

Glossary of terms

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 one of many kinds of tangible by-products produced during the development of software. Some artifacts (e.g., use cases, class diagrams, and other Unified Modeling Language (UML) models, requirements and design documents) help describe the function, architecture, and design of software. Other artifacts are concerned with the process of development itself—such as project plans, business cases, and risk assessments.

Bidirectional

This is 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.

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: (i) MagicDraw®, (ii) IBM® Rational® DOORS®, (iii) IBM® Rational® DOORS® Next Generation, and (iv) data repository in CSV format.

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.

DataHub Operation Mode

You can use DataHub operation mode to copy data, copy data with sync, create OSLC Link, copy data and create trace, or create trace. The table below describes the function of the DataHub operation mode.

Operation Mode	Function
Copy Data	To copy data without creating any DHLink.
Copy Data With Sync	To copy data and create the DHLink which is used for synchronization.
Copy Data with DHTrace	To copy data and also create trace links between source and target.
Create DHTrace	Only creates trace link without any data.
Create OSLC Link	Creates OSLC Link

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 – for example, a requirement, defect, test case, source file, or development plan and so on – is an HTTP resource that is manipulated using the standard methods of the HTTP specification (GET, PUT, POST, DELETE).

OSLC Query

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

OSLC Link

The OSLC specification defines relationships and that relationships should be modeled as links. The types of unidirectional links that are supported include *affectedBy*, *constrainedBy*, *constrains*, *decomposedBy*, *decomposes*, *elaborates*, *elaboratedBy*, *implementedBy*, *satisfiedBy*, among others.

Module

A *DOORS* container within which hierarchically arranged sub-modules may be contained, typically, this is a requirement specification tree.

Relationship

A relationship is said to exist between two resources if there is something that connects them; they work together, belong together, are similar or should be considered together. There may be different types of relationships.

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 is required to describe and represent all these multiple views.

Requirements Traceability

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

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

Link

A link is a URI reference from one resource, the subject or source, to another resource, said to be the object or target. In RDF and in OSLC we use links to model relationships and like relationships, links are unidirectional.

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 as DataHub itself consists of four drivers (MagicDraw®, DOORS, and DOORS Next Generation and CSV) and each driver has one or more data sources, except DOORS.

Driver

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

Global ID

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

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 (see "**Data**" above).

A link is an item whose specific format makes it possible to link one data to another and also enables it to be synchronized with other links (see "**Link**" below).

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, a data source type, an IBM DOORS requirement type, a MagicDraw® stereotype, or a CSV column. Each item type has a unique Type ID.

Link

A link is an item whose specific format makes it possible to 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.

Recursively

When this option is chosen, all the child nodes under the selected node will be included. If not, then only a single element will be processed.

Synchronized

Synchronized refers to the status of relations that the related nodes need to be the same as the other side.

Excluded

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

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.

PendingDelete

PendingDelete refers to the status of an item that occurs because one or more related items have been deleted. Pending delete status is similar to Orphan status, only it 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 PendingUpdate status can be either accepted to update the properties or discarded.

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 mode box in DataHub Explorer.

There are two kinds of DHLink: (i) unidirectional and (ii) bidirectional.

(i) Unidirectional is either from source to target or vice-versa. Data is synchronized in only one direction and not both directions.

(ii) Bidirectional DHLink Synchronization refers to two-way synchronization meaning that whenever data in the source is updated, data in the target item will be updated as well and vice versa. When you synchronize data with bidirectional DHLink,

DataHub will establish consistency in data hierarchy in both items.

If two data items have an association, they can be opened from one another.

Synchronized

Synchronized refers to the status of an item that does not have any pending changes.

Schema Map

A Schema Map refers to the mapping of attributes between the source and target types. The source or 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 the MagicDraw® and DOORS formal module.

MagicDraw® SysML Requirements	Functional Requirements Module
Name	Object Heading
Id	Object Identifier
Text	Object Text

Mapping Mode

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

Status

Status refers to the status of an association item. The status of an item will vary depending on the association type (Sync DHLINK or Trace). If the association type is a Sync DHLINK, the item status will be **Synchronized**, **PendingUpdate**, **PendingDelete**, **Excluded**, or **Orphan**, or if it is a trace, its status might be suspect.

Trace

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

As changes can occur in both directions, the order of this Dependency is usually ignored. The relationship's properties can specify the trace mapping, but the trace is usually bi-directional, informal, and rarely computable.

Unidirectional

This is a one way synchronization either from source to target or target to source. Unidirectional sync will change only one side of source and target.