

Glossary

On this page

- [Alias](#)
- [Artifact](#)
- [Bidirectional](#)
- [Data](#)
- [DataHub operations](#)
- [DataHub tree](#)
- [Data Source](#)
- [DHTrace](#)
- [Driver](#)
- [Excluded](#)
- [Global ID](#)
- [HP Application Lifecycle Management \(HP ALM\)](#)
- [Item](#)
- [Item type](#)
- [Link](#)
- [Mapping mode](#)
- [Module](#)
- [Node](#)
- [Orphan](#)
- [OSLC](#)
- [OSLC link](#)
- [OSLC query](#)
- [PendingDelete](#)
- [PendingUpdate](#)
- [Recursively](#)
- [Relationship](#)
- [Requirements traceability](#)
- [Schema map](#)
- [Status](#)
- [Synchronized](#)
- [Sync relation](#)
- [System model](#)
- [Unidirectional](#)

Alias

Alias is a name that is assigned to a node in the DataHub Explorer tree or MagicDraw[®] Containment area tree. This node can then be accessed or removed via the [Alias Manager](#) dialog.

Artifact

An artifact is a kind of tangible by-products produced during software development. Some artifacts, e.g., Use Cases, Class diagrams, and other Unified Modeling Language (UML) models, and requirements and design documents, help describe the function, architecture, and design of software. Other artifacts are concerned with the process of development itself, e.g., project plans, business cases, and risk assessments.

Bidirectional

A source-to-target link. A bidirectional link will update both source and target nodes.

Data

Data refers to an item whose specific format enables it to be synchronized with and referenced to and from another data. Data can be a MagicDraw[®] element or relation, a DOORS node or relation, or a CSV row.

DataHub operations

You can use DataHub operations to copy data, copy data with sync, create OSLC Link, copy data and create DHTrace, or create DHTrace. The table below describes the functions of the DataHub operations.

Operation	Function
Copy Data	To copy data without creating any DHLinks.
Copy Data With Sync	To copy data and create a DHLink for synchronization.
Copy Data with DHTrace	To copy data and also create DHTrace links between source and target.
Create DHTrace	To create only a DHTrace link without any data.

Create OSLC Link	To create an OSLC Link.
------------------	-------------------------

DataHub tree

A DataHub tree refers to a tree structure designed to show drivers, Data Sources, and items of the connected Data Sources in DataHub. The tree within DataHub Explorer has two top levels: DataHub and Data Sources respectively. The tree consists of four drivers (MagicDraw[®], DOORS, DOORS Next Generation, and CSV) and each driver, except DOORS, has one or more Data Sources.

Data Source

A Data Source refers to the project path of an application. A Data Source can be any of the following

- MagicDraw[®] project
- IBM[®] Rational[®] DOORS[®] client
- IBM[®] Rational[®] DOORS[®] Next Generation client
- CSV file

An application can have more than one Data Source. Each Data Source refers to a specific project path and has a unique Data Source ID.

You can connect DataHub to the following applications

- MagicDraw[®]
- IBM[®] Rational[®] DOORS[®]
- IBM[®] Rational[®] DOORS[®] Next Generation
- Data repository in the CSV format.

DHTrace

The DHTrace relationship is a specialization of a Dependency, connecting model elements or sets of elements that represent the same concept across models. A DHTrace is often used to track requirements and model changes, typically in the Traceability diagram, a Class, Use Case, Object, or the Composite Structure diagram.

As changes can occur in both directions, the order of this Dependency is usually ignored. The properties of the relationship can specify the DHTrace mapping, but the DHTrace is usually bidirectional, informal, and rarely computable.

Driver

A driver refers to a connector to either an application or to a file. Each driver has a unique Driver ID. There are three applications (MagicDraw[®], DOORS, and Excel) and one data repository in the CSV format. Each application uses a specific driver. For example, the *DOORS* driver connects to *DOORS* and *DOORS* Next Generation driver connects to *DOORS* Next Generation.

Excluded

Excluded refers to the status of an ignored node only in Sync relations. The excluded nodes depend on the Sync relation that one node can have both Excluded and the other synchronization status.

Global ID

A global ID consists of a Driver ID, Data Source ID, Type ID, and Item ID arranged in order and separated by a delimiter “/”. Each item or item type in the DataHub tree has a unique global ID.

HP Application Lifecycle Management (HP ALM)

HP ALM is a set of software tools developed and marketed by the HP Software Division for application development and application testing. It includes tools for requirements management, test planning and functional testing, performance testing, developer management and defect management.

Item

An item refers to a particular element in the DataHub tree. Each item has a unique item ID. Elements such as drivers, data sources, folders, packages, DOORS formal modules, requirements (data), and relationships (links) in the DataHub Explorer tree are called items. A node may have nested children as items.

[Data](#) refers to an item whose specific format enables it to be synchronized with and traced to and from another data.

A [link](#) is an item whose specific format can link one data to another and also enables it to be synchronized with other links.

Item type

An item type consists of a list of properties. For example, a SysML Requirement type consists of two properties (ID and Text). Thus, every item with the same item type shares the same property list. An item type can be a driver type, Data Source type, IBM DOORS requirement type, MagicDraw[®] stereotype, or CSV column. Each item type has a unique Type ID.

Link

A link is an item whose specific format can link one data to another and also enables it to be synchronized with other links. A link can only be a MagicDraw[®] relationship, an IBM[®] Rational[®] DOORS[®] link, or an OSLC Link. Unlike data, links can only be synchronized. A link is also a URI reference from one resource, subject, or source to another resource, and can be an object or target. In RDF and OSLC, we use links to model relationships. Links are unidirectional like relationships.

Mapping mode

DataHub has two types of mapping, **Individual Type Mapping** and **Group Type Mapping**. The **Individual Type Mapping** shows the same node structure as the dragged nodes, and you need to map each source node to each target node. The **Group Type Mapping** groups nodes according to their types and allows you to map a source type to a target type.

Module

A DOORS container which may contain hierarchically arranged sub-modules. Typically, this is a requirement specification tree.

Node

A node is like a leaf in a tree. It represents a distinct point within the tree structure. A node can have children or be a child itself.

Orphan

Orphan refers to the status of an updated item that attempts to synchronize to the other side of the relation that has been deleted.

OSLC

DataHub, as of this version facilitates OSLC queries. Open Services for Lifecycle Collaboration (OSLC) is an open community creating specifications for integrating tools. The goal of OSLC is to create specifications for interactions between tools.

In OSLC, each artifact in the lifecycle, e.g., a requirement, defect, test case, source file, or development plan, is an HTTP resource that is manipulated using the standard methods of the HTTP specification (GET, PUT, POST, DELETE).

OSLC link

The OSLC specification defines relationships, which can be modeled as links.

Types of unidirectional links supported include *affectedBy*, *constrainedBy*, *constrains*, *decomposedBy*, *decomposes*, *elaborates*, *elaboratedBy*, *implementedBy*, and *satisfiedBy*.

OSLC query

DataHub supports two types of OSLC queries - basic query and advanced query. The basic query is a simple text based on searching, while the advanced query syntax follows SQL or SPARQL.

PendingDelete

PendingDelete refers to the status of an item that occurs because one or more related items have been deleted. The PendingDelete status is similar to the Orphan status, but the PendingDelete occurs at synchronization.

PendingUpdate

PendingUpdate refers to the status of an item that occurs because one or more related items have been updated (changes have been made to the item(s)). Items in the PendingUpdate status can be either accepted to update the properties or discarded.

Recursively

When this option is chosen, all the child nodes under the selected node are included. Otherwise, only a single element is processed.

Relationship

A relationship exists between two resources when there is something that connects them. The two resources work together, belong together, are similar, or should be considered together. There are different types of relationships.

Requirements traceability

Requirements traceability is concerned with documenting the life of a requirement and providing bidirectional traceability between various associated requirements. It enables you to find the origin of each requirement and track every change that has been made to this requirement.

Requirements include design artifacts, implementation, and are finally verified. The artifacts tied to the latter stages can be traced back to the requirements as well. This is typically done via a Requirements Traceability matrix.

Schema map

A schema Map refers to the mapping of attributes between the source and target types. The source and target types can be identical or different. Apart from mapping, you can also use schema maps to copy items from different item types and update them for attribute synchronization.

The table below is an example of attribute mapping between MagicDraw® and the DOORS formal module.

MagicDraw® SysML requirements	Functional requirements module
Name	Object Heading
Id	Object Identifier
Text	Object Text

Status

Status refers to the status of an association item. The status of an item varies depending on the association type (**Sync DHLink** or **DHTrace**). If the association type is a **Sync DHLink**, the item status will be **Synchronized**, **PendingUpdate**, **PendingDelete**, **Excluded**, or **Orphan**. If it is a trace, its status might be **Suspect**.

Synchronized

Synchronized refers to the status of relations that the related nodes need to be the same as the other side or the status of an item that does not have any pending changes.

Sync relation

DHLink synchronizations are associations between data in both source and target items. Items with DHLink can synchronize one another. DHLink synchronization is available in the **Operation** drop-down list in DataHub Explorer. There are two kinds of DHLink: Unidirectional and bidirectional.

- **Unidirectional DHLink synchronization** is either from a source to a target or vice-versa. Data is synchronized in only one direction and not both directions.
- **Bidirectional DHLink synchronization** refers to two-way synchronization. When the data in the source is updated, the data in the target item will be updated as well and vice versa. When you synchronize data with bidirectional DHLink synchronization, DataHub will establish the consistency in data hierarchy of both items.

If two data items have an Association, they can be open from one another.

System model

A system model is the conceptual model that describes and represents a system. A system is comprised of multiple views such as planning, requirement (analysis), design, implementation, deployment, structure, behavior, input data, and output data views. A system model must describe and represent all these multiple views.

Unidirectional

This is a one-way synchronization either from a source to a target or from a target to a source. Unidirectional synchronization changes only one side of the source and target.