|  |  |  |
| --- | --- | --- |
| Graphical user interface, website  Description automatically generated | | |
|  | | |
| Document title: | | |
| **NOAKA DEXPI Pilot Transfer Requirements** | | |
| Document no.: | Rev.: | Page: |
|  | **2.4** | **1 of 58** |

NOAKA DEXPI Pilot Transfer Requirements

*Prepared for*NOAKA DEXPI Pilot Group

*Prepared by*

Pedersen, Tonia

**Revision and Signoff Sheet**

**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Reason for issue, key changes and or decisions** | Prepared |
| 0.1.1 |  | Initial | TLP |
| 0.1.2 | 28.10.21 | Updated after comments from Manfred and added ItemTag attribute for all types other than Equipment. | TLP |
| 1.0 | 02.11.21 | Update to include equipment box definitions | TLP |
| 2.0 | 29.11.21 | Equipment box not required to transfer. Updates to OPC and slope sections. | TLP |
| 2.1 | 01.12.21 | Pipe graphical model, Flow Arrow & Virtual Nozzle. | TLP |
| 2.2 | 10.12.21 | Actuated valve example | TLP |
| 2.3 | 14.12.21 | Update metadata example | TLP |
| 2.4 | 17.12.21 | Shutdown and alarm models added | TLP |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Technical Advisor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Email** |
| Manfred Theißen | DEXPI technical resource | theissen@plants-and-bytes.de |

**Table of Contents**

[Overview 5](#_Toc90650802)

[Purpose 5](#_Toc90650803)

[Abbreviations 5](#_Toc90650804)

[References 5](#_Toc90650805)

[Constraints/Assumptions 6](#_Toc90650806)

[Process and Instrumentation Diagram Transfer 7](#_Toc90650807)

[Overview 7](#_Toc90650808)

[P&ID Transfer Breakdown Grouping 7](#_Toc90650809)

[P&ID C01 Example redrafted by Aibel 8](#_Toc90650810)

[DEXPI Standard 9](#_Toc90650811)

[Analysis: 9](#_Toc90650812)

[Proteus Schema 9](#_Toc90650813)

[Analysis 9](#_Toc90650814)

[Transfer Requirements 10](#_Toc90650815)

[Drawing Metadata 11](#_Toc90650816)

[Symbols 13](#_Toc90650817)

[Equipment 17](#_Toc90650818)

[Piping 19](#_Toc90650819)

[Instrumentation 24](#_Toc90650820)

[Equipment Box 37](#_Toc90650821)

[Annotations 39](#_Toc90650822)

[Miscellaneous Graphics with Symbol Reference 41](#_Toc90650823)

[ANNEX A: Custom class definitions 53](#_Toc90650824)

[Drain box 53](#_Toc90650825)

[Virtual Piping Connector 53](#_Toc90650826)

[Custom Class Type Details 53](#_Toc90650827)

[ANNEX B: Custom attribute defintions 54](#_Toc90650828)

[ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’ 56](#_Toc90650829)

[ANNEX D: Symbol Rotation and mirroring example 58](#_Toc90650830)

# 

# Overview

## Purpose

This document provides the requirements for the NOAKA DEXPI pilot project transfer of the P&ID between intelligent engineering systems using the DEXPI 1.3 standard.

## Abbreviations

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Comment** |
| DEXPI | Data Exchange in the Process Industry |  |
| P&ID | Process & Instrument Diagram |  |
| OPC | Off Page Connector |  |
| PIF | ProcessInstrumentFunctions |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref** | **Document / Standard** | **Title / Description** | **Rev** |
| [[1](https://15926.org/home/)] | ISO 15926 | Interoperability standard for process plants |  |
| [[2](https://www.standard.no/fagomrader/energi-og-klima/petroleum/norsok-standard-categories/z-technical-info/z-004/)] | Z-004 | NORSOK CAD Symbol Libraries |  |
| [3] | TR3111 | Equinor LCI Requirements - Data Content and Transfer |  |
| [[4](https://github.com/ProteusXML/proteusxml)] | Proteus schema | Transfer schema | 4.1 |
| [[5](https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf)] | DEXPI | DEXPI P&ID Specification Standard | 1.3 |
| [[6](https://gitlab.com/dexpi/Specification/raw/master/specification/DEXPI%20Specification%201.2.pdf)] | DEXPI | DEXPI P&ID Specification Standard | 1.2 |
| [[7](https://toniapedersen.github.io/DEXPI/Symbols.xlsm)] | Symbols.xlsm | NOAKA DEXPI Pilot Symbol Library | 1.0 |
| [[8]](https://github.com/ProteusXML/proteusxml/blob/master/additional_documents/P%26ID%20Profile%20file%20specification%203.3.3.doc) | P&ID File Specification | ADI / IDS ISO-15926 P&ID file specification | 1.5 |
| [9] |  |  |  |

## Constraints/Assumptions

The following points indicate general constraints/assumptions made when writing this document that were used to determine the scope and should be referenced when determining the design solution.

1. DEXPI 1.3 standard shall be used within the project.
2. It is required that all parties shall use the same graphical representation and ID of each symbol as defined in the NOAKA DEXPI Pilot Symbol Library (Ref: [7] ). It is vital that dimension, origo, rotation and mirroring point information is correctly defined in the symbol as per the definition.
3. Each DEXPI class element shall include a maximum of one symbol reference.
4. No graphical primitive information shall be transferred when the Symbol Reference ID is provided for the object.
5. Current drawing revision number shall be transferred – no historical drawing revision information shall be transferred.
6. It is assumed that all main objects; pipelines, equipment, instruments and main piping components shall exist in the target system.
7. Assume DEXPI diagram item will always have MinX = 0, MinY = 0, BackgroundColor = “white”
8. Project shall assume ‘en’ as the default language for MultiLanguageString types it is therefor not required to transfer ‘Language’ value for these types.
9. Equipment box/table shall not be transferred.

**Constraints that are ONLY applicable for the first stage of the pilot:**

1. No symbol transformation information (mirroring, rotation, scaling) shall be transferred.
2. Dynamic symbol support shall only be required for GateValve types.

# Process and Instrumentation Diagram Transfer

## ****Overview****

The P&ID is a detailed graphical representation of the process flow and interconnection of the involved components, including piping, equipment, valves, instrumentation, and other process components.

Within the NOAKA DEXPI pilot project the content of the P&ID has been broken down into the groups shown in the figure below; requirements regarding the transfer of each group are detailed in the sections below with a focus on the C01 example as redrafted by Aibel.

### ****P&ID Transfer Breakdown Grouping****

Diagram, engineering drawing

Description automatically generated

Figure 1: P&ID NOAKA DEXPI Pilot breakdown

## ****P&ID C01 Example redrafted by Aibel****

As part of the DEXPI initiative the following example P&ID drawings has been provided. This drawing shall be reproduced in each of the drawing tools as per the example below.

[Diagram

Description automatically generated](https://toniapedersen.github.io/DEXPI/Aibel_export.pdf)

Figure 2: Example C01 redrafted by Aibel

## ****DEXPI Standard****

The DEXPI P&ID Specification defines an information model for P&IDs as well as a mapping to the exchange format Proteus Schema.

For the NOAKA DEXPI pilot project all transfers and configuration shall be based on the DEXPI 1.3 standard.

Changes to the DEXPI standard from 1.2 to 1.3 include:

* Equipment types added for solid processes
* Units of measurement cleaned up
* More comprehensive specification of graphics
* Introduction of CustomClass and CustomAttribute elements

### Analysis:

The following sub-section details weaknesses with the DEXPI 1.3 standard that should be addressed with the DEXPI group.

#### Engineering Notes

Engineering notes may be applied to one or many objects on a P&ID to provide additional information to the end user. Notes are an import method of communicating additional information about the process or requirements and are commonly used on the P&ID.

**Issue:** The current solution to provide for the transfer of notes relies on the CustomAttribute element and specialized rules to define the association of the note and the graphical object.

**Suggestion**: A suggestion would be to introduce a new object type within DEXPI

## Proteus Schema

Proteus 4.1 is the current exchange format used to implement the DEXPI 1.3 standard for the P&ID. This format may be replaced in future versions of DEXPI. The analysis below shall be provided as feedback to the DEXPI group to support discussions related to enhancing or replacing Proteus going forward.

### Analysis

The following sub-section details weaknesses with the Proteus 4.1 standard that should be addressed with the DEXPI group.

#### Object Symbol Reference

P&ID design can require the use of more than one symbol to graphically define a single tagged object.

**Issue:** The Proteus format does not support more than one symbol reference per class definition, requiring some the definition of new combined symbols within the pilot project.

**Suggestion**: Create a new symbol as a combination of symbols where two or more symbols are used to represent a single DEXPI class.

## Transfer Requirements

The NOAKA DEXPI pilot project requires that each equipment object displayed on the P&ID shall be transferred within the DEXPI format with the required class mapping, symbol reference ID mapping, attribute mapping and association references to the piping / instrumentation systems as shown on the P&ID. The following sections provide details for the class and attribute mapping and implementation of the standard to reflect the objects and topology of the C01 example based on the group in section: 0. Any additional rules or assumptions to be applied will also be noted in these sections.

**General Requirement Details:**

* PersistentID referencing shall be used to uniquely identify each class object. The context given within the PersistentID shall include a reference to the exporting application.
* DexpiCustomAttributes shall be used as the GenericAttributes ‘Set’ value within the Proteus export to group custom attributes that are defined within ANNEX B: Custom attribute defintions

### Drawing Metadata

Drawing Metadata is the information that is file level specific and is often found as part of the title block of the drawing.

A picture containing diagram

Description automatically generated

Figure : DEXPI MetaData model example

#### Requirement Details:

* Only current revision information shall be transferred in the export file.
* Each DEXPI transfer file shall include the following attributes as part of the drawing metadata:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| DrawingName | http://data.posccaesar.org/rdl/RDS2102503531 | PIPING AND INSTRUMENT DIAGRAM |  |
| DrawingSubTitle | http://sandbox.dexpi.org/rdl/DrawingSubTitleAssignmentClass | TEST P&ID |  |
| DrawingNumber | http://sandbox.dexpi.org/rdl/DrawingNumberAssignmentClass | PID.001 |  |
| ProcessPlantIdentificationCode | http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass | D |  |
| ProcessPlantName | http://sandbox.dexpi.org/rdl/ProcessPlantNameAssignmentClass | KRAFLA |  |
| PlantSystemIdentificationCode | http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass | 20 |  |
| PlantAreaName | http://sandbox.dexpi.org/rdl/AreaIsa95NameAssignmentClass |  |  |
| ProjectNumber | http://sandbox.dexpi.org/rdl/ProjectNumberAssignmentClass |  |  |
| ApprovalDateRepresentation | http://sandbox.dexpi.org/rdl/ApprovalDateRepresentationAssignmentClass |  |  |
| RevisionNumber | http://sandbox.dexpi.org/rdl/RevisionNumberAssignmentClass | 01 |  |
| SheetFormat | http://sandbox.dexpi.org/rdl/SheetFormatAssignmentClass | NTS at A1 |  |
| ApprovalDescription | http://sandbox.dexpi.org/rdl/ApprovalDescriptionAssignmentClass |  |  |
| CreatorName | http://sandbox.dexpi.org/rdl/CreatorNameAssignmentClass |  |  |
| CheckerName | http://sandbox.dexpi.org/rdl/CheckerNameAssignmentClass |  |  |
| ApproverName | http://sandbox.dexpi.org/rdl/ApproverNameAssignmentClass |  |  |

Table 1: DEXPI file metadata Attribute Requirements

### Symbols

All DEXPI class elements that are represented by a graphical symbol shall be represented by and contain a reference to a single symbol ID within the P&ID mapping and final transfer file.

Symbol Reference ID values are given in the NOAKA DEXPI pilot symbol library reference file: [Symbols.xlsm](https://toniapedersen.github.io/DEXPI/Symbols.xlsm)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | **Label Details** | **Label Attribute(s)** | **Options** |
| **DEXPI ID** | **Graphic** | **Description** | **Rotation** | **Mirroring** | **ResizingX** | **ResizingY** | **TR1970** | **Z004** | **Symbol** | **A** | **1** |
| [ND0011](https://toniapedersen.github.io/DEXPI/Symbols/ND0011_Detail.svg) |  | Spring Actuated Safety Valve | Yes | Yes | No | No |  |  | IM005A |  |  |
| [PV019A](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Detail.svg) |  | Valve Ball | Yes | Yes | No | No | STLV008 | PV019A |  | <ObjectDisplayName> | [ValvePosition = 'NC'](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Option1.svg) |

Table 2: Extract from Symbols.xlsm: NOAKA DEXPI reference symbol legend

#### Requirement Details:

* SymbolRegistrationNumber attribute within the Shape element shall be used to provide the reference to the NOAKA DEXPI pilot symbol reference ID.
* Name attribute of the Shape element shall be used as an internal file identifier to link the Shape symbol reference of the object to the Graphical representation instance of the object. Name is implemented in Proteus using ComponentName attribute.
* Location of the instance shall be given by the Shape Usage (Position / Reference) attributes for X & Y axiis. The X & Y points are the location reference to the origo of the symbol as defined in the symbol library for the project.
* Project shall use Label type ‘TagNameLabel’ for any item labels that are represented by a symbol e.g. for SafetyReliefValve.
* Transformation definition for the symbol shall use the definitions and rules as defined within the DEXPI standard (This is described in Ref [6] p. 19). Although any rotation shall be allowed and possible to transfer in the DEXPI file it is preferrable that only 0,90,180 & 270 deg anti-clockwise rotation of the symbol is used.
  + X-axis mirroring is provided for within DEXPI 1.3 through the combination of attributes: ‘IsMirrored=True’ and ‘Rotation=180.0’ as per example (Ref: ANNEX D: Symbol Rotation and mirroring example)
* Target application shall apply functionality to ensure that any text given as part of the symbol is displayed from left-to-right or bottom-to-top based on the rotation of the graphic.
* Target application shall use the rotation values from the transfer file and interpret the information to ensure graphical best fit with a focus on ensuring any symbol rotation transferred can be imported and displayed.
* All connection points shall be on the grid. This will ensure correct placement and management of the drawing connections for import.

#### DEXPI Model Examples

##### DEXPI w/ Proteus implementation: Shape SymbolRegistrationNumber

|  |  |  |
| --- | --- | --- |
| DEXPI definition |  | Proteus Implementation |
| Figure 4: DEXPI Shape SymbolRegistrationNumber |  | Figure 5: Proteus Implementation SymbolRegistrationNumberAssignmentClass |

##### DEXP Model: Shape with label symbol

Diagram

Description automatically generated

Figure 6: DEXPI model example Safety Valve Label Symbol Reference

### Equipment

Equipment is the main items shown in a P&ID to perform the process required treatment. The plant equipment is shown in the P&ID by an icon showing the equipment in basic manner. Equipment is usually identified by a name and unique tag (Unique identifier that is assigned to a field device, skid or equipment).

Diagram

Description automatically generated

Figure 7: DEXPI Equipment model example

#### Requirement Details:

* Each Equipment DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TagName | http://sandbox.dexpi.org/rdl/TagNameAssignmentClass | D-20PA001 | Tag name as stored in the tag register system. |
| ObjectDisplayName | http://sandbox.dexpi.org/rdl/ObjectDisplayName | D-20PA001 | Label text as displayed on the P&ID |
| EquipmentDescription | <http://data.posccaesar.org/rdl/RDS2181987301> | GEAR PUMP | Functional service description of the tagged item. |
| TagType | http://sandbox.dexpi.org/rdl/TagType | PA | Letter code indicating the function of the item. |
| Sequence | http://sandbox.dexpi.org/rdl/Sequence | 0001 | Sequence number which is part of the tag name. |

Table 3: DEXPI Equipment Attribute Requirements

#### DEXPI Class and Symbol References

The table below outline the required equipment objects and the required references to be used within the DEXPI transfer file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| D-20PA001 | RotaryPump | http://data.posccaesar.org/rdl/RDS420749 | [PP003A](https://toniapedersen.github.io/DEXPI/Symbols/PP003A_Detail.svg) | NA |
| D-20HA001 | PlateHeatExchanger | http://data.posccaesar.org/rdl/RDS420749 | [PE010A](https://toniapedersen.github.io/DEXPI/Symbols/PE010A_Detail.svg) | NA |
| D-20VA001 | Separator | http://data.posccaesar.org/rdl/RDS2194378711 | [PT002A](https://toniapedersen.github.io/DEXPI/Symbols/PT002A_Detail.svg) | NA |
| D-20PA002 | CentrifugalPump | http://data.posccaesar.org/rdl/RDS416834 | [PP001A](https://toniapedersen.github.io/DEXPI/Symbols/PP001A_Detail.svg) | NA |
| D-20HA002 | TubularHeatExchanger | http://data.posccaesar.org/rdl/RDS13971182 | [PE037A](https://toniapedersen.github.io/DEXPI/Symbols/PE037A_Detail.svg) | NA |
| D-20TD414 | CustomEquipment | Refer: Drain box | [PZ013A](https://toniapedersen.github.io/DEXPI/Symbols/PZ013A_Detail.svg) | NA |

Table 4: DEXPI Equipment Symbol Reference Requirements

### Piping

A Piping System is an assembly of various components put together with a proper method of joints, functionally to transport fluid from its source to destination. The different components put together are defined as piping components. They are designed for withstanding the operating and design conditions specified in the process parameters. The following sub-sections detail specific subsets of requirements based on component group types within the piping system.

Diagram

Description automatically generated

Figure 8: DEXPI PipingNetworkSystem model example

#### Piping

This sub-section details the transfer requirements related to the PipingNetworkSystem, PipingNetWorkSegment and Pipe elements.

##### Requirement Details:

* Each pipeline shall be represented by a separate PipingNetworkSystem
* Each pipeline shall contain one or more PipingNetworkSegments where the topology of the PipingNetworkSegments is defined in ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’
* Each PipingNetworkSystem DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | D-20L00001A | The pipeline tag number. |
| ObjectDisplayName | http://sandbox.dexpi.org/rdl/ObjectDisplayName | D-20L00001A-1800PL-AD200- | Label text as displayed on the P&ID |
| LineDescrption | http://sandbox.dexpi.org/rdl/LineDescription | D-20L00001A-1800PL-AD200- | Complete Line number as indicated on the P&ID. |
| NominalDiameterNumericValueRepresentation | http://sandbox.dexpi.org/rdl/  NominalDiameterNumericalValueRepresentationAssignmentClass | 0800 | Nominal diameter for the line |
| PipingClassCode | http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass | AD750 | Piping class for the line |
| ProductCode | http://sandbox.dexpi.org/rdl/ProductCode | PL | Product service code for the line |
| LineSuffix | http://sandbox.dexpi.org/rdl/LineSuffix | A | Size indicator suffix |

Table 5: DEXPI PipingNetworkSegment Attribute Requirements

#### Piping Components

Piping components are those components that are connected in-line with the pipe to support the transport of fluid from its source to destination. Piping components referred to in this section can be (but are not limited to) one of the following types:

* Pipe fittings
* Flanges
* Gaskets
* Manually Operated Valves
* Special Items
* Nozzles

The Piping Components group as discussed here does not include pipes, safety valves or actuated valves. Safety valves and actuated valves are detailed in Instrumentation section.

##### Requirement Details:

* NozzleTee symbol shall be used to represent all nozzles on the P&ID
* Where a nozzle has not been included on the drawing between the equipment and the pipe/measuring line the transfer shall include a ‘virtual’ nozzle to ensure compliance with DEXPI. A ‘virtual’ nozzle shall be identified using the Nozzle custom attribute ‘IsVirtual= true’ (Ref: ANNEX B: Custom attribute defintions)
* BlindFlange DEXPI class mapping shall be used in the transfer file for the flange where there is one PipingNode connection
* Flange DEXPI class mapping shall be used in the transfer file for the flange where there are two PipingNode connections.
* VirtualPipingConnector custom class shall be used to provide for direct pipe to pipe connections i.e., where there is no graphical representation of a connecting piping compontent between the two pipes shown on the P&ID. (Ref: Virtual Piping Connector)
* Each PipingComponent and Nozzle DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | D-VB20-0002 | Tag name as stored in the tag register |
| ObjectDisplayName | http://sandbox.dexpi.org/rdl/ObjectDisplayName | D-VB20-0002 | Label text as displayed on the P&ID |
| TagType | http://sandbox.dexpi.org/rdl/TagType | VB | Letter code indicating the function of the item. |
| Sequence | http://sandbox.dexpi.org/rdl/Sequence | 0002 | Sequence number which is part of the tag number. |

Table 6: DEXPI PipingComponent Attribute Requirements

##### DEXPI Class and Symbol References

The table below details the required piping component objects and the required references to be used within the DEXPI transfer file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Piping Component** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| Nozzle | Nozzle | http://data.posccaesar.org/rdl/RDS415214 | [ND0002](https://toniapedersen.github.io/DEXPI/Symbols/ND0002_Detail.svg) | NA |
| Manway | Nozzle | <http://data.posccaesar.org/rdl/RDS415214> | [PZ003A](https://toniapedersen.github.io/DEXPI/Symbols/PZ003A_Detail.svg) | NA |
| D-VF20-0003 | ButterflyValve | <http://data.posccaesar.org/rdl/RDS416609> | [PV018A](https://toniapedersen.github.io/DEXPI/Symbols/PV018A_Detail.svg) | NA |
| D-VC20-0001 | CheckValve | http://data.posccaesar.org/rdl/RDS292229 | [PV013A](https://toniapedersen.github.io/DEXPI/Symbols/PV013A_Detail.svg) | NA |
| D-VB20-0001 | BallValve | http://data.posccaesar.org/rdl/RDS416654 | [PV019A](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Detail.svg) | NA |
| Reducer | PipeReducer | http://data.posccaesar.org/rdl/RDS416294 | [PE001A](https://toniapedersen.github.io/DEXPI/Symbols/PE001A_Detail.svg) | NA |
| BlindFlange | BlindFlange | http://data.posccaesar.org/rdl/RDS414719 | [PV002A](https://toniapedersen.github.io/DEXPI/Symbols/PV002A_Detail.svg) | NA |
| Flange | Flange | http://data.posccaesar.org/rdl/RDS13307654 | [PV002A](https://toniapedersen.github.io/DEXPI/Symbols/PV002A_Detail.svg) | NA |
| D-VB20-0003 | BallValve | http://data.posccaesar.org/rdl/RDS416654 | [PV019A](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Detail.svg) | NA |
| D-VB20-0002 | BallValve | http://data.posccaesar.org/rdl/RDS416654 | [PV019A](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Detail.svg) | NA |
| D-VG20-0001 | GateValve | http://data.posccaesar.org/rdl/RDS416519 | [PV005A](https://toniapedersen.github.io/DEXPI/Symbols/PV005A_Detail.svg) | NA |
| D-VG20-0002 | GateValve | http://data.posccaesar.org/rdl/RDS416519 | [PV005A](https://toniapedersen.github.io/DEXPI/Symbols/PV005A_Detail.svg) | NA |
| *‘VirtualPipingConnector’* | CustomPipingComponent | Ref: Virtual Piping Connector | NA | NA |

Table 7: DEXPI Piping Component Symbol Reference Requirements

##### Requirements Details: Dynamic Gate Valve Symbol

DEXPI transfer and application functionality shall support dynamic symbols, symbols that change their graphical appearance based on attribute values of the associated item.

* Each GateValve DEXPI transfer object shall ALSO include the following attribute in the transfer file if available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ValvePosition | <http://sandbox.dexpi.org/rdl/ValvePosition> | NC | Symbol options provided in the symbol library show how this will be represented. Ref [7] |

Table 8: DEXPI GateValve ADDITIONAL attribute requirements

### Instrumentation

Instrumentation is the items shown in a P&ID required to run, monitor and control a specific process. E.g. Indicators, Recorders, Controllers , including: pressure, temperature and flow instruments, control valves, pressure safety valves, meters etc.

#### Instrumentation (not incl. actuated valves or Safety Valves)

Diagram

Description automatically generated

Figure 9:DEXPI Instrumentation model example

##### Requirement Details:

* All instruments (not incl. actuated and safety valves) shall be represented by a ProcessInstrumentFunction (PIF) class element in DEXPI.
* InstrumentationLoopFunction class must be used as a 'grouping' mechanism for ProcessInstrumentFunctions (PIF) where the loop identifier for those PIF items is known.
* Signal lines shall be represented using SignalLineFunction class
* SignalConveyingFunction Parent shall be the ProcessInstrumentationFunction item associated with the ‘sending’ PIF. \*Ensures consistency with cases where the PIF is associated with an ActuatingFunction.
* Measure lines between instrument and piping component shall be represented using MeasuringLineFunction class.
* DEXPI MeasuringLineFunction shall have an ProcessSignalGeneratingFunction as its Source
* DEXPI ProcessInstrumentationFunction reference shall contain a ProcessInstrumentationFunctionType descriptor attribute where both a field device and shared display/shared control element exists for the object and the tag name for these objects is the same.
* Each InstrumentationLoopFunction DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| InstrumentationLoopFunctionNumber | <http://sandbox.dexpi.org/rdl/>InstrumentationLoopFunctionNumberAssignmentClass | 0003 | Loop number use to group associated instruments. |

Table 9: DEXPI InstrumentationLoopFunction Attribute Requirements

* Each Instrument (PIF) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | D-20TI-0003 | Tag name as stored in the tag register system. |
| TagType | http://sandbox.dexpi.org/rdl/TagType | TI | Letter code indicating the function of the item. |
| ProcessInstrumentationFunctionCategory | http://sandbox.dexpi.org/rdl/  ProcessInstrumentationFunctionCategoryAssignmentClass | T | Function category |
| ProcessInstrumentationFunctions | http://sandbox.dexpi.org/rdl/  ProcessInstrumentationFunctionsAssignmentClass | I | Additional functions |
| ProcessInstrumentationFunctionNumber | http://sandbox.dexpi.org/rdl/  ProcessInstrumentationFunctionNumberAssignmentClass | 0003 | Sequence number |
| ProcessInstrumentationFunctionType | http://sandbox.dexpi.org/rdl/  ProcessInstrumentationFunctionTypeAssignmentClass | Discrete / SharedDisplaySharedControl | Differentiate between field device and control function block |

Table 10: DEXPI Instrument (PIF) Attribute Requirements

##### DEXPI Class and Symbol References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| D-20HV-0001 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| D-20TI-0003 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| D-20TIC-0003 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| D-20PI-0005 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| D-20PIC-0004 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| D-20PI-0004 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |

Table 11: DEXPI PIF Symbol Reference Requirements

#### Instrumentation (Safety valves)

Safety Valves shall be transferred using the DEXPI SafetyValveOrFitting class or one of the SafetyValveOrFitting subtype class within the PipingNetworkSystem.

Diagram

Description automatically generated

Figure 10: Instrumentation (Safety Valve) DEXPI model example

##### Requirement Details:

* Safety Valve label symbol reference shall be transferred as a separate reference on the label element as per the example given in Figure 6: DEXPI model example Safety Valve Label Symbol Reference
* Each Instrument (Safety Valve) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | D-20PSV-0002 | Tag name as stored in the tag register system. |
| Sequence | http://sandbox.dexpi.org/rdl/Sequence | 0002 | Sequence number which is part of the tag number. |
| TagType | http://sandbox.dexpi.org/rdl/TagType | PSV | Letter code indicating the function of the item. |

Table 12: DEXPI Instrument (Safety Valves) Attribute Requirements

##### DEXPI Class and Symbol References

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | **RDS** | | **Item Symbol** | **Label Symbol** |
| D-20PSV-0002 | SpringLoadedAngleGlobeSafetyValve | | http://sandbox.dexpi.org/rdl/SpringLoadedAngleGlobeSafetyValve | [ND0011](https://toniapedersen.github.io/DEXPI/Symbols/ND0011_Detail.svg) | IM005A |

Table 13: DEXPI Safety Valve Symbol Reference Requirements

#### Instrumentation (actuated valves)

Actuated Valves are complex DEXPI structures as the valve is part of the piping package while instrumentation provides the automation. The special DEXPI modelling is shown below. Note that for actuated valves there are a number of RepresentationGroups required: including the representation for the valve, the actuator and the instrument bubble symbol.

Diagram

Description automatically generated

Figure 11: DEXPI Actuated Valve model example

A screenshot of a computer

Description automatically generated with medium confidence 

Figure 12: Proteus Implementation Actuated Valve (Ref: included example file ‘ActuatedValve\_Example.xml’)

##### Requirement Details:

* InstrumentationLoopFunction class association for an actuated valve shall be via the associated ProcessInstrumentationFunction object Ref: Figure 11: DEXPI Actuated Valve model example.
* DEXPI ActuatingFunction shall have an associated SignalLineFunction. The ActuatingFunction shall be the Target of the SignalLineFunction.
* The ActuatingSystem class shall contain the information relevant for the tag.
* Each Instrument (ActuatingSystem) DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | D-20PSV-0002 | Tag name as stored in the tag register system. |
| TagType | http://sandbox.dexpi.org/rdl/TagType | PSV | Letter code indicating the function of the item. |
| TypicalInformation | http://sandbox.dexpi.org/rdl/TypicalInformationAssignmentClass |  | Code identifying the associated Typical |

Table 14: DEXPI Instrument Attribute Requirements

##### DEXPI Class and Symbol References

The following class and symbol references provides the details for the Instrumentation (Actuated Valves) as a complex mapping type consisting of three DEXPI class elements that together represent the actuated valve.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Detail** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| Label showing valve tag name details | ActuatingSystem | http://sandbox.dexpi.org/rdl/ActuatingSystem | [IM005A](https://toniapedersen.github.io/DEXPI/Symbols/IM005A_Detail.svg) | NA |
| Actuator symbol snapped to the valve | ControlledActuator | http://sandbox.dexpi.org/rdl/ControlledActuator | [PA001A](https://toniapedersen.github.io/DEXPI/Symbols/PA001A_Detail.svg) | NA |
| The basic valve symbol. | GateValve | http://data.posccaesar.org/rdl/RDS416519 | [PV005A](https://toniapedersen.github.io/DEXPI/Symbols/PV005A_Detail.svg) | NA |

Table 15: DEXPI Actuated Valve Symbol Reference Requirements

#### Instrumentation (Alarm and shut-down ‘cause’)

The following section details additional modelling and transfer requirements for instrument control functions that have alarm points and/or associated shutdown function(s).

Diagram

Description automatically generated

Figure 13: DEXPI Instrumentation Alarm & Shut-down ‘cause’ model

##### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (not incl. actuated valves or Safety Valves)

##### DEXPI Class and Symbol References

The following class and symbol references provides the details for the Instrumentation (Alarm and shut-down ‘cause’) example above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| D-20LST-0078 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| A-PSD4-20-52 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [IM017A](https://toniapedersen.github.io/DEXPI/Symbols/IM017A_Detail.svg) | NA |

Table 16: DEXPI Symbol Reference Requirement for example

#### Instrumentation (Shut-down ‘effect’ with multiple signal lines)

The following section details additional modelling and transfer requirements for instrument shutdown function(s) ‘effect’ elements.

Note: each shutdown ‘cause’ object may have one or more ‘effect’ objects.

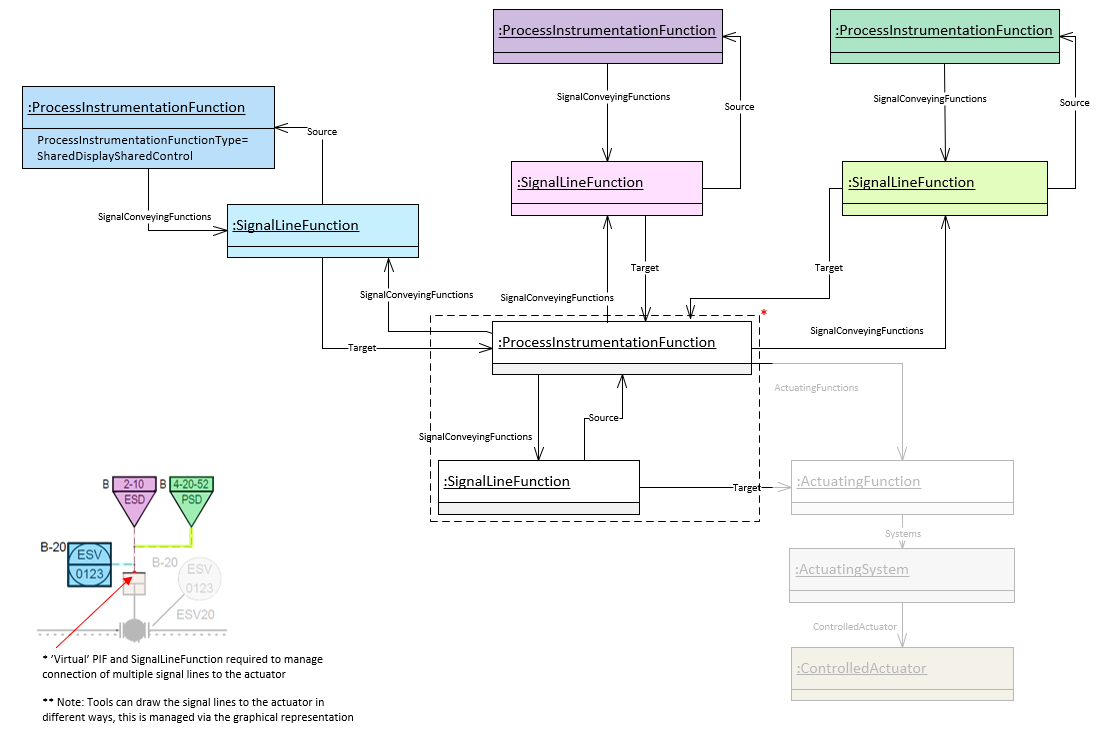


Figure 14: DEXPI Instrumentation Shut-down ‘effect’ model with multiple signal lines

##### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (not incl. actuated valves or Safety Valves)
* Where more than one signal is connected to an actuator the mapping shall include a ‘virtual’ ProcessInstrumentationFunction and SignalLineFunction as a method of ‘collecting’ the signal lines before associating them with the ActuatingFunction element.

Note: Open DEVOPS task: [TASK 58517](https://dev.azure.com/EquinorASA/Spine/_workitems/edit/58517) to determine management of overlapping signal lines.

##### DEXPI Class and Symbol References

The following class and symbol references provides the details for the Instrumentation (Shut-down ‘effect’ with multiple signal lines) example above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| B-20ESV-0123 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [ND0006](https://toniapedersen.github.io/DEXPI/Symbols/ND0006_Detail.svg) | NA |
| B-ESD2-10 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [IM017A](https://toniapedersen.github.io/DEXPI/Symbols/IM017A_Detail.svg) | NA |
| B-PSD4-20-52 | ProcessInstrumentationFunction | http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction | [IM017A](https://toniapedersen.github.io/DEXPI/Symbols/IM017A_Detail.svg) | NA |

Table 17: DEXPI Symbol Reference Requirement for example

### Equipment Box

**Equipment box shall not be transferred**.

This section has been left as a reference in the event that the project determines it is necessary to transfer equipment box information.

Equipment Box/Table is the addition of a tabulated list of attribute name and value pairs that are associated with an equipment shown on the P&ID. The extent of the requirements for this list of attributes shall be as per the project requirements. The Equipment box transfer will be managed similarly to the symbol transfer via a reference ID.

Diagram

Description automatically generated

Figure 15: DEXPI Equipment box model example

#### Requirement Details:

* Any attributes that are shown in the equipment box shall be transferred with the associated item in the transfer file.
* During the pilot project only Tag Name and Description are required to be transferred using the DEXPI attribute definitions below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TagName | http://sandbox.dexpi.org/rdl/TagNameAssignmentClass | D-20PA001 | Tag name as stored in the tag register system. |
| EquipmentDescription | <http://data.posccaesar.org/rdl/RDS2181987301> | GEAR PUMP | Functional service description of the tagged item. |

Table 18: DEXPI Equipment Box Attribute Requirements

#### DEXPI Class and Equipment Box Symbol References

The table below outlines the required equipment objects and the required equipment box references to be used within the DEXPI transfer file:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Example Display Name** | **DEXPI Class** | | | **Equipment Box Symbol** | |
| D-20PA001 | RotaryPump | | | [STPE102](https://toniapedersen.github.io/DEXPI/Symbols/STPE102_Detail.svg) | |
| D-20HA001 | | PlateHeatExchanger | [STPE114](https://toniapedersen.github.io/DEXPI/Symbols/STPE114_Detail.svg) | |
| D-20VA001 | | Separator | [STPE101](https://toniapedersen.github.io/DEXPI/Symbols/STPE101_Detail.svg) | |
| D-20PA002 | | CentrifugalPump | [STPE102](https://toniapedersen.github.io/DEXPI/Symbols/STPE102_Detail.svg) | |
| D-20HA002 | | TubularHeatExchanger | [STPE113](https://toniapedersen.github.io/DEXPI/Symbols/STPE113_Detail.svg) | |

Table 19: DEXPI Equipment Box Symbol Reference Requirements

### Annotations

Annotations is the text added to the P&ID to provide additional information about an item or a group of items on the drawing. Each annotation shall be linked to at least one item on the drawing, however, the modelling described below will allow for annotation that are not connected.. An item on the drawing can have more than one annotation.

DEXPI standard does not currently contain an annotation class element, this has been identified as a gap within DEXPI. Annotation shall instead be modelled with Name, Text and an arbitrary number of PersistentIDs as detailed below within the MetaData object.

Notes are modeled as objects with a Name (string, e.g., "NOTE 1") and a Text (string, e.g., "CAUTION, HOT!") and an arbitrary number of PersistentID objects as an array (not ordered in terms of UML).

#### Requirement Details:

* Each note shall be identified via an ID (xml:id, e.g., "Note-1"). The ID has no external meaning, it is only used to refer to a note within a Proteus file.
* Each note shall have at least one PersistentID with a Context (string, e.g., "PidMaker123") and an Identifier (string, "hw893"). PersistentIDs are transferred within an array to allow for more than one PersistentID value pair per note.
* Notes shall be implemented as part of the MetaData object e.g. they are "file global" (w.r.t. a P&ID/DexpiModel/Proteus file)
* Any engineering object (e.g., a CentrifugalPump) can refer to any number of notes.
* Each note can be referred to by any number of engineering object

##### DEXPI w/ Proteus implementation: Annotations

|  |  |  |
| --- | --- | --- |
| DEXPI definition |  | Proteus Implementation |
| Figure 16: DEXPI model for Annotation |  | Timeline  Description automatically generated with low confidence  Figure 17: Proteus Implementation Annotations |

### Miscellaneous Graphics with Symbol Reference

#### Graphics Modelled within Dexpi

The following section contains details for those miscellaneous elements that are modelled within DEXPI 1.3 standard and are defined with a symbol reference.

##### PropertyBreak

Property break graphic shall be used to represent the break of ‘one’ attribute only i.e. each break type shall be represented by a separate property break graphic.

The type of break being represented by the property break graphic shall be identified by setting the associated property break attribute as defined below. The graphic shall display the ‘break’ attribute from the pipeline on either side of the property break graphic.

\*\*Note: PropertyBreak class is a PipingNetworkSegmentItem subtype in DEXPI 1.3



Figure 18: DEXPI PropertyBreak model example

###### Requirement Details:

* Each PropertyBeak DEXPI transfer object shall use one (and only one) of the following attributes to identify the type of property break:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Transfer Value** | **Value RDS** |
| InsulationBreak | http://sandbox.dexpi.org/rdl/InsulationBreakSpecialization | InsulationBreak | http://sandbox.dexpi.org/rdl/InsulationBreak |
| NominalDiameterBreak | http://sandbox.dexpi.org/rdl/NominalDiameterBreakSpecialization | NominalDiameterBreak | http://sandbox.dexpi.org/rdl/NominalDiameterBreak |
| PipingClassBreak | http://sandbox.dexpi.org/rdl/PipingClassBreakSpecialization | PipingClassBreak | http://sandbox.dexpi.org/rdl/PipingClassBreak |
| HeatTracingBreak | http://sandbox.dexpi.org/rdl/HeatTracingBreakSpecialization | HeatTracingBreak | NA |
| UnderAboveGroundBreak | http://sandbox.dexpi.org/rdl/UnderAboveGroundBreakSpecialization | UnderAboveGroundBreak | NA |
| FloorModuleBreak | http://sandbox.dexpi.org/rdl/FloorModuleBreakSpecialization | FloorModuleBreak | NA |
| AreaBreak | http://sandbox.dexpi.org/rdl/AreaBreakSpecialization | AreaBreak | NA |

Table 20: DEXPI Equipment Attribute Requirements

###### DEXPI Class and Symbol References

The table below outlines the required property break objects and the required references to be used within the DEXPI transfer file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Type** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| Property Break | PropertyBreak | http://sandbox.dexpi.org/rdl/PropertyBreak | [ND0007](https://toniapedersen.github.io/DEXPI/Symbols/ND0007_Detail.svg) | NA |

Table 21: DEXPI Equipment Symbol Reference Requirements

##### Piping Off Page Connector (Out)

Piping Off Page Connector (OPC) graphic shall be used to represent a pipeline that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘out’.

Diagram

Description automatically generated

Figure 19: DEXPI FlowOutPipeOffPageConnector model example

###### Requirement Details:

* The piping off page connector (Flow Out) shall use the correct symbol to indicate flow direction out dependant on which side of the drawing the graphic is placed.
* Each FlowOutPipeOffPageConnector shall be transferred with an associated PipeOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each PipeOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

###### DEXPI Class and Symbol References

The table below outlines the required off page connector class type and the required symbol reference to be used within the DEXPI transfer file for a given flow direction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Type** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| OPC (Flow Out Right) | FlowOutPipeOffPageConnector | http://sandbox.dexpi.org/rdl/FlowOutPipeOffPageConnector | [ND0008](https://toniapedersen.github.io/DEXPI/Symbols/ND0008_Detail.svg) | NA |
| OPC (Flow Out Left) | FlowOutPipeOffPageConnector | <http://sandbox.dexpi.org/rdl/FlowOutPipeOffPageConnector> | [ND0009](https://toniapedersen.github.io/DEXPI/Symbols/ND0009_Detail.svg) | NA |

##### Piping Off Page Connector (In)

Piping Off Page Connector (OPC) graphic shall be used to represent a pipeline that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘in’.

Diagram

Description automatically generated

Figure 20: DEXPI FlowInPipeOffPageConnector model example

###### Requirement Details:

* The piping off page connector (Flow In) shall use the correct symbol to indicate flow direction in dependant on which side of the drawing the graphic is placed.
* Each FlowInPipeOffPageConnector shall be transferred with an associated PipeOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each PipeOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

###### DEXPI Class and Symbol References

The table below outlines the required off page connector class type and the required symbol reference to be used within the DEXPI transfer file for a given flow direction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Type** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| OPC (Flow In Left) | FlowInPipeOffPageConnector | http://sandbox.dexpi.org/rdl/FlowInPipeOffPageConnector | [ND0008](https://toniapedersen.github.io/DEXPI/Symbols/ND0008_Detail.svg) | NA |
| OPC (Flow In Right) | FlowInPipeOffPageConnector | <http://sandbox.dexpi.org/rdl/FlowInPipeOffPageConnector> | [ND0009](https://toniapedersen.github.io/DEXPI/Symbols/ND0009_Detail.svg) | NA |

##### Signal Off Page Connector (Out)

Signal Off Page Connector (OPC) graphic shall be used to represent a signal that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘out’.

Diagram

Description automatically generated

Figure 21: DEXPI FlowOutSignalOffPageConnector model example

###### Requirement Details:

* The signal off page connector (Flow Out) shall use the correct symbol to indicate flow direction out dependant on which side of the drawing the graphic is placed.

###### DEXPI Class and Symbol References

The table below outlines the required off page connector class type and the required symbol references to be used within the DEXPI transfer file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Type** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| OPC (Flow Out Right) | FlowOutSignalOffPageConnector | http://sandbox.dexpi.org/rdl/FlowOutSignalOffPageConnector | [ND0008](https://toniapedersen.github.io/DEXPI/Symbols/ND0008_Detail.svg) | NA |
| OPC (Flow Out Left) | FlowOutSignalOffPageConnector | http://sandbox.dexpi.org/rdl/FlowOutSignalOffPageConnector | [ND0009](https://toniapedersen.github.io/DEXPI/Symbols/ND0009_Detail.svg) | NA |

##### Signal Off Page Connector (In)

Signal Off Page Connector (OPC) graphic shall be used to represent a signal that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘in’.

Diagram

Description automatically generated

Figure 22: DEXPI FlowInSignalOffPageConnector model example

###### Requirement Details:

* The signal off page connector (Flow In) shall use the correct symbol to indicate flow direction in dependant on which side of the drawing the graphic is placed.

###### DEXPI Class and Symbol References

The table below outlines the required off page connector class type and the required symbol references to be used within the DEXPI transfer file:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Type** | **DEXPI Class** | **RDS** | **Item Symbol** | **Label Symbol** |
| OPC (Flow In Left) | FlowInSignalOffPageConnector | http://sandbox.dexpi.org/rdl/FlowInSignalOffPageConnector | [ND0008](https://toniapedersen.github.io/DEXPI/Symbols/ND0008_Detail.svg) | NA |
| OPC (Flow In Right) | FlowInSignalOffPageConnector | http://sandbox.dexpi.org/rdl/FlowInSignalOffPageConnector | [ND0009](https://toniapedersen.github.io/DEXPI/Symbols/ND0009_Detail.svg) | NA |

##### Slope

Pipeline slope graphic shall be used to represent the slope for the pipeline.

Graphical user interface

Description automatically generated

Figure 23: DEXPI Slope model example

###### Requirement Details:

* The pipeline slope graphic shall use the ‘PipeSlopeSymbol’ type as per the model example above.
* The pipeline slope graphic shall reference the SymbolRegistrationNumber ‘STPL008’ within the transfer file as per the model example above.
* The association between the PipingNetworkSegment and the pipeline slope graphic shall be provided in the DEXPI transfer file following the model example above.

##### Piping Flow Direction Arrow

Pipeline flow direction arrow graphic shall be used to represent the flow direction for the pipe.

Graphical user interface, application

Description automatically generated

Figure 24: DEXPI Slope model example

###### Requirement Details:

* The pipeline flow arrow graphic shall use the ‘PipeFlowArrow’ type as per the model example above.
* The pipeline flow arrow graphic shall reference the SymbolRegistrationNumber ‘ND0010’ within the transfer file as per the model example above.
* The association between the PipingNetworkSegment and the pipeline flow arrow graphic shall be provided in the DEXPI transfer file following the model example above.

#### Graphics Items Without Symbol Reference

The following section details any special requirements regarding graphical elements that are not represented by a symbol.

##### Pipe

Pipe elements are represented in DEXPI as a connection between points: ConnectorLine InnerPoints and/or PipingNodePosition Position points.

Diagram

Description automatically generated

Figure 25: DEXPI Pipe with angle model example

###### Requirement Details:

* The PipingNodePosition Position point of the connected PipingNode shall represent a point on the PipingNodeOwner e.g. GateValve / Nozzle

##### LeaderLines

TBD

###### Requirement Details:

# ANNEX A: Custom class definitions

The following section details the custom class definitions that shall be available within the NOAKA DEXPI pilot profile for verification within the pilot project.

DEXPI type definitions for each custom class type can be found in Table 22: NOAKA DEXPI Pilot Custom Class

## Drain box

DrainBox type is defined within TR0052 as part of the ‘Atmospheric storage tanks and containment functions’ grouping.

The Drain box are defined as having only one piping connection point, typically at the base of the symbol for ongoing pipe connections and no connection point at the top.

## Virtual Piping Connector

VirtualPipingConnector type is defined as a special piping connector type. This connector type is required in the case where the P&ID shows two pipes connecting directly with each other i.e., there is no graphical representation of a connecting piping compontent between the two pipes shown on the P&ID.

This type has a Supertype ‘PipingNodeOwner’ and thus provides the necessary PipingNode connection points between PipingNetworkSegments.

This type has no graphical representation.

## Custom Class Type Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag Type | DEXPI class | URI | TypeName | TypeURI |
| Drain Box | CustomEquipment | http://sandbox.dexpi.org/rdl/CustomEquipment | DrainBox | http://data.posccaesar.org/rdl/RDS298844 |
| Virtual Piping Connector | CustomPipingComponent | http://sandbox.dexpi.org/rdl/  CustomPipingComponent | VirtualPipingConnector | http://sandbox.dexpi.org/rdl/  VirtualPipingConnector |

Table 22: NOAKA DEXPI Pilot Custom Class

# ANNEX B: Custom attribute defintions

The following section details the custom attribute definitions that shall be available within the NOAKA DEXPI pilot profile for verification within the pilot project.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | URI | | Example | Type | Multi. | Description |
| ItemTag | http://sandbox.dexpi.org/rdl/ItemTagAssignmentClass | | D-20L00001A | String | 0..1 | Tag Name attribute as shown in the tag management register. |
| ObjectDisplayName | http://sandbox.dexpi.org/rdl/ObjectDisplayName | | D-20L00001A-1800PL-AD200- | String | 0..1 | Label text as displayed on the P&ID |
| LineDescription | http://sandbox.dexpi.org/rdl/LineDescription | | D-20L00001A-1800PL-AD200- | String | 0..1 | Complete Line number as indicated on the P&ID. |
| ValvePosition | http://sandbox.dexpi.org/rdl/ValvePosition | | NC / NO | String | 0..1 | Options provide in the symbol library detail how this will be interpreted. Ref [7] |
| TagType | http://sandbox.dexpi.org/rdl/TagType | | VB | String | 0..1 | Letter code indicating the function of the item. |
| Sequence | http://sandbox.dexpi.org/rdl/Sequence | | 0002 | String | 0..1 | Sequence number which is part of the tag number. |
| Notes | http://sandbox.dexpi.org/rdl/Notes | | {  ID="Note-1",  Name="NOTE 1",  Text="CAUTION, HOT!",  PersistentIDs=[  {Identifier="hw893" Context="PidMaker123"},  {Identifier="1" Context="PID Fix Pro"}]} | String | 0..\* |  |
| ProductCode | http://sandbox.dexpi.org/rdl/ProductCode | PL | | String | 0..1 | Product service code for the line |
| LineSuffix | http://sandbox.dexpi.org/rdl/LineSuffix | A | | String | 0..1 | Size indicator suffix |
| HeatTracingBreak | http://sandbox.dexpi.org/rdl/  HeatTracingBreakSpecialization | HeatTracingBreak | | String | 0..1 | Indicates HeatTrace break |
| UnderAboveGroundBreak | http://sandbox.dexpi.org/rdl/  UnderAboveGroundBreakSpecialization | UnderAboveGroundBreak | | String | 0..1 | Indicates Under/Above ground break |
| FloorModuleBreak | http://sandbox.dexpi.org/rdl/  FloorModuleBreakSpecialization | FloorModuleBreak | | String | 0..1 | Indicates Floor/Module break |
| AreaBreak | http://sandbox.dexpi.org/rdl/AreaBreakSpecialization | AreaBreak | | String | 0..1 | Indicates Area break |
| IsVirtual | http://sandbox.dexpi.org/rdl/IsVirtual | true | | String | 0..1 | This object is not represented graphically on the drawing. |
| ProcessInstrumentationFunctionType | http://sandbox.dexpi.org/rdl/  ProcessInstrumentationFunctionTypeAssignmentClass | Discrete / SharedDisplaySharedControl | | String | 0..1 | Differentiate between field device (Discrete) and Function block (SharedDisplaySharedControl) |

Table 23: NOAKA DEXPI Pilot Custom Attributes

# ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’

**(Ref: [8]) Section 2.2.1 for DEXPI 1.3)**

**PipingNetworkSegment Topology (Connection element)**

Components within a PipingNetworkSegment are considered to be implicitly connected, by their main flow in and flow out connections points, in the order that they are represented in the PipingNetworkSegment. This ordering may differ between a P&ID and 3D model (see **Error! Reference source not found.**)

Each PipingNetworkSegment is a collection of PipingNetworkSegmentItems (e.g., PipingComponents such as Valves) and PipingConnections (e.g., Pipes) with common engineering properties that define a single process flow. Where there is a junction in the flow or a change of specification (e.g., piping class or nominal diameter), the PipingNetworkSegment will terminate.

A PipingNetworkSegment, as its SourceItem, will reference a Nozzle, PipingComponent, or PropertyBreak that it doesn’t contain or it will reference a FlowInPipeOffPageConnector that it contains.

Connection from :

Nozzle

PipingComponent

PropertyBreak

PipingNetworkSegment

…

PipingNetworkSegment

FlowInPipeOffPage­Connector

…

Connection from :

FlowInPipeOffPage­Connector

A PipingNetworkSegment, as its TargetItem, will reference a Nozzle, merging component (a PipingComponent such as a Tee) that it doesn’t contain or it will reference a PipingComponent, PropertyBreak, or FlowOutPipeOffPageConnector that it contains as its last component.

PipingNetworkSegment

PipingNetworkSegment

Connection to :

Nozzle

Merging component

…

PipeConnectorSymbol

Reducer

Splitting Component

…

Connection to :

PipingComponent, PropertyBreak  
FlowOutPipeOffPage­Connector

If the TargetItem of a PipingNetworkSegment is a contained PipingNetworkSegmentItem, the TargetNode of the PipingNetworkSegment is the main downstream PipingNode of this PipingNetworkSegmentItem, if applicable (i.e. the main flow out of the segment).

# ANNEX D: Symbol Rotation and mirroring example

The following table shows the rotation and mirroring of an example symbol and demonstrates that mirroring the base symbol (shown in RED) on the Y-axis and then rotating 180o provides the same result as if the base symbol had been mirrored on the X-Axis.

Shape, polygon

Description automatically generated

Figure 26: Symbol Rotation Attribute Example