P&ID.

Task:

Measurement function: `Connect` only calls function `SetCDev` from `UserScriptBlock1`. If a function is connected to a process device (pipe, device), the process connector is generated and connected dynamically and automatically. Therefore, in this case, `Connect` is also called.

Actuating function: `Connect` calls `ConnectedObjectAsChild`. An "I" connector or a "W" connector is created at the function, depending on the context.

See also section Connectors (Page 78).

"DisConnect"

Call: When disconnecting two connectors. If you reduce the number of process connectors, the connectors are separated as well. Thus `DisConnect` is called as well. Only with measurement functions.

When the call is made, the "SYS.AutoDeleteFreeConnectors" attribute (type: Boolean, i.e. checkbox) is queried on reports at measurement functions. If the attribute exists and is set to "1", measurement function connectors which are not connected or are free are deleted. If you wish to use the attribute, you must create it yourself.

Task:

`DisConnect`calls function `SetCDev` from `UserScriptBlock1`.

Comment in line 4: `Workset.Lib.Elo.DelChapterByConIndex "BAS020",Connector`

Safety prompt for the case in which measuring and actuating functions have been mixed (these are separate branches in the COMOS DB). "BAS020" is the substance data of the actuating function. This tab is not required for the measurement function and can be dispensed with.

"OnEditOk"

Call: Whenever "OK" or "Apply" is pressed in the properties window. Only with measurement functions.

Task:

Measurement function: `OnEditOk` only calls function `SetCDev` from `UserScriptBlock1`.

"OnReferencedByDevice"

Call: When creating the function and on every change of base object.

Task:

Measurement function: When the function is created, it takes the first letter of the label of the position and thus becomes a suitable measurement function. If a mask is defined for the label at the base object, the mask wins out.

After that, `SetCDev` is called from `UserScriptBlock1`. The base data can be simplified in this way: The same neutral functions are always located below the positions and the appropriate measurement function is created as a result of automatically taking over the label (and the subsequent change of base object in `UserScriptBlock1`).

Actuating function: Same as the measurement function, but here the actuating function takes the first letter from the label of the position and appends a "`V`".

"UserScriptBlock1"

This script block sets script variables and implements the `SetCDevice` function. The script variables do not necessarily have to be located in `UserScriptBlock1`.

1. `PositionCDevFullName`

   Automatically generates a position below the report if a function is placed on the P&ID report from the base data. The measurement function lies below the position. In order to put the mechanism into effect following prerequisites have to be met:

   – This function needs to have a valid function letter in the label. If you use the base object of a very general function in the COMOS DB, it will not have any function letters. Function letters are not added to the labels until lower levels. If you use a base object with function letters (and the other prerequisites are met), then the position is generated at the time the engineering object is created. If this is a general object you have to enter a function label on the engineering side. A base object change (pressure, temperature, etc.) takes place in response to the new function letter and the position is created.

   – The function has to have the global variable `PositionCDevFullName` set in the script. Example: `PositionCDevFullName = "@P|02"`.

     It does not matter in which script block this global variable was set. The global variable states which position is used.

2. `PositionCDevice`

   Defines which base object the owner position of the function gets. See also section Automatically (Page 58).

3. `SetCDevice`:

   Only for measurement function:

   – Call:

     If the label is changed. The label entered is automatically written to the "ICF100 Function" attribute. This attribute is queried in `SetCDev`.

     If the number of process connectors has been changed. The process connectors are controlled by the `CProcess` number of process connectors attribute.

   – Task: `SetCDevice` checks whether the base object needs to change and determines the new base object.

     The variable `BaseCDevFullName` identifies the base object node under which the search for the function base object should be carried out.

     `SetCDevice` then passes the label and the number of process connectors to the `SetCDevBySpecs` function. `SetCDeviceBySpecs` evaluates these parameters and changes the base object if necessary:

     First level below the node specified by `BaseCDevFullName`: depending on the function code.

     Second level: depending on the number of process connectors.

"**UserScriptBlock2**"

AddToGraficalParameter: Adds the following attributes to the functions context menu:
ICF101 Processing with
ICF102 Output and operation

14.16 Changing color settings globally

14.16.1 Overview

The option to change color settings globally gives the user the opportunity to temporarily alter the color settings of the symbols of objects placed on the report. To do this, COMOS makes reference to predefined rules which are stored in one or a number of queries.

14.16.2 Queries

"@System > @D > @ColorQuerySets > RI2 Color set objects character type"

By default, three queries for changing color settings globally are predefined in the COMOS DB. The queries are found in the base object, in the Navigator on the "Base objects" tab, under the "@System > @D > @ColorQuerySets > RI2 Color set objects character type" node.

Base object

Each color query has a base object query "request" as an owner. Each color query is based on an engineering object query to which a color has been assigned. You can customize the color queries so that the color of the symbols only changes under certain conditions.

14.16.3 "Select color" button

"Attributes" tab for query

The "Select color" button appears on the "Attributes" tab in the query properties. Use this button to assign a color to the query.

The following properties must be assigned to the button:

Name	Description	Display type	Type
Color	Select color	Button	Number

14.16.4 Activating "UseColorSets" mode

Introduction

When `UseColorSets` mode is activated, the user can make temporary color settings for symbols via the context menu of the report.

Procedure

To activate `UseColorSets` mode, proceed as follows:

1. Open the report template for the report for which you wish to activate the mode.
2. Right-click on the working area of the report template.

   The context menu opens.
3. Select the "Options" command from the context menu.

   The "Options" window opens.
4. Find the `UseColorSets` entry.
5. Change the value of `UseColorSets` mode from `False` to `True`.
6. Click "OK" to save your input and close the "Options" window.
7. Click "Save" in the report template.

Result

`UseColorSets` mode is activated for all reports which access the selected report template. The next time the user opens a corresponding report, the mode is available.

14.17 PDS 2D

14.17.1 Creating a DB choice object

Procedure

To create a DB choice object, proceed as follows:

1. If required: Import a DB choice object into your database. Perform a NOXIE import or import a working layer.
2. Click on the "Base objects" tab in the Navigator.
3. Create a base object under the following node that inherits from the DB choice object, or import the DB choice object directly under the node:

   "Import > @DGN base objects for external data import"

4. In the properties of the DB choice object, enter a name and a description in the "General" control group of the "System" tab.
5. Save your entries.

14.17.2 Managing links of the P&IDs to DB choice objects in the base data

Procedure

There are two ways to manage the links of the P&IDs to DB choice objects in the base data:

- If you only use one P&ID template and one P&ID report base object:
  - Link the P&ID report base object to a DB choice object.
  - Configure the P&ID report base object in such a way that the user can overwrite the default DB choice link.
- If you use multiple P&ID templates and P&ID base objects:
  - Link the P&ID report base object to a DB choice object.
  - Configure the P&ID report base object in such a way that the user cannot overwrite the default database choice link.

See also

Linking the P&ID to the DB choice object (Page 159)

14.17.3 Configuring a DB choice object

Procedure

To define the import settings for the import from a PDS 2D database, proceed as follows:

1. Open the properties of the DB choice object.
2. Configure the "Attributes > PDS import settings" tab.
3. Configure the "Attributes > PDS unit mapping" tab.

Import settings

Make the following settings:

- Enter information regarding the PDS 2D database. COMOS needs this information for the import.

  For example:

  – Full path to the PDS 2D database
  – Table prefix
  – The code list directory
  – …
- Specify the to be used P&ID base objects as the standard for the following objects:
  – Pipes
  – Pipe branches
  – Pipe segments
  – Nozzles
  – Positions
- Specify the name of the XML file which fully logs the import.
- If you are working with script variants: Enter the name of a script variant. This script variant runs in the import scripts of the P&ID base objects during import.
- You can define a deviation from the default import behavior for functions and control valves: Determine whether each function and each control valve is to be imported under its own separate position.
- Define the unit mapping between COMOS and PDS 2D.

See also

Attributes of the DB choice object (Page 268)

14.17.4 Adjusting the base data structure to the DB choice structure

Initial situation

If you performed DGN imports with a COMOS version prior to COMOS 9.0 in a COMOS project, you did not need a DB choice object.

To run a PDS 2D import in such a project, it is necessary that you adapt the base data structure to the DB choice structure.

Procedure

To adapt the base data structure, proceed as follows:

1. Create a DB choice object as described above.
2. Open the properties of the DB choice object.
3. Switch to the "Attributes > PDS import settings" tab.
4. Click the "Move base objects" button.

Result

COMOS moves all base objects located under "Import > @DGN" to the DB choice object.

14.17.5 Configuring DGN import objects

Introduction

COMOS converts as many import engineering objects as possible into P&ID objects during the PDS 2D import.

One place you can define how COMOS is to do this is in the import base objects.

Procedure

To configure the import base objects, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Open the node of the DB choice object that is linked with the P&ID, with which you imported the cell library.
3. Open the properties of an import base object, which is located under this DB choice object.
4. Switch to the "Attributes > Data from PDS database" tab.
5. Configure the control elements of the "Set at import base object" control group.
6. Repeat steps 3 and 5 for all import base objects under the DB choice object.

Import settings of the "Set at import base object" control group

Make the following settings:

- "Assigned P&ID base object" field:

  Set a reference to a P&ID base object.

  A base object change takes place during the PDS 2D import: All import engineering objects based on this import base object get the P&ID base object referenced here as a new base object. This converts them into P&ID engineering objects.

- "Delete object" option:

  Determine whether the DGN engineering object is to be deleted. If the DGN engineering object is deleted, it will not be further processed by the PDS import.

- "Delete local graphic" option:

  Determine whether the P&ID engineering object is to use the symbol of P&ID base object or the DGN cell.

  See also section Configuring the symbol display (Page 233).

Editing mode for users on the engineering side

Depending on the editing mode you set for the attributes of the import base objects, users can overwrite some of their default settings in the derived import engineering objects prior to a PDS 2D import.

Further information

You find a complete overview of all import options and their function in section Attributes of the "Data from the PDS database" tab (Page 270).

14.17.6 Configuring the symbol display

Introduction

If the user has selected the "Always import as local script" option in the DGN import settings, the graphic of the DGN cell is checked in as a local script at the import engineering object.

You can decide whether the P&ID engineering object generated by the PDS 2D import is to use the graphic of the DGN cell or that of the P&ID base object.

Procedure

To determine which graphic the P&ID engineering objects uses, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Open the node of the DB choice object linked to the P&ID you used to import the cell library.

3. Open the properties of an import base object located under this DB choice object.
4. Switch to the "Attributes > Data from PDS database" tab.
5. Control group "Set at import base object", option "Delete local graphic":
   – Enabled: The graphic stored at the P&ID base object is used. The local script which stores the image of the cell is deleted.
   – Disabled: The graphic of the DGN cell is used. It is stored at the P&ID engineering object as a local script.

14.17.7 Configuring P&ID base objects

Introduction

COMOS converts as many import engineering objects as possible into P&ID objects during the PDS 2D import. One place you can define how COMOS is to do this is in the P&ID base objects.

You define the following points in the P&ID base objects:

- The import behavior, if COMOS detects that objects with the same name already exist in the engineering data during the conversion from import objects into P&ID objects
- The import scripts that are executed during conversion

Procedure

To configure the P&ID base objects, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Open the "@01 > PID Piping and instrumentation" node.
3. Open the properties of a P&ID base object.
4. Switch to the "Script" tab.
5. Open one of the UserScriptBlocks.
6. Implement the following methods in the script editor:

   `PDSImport`

   `PDSImportDone`

   Optional: `PDSAfterMove`
7. Switch to the "Attributes > Data from PDS database" tab.
8. Configure the control elements of the "Set at COMOS base object" control group.
9. Repeat steps 3 to 8 for all P&ID base objects that are involved in the import.

"Set at COMOS base object" control group

In the "Set at COMOS base object" control group, you specify how COMOS should proceed if it finds an engineering object with the same name.

See also

Handling the configuration for already existing P&ID engineering objects (Page 235)

Implementing the import script (Page 238)

14.17.8 Handling the configuration for already existing P&ID engineering objects

Introduction

When import engineering objects are converted into P&ID engineering objects, they are assigned the name of the imported cell.

The general COMOS rule applies here: Objects that are located on the same level in the tree structure of the Navigator cannot have the same name.

Your settings determine how COMOS proceeds if it detects during the conversion that there already is an object with the same name in the engineering data.

Procedure

To define how the import behaves if COMOS finds an object with the same name in the engineering data, proceed as follows for each P&ID base object that is involved in the import:

1. Open the properties of the P&ID base object.
2. Switch to the "Attributes > Data from PDS database" tab.
3. Configure the control elements of the "Set at COMOS base object" control group.

Import settings of the "Set at COMOS base object" control group

Select one of the following values from the "If object available" list:

Value	Result
"Create new object"	• The base object of the import engineering object is changed as specified in the "Assigned P&ID base object" field in the properties of the import engineering object. The import engineering object is thereby transformed into a P&ID engineering object. • The name of the P&ID engineering object is incremented by a number. • A log entry is created. • The existing P&ID engineering object remains unchanged.
"Keep only name and description"	• There is no base object change at the import engineering object. The import engineering object is deleted. • Instead, the existing P&ID engineering object is used. • The name and description of the P&ID engineering object is retained. • Which graphic is used depends on how the "Delete local graphic" option is set on the Attributes > Data from PDS database" tab at the import engineering object: – Enabled: The local graphic that was stored at the import engineering object is deleted. Instead, the graphic of the base object that is linked to the P&ID engineering object is used. Depending on how the "Exchange base object of existing object" option was configured at the import engineering object, this is either the original base object of the P&ID engineering object or the P&ID base object that was referenced at the import engineering object. – Disabled: The local graphic of the import engineering object is checked in and used at the existing P&ID engineering object as a local graphic. • The PDS 2D data of the import engineering object is written to the P&ID engineering object. The script stored at the base object of the P&ID engineering object is used for mapping between PDS 2D data and COMOS attributes. Depending on how the "Exchange base object of existing object" option was configured at the import engineering object, this is either the original base object of the P&ID engineering object or the P&ID base object that was referenced at the import engineering object. • No log entry is created.

Value	Result
"Connect to imported graphic"	• There is no base object change at the import engineering object. The import engineering object is deleted. • Instead, the existing P&ID engineering object is used. • The name and description of the P&ID engineering object are overwritten. • The graphics of the import engineering object is checked in as a local script at the P&ID object and used, even if "Delete local graphic" is enabled. • The PDS 2D data of the import engineering object is written to the P&ID engineering object. The import script stored at the base object of the P&ID engineering object is used for mapping between PDS data and COMOS attributes. Depending on how the "Exchange base object of existing object" option was configured at the import engineering object, this is either the original base object of the P&ID engineering object or the P&ID base object that was referenced at the import engineering object. • Deviations from the previously at the P&ID object stored values are logged in the XML file.
"Don't search for available object"	• The base object of the import engineering object is changed as specified in the "Assigned P&ID base object" field in the properties of the import engineering object. The import engineering object is thereby transformed into a P&ID engineering object. • The name of the DGN cell is not used as the name. Instead, COMOS generates the name from the name mask defined in the P&ID base object. • Unlike the rest of the settings, there is no check for duplicate names. The PDS 2D import is therefore faster.

- "Exchange base object of existing object" option:

  This option is only evaluated if one of the following values is set in the "If object exists" list:

  – "Keep only name and description"

  – "Connect to imported graphic"

Value	Description
Enabled	The existing P&ID engineering object is used for the PDS 2D import, but its base object link is changed: The based object assignment under "Mapped P&ID base object" at the import engineering object is assigned as a new base object. Consequence: • Depending on how the "Delete local graphic" option is configured, this has an effect on which graphic is used. • The script of the new P&ID base object is used to import the PDS 2D data into the COMOS attributes.
Disabled	The existing P&ID engineering object is used for the PDS 2D import. There is no base object change.

14.17.9 Importing text cells as objects

Procedure with "Import empty cells"

If the "Import empty cells" option is enabled in the DGN import settings, import base cells are created during the DGN import for the pure text cells that are placed on the DGN drawing.

Similar to the other import base objects, you also link these import base objects to the appropriate P&ID base objects.

For pure text cells, select base objects for P&ID label symbols. Base objects for P&ID label symbols must have the following property: "System" tab, "General" control group, "Class" property: "Dataset" value

Label symbols are then generated during the PDS 2D import and are linked to the components in the database whose data they readout.

Procedure without "Import empty cells"

If the "Import empty cells" option is disabled in the DGN import settings, no import base cells are created during the DGN import for the pure text cells that are placed on the drawing.

With the PDS 2D import, label symbols can also be generated and linked to the components in the database whose data they readout.

Proceed as follows:

1. Manually create import base objects for the text cells.
2. Regard that the name of the base object is identical to the name of the cell.
3. Then link the base objects as described above for the base objects for P&ID label symbols.
4. Perform the PDS 2D import.

14.17.10 Implementing the import script

Overview

Implement the following methods in the `UserScriptBlock` of a P&ID base object:

- `PDSImport`
- `PDSImportDone`
- Optional: `PDSAfterMove`

The methods are called by COMOS when a DGN engineering object with a link to this P&ID base object is converted into a P&ID object.

The methods fulfill the following tasks:

- `PDSImport`:

  The method defines which value is to be entered in the "Name" property of the P&ID object. Take over the value directly from a column of a PDS 2D table or process it before you write it in the name.

- `PDSImportDone`:

  The method defines the attribute mapping between the attributes of the P&ID engineering objects and the PDS 2D data.

- `PDSAfterMove`:

  If the object was sorted under a different P&ID object: The method implements a processing type that considers the new owner structure.

Procedure

Proceed as follows:

1. Open the properties of the P&ID base object, "Script" tab.
2. Open the script editor for one of the UserScriptBlocks.
3. Implement the three methods as described further below.
4. Repeat these steps for all P&ID base objects that are linked to DGN objects.

Using script variants

If you are importing from multiple PDS 2D databases, you can implement a separate script variant for each database.

For each database from which you want to import, proceed as follows:

1. Open the properties of the DB choice object in which you want to manage the import settings for the database.
2. Switch to the "PDS import settings" tab.
3. Type a name into the "Script variant" field.
4. For each P&ID base object that is linked to a DGN object, first proceed as described above. During the implementation of the methods, use a GetScriptVariant call for every method call to check which option is to be executed. Then insert a Select/Case statement for your variant, in which you define the implementation of the script variant.

Methods

Implement the methods listed in the table below in the UserScriptBlock. You can access additional methods and variables for the implementation of the methods via `Node` . See also section Variables and methods for writing the import script (Page 241).

Method	Parameter	Description
`PDSImport`	`Node` The COMOS object (type: Device) from where the method is called	Set the name of the P&ID engineering object via `Node.ReadData` and `Node.SetName`. This step is required for the COMOS-internal check to determine whether an engineering object with the same name already exists.
`PDSImport-Done`	`Node` See above.	Define the attribute mapping via `Node.ReadData` and `Node.WriteData`. If required: Edit the value imported via `ReadData` before calling `WriteData` – e.g. by running `SeparateUnitAndValue`. The method is called after `PDSImport`, but not until COMOS has checked whether an engineering object with the same name already exists in the engineering view. No import is performed for incompatible data types. A corresponding entry is created in the error log. All PDS 2D data is documented at the P&ID engineering object in the "PDSData.Data" list attribute, regardless of any existing assignment for this data: • "Name" column: Name of the PDS attribute • "Description" column: Description of the attribute in PDS • "Value" column: Value from the PDS 2D database
`PDSAfter-Move`	`Node` See above.	The method is only available at the following objects: • Vessel nozzles • Vessel installations • Measuring equipment • Pipes and pipe branches It is executed after the object has been moved under a new P&ID owner object. The method is only called for the sorted object. This method gives you the chance to implement the processing of the object which considers the new owner structure. Implementation optional.

Example script

```
Sub PDSImport(Node)
'Only apply if Device:
If ThisObj.SystemType = 8 Then
        strValue=node.ReadData (FieldName)
        node.SetName strValue
    End If
End Sub
```

```
Sub PDSImportDone(Node)
'Only apply if Device:
If ThisObj.SystemType = 8
        strValue = node.ReadData (FieldName)

        'optional: process strValue in any way that is required -
e.g. call
        'SeparateValueAndUnit

        node.WriteData strValue, SpecName, ComosUnit, PDSUnit, Index
    End If
End Sub

Sub PDSAfterMove(Node)
'Only apply if Device:
If ThisObj.SystemType = 8 Then
        strValue = node.ReadData (FieldName)
        node.WriteData strValue, SpecName, ComosUnit, PDSUnit, Index
    End If
End Sub
```

14.17.11 Variables and methods for writing the import script

Methods from `Node`

The following methods are available for implementing the `PDSImport`, `PDSImportDone` and `PDSAfterMove` methods via the `Node` parameter.

Method	Parameter	Description
`ReadData`	`FieldName as String`: The name of the PDS 2D field that is being processed	`ReadData` returns the value of the field addressed with `FieldName` in the form of a string. The value originates from the PDS 2D dataset referenced by `Node`.
`SetName`	`ObjectName as String`: The name that Node should get	`SetName` assigns the COMOS object the string passed by `Node` in `ObjectName` as a name.

Method	Parameter	Description
`WriteData`	• `strValue as String`: The value that writes `WriteData` to `Node` • `SpecName as String`: – The `ComosNestedName` of an attribute from `Node`, for example: "TD.ND0001" – Special case: The variables `ComosDescription` or `ComosLabel` • Optional: `ComosUnit as String`: Default: "" The target unit in COMOS Only required if there is no matching entry in the unit mapping of the DB choice object. • Optional: `PDSUnit as String`: Default "" The target unit COMOS searches for the appropriate entry in the unit mapping of the DB choice object • Optional: `Index as lo`ng: Default: -1 If `SpecName` addresses a list attribute: The index which indicates to which position in the `strValue` list it is written	`WriteData` writes the string passed in `strValue` to the attribute from `Node` addressed by `SpecName`. Special case: The following variables are used as `SpecName`: • `ComosDescription` Result: `WriteData` writes `strValue` to the description of `Node`. • `ComosLabel`: Result: `WriteData` writes `strValue` to the label of `Node`. The attribute is reset to the unit determined by `PDSUnit` or `ComosUnit` and `strValue` is converted accordingly. If both `PDSUnit` and `ComosUnit` were passed, `ComosUnit` takes priority.
`GetTableName`	/	`GetTableName` returns the name of the PDS 2D table as a string.
`GetScript-Variant`	/	`GetScriptVariant` evaluates which script variant is set at the DB choice object and returns it as a string.
`ComosDev`	/	`ComosDev` returns the COMOS object on which the script is currently running. Return value: `Device`

Method	Parameter	Description
`Separate-ValueAndUnit`	`strValue as String`: The method expects that first the value and then the unit are given in `strValue`.	`SeparateValueAndUnit` divides the string passed in `strValue` into three substrings and returns them as `Collection`: Item(1): The substring up to the first number (exclusive) Item(2): The numerical value, consisting of: • Digits • If available: Delimiter (dot or comma) • If available: Decimals The substring ends before the first character that is neither a number, period or comma, or with the second period or comma. Item(3): The remainder of the string.
`ReadData-FromPCP`	• `FieldName As String`: Name of the field in the PCP table that is being processed • `ConnectorNumber As Long`: The ID of the connector ("PCP_NO" field)	`ReadDataFromPCP` based on the passed connector, reads out a from forwarded connector ID a value from the PCP table of an object. Call this function only for objects which take data from the EQ_NOZZ, PIPING_COMP and INSTR_COMP tables. This usually involves the following objects: • Nozzles under the equipment • Valves in pipes • Measuring equipment Return value: `String`

See also

Implementing the import script (Page 238)

14.18 Configuration file for the DGN import

14.18.1 Configuration file "DGNImport1.xml"

Path to the configuration file

The standard configuration file "DGNImport.xml" can be found in the Comos installation directory on the following path:

"<COMOS installation directory>\Current\Bin\OCX\WSP"

If you have deleted or renamed the configuration file, it is created automatically. Click on the "Save" button in the "Options" window but do not make an entry in the "Configuration file" field.

See also

"Options" window (Page 279)

14.18.2 Assigning a DGN line type to a COMOS line type

Introduction

Assign the DGN line types to the line types available for selection in COMOS on the "Line types" tab of the "Options" window. You can usually use the operating system line types you selected on the interactive report (using the context menu command "Options > Graphical properties).

Requirement

The "Options" window is open.

Procedure

To assign a DGN line type to a COMOS line type, proceed as follows:

1. Select the required line type from the "COMOS line types" table.
2. Select the required DGN line type from the "Used DGN line types" table.

   Repeat steps one and two for each DGN line type to which you wish to assign a COMOS line type. Multiple selection is not possible.
3. To assign the selected line types to one another, click the "<<Assign" button.
4. Select one of the following options:
   - Click "Save" to save the settings in the existing configuration file.
   - Click "Save As" to save the settings in a new configuration file. Select the required directory. Enter a file name. Click "Save".

Result

Both line types are assigned to each other and added to the "Assigned line types" list. The settings are written to the configuration file and saved as the default settings for DGN imports.

14.18.3 Assigning a DGN layer to a COMOS layer

Introduction

Assign the DGN layers to the COMOS layers available in the drawing type currently in use on the "Layers" tab of the "Options" window.

Requirement

The "Options" window is open.

Procedure

To assign a DGN layer to a COMOS layer, proceed as follows:

1. Select one or more DGN layers from the "Used DGN layers" table.
2. Select the required COMOS layer from the "Layers for the current symbol type" table.
3. To assign the selected layers to one another, click the "<<Assign" button.
4. Select one of the following options:
   - Click "Save" to save the settings in the existing configuration file.
   - Click "Save As" to save the settings in a new configuration file. Select the required directory. Enter a file name. Click "Save".

Result

The layers are assigned to each other and added to the "Assigned layers" list. The settings are written to the configuration file and saved as the default settings for DGN imports.

14.18.4 Assigning a DGN font to a Windows font

Introduction

Assign the DGN fonts to the selected Windows fonts on the "Fonts" tab of the "Options" window

Requirement

The "Options" window is open.

Procedure

To assign a DGN font to a Windows font, proceed as follows:

1. Click on the "..." button next to the "Path to the font resource files:" field.

   The "Search folder" window opens.
2. Select the directory in which the DGN fonts are saved.
3. Click "OK".

   You return to the "Fonts" tab. The DGN fonts in the selected directory are listed in the table on the right-hand side.
4. Click "..." next to the mapping table.

   The "Font" window opens.
5. Select the required font.
6. If applicable, select the required font mode and font size.
7. Click "OK".

   You return to the "Fonts" tab.
8. Select one or more DGN fonts from the table on the right-hand side.
9. To assign the selected DGN fonts to the selected Windows font, click the "<<Assign" button.
10. Select one of the following options:
    - Click "Save" to save the settings in the existing configuration file.
    - Click "Save As" to save the settings in a new configuration file. Select the required directory. Enter a file name. Click "Save".

Result

The fonts are assigned to one another and displayed in the mapping table. The settings are written to the configuration file and saved as the default settings for DGN imports.

14.18.5 Undoing an assignment

Requirement

You have made assignments in the "Options" window.

Procedure

To undo the assignment of the entries in the mapping table, proceed as follows:

1. Select one or more entries in the mapping table.
2. Click on the ">> Detach" button.

Result

The selected entries are removed from the assignment list.

14.19 Pipe Easy

14.19.1 Adding the "Display PipeEasy conversion dialog" button to the toolbar

Introduction

The  "Display PipeEasy conversion dialog" button is not displayed on the report toolbar by default. Use a script to add the button to the toolbar.

Requirement

The "std_PipeEasy" icon is available in the pictures library.

Procedure

To add the "Display PipeEasy conversion dialog" button to the toolbar, proceed as follows:

1. Open the report template for the report containing the toolbar you wish to add the button to.
2. Right-click in the white area of the report template.
3. Select the "Options" command from the context menu.

   The "Options" window opens.
4. Enter the following script at the end of the '`**PipeEasy Script` section:

   ```
   UIObject.AddMenu

   UIObject.AddMenu "ComosImportUtilities.ImportUtilities:ShowSetupForm", "",
   "std_PipeEasy", MenuText
   ```
5. Click "OK" to save your input and close the "Options" window.

Result

The "Display PipeEasy conversion dialog" button is added to the report toolbar.

14.19.2 Tabs in the "Conversion settings" window

Overview

The "Conversion settings window comprises four tabs. You can hide tabs and define standard actions for text rules in the "Options" window for the report template.

Example

Executing the following script causes only the "Assign texts" tab to be displayed and the "Use rules" action to be started:

```
UIObject.AddMenu
"ComosImportUtilities.ImportUtilities:ShowSetupForm",
"DisabledTabs:0;2;3|DEFAULTACTION:1", "std_PipeEasy",
"Show conversion form"
```

14.19.3 Importing data

Create graphical data on the report for test purposes.

See also

DGN import (Page 160)

14.19.4 Data management and analysis

Different rights are required, depending on which function is selected.

Some functions execute user-defined scripts. To write a script, you need good knowledge of VB script.

14.19.5 Creating base objects for text rules

Introduction

First create an action base object under which the base objects for text rules are collected. Underneath this, create the number of base objects you require, with scripts describing texts.

Requirement

The base project is open.

Creating an action base object

To create an action base object for text rules, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. To do this, right-click on the node below which you wish to create the base object and select the "New > New base object" command from the context menu.

   The properties of the new object open.
3. Select "Action" from the "Class" list.
4. Select "Script" from the "Subclass" list.
5. Enter the desired name for the object, e.g. "T", in the "Name" field.
6. Enter the desired description for the object, e.g. "Text assignment rules", in the "Description" field.
7. Click "OK" to create the object and close the properties window.

Result

The action base object is created.

Creating base objects with scripts describing texts

To create a base object with scripts describing texts, proceed as follows:

1. Select the "Base objects" tab in the Navigator.
2. Right-click on the desired node of the action base object.
3. Select the "New > New base object" command from the context menu.

   The properties of the new object open.
4. Select "Action" from the "Class" list.
5. Select "Script" from the "Subclass" list.
6. Enter the desired name for the object in the "Name" field.

   Every base object must have the same name as its script function. Example: The base object "SetName" must have a script function called "SetName".
7. Enter the desired description for the object, e.g. "Rename object", in the "Description" field.

8. Add the required script to the "Script" tab.

   Make sure that the script function has exactly the same name as the base object.

   The script function must have a parameter array:

   – Params(1): Comos object of the symbol
   – Params(2): Text
   – Params(3): ReportObject
   – Params(4) ReportText (if selected) or Nothing (if name of an attribute is set)

9. Click "OK" to create the object and close the properties window.

Result

The base object is created. If the function is completed successfully, an empty string ("") is returned. If the function is not completed successfully, a text is returned describing the error.

Script example 1:

`ChangeOwner`: moves the object to unit UJ50

```
Function ChangeOwner(Params)
    ChangeOwner = ""

    Set Device = Params(1)
    Text= Params(2)

    ' Device->Document->DocFolder->Unit
    Set StartDev = Device.owner.owner.owner

    For i = 1 To Len(Text)
        s = Mid(Text, i, 1)

        If IsNumeric(s) Then
            s = Mid(Text, i)
            Set StartDev = StartDev.Devices.Item(s)
            If StartDev Is Nothing Then
                ChangeOwner = "Unit " & Text & " not found"
                Exit Function
            Else
                Exit For
            End If

        End If

        Set StartDev = StartDev.Devices.Item(s)
        If StartDev Is Nothing Then
            ChangeOwner = "Unit " & Text & " not found"
            Exit Function
        End If
    Next

    Device.OwnerCollection.Remove Device
```

```
    StartDev.Devices.Add Device
End Function
```

Script example 2:

`SetName`: Rename COMOS object

```
Function SetName(Params)
    SetName = ""
    Set Device = Params(1)
    Text= Params(2)

    Device.Name = Text
    If Device.Check <> 0 Then
      SetName = "Invalid name"
    End If
End Function
```

14.20 Symbols

14.20.1 Setting the size of connectors for P&ID objects

Introduction

COMOS offers the possibility of adapting the size of the connectors for P&ID objects to your needs.

Requirement

An interactive report has been created.

Procedure

To specify the connectors of the P&ID objects, follow these steps:

1. Select the command "Report template > Open report template" in the context menu of the interactive report in the Navigator.

   The report template opens.

2. Select the "Options" command from the context menu of the report template's working area.

   The "Options" window opens.

3. Add a line at the end of the script, in which you assign the desired value to the `ConnectorRadius` property.

   Example: `ConnectorRadius = 5`

4. Close the dialog box by clicking "OK" and save the report template.

14.20.2 Standard pipe flags

"Symbols" tab

Define the standard pipe flag in the text script on the "Symbols" tab for the base object of the pipe segment.

COMOSDB

In the COMOS DB, the pipe flag outputs the following information:

- Name of the corresponding pipe
- Nominal diameter
- Nominal pressure
- Pipe spec

Report template

If, in the options script of the report template, `SetPipeFlagOnCreate` is set to `TRUE`, it is evaluated and displayed automatically when a pipe is drawn in.

The options script of the report template provides a number of options for controlling the behavior of the standard pipe flag. See also section Report template script options (Page 219).

Additional pipe flags

Objects for other pipe flags have been predefined under "@01 > PID > 99 > 01 P&ID labeling symbols" in the COMOS DB. See also section Base object "@01 > PID > 99 > 01 P&ID labeling symbols" (Page 185).

14.20.3 Symbols: End symbols

Display of PipeEndSymbols:

Scenario 1

Evaluation of "PIA602"

- "SYS.PIA602" with Value <> 1

  Meaning: The pipe is connected to a non-segmenting component. This corresponds to the "Branch separative" or "Pipe separative" behavior.

  Behavior: The end symbol script is evaluated.

- "SYS.PIA602" with Value = 1

  Meaning: The pipe is connected to a segmenting component.

  Behavior: The end symbol script is not considered.

Scenario 2

Attribute "PIA602" not available

- The old state is queried due to compatibility reasons.

  Old state:

  If the pipe is connected to a component which is not a PipeFitting (Dev.RIClass <> RIClassFitting), the end symbol script is evaluated.

See also

Showing and hiding additional pipe symbols (Page 42)

14.21 Mounting a rotation correction for PFD symbols

Procedure

To mount a rotation correction for PFD symbols when opening a P&ID document, proceed as follows:

1. Click the "Extra > Object debugger" menu.

   The object debugger opens.

2. Drag&drop the project from the Navigator into the "Object A" field.

3. Enter the following line in the "Script" field:

   ```
   A.workset.globals.CorrectRIRotation
   ```

4. Click on the "Execute" button.

Result

When you open a P&ID document, rotation correction is applied for PFD symbols. Once you terminate COMOS, this setting is not applied the next time you start the program.

14.22 Mounting a scaling correction for PFD/P&ID symbols

Procedure

To mount a scaling correction for PFD/P&ID symbols when loading a P&ID symbol script, proceed as follows:

1. Click the "Extra > Object debugger" menu.

   The object debugger opens.
2. Drag&drop the project from the Navigator into the "Object A" field.
3. Enter the following line in the "Script" field:

   ```
   A.workset.globals.RiConversionScale = 0.8
   ```
4. Click on the "Execute" button.

Result

When you load a P&ID symbol script, scaling correction is applied for PFD/P&ID symbols. Once you terminate COMOS, this setting is not applied the next time you start the program.

14.23 Search functions for P&ID reports

14.23.1 GetDeviceConnectedToPipe

Application

The `GetDeviceConnectedToPipe(ByVal Pipe As IComosDDevice, ByVal StrConnectorName As String, ByVal SearchMode As Integer) As IComosDDevice` function gets a component that is connected to the specified connector of a pipe object.

The following table shows the meaning of the parameters used:

Parameter	Meaning
`Pipe`	Pipe object for which the search is conducted.
`StrConnectorName`	Specifies whether to search for a connected component at the input or output of a pipe object. I1: Input of the pipe object O1: Output of the pipe object
`SearchMode`	Number of the desired search mode. COMOS provides three search modes.

See also

Search modes (Page 256)

14.23.2 GetConnectorOfDeviceConnectedToPipe

Application

The `GetConnectorOfDeviceConnectedToPipe (ByVal Pipe As IComosDDevice, ByVal StrConnectorName As String, ByVal SearchMode As Integer) As IComosDDevice` function gets the device connector, which is connected to the specified connector of a pipe object.

The following table shows the meaning of the parameters used:

Parameter	Meaning
`Pipe`	Pipe object for which the search is conducted.
`StrConnectorName`	Specifies whether to search for a connected component at the input or output of a pipe object. I1: Input of the pipe object O1: Output of the pipe object
`SearchMode`	Number of the desired search mode. COMOS provides three search modes.

See also

Search modes (Page 256)

14.23.3 Search modes

Meaning of the search modes

COMOS provides three search modes for the `GetDeviceConnectedToPipe` and `GetConnectorOfDeviceConnectedToPipe` search functions.

In all three search modes, COMOS determines which component or component connection is connected at input I1 or output O1 of a specified pipe object. You can restrict the search using attributes depending on the search mode. You cannot specify all pipe objects in every search mode.

The following table shows the pipe objects that can be specified in a given search mode and how you can restrict the search using attributes.

Search mode	Pipe	Pipe branch	Pipe segment
1	Yes	Yes	Yes
2	Yes[1]	No	No
3	Yes[1,2]	Yes[2]	Yes[2]

[1] The COMOS search only considers the pipe branches of the pipe, which are identified as a main branch. In the "Attributes > P&ID options" tab of the "Pipe information" control group, you designate the pipe branch as a main branch using the "Main pipe branch" attribute (attribute: RI.PEA935).

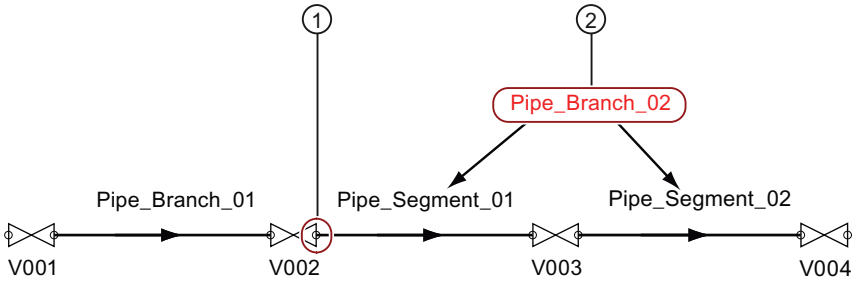
COMOS returns only those components or component connectors, which are connected to a main pipe branch.

[2] COMOS only returns those components or component connectors, for which the "Main equipment class" attribute is enabled. You enable the "Main equipment class" attribute (attribute: RI.PEA936) in the "Attributes > P&ID Options" tab.

Example

This example shows the two search functions for search mode 1.

The following graphic shows an example of the parameters that are specified for the search function.



① Parameter "StrConnectorName"
② Parameter "Pipe"

Function calls:

`GetDeviceConnectedToPipe(Pipe_Branch_02, "I1",1)` returns the component "V002" as the search result.

`GetConnectorOfDeviceConnectedToPipe(Pipe_Branch_02, "I1",1)` returns the component connector "O1" of the "V002" device as the search result.

See also

GetDeviceConnectedToPipe (Page 254)

GetConnectorOfDeviceConnectedToPipe (Page 255)

14.24 Configuring automatic assignment of line types

14.24.1 Basics of automatic assignment of line types

Using automatic line types

COMOS allows you to set the line type automatically when drawing a connection. This requires you to configure the following:

1. Specify the components to be used for which automatic line types.

   See also Specifying component types to be used (Page 257).

2. Specify the components between which automatic line types are to be created.

   See also Specifying line types (Page 258).

3. Add the "SYS.LineTypeClass" attribute in the attributes of the base objects of the components and functions.

   See also Adding the "SYS.LineTypeClass" attribute to the attributes of the base objects (Page 259).

See also

Using automatic assignment of line types (Page 24)

14.24.2 Specifying component types to be used

Procedure

To specify the component types to be used, follow these steps:

1. Select the "Assign line types" command in the "Plugins > Basic" menu.

   The "Assign line types" window opens.

2. Select the "Component types" tab.

3. Click the "Add component type" button.

   The "Add component type" window opens.
4. Enter the key of component or function, for example ARM.
5. Enter a description of the component or function in necessary.
6. Close the dialog box by clicking "OK".

14.24.3 Specifying line types

Requirement

The used component types are created.

Specifying the connection type

To specify the connection type, follow these steps:

1. Select the "Assign line types" command in the "Plugins > Basic" menu.

   The "Assign line types" window opens.
2. Select the "Connection types" tab.
3. Click the "Add connection type" button.

   The "Add connection type" window opens.
4. Enter the source and target object of the desired connection.
5. Close the dialog box by clicking "OK".

Specifying the line type of the connection

To specify the line type of the connection, follow these steps:

1. Select the desired connection type in the top table.
2. In the "Line settings" area, specify the line type, line width, line color and the level of connection.

Result

The line types are specified. COMOS generates the following standard tables:

- @SYSTEM > @LINETYPEMAPPING > @EQUIPMENTS
- @SYSTEM > @LINETYPEMAPPING > @FUNCTIONS
- @SYSTEM > @LINETYPEMAPPING > @LINETYPES

14.24.4 Adding the "SYS.LineTypeClass" attribute to the attributes of the base objects

Procedure

To add the "SYS.LineTypeClass" attribute to the attributes of the base objects, follow these steps:

1. Navigate in the base project to the base object of the P&ID object, for which you want to use automatic line types.
2. Open the base object properties.
3. Select the following tabs:
   - For components: "Attributes > System information"
   - For functions: "Attributes > System"
4. Select the "Design mode" command in the context menu of the tab.

   The design mode is activated.
5. Select the "New > Attribute" menu command in the context menu in the tab.
6. Enter "LineTypeClass" as the name.
7. Enter "Component type" as a description.
8. Link this attribute with the standard table:
   - For components:

     "@SYSTEM > @LINETYPEMAPPING > @EQUIPMENTS"
   - For measuring functions

     "@SYSTEM > @LINETYPEMAPPING > @FUNCTIONS"
9. Close the dialog box by clicking "OK".
10. Save the base object.

User interface reference

15

15.1 "Pipe spec mapping" window

The "Pipe spec mapping" window contains the following control elements:

Control element	Description
"Display" list	• Display all: Displays all P&ID objects. • Only implemented: Displays only P&ID objects already assigned to a component. • Only not implemented: Displays only P&ID objects not yet assigned to a component.
"Execute now" button	Available components are determined for all P&ID objects in the "Pipe spec mapping" area.
"Pipe spec mapping" area	Contains all P&ID objects of the current selection • Gray font color: P&ID object without implementation • Black font color: P&ID object with implementation. The currently assigned components are organized under the P&ID objects.
"Existing components" area	Contains all the implementations that are available for the P&ID objects of the current selection.
"Assign" button	Assigns the P&ID object to the selected components.
"Detach" button	Detaches a device from the P&ID object.

15.2 Options in the "Create connection" window

The following table describes the options in the "Create connection" window:

Option	Description
"Do not create connection"	If this option is activated, the pipe and the P&ID object appear to be connected on the report, but a connection is not created in the database.
"Create connection"	If this option is activated, the P&ID object and the pipe are joined directly by means of their connectors.
"Create nozzle"	If this option is activated, a nozzle is created. The nozzle is created in the Navigator, underneath the P&ID object. The pipe is connected with the nozzle. This option is only available if your administrator has prepared a nozzle for the P&ID object.

In the upper half of the dialog you determine how the pipe input is to be connected, and in the lower half how the output is to be connected.

15.3 Project properties

15.3.1 Control elements on the "Process engineering" tab

Overview

The following table describes the control elements on the "Process engineering" tab:

Control element	Description
"Base object for pipe" field	In this field, you define which base objects are used for pipes.
"Base object for pipe branch" field	In this field, you define which base objects are used for pipe branches and when drawing with the "Connection" tool.
"Name for implementation" field	In this field, you define the keyword used to identify implementations of a PFD report as belonging together.
"Disable automatic generation of COMOS connectors" option	This option enables and disables the automatic generation of connectors. If the option is activated, when a component is placed on a diagram, any connectors which have been defined in the component's symbol script but are not available in the component data are not generated. This means that when a component is placed, these connectors are neither generated on the data side nor displayed on the report. The option does not affect the measurement functions or the creation of dynamic connectors.
"Always generate a new name at unit assignment" option	If this option is activated, a new object name is always generated whenever you assign an object to a new unit.
"Company logo" field	Enter the path to the company logo to be used in the report in this field.
"Pipe spec" field	If you wish to use a pipe spec, enter the desired pipe spec in this field.
"Nominal diameter specification for the first connector (input)" field	Enter the nominal diameter of the pipe spec for the first input connector in this field.
"Nominal diameter specification for other connector (output)" field	Enter the nominal diameter of the pipe spec for all output connectors in this field.
"Only apply values linked through the GD tab" option	This option is only relevant for older databases in which the P&ID objects have a "GD" tab. Use this option to specify the method used to apply the attribute values for connected components.

15.4 Tabs and attributes of functions

15.4.1 "SYS System" tab

Control groups

The "SYS system" tab contains two control groups for attributes:

- "System settings - IC": These attributes are part of the preparation for I&C planning. You find additional information on this topic in the "E&IC" manual, keyword ""System" tab".
- "Function settings - PI": These attributes belong to the base data administration of P&ID planning.

Attribute "FunctionType Function type"

The attribute displays whether the function is a measurement function, an actuating function or a neutral function.

Query this attribute via `specs.FD. F100.value`.

If the user has left the field blank and this attribute does not have a value, a query returns `device.name`, i.e. the name of the engineering object.

Attribute "PCC Component for process coupling"

This attribute determines whether the function is created with an inline device, a nozzle, or a valve when connecting to an item of process equipment (a pipe, as a rule) on a P&ID diagram. In the report template, set a parameter for this purpose in the options script.

See also section Automatically creating process coupling (Page 81).

In the COMOS DB, the tab is originally located in the collection of tabs "@10 > BAS > Y > 1 > SYS > SYS system".

The attributes of the tab originate from the base object "@10 > BAS > Y > 1 > SYS > SYS system".

Attribute "DevOwnerByConnectorDisabled"

Components are moved automatically in the following cases:

- The component is located below the report.
- The component is connected to a pipe.
- The pipe is assigned to a unit.

The RILIB function `SetDeviceOwnerByConnector` provides the basis for the move function. This function assumes that the pipe, rather than the document, is to be the owner of a component. If you do not want to use this method for components, you can deactivate automatic assignment with the attribute "DevOwnerByConnectorDisabled". To do this, create the "DevOwnerByConnectorDisabled" attribute as a checkbox type on the "SYS system" tab and activate it.

15.4.2 "RI Presentation" tab

Control groups

This tab contains two control groups. The first controls attributes that are always displayed on the P&ID diagram, e.g. "Line thickness" or "Color".

The "Optional display" control group contains control elements which are only available for drawing type PID2, e.g. reports P+ID EN 10628.

"ICF200 Graphic for relevant settings"

If this option is activated, the following commands are available for selection under the "Graphical settings" command in the context menu on the report:

- "> Safety-relevant (graphic)":

  Sets the value of attribute "ICA110 Safety-relevant (graphic)" on the "RI Presentation" tab.

  Standard table "IC > Y > 0 > 02 > ICF103 Safety-relevant" is stored at the attribute. When you select an entry from this list, the "Safety-relevant (graphic)" attribute is displayed on the report.

- "> Calibration-proofed measurement":

  Use this command to set the value of the "BAS1113 Calibration-proofed measurement" attribute on the "General data" tab. It is stored with a base standard table from the IC branch and functions in the same way as "ICA110".

- "> Quality-/Environment-relevant":

  Sets the value of the "BAS1111 Quality-/Environment-relevant" attribute on the "General data" tab. It is stored with a base standard table from the IC branch and functions in the same way as "ICA110".

"ICF201 Alarm display:"

Standard table "IC > Y > 0 > 02 > ICF201 Alarm" is stored at the attribute. If this option is activated, the values entered on the "RI Presentation" tab in attributes "ICF202", "ICF202a", "ICF292a2", "ICF202b", "ICF202b1", and "ICF292b2" are mapped.

Edit attributes

You can edit these attributes via the function properties. If you activate the attributes for the additional graphics, you also activate the "Graphic for relevant settings" and "Alarm display" attributes.

Alternatively, you can also select the function on the P&ID diagram and make the setting via the context menu.

Position in the ComosDB

In the ComosDB, the tab is originally located in the collection of tabs "@10 > BAS > Y > 1 > RI > 01 > RI Presentation".

The attributes of the tab originate from the base object "@10 > BAS > Y > 1 > RI RI Presentation".

See also

Symbols: Function symbols (Page 69)

15.4.3 "IC110 Function data measuring point" and "IC110 Function data actuating point" tabs

These tabs are available for functions starting at base object level "@02 > 030 > 00 > 01 Measurement function" and "@02 > 030 > 00 >02 Actuating function".

The most important attributes of this tab are:

- "ICF100 Function": Identifies which function is involved. This attribute returns the DisplayValue or nothing, if no value has been set.
- "ICF102 Output and operation" and "ICF101 Process with": Saves the graphical settings of the symbol. Alternatively, you can also edit this attribute in the P&ID diagram via the "Graphical settings" context menu. See also section Display on reports (Page 72) and section "RI Presentation" tab (Page 264).
- "CProcess number of process connectors": Determines the number of "Substance data" and "Process connector" tabs. This attribute is only available for measurement functions. See also section Number of process connectors (Page 81).

Some of the attributes of the tab cannot be edited. They serve as background information.

Fewer attributes have been predefined in the actuating functions as for the measurement functions.

In the COMOS DB, the tab is originally located in the following collection of tabs: "@10 > BAS > Y > 1 > IC110/210 > IC110 > 02 > IC110 Function data measuring point"

and "@10 > BAS > Y > 1 > IC110/210 > IC110 > 03 > IC110 Function data actuating point"

The attributes of the tab originate from the base object "@10 > BAS > Y > 0 > 04 Function data".

15.4.4 "BAS021 Substance data" and "BAS031 Process connector" tabs

15.4.4.1 General

If the function is assigned a process connector, the "BAS021 Substance data" and "BAS031 Process connector" tabs are created. The number of process connectors controls how many tabs are created in each case.

"BAS021 Substance data" tab

Some attribute values on the "BAS021 Substance data" tab of the function are taken from the "PI010 Substance data" tab of the connected pipe branch.

The following path leads to the base object on the "BAS021 Substance data" tab: "@10 > BAS > Y > 1 > BAS020 > 01 > BAS020 Substance data".

The attributes originate from the base object "@10 > BAS > Y > 1 > BAS020 > 01 > BAS020 Substance data".

"BAS030 Process connector" tab

Some attribute values on the "BAS031 Process connector" tab of the function are take from the "PI030 Technical data" tab of the connected pipe branch.

In the ComosDB, the tab is originally located under the following node:

"@10 > BAS > Y > 1 > BAS030 > 02 > BAS030 Process connector data"

The attributes originate from the base object "@10 > BAS > 0 > BAS > 21 BAS1001-1050".

See also

Creating a process connector (Page 75)

Process coupling for measuring functions (Page 77)

Process coupling actuating function (Page 80)

15.4.4.2 Measuring function with process coupling nozzle

If a function with a nozzle as a process coupling is assigned a process connector, the following new tabs are created:

"@10 > BAS > Y > 1 > BAS020 > 01 > BAS021 Substance data 1"

"@10 > BAS > Y > 1 > BAS030 > 02 > BAS031 Process connector 1(+)"

The number of process connectors controls how many tabs the measurement function has for substance data and process connectors.

If there is more than one connector, the tabs are incremented.

15.4.4.3 Measuring function with process coupling inline device

Flow measurements have an inline device as a process coupling. If a flow measurement is assigned a process connector, then among other things the following new tabs are created:

- `Process connector P1(P+)`
- `Process connector P2(P-)`
- `Substance data 1`

The number of these tabs is fixed and cannot be increased, in contrast to that of the other measurement functions.

15.4.4.4 Actuating function

When they are created in the Navigator, actuating functions are assigned a process connector directly, and hence among other things the following tabs:

- 2-way valve

  "@03 > BAS > IEC > @F > 02 > 01 > BAS031 Process connector 1(+)"

  "@03 > BAS > IEC > @F > 02 > 01 > BAS032 Process connector 2 (-)"

- 3-way valve

  "@03 > BAS > IEC > @F > 02 > 01 > BAS031 Process connector 1(+)"

  "@03 > BAS > IEC > @F > 02 > 01 > BAS032 Process connector 2 (-)"

  "@03 > BAS > IEC > @F > 02 > 02 > BAS033 Process connector 3"

- 4-way valve

  "@03 > BAS > IEC > @F > 02 > 01 > BAS031 Process connector 1(+)"

  "@03 > BAS > IEC > @F > 02 > 01 > BAS032 Process connector 2 (-)"

  "@03 > BAS > IEC > @F > 02 > 02 > BAS033 Process connector 3"

  "@03 > BAS > IEC > @F > 02 > 03 > BAS034 Process connector 4"

"(+)" stands for the input and "(-)" for the output. The number of tabs cannot be increased.

15.4.5 "Process connection data" tab

Regardless of how many process connectors a function has, it only has one "Process connection data" tab.

Only actuating functions and measurement functions that have an inline device as a process coupling have this tab. In the case of actuating functions, the tab is created directly with the engineering object. In the case of measurement functions with an inline device, the tab is not created until they are assigned a process connector. See also section Creating a process connector (Page 75).

Data flow

The two attributes "ICP109 Operating temperature, valid" and "ICP110 Operating pressure, valid" obtain their value via a static link to the connected pipe branch.

All other values are entered by the user at the function.

Position in the ComosDB

The tab is originally located in the base data under "@10 > BAS > Y > 1 > IC110/210 > IC210 > 02 > IC220 Process connector data".

The attributes originate from "@10 > BAS > Y > 0 > 02 Process connector data".

15.5 PDS 2D

15.5.1 Attributes of the DB choice object

Content

This section lists all attributes that are available on the "PDS import settings" and "PDS unit mapping" tabs at the DB choice object.

"PDS import settings" tab

Name: "PDSSettings"

Description: "PDS import settings" tab

Attributes:

Name	Description	Function
"Cmd1"	"Create objects from rdb_items"	Obsolete Do not use
"CmdStart"	"Start import"	Button Starts the PDS import for all reports that are linked to the DB choice object.
"DefPos"	"Position"	Reference The base object that is used for generating positions.
"LFN"	"Log file name"	Field Name of the XML file that is stored in the user's local "Temp" directory and logs all imports, which run via the DB choice object.
"MoveDevs"	"Move base objects"	Button Moves all base objects that are located under the "Import > @DGN" node under the DB choice object.

Name	Description	Function
"MultiPosition"	"Create position for each loop item"	Option Disabled: Default behavior - COMOS checks under which position a function or control valve is located in PDS 2D and imports the function or control valve under a position with the same name. If there is no position with a same name, one is created. Enabled: Deviation from the default import behavior, each function and each control valve is imported under its own separate position. Application case: A position was assigned to multiple functions and control valves in PDS 2D. In COMOS, there is no way to correctly assign the functions and control values to each other.
"Nozzle"	"Nozzles"	Reference The P&ID base object used for generating nozzles.
"PathAttributes"	"'Attributes' file"	File selection The full path to the "Attributes" file of the PDS 2D database entered in "Rdb database".
"PathCodeLists"	"'Code lists' directory"	Field The full path to the "Code lists" directory of the PDS 2D database entered in "Rdb database".
"PathRdbItems"	"'RdbItems' file"	Field The full path to the "RdbItems" file of the PDS 2D database entered in "Rdb database".
"Pipe"	"Pipe"	Reference The P&ID base object used for generating pipes.
"PipeBranch"	"Section"	Reference The P&ID base object, which is used for generating pipe branches.
"PipeGroupField"	"Field name for grouping pipe runs"	Edit field The field name from the PDS 2D database that is used for grouping pipe runs.
"RdbDB"	"Rdb database"	File selection The full path to the PDS 2D database.
"ScriptVariant"	"Script variant"	Field The name of the script variant to be used for the import.
"Seg"	"Segment"	Reference The P&ID base object used for generating pipe segments.
"TablePrefix"	"Table prefix"	Field If the tables of the in the "Rdb database" entered PDS 2D database have a prefix: The prefix

"PDS unit mapping" tab

Name: "PDSUnitMapping"

Description: "PDS unit mapping"

Attributes:

Name	Description	Function
"UnitMap"	"Unit mapping"	List Table in which you map units between PDS 2D and COMOS.
"UnitMap.ComosUnit"	"Comos unit"	Column in which you enter the COMOS unit using the following notation: <Name of the unit group>.<Name of the unit>
"UnitMap.PDSUnit"	"PDS unit"	Column in which you enter the PDS 2D unit as defined in PDS 2D.

"Data from the PDS database" tab

Name: "PDSData"

Description: "Data from the PDS database"

This tab was added to the DB Choice object for inheritance reasons. It is inherited from the DB choice object to the underlying DGN base objects. It is not configured at the DB choice object.

15.5.2 Attributes of the "Data from the PDS database" tab

Attributes

Name: "PDSData"

Description: "Data from the PDS database"

Use: The tab is used at the following objects:

- The DGN import objects
- The P&ID engineering objects

Attributes:

Name	Description	Function
"Behavior"	"If object available"	List Set at P&ID set base object Controls how the import proceeds if an engineering object with the same name already exists.
"CPData"	"CP data"	List Imports the nominal diameters of the DGN connector points for the following objects: • Nozzles under the equipment • Valves in pipes • Measuring equipment
"CPData.No"	"Number"	The ID of the connector point
"CPData.NPD"	"Diameter"	The nominal diameter of the connector point
"Data"	/	List The table is filled automatically during the import in the engineering view. It logs all PDS data from the PDS 2D dataset referenced by the DGN object.
"Data.Desc"	"Description"	Description from PDS 2D
"Data.Name"	"Name"	Field name from PDS 2D
"Data.Value"	"Value"	Value from PDS 2D
"DelGraph"	"Delete local graphic"	Option Evaluated at the import engineering object Controls the symbol display.
"DelObject"	"Delete object"	Option Set at import base object.
"ExCDev"	"Replace base object of the existing object"	Option Set at P&ID set base object Controls how the import proceeds if an engineering object with the same name already exists.
"MappedBO"	"Assigned P&ID base object"	Reference Set at import base object Controls into which P&ID object the import object is converted.
"Pkey"	"Primary key"	Only in the Navigator Only for system internal use
"TabName"	"Table name"	Only in the Navigator Only for system internal use

See also

Configuring DGN import objects (Page 232)
Configuring the symbol display (Page 233)
Handling the configuration for already existing P&ID engineering objects (Page 235)

15.6 XMpLant

15.6.1 Attributes of the "XMpLant" tab

Overview of the attributes

The "XMpLant" tab has the following attributes:

Name	Description	Function	Values	Default value
"PlantItem-Node-Name" *	"PlantItem-NodeName"	List Defines as which PlantItem type the object is exported.	Values in the standard table "@SYSTEM > @XMpLant XMpLant configuration > XMPlantNode-Names"	"Equipment"
"Tag" *	"Tag"	Field Contains the component tag of the object.	Free text	The "Label" property of the object
"Specification" *	"Specification"	Field Contains a unique identifier across all projects.	Free text	Empty string
"Stock-number" *	"Stock-number"	Field Contains the stock number of the object.	Free text	Base object of the object, "SystemFull-Name" properties
"ComponentClass" *	"Component-Class"	Currently not supported.	Free text	Empty string
"ComponentType" *	"Component-Type"	List Specifies how the symbol definition is to be carried out.	Values of the "@SYSTEM > @XMpLant XMpLant configuration > XMpLantPlantItem-ComponentTypeAttri-butes" standard table	• ShapeCatalog-Item: "Explicit" • Outside the shape catalog: "Normal"
"ComponentName" *	"Component-Name"	Field Contains the ID of the shape catalog item referenced by the object.	Free text	The ID of the ShapeCatalog-Item: "XMC_<n>"
"Revision" *	"Revision"	Field Contains the revision state of the object.	Free text	Empty string

Name	Description	Function	Values	Default value
"Export-ListedAttri-butes"	"Create only attributes from the following list"	Option Specifies which P&ID object attributes from the subtabs on the "Attribute" tab are exported as generic attributes. Deactivated (default setting): All attributes whose "Value" is set are exported. Activated: Only attributes that are part of the "AttributesList" are exported.	0 = deactivated 1 = activated	0
"Attributes-List"	-	List Specify which attributes are exported as generic attributes.	The "NestedNames" of the attributes: <tab name>.<attribute name>	-
"Status" *	"Status"	Field Contains the working status of the object.	Values in the "@SYSTEM > @XMpLant XMpLant configuration > XMpLantPlantItem-StatusAttributes" standard table	"Current"

Attributes indicated by an asterisk ("*") are only described in English because this is the exact description used to define these attributes in ISO standard 15926.

15.6.2 Standard tables

Principle

For certain attributes in the PlantItem node, the XMpLant standard does not permit free text, but rather expects a specific, predefined series of values.

To prevent users from making incorrect entries, the attributes in the COMOS DB are stored with standard tables that only contain permitted values.

Since XMpLant expects the values in English, these standard tables are available in English only.

Standard table "PlantItemNodeName"

In the COMOS DB, you find the standard table in the following node:

"@SYSTEM > @XMpLant XMpLant Konfiguration > PlantItemNodeName"

The standard table has the following values:

Name	Description	Value 1	Comment
"<default>"	"<default>"	"<default>"	Default value: "Equipment"
"Component"	"Component"	"Component"	-
"Equipment"	"Equipment"	"Equipment"	-
"InstrumentLoop"	"InstrumentLoop"	"InstrumentLoop"	Always used for positions
"Nozzle"	"Nozzle"	"Nozzle"	-
"PipeConnector"	"PipeConnector"	"PipeConnector"	-
"PipeFlowArrow"	"PipeFlowArrow"	"PipeFlowArrow"	-
"PipingComponent"			Always used for pipes
"PipingNetworkSegment"	"PipingNetworkSegment"	"PipingNetworkSegment"	Always used for pipe branches
"PiningNetworkSystem"	"PiningNetworkSystem"	"PiningNetworkSystem"	-
"PlantArea"	"PlantArea"	"PlantArea"	-
"ProcessInstrument"	"ProcessInstrument"	"ProcessInstrument"	Always used for functions
"SignalConnectorSymbol"	"SignalConnectorSymbol"	"SignalConnectorSymbol"	-
"SignalLine"	"SignalLine"	"SignalLine"	Data lines and effect lines are detected by the software during export and are exported as PlantItems of the type "SingalLine".

Standard table "ComponentType"

In the COMOS DB, you find the standard table in the following node:

"@SYSTEM > @XMpLant XMpLant Konfiguration > ComponentType"

The standard table has the following values:

Name	Description	Value 1	Comment
"<default>"	"<default>"	"<default>"	Default value: "Normal"
"Explicit"	"Explicit"	"Explicit"	The symbol definition is done explicitly.
"Normal"	"Normal"	"Normal"	The symbol definition from the shape catalog is used.
"Parametric"	"Parametric"	"Parametric"	The symbol definition is parameterized.

Standard table "Status"

In the COMOS DB, you find the standard table in the following node:

"@SYSTEM > @XMpLant XMpLant Konfiguration > Status"

The standard table has the following values:

Name	Description	Value 1	Comment
"<default>"	"<default>"	"<default>"	Default value: "Current"
"Current"	"Current"	"Current"	-
"Deleted"	"Deleted"	"Deleted"	-
"Modified"	"Modified"	"Modified"	-
"New"	"New"	"New"	-

15.7 DGN

15.7.1 "Import and dissolve drawing" window

Overview

The following table describes the control elements of the "Import and dissolve drawing" window:

Control element	Description
"Drawing type" list	Select the desired drawing type from this list. You can import files in the following formats: • DGN files • PID files • DWG files • DXF files • XML files
"Create base objects only" option	If this option is activated, only base objects and no engineering objects are created. The setting of the option is stored but not evaluated at this point. For the DGN import you can also activate or modify this option in the second step.
"Import file" field	This field displays the file currently selected for the import.
"…" button	Click on this button to open the "Select import file" window.
"OK" button	Click on this button to confirm your inputs.

15.7.2 "Convert DGN drawing to engineering objects" window

Overview

In the "Convert DGN drawing to engineering objects" window, you define how to conduct the import of the selected file in detail. Your settings only apply to the current import and are not stored.

Furthermore, you can configure the general import settings by clicking on the "Options" button. These settings do not only apply for the current import but for all DGN imports. See also section "Options" window (Page 279).

The window is divided into four parts:

- Upper window area
- Control group "Unit conversion (in mm)"
- Detail settings for the current import
- Buttons in the lower area of the window

Upper window area

The data displayed here is only read out. You cannot change this file.

The following table describes the upper area of the window:

Field	Description
"Options"	This field displays the configuration file currently set. To change the configuration file, click on the "Options" button on the "General" tab in the "Options" window.
"File name"	This field displays the name of the import file selected in the first step.
"File version"	This field displays the version number of MicroStation. COMOS also reads if it is a 2D or a 3D drawing. You can only import 2D drawings. If you attempt to import a 3D drawing, an error message is displayed. The import supports DGN Version 7 and DGN Version 8.

Control group "Unit conversion (in mm)"

Here you can scale the imported DGN drawing so that it matches the scaling of the interactive report.

For a better overview, the print area of the DGN drawing is displayed on the right of the interactive report in millimeters.

The following table describes the options of the "Unit conversion (in mm)" control group:

Option	Description
"1:1"	If this option is activated, the DGN drawing is not scaled.
"1:25.4 (inch ->mm)"	If this option is activated, one inch on the DGN drawing is represented as 25.4 mm on the report.
"1:1000 (meter->mm)"	If this option is activated, one meter on the DGN drawing is represented as 1,000 mm on the report.
"1: X"	If this option is activated, you are free to select how many millimeters one DGN length unit is converted into. Enter the required scaling in the field. Entering a value less than 1 scales the DGN drawing down.

Detail settings for the current import

The following table describes the options used to make the detailed settings for the current import:

Option	Description
"Create base objects only"	If this option is activated, base objects are generated for the cells placed on the DGN drawing, but no engineering objects. See also section Undoing the import (Page 162). Also the graphical information of the drawing is not imported to the interactive report.
"Only graphical import"	If this option is activated, the graphical information of the DGN drawing is imported. New objects are not created. See also section Overview (Page 160). Use the "PipeEasy" tool to convert the imported graphical symbols into objects. See also section Pipe Easy (Page 133).

Option	Description
"Search for changed cells"	If this option is activated, COMOS checks whether the import set contains changed cells. Graphical attributes provide the basis for comparison, e.g.: • Point of origin • Width • Height • Line color If there are changed cells, a separate base object is created for each one detected. Name: [Name of the base object, which is created for the first cell of this name] + "_" + [counter] As an alternative to this option, you can activate the "Import always as local script" option. See also section Algorithm for generating the DGN base objects (Page 163).
"Create XML log"	If this option is activated, a log file is saved at the "XML value" attribute of the report.
"Ignore construction classes"	If this option is activated, MicroStation construction classes are not imported.
"Import hidden layers, too"	If this option is activated, hidden layers of the DGN drawing are imported.
"Import always as local script"	If this option is activated, a base object is created for the first cell imported if there is more than one cell with the same name in the import file. The graphic is stored as a script in the properties of the base object, on the "Symbols" tab. The graphic of the following cells is stored as a local script at the engineering object. This is regardless of whether the cell is mutated or not. This option guarantees the best quality of the imported graphic. Alternatively, you can activate the "Search for mutated cells" option. See also section Algorithm for generating the DGN base objects (Page 163).
"Use 3D rotation matrix"	Activate this option if the import file is a 2D drawing you generated by converting a MicroStation 3D drawing into a 2D drawing.
"Ignore colors"	If this option is activated, the drawing is displayed in black and white in COMOS.
"Import empty cells"	If this option is activated, MicroStation placeholders containing attributes are also imported.

Construction classes are always included in the import.

Buttons in the lower area of the window

The following table describes the buttons in the lower area of the "Convert DGN drawing to engineering objects" window:

Button	Description
"Options"	Click on this button to open the "Options" window where you make the basic settings for all DGN imports.
"OK"	Click on this button to start the import.
"Cancel"	Click on this button to close the "Convert DGN drawing to engineering objects" window. Your settings are not saved.

15.7.3 "Options" window

Overview

The "Options" window is where you make settings for the configuration file. The window comprises the following four tabs:

- "Common"
- "Line types"
- "Layers"
- "Fonts"

"Common" tab

The following table describes the control elements of the "Common" tab:

Control element	Description
"Configuration file" field	When you open the window, this field displays the last configuration file selected. The configuration file saves the assignments made in the "Options" window.
"…" button	Click on this button to set the configuration file.
"Current symbol type" list	This list contains all drawing types currently defined in COMOS with their different symbol types. Dependent upon which entry you select, the corresponding symbols are created on the "Symbols" tab of the new base objects. By default, the list displays the drawing type of the report which is currently open. Recommendation: Create a separate configuration file for each drawing type.

Control element	Description
"Save" button	Click on this button to save the changes to the configuration file.
"Save As..." button	Click on this button to create a new configuration file and save it below the stated name. The new file is not automatically input to the "Configuration file" field. Click on the "..." button to select the file.
"Cancel" button	Click on this button to close the window without saving the changes to the configuration file.

"Line types" tab

The following table describes the control elements of the "Line types" tab:

Control element	Description
"Assigned line types" table	This table displays the current assignments between COMOS line types and DGN line types. A DGN line type may occur in exactly one mapping entry. A COMOS line type can be assigned to multiple DGN line types.
"COMOS line types" table	This table displays the COMOS line types.
"Used DGN line types" table	This table displays the DGN line types which are available in the currently selected import file. The "*" entry is a placeholder for all other DGN line types whose index number is not yet known. No COMOS line type has been assigned to these line types. This entry is always displayed.
"<< Assign" button	Click this button to assign a line type from the "COMOS line types" table to a line type from the "Used DGN line types" table.
"Detach >>" button	Click on this button to undo the assignment for the selected entry in the "Assigned line types" table.

"Layers" tab

The following table describes the control elements of the "Layers" tab:

Control element	Description
"Assigned layers" table	This table displays the current assignments between COMOS layers and DGN layers.
"Used DGN layers" table	This table displays the DGN layers which are available in the currently selected import file. The "*" entry is a placeholder for all other DGN layers whose index number is not yet known. No COMOS layer has been assigned to these layers. This entry is always displayed.

Control element	Description
"Layers for the current symbol type" table	This table displays all COMOS layers that are available on the "Common" tab for the currently selected drawing type. The selected drawing type is displayed in the column header.
"<< Assign" button	Click on this button to assign a layer from the "Layers for the current symbol type" table to a layer from the "Used DGN layers" table.
"Detach >>" button	Click on this button to undo the assignment for the selected entry in the "Assigned layers" table.

"Fonts" tab

The following table describes the control elements of the "Fonts" tab:

Control element	Description
"Path to the font resource files:" field	This field displays the path to the selected file containing the DGN fonts.
"…" button	Click on this button next to the "Path to the font resource files:" field to open the "Browse folders" window.
"COMOS DGN" table	This table displays the current assignments between Windows fonts and DGN fonts.
"…" button	Click on this button next to the "Path to the font resource files:" field to open the "Font" window.
Table for DGN fonts	This table shows the DGN fonts which are available in the currently selected DGN fonts directory.

See also

Configuration file "DGNImport1.xml" (Page 243)

15.8 Pipe Easy

15.8.1 "Create pipes" tab

The following table describes the control elements of the "Create pipes" tab:

Control element	Description
"Base object for pipe break" field	In the COMOS DB, a pipe break can be found on the "Base objects" tab under the following node: "@01 > PID > 99 > 02 > 01 Pipe break". This base object accesses the settings of the "Maximal distance in mm between" control group. If two lines are further apart than specified in the "Two lines" field and closer together than specified in the "Two break ends" field, the following objects are created: • One pipe branch • Two pipe segments • One engineering object for the pipe break This engineering object is created parallel to the pipe branch. The pipe is displayed graphically broken on the report.
"Connector point and line" field	This field displays the maximum permissible X/Y variation of the coordinate comparison between the connector of a report object and a line. The line direction is considered.
"Two lines" field	This field displays the maximum permissible distance between two line ends so the pipe branch can be continued. Bent lines are permissible. The pipe branch created follows the bend of the lines. Note: T connections are not created. If you wish to continue a line beyond a bend, the line must be placed at the end of another line.
"Two break ends" field	This field displays the maximum permissible X/Y variation of the coordinate comparison between two break ends. Collinear liens are not considered. The line must continue on an exact vertical or horizontal plane at the break. The angle deviation must not exceed the value in the "Maximum angle inaccuracy in degree" field.
"Maximum angle inaccuracy" field	This field displays the maximum value up to which COMOS is to interpret two objects as right-angled or collinear.
"Layer for converted lines" field	Enter a layer which has not previously been used in this field. To check the layers that have previously been used in the report, click on the "Layers" button on the report toolbar. The value specified for the layer in this field applies to the next action in each case. If you use different layers for different actions, you must make a note of which lines belong to which layers so that they can subsequently be edited separately. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Layers".
"Permit perpendicular lines only" option	If this option is activated, the direction of the lines is not considered. The "Maximum angle inaccuracy" field is evaluated instead. If the angle inaccuracy remains within the permitted range, the created pipe branch likewise has the angle of the converted lines. Right-angled or collinear lines are not scaled. If the deviation from the default value is too great, the pipe branch breaks off at this point.

Control element	Description
"Consider graphical attributes" option	If this option is activated, the pipe branch breaks off if a line changes its color, line type, or line width. For each graphical representation, a separate pipe segment is created. Note: The DGN import can convert the graphical attributes of the DGN files to COMOS-specific colors and line types. See also section Configuration file for the DGN import (Page 243).
"Action" list	See also section "Create pipes" action (Page 283). See also section "Delete converted lines", "Hide converted lines", and "Show converted lines" actions (Page 284). See also section "Delete short lines" action (Page 284). See also section "Recollect lines" action (Page 284).

15.8.1.1 "Create pipes" action

Modes

The "Create pipes" action knows two modes, dependent upon whether you have selected one or a number of objects on the report:

- No object selected

  If you have not selected an object on the report, pipes are created based on the symbols. Then, further pipes are created based on these generated pipes.

  If no symbols have been placed on the report, this mode is ineffective.

- Selection set

  The selected report objects or lines are the start objects of the conversion. The created COMOS objects are created below the document under the "__IMPORT" object.

Operator input

It is not necessary to click on the "Execute" button every time. Instead, you can select multiple objects on the report and execute the action immediately by pressing the <ENTER> key.

Note

The sequence of the line conversion affects the created pipes.

See also

"Create pipes" tab (Page 282)

15.8.1.2 "Delete converted lines", "Hide converted lines", and "Show converted lines" actions

Display of converted lines

Converted lines are not deleted; they are displayed in green and two millimeters wide. Furthermore, the lines are assigned to the selected layer.

Layers

Use the "Delete converted lines", "Hide converted lines", and "Show converted lines" actions to edit the layer you entered in the "Layer for converted lines" field. Each time, the currently set level is considered.

Example

Thus, the "Delete converted lines" action does not delete all converted lines, but only the converted lines on the level that is entered at present in the "Layers" field.

See also

"Create pipes" tab (Page 282)

15.8.1.3 "Delete short lines" action

DGN files contain a relatively high number of lines shorter than 1 mm. Use the "Delete short lines" action to delete these lines. This increases report processing performance.

15.8.1.4 "Recollect lines" action

Collection

When you execute the "Create pipes" action, all lines of the report are collected. Afterwards, only this collection is processed. All already converted lines will be removed from the collection.

"Undo" function

When you execute the "Undo" function in the report, only the conversion of the lines is undone. The lines are not put back into the collection. When you resume work, these lines are no longer considered by the "Create pipe" action.

When you execute the "Recollect lines" action, the collection of the non-converted lines is updated. This action should only be used if you have previously used the "Undo" function.

15.8.2 "Assign texts" tab

The following table describes the control elements of the "Assign texts" tab:

Control element	Description
"Base object for text rules" field	This field displays the action base object under which the base objects for text rules are located.
"Attribute for text rules" field	This field displays the nested name of the attribute used when searching the objects. If the Comos object has the specified attribute, the value of the attribute is transferred to the corresponding script function as text.
"Maximal distance inaccuracy in mm" field	This field displays the value which defines the tolerance within which Comos interprets a text as still belonging to a symbol. You can rotate the text or the symbol without changing the distance. Rotation always takes place around the placement point and does not change the position of the placement point.
"Layer for converted texts" field	Enter a layer which has not previously been used in this field. To check the layers that have previously been used in the report, click on the "Layers" button on the report toolbar. The value specified for the layer in this field applies to the next action in each case. If you use different layers for different actions, make a note of which lines belong to which layers so that they can subsequently be edited separately. You find additional information on this topic in the "Reports - Basic Operation" manual, keyword "Layers".
"Rule" list	This list shows the base objects for text rules which are located under the action base object.
"Action" list	See also section "Use rule" action (Page 285). See also section "Use all rules" action (Page 286). See also section "Delete converted texts", "Hide converted texts", and "Show converted texts" actions (Page 286). See also section "Delete empty texts" action (Page 286).

15.8.2.1 "Use rule" action

Symbol and text

You have created a rule using symbol and text. Use the "Use rule" action to apply the rule selected in the "Rule" list to all report objects. Selected objects are not evaluated.

All symbols and texts are searched. The same rule is used for symbols and texts with the same relation. The value of the "Maximal distance inaccuracy in mm" field for the relative distance (x and y) is considered here.

Object and attribute

You have created a rule using an attribute. When you select a pipe and a text on the report or specify an attribute, the rule selected in the "Rule" list is only applied once.

If the return value is = "", the text is displayed in green and is set to the layer specified in the "Layer for converted texts" field.

Otherwise, a new red text with the return value of the function is created vertical to the original text and is also set to the layer specified in the "Layer for converted texts" field.

There is no restriction on the number of text rules with different relations which can be created for each action. The relations are saved in the base project on the "Base objects" tab under the "@System > @Profiles > @AllUsers" node.

See also

Creating rules (Page 135)

15.8.2.2 "Use all rules" action

See also section "Use rule" action (Page 285).

When you execute this action, all text rules are applied to all relations.

15.8.2.3 "Delete converted texts", "Hide converted texts", and "Show converted texts" actions

Converted texts are not deleted automatically. Instead, they are set to the currently selected layer.

As is the case with converted lines, you can show, hide, or delete converted texts.

15.8.2.4 "Delete empty texts" action

Dependent upon which rule is used, placeholders can also be processed. Use the "Delete empty texts" action to delete any placeholder texts.

15.8.3 Control elements on the "Create symbol" tab

The following table describes the control elements of the "Create symbol" tab:

Control element	Description
"Object for symbol" field	This field displays the base object used to crate an engineering object from a symbol on a report. This field is a mandatory field.
"At base object" option	If this option is activated, a DevSymbol with the name of the symbol type of the document is created at the base object. The script generated in the selection is used. The drawing type is taken from the report that is currently open. Note: To select the option, you need appropriate rights.
"At report object" option	The DevSymbol at the base object is not created or changed. The script is saved locally at the report object.
"Use existing base object symbol" option	If this option is activated, the symbol graphic is taken from the DevSymbol of the base object. COMOS does not check if the DevSymbol and the selected lines match. It is your administrator's responsibility to check if the selected lines match the symbol from the base object.

15.8.4 Control elements on the "Assignment" tab

The following table describes the control elements of the "Assignment" tab:

Control element	Description
"Imported base object" field	This field displays the imported base object that you want to link to a COMOS base object. When you drag an engineering object into this field, COMOS automatically uses the base object of the engineering object.
"COMOS base object" field	This field displays the COMOS base object that you want to link to an imported base object.
"Link all mutations" option	When a DGN drawing is imported, cells that have identical names but different graphical representations are detected. The base objects of these cells are given a special name. If this option is selected, these objects are linked to a COMOS object.
"Scale linked objects immediately" option	If this option is selected, the imported symbol script is deleted and replaced by the COMOS symbol script. The object is scaled based on the values entered in the "Scale linked objects" control group.
"Carry out zero point correction" option	If this option is activated, the report object is set to its original coordinates.