|  |  |  |
| --- | --- | --- |
| Graphical user interface, website  Description automatically generated | | |
|  | | |
| Document title: | | |
| **NOAKA DEXPI Transfer Requirements** | | |
| Document no.: | Rev.: | Page: |
|  | **3.0** | **1 of 58** |

NOAKA DEXPI 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 |
| 2.5 | 19.01.22 | Added in-line instrument and updated instrumentation section headers, heat tracing & connected equipment details. | TLP |
| 3.0 | 26.01.22 | A number of updates to be in alignment with symbol legend mapping to DEXPI | TLP |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Technical Advisor**

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

**Table of Contents**

[Overview 5](#_Toc94113062)

[Purpose 5](#_Toc94113063)

[Abbreviations 5](#_Toc94113064)

[References 5](#_Toc94113065)

[Constraints/Assumptions 6](#_Toc94113066)

[Process and Instrumentation Diagram Transfer 7](#_Toc94113067)

[Overview 7](#_Toc94113068)

[P&ID Transfer Breakdown Grouping 7](#_Toc94113069)

[P&ID C01 Example redrafted by Aibel 8](#_Toc94113070)

[DEXPI Standard 9](#_Toc94113071)

[Analysis: 9](#_Toc94113072)

[Proteus Schema 9](#_Toc94113073)

[Analysis 9](#_Toc94113074)

[Transfer Requirements 10](#_Toc94113075)

[Symbols 11](#_Toc94113076)

[File Metadata 15](#_Toc94113077)

[Functional System & Plant Area 17](#_Toc94113078)

[Equipment 19](#_Toc94113079)

[Piping 21](#_Toc94113080)

[Instrumentation 26](#_Toc94113081)

[Equipment Box 40](#_Toc94113082)

[Annotations 42](#_Toc94113083)

[Miscellaneous Graphics with Symbol Reference 44](#_Toc94113084)

[ANNEX A: Custom class definitions 54](#_Toc94113085)

[Drain box 54](#_Toc94113086)

[Virtual Piping Connector 54](#_Toc94113087)

[Custom Class Type Details 54](#_Toc94113088)

[ANNEX B: Custom attribute defintions 55](#_Toc94113089)

[ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’ 56](#_Toc94113090)

[ANNEX D: Symbol Rotation and mirroring example 58](#_Toc94113091)

# Overview

## Purpose

This document provides the requirements for the NOAKA DEXPI 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 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] | POSC Caesar Reference Data Library | [Reference Data Service (posccaesar.org)](http://data.posccaesar.org/rdl/) |  |

## 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 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. NOAKA DEXPI Symbol Library symbols shall be used as the basis for mapping towards DEXPI class definitions.
4. Any Symbol that does not have a match to a DEXPI class definition will be defined with a DEXPI custom class and contain a TypeName and TypeURI reference to provide the type details. POSCCAESAR Ref: [9] will be used as the reference library where possible.
5. Each DEXPI class element shall include a maximum of one symbol reference.
6. No graphical primitive information shall be transferred when the Symbol Reference ID is provided for the object.
7. Current drawing revision number shall be transferred – no historical drawing revision information shall be transferred.
8. It is assumed that all main objects; pipelines, equipment, instruments and main piping components shall exist in the target system.
9. Assume DEXPI diagram item will always have MinX = 0, MinY = 0, BackgroundColor = “white”
10. Project shall assume ‘en’ as the default language for MultiLanguageString types it is therefor not required to transfer ‘Language’ value for these types.
11. Equipment box/table shall not be transferred.

**Constraints that are ONLY applicable for the March deliverables within the project:**

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 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 breakdown

## ****P&ID Example redrafted by Aibel.. need to add new example****

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

## ****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 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 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

Within the NOAKA DEXPI project it has been decided that the symbols shall form the basis of the mapping toward DEXPI to support the initial decision that we shall only transfer the graphical information from the P&ID in the DEXPI transfer file.

The NOAKA DEXPI 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 NOAKA process engineering drafting standards and symbols. 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

### Symbols

The NOAKA DEXPI symbol legend is based on Z004 NORSOK standard with some additions to support the DEXPI transfer, this project symbol legend is referred to as the NOAKA DEXPI symbol legend.

All symbols within this legend are mapped to a corresponding DEXPI class definition including any additional attributes to support the definition of the object.

A key feature of the NOAKA DEXPI symbol legend is the addition of a defined origo point for all symbols. A standard origo point definition across NOAKA will ensure correct placement of the symbol based on a single point element (x,y) for the shape in the transfer file.

The origo point placement is based on a few general rules:

* In-line components with 2 connectors the origo is placed on the center of the line transecting the symbol along the plane of the to-be connected pipeline to ensure ease of rotation on the pipeline and ease of substitution for intelligent attribute based symbols.
* In-line components with 3 or 4 connectors the origo is placed at the intersection point of the connectors
* In-line components with 1 connection point the origo is placed at the connection point

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | **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/Detail/ND0011_Detail.svg) |  | Spring Actuated Safety Valve | Yes | Yes | No | No |  |  | IM005A |  |  |
| [PV019A](https://toniapedersen.github.io/DEXPI/Symbols/Detail/PV019A_Detail.svg) |  | Valve Ball | Yes | Yes | No | No | STLV008 | PV019A |  | <ObjectDisplayName> | [ValvePosition = 'NC'](https://toniapedersen.github.io/DEXPI/Symbols/PV019A_Option1.svg) |

#### Requirement Details:

* SymbolRegistrationNumber attribute within the Shape element shall be used to provide the reference to the NOAKA DEXPI 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 3: DEXPI Shape SymbolRegistrationNumber |  | Figure 4: Proteus Implementation SymbolRegistrationNumberAssignmentClass |

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

Graphical user interface

Description automatically generated

Figure 5: DEXPI model example Safety Valve Label Symbol Reference

### File 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 6: 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** |
| ApprovalDateRepresentation | http://sandbox.dexpi.org/rdl/ApprovalDateRepresentationAssignmentClass |  |  |
| ApprovalDescription | http://sandbox.dexpi.org/rdl/ApprovalDescriptionAssignmentClass |  |  |
| ApproverName | http://sandbox.dexpi.org/rdl/ApproverNameAssignmentClass |  |  |
| CheckerName | http://sandbox.dexpi.org/rdl/CheckerNameAssignmentClass |  |  |
| CreatorName | http://sandbox.dexpi.org/rdl/CreatorNameAssignmentClass |  |  |
| DrawingName | http://data.posccaesar.org/rdl/RDS2102503531 | PIPING AND INSTRUMENT DIAGRAM |  |
| DrawingNumber | http://sandbox.dexpi.org/rdl/DrawingNumberAssignmentClass | PID.001 |  |
| DrawingSubTitle | http://sandbox.dexpi.org/rdl/DrawingSubTitleAssignmentClass | TEST P&ID |  |
| PlantAreaName | http://sandbox.dexpi.org/rdl/AreaIsa95NameAssignmentClass |  |  |
| PlantSystemIdentificationCode | http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass | 20 |  |
| ProcessPlantIdentificationCode | http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass | D |  |
| ProcessPlantName | http://sandbox.dexpi.org/rdl/ProcessPlantNameAssignmentClass | KRAFLA |  |
| ProjectNumber | http://sandbox.dexpi.org/rdl/ProjectNumberAssignmentClass |  |  |
| RevisionNumber | http://sandbox.dexpi.org/rdl/RevisionNumberAssignmentClass | 01 |  |
| SheetFormat | http://sandbox.dexpi.org/rdl/SheetFormatAssignmentClass | NTS at A1 |  |

### Functional System & Plant Area

System numbering codes and plant area codes are both given on the P&ID to provide information about the engineering elements. DEXPI provides for the transfer of these elements via the PlantSystem & PlantArea class definitions.

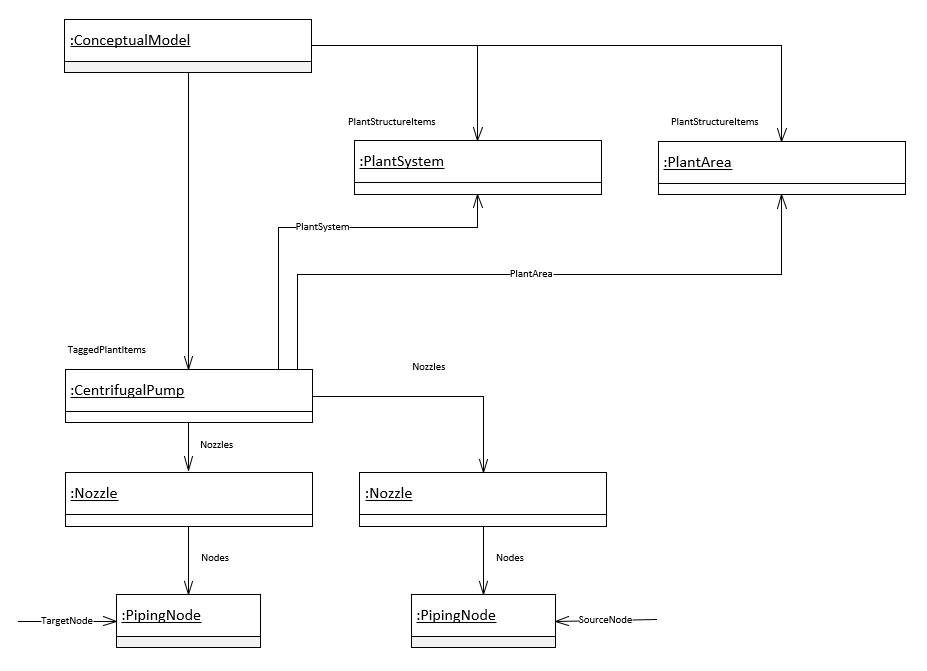


Figure 7: DEXPI Functional System model example

#### Requirement Details:

* PlantSystem association shall be used where this information is available for an item.
* Each PlantSystem DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PlantSystemIdentificationCode | http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass | 20 | System code associated with the item |

* PlantArea association shall be used where this informationis available for an item.
* Each PlantArea DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PlantAreaIdentificationCode | http://sandbox.dexpi.org/rdl/PlantAreaIdentificationCodeAssignmentClass | B23 | Area code associated with the item |

### 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 8: DEXPI Equipment model example

#### Requirement Details:

* Direct connections between Equipment shall be identified within the DEXPI transfer file by transferring the secondary equipment (e.g. motor) unique ID value in the ConnectedEquipment attribute on the primary equipment (e.g. pump) element.
* Each Equipment DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ConnectedEquipmentID | http://noaka.org/rdl/ConnectedEquipmentIDAssignmentClass |  | Proteus XML ID value of the connected equipment element |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | D-20PA001 | Label text as displayed on the P&ID |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 001 | Sequence number which is part of the tag name. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | PA | Letter code indicating the function of the item. |
| EquipmentDescription | http://data.posccaesar.org/rdl/RDS2181987301 | GEAR PUMP | Functional service description of the tagged item. |
| TagName | http://sandbox.dexpi.org/rdl/TagNameAssignmentClass | D-20PA001 | Tag name as stored in the tag register system. |

* For each Equipment DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TypeCode | http://noaka.org/rdl/TypeCodeAssignmentClass | E | Text code to be displayed within the symbol. Ref: [7] ‘Label Attributes’ |

### 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 9: 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
* Where heat tracing is utilized on a pipingnetworksystem this information shall be transferred via DEXPI using the HeatTracingType attribute on the PipingNetworkSystem. All elements within the PipingNetworkSystem shall inherit the heat tracing setting using one of the following values:
  + NULL: heat tracing system has not yet been evaluated
  + HeatTracingSystem: Heat tracing will be used on the piping network system
  + NoHeatTracingSystem: Heat tracing will not be used on the piping network system
* PipingNetworkSystems shall contain all PipingNetworkSegments that share the same pipeline name ‘ItemTag’ attribute value.
* 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://noaka.org/rdl/ItemTagAssignmentClass | D-20L00001A | The pipeline tag number. |
| LineDescription | http://noaka.org/rdl/LineDescriptionAssignmentClass | D-20L00001A-1800PL-AD200- | Complete Line number as indicated on the P&ID. |
| LineSuffix | http://noaka.org/rdl/LineSuffixAssignmentClass | A | Size indicator suffix |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | D-20L00001A-1800PL-AD200- | Label text as displayed on the P&ID |
| ProductCode | http://noaka.org/rdl/ProductCodeAssignmentClass | PL | Product service code for the line |
| SegmentLineTypeRepresentation | http://noaka.org/rdl/ SegmentLineTypeRepresentationAssignmentClass | Primary/Secondary/Utility | Indicate line type to use |
| HeatTracingType | http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization | NoHeatTracingSystem | Indicates that no heat tracing is used. |
| NominalDiameterNumericValueRepresentation | http://sandbox.dexpi.org/rdl/ NominalDiameterNumericalValueRepresentationAssignmentClass | 800 | Nominal diameter for the line |
| PipingClassCode | http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass | AD750 | Piping class for the line |

#### 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 DEXPI transfer object shall include the following attributes when available/applicable:

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

* Each Nozzle DEXPI transfer object shall include the following attributes when available/applicable:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| IsVirtual | http://noaka.org/rdl/IsVirtualAssignmentClass | TRUE | Set to ‘true’ if the Nozzle is included in the DEXPI modelling but does not have a graphical representation on the drawing. |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | M01 | Tag name as stored in the tag register |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | M01 | Label text as displayed on the P&ID |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 01 | Sequence number which is part of the tag number. |
| TagType | http://noaka.org/rdl/TypeCodeAssignmentClass | M | Letter code indicating the function of the item. |

* For each PipingComponent DEXPI transfer object that is defined as a special item shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SpecialItemNumber | http://noaka.org/rdl/SpecialItemNumberAssignmentClass | 3029 | Text for special item identification to be shown in SpecialItem label. |

##### 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://noaka.org/rdl/ValvePositionAssignmentClass | NC | Symbol options provided in the symbol library show how this will be represented. Ref [7] |

### 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 (Off-Line Instrumentation)

Diagram

Description automatically generated

Figure 10: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 conveying lines shall be represented using DEXPI SignalConveyingFunction 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. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20TI-0003 | Tag name as stored in the tag register system. |
| ProcessInstrumentationFunctionLocation | http://noaka.org/rdl/ ProcessInstrumentationFunctionLocationAssignmentClass | Field / Primary / Auxiliary / Inaccessable | Location information for the instrument. |
| ProcessInstrumentationFunctionType | http://noaka.org/rdl/ ProcessInstrumentationFunctionTypeAssignmentClass | Discrete / SharedDisplaySharedControl | Differentiate between field device and control function block |
| TagSuffix | http://noaka.org/rdl/TagSuffixAssignmentClass | A | Suffix code |
| ProcessInstrumentationFunctionCategory | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionCategoryAssignmentClass | T | Function category |
| ProcessInstrumentationFunctionNumber | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionNumberAssignmentClass | 0003 | Sequence number |
| ProcessInstrumentationFunctions | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionsAssignmentClass | I | Additional functions |
| ProcessInstrumentationFunctionCategory | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionCategoryAssignmentClass | T | Function category |

* For each ProcessInsrumentationFunction DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TypeCode | http://noaka.org/rdl/TypeCodeAssignmentClass | PSD | Text code to be displayed within the symbol. Ref: [7] ‘Label Attributes’ |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SignalConveyingFunctionTypeRepresentation | http://noaka.org/rdl/SignalConveyingFunctionTypeRepresentationAssignmentClass | SignalConveying | Linestyle type representation as defined in the Symbols legend for ‘SignalAndLineStyles’ |

Calendar

Description automatically generated with low confidence

Text

Description automatically generated

Figure 11: Instrumentation symbols with reference to FunctionType & FunctionLocation

##### Instrumentation (Off-Line instrument 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 12: DEXPI Instrumentation Alarm & Shut-down ‘cause’ model

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Alarm levels shall be represented by DEXPI SignalConveyingFunction class type and shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PortStatus | http://sandbox.dexpi.org/rdl/PortStatusSpecialization | StatusLowLowPort | Alarm type |

##### Instrumentation (Off-line Instrument 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.

Diagram

Description automatically generated

Figure 13: DEXPI Instrumentation Shut-down ‘effect’ model with multiple signal lines

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Where more than one signal is connected to an actuator the export system shall ensure that there is 1 ‘owner’ of the ActuatingFunction and that this shall be a SharedDisplaySharedControl type PIF if available.

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

##### Instrumentation Electrical Powered Equipment

This is a special grouping for electrical powered objects that receive/send signals from other electrical powered objects, controllers, shutdowns and equipment.

Diagram

Description automatically generated

Figure 14: DEXPI Instrumentation Electrical Powered Equpment model

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Electrical signal lines between ‘instrument electrical powered equipment’ object and equipment shall be represented using SignalConveyingFunction class which has a DEXPI ActuatingElectricalFunction class as its Target.

#### Instrumentation (In-Line Instrumentation)

\*\*Does not include instrument safety valves or actuated valves.

In-line instrumentation are those elements that are part of the piping network and provide an instrumentation function.

Graphical user interface, table

Description automatically generated

Figure 15: Example In-line thermowell with associated transmitter.

##### Requirement Details:

* In-line instrument on the P&ID shall be represented as both an in-line instrument **and** a separate off-line instrument as a ‘short-cut’ to allow for signal connections. In the case where a signal connection is required this shall be via a separate off-line instrument connected to the in-line instrument.
* All in-line instruments (not incl. actuated and safety valves) shall be represented by an InlinePrimaryElement (or subtype) class element in DEXPI.
* Measure lines between in-line instrument and off-line instrument shall be represented using MeasuringLineFunction class.
* DEXPI MeasuringLineFunction shall have an ProcessSignalGeneratingFunction as its Source
* Each InlinePrimaryElement (or subtype) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| InlinePrimaryElementFunctionCategory | http://noaka.org/rdl/InlinePrimaryElementFunctionCategoryAssignmentClass | T | Function category |
| InlinePrimaryElementFunctions | http://noaka.org/rdl/InlinePrimaryElementFunctionsAssignmentClass | W | Additional functions |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20TI-0003 | Tag name as stored in the tag register system. |
| TagType | http://noaka.org/rdl/TypeCodeAssignmentClass | TI | Letter code indicating the function of the item. |

* For each InlinePrimaryElement DEXPI transfer object that is defined as a special item shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SpecialItemNumber | http://noaka.org/rdl/SpecialItemNumberAssignmentClass | 5068 | Text for special item identification to be shown in SpecialItem label. |

#### Instrumentation (Safety and Self Acting Valves)

Instrumentation (Safety and Self-Acting Valves) shall be transferred as a DEXPI PipingComponent suptype within the DEXPI PipingNetworkSystem.

Diagram

Description automatically generated

Figure 16: Instrumentation (Safety Valve) DEXPI model example

##### Requirement Details:

* Label symbol reference shall be transferred as a separate reference on the label element as per the example given in Figure 5: DEXPI model example Safety Valve Label Symbol Reference
* Each Instrument (Safety and Self-Acting Valve) shall be represented by DEXPI SafetyValveOrFitting (or subtype) include the following attributes when available:

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

#### 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 17: DEXPI Actuated Valve model example

A screenshot of a computer

Description automatically generated with medium confidence 

Figure 18: 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 17: DEXPI Actuated Valve model example.
* DEXPI ActuatingFunction shall have an associated SignalLineFunction. The ActuatingFunction shall be the Target of the SignalLineFunction.
* The OperatedValve (or subtype) class shall contain the information relevant for the tag.
* The OperatedValve (or subtype) class element shall ‘own’ the link to the label for this item type.
* Each OperatedValve (or subtype) DEXPI transfer object shall include the following attributes when available:

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

* For each Actuator (ControlledActuator) DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | M | Letter code indicating the function of the item. |

### 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.

Graphical user interface, application, table

Description automatically generated

Figure 19: 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 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. |

### 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).

|  |  |  |
| --- | --- | --- |
| DEXPI definition |  | Proteus Implementation |
| Diagram  Description automatically generated  Figure 20: DEXPI model for Annotation |  | Timeline  Description automatically generated with low confidence  Figure 21: Proteus Implementation Annotations |

#### 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
* The custom attribute ‘Notes’ shall be used for both the Metadata details of the note and the reference to the note from the process item as per the models above and the examples in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| Notes | http://noaka.org/rdl/NotesAssignmentClass | Note-1 | Equipment reference ID to Metadata Notes parameterized value text. |
| Notes | http://noaka.org/rdl/NotesAssignmentClass | Value = "{  ID="Note-1",  Name="NOTE 1",  Text="CAUTION, HOT!",  PersistentIDs=[  {Identifier="hw893" Context="PidMaker123"},  {Identifier="1" Context="PID Fix Pro"}]}" | Metadata attribute to transfer parameterized details of a note. To be referred to via the ID value. |

### 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 22: DEXPI PropertyBreak model example

###### Requirement Details:

* Each PropertyBeak DEXPI transfer object shall use the following attributes to identify the type of property break:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Transfer Value** | **Value RDS** |
| AreaBreak | http://noaka.org/rdl/AreaBreakSpecializationAssignmentClass | AreaBreak | NA |
| FloorModuleBreak | http://noaka.org/rdl/FloorModuleBreakSpecializationAssignmentClass | FloorModuleBreak | NA |
| HeatTracingBreak | http://noaka.org/rdl/HeatTracingBreakSpecializationAssignmentClass | HeatTracingBreak | NA |
| UnderAboveGroundBreak | http://noaka.org/rdl/UnderAboveGroundBreakSpecializationAssignmentClass | UnderAboveGroundBreak | NA |
| 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 |

##### TBD: 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 23: 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. |

##### TBD: 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 24: 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. |

##### TBD: 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 25: 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.
* Each SignalOffPageConnectorReferenceByNumber 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. |

##### TBD: 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 26: 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.
* Each SignalOffPageConnectorReferenceByNumber 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. |

##### Slope

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

Graphical user interface

Description automatically generated

Figure 27: 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 28: 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 29: 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

##### Leader Lines

Leader lines are represented in DEXPI as a polyline with an ordered set of points depicting the start, end and any intermediate points of the line.

Diagram

Description automatically generated

Figure 30: Leader line graphic representation example

###### Requirement Details:

* Leader lines shall be ‘owned’ by the label as shown above.

# ANNEX A: Custom class definitions

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

DEXPI type definitions for each custom class type can be found in Table 1: NOAKA DEXPI 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 1: NOAKA DEXPI Custom Class

# ANNEX B: Custom attribute defintions

Refer to NOAKA DEXPI Symbol legend file Ref: [7]

# 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 31: Symbol Rotation Attribute Example