

IBM InfoSphere Information Server
Version 8 Release 7

Guide to Managing Common Metadata



IBM InfoSphere Information Server
Version 8 Release 7

Guide to Managing Common Metadata



Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 103.

Contents

Chapter 1. Common metadata assets . . . 1

Types of common metadata assets	2
Implemented data resources	2
Logical data model assets	6
Physical data model assets	11
Business intelligence assets	14
Miscellaneous common metadata assets	18
Implementation relationships	19
Asset identity.	22
Duplicate assets	24
Disconnected assets.	25
Delete rules for assets	26

Chapter 2. Importing and managing assets by using IBM InfoSphere

Metadata Asset Manager 29

Overview of InfoSphere Metadata Asset Manager	29
Roles for using InfoSphere Metadata Asset Manager	31
Administering InfoSphere Metadata Asset Manager	31
Topology diagram for InfoSphere Metadata Asset Manager	32
Preparing to use InfoSphere Metadata Asset Manager	33
Metadata interchange servers	33
Import settings for InfoSphere Metadata Asset Manager	34
Importing and sharing assets	36
Preparing to import assets with InfoSphere Metadata Asset Manager	36
Importing metadata by using InfoSphere Metadata Asset Manager	37
Working with staged imports	41
Sharing imports to the metadata repository.	47
Command-line interface for InfoSphere Metadata Asset Manager	48
Creating an import area by using the command line	55
Escaping characters.	57
Managing common metadata assets	57

Finding metadata assets in the repository	57
Viewing asset details	59
Deleting metadata assets	61
Managing disconnected assets	62
Managing duplicate metadata assets	63
Setting or removing an implementation relationship	65
Merging assets	67
Deleting disconnected link points	69

Chapter 3. Exchanging metadata by using IBM InfoSphere Metadata

Integration Bridges 71

How bridges work	71
Requirements for using bridges.	72
Which tool do I use to run the bridges?	73
Importing BI metadata from IBM Cognos	73
Bridges that use IBM MetaBroker technology	79
Exporting and importing metadata by using Import Export Manager	84
Exporting metadata by using Import Export Manager	84
Importing metadata by using Import Export Manager	85
Importing metadata by using the command line	86
Importing metadata by using the Designer client	94

Contacting IBM 97

Accessing product documentation. 99

Product accessibility 101

Notices and trademarks 103

Index 107

Chapter 1. Common metadata assets

Common metadata assets are stored in the metadata repository and shared between tools in the IBM® InfoSphere® Information Server suite. Common metadata assets include logical and physical data model assets, implemented data resources, and business intelligence (BI) assets.

You can use bridges and connectors to import common metadata assets. Then you can work with the common metadata assets in suite tools:

- In InfoSphere DataStage® and QualityStage® Designer, you can use table definitions that are created from database tables and data file structures.
- In InfoSphere Information Analyzer, you can analyze database columns, data file fields, and design columns.
- In InfoSphere FastTrack, you can use database tables, data files structures, and design tables to create mappings for jobs.
- In InfoSphere Metadata Workbench, you can create reports that trace data lineage from database tables and columns, across multiple jobs, and into the BI reports that are based on database tables. You can trace a database table back to the design table that it implements and to the logical entity that was the source of the design table.
- In InfoSphere Business Glossary, you can create categories and terms that add business meaning to common metadata assets.
- In InfoSphere Metadata Asset Manager, you can analyze, preview, and refine your common metadata imports before you share them to the metadata repository. You can search or browse for common metadata assets in the metadata repository, find and eliminate duplicates, and delete disconnected assets.
- By using the istool command line, you can transfer common metadata assets to other installations of InfoSphere Information Server, for example, between development, test, and production environments. For information on using the istool command line, see the *IBM InfoSphere Information Server Administration Guide*.

Related information:

Overview

You use InfoSphere Metadata Asset Manager to import and manage common metadata assets, which are vital to all components of the InfoSphere Information Server suite. You can import assets to a staging area before you share them to the metadata repository. In the metadata repository, you can browse and search for common metadata assets, set implementation relationships between them, and merge duplicates.

Managing common metadata assets

You can use the **Repository Management** tab in InfoSphere Metadata Asset Manager to view, compare, merge, and delete assets that are stored in the metadata repository. You can also set implementation relationships between logical and physical data models and implemented data resources.

Types of common metadata assets

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Implemented data resources

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

Implemented data resources include the table-level and column-level metadata that IBM InfoSphere Information Server suite tools use in jobs and analyses.

Implemented data resources represent actual databases and data files, unlike physical data models, which represent designs for storing data. Implemented data resources can implement physical data models and logical data models.

You can import or create implemented data resources in the following ways:

- Import implemented data resources from third-party tools, files, and databases by using bridges and connectors.
- Create shared database tables from table definitions in InfoSphere DataStage and QualityStage.

In previous versions of InfoSphere Information Server, implemented data resources were called *physical data resources (PDR)*.

Asset types

The following table lists and defines the types of implemented data resources that can be stored in the metadata repository. Each asset has a unique identity that is determined by the identity components.

Table 1. Implemented data resources

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Host	A computer where a database or data file exists.	Host name	Database and data file Hosts can also contain assets that are not implemented data resources, such as jobs, connectors, and data connections.
 Database	A relational storage collection that is organized by schemas and procedures. A database stores data that is represented by tables.	<ul style="list-style-type: none"> • Database name • DBMS name • Database server name • Identity of the host <p>Note: The DBMS name and database server name components are not displayed as part of the identity string in user interfaces or on the command line.</p>	Database schema
 Database schema	A named collection of related database tables and integrity constraints. A schema defines all or a subset of the data that is in a database. A database schema can implement a logical data model or a physical data model.	<ul style="list-style-type: none"> • Schema name • Identity of the database <p>If a database schema contains a subschema, the identity of the subschema is the following:</p> <ul style="list-style-type: none"> • Subschema name • Identity of the containing schema 	Database table, stored procedure, and database domain Database schemas can also contain subschemas
 Database table	A structure for representing and storing data objects in a database. A database table can implement a logical entity or a design table.	<ul style="list-style-type: none"> • Database table name • Identity of the schema 	Database column, candidate key, and foreign key

Table 1. Implemented data resources (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Database column	A column in a database table. A database column can implement an entity attribute or a design column.	<ul style="list-style-type: none"> • Database column name • Identity of the database table 	
 Candidate key	A set of one or more columns that uniquely identifies the rows of a database table. Every database table must have at least one candidate key. One of them is designated as the primary key. All others are called alternative keys.	<ul style="list-style-type: none"> • Candidate key name • Identity of the database table 	
 Foreign key	A non-unique identifier that defines a relationship between two database tables. A foreign key in one table typically matches one of the candidate keys in the related table.	<ul style="list-style-type: none"> • Foreign key name • Identity of the database table 	
 Stored procedure	A procedure that is defined and stored within a database to retrieve or manipulate data in that database, or to enforce constraints. Stored procedures can implement design stored procedures.	<ul style="list-style-type: none"> • Stored procedure name • Identity of the schema 	Stored procedure parameter
 Stored procedure parameter	A parameter that is used by a stored procedure.	<ul style="list-style-type: none"> • Stored procedure parameter name • Identity of the stored procedure 	
 Data file	An information asset that represents a collection of fields that is stored in a single file. This asset could be a sequential file (a flat file that has no hierarchical structure) or a complex flat file (a file that has hierarchical structure). Examples of complex flat files include, COBOL copybooks and XML files. A data file can implement a physical data model.	<ul style="list-style-type: none"> • Data file name • Data file path • Identity of the host 	Data file structure

Table 1. Implemented data resources (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Data file structure	A collection of related fields in a data file. A data file structure is the file equivalent of a database table. A data file structure can implement a design table.	<ul style="list-style-type: none"> • Data file structure name • Identity of the data file 	Data file field
 Data file field	A field within a data file structure. A data file field is equivalent to a database column. It is the smallest data unit that is used to store the data values of an object. A data file field can implement a design column.	<ul style="list-style-type: none"> • Data file field name • Identity of the data file structure 	
 Database domain	A user-defined datatype that is contained by a database schema. Database domains can implement design domains and logical domains.	<ul style="list-style-type: none"> • Database domain name • Database domain name qualifier • Identity of database schema 	Data file field
 Data item definition	An information asset that represents user-defined types and intermediate elements in the hierarchy of complex data structures. Examples are COBOL structured fields, SAP intermediate segments in IDoc structures, and XML type structures.	<ul style="list-style-type: none"> • Data item definition name • Data item definition name qualifier 	Domain data field
 Domain data field	A data field that is contained by a data item definition or by a physical domain.	<ul style="list-style-type: none"> • Domain data field name • Identity of data item definition or physical domain 	

Related concepts:

“Logical data model assets”

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

“Physical data model assets” on page 11

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

“Asset identity” on page 22

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

“Implementation relationships” on page 19

You can set and explore relationships between logical data models, physical data models, and implemented data resources.

Logical data model assets

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

A logical data model can be represented graphically in a logical data model diagram.

Logical data models can be implemented by both physical data models and database schemas. You can use InfoSphere Metadata Asset Manager to set implementation relationships between logical data models, physical data models, database schemas, and their contained assets.

You can use bridges to import logical data models from design tools such as InfoSphere Data Architect and CA ERwin Data Modeler.

If you import logical data models, you have an end-to-end view of the metadata assets that control data flow. You can trace a logical entity to the table that implements the logical entity, through the jobs that use the columns of the table, and on to the business intelligence (BI) report that is based on the table.

Asset types

The following table lists and defines the types of logical data model assets that are stored in the metadata repository.

Table 2. Logical data model assets

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Logical data model	<p>A logical representation of the data objects that are related to a business domain and the rules or constraints that govern their associations in real-world applications. Logical data models consist of a set of entities and relationships. A logical data model can be implemented by a physical data model or a database schema.</p>	<ul style="list-style-type: none"> • Logical data model name • Model namespace: a value specified during import to help uniquely identify the logical data model <p>If a logical data model has a submodel, the identity components of the submodel are the following:</p> <ul style="list-style-type: none"> • Name of the contained logical data model • Namespace of the contained logical data model • Identity of the containing logical data model 	<p>Diagram, subject area, logical entity, logical relationship, entity generalization hierarchy, and logical domain</p> <p>Note: Logical data models can also contain submodels.</p>
 Diagram in logical data model	<p>A graphical representation of a logical data model.</p>	<ul style="list-style-type: none"> • Diagram name • Identity of the logical data model 	
 Diagram in subject area	<p>A graphical representation of subject area.</p>	<ul style="list-style-type: none"> • Diagram name • Identity of the subject area 	

Table 2. Logical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Subject area	A grouping of related logical entities that focus on a particular business area. A logical entity can be included in more than one subject area to better differentiate it from other logical entities in the logical data model. Subject areas can be represented graphically in subject area diagrams.	<ul style="list-style-type: none"> • Subject area name • Identity of the logical data model 	<p>A subject area can include but does not contain logical entities, logical relationships, and entity generalization hierarchies. Deleting the subject area does not delete assets of these types.</p> <p>A subject area can contain a diagram.</p>
 Logical entity	An asset that represents the data structure in the logical data model. A logical entity defines entity attributes, entity keys, and entity constraints. A logical entity can be implemented by a design table in a physical model or by a database table.	<ul style="list-style-type: none"> • Logical entity name • Identity of the logical data model 	Entity attribute, entity key, and entity constraint
 Entity attribute	A relevant property or characteristic of an entity that defines the meaning and purpose of a unit of data. An entity attribute can be implemented by a design column in a physical data model or by a database column.	<ul style="list-style-type: none"> • Entity attribute name • Identity of the logical entity 	Validation constraint
Validation constraint	A validation constraint is an abstract supertype that can be a validation rule, validation range, or validation list.	See validation rule, validation range, or validation list.	
 Validation rule	An expression that defines the valid values for an entity attribute or a logical domain.	<ul style="list-style-type: none"> • Validation rule name • Validation rule name qualifier • Identity of the logical data model 	

Table 2. Logical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Validation range	A range of values that defines the valid values for an entity attribute or a logical domain.	<ul style="list-style-type: none"> • Validation range name • Validation range name qualifier • Identity of the logical data model 	
 Validation list	A list of discrete values that defines the valid values for an entity attribute or a logical domain.	<ul style="list-style-type: none"> • Validation list name • Validation list name qualifier • Identity of the logical data model 	Validation value
 Validation value	A discrete value in a validation list.	<ul style="list-style-type: none"> • Validation value name • Validation value sequence • Identity of the validation list 	
Entity key	A semantic identifier that consists of one or more entity attributes. An entity key is an abstract supertype that can be a unique key, a reference key, or an inversion entry.	See unique key, reference key, or inversion key.	
 Unique key	A key that specifies the values of the entity attributes that uniquely identify the instances of the logical entity. A unique key can be implemented as a candidate key in a design table or a database table.	<ul style="list-style-type: none"> • Unique key name • Identity of the logical entity 	
 Reference key	A key that specifies a referential integrity constraint that is associated with a relationship. A reference key can be implemented as a foreign key in a design table or a database table.	<ul style="list-style-type: none"> • Reference key name • Identity of the logical entity 	

Table 2. Logical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Inversion key	A non-unique logical identifier that defines an inversion entry. An inversion key can be implemented by an index in a design table or a database table.	<ul style="list-style-type: none"> • Inversion key name • Identity of the logical entity 	
 Entity key component	An association between an entity key and one of its attributes.	<ul style="list-style-type: none"> • Identity of the entity key • Identity of the entity attribute 	
 Entity constraint	A business rule in the form of a constraint that is attached to a logical entity.	<ul style="list-style-type: none"> • Entity constraint name • Identity of the logical entity 	
 Logical relationship	An asset that represents the set of business rules that define the associations between two logical entities. A logical relationship can be implemented by a design foreign key or a foreign key for a database table.	<ul style="list-style-type: none"> • Logical relationship name • Identity of the logical model • Identity of the owner entity <p>The identity of the owner entity is not a mandatory component. Not all design tools use this concept.</p>	Relationship end
 Relationship end	A connection between the entity and the relationship that expresses how the logical entity participates in the relationship.	<ul style="list-style-type: none"> • Name of the relationship end • Identity of the relationship • Identity of the logical entity 	
 Entity generalization hierarchy	An asset that represents the inheritance associations that classify logical entities into subtypes and supertypes. A hierarchy supertype is a logical entity that is the supertype or parent entity in the hierarchy.	<ul style="list-style-type: none"> • Name of the entity generalization hierarchy • Identity of the parent entity • Identity of the logical model 	Hierarchy supertype and hierarchy subtype

Table 2. Logical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Hierarchy subtype	An asset that connects a logical entity as a subtype in an entity generalization hierarchy. An entity can be a subtype in more than one hierarchy.	<ul style="list-style-type: none"> • Name of the hierarchy subtype • Identity of the entity generalization hierarchy • Identity of the logical entity that is the subtype 	
 Logical domain	A user-defined data type or global attribute that can be reused in multiple logical entities. A logical domain can be implemented by a physical domain or a database domain.	<ul style="list-style-type: none"> • Logical domain name • Logical domain name qualifier • Identity of the logical data model 	

Related concepts:

“Implemented data resources” on page 2

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

“Physical data model assets”

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

“Asset identity” on page 22

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

“Implementation relationships” on page 19

You can set and explore relationships between logical data models, physical data models, and implemented data resources.

Physical data model assets

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

Physical data models are typically generated from logical data models by using modeling tools, although they can be reverse-engineered from existing databases. Physical data models represent designs for storing data, unlike implemented data

resources, which represent actual databases and data files. A physical data model can implement multiple logical data models and can be implemented by multiple database schemas and data files.

You can use bridges to import physical data models from design tools such as IBM InfoSphere Data Architect and CA ERwin Data Modeler. When you import a physical data model by using a bridge, a corresponding database schema that implements the physical data model is created in the metadata repository for use in InfoSphere DataStage and QualityStage jobs.

You can use InfoSphere Metadata Asset Manager to set implementation relationships between logical data models, physical data models, database schemas, data files, and their contained assets.

Asset types

The following table lists and defines the types of physical data model assets that can be stored in the metadata repository.

Table 3. Physical data model assets

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Physical data model	A design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. A physical data model can implement a logical data model and can be implemented by a database schema or a data file.	<ul style="list-style-type: none"> Physical data model name Model namespace: a value specified during import to help uniquely identify the physical data model 	Design table, design stored procedure, and physical domain
 Design table	An asset that represents a table structure in the physical data model. The design table defines the design column, the design candidate key, and the design foreign key. A design table can implement a logical entity and can be implemented by a database table or data file structure.	<ul style="list-style-type: none"> Design table name Identity of the physical data model 	Design column, design candidate key, and design foreign key

Table 3. Physical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Design column	A relevant property or characteristic of a design table that defines the meaning and purpose of a unit of data. A design column can implement an entity attribute and can be implemented by a database column or a data file field.	<ul style="list-style-type: none"> • Design column name • Identity of the design table 	
 Design candidate key	A unique semantic identifier that defines the identity constraint of a design table.	<ul style="list-style-type: none"> • Design candidate key name • Identity of the design table 	
 Design foreign key	A non-unique identifier that defines a relationship between two design tables.	<ul style="list-style-type: none"> • Design foreign key name • Identity of the design table 	
 Design stored procedure	An asset that represents the stored procedure structure in the physical data model. The design stored procedure also defines the design stored procedure parameters. A design stored procedure can be implemented by a stored procedure.	<ul style="list-style-type: none"> • Design stored procedure name • Identity of the physical data model 	Design stored procedure parameter
 Design stored procedure parameter	A parameter that is used by a design stored procedure.	<ul style="list-style-type: none"> • Design stored procedure parameter name • Identity of the design stored procedure 	
 Physical domain	A user-defined data type or global attribute that can be reused in multiple design tables. A physical domain can implement a logical domain and can be implemented by a database domain.	<ul style="list-style-type: none"> • Physical domain name • Physical domain name qualifier • Identity of the physical data model 	Domain data field

Table 3. Physical data model assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Domain data field	A data field that is contained by a data item definition or by a physical domain.	<ul style="list-style-type: none"> • Domain data field name • Identity of physical domain or data item definition 	

Related concepts:

“Implemented data resources” on page 2

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

“Logical data model assets” on page 6

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

“Asset identity” on page 22

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

“Implementation relationships” on page 19

You can set and explore relationships between logical data models, physical data models, and implemented data resources.

Business intelligence assets

Business intelligence (BI) assets are used by BI tools to organize reports and models that provide a business view of data. These assets include BI reports, BI models, BI collections, and cubes.

You can use bridges to import BI assets from tools such as IBM Cognos® and SAP BusinessObjects.

It is good practice when importing BI assets to simultaneously import the database tables that BI reports are based on. You can then use InfoSphere Metadata Workbench to create data lineage reports that show the relationship between the database tables, the jobs that use the database tables, and the BI reports that are based on the tables.

You can use InfoSphere Metadata Asset Manager to browse and delete BI assets and manage duplicate BI assets.

Asset types

The following table lists and defines the types of BI assets that are stored in the metadata repository of InfoSphere Information Server.

Table 4. BI assets

Asset type	Definition	Components of the identity of the asset	Contained asset types
 BI server	<p>When a BI tool supports multiple servers on a single host computer, the BI server value is the name of the source tool server. When a BI tool supports a single server per host computer, the BI server value is the name or IP address of the host system.</p> <p>BI servers are displayed in InfoSphere Metadata Workbench and on the Import tab of InfoSphere Metadata Asset Manager.</p>	<ul style="list-style-type: none"> • BI server name • BI server namespace 	BI folder
 BI folder	<p>The folder structure that contains BI models, or BI reports, or both in the source tool. BI folders can also contain other BI folders.</p> <p>BI folders are displayed in InfoSphere Metadata Workbench and on the Import tab of InfoSphere Metadata Asset Manager.</p>	<ul style="list-style-type: none"> • BI folder name • Identity of the BI server, or, for subfolders, the identity of the BI folder that contains the subfolder 	BI folder, BI model, and BI report
 BI model	<p>A grouping of BI data collection views that are relevant to a BI application.</p>	<ul style="list-style-type: none"> • BI model name • BI model namespace 	Cube, BI collection, BI join, BI hierarchy, and BI filter
 BI collection	<p>A data structure that provides a view of data that is stored in databases and files. In dimensional modeling, these structures are known as dimensions and fact tables. BI collections are the data sources of BI reports.</p>	<ul style="list-style-type: none"> • BI collection name • BI collection namespace • Identity of the BI model that contains the collection, or, for subcollections, the identity of the BI collection that contains the subcollection 	BI collection member, BI level, BI hierarchy, and BI filter Note: BI collections can contain other BI collections.

Table 4. BI assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 BI collection member	<p>The basic abstraction of a data value that is projected from a database column. BI collection members define the structure of the collection that owns them. There are two types of members: regular and measure. Regular members are dimension attributes that describe the characteristics and semantics of the owner collection. Measures represent analytic values that define a measurement entity in a fact collection.</p>	<ul style="list-style-type: none"> • BI collection member name • BI collection member namespace • Identity of the BI collection 	
 BI level	<p>An asset that defines a logical step in the order of a BI hierarchy. A BI level consists of one or more BI collection members of the same BI collection that are related and function as a logical unit.</p>	<ul style="list-style-type: none"> • BI level name • BI level namespace • Identity of the BI collection 	BI level element
 BI level element	<p>An associative class that assigns a BI collection member to a specific level within the collection.</p>	<ul style="list-style-type: none"> • Identity of the BI collection member • Identity of the BI level 	
 Cube	<p>A subset of a BI model that consists of a set of related analytic values that share the same dimensionality.</p>	<ul style="list-style-type: none"> • Cube name • Cube namespace • Identity of the BI model 	Cube dimension and cube measure
 Cube dimension	<p>An associative class that connects a cube to dimensions in the BI collection that are relevant to the analytic values of the cube. A cube dimension references the BI collection from which the dimension is derived and the relevant dimension hierarchy of the cube.</p>	<ul style="list-style-type: none"> • Identity of the cube • Identity of the BI collection 	

Table 4. BI assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 Cube measure	An associative class that connects a cube to BI collection members that are measures.	<ul style="list-style-type: none"> • Identity of the cube • Identity of the BI collection member 	
 BI filter	A filtering constraint on the source data that is viewed through a BI collection. Filters are either local or global. A local filter is owned by a single BI collection. A global filter is owned by the BI model and by one or more collections.	<ul style="list-style-type: none"> • BI filter name • BI filter namespace • Either the identity of the containing BI model or the containing BI collection 	
 BI hierarchy	An organizational structure that defines an ordering or relationship of data within a BI collection.	<ul style="list-style-type: none"> • BI hierarchy name • BI hierarchy namespace • Either the identity of the containing BI model or the containing BI collection 	BI hierarchy member
 BI hierarchy member	An asset that orders BI levels within a hierarchy structure.	<ul style="list-style-type: none"> • Identity of the BI hierarchy • Identity of the BI level 	
 BI join	An asset that joins two database tables (a physical join) or two BI collections (a logical join). The physical join defines the data source of a BI collection and the logical join is used in a star schema between fact and dimension collections.	<ul style="list-style-type: none"> • BI join name • BI join namespace • BI join condition • Identity of the BI model 	BI report section, BI report field, and BI report query
 BI report	A business intelligence report that is based on information in a database or a BI model.	<ul style="list-style-type: none"> • BI report name • BI report namespace 	BI report section and BI report query

Table 4. BI assets (continued)

Asset type	Definition	Components of the identity of the asset	Contained asset types
 BI report section	An asset that defines the presentation of a section of a BI report. A BI report section is a grouping of BI report fields.	<ul style="list-style-type: none"> • BI report section name • BI report section namespace • Either the identity to the BI report or the identity of the containing BI report section 	BI report field
 BI report field	A field in a BI report that is typically based on a database column. Some BI report fields, including page numbers and section headers, are not data fields.	<ul style="list-style-type: none"> • BI report field name • Identity of the BI report section 	
 BI report query	A query on a database or a BI model whose result set populates a BI report section.	<ul style="list-style-type: none"> • BI report query name • BI report query namespace • Identity of the BI report 	BI report query item
 BI report query item	An asset that defines the data values that are associated with a BI report field by defining a column in a BI report query.	<ul style="list-style-type: none"> • BI report query item name • Identity of the BI report query 	

Related concepts:

“Asset identity” on page 22

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Miscellaneous common metadata assets

Common metadata assets include several asset types that do not fit the major categories. These assets are data connections, custom attributes, and contract libraries.

Asset types

The following table lists and defines the miscellaneous common metadata assets that are stored in the metadata repository. You can import and export assets of these types by using the istool command line.

Table 5. Miscellaneous common metadata assets

Asset type	Definition	Components of the identity of the asset
Data connection	A connection for accessing a database or file. For example, an ODBC or Oracle connection.	<ul style="list-style-type: none"> • Data connection name • Identity of the database or data file • Identity of the host
Custom attribute definition	A custom attribute that is created in IBM InfoSphere Metadata Workbench. Custom attributes are created by users as properties of implemented data resources, logical data resources, extended data resources, and mappings.	Custom attribute definition name
Contract library	A group of related XML schemas that are imported and used by the XML stage to transform data.	Contract library name

Implementation relationships

You can set and explore relationships between logical data models, physical data models, and implemented data resources.

InfoSphere Information Server tools analyze and transform data that comes from databases and data files. The metadata that describes databases and data files and their contents is stored in the metadata repository as a hierarchy of *implemented data resources*. In this hierarchy, a host computer contains one or more databases, which contain schemas. The database schemas contain tables, which contain columns. A host can also contain a data file, which contains data file structures. Data file structures contain data file fields, which are the equivalent of columns.

The organization within these data hierarchies is often determined by decisions that data modelers make when they use design tools such as CA ERwin Data Modeler or InfoSphere Data Architect. As a data modeler, you use a modeling tool to create *logical data models* that capture the business definition of information assets and the relationships between them. By using the same tool, you can then implement logical data models as *physical data models*, transforming the logical concepts into the design for a database or sometimes the design for a data file. Finally, you can implement the physical data models as schemas in real-world databases, or as data files. You can also implement the logical data models directly as database schemas.

Although database and data file assets are placed in the category of *implemented data resources*, they do not always have implementation relationships to data model assets. Some databases, for example, are constructed without reference to a logical data model or a physical data model. In other cases, the databases are imported separately from the data models and the relationship between them is not established unless you set it manually in InfoSphere Metadata Asset Manager.

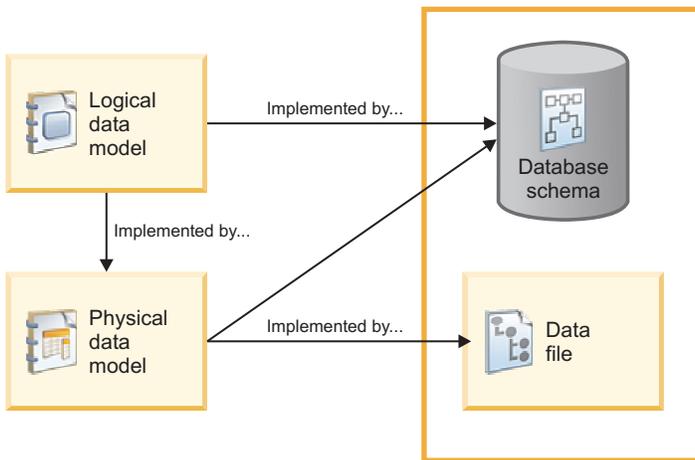


Figure 1. Implementation relationships between types of assets

By importing and storing the metadata for logical and physical data models and implemented data resources, InfoSphere Information Server provides a unified view of your data flow, from the logical conception through multiple transformations in jobs.

Setting implementation relationships during bridge imports

When you use a bridge to import a logical data model and a related physical data model from a design tool, the bridge automatically sets implementation relationships between the corresponding assets in the models. For example, entities and attributes in the logical data model are connected by *implementation relationships* to corresponding design tables and design columns in the physical data model.

An implementation relationship is a two-way relationship:

- A logical data model is *implemented by* a physical data model.
- A physical data model *implements* a logical data model.

When you import a physical data model from a design tool, the import bridge also creates an additional set of implemented data resources, based on identity parameters that you specify during the import. This process transforms the physical data model into a database schema with database tables and database columns that correspond to the design tables and design columns in the model. Both the implemented data resources and the physical data model assets are saved to the metadata repository. The database tables and columns can be used by developers of InfoSphere DataStage and QualityStage jobs. The bridge sets implementation relationships between the physical data model assets and the corresponding assets in the database schema.

You can use these implementation relationships to trace the source of your metadata back to its logical origin in a design tool. For example, by using data lineage in InfoSphere Metadata Workbench, you can trace the definition of a database column back to the design column that it implements. For each column, you can view the related entity attribute or design column. You can also view and follow these relationships in InfoSphere Metadata Asset Manager. This gives you an end-to-end perspective on your information flow that can alert you to changes in your data structure. For example, if you change the properties of an entity attribute in the logical data model, you can see which database columns are

affected downstream by the change. Or you can trace implementation relationships to ensure that all the entity attributes in a logical data model are implemented as database columns.

Additional ways of setting implementation relationships

You can manually set relationships between logical data models, physical data models, and implemented data resources that are stored in the metadata repository. Setting implementation relationships manually is useful when you import logical data models and physical data models separately, or when you import database or data file assets and you want to indicate that they implement logical or physical data model assets.

On the **Repository Management** tab of InfoSphere Metadata Asset Manager, you can specify that a logical data model is implemented by one or more physical data models or database schemas. You can also specify that a physical data model is implemented by one or more database schemas or data files. You can set additional implementation relationships between the contained assets. For example, if you create an implementation relationship between a logical data model and a database schema you can specify that a logical entity in the data model is implemented by one or more database tables in the schema.

When you analyze a data source in InfoSphere Information Analyzer, you can use the analysis results to create a physical data model. This sets implementation relationships between corresponding assets in the physical data model and the database schema that was analyzed. You can then create a rule, associate the physical data model assets with default rule bindings, and use the implementation relationships to view which database tables or database columns you can bind to.

Related concepts:

“Logical data model assets” on page 6

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

“Physical data model assets” on page 11

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

“Implemented data resources” on page 2

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

Related information:

Setting or removing an implementation relationship

Logical data models and physical data models can be implemented by implemented data resources. You can set or remove an implementation relationship between either logical data model assets or physical data model assets and implemented data resources. You can also set or remove an implementation relationship between logical data model assets and physical data model assets.

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

When you use the suite tools of InfoSphere Information Server to work with assets that are stored in the metadata repository, it is important that each asset has a unique identity. This unique identity ensures, for example, that the database column that you use in a job is the same database column that you assign a steward to or that is assigned to a glossary term.

The identity of an asset in the metadata repository is often based on its name and its containment hierarchy. For example, the identity of a database table is the name of the table plus the identity of the database schema, database, and host computer that contain it. You cannot have two different database tables with the same name in the same schema. If you know the identity of the asset you can more easily browse to the asset or search for it.

In some cases, the *namespace* or *name qualifier* of an asset is also used as part of its identity. These two identity components are usually optional values that are provided by some of the tools from which the assets are imported. For example, some BI tools provide namespace information in the form of folder paths for certain BI assets that you import into the metadata repository. This additional value helps uniquely identify the BI asset. When you import logical data models or physical data models, you must specify a namespace value that helps to uniquely identify the data model. This prevents data models from being overwritten when you import a different data model that has the same name.

The identity components of each type of common metadata asset are listed in the separate topics on those asset types. Knowing the identity of an asset type can help you understand the browsing structure in suite tools. Figure 1 illustrates that the

browsing path to a database table on the Repository Management tab of InfoSphere Metadata Asset Manager is the same as the identity property that is listed for the database table.

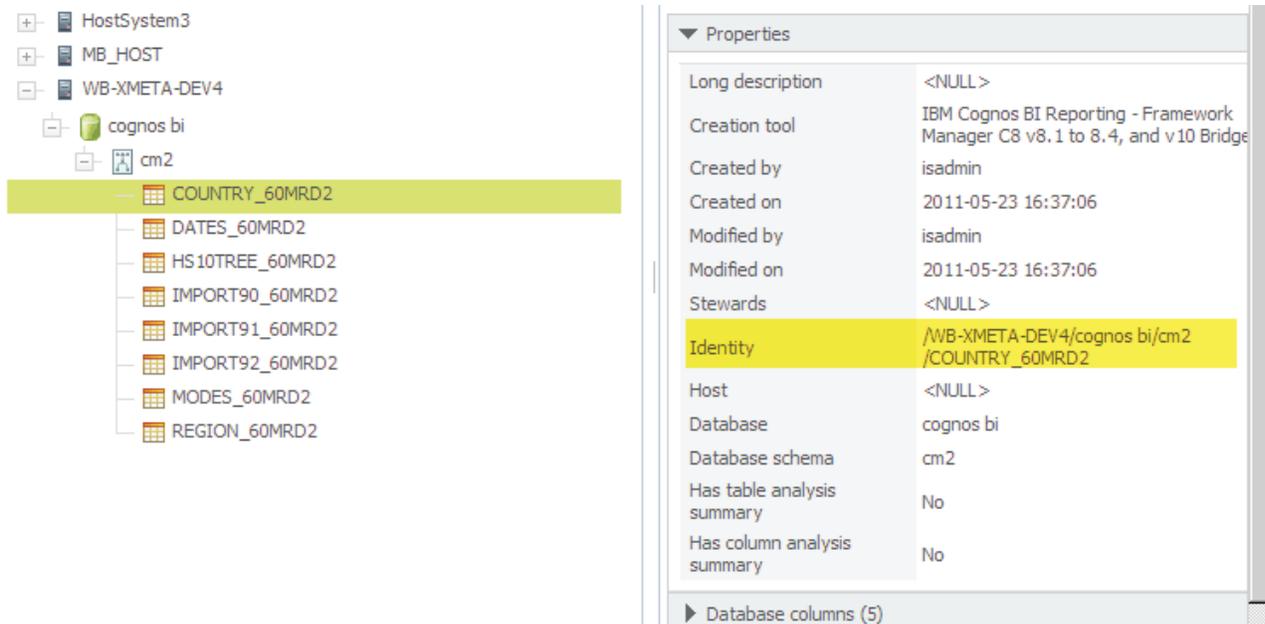


Figure 2. Browse path and identity of a database table

In the **Properties** list in the graphic above, the different components of the identity of the database table are separated by forward slashes in the form `host_name/database_name/database schema_name/database_table_name`. This representation is called an *identity string*. Identity strings are used to specify individual assets when you use the `istool` command line to import, export, query, or delete common metadata assets.

Related concepts:

“Implemented data resources” on page 2

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

“Logical data model assets” on page 6

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

“Physical data model assets” on page 11

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

“Business intelligence assets” on page 14

Business intelligence (BI) assets are used by BI tools to organize reports and models that provide a business view of data. These assets include BI reports, BI models, BI collections, and cubes.

Duplicate assets

Duplicate assets are assets that have the same identity or assets that have different identities but can be considered identical. You can manage duplicate assets by merging or deleting them.

Under most circumstances, suite tools prevent you from creating duplicate assets or importing duplicate assets into the metadata repository. However, there are some cases where duplicate assets might occur in the metadata repository. Consider these cases where duplicates might occur:

- When you use two different types of import tools, for example a connector and a bridge, to import an asset from the same source at different times, one tool might report the identity differently from the other tool. For example, the database management system (DBMS) might be interpreted differently by two import tools, and as a result the same imported assets are contained by two different databases.
- When you import the same asset twice from different host computers or specify different identity parameters, you could end up with the same database tables being part of a different host, database, or schema.
- When you override the default administration setting in InfoSphere Metadata Asset Manager that prevents the import of duplicates, you can create duplicates in the metadata repository.

There are two basic types of duplicates in the metadata repository:

- Duplicates that have the same identity. For example, a database table that has the same name as another database table that is contained in the same database schema.
- Duplicates that have different identities but can really be considered the same object. For example, two database tables that have the same columns and perhaps the same name, but are contained in different schemas, databases, or hosts.

When you import assets by using InfoSphere Metadata Asset Manager, you are alerted to any duplicates in the import event that have the same identity. You can inspect those duplicates in the staging area. But unless your administration settings

allow you to import duplicates, you must fix the duplicates in the source tool and reimport before the import event can be shared with the metadata repository. If you choose to allow duplicates during imports, be aware that every reimport of the same content increases the number of duplicates in the repository.

When duplicates exist in the metadata repository, you can manage them by using the **Repository Management** tab of InfoSphere Metadata Asset Manager. By clicking **Manage Duplicates**, you can view and compare assets of specific types that have the same name. You can merge duplicate databases that are contained by the same host computer and duplicate database schemas that are contained by the same database. Contained database tables and database columns that have the same name are also merged. You can delete duplicate assets of all types.

You can also browse the repository to inspect assets that you think might be duplicates but do not have the same name.

To help you decide what action to take, the **Repository Management** tab provides information about the usage of assets. Before you merge or delete a duplicate asset, check the *usage* of the asset by the suite tools. For example, if a duplicate asset that you want to delete is assigned to a term, you might want to assign the surviving asset to the same term. If a duplicate asset is used in a job, you can change the job properties in InfoSphere DataStage and QualityStage to use the surviving asset instead of the deleted asset.

Related information:

Managing duplicate metadata assets

You can compare, merge, or delete duplicate assets in the metadata repository. Deleting duplicate assets helps ensure that suite tool users can choose the correct assets to use.

Merging assets

After browsing for assets or managing duplicate assets, you can merge two assets into one asset. Merging duplicate metadata assets makes it easier for suite tool users to choose the correct metadata to use in jobs. You can merge databases or database schemas.

Disconnected assets

Assets that become disconnected from their immediate parent asset in the identity string are called *disconnected assets*.

Assets can become disconnected when parent assets are deleted and the child asset has relationships to other assets that prevent its deletion.

Disconnected assets can cause problems when they are used in suite tools because they do not have complete identities. The solution is to delete the disconnected asset and reimport the original asset. You can browse and delete disconnected assets by using the **Repository Management** tab of InfoSphere Metadata Asset Manager.

Be sure to view the usage of the disconnected asset so that you can recreate the necessary relationships to other suite tools when you reimport the asset.

Related information:

Managing disconnected assets

You can view the details of a disconnected implemented data resource or business intelligence (BI) asset or delete disconnected assets from the metadata repository. Disconnected assets do not have a complete identity because one or more assets in the identity hierarchy do not exist.

Delete rules for assets

Before deleting an asset from the metadata repository, make sure that you understand these delete rules for the asset.

Delete rules for implemented data resources

The following table describes the delete rules that apply when you delete an implemented data resource.

Table 6. Delete rules for implemented data resources

Deleted asset	Delete rule
Host	You cannot delete a host if it contains any of the following assets: databases, data files, jobs, connectors, or data connections.
Database	All database schemas that are contained by the database are also deleted.
Database schema	All database tables and stored procedures that are contained by the database schema are also deleted.
Database table	All columns, candidate keys, and foreign keys that are contained by the database table are also deleted. Any table analysis and column analysis summaries based on the database table are also deleted. You cannot delete a database table if it has a foreign key constraint that points to a primary key in another table.
Data file	All data file structures and data file fields that are contained by the data files are also deleted.
Data file structure	All data file fields that are contained by the data file structure are also deleted.

Delete rules for BI assets

The following table describes the delete rules that apply when you delete a BI asset.

Table 7. Delete rules for BI assets

Deleted asset	Delete rule
BI model	All BI dimensions, BI collections, BI joins, BI hierarchies, and cubes that are contained by the BI model are also deleted.

Table 7. Delete rules for BI assets (continued)

Deleted asset	Delete rule
BI report	All BI report queries and report layouts that are contained by the BI report are also deleted.
BI collection	All BI collection members, BI levels, and BI hierarchies that are contained by the BI collection are also deleted.
Cube	All BI dimensions and BI collection members that are contained by the cube are also deleted.
BI report query	All BI report query items that are contained by the BI report query are also deleted.

Delete rules for logical data models

The following table describes the delete rules that apply when you delete logical data model assets.

Table 8. Delete rules for logical data models

Deleted asset	Delete rule
Logical data model	All diagrams, subject areas, logical relationships, logical entities, and entity generalization hierarchies contained in the logical data model are also deleted.
Logical domain	You cannot delete a logical domain until all associated entity attributes are deleted.
Logical entity	All logical relationships, logical attributes, keys, and validation constraints associated with the logical entity are also deleted.
Entity generalization hierarchy	All subtype elements associated with the entity generalization hierarchy are deleted. Logical entities associated with the entity generalization hierarchy are not deleted.
Logical relationship	All relationship ends connected with the logical relationship are deleted. Logical entities associated with the relationship are not deleted.
Subject area	Deleting a subject area does not affect associated assets.
Diagram	Deleting a diagram does not affect associated assets.

Delete rules for physical data models

The following table describes the delete rules that apply when you delete physical data model assets.

Table 9. Delete rules for physical data models

Deleted asset	Delete rule
Physical data model	All physical domains, design tables, and design procedures contained in the physical data model are also deleted.
Physical domain	You cannot delete a physical domain until all associated attributes are deleted.
Design table	All design columns, design candidate keys, and design foreign keys associated with the design table are also deleted.
Design stored procedure	All design stored procedure parameters are also deleted.

Chapter 2. Importing and managing assets by using IBM InfoSphere Metadata Asset Manager

You can review and preview imports of metadata assets, and you can manage assets that are in the metadata repository.

Overview of InfoSphere Metadata Asset Manager

You use InfoSphere Metadata Asset Manager to import and manage common metadata assets, which are vital to all components of the InfoSphere Information Server suite. You can import assets to a staging area before you share them to the metadata repository. In the metadata repository, you can browse and search for common metadata assets, set implementation relationships between them, and merge duplicates.

When you *share* imports to the metadata repository, the imported assets are available to users of other suite tools. Other users can analyze the assets, use them in jobs, assign them to terms, or designate stewards for the assets. Until you share the import, the assets are not visible in the metadata repository and cannot be used by suite tools.

Common metadata assets include the following types:

- Implemented data resources, such as database tables, database columns, and data file fields
- Physical data model assets, such as design tables and design columns
- Logical data model assets, such as logical entities, entity attributes, and relationships
- Business intelligence (BI) assets, such as BI models, BI collections, and BI reports that are based on database tables

InfoSphere Metadata Asset Manager uses the following types of import tools:

- InfoSphere Metadata Integration Bridges, which import metadata from many types of tools, files, and databases, including IBM Cognos, CA ERwin, InfoSphere Data Architect, and SAP BusinessObjects. Some bridges import additional types of metadata, such as analysis information and user information.
- InfoSphere Information Server connectors, which import database metadata from databases such as Teradata, Oracle, and DB2®, and by using ODBC connections.

The tasks that you perform in InfoSphere Metadata Asset Manager are divided into three main areas, each of which has its own tab:

- **Import**, where you analyze and preview the contents of the import in a staging area before you share it to the metadata repository
- **Repository Management**, where you find, inspect, and manage assets that are in the metadata repository
- **Administration**, where you set import policies and specify the metadata interchange servers where bridges and connectors are installed

Importing assets to the staging area

The contents and parameters of each import are saved to an *import area* in the staging area, regardless of whether you choose to share the import to the metadata repository. When you run a new import, you create an import area that contains the *staged import*. When you reimport from an import area, it creates a new staged import in the same import area.

Importing assets involves the following tasks:

1. Naming the import area and choosing the metadata interchange server that you want to run bridges and connectors from. You can choose a local or remote computer.
2. Selecting a bridge or connector to import from the database, file, or tool that contains the assets that you want to import.
3. Specifying parameter values for the bridge or connector, and in some cases selecting the specific assets to import.
4. Specifying identity parameters if you are importing database assets. The information you enter in these parameters ensures that the database tables that you import are linked to the correct schema, database, and host computer.
5. Selecting whether to perform an express import or a managed import.

Express[®] import

The source metadata is imported to the staging area, where it is analyzed and previewed automatically. Your administrator's settings determine whether the import is shared automatically to the metadata repository or if you are required to view a preview before you manually share the import to the metadata repository.

Managed import

The source metadata is imported to the staging area, where you manually analyze the import contents and preview the result of sharing it to the metadata repository. When you are satisfied with the contents of the import, you share the import assets to the metadata repository.

Analyzing, previewing, and sharing to the metadata repository

In the staging area, you can analyze the import contents to see if there are duplicates or identity issues. If such issues exist, you can fix them in the source tool and then reimport the assets. Reimporting creates a new staged import in the same import area. You can compare the contents of the newest staged import with previous staged imports.

When the analysis results are satisfactory, you can preview the result of sharing the import to the metadata repository. You can browse to see which existing assets in the repository will be created and merged as a result of the import, and you can view the list of deleted assets. You can open InfoSphere Metadata Workbench, if it is installed, to further examine assets that are in the repository.

You can reimport and, before sharing, change the import parameters to ensure that you are importing the precise chosen content. When you are satisfied with the preview, you can share the staged import to the metadata repository. You can then browse the results of the sharing process.

Managing assets in the metadata repository

On the **Repository Management** tab, you can browse and search for common metadata assets.

If there are duplicate assets in the metadata repository, you can find them and then merge or delete them. If there are disconnected assets, which are assets that have become separated from their original containing assets, you can inspect and delete them.

You can view and set implementation relationships between logical data models, physical data models, database schemas, and data files. These relationships help you trace the evolution of assets from their conception in design tools to their implementation in databases and data files.

Roles for using InfoSphere Metadata Asset Manager

To use InfoSphere Metadata Asset Manager, you must have the role of Common Metadata User, Common Metadata Importer, or Common Metadata Administrator.

A suite administrator can assign the following roles on the **Administration** tab of the InfoSphere Information Server Web console.

Common Metadata User

This user can access the **Repository Management** tab to browse, search for, and inspect assets that are in the metadata repository.

Common Metadata Importer

On the **Import** tab, this user can perform the following tasks:

- Create import areas
- Import metadata to the staging area
- Work with staged imports that this user creates, including analyzing, previewing, reimporting, and sharing imports to the metadata repository
- This user can view, work in, and delete only those import areas that this user creates.

On the **Repository Management** tab, this user can browse, search for, and inspect assets that are in the metadata repository.

Common Metadata Administrator

On the **Administration** tab, this user can perform the following tasks:

- Specify import policies
- Configure metadata interchange servers

On the **Import** tab, this user has all the privileges of the Common Metadata Importer. In addition, this user can view, work in, and delete import areas that are created by any user.

On the **Repository Management** tab, this user can merge and delete assets and set implementation relationships. In addition, this user can browse, search for, and inspect assets that are in the metadata repository.

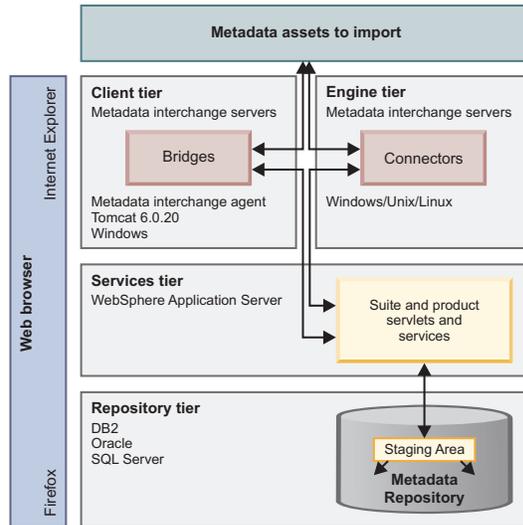
Administering InfoSphere Metadata Asset Manager

The Common Metadata Administrator configures InfoSphere Metadata Asset Manager for use and sets import policies.

Topology diagram for InfoSphere Metadata Asset Manager

The components of InfoSphere Metadata Asset Manager are distributed across multiple tiers. Components are installed automatically on the services, and repository tier, and manually on the client tier.

The following graphic displays the topology and import flow for InfoSphere Metadata Asset Manager.



InfoSphere Metadata Asset Manager is a web client that runs on supported Firefox and Internet Explorer browsers.

You import metadata from tools, databases, and files by using bridges or connectors.

To use bridges, you install the metadata interchange agent and bridges on a Microsoft Windows computer on the client tier. Apache Tomcat V. 6.0.20 is installed with the metadata interchange agent.

To use connectors, you install the connectors by installing InfoSphere DataStage and QualityStage on a Windows, Unix, or Linux computer on the engine tier.

Before you import metadata, the Common Metadata Administrator designates the computers that the bridges and connectors are installed on as metadata interchange servers.

Servlets and services for InfoSphere Metadata Asset Manager are installed on the services tier, along with suite services.

The staging area schema for InfoSphere Metadata Asset Manager is created automatically in the metadata repository when you install the suite and create the default DB2 database for the metadata repository. You must create it manually as part of configuring Oracle or SQL Server databases for the metadata repository.

When you import metadata by using bridges or connectors, the contents of the import are stored in the staging area, where you can analyze the assets and preview the results of sharing the import to the metadata repository. Until you

share the import, other users cannot see or work with the import contents. When you share the import to the metadata repository, other suite users can view and work with the imported assets.

Related reference:

“Metadata interchange servers”

Metadata interchange servers are computers that are used by InfoSphere Metadata Asset Manager to run imports with bridges and connectors. Before users can import metadata, the Common Metadata Administrator must define and enable one or more metadata interchange servers on the **Administration** tab of InfoSphere Metadata Asset Manager.

Preparing to use InfoSphere Metadata Asset Manager

Before users run imports in a new installation of InfoSphere Metadata Asset Manager, the Common Metadata Administrator must ensure that all necessary components and tools are installed and configured correctly. Ensure that the required roles are assigned so that users can import metadata and browse the metadata repository.

About this task

For information on installing InfoSphere Metadata Asset Manager, see the *IBM InfoSphere Information Server Planning, Installation, and Configuration Guide*.

Procedure

1. Assign the appropriate metadata user roles in the InfoSphere Information Server Web console.
2. On the **Administration** tab of InfoSphere Metadata Asset Manager, create and enable metadata interchange servers. Metadata interchange servers are the computers where the bridges and connectors are installed that you use to import metadata. You must have the role of Common Metadata Administrator to use the Administration tab of InfoSphere Metadata Asset Manager.
3. Ensure that all required software is in place on each metadata interchange server:
 - Ensure that each computer where bridges are installed meets the system requirements for using bridges. For a list of installation prerequisites, see the technote *System requirements for InfoSphere Metadata Integration Bridges*: <http://www.ibm.com/support/docview.wss?&uid=swg27021981>.
 - Check the documentation for the connectors that you use to ensure that you have met all of their prerequisites for use.
 - Ensure that you have installed any additional required software for each bridge that you use. Some bridges also require that you install the source tool on the same computer that the bridge is installed on. For a list of supported bridges and their individual prerequisites, see the technote *List of supported bridges for InfoSphere Information Server Version 8.7*: <http://www.ibm.com/support/docview.wss?&uid=swg27021980>.
4. On the **Administration** tab of InfoSphere Metadata Asset Manager, specify import settings.

Metadata interchange servers

Metadata interchange servers are computers that are used by InfoSphere Metadata Asset Manager to run imports with bridges and connectors. Before users can

import metadata, the Common Metadata Administrator must define and enable one or more metadata interchange servers on the **Administration** tab of InfoSphere Metadata Asset Manager.

You must have the role of Common Metadata Administrator to define, enable, or disable metadata interchange server.

To enable bridge imports, specify a computer where the metadata interchange agent is installed and specify the HTTPS port for the computer. Use the port number that was specified when the metadata interchange agent was installed. The default HTTPS port is 19443.

To enable connector imports, specify a computer where connectors are installed. Do not specify an HTTPS port if only connectors are installed on the metadata interchange server.

You can specify a single metadata interchange server for both bridges and connectors if the metadata interchange agent, bridges, and connectors are installed on the same computer. Specify the HTTPS port for this server.

When you enable a metadata interchange server, users with the Common Metadata Importer role can select it on the **New Import Area** screen. They can then use a bridge or connector that is installed on that computer to run an import.

Metadata interchange servers are listed as **in use** if one or more import areas have been created by using that server. You cannot delete a metadata interchange server if it is listed as in use.

If you disable a metadata interchange server, users cannot reimport staged imports that were created by using that server.

Related concepts:

“Topology diagram for InfoSphere Metadata Asset Manager” on page 32
The components of InfoSphere Metadata Asset Manager are distributed across multiple tiers. Components are installed automatically on the services, and repository tier, and manually on the client tier.

Related tasks:

“Importing metadata by using InfoSphere Metadata Asset Manager” on page 37
You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Import settings for InfoSphere Metadata Asset Manager

The Common Metadata Administrator sets global policies for imports on the **Administration** tab. These policies include rules for running express imports and for allowing duplicates.

Import settings are designed to let you establish procedures that match the needs of your enterprise. Perhaps the most important choices are whether to allow duplicates to be imported and when to require users to preview metadata during express imports.

These settings apply to all imports. There is no way to override them except by changing the settings in the **Administration** tab.

You can choose to save import passwords in encrypted format so that the importer does not have to specify the password again for reimports, testing connections, and filtering the source metadata.

If you enable additional import debugging, additional XMI information is saved. The import log, which you can download from the **Summary** section of the **Staged Imports** tab, contains a link to the XMI information.

If users delete import areas where staged imports have been shared to the repository, the history of the import is lost. But the shared metadata is not deleted from the repository.

Allowing duplicates

If you allow duplicate assets to be imported into the metadata repository, you can cause confusion and create a situation that is prone to error. For example, an InfoSphere DataStage job developer might create a table definition from a duplicate database table and use that table definition in a job. Another suite user might assign the other duplicate database table to a term, assuming it is the same database table that was referenced in the job. In addition, every time you reimport the same source, you create more duplicates.

For this reason, the default import setting is to not allow duplicate assets to be shared to the metadata repository.

However, there might be circumstances when you want to allow duplicate assets to be imported. For example, you might decide that the assets in your source that are duplicates are not critical to your work flow. You can find those assets and delete them later in the **Repository Management** tab. The alternative to importing them would be to fix them in the source, and there might be cases where that would be time-consuming. It could be more important for you to get the import completed and deal with the duplicates later. In that case, you might want to change the default setting and allow duplicate assets to be shared to the repository.

Previewing express imports

When you run an express import, the source metadata is imported to the staging area, where it is analyzed and previewed automatically. The import settings determine whether the import is shared automatically to the metadata repository or if users are required to view a preview before you manually share the import to the metadata repository.

The freedom that you give users to perform express imports should be proportional to the confidence that you have in the source metadata and in the result of sharing it to the metadata repository.

By default, users are required to preview express imports only if assets will be deleted as a result of the import. There might be cases where your importers have such thorough knowledge of the metadata in both the source and the metadata repository that you could override this default and not require a preview if assets are deleted.

By default, previews are not required if assets will be created or merged as a result of the import. Assets are frequently created in the metadata repository during import, and creating an asset does not affect the existing assets in the metadata repository. Assets are also frequently merged during reimports, often without

changing existing assets. Changing the default settings creates an import work flow that is closer to a managed import but still automates the analysis and preview steps. This ensures that the importer can preview and fully understand the results of the import before sharing it to the metadata repository.

By default, if you allow duplicates to be shared to the metadata repository, you must preview the import before sharing. Because of the potential effects of sharing duplicates, you should change this default only if you have detailed knowledge of the source metadata and only if you have full confidence in the results of sharing it.

Related tasks:

“Importing metadata by using InfoSphere Metadata Asset Manager” on page 37
You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Importing and sharing assets

You can run a managed or express import, work with the assets in the staging area, preview the result of sharing, and share the import to the metadata repository.

Preparing to import assets with InfoSphere Metadata Asset Manager

Before you import assets, review your access privileges, review the parameters that must be set, and determine the type of import that you want to use. These optional steps can help you avoid mistakes when you import assets.

Before you begin

You must have the role of Common Metadata Administrator or Common Metadata Importer.

About this task

Before you run an import from a particular source, it is important to understand the following things:

- The source metadata
- The bridge or connector that you use to import the source metadata
- What type of import is most appropriate for this metadata

These optional steps can help you avoid mistakes when you import.

Procedure

1. Identify the assets that you want to import and ensure that you have the credentials to connect to them.
2. To determine what additional information you need, start a trial run:
 - a. On the **Import** tab, click **New Import Area**. Name the import area and specify a metadata interchange server where the bridge or connector is installed that you will use to import the assets.
 - b. Choose the correct bridge or connector. You choose the bridge or connector based on the source metadata that it imports. If you choose a connector,

read the documentation for the connector. If you choose a bridge, the documentation is embedded in the import screens of the InfoSphere Metadata Asset Manager interface.

- c. Review the import parameters in advance. Page through the parameter screens. Check the help for each parameter to ensure that you are prepared with enough information to specify values for the import parameters.
3. Consider whether this import is better run as an express import or a managed import.

Express import

Depending on the settings that the Common Metadata Administrator specifies in the **Administration** tab, express imports can let you import metadata directly into the metadata repository without manual analysis or preview. An express import might be appropriate for cases where you have thorough knowledge of the assets to be imported and confidence in the results of saving them to the metadata repository.

Managed import

A managed import is appropriate in cases where you want to be sure of the contents and of their possible effect on other assets in the repository.

4. Cancel the import and obtain any missing information before you run the import. No import area is created if you cancel the import before it runs.

Importing metadata by using InfoSphere Metadata Asset Manager

You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Before you begin

You must have the role of Common Metadata Importer or Common Metadata Administrator.

About this task

You create an import area by running an import. The import area is created whether you run an express import or a managed import.

When you run a managed import, you can closely study the metadata assets in the import. You can preview the effects that the import might have on the repository. You can take time out to fix problems in the source metadata, or reimport with other parameters.

An express import saves time when you have high confidence in the contents of your import source. By default, an express import requires that you preview the result of sharing to the metadata repository if repository assets will be deleted as a result of the import. Your administrator can change the defaults to allow automatic sharing in all cases or to require previews in all cases.

Procedure

1. On the **Import** tab, click **New Import Area**.
2. In the **New Import Area** window, take the following steps:

- a. Specify a unique name and a description for the import area.
 - b. Select the metadata interchange server from which you want to run the import.
 - c. Select the bridge or connector to import with. You choose the bridge or connector depending on the source metadata that it imports. Help for the selected bridge or connector appears in the **Import Help** pane.
 - d. Click **Next**.
3. Specify import parameters for the selected bridge or connector. Help for each parameter appears when you hover over the value field.
 - a. For imports from databases and repositories, browse to select the specific assets that you want to import.
 - b. Click **Next**.
 4. If you are importing database tables or physical data models, specify identity parameters to identify the host computer and database that contain the tables. Consult the help for each selected parameter. Click **Next**.
 5. Type a description for the import event and specify whether to run an express import or a managed import.
 6. Click **Import**. The import area is created. The import runs and status messages appear.

Leave the import window open to avoid the possibility that long imports time out.

Results

When the import completes, if you ran a managed import, analyze the imported assets in the **Staged Imports** tab of the import area.

If you ran an express import, take one of the actions that are listed in the following table.

Table 10. Choices after an express import

In this case	Take this action
If the analysis shows problems that you must fix	The Staged Imports tab is displayed. Review the analysis results. If necessary, reimport the staged import.
If your administration settings require a preview	The View Share Preview screen is displayed. Preview the result of the import.
If your administration settings do not require a preview	The assets are shared to the metadata repository. The Shared Imports tab is displayed. You can browse the assets on the Repository Management tab and work with them in other suite tools.

Related tasks:

“Creating an import area from an existing staged import” on page 45
You can use the parameters of an existing staged import to reimport metadata into a new import area.

“Reimporting staged imports” on page 42

You can reimport a staged import in an import area. You can adjust the import parameters for the reimport to better specify the metadata that you import.

Related reference:

“Import settings for InfoSphere Metadata Asset Manager” on page 34
The Common Metadata Administrator sets global policies for imports on the **Administration** tab. These policies include rules for running express imports and for allowing duplicates.

“Metadata interchange servers” on page 33

Metadata interchange servers are computers that are used by InfoSphere Metadata Asset Manager to run imports with bridges and connectors. Before users can import metadata, the Common Metadata Administrator must define and enable one or more metadata interchange servers on the **Administration** tab of InfoSphere Metadata Asset Manager.

Related information:

Importing and exporting metadata by using bridges

You can export implemented data resources, logical and physical data model assets, and user information from the metadata repository. You can also import metadata by using Import Export Manager, but InfoSphere Metadata Asset Manager is the preferred import tool because of its advanced functionality.

Identity parameters for imported assets

You specify values for identity parameters when you import database assets. Specifying identity values ensures that imported tables and schemas are linked to the correct database and host system.

The bridges and connectors that import metadata from remote databases and repositories occasionally provide incomplete or inconsistent information about the host systems and databases that contain the assets that you are importing.

The identity of a database table has the following components:

- Host system name
- Database identity, consisting of database name, DBMS name, and DBMS server instance name
- Database schema name
- Database table name

Thus, host name, database name, database management system (DBMS) name, and DBMS server instance name are critical parts of the identity of the tables and schemas that they contain. If you were to import the same database assets by two different methods, and each method produced a different DBMS name, you could create duplicate assets in the metadata repository. Assets that have technically different identities but have identical content on contain the same child assets are potential duplicates.

To avoid duplicates and to ensure that your database assets are correctly identified, you specify values for identity parameters during the import process. Identity parameters are also valuable when you want the imported assets to be associated with a different host system, database, or DBMS than you imported them from. For

example, you might want the assets to be identified with the host system and database that you use for test and production. For the host system, database, and schema parameters you can either type a name or select a host, database, or schema that is already in the metadata repository to associate the imported assets with.

When you use consistent values for identity parameters, suite users can readily identify which assets to use for such tasks as creating jobs, designating stewards, or assigning to terms.

If you are importing a physical data model from a design tool, the physical data model is automatically turned into a database schema with database tables that can be used in InfoSphere DataStage and QualityStage jobs. The identity parameters that you specify are used to create a valid identity for the database schema. Both the physical data model and the database schema are stored in the metadata repository on import.

Depending on the bridge or connector that you choose for import, you can specify values for the following identity parameters:

Host system name

The name of the computer that hosts the database. If you are importing from a database, you can specify a different name than the name of the source computer. For example, you might specify the computer that will host this database during development or production.

Database name

The name of the database that contains the imported schemas and tables. You can specify a different name than the name of the source database.

DBMS name

The name of the DBMS that hosts the database. The DBMS name is part of the identity of the target database. You can specify a different DBMS than the one that hosts the source database. The DBMS name should be appropriate for the value that you enter for database name.

DBMS server instance name

The name of the DBMS server instance, if it exists. Some database management systems do not have the concept of DBMS server instance. DBMS server instance name is part of the identity of the target database. You can specify a different DBMS server instance name than the one that hosts the source database. The DBMS server instance name should be appropriate for the value that you enter for database name.

Schema name

The name of the schema that the imported tables belong to.

Related information:

Implemented data resources

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Working with staged imports

You can analyze, preview, compare, and reimport metadata before you share it to the metadata repository.

Analyzing imports in InfoSphere Metadata Asset Manager

You analyze imports to see if there are duplicate assets or assets with invalid identities.

Before you begin

You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.

About this task

All imports must be analyzed before they can be previewed or shared. If you run an express import, the import content is automatically analyzed. If you run a managed import, you must manually run analysis as described in this topic.

If an import contains assets with invalid identities, such as objects that do not have full identity strings, you cannot share the import to the metadata repository. To resolve this problem, fix the identities in the source and reimport. If fixing the identities in the source is too difficult, see the technote on auto-cleaning invalid identities: <http://www.ibm.com/support/docview.wss?uid=swg21509193> .

If an import contains duplicate assets, by default the import cannot be shared to the metadata repository. You must remove the duplicates from the source and reimport. However, the Common Metadata Administrator can override the default import settings and allow duplicates to be shared.

Procedure

1. On the **Import Areas** screen of the **Import** tab, double-click an import area to open it. The **Statistics** table displays the asset types that are contained in the import.
2. Select the **Staged Imports** tab for the import area and select the most recent staged import. You cannot analyze an import that has previously been analyzed.
3. Click **Analyze**. Columns that list the number of duplicate asset sets and assets with invalid identities are added to the **Statistics** table.
4. Click the links for any duplicate sets or invalid identities to review the problems.

What to do next

If the analysis shows no problem assets, you can preview the results of sharing the import to the metadata repository.

If invalid identities or duplicate assets prevent you from previewing or sharing the import, fix the problems in the source and reimport.

Related information:

Duplicate assets

Duplicate assets are assets that have the same identity or assets that have different identities but can be considered identical. You can manage duplicate assets by merging or deleting them.

Disconnected assets

Assets that become disconnected from their immediate parent asset in the identity string are called *disconnected assets*.

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Reimporting staged imports

You can reimport a staged import in an import area. You can adjust the import parameters for the reimport to better specify the metadata that you import.

Before you begin

You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.

About this task

After you review assets in a staged import, you might find that the import includes assets that you do not want to import and is missing others that you do want to import. You might fix these problems by reimporting from the same source while changing the import parameter settings.

You might also need to fix problems in the import source, such as duplicate assets or invalid identities, and then reimport.

You cannot choose a different bridge or connector when you reimport. You cannot change the original identity parameters. If you want to create a similar import with a different bridge or connector, or different identity parameters, create a new import area.

For imports from databases and repositories, if you have shared the staged import to the metadata repository, you cannot change the selection of assets when you reimport. For file imports, you can change the location that you import the file from.

Note: When you reimport from a file that you have previously imported and shared to the metadata repository, ensure that the file has the same basic contents as the original import. You can add, change, and delete assets in the source tool before reimporting the file, but do not use a reimport to import a completely different file of the same name, because previously imported assets could be deleted inadvertently.

Procedure

1. On the **Import** tab, double-click an import area to open it.
2. Select the **Staged Imports** tab, select a staged import, and click **Reimport**.
3. Specify import parameters. If you are importing a file from the local computer, you must select the file again. You can change the location that you import the file from.
4. Click **Next** to review identity parameters.
5. Click **Next** and specify a description for the import.
6. Specify whether to run an express import or a managed import.
7. Click **Reimport**. To avoid timeouts in long imports, keep the window open until the import completes.

Results

The import runs and a new staged import is created in the import area.

What to do next

You can work elsewhere in the product while the import runs.

When the import completes you can see the results:

- If you specified a managed import you are returned to the staging area where can analyze the import.
- If you specified an express import, a preview is displayed, or the results of sharing the import are displayed, depending on your administration settings for express imports.

Related concepts:

“Command-line interface for InfoSphere Metadata Asset Manager” on page 48
Use the InfoSphere Metadata Asset Manager command-line interface (CLI) to perform operations from a Windows batch (.bat) script. You can use the command-line interface to import metadata, reimport metadata, delete staged imports from import areas, and generate a list of import areas.

“Overview of InfoSphere Metadata Asset Manager” on page 29

You use InfoSphere Metadata Asset Manager to import and manage common metadata assets, which are vital to all components of the InfoSphere Information Server suite. You can import assets to a staging area before you share them to the metadata repository. In the metadata repository, you can browse and search for common metadata assets, set implementation relationships between them, and merge duplicates.

Related tasks:

“Importing metadata by using InfoSphere Metadata Asset Manager” on page 37
You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Comparing staged imports

You can compare staged imports that are in the same import area. Comparing shows you the differences in assets between each staged import.

Before you begin

You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.

About this task

When you compare staged imports, you can determine whether a reimport has achieved its goal of collecting the assets that you want to share to the repository. You can also get a detailed view of the changes to particular assets, including the different property values of assets that are otherwise identical.

Assets of a single asset type share the same properties. For example, the properties of database tables include long description and short description. Two database tables that have the same identity but are in different staged imports can have different values for their short descriptions or long descriptions. You can review such value differences when you compare staged imports.

You can compare any staged import to any previous staged import in the same import area.

Procedure

1. On the **Import** tab, double-click an import area to open it.
2. Select the **Staged Imports** tab.
3. Select a staged import, and click **More Actions > Compare with Previous**. The **Compare with Previous Staged Import** displays a comparison of the selected stage import with the staged import that immediately preceded it.
4. If appropriate, select a different previous staged import from the list for comparison.
5. In the **Comparison Results** table, click the links to review the matched, added, and deleted assets. *Matched assets* are assets that have the same identity but might have different values for some properties.

To do this comparison	Take this action
Compare the values of the properties of assets that were matched	<ol style="list-style-type: none">1. Click a link in the Matched column.2. On the Matched Asset Property Differences screen, select an asset to analyze it, or click Analyze All. The differences icon () next to an asset name indicates that there are differences between values for corresponding properties.3. Select an asset where differences exist and, select View All Properties to compare the values for each property.
View lists of assets that were added or deleted between the previous staged import and the selected staged import	Click a link in the Added or Deleted columns.

What to do next

Depending on the results of the comparison, you might want to do any of the following actions:

- Compare the staged import to another previous staged import.
- Edit the source metadata in the source tool, file, or database.
- Reimport the staged import.
- Preview the results of sharing the staged import to the metadata repository.
- Share the staged import.

Related information:

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Creating an import area from an existing staged import

You can use the parameters of an existing staged import to reimport metadata into a new import area.

Before you begin

You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.

About this task

When you are working in a particular import area, you might want to start a new import area that is based on a particular staged import. For example, you might want to reimport the source metadata and change the identity parameters, which you cannot do with a reimport to the same import area. Or you might want to import a different set of metadata by using the same bridge or connector and most of the same parameter values.

By using the **Import into New Import Area** option for a staged import, you can start a new import to a new import area. The import screens are populated with the parameter values from the staged import. You can change all import parameter values, including the bridge or connector, the metadata interchange server, and the identity parameters.

Procedure

1. On the **Import** tab, double-click an import area to open it.
2. Select the **Staged Imports** tab, select a staged import, and click **More Actions > Import into New Import Area**.
3. Specify parameters on the import screens and click **Import**.

Results

A new import area is created. You can view the assets on the **Staged Imports** tab. If you ran an express import, you can preview the results of sharing to the metadata repository, or view the results, depending on your administration policy.

Related tasks:

“Importing metadata by using InfoSphere Metadata Asset Manager” on page 37
You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Previewing the result of an import

Before you share an import to the metadata repository, you can see what effect the import will have on assets that are already in the repository.

Before you begin

You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.

You must have analyzed the import manually or you must have run an express import that performed analysis.

About this task

You preview before sharing to understand the effect the import will have on existing assets in the repository. The preview tells you which existing assets will be deleted or merged when the import is shared.

For managed imports, previews are required before you can share the import. For express imports, previews are required by default if sharing the import will delete assets. The Common Metadata Administrator can change the default setting to allow sharing without a preview or to require previews for all express imports.

You can preview and share only the most recent staged import in an import area.

Procedure

1. On the **Import** tab of InfoSphere Metadata Asset Manager, double-click an import area to open it.
2. Select the **Staged Imports** tab, select the most recent staged import and click **Preview**. The preview results are displayed on the **View Share Preview** screen:
 - The **Statistics** table lists each type of asset that will be created, merged, or deleted by the import.
 - The **Resulting Assets** tree displays the imported assets as they would appear if you shared them to the repository.

The actual results of sharing the import might differ from the preview of those results, depending on transactions within the metadata repository between the time of the preview and the share. You can refresh the preview to take into account changes to the metadata repository since the preview was created.

3. If the preview was created previously and you want to see an updated view, click **Refresh**. Always refresh old previews before you share the import to the repository.
4. Examine the assets in the **Statistics** table:
 - a. Click a link for an asset type to view a list of assets of that type that will be created, merged, or deleted.

- b. Select an asset in the list. The **Resulting Assets** tree displays the asset as it will appear in the metadata repository when you import it. If the asset will be deleted, the tree opens to the nearest merged parent asset.
5. Browse the **Resulting Assets** tree to see assets that will be created or merged. Hover over the icon for an asset to see what type it is and whether it will be merged or created.
6. Right-click an asset in the **Statistics** table or the **Resulting Assets** tree to take any of the following actions:
 - Search for assets with similar names in the metadata repository.
 - View the asset in the Repository Management tab, if it exists in the metadata repository.
 - View the asset in InfoSphere Metadata Workbench if it is installed.

What to do next

When you finish previewing, you can click **Share to Repository** on the **View Share Preview** screen. Sharing the import to the metadata repository makes the assets in the import available to other suite tools.

If you are not ready to share the import, you can continue working on the import contents. You might want to take one or more of the following actions:

- Edit the assets in the source tool.
- Reimport with different parameter values.
- Create a new import based on the parameter values of the staged import.
- Delete existing assets from the metadata repository.

Sharing imports to the metadata repository

You can share staged imports to the metadata repository where suite users can access the imported assets.

Before you begin

You must meet the following prerequisites:

- You must have the Common Metadata Administrator role, or you must be the Common Metadata Importer who created the import area.
- You must have analyzed and previewed the import. Analysis is automatic if you run an express import.

About this task

Sharing the import to the metadata repository is the final step in importing when you use InfoSphere Metadata Asset Manager. Users of other suite tools can work with the assets that you share to the repository. For example, users of InfoSphere DataStage and QualityStage can create table definitions from shared database tables and use them in jobs. Users of InfoSphere Business Glossary can assign the imported assets to terms and designate stewards for the assets.

You can share only the most recent staged import in an import area.

Procedure

1. On the **Import** tab of InfoSphere Metadata Asset Manager, double-click an import area to open it.

2. Select the **Staged Imports** tab and select the most recent staged import.
3. If you have not recently previewed the results of sharing the staged import, run the preview again and then click **Refresh**. Always refresh old previews before you share the import to the repository. It is possible that objects in the repository might have been changed by imports or deletions since the original preview was run.
4. Click **Share to Repository**. The results are displayed on the **Shared Imports** tab:
 - The **Statistics** table lists each type of asset that is created, merged, or deleted by the import. You can filter the results to see the specific assets that are created, merged, or deleted.
 - The **Resulting Assets** tree displays the imported assets. You can expand the tree and hover over the icon for an asset to display the asset type and whether it was created or merged.

The actual results of the share might differ from the preview, depending on transactions within the repository between the time of the preview and the share.

What to do next

You can browse and manage the imported assets on the **Repository Management** tab or work with them in other suite tools.

If the contents of the repository change and you want to share the staged import again, select the shared import on the **Shared Imports** tab, and click **Reshare to Repository**. It is best practice to preview the results again before you reshare. You can reshare only the most recent staged import in an import area.

Command-line interface for InfoSphere Metadata Asset Manager

Use the InfoSphere Metadata Asset Manager command-line interface (CLI) to perform operations from a Windows batch (.bat) script. You can use the command-line interface to import metadata, reimport metadata, delete staged imports from import areas, and generate a list of import areas.

The command-line interface is supported only on Windows computers.

If you have the Common Metadata Administrator role, you can generate a list of import areas, import metadata, delete staged imports from import areas, and reimport any staged import. If you have the Common Metadata Importer role, you can list only those import areas that you created, delete staged imports only from import areas that you created, and reimport staged imports only in import areas that you created.

Credentials

Command-line imports might require separate sets of credentials for connecting to InfoSphere Information Server and for connecting to the source of the metadata that you are importing:

- The user name and password are required for connecting to InfoSphere Information Server. If you do not enter the user name and password as command parameters, you are prompted to enter them.
- When you are re-importing, if the original import used a bridge, and the password was not saved, you will not be able to invoke the reimport by using

the **reimport** action. You can enable saving of passwords on the Administration tab of InfoSphere Metadata Asset Manager.

Using imam command line scripts

Use the `imam.bat` script file to run the command line on Windows systems.

You can run the script from the `InformationServer\ASBNode\bin` folder for client or engine tier installations, and in the `InformationServer\ASBServer\bin` folder for services tier installations. For example, if the InfoSphere Information Server home directory is `C:\IBM\InformationServer`, the **imam** command in a services tier installation would be `C:\IBM\InformationServer\ASBServer\bin\imam.bat`.

HTTP and HTTPS configuration

The command line interface uses HTTP or HTTPS to communicate with the services tier computer, depending on the configuration of the services tier computer. To determine whether the services tier computer is configured to use HTTPS, look for the file `https.conf` in one of the following locations:

- `install\dir\ASBServer\conf`
- `install\dir\ASBNode\conf`

If the services tier computer is configured to use HTTP, but you want to run the command line by using HTTPS, you must manually change the setting, and configure HTTPS for the entire suite. To manually configure HTTPS, see the topic *Manually configuring InfoSphere Information Server components to use HTTPS* in the *IBM InfoSphere Information Server Planning, Installation, and Configuration Guide*.

Command syntax

```
imam --help --action action
[action options] [--username username] [--password password] [--server server_name]
[--port port_number] [--authfile authfile_path]
```

Command parameters

You must enter two dashes (--) before all command parameters, when you use the long name of the parameter. For example, `--action`. When you use short names for the parameters, specify only one dash (-). For example, `-a`.

--username *user_name*

Optional. The user name that is required for logging into InfoSphere Information Server. The user must have the role of Common Metadata Administrator or Common Metadata Importer. You can use the **--authfile** parameter instead of specifying the **--username** and **--password** parameters. You can enter a username without entering a password. If you do this you are prompted for a password. The short name of this parameter is **-u**.

--password *password*

Optional. The password for the specified user name to log into InfoSphere Information Server. You can use the **--authfile** parameter instead of specifying the **--username** and **--password** parameters. The short name of this parameter is **-w**.

--server *server_name*

Optional. Name of the services tier computer. This parameter is mandatory if you are using the command-line interface on the client tier computer on Windows where the metadata interchange agent is installed. It is optional if you are using the command-line interface on the services or engine tiers on

Windows. If you specify a server, then you must specify a port. If you do not specify `--server` or `--port`, then you can specify the primary server that you want to use in the `registered-servers.xml` file. Specify `is-primary-server="true"` for the primary server. The short name of this parameter is `-s`.

--port *port_number*

Optional. The port number to use on the services tier computer. If your InfoSphere Information Server suite installation is configured for HTTP, the default port used is 9080. If you do not specify a port, and your InfoSphere Information Server suite installation is configured for HTTPS, the default port used is 9443. If you specify a port, you must specify a server. The short name of this parameter is `-p`.

--authfile *authfile_path*

Optional. Path to a file that contains the encrypted or non-encrypted credentials for logging on to InfoSphere Information Server. If you use the `--authfile` parameter, you do not need to specify `--username` or `--password` on the command line. If you specify both the `--authfile` option and the explicit user name and password options, the explicit options take precedence over what is specified in the file. For more information see the topic *Encrypt command* in the *IBM InfoSphere Information Server Administration Guide*.

--help

Optional. Prints the list of actions and parameters. The help command is automatically issued when you issue a command that contains a syntax error, such as a typographical error, an improperly cased option or argument, or when the command is missing a mandatory parameter. The short name of this parameter is `-h`.

--action *action*

Required. Specifies the type of action that you want to run by using the command line. The short name of this parameter is `-a`. You can specify the following options for each action:

Table 11. Options for the `--action` parameter

Long name	Parameters	Description
list	<code>--type type</code>	<p>--type <i>type</i></p> <p>Provides the list of the specified type. The only type that is currently supported is <i>area</i>. This is a mandatory parameter for the list action. You must specify <i>area</i> to produce a list of import areas. If you have the Common Metadata Administrator role, the list includes import areas created by all users. If you have the Common Metadata Importer role, the list only includes import areas that you created. The short name for this parameter is <code>-t</code>.</p>

Table 11. Options for the --action parameter (continued)

Long name	Parameters	Description
import	<p>--importArea <i>import_area_name</i></p> <p>--paramsFile <i>parameter_file_name</i></p> <p>--mis_name <i>metadata_interchange_server_name</i></p> <p>--localFile <i>local_file_name</i></p> <p>--area_description <i>description_of_import_area</i></p> <p>--import_description <i>description_of_import</i></p>	<p>Creates a new import area and then completes an express import of metadata by using the import bridge that is specified in the parameters file.</p> <p>Note: To use the import action, you must install prerequisite software that is listed in this technote: http://www-01.ibm.com/support/docview.wss?uid=swg27038192.</p> <p>Depending on your import settings in the Administration tab, the import might be directly shared to the metadata repository, or you might be required to preview the import first. If there are errors, the imported metadata appears as a new staged import in the staging area, but is not shared with the metadata repository.</p> <p>For information about creating a parameters file, see “Creating an import area by using the command line” on page 55. You must specify the following parameters on the command line:</p> <p>--importArea <i>import_area_name</i> Name of the import area that you want to create. The parameters for the import area are the parameters that are specified in the parameters file. If the import area names contain blank spaces, you must put quotation marks around the blank spaces. If the name of the import area contains quotation marks, then you must enter a backslash character (\) before each of the quotation marks. For information about escaping characters, see “Escaping characters” on page 57. The short name for this parameter is -i.</p> <p>--paramsFile <i>parameter_file_name</i> The name and location of the parameter file. For example, C:\Files\parameters_file.xml. The short name for this parameter is -pf.</p> <p>--mis_name <i>metadata_interchange_server_name</i> The name of the metadata interchange server that you want to use for the import. It is required only if you want to import a file that is on the metadata interchange server. The short name for this parameter is -mn.</p> <p>The following parameters are optional:</p> <p>--localFile <i>local_file_name</i> The path to a local file to import. It is required only if:</p> <ul style="list-style-type: none"> You are importing metadata by using a CWM XMI bridge. You want to import a file that is on the local computer instead of the metadata interchange server. <p>The --localFile option is case sensitive, so you must use a capital "F" when you specify this option.</p> <p>The short name for this parameter is -l.</p> <p>--area_description <i>description_of_import_area</i> A description of the import area. The description is displayed on the Import tab in InfoSphere Metadata Asset Manager. The short name for this parameter is -ad.</p> <p>--import_description <i>description_of_import</i> A description of the import. The description is displayed on the Staged Imports tab in InfoSphere Metadata Asset Manager. The short name for this parameter is -id.</p>

Table 11. Options for the --action parameter (continued)

Long name	Parameters	Description
reimport	<p>--importArea <i>import_area_name</i></p> <p>--localFile <i>local_file_name</i></p>	<p>Reimports the most recent staged import as an express import. Depending on your import settings in the Administration tab, the import might be directly shared to the metadata repository, or you might be required to preview the import first. If there are errors, the reimported metadata appears as a new staged import in the staging area, but is not shared with the metadata repository. You must specify the following parameter:</p> <p>--importArea <i>import_area_name</i> Name of the import area in which the reimport runs. The reimport uses the parameters from the most recent import in the import area. If the import area names contain blank spaces, you must put quotation marks around the blank spaces. If the name of the import area contains quotation marks, then you must enter a backslash character (\) before each of the quotation marks. For information about escaping characters, see "Escaping characters" on page 57. The short name for this parameter is -i.</p> <p>The following parameter is optional:</p> <p>--localFile <i>local_file_name</i> The path to a local file to import. It is required only when you are reimporting a file that was initially imported on the local computer instead of the metadata interchange server. The --localFile option is case sensitive, so you must use a capital "F" when you specify this option. The short name for this parameter is -l.</p> <p>Note: If password parameters are not saved as part of the previous import, then you will not be able to invoke the reimport by using the reimport parameter. You can save the password parameters on the Administration tab in InfoSphere Metadata Asset Manager.</p>
purge	<p>--area_name <i>import_area_name</i></p> <p>--keep_events <i>number_to_keep</i></p>	<p>Deletes staged imports from an import area. In all cases, at least one staged import remains in the import area after the rest are deleted. If there are shared imports in the area, the most recent shared import is always kept. You must specify the following parameter:</p> <p>--area_name <i>import_area_name</i> Name of the import area from which you are deleting staged imports. If the name of the import area contains blank spaces, you must put quotation marks around the blank spaces. If the name of the import area contains quotation marks, you must enter a backslash character (\) before each of the quotation marks. For information about escaping characters, see "Escaping characters" on page 57. The short name for this parameter is -i.</p> <p>The following parameter is optional:</p> <p>--keep_events <i>number_to_keep</i> The number of staged imports that you want to keep in the import area. Staged imports are kept in the following order: the most recent shared import, the most recent staged import (shared or unshared), the next most recent staged import, and so on.</p> <p>If you use purge without using --keep_events, all but three of the staged imports are deleted from the import area. If there are three or fewer staged imports in the import area when you run the command, no action is taken. If you use --keep_events, you must specify a number greater than zero. If the number of staged imports in the import area is equal to or less than the number specified, no action is taken. The short name for this parameter is -ke.</p>

List command example

The following list command generates a list of all import areas.

```
imam --username isadmin --password isadmin --action list --type area
```

Import command example 1

The following import command creates the import area *area57* and uses the parameters file *ErwinData1_parameters.xml* to import metadata into InfoSphere Metadata Asset Manager.

```
imam -i area57 --paramsFile "C:\Files\ErwinData1_parameters.xml" --action import
--area_description Erwin files --import_description initial import
--mis_name Default --username isadmin --password mypassword
```

Import command example 2

The following import command creates the import area *test20* and uses the parameters file *XML.xml* to import metadata into InfoSphere Metadata Asset Manager. The *ASCL1.xmi* file is imported from the local computer, rather than from the metadata interchange server.

```
imam -i test20 --paramsFile C:\Files\XML.xml -l C:\Files\ASCL1.xmi --action import
--area_description someDesc --import_description stagedDesc --mis_name Default
--username isadmin --password mypassword
```

Import command example 3

The following import command creates the import area *"Department"* and uses the parameters file *MyData.xml* to import metadata into InfoSphere Metadata Asset Manager. The name of the import area contains quotation marks, so backslash characters (\) are included before each of the quotation marks.

```
imam -i \"Department\" --paramsFile "C:\Files\MyData.xml" --action import
--area_description Department names --import_description initial import
--mis_name Default --username isadmin
```

Reimport command example 1

The following reimport command reimports the most recent import event in the import area *area51* into InfoSphere Metadata Asset Manager.

```
imam --action reimport --importArea area51 --server 9.184.184.153 --port 9080
--localFile C:\data.xml --authfile C:\authfile.txt
```

Reimport command example 2

The following reimport command reimports the most recent import event in the import area *"Customer"* into InfoSphere Metadata Asset Manager. The short name commands for all parameters are used. The name of the import area contains quotation marks, so backslash characters (\) are included before each of the quotation marks.

```
imam.bat -u isadmin -w password100 -s mds-qa1 -p 9443 -a reimport
-i \"Customer\"
```

Reimport command example 3

The following reimport command reimports the most recent import event in the import area *'salary'* into InfoSphere Metadata Asset Manager. The short name commands for all parameters are used, and the server and port parameters are

omitted because the command is being run on the local InfoSphere Metadata Asset Manager Server machine on the services tier.

```
imam.bat -a reimport -i 'salary' -u admin1 -w password1
```

Reimport command example 4

The following reimport command reimports the most recent import event in the import area *"Department heads"* into InfoSphere Metadata Asset Manager. The short name commands for all of the parameters are used. The import area name *"Department heads"* has a space in it, so the whole name needs to be surrounded by quotation marks, and there is a quotation mark in the name that needs to be escaped with a backslash character (\).

```
imam.bat -a reimport -i "\"Department heads" -u admin1  
-w password1 -s mds-qa1 -p 9443
```

Reimport command example 5

The following reimport command reimports the most recent import event in the import area *"test" model* into InfoSphere Metadata Asset Manager.

```
imam.bat -a reimport -i "Import of \"test\" model" -u admin1  
-w password -s mds-qa1 -p 9443
```

Purge command example 1

The import area *MyImports* contains the following staged imports, listed in the order in which they were imported.

- Import 1 (not shared), the oldest import
- Import 2 (shared)
- Import 3 (not shared)
- Import 4 (shared)
- Import 5 (not shared)
- Import 6 (not shared)
- Import 7 (not shared), the most recent import

The following command does not specify **-keep_events**. By default, it keeps three staged imports, always including the most recent shared import.

```
imam --action purge --area_name MyImports
```

In this example, imports 7, 6, and 4 are kept. Import 4 is kept because it is the most recent shared import. Imports 6 and 7 are kept because they are the most recent remaining staged imports.

Purge command example 2

The following command deletes all but two staged imports from the import area *MyImports*.

```
imam --action purge --area_name MyImports --keep_events 2
```

In this example, imports 7 and 4 are kept, because import 4 is the most recent staged import and import 7 is the most recent import.

Purge command example 3

The following command deletes all but six staged imports from the import area *MyImports*.

```
imam --action purge --area_name MyImports --keep_events 6
```

Only import 1 is deleted.

Purge command example 4

The following command attempts to delete all seven staged imports.

```
imam --action purge --area_name MyImports --keep_events 0
```

The command fails because you must enter a number greater than zero. You cannot delete all staged imports by using the command line.

Purge command example 5

For the following command, the import area *MyImports2* has seven staged imports, none of which has been shared.

```
imam --action purge --area_name MyImports2 --keep_events 2
```

The command deletes all but two staged imports from the import area, leaving the two most recent unshared staged imports.

Help command example

The following help command generates a list of commands.

```
imam --help
```

Related tasks:

“Reimporting staged imports” on page 42

You can reimport a staged import in an import area. You can adjust the import parameters for the reimport to better specify the metadata that you import.

Creating an import area by using the command line

You can prepare a parameters file to run a new import on the command line of InfoSphere Metadata Asset Manager.

Before you begin

Before you can save a parameters file for use on the command line, you must create or have access to an import area in InfoSphere Metadata Asset Manager. To save the parameters file you must have the role of Common Metadata Administrator or you must be the common metadata importer who created the import area.

Before you can run a command-line import that uses the parameters file, you must meet the following prerequisites:

- Install prerequisite software that is listed in this technote: <http://www-01.ibm.com/support/docview.wss?uid=swg27038192>.
- You must have the role of Common Metadata Administrator or Common Metadata Importer.

About this task

When you run an import in InfoSphere Metadata Asset Manager the parameter values that you specify in the import wizard are saved internally.

If you want to create another import area based on the original staged import, you can create a parameters file that contains the saved parameter values of the staged import. You can specify the parameters file on the command line and import the metadata into a new import area by using the command line. The parameter information is specific to the bridge that you used for the original import.

You can edit the parameters file to change or add values.

To create a parameters file without actually importing the metadata into the metadata repository, run the import as a managed import and do not share the resulting staged import.

For security reasons, password values that you entered in the import wizard are not saved. You must enter them manually in the import parameters file. You can encrypt any value by using the encryption tool. See the topic [Encrypt command](http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/topic/com.ibm.swg.im.iis.found.admin.common.doc/topics/encrypt_ovw.html) (http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/topic/com.ibm.swg.im.iis.found.admin.common.doc/topics/encrypt_ovw.html).

Procedure

1. Open an import area and select the **Staged Imports** tab.
2. Select the staged import whose parameters you want to use and click **More Actions > Save Parameters File**.
3. Download the parameters file and open it in a text editor.
4. Password values are not automatically displayed in the generated file. Add any necessary password values between the `<value>` and `</value>` tags for password parameters.
5. Optional: As necessary, change any other parameter values that are within the value tags.

Note: The parameter values within the value tags are not always the same as the drop-down choices that are displayed in the user interface. If you are not sure of the value to specify in the file, run the import again as a managed import, choose the desired value in the interface, and save the parameters file.

6. If you want to import a file on your local computer, take one of the following actions:

Option	Description
If the original import specified a local file	Specify the local file path as a command-line option. Do not specify the path in the parameters file.
If the original import specified a file on the metadata interchange server	<ol style="list-style-type: none">1. Delete the value for the File parameter in the parameters file, by deleting the content between the <code><value></code> and <code></value></code> tags.2. Specify the local file path as a command line option. Do not specify the path in the parameters file.

7. When you are ready to run a command-line import, specify the parameters file on the command line and use the **import** option with the **action** parameter of the **imam** command. For information about command-line syntax, see *Command-line options for InfoSphere Metadata Asset manager*.

Results

When the import runs, a new import area is created. The imported metadata is displayed in the import area.

Command-line imports are run as express imports. Depending on your import settings in the Administration tab, the import might be directly shared to the metadata repository or you might be required to preview the import first.

Escaping characters

In cases where asset names contain characters or blank spaces that conflict with the InfoSphere Metadata Asset Manager command-line interface syntax, you must escape those characters in the command line.

The InfoSphere Metadata Asset Manager command-line interface requires that you escape certain characters.

To escape the following characters, you must insert double quotation marks (") around the asset names that contain the characters that must be escaped:

- ^ (hat, circumflex)
- & (ampersand)
- < (less than)
- > (greater than)
- | (logical or)
- Blank spaces

For example, an ampersand character (&) that is part of an asset name is not accurately read by the command-line unless it is surrounded by double quotation marks. Therefore, the asset name A&C must be written as "A&C" on the command line.

When an asset name contains double quotation marks, use a backslash before the double quotation marks (\) to escape them. For example, the asset name Customer"Fixes" must be written as Customer\"Fixes\".

Managing common metadata assets

You can use the **Repository Management** tab in InfoSphere Metadata Asset Manager to view, compare, merge, and delete assets that are stored in the metadata repository. You can also set implementation relationships between logical and physical data models and implemented data resources.

Finding metadata assets in the repository

You can find assets in the metadata repository by browsing the repository or by searching for asset names that match a specific character string.

Browsing the metadata repository to find assets

You can find assets in the metadata repository by browsing implemented data resources, business intelligence (BI) assets, physical data model assets, or logical data model assets.

Before you begin

To browse the metadata repository you must have the Common Metadata User role, the Common Metadata Importer role, or the Common Metadata Administrator role.

Procedure

1. Click the **Repository Management** tab in InfoSphere Metadata Asset Manager.
2. In the Navigation pane, expand **Browse Assets**, and then click **Logical Data Models**, **Physical Data Models**, **Implemented Data Resources**, **Data Item Definitions**, **Business Intelligence Models**, or **Business Intelligence Reports**.

Related tasks:

“Viewing asset details” on page 59

You can view the details of an asset in the metadata repository. Asset details include the properties of the asset, such as its name, type, and description, the list of assets that it contains, and other related assets such as stewards, creation tools, and other assets that implement the selected asset. You might want to view asset details before deleting an asset from the metadata repository or before merging it with another asset.

Related information:

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Searching the repository to find assets

You can find implemented data resources, business intelligence (BI) assets, data item definitions, logical data model assets, and physical data model assets in the metadata repository by searching for assets whose names contain a specific character string. Searches do not return disconnected assets.

Before you begin

To search the metadata repository you must have the Common Metadata User role, the Common Metadata Importer role, or the Common Metadata Administrator role.

About this task

The search is case-sensitive. Two wildcards are supported when you enter a search string in the **Search** section of the Navigation menu. An asterisk (*) finds one or more characters in the specified position. An exclamation point (!) finds any single character at the current position. If you enter a backslash (\) immediately before a wildcard character, the wildcard is treated as a literal value.

Procedure

1. Click the **Repository Management** tab in InfoSphere Metadata Asset Manager .
2. Expand the **Search** section in the Navigation pane.

3. Type a character string in the **Asset Name** box, and click **Search**.
4. Optional: To filter the search results by a specific asset type, select an asset type from the **Asset Type** list.
5. Click **Search**.
6. Optional: To narrow the scope of the search results, click **Search Options** and enter values in the following fields on the Search Options pane:

Option	Description
Asset type	Select the type of asset you want to search for in the Asset type menu.
Asset name	Enter the name of an asset in the asset name field. The search is case-sensitive and you must enter an exact value.
Where	Select a Where option to modify search filters in all of the fields displayed in the search options area. Exact Match (Not Case Sensitive) makes the search not case-sensitive and only returns exact matches. Exact Match only returns exact matches. Contains returns search results for any string that you specify. Prefix returns matches that occur in the beginning of a search string. Suffix returns matches that occur at the end of a search string.
Host	Enter a host name to limit search results to a specific computer. The search results include only those assets that the specified host contains.
Database	Enter a database name to limit search results to a specific database. The search results include only those assets that the specified database contains.

Related tasks:

“Viewing asset details”

You can view the details of an asset in the metadata repository. Asset details include the properties of the asset, such as its name, type, and description, the list of assets that it contains, and other related assets such as stewards, creation tools, and other assets that implement the selected asset. You might want to view asset details before deleting an asset from the metadata repository or before merging it with another asset.

Related information:

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Viewing asset details

You can view the details of an asset in the metadata repository. Asset details include the properties of the asset, such as its name, type, and description, the list of assets that it contains, and other related assets such as stewards, creation tools,

and other assets that implement the selected asset. You might want to view asset details before deleting an asset from the metadata repository or before merging it with another asset.

Before you begin

Find the asset by browsing or searching the metadata repository.

To view asset details you must have the Common Metadata User role, the Common Metadata Importer role, or the Common Metadata Administrator role.

Procedure

1. In the list of assets, select an asset. The details that are displayed vary depending on the type of asset that is selected.
2. Optional: To retrieve the latest usage information for all assets, click **More Actions > Retrieve Usage for All Assets**. You can view general usage information for top-level assets, such as databases or schemes, and detailed usage information for children assets, such as tables and stored procedures.
3. Optional: To view the selected asset in InfoSphere Metadata Workbench, click **More Actions > Show Asset In Metadata Workbench**. By viewing an asset in InfoSphere Metadata Asset Manager, you can see more information about how the asset is used in other InfoSphere Information Server products. In addition, you can run queries on the asset and other related assets.
4. Optional: To hide the details of a selected asset, click **More Actions > Hide Details**.

Related tasks:

“Searching the repository to find assets” on page 58

You can find implemented data resources, business intelligence (BI) assets, data item definitions, logical data model assets, and physical data model assets in the metadata repository by searching for assets whose names contain a specific character string. Searches do not return disconnected assets.

“Browsing the metadata repository to find assets” on page 58

You can find assets in the metadata repository by browsing implemented data resources, business intelligence (BI) assets, physical data model assets, or logical data model assets.

“Managing disconnected assets” on page 62

You can view the details of a disconnected implemented data resource or business intelligence (BI) asset or delete disconnected assets from the metadata repository. Disconnected assets do not have a complete identity because one or more assets in the identity hierarchy do not exist.

Related information:

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Deleting metadata assets

You can delete assets from the metadata repository of InfoSphere Information Server. When you delete an asset, all assets that are contained by that asset are also deleted. You might want to delete assets if the metadata repository contains duplicate assets, disconnected assets, or assets that are no longer required.

Before you begin

You must have the Common Metadata Administrator role to delete assets from the metadata repository.

Find the asset by browsing or searching for assets in the metadata repository. You can also find the asset by comparing duplicates or identifying disconnected assets.

Before deleting hosts that contain other assets, you must delete all of the contained assets. For example, if you want to delete a host that contains schemas, you must delete all of the schemas before you can delete the host.

Procedure

1. To make sure that you want to delete the asset, complete these steps:
 - a. View the details of the asset.
 - b. View the usage information to see the tools that use the asset.
2. Select one or more assets that you want to delete, and then click **Delete**. In the Delete Confirmation window that is displayed, click **Yes**.

Results

When you delete an asset, it is removed from the metadata repository.

Related information:

Delete rules for assets

Before deleting an asset from the metadata repository, make sure that you understand these delete rules for the asset.

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Managing disconnected assets

You can view the details of a disconnected implemented data resource or business intelligence (BI) asset or delete disconnected assets from the metadata repository. Disconnected assets do not have a complete identity because one or more assets in the identity hierarchy do not exist.

Before you begin

You must have the Common Metadata Administrator role to delete assets from the metadata repository. Before deleting an asset, make sure that you understand the delete rules for assets.

About this task

If an asset is used in by InfoSphere Information Server tools, review the usage details before deleting the asset. Disconnected assets that are being analyzed by other tools should probably be deleted, because they might skew analysis results. You might want to reimport the original asset for use with the same suite tools before you delete the disconnected asset.

Some disconnected assets cannot be identified precisely by asset type when their containing assets are missing. For this reason, database tables, design tables, and data file structures are identified as *data collections* in the list of disconnected assets.

Procedure

1. Click the **Repository Management** tab in InfoSphere Metadata Asset Manager.
2. Expand the **Manage Disconnected Assets** section of the Navigation menu and select the type of asset that you want to view disconnected assets for. A list of disconnected assets is displayed on the Search Results page.
3. Optional: View the details of the asset to verify that you want to delete it.
4. Delete the disconnected asset. You can delete the asset in the following ways:
 - After finding a disconnected asset, in the list of disconnected assets, select assets that you want to delete, and click **Delete**.
 - While viewing the details of the disconnected asset, click **Delete** to delete the asset that you are viewing.

Related concepts:

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Related tasks:

“Viewing asset details” on page 59

You can view the details of an asset in the metadata repository. Asset details include the properties of the asset, such as its name, type, and description, the list of assets that it contains, and other related assets such as stewards, creation tools, and other assets that implement the selected asset. You might want to view asset details before deleting an asset from the metadata repository or before merging it with another asset.

“Deleting disconnected link points” on page 69

You can delete disconnected link points from the metadata repository. Deleting disconnected link points improves performance and frees disk space on your system.

Related information:

Delete rules for assets

Before deleting an asset from the metadata repository, make sure that you understand these delete rules for the asset.

Disconnected assets

Assets that become disconnected from their immediate parent asset in the identity string are called *disconnected assets*.

Managing duplicate metadata assets

You can compare, merge, or delete duplicate assets in the metadata repository. Deleting duplicate assets helps ensure that suite tool users can choose the correct assets to use.

Before you begin

You must have the Common Metadata Administrator role to delete or merge assets. Before deleting an asset, make sure that you understand the delete rules for assets.

About this task

You can manage duplicates for all common metadata assets except BI assets. Assets are considered duplicate candidates based on their names and their hierarchical identities. For example, if two databases are named, "Customer" they would be considered potential duplicates based on the fact that they have the same name. You can determine whether potential duplicates are true duplicates on the side-by-side comparison screen where the properties of each potential duplicate asset are displayed.

Procedure

1. Click the **Repository Management** tab in InfoSphere Metadata Asset Manager.
2. Find duplicate assets in one of the following ways:

Option	Description
Use the manage duplicates workspace to compare potential duplicates that have the same name.	<ol style="list-style-type: none"> 1. In the Manage Duplicates Navigation pane click Logical Data Models, Physical Data Models, or Implemented Data Resources. 2. In the right pane click View Duplicates for a particular asset type. 3. Select up to five assets of the same name and click Compare.
Search or Browse the repository to find an asset and compare it to other assets.	<ol style="list-style-type: none"> 1. Select an asset and click Compare. 2. Select up to four assets to compare to the original asset: <ul style="list-style-type: none"> • Click Duplicate Candidates to compare assets that have the same name. • Browse or search to compare assets whose names are different from the originally selected asset. 3. Click Add to Selected and click OK.

3. Carefully compare the properties of the assets, the relationships the assets have to other assets, and the usage information to make sure that the assets are duplicates.
4. When you are satisfied that the asset is a duplicate, you can delete it or merge it with another asset.

Option	Description
To delete the duplicate asset	Select the asset that you want to delete and click Delete . You can select up to five assets to delete.
To merge the duplicate asset with another asset	Select the assets that you want to merge and click Merge . You must select two assets to merge. You can merge only databases that are contained by the same host and database schemas that are contained by the same database.

Click **Close** to close the workspace.

Related tasks:

“Merging assets” on page 67

After browsing for assets or managing duplicate assets, you can merge two assets into one asset. Merging duplicate metadata assets makes it easier for suite tool users to choose the correct metadata to use in jobs. You can merge databases or database schemas.

Related information:**Merging assets**

After browsing for assets or managing duplicate assets, you can merge two assets into one asset. Merging duplicate metadata assets makes it easier for suite tool users to choose the correct metadata to use in jobs. You can merge databases or database schemas.

Duplicate assets

Duplicate assets are assets that have the same identity or assets that have different identities but can be considered identical. You can manage duplicate assets by merging or deleting them.

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Asset identity

Each asset in the metadata repository needs to have a unique identity. You can merge or delete assets that do not have unique identities, such as duplicates and disconnected assets.

Setting or removing an implementation relationship

Logical data models and physical data models can be implemented by implemented data resources. You can set or remove an implementation relationship between either logical data model assets or physical data model assets and implemented data resources. You can also set or remove an implementation relationship between logical data model assets and physical data model assets.

Before you begin

You must have the Common Metadata Administrator role to set implementation relationships between two assets.

About this task

When you set an implementation relationship between two assets, the contained assets are automatically matched based on their names and type, and implementation relationships are set for both the top-level assets, and all the assets that the top-level assets contain. Assets that are not successfully matched appear as unmatched assets. To manually match unmatched assets, go to the View Contained Relationships workspace.

Procedure

1. Click the **Repository Management** tab.
2. In the Navigation pane search for an asset or browse for an asset. You can set implementation relationships between:
 - Logical data models and physical data models or database schemas
 - Physical data models and database schemas or data files

When you set an implementation relationship for an asset, corresponding implementation relationships are set between contained assets of like types that have the same name.

3. Select the logical data model or physical data model that you want to set an implementation relationship for and expand the **Implemented by** section on the right side of the screen.
4. Click **New Implementation Relationship**.
5. Select the asset that implements the original asset that you selected.

Option	Description
Browse for assets	Click the Browse for Asset tab, select the asset you want to set the relationship for, and click OK .
Search for assets	Click the Search for Asset tab, select the asset type that you want to search for from the drop-down menu, and click Search .

6. Review the status of the implementation relationship for the contained assets. Select **View contained relationships** before clicking **OK** to review the implementation relationships for matched contained assets or to set implementation relationships for unmatched assets.
7. In the View Contained Relationships workspace, click an asset in the implementation candidates column and drag it onto a target asset that you want to create an implementation relationship with. You can filter the source and target assets by type in the drop-down menus.
8. Optional: Click **Enable one to many matching** to allow one source asset to be matched with multiple target assets.
9. Review the implementation relationships in the **Implementation Relationships** workspace. Click **Remove** to remove an implementation relationship.

Related information:

Implementation relationships

You can set and explore relationships between logical data models, physical data models, and implemented data resources.

Logical data model assets

Logical data models capture the business definition of information assets by using the entity-relationship modeling approach. The logical data model consists of a set of related entities and their business associations.

Physical data model assets

A physical data model is a design schema for information assets that defines the physical structures and relationships of data within a subject domain or application. Physical data models are independent of implementation or platform details.

Implemented data resources

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

Merging assets

After browsing for assets or managing duplicate assets, you can merge two assets into one asset. Merging duplicate metadata assets makes it easier for suite tool users to choose the correct metadata to use in jobs. You can merge databases or database schemas.

Before you begin

You must have the Common Metadata Administrator role to merge assets. You should review merge rules before merging assets.

About this task

You can merge the following types of assets:

- Databases that are contained by the same host
- Database schemas that are contained by the same database

Procedure

1. In the Navigation pane on the **Repository Management** tab, click **Browse Assets** or **Manage Duplicates**.

Option	Description
To merge two assets from the Browse Assets section	<ol style="list-style-type: none"> 1. Select Implemented Data Resources in the Browse Assets section, and browse to select the database or database schema that you want to merge with another asset. 2. Click Merge. 3. In the Select an Asset to Merge window, click Duplicate Candidates, Browse for Asset, or Search for Asset to locate the asset that you want to merge the asset with. 4. Select the asset that you want to merge the starting asset with, and click OK.
To merge two assets from the Compare workspace when you are browsing assets or managing duplicates	Select two assets on the Compare workspace and click Merge .

The Merge workspace opens.

2. Optional: To swap the **Starting Asset** asset and the **Merge Into** asset, at the top of the **Merge** workspace, click **Swap Order**.
3. Review the usage for the starting asset, and then specify a usage option. Usage information indicates which tools have relationships with the starting asset. If you choose **Transfer specific usage information to merged asset**, you must then select the type of information to transfer.
4. Optional: Unselect **Delete Asset After Merge** if you do not want to delete the starting asset after the merge is complete.
5. Review the resulting asset details in the **Resulting Asset** column, and then click **Merge**. You see the following message when the merge completes successfully:

asset name successfully merged into *asset name*

Related tasks:

“Managing duplicate metadata assets” on page 63

You can compare, merge, or delete duplicate assets in the metadata repository.

Deleting duplicate assets helps ensure that suite tool users can choose the correct assets to use.

Related information:

Duplicate assets

Duplicate assets are assets that have the same identity or assets that have different identities but can be considered identical. You can manage duplicate assets by merging or deleting them.

Common metadata asset types

Each type of common metadata asset has a unique definition and unique identity components. Common metadata assets include implemented data resources, logical data model assets, physical data model assets, business intelligence assets, and some miscellaneous assets.

Merge rules

You use the merge functionality to combine two assets that you select into a single, merged asset. You can merge only databases or database schemas.

The ability to merge two assets gives you control over how to manage data that is stored in the metadata repository. When you merge two assets, the following

properties and relationships of the starting asset are overwritten by the properties of the asset that the starting asset is being merged into:

- Name
- Long description
- Creation tool
- Created by
- Created on
- Modified by
- Modified on
- Identity
- Host
- DBMS

For example, if the starting asset was created on Thursday January 13, 2010 at 9:00 AM and was created by admin100, and the asset you are merging the starting asset into was created on Friday January 14, 2010 at 10:22 AM and was created by admin202, the resulting asset would have a creation date of Friday January 14, 2010 at 10:22 AM and the created by field would list admin202.

Stewards from both assets are carried over to the resulting asset.

All assets contained by the databases, database schemas, and database tables that you are merging are carried over to the resulting asset.

You can control the usage relationships that get transferred to the resulting asset. Review the usage relationships to determine whether an asset is used in InfoSphere Information Server products, such as InfoSphere Information Analyzer or InfoSphere Business Glossary. You can transfer usage relationships from both assets to the resulting asset, transfer only specific usage relationships to the resulting asset, or you can choose to not transfer any usage relationships.

Deleting disconnected link points

You can delete disconnected link points from the metadata repository. Deleting disconnected link points improves performance and frees disk space on your system.

Before you begin

You must have the Common Metadata Administrator role to delete assets from the metadata repository. Before deleting an asset, make sure that you understand the delete rules for assets.

About this task

Disconnected link points occur when assets are exported from the metadata repository. When they are exported, they are exported with links to related assets that exist in the target system. The links are stored in the metadata repository, so that if the target asset is ever imported, the links are still active and working. The only way to clean up these links is to delete all links that are disconnected.

Procedure

1. Click the **Repository Management** tab in the InfoSphere Metadata Asset Manager.

2. Expand the **Manage Disconnected Assets** workspace in the Navigation menu.
3. Click **Delete Disconnected Link Points**.
4. Click **Yes** in the Delete Confirmation window.

Related tasks:

“Managing disconnected assets” on page 62

You can view the details of a disconnected implemented data resource or business intelligence (BI) asset or delete disconnected assets from the metadata repository. Disconnected assets do not have a complete identity because one or more assets in the identity hierarchy do not exist.

Chapter 3. Exchanging metadata by using IBM InfoSphere Metadata Integration Bridges

You can import metadata into the metadata repository of IBM InfoSphere Information Server from tools, files, and databases by using InfoSphere Metadata Asset Manager. You can export metadata from the metadata repository to various tools, files, and databases by using Import Export Manager.

Related information:

Importing and managing assets by using IBM InfoSphere Metadata Asset Manager
You can review and preview imports of metadata assets, and you can manage assets that are in the metadata repository.

How bridges work

You can use bridges to import or export metadata. Some bridges use IBM MetaBroker® technology.

Metadata is data about data. Metadata can describe the design, development, and implementation of information systems, and the movement of data. For example, the description of a database table is metadata. The relationships between database objects are metadata. The definitions of the columns in the table, the data types of the columns, and the derivations that transform the columns are metadata. Operational metadata is collected when you run InfoSphere DataStage and QualityStage jobs.

You can use bridges to import metadata from a particular tool, file, or database into the metadata repository, and to export metadata from the metadata repository. The bridge converts the information by translating the semantics of the source tool into the semantics of the target tool.

After you import into the metadata repository, users of suite tools such as InfoSphere Information Analyzer, InfoSphere Business Glossary, InfoSphere Metadata Workbench, and InfoSphere DataStage and QualityStage Designer can manipulate these metadata objects as required. When you use the Designer client, for example, you can use the imported tables in table definitions to create jobs. When you use the glossary, you can apply business terms to database columns and other objects. When you use the metadata workbench you can view the properties and relationships of individual metadata objects and create reports that show the flow of data through assets such as database columns, stages, jobs, and business intelligence (BI) reports. Metadata objects that are stored in the metadata repository of InfoSphere Information Server are referred to as *information assets*, or simply *assets*.

When suite tools create metadata, such as table definitions, database schemas, categories and terms, or physical data models, that metadata is automatically stored in the metadata repository.

Because bridges support Unicode, you can process data in a wide variety of languages and display data in any character set.

You can use bridges with InfoSphere Metadata Asset Manager to import the following types of assets:

- Implemented data resources, such as databases and data files and their contents
- Logical data models and physical data models from design tools such as CA ERwin and InfoSphere Data Architect
- Analysis results from profiling tools such as InfoSphere Discovery
- Dimensional models and reports from business intelligence (BI) tools such as IBM Cognos and SAP BusinessObjects
- Information about users of InfoSphere Information Server who can become stewards for assets in the metadata repository

You can use bridges with Import Export Manager to export the following types of assets:

- Implemented data resources
- Logical data models and physical data models to InfoSphere Data Architect
- Lists of users of InfoSphere Information Server

For a complete list of supported bridges and the types of assets they can import and export, see the technote *List of supported bridges for InfoSphere Information Server Version 8.7*: <http://www.ibm.com/support/docview.wss?&uid=swg27021980>.

Related tasks:

“Exporting metadata by using Import Export Manager” on page 84
 You can export implemented data resources, logical and physical data model assets, and user metadata from the metadata repository to tools and files.

“Running command-line imports” on page 87
 You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Related information:

Common metadata assets
 Common metadata assets are stored in the metadata repository and shared between tools in the IBM InfoSphere Information Server suite. Common metadata assets include logical and physical data model assets, implemented data resources, and business intelligence (BI) assets.

Importing and managing assets by using IBM InfoSphere Metadata Asset Manager
 You can review and preview imports of metadata assets, and you can manage assets that are in the metadata repository.

Requirements for using bridges

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

Requirements for using bridges apply to the client tier computer where the metadata interchange agent and the bridges are installed. For information on installing the metadata interchange agent and InfoSphere Metadata Integration Bridges, see the *IBM InfoSphere Information Server Planning, Installation, and Configuration Guide*.

For a list of supported bridges and their individual prerequisites, see the technote *List of supported bridges for InfoSphere Information Server Version 8.7*: <http://www.ibm.com/support/docview.wss?&uid=swg27021980>.

For a list of installation prerequisites, see the technote *System requirements for InfoSphere Metadata Integration Bridges*: <http://www.ibm.com/support/docview.wss?&uid=swg27021981>.

Related tasks:

“Exporting metadata by using Import Export Manager” on page 84
You can export implemented data resources, logical and physical data model assets, and user metadata from the metadata repository to tools and files.

“Running command-line imports” on page 87

You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Related information:

Importing and managing assets by using IBM InfoSphere Metadata Asset Manager
You can review and preview imports of metadata assets, and you can manage assets that are in the metadata repository.

Which tool do I use to run the bridges?

Use InfoSphere Metadata Asset Manager for imports. Use Import Export Manager for exports.

InfoSphere Metadata Asset Manager provides robust functionality for importing metadata. It is the preferred tool for bridge and connector imports.

You can use InfoSphere Metadata Asset Manager to import and manage metadata assets from applications, files, and databases. You import the metadata to a staging repository where you can analyze the source metadata and preview the results before you share it to the metadata repository. You can browse and search the metadata repository and manage common metadata assets.

Import Export Manager is used to export metadata. You can also use Import Export Manager to import metadata, but it does not have the advanced features of InfoSphere Metadata Asset Manager.

InfoSphere Metadata Asset Manager and Import Export Manager are available on client tier computers where the bridges are installed.

On client tier computers where both InfoSphere DataStage and QualityStage Designer and the bridges are installed, you can start bridge imports of implemented data resources from a menu in the Designer client.

Related tasks:

“Exporting metadata by using Import Export Manager” on page 84
You can export implemented data resources, logical and physical data model assets, and user metadata from the metadata repository to tools and files.

Related information:

Importing and managing assets by using IBM InfoSphere Metadata Asset Manager
You can review and preview imports of metadata assets, and you can manage assets that are in the metadata repository.

Importing BI metadata from IBM Cognos

You can use bridges to transfer metadata between IBM Cognos versions 8 and 10 and the metadata repository of IBM InfoSphere Information Server. You can import business intelligence (BI) reports, BI queries, BI models, and physical models. The import maintains the relationships between the BI models and the physical models.

Before you begin

To import from Cognos when using InfoSphere Information Server, Version 8.7, install prerequisite software that is listed in this technote: <http://www-01.ibm.com/support/docview.wss?uid=swg27038192>.

Related concepts:

“How bridges work” on page 71

You can use bridges to import or export metadata. Some bridges use IBM MetaBroker technology.

“Which tool do I use to run the bridges?” on page 73

Use InfoSphere Metadata Asset Manager for imports. Use Import Export Manager for exports.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

IBM Cognos BI Reporting - Content Manager bridge reference

Prerequisites, frequently asked questions, troubleshooting, and parameter information for the Cognos Content Manager bridge.

About this bridge

The IBM Cognos BI Reporting - Content Manager bridge supports Cognos 8 BI Reporting Content Manager functionality in Cognos ReportNet and Cognos versions 8 to 8.4 and 10. The bridge imports a complete package of business intelligence (BI) models, BI reports, and related implemented data resources.

BI reports are report templates that are created within the reporting tools Cognos Report Studio and Cognos Query Studio. BI reports include report fields. Some report fields are non-data fields, including page numbers and section headers, while others are data fields that retrieve or calculate data from a data source. BI reports include BI queries and query members that source and aggregate the information to display from BI models that are created in Cognos Framework Manager.

Important: When you upgrade to the latest version of the bridge by installing the prerequisite software, see the following technote for information on ensuring that your new imports reconcile with previous imports: <http://www-01.ibm.com/support/docview.wss?uid=swg21622045>.

Prerequisites

Frequently asked questions

Troubleshooting

“Import parameters” on page 77

Prerequisites

Install prerequisite software that is listed in this technote: <http://www-01.ibm.com/support/docview.wss?uid=swg27038192>.

Ensure that the following additional prerequisites are met before you run the bridge:

Connectivity

Before you can access Cognos Content Manager, its web services must be operational. Setting up these web services might require working with the networking proxy and firewall.

To test the connection, connect to Cognos Content Manager by using a web browser to make sure that it is accessible from the client computer. An example URL is `http://localhost:9300/p2pd/servlet`. If Cognos Content Manager is running and accessible, you can see a status page. The state of the server must be **running**.

To test if your authentication parameters work, use the web client tool from Cognos to verify the availability of the connection and authentication. An example URL is: `http://localhost/c8/cm_tester.htm`, where `localhost` is replaced with the appropriate IP name and port that is provided by Cognos. The bridge does not work if the connection and authentication are not available.

Cognos must be able to locate a gateway or dispatcher that is running on a web server that supports chunking and attachments to handle large volumes of data. If there is no firewall between users and Cognos, components use the default setting. If there is a firewall, you must have access to at least one web server that supports chunking outside of the firewall.

The `http` or `https` protocol prefix indicates whether SSL is required. You can find the value in the Cognos installation directory in the file `configuration\cogstartup.xml` file. For example:

```
<crn:parameter name="cdk"
<crn:value xsi-Type="xsd:anyURI">
http://localhost:9300/p2pd/servlet/dispatch</crn:value>
</crn:parameter>
```

Contact your Cognos administrator or Cognos support if necessary.

Permissions for users

IBM Cognos has five types of permissions: Read, Execute, Traverse, Write, and Set Policy. These permissions can be assigned or restricted for a user, group, or role.

Ensure that the user has the Read, Execute, and Traverse permissions assigned for all entries that are included in the import, (not just Execute and Traverse, as often recommended by the IBM Cognos documentation). Such entries include folders, reports, queries, analysis, packages, and connections. These permissions are read-only and do not change the Cognos contents. Many entries depend upon others. For example, packages use connections, reports use packages, and so on.

Data sources in IBM Cognos can be secured against multiple namespaces. In some environments, the namespace that is used to secure the data source is not the primary namespace that is used for access to IBM Cognos Connection. The bridge might need to access a report or other entry that is associated with a data source that secured against multiple namespaces. In such cases you must specify a user that has permissions for the required primary namespace. See the IBM Cognos documentation on permissions and security for more details.

Frequently asked questions

Why are multiple versions of a package extracted from Content Manager?

You can edit or update any design model in Cognos Framework Manager and publish it as a new version of a Framework Manager package in Content Manager. The Cognos development lifecycle requires that you then migrate any related reports to use this new version of the package in Content Manager. If you do not complete that migration for all such reports, some reports might still use an old version of a package. Remove old versions of a package that are no longer used. Some new versions of a package might not be used yet by any version. In such cases, multiple versions of a package might be used by different reports and thus imported.

How can I extract only the latest version of a package from Content Manager?

Select a single package to import, and set the **Add dependent objects** parameter to **False**. Only the latest version of the package is extracted.

Why are copied reports not imported when I specify an incremental harvesting date and time?

In Cognos, when you copy a folder that contains one or more reports, the modification time of each report is not updated. Incremental harvesting is based upon the report modification time. Copies are not imported with incremental harvesting whenever the date and time specified is later than the modification time of the original report.

Troubleshooting

To provide metadata for support cases, export your metadata from the Cognos 8.4 or Cognos 10 server into an archive file, such as a ZIP file:

1. Connect to IBM Cognos Connection by using the web browser.
2. Click **Launch > IBM Cognos Administration**.
3. Click **Configuration**.
4. Click **Content Administration**.
5. Click **New Export**.
6. Follow the instructions in the export wizard to export the metadata.
 - Avoid exporting the whole content store. To export specific folders and directory content, click **Select public folders and directory content** and follow the wizard pages to choose only the packages and folders that you want.
 - Do not include access permissions and references to namespaces other than IBM Cognos.
 - Set entry ownership to the user who performs the import in the target environment.
 - If you create a deployment archive, do not use spaces in the name.
7. After you run the export, send the archive file to the support team. The export can result in a single compressed file or a multi-volume archive file. The exported archive is usually located on the server in the folder `C:\Program Files\cognos\c8\deployment`.

For more information, see the IBM Cognos documentation for details about exporting metadata from Cognos.

Import parameters

The Cognos BI Reporting - Content Manager bridge uses the following import configuration parameters.

Version

Required. Select the version of Cognos server that you want to import from.

Options

- **Cognos 10.1**
- **Cognos 8.4**
- **Cognos 8.3**
- **Cognos 8.1 and 8.2**
- **Cognos ReportNet 1**

Dispatcher URL

Required. Type the URI that is used by the Framework Manager, Metrics Designer, or SDK to send requests to Cognos.

The value typically corresponds to the External dispatcher URI of one of the dispatchers in your installation, for example `http://Server:9300/p2pd/server/dispatch`. You must use the specific network host name or IP address instead of localhost. If the Framework Manager, Metrics Designer, or SDK clients connect to Cognos through an intermediary such as a load balancer or proxy, specify the host and port of the intermediary. For more information, see the bridge prerequisites.

Namespace

Leave blank if Cognos authentication is not configured.

A namespace defines a collection of user accounts from an authentication provider. See *Authentication Providers* in the *Cognos ReportNet Installation and Configuration Guide*.

User name

Type the user name to use if Cognos authentication is configured. Leave blank if Cognos authentication is not configured. This import bridge is read-only and never affects the IBM Cognos contents. It is safe to attempt the initial metadata import as Administrator in order to ensure that the entire content is extracted without access permission issues. Eventually, the administrator can set up a read-only user or group. For more information, see the bridge prerequisites.

Password

Type the password if Cognos authentication is configured. Leave blank if Cognos authentication is not configured.

Cognos assets to import

Required. Browse to select the assets that you want to import, or type a content string that specifies one or more search paths. You can select a smaller set of objects than the whole server content.

A content string is a semicolon-separated list of individual Cognos search paths that are used to retrieve objects from Cognos. The following object types are supported: package, folder, model, report, query, and shortcut. See the Cognos documentation for the full search path syntax.

Search paths that attempt to retrieve everything under a certain folder or content root are inefficient. The import might run for a long time or cause errors on the Cognos server. Instead of using `/**`, use more specific search paths, such as these search paths:

- `/**[@objectClass='query'`
- `@objectClass='report'`
- `@objectClass='model']`

Use a backslash character (`\`) to escape each semicolon (`;`) and or backslash character in the content string. You can retrieve models by package name, for example `/content/package[@name='GO Sales and Retailers']/model`. If there are multiple published versions, the latest is imported.

You can retrieve reports by using the complete search path. To find the complete search path in Cognos, click **View the search path** on the properties page for the report. An example is `/content/package[@name='GO Sales and Retailers']/folder[@name='Documentation Report Samples']/report[@name='Create a Prompt']`.

If a query returns multiple models or reports, only the last model or report is imported. The following queries return multiple reports:

- `//report` returns all reports
- `/content/package[@name='GO Sales and Retailers']//report` returns all reports in a package

Add dependent objects

You can add dependent objects to the initial selection of Cognos objects that you selected with the parameter **Cognos assets to import**. By default, this option is selected. Only the first level of dependent objects is added. That is, when a report search path is specified, its source model is imported, but other reports that depend on the model are not imported.

Folder representation

Accept the default value, **Ignore**, which ignores folders from Cognos Framework Manager. The default automatically captures the Cognos server and the package or folder location of the report.

Import joins

You can import joins that are defined in the BI model. By default, joins are not imported.

Import levels

You can import levels and hierarchies. When the option is selected, levels and hierarchies that are defined in the BI model are imported. By default, they are not imported.

Related information:

Business intelligence assets

Business intelligence (BI) assets are used by BI tools to organize reports and models that provide a business view of data. These assets include BI reports, BI models, BI collections, and cubes.

Implemented data resources

Implemented data resources include databases and their contents, data files and their contents, and host computers that contain databases and data files. Data item definitions, which represent complex data structures, are also implemented data resources.

Importing metadata by using InfoSphere Metadata Asset Manager

You can import by using a bridge or connector that is on any computer that is designated as a metadata interchange server. You specify connection information and information about the source metadata, and choose to run either an express import or a managed import.

Bridges that use IBM MetaBroker technology

Some bridges use MetaBroker technology to extend their functionality. You can use them to import implemented data resources, analysis information, logical and physical data model assets, and user information.

DBM File MetaBroker

You can use DBM File MetaBroker to import DBM (database model) files from InfoSphere Discovery into the metadata repository of InfoSphere Information Server. The files can contain implemented data resources and analysis results.

The bridge imports DBM files that are created by InfoSphere Discovery, Version 4.1.1.6 or later.

Use IBM InfoSphere Data Architect MetaBroker to import DBM files from InfoSphere Data Architect.

It is not necessary to install InfoSphere Discovery to import a DBM file by using DBM File MetaBroker.

Important: Use DBM File MetaBroker to import files that contain tables from a single database, not tables from multiple databases. You must enter the host computer name, database name, database management system (DBMS), and DBMS instance name as parameters for the import.

DBM File MetaBroker supports the three model types that are produced by InfoSphere Discovery : generic SQL, IBM DB2, and Oracle.

DBM File MetaBroker does not import categories and terms or references to categories and terms.

You can import the following types of assets into the metadata repository when you use DBM File MetaBroker:

- Implemented data resources:
 - host computers
 - databases
 - database schemas
 - stored procedures

- database tables
- database columns
- foreign keys
- candidate keys
- indexes
- Analysis assets:
 - table analysis properties
 - column analysis properties
 - data classification codes
 - inferred candidate keys
 - inferred foreign keys

The following table lists the product modules that can view and work with the implemented data resources and analysis assets that are stored in the metadata repository:

Table 12. Product modules that work with metadata that is imported by using DBM File MetaBroker

Product module	Types of metadata that you can work with
InfoSphere Information Analyzer	Implemented data resources and analysis assets
InfoSphere DataStage and QualityStage	Implemented data resources with annotations that include analysis information
<ul style="list-style-type: none"> • InfoSphere Business Glossary • InfoSphere FastTrack • InfoSphere Metadata Workbench 	<ul style="list-style-type: none"> • Implemented data resources • Table analysis summaries and column analysis summaries • Inferred keys displayed as attributes of analysis summaries

Scenario for importing and using DBM files

A user of InfoSphere Discovery highlights relationships between database tables that comprise a data set in a project. By choosing the **Export to Optim™** option, the user creates a DBM file that contains the data set.

A data analyst uses DBM File MetaBroker to import the file into InfoSphere Information Server.

Working in InfoSphere Information Analyzer, the data analyst assesses the data quality and provides reports and annotations to subject matter experts and developers. The analyst also establishes data rules for ongoing validation.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

IBM InfoSphere Data Architect MetaBroker

You can use IBM InfoSphere Data Architect MetaBroker to transfer metadata between InfoSphere Data Architect and the metadata repository of InfoSphere Information Server.

The bridge supports the following versions of InfoSphere Data Architect:

- 7.5.1, 7.5.2, 7.5.3.
- 7.6.
- 8.1.

Imports are not supported from IBM Rational Data Architect, Version 7.5.

Support for versions 7.6 and 8.1 requires that you install prerequisite software that is listed in this technote: <http://www-01.ibm.com/support/docview.wss?uid=swg27038192>.

The bridge can transfer the following types of metadata:

- Physical data model assets (DBM files)
- Logical data model assets (LDM files)

Whether you import related physical data models and logical data models separately or at the same time, the relationships between them are preserved when both have been imported into the metadata repository.

When you use the bridge to import a DBM file from InfoSphere Data Architect, a corresponding set of implemented data resources is created, including host, database, database schema, and database tables, based on identity parameters that you specify values for. Users of InfoSphere DataStage and QualityStage can create table definitions from the database tables and use them in jobs.

The IBM InfoSphere Data Architect MetaBroker does not support transferring naming models and glossary metadata between the metadata repository and InfoSphere Data Architect. For that purpose, use the import and export functionality in InfoSphere Business Glossary.

You can run the bridge from the following tools:

- InfoSphere Metadata Asset Manager (imports to the metadata repository)
- Import Export Manager (exports from the metadata repository)

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

ODBC 3.0 MetaBroker

You can use ODBC 3.0 MetaBroker to import metadata from databases into the metadata repository of InfoSphere Information Server.

With the ODBC 3.0 MetaBroker, you can import metadata from databases that support the ODBC 3.0 protocol. When you import, you can prefilter the metadata to select only some of the objects in the database.

You start the import process in InfoSphere Metadata Asset Manager or in IBM InfoSphere DataStage and QualityStage Designer.

If you import by using the Designer client, the imported metadata is limited to tables and columns. The tables become shared tables in the metadata repository that are linked to table definitions in the Designer client.

If you import by using InfoSphere Metadata Asset Manager, additional metadata objects are stored in the metadata repository. However, you must register interest in the imported tables to use them in the Designer client.

The ODBC 3.0 MetaBroker does not support imports from Microsoft Access.

Note: For better performance on large imports, use the ODBC Connector instead of the ODBC 3.0 MetaBroker. You can import by using the ODBC Connector in InfoSphere Metadata Asset Manager.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

User Information MetaBroker

You can import files that contain people who can be designated as stewards. You can export user information from the metadata repository.

Some people who are responsible for assets that are stored in the metadata repository of InfoSphere Information Server are not users of the suite tools. If you want to designate such people as stewards for one or more assets, you can import the contact information for those people by using the User Information MetaBroker.

This import saves the time that it takes to enter the information individually for each user. You can use InfoSphere Business Glossary to designate these users as stewards.

The users that are imported by this bridge cannot be used to authenticate with the repository and are not displayed as users in the Administration tab of InfoSphere Information Server Web console.

You cannot use the bridge to import or export password information. The bridge does not import or export information about relationships between users or between users and other objects in the metadata repository.

When you import the user information, the import file must have the appropriate comma-separated values (CSV) file format. Sample CSV files and CSV template files are provided in the installation folder. By default, the installation folder is C:\IBM\InformationServer\Clients\MetaBrokersAndBridges\User Information MetaBroker. By using these files, you can type in the same input that you can supply in the Administration tab of the Web console.

When you export user information, a CSV file of the same format is created.

You import by using InfoSphere Metadata Asset Manager and export by using Import Export Manager.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

File format for User Information MetaBroker:

You can create comma-separated value (CSV) files with this format to import information about users in InfoSphere Information Server.

Purpose

You can import and export information about users of InfoSphere Information Server by using User Information MetaBroker. The CSV file that you import and export contains rows of users. Each cell of the row has an attribute for that user, for example, e-mail address and phone number.

The files that you import into the metadata repository must match the format that is described in this topic. See the sample CSV files in the bridge installation folder for more details. By default, the installation folder is C:\IBM\InformationServer\Clients\MetaBrokersAndBridges\User Information MetaBroker. Files that you export from the metadata repository are created in the format of the sample files.

Format of User Information CSV file

Each row in the CSV file is a header row, a comment row, or a value row.

Header rows begin with the name of the class of metadata to be imported, followed by the names of attributes and relationships that you can populate.

Comment rows begin with a number sign (#). In the sample file, a comment row immediately beneath a header row shows the types of information that must be furnished for each column, the valid values for the column, and whether a value is optional or mandatory.

Comment rows in the sample file can include the following elements:

- Value indicators indicate where values are placed. They are surrounded by angle brackets.

- Indicators for optional values are surrounded by angle brackets and square brackets, for example, [<value>].
- The pipe symbol (|) is used to indicate or in a list of values.
- The word value indicates that the value is a string. For example, <value> indicates that the value is a string and is mandatory, and [<value>] indicates that the value is a string and is optional.
- <ID> indicates that the value is the unique, mandatory, numeric identifier for each row.

Each value row begins with a unique numeric identifier that identifies the object to be imported. Value rows relate to the header row that precedes them. If a value row does not populate the last properties that are specified by the header row, you can use trailing commas to indicate empty values. You must use commas to indicate empty values in the middle of a value row.

Exporting and importing metadata by using Import Export Manager

You can export implemented data resources, logical and physical data model assets, and user information from the metadata repository. You can also import metadata by using Import Export Manager, but InfoSphere Metadata Asset Manager is the preferred import tool because of its advanced functionality.

Exporting metadata by using Import Export Manager

You can export implemented data resources, logical and physical data model assets, and user metadata from the metadata repository to tools and files.

Before you begin

Ensure that you meet the software requirements for using bridges.

For a list of supported bridges and their individual prerequisites, see the technote *List of supported bridges for InfoSphere Information Server Version 8.7*: <http://www.ibm.com/support/docview.wss?&uid=swg27021980>.

The length of any parameter value that you enter in the Import Export Manager user interface cannot exceed 30,000 characters or the export fails.

Procedure

1. Double-click the Import Export Manager icon in your desktop to open Import Export Manager.
2. Click **Export metadata**.
3. Select the bridge from the **Target** list and click **Start**.
4. In the Parameter Selection window, specify parameters for connecting to InfoSphere Information Server. To display help for any parameter, click the name of the parameter.
5. In the Metadata Prefilter window, select the metadata instances to export and click **OK**. The Status window opens and the bridge begins reading the data from the metadata repository.
6. When the process is complete, click **Select All** to export all the metadata, or click **Filter** to open the Metadata Selection window and select some of the metadata to export.
7. In the Parameter Selection window, specify parameters for exporting metadata to the target.

- Optional: If the target tool requires, specify credentials and connection information. The Status window opens and the bridge begins to write data to the target.
- When the process is complete, click **Finish** in the Status window.

What to do next

You can view the exported metadata in the file or tool that you exported it to.

Related concepts:

“How bridges work” on page 71

You can use bridges to import or export metadata. Some bridges use IBM MetaBroker technology.

“Which tool do I use to run the bridges?” on page 73

Use InfoSphere Metadata Asset Manager for imports. Use Import Export Manager for exports.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

Importing metadata by using Import Export Manager

You can import implemented data resources, logical and physical data model assets, business intelligence assets, analysis metadata, and user metadata into the metadata repository.

Before you begin

You must meet the software requirements for the bridge that you want to use.

For a list of supported bridges and their individual prerequisites, see the technote *List of supported bridges for InfoSphere Information Server Version 8.7*:

<http://www.ibm.com/support/docview.wss?&uid=swg27021980>.

The length of any parameter value that you enter in the Import Export Manager user interface cannot exceed 30,000 characters or the import will fail. If asset filtering strings exceed that length, use the command line for the import.

Procedure

- Double-click the Import Export Manager desktop icon to open Import Export Manager.
- Select the **Import metadata** tab.
- Select the bridge from the **Source** list and click **Start**.
- In the Parameter Selection window, specify the parameters to connect to IBM InfoSphere Information Server, and specify parameters for logging and handling of duplicate assets. To display help for any parameter, click the name of the parameter. When you click **OK**, the parameter selection window for the import bridge opens.
- In the Parameter Selection window for the import bridge, specify parameters for importing metadata from the source tool, file, or database.
 - Type the credentials and connection information for the import source.

- b. Type values for additional parameters in the Additional Parameters pane.
The information that you provide in this pane is critical to reconcile the imported metadata with previously imported metadata from the same source. This information is also critical to the ability of IBM InfoSphere Metadata Workbench to match imported database assets to jobs and to other assets in the metadata repository.
 - c. Optional: Save the parameter settings to a file that you can use to run imports on the command line.
6. Click **OK**. The Status window opens and the bridge begins to read the data from the source tool. For some bridges, the Metadata Prefilter window opens and you can prefilter the metadata before it is retrieved from the source tool.
7. When the process is complete, click **Select All** to import all the metadata, or click **Filter** to open the Metadata Selection window and select some of the metadata to import. If you filter the metadata, you can save your filter settings to a file to use on the command line.
8. When the import is complete, click **Finish** in the Status window.

What to do next

When you import tables from physical data models, databases, or business intelligence tools directly into the metadata repository, the database tables that are imported are shared tables. To use these tables in IBM InfoSphere DataStage and QualityStage jobs, you must create table definitions from them in the Designer client.

You can view and manage the imported assets on the **Repository Management** tab of InfoSphere Metadata Asset Manager. You can work with the imported metadata in other suite tools.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

Related information:

Common metadata assets

Common metadata assets are stored in the metadata repository and shared between tools in the IBM InfoSphere Information Server suite. Common metadata assets include logical and physical data model assets, implemented data resources, and business intelligence (BI) assets.

Importing metadata by using the command line

You can use the Import Export Manager command-line interface to run bridges that import metadata from tools, files, and databases. For example, you can issue command-line parameters that run a batch collection of imports with different bridges. Or you can include commands in a script, and use third-party scheduling software to schedule an import process that runs unattended.

Related tasks:

“Importing metadata by using Import Export Manager” on page 85

You can import implemented data resources, logical and physical data model assets, business intelligence assets, analysis metadata, and user metadata into the metadata repository.

Running command-line imports

You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Before you begin

Ensure that your computer meets the software requirements for the bridge that you want to use.

About this task

You use the Import Export Manager tool to save parameter values and filtering criteria to files, and then you pass those files as arguments when you issue the **ImportExportManager** command to import metadata.

You can import metadata directly into the metadata repository, or you can import the metadata in stages by using an intermediate file. When you use an intermediate file, you bypass the typical process that is used to encode and decode the metadata. Instead of writing metadata directly to the metadata repository, the import process creates an intermediate file that you can use in subsequent imports. This approach is useful if you are importing a large metadata file (larger than 1 MB), and you expect to run the import process multiple times. Using an intermediate file also provides greater flexibility for comparing multiple imports.

Procedure

1. Create a parameter XML file by performing the following steps. The parameter XML file provides parameter values that are used by the bridges when you import metadata by using the command line.
 - a. Double-click the Import Export Manager desktop icon to open Import Export Manager.
 - b. Select the **Import metadata** tab.
 - c. Select the bridge from the **Source** list and click **Start**.
 - d. In the Parameter Selection window, specify the parameters to connect to InfoSphere Information Server, and specify parameters for logging and handling of duplicate assets. To display help for any parameter, click the name of the parameter. Click **OK**.
 - e. In the Parameter Selection window for the import bridge, specify parameters for importing metadata from the source tool, file, or database. For more information about specifying these parameters, see the topic about using Import Export Manager to import metadata.
 - f. Click the **Save Parameters** check box, and click **OK**.
 - g. Browse and save the parameter XML file.

Tip: To create a parameter XML file that contains parameters for more than one bridge, use Import Export Manager to create multiple parameter XML

files. After creating the files, manually consolidate the files into a single file that has multiple <Import> elements enclosed by a single <Parameters> element.

2. Optional: If you want to filter the contents of the metadata, create a filter file by performing the following steps:
 - a. Continue in Import Export Manager until you see the Status window.
 - b. In the Status window, click the **Filter** button.
 - c. In the Metadata Selection window, select the assets that you want to include when you import the metadata.
 - d. Select the **Save Filter** check box and click **OK**.
 - e. Browse and save the file.
 - f. Use a text editor to edit the parameter XML file that you created earlier, and add the following line to the <Bridge> element:

```
<Param Name="Filter_File">C:\filter.flt</Param>
```

where *C:\filter.flt* is the filter file.

3. Click **Cancel** to close Import Export Manager.
4. Use an XML editor or a text editor to create the IS parameter XML file. The IS parameter XML file provides mandatory parameter values that you must pass to the **ImportExportManager** command to access the metadata repository. For more information, see “Format of the IS parameter XML file” on page 93.
5. Validate the parameter XML files by issuing the **ImportExportManager** command with the **-validate** parameter specified.

For example, the following command validates the parameter XML file, param.xml:

```
ImportExportManager -validate -paramfile C:\param.xml
```

The following command validates the IS parameter XML file, isparam.xml:

```
ImportExportManager -validate -paramfile C:\isparam.xml
```

6. Import the metadata by issuing the **ImportExportManager** command with one of the following import parameters specified. For more information about the syntax of this command, see “Parameters for importing metadata” on page 90.

Option	Description
-import	Imports metadata into the metadata repository by using a specific bridge. Optional: To create an intermediate file rather than writing metadata directly to the metadata repository, also specify the -importtofile parameter.
-importall	Imports metadata into the metadata repository by using all bridges that are specified in the parameter XML file.
-directimport	Uses the intermediate file that you specified with the -importtofile parameter to import metadata into the metadata repository.

For example, the following command uses the bridge associated with the ID “IDAMB1” in the parameter XML file, param.xml, to import metadata into the metadata repository:

```
ImportExportManager -import IDAMB1 -paramfile C:\param.xml  
-isparamfile C:\isparam.xml -server Server26 -verbose N
```

What to do next

When you import tables from physical data models, databases, or business intelligence tools directly into the metadata repository, the database tables that are imported are shared tables. To use these tables in IBM InfoSphere DataStage and QualityStage jobs, you must create table definitions from them in the Designer client.

You can view and manage the imported assets on the **Repository Management** tab of InfoSphere Metadata Asset Manager. You can work with the imported metadata in other suite tools.

Related reference:

“Parameters for importing metadata” on page 90

The **ImportExportManager** command includes parameters that control the behavior of the import process.

“Parameters for validating the parameter XML files” on page 92

The **ImportExportManager** command includes parameters for validating the parameter XML file and the IS parameter XML file. You must validate these files to encapsulate hidden parameters such as passwords.

“Format of the IS parameter XML file” on page 93

The IS parameter XML file contains parameter values that are required by a bridge to access the metadata repository. When you run the **ImportExportManager** command to import metadata, you use the **-isparamfile** parameter to specify the fully qualified name of this file.

ImportExportManager command

Use the **ImportExportManager** command to import metadata into the IBM InfoSphere Information Server metadata repository.

Purpose

The **ImportExportManager** command imports the following types of metadata into the metadata repository:

- Implemented data resources
- Logical and physical data model assets
- Business intelligence (BI) reports and models
- Analysis information
- Information about users of InfoSphere Information Server

The command also includes parameters for validating the structure of the parameter XML files that are used by bridges to control the import of data.

Run the **ImportExportManager** command from a command window in the `MetabrokersAndBridges` folder, which is installed by default in: `C:\Program Files\Common Files\IBM\InformationServer\MetabrokersAndBridges`. The command invokes `ImportExportManager.exe`, which is located in that folder.

Tip: You can access help from the command line by issuing the following command:

```
ImportExportManager -help | -h
```

Parameters for importing metadata:

The **ImportExportManager** command includes parameters that control the behavior of the import process.

You can import metadata directly into the metadata repository, or you can import the metadata in stages by using an intermediate file. When you use an intermediate file, you bypass the typical process that is used to encode and decode the metadata. Instead of writing metadata directly to the metadata repository, the import process creates an intermediate file that you can use in subsequent imports. This approach is useful if you are importing a large metadata file (larger than 1 MB), and you expect to run the import process multiple times. Using an intermediate file also provides greater flexibility for comparing multiple imports.

Syntax for importing metadata

```
ImportExportManager {-import {ID | Bridge Name} | -importall [-continue]}  
-paramfile Parameter File -isparamfile IS Parameter File -server Server Name  
[-verbose Y | N] [-logfile Log File]
```

Syntax for creating an intermediate file for importing metadata

```
ImportExportManager -import {ID | Bridge Name} -paramfile Parameter File  
[-isparamfile IS Parameter File] [-server Server Name] -importtofile Import File  
[-verbose Y | N] [-logfile Log File]
```

Syntax for importing metadata by using an intermediate file

```
ImportExportManager -directimport Import File -isparamfile IS Parameter File -server  
Server Name [-verbose Y | N] [-logfile Log File]
```

Parameters

-import

Imports metadata by using bridge and associated parameter values that are specified for *Bridge Name* or *ID* in the parameter XML file, *Parameter File*.

Bridge Name

The name of the bridge enclosed in quotation marks. For example, "SAP Business Objects Designer (Repository)" is a valid bridge name.

ID The ID that is specified for the block in the parameter XML file, *Parameter File*, where parameters for the bridge are defined.

-importall

Imports metadata by using all bridges and associated parameter values that are specified in the parameter XML file, *Parameter File*.

-continue

Specifies that the import process continues when a failure is encountered. If this parameter is not specified, the import process ends if it encounters a failure.

-paramfile *Parameter File*

Specifies the fully qualified name of the parameter XML file, *Parameter File*. This file provides parameter values that are used by the bridges to import metadata. You must create a parameter XML file before you can issue the **ImportExportManager** command to import metadata. For information about creating this file, see "Running command-line imports" on page 87.

-isparamfile *IS Parameter File*

Specifies the fully qualified name of the IS parameter XML file, *IS Parameter File*. This file provides parameter values that are required to access the metadata repository when you import metadata by using the command line. You must create this file manually. For more information, see “Format of the IS parameter XML file” on page 93.

-server *Server Name*

The ID that is specified for the <ImportExport> element in *IS Parameter File*. The <ImportExport> element contains parameter values that specify the user ID, password, host, and port number for accessing the metadata repository.

-importtofile *Import File*

Creates an intermediate file, *Import File*, rather than writing metadata directly to the metadata repository. You pass the name of this file to the **-directimport** parameter. The name of *Import File* must be fully qualified.

-directimport *Import File*

Uses the intermediate file, *Import File*, that you specified with the **-importtofile** parameter to import metadata into the metadata repository. The name of *Import File* must be fully qualified.

-verbose *Y|N*

Specifies whether logging information is saved to a temporary folder for all bridges that are used during the import sequence. If no value is specified, the default is Y.

-logfile *Log File*

Specifies the fully qualified name of the log file, *Log File*, to which the **Import Export Manager** writes logging information. If you do not specify this parameter, the log file is written to `ImpExpManagerCmdRun.log` in the `%temp%` directory.

Examples of importing metadata into the metadata repository

The following command uses the SAP Business Objects Designer (Repository) bridge to import metadata into the metadata repository:

```
ImportExportManager -import "SAP Business Objects Designer (Repository)"  
-paramfile C:\param.xml -isparamfile C:\isparam.xml -server Server26 -verbose N
```

The following command uses the bridge associated with the ID “erwin1” in the parameter XML file, `param.xml`, to import metadata into the metadata repository:

```
ImportExportManager -import erwin1 -paramfile C:\param.xml  
-isparamfile C:\isparam.xml -server Server26 -verbose N
```

The following command imports metadata for all bridges that are specified in the parameter XML file, `param.xml`. If a failure occurs, the import process continues.

```
ImportExportManager -importall -continue -paramfile C:\param.xml  
-isparamfile C:\isparam.xml -server Server26 -verbose N
```

Examples of importing metadata by using an intermediate file

The following command uses the bridge associated with the ID “erwin1” in the parameter XML file, `param.xml`, to create an intermediate file, `erwin1.encoder.xml`:

```
ImportExportManager -import erwin1 -paramfile C:\param.xml  
-verbose Y -importtofile C:\erwin1.encoder.xml
```

The following command uses the intermediate file, `erwin1.encoder.xml`, to import metadata into the metadata repository:

```
ImportExportManager -directimport C:\erwin1.encoder.xml
-isparamfile C:\isparam.xml -server Server26
-logfile C:\XML29.tmp.xml.log
```

Related tasks:

“Running command-line imports” on page 87

You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Related reference:

“Format of the IS parameter XML file” on page 93

The IS parameter XML file contains parameter values that are required by a bridge to access the metadata repository. When you run the **ImportExportManager** command to import metadata, you use the **-isparamfile** parameter to specify the fully qualified name of this file.

Parameters for validating the parameter XML files:

The **ImportExportManager** command includes parameters for validating the parameter XML file and the IS parameter XML file. You must validate these files to encapsulate hidden parameters such as passwords.

Syntax

```
ImportExportManager -paramfile {Parameter File | IS Parameter File} -validate
[-logfile Log File]
```

Parameters

-paramfile *Parameter File* | *IS Parameter File*

Specifies the fully qualified name of the parameter XML file, *Parameter File* or *IS Parameter File*, that contains parameter values that are used by the bridges to import metadata.

-validate

Validates the XML file to encapsulate hidden parameters such as passwords.

-logfile *Log File*

Specifies the fully qualified name of the log file, *Log File*, to which the **Import Export Manager** writes logging information. If you do not specify this parameter, the log file is written to `ImpExpManagerCmdRun.log` in the `%temp%` directory.

Example

The following command validates the parameter XML file, `param.xml`:

```
ImportExportManager -validate -paramfile C:\param.xml
```

The following command validates the IS parameter XML file, `isparam.xml`:

```
ImportExportManager -validate -paramfile C:\isparam.xml
```

Related tasks:

“Running command-line imports” on page 87

You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Related reference:

“Format of the IS parameter XML file”

The IS parameter XML file contains parameter values that are required by a bridge to access the metadata repository. When you run the **ImportExportManager** command to import metadata, you use the **-isparamfile** parameter to specify the fully qualified name of this file.

Format of the IS parameter XML file:

The IS parameter XML file contains parameter values that are required by a bridge to access the metadata repository. When you run the **ImportExportManager** command to import metadata, you use the **-isparamfile** parameter to specify the fully qualified name of this file.

Purpose

The IS parameter XML file provides mandatory parameter values that you must pass to the **ImportExportManager** command to access the metadata repository. Because the IS parameter XML file contains sensitive information, such as user names and passwords, you can specify parameter values in this file and then run the **ImportExportManager** command with the **-validate** parameter specified to encrypt sensitive information. The IS parameter XML file must reside in a directory that is accessible to the **ImportExportManager.exe** program.

Format

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
  <ImportExport id="Server Name">
    <Bridge>
      <Param Name="ISUser">User ID</Param>
      <Param Name="ISPassword" Hidden="true" Encrypted="false">Password</Param>
      <Param Name="ISHost">Host</Param>
      <Param Name="ISPort">Port</Param>
    </Bridge>
  </ImportExport>
</Parameters>
```

<Parameters> element

The root element of the IS parameter XML file. This element contains one or more <ImportExport> elements.

<ImportExport> element

Contains one <Bridge> element. The value of the ID attribute, *Server Name*, uniquely identifies each <ImportExport> element in the XML file.

<Bridge> element

Contains four <Param> child elements. The <Param> elements specify the following connection details for the metadata repository:

User ID

Specifies a valid InfoSphere Information Server user name on the computer where the metadata repository resides.

Password

Specifies the password for the user specified as *User ID*

Host

Optional. Specifies the host name of the computer where the metadata repository resides. If no value is specified, the default is localhost.

Port

Optional. Specifies the port number of the computer where the metadata repository resides. If no value is specified, the default port number is 9080.

To indicate that the content of a <Param> element needs to be encrypted, specify the attribute `Hidden="true"`, and run the **ImportExportManager** command with the **-validate** parameter specified. When you run this command, all hidden parameters in the file are encrypted, and the `Encrypted` attribute is set to "true".

The following IS parameter XML file has been validated. It specifies connection parameters for a server where the metadata repository resides:

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
  <ImportExport id="Server26">
    <Bridge>
      <Param Name="ISUser">admin</Param>
      <Param Name="ISPassword" Hidden="true" Encrypted="true">LD8@1:VA<9:<0FD</Param>
      <Param Name="ISHost">9.184.184.26</Param>
      <Param Name="ISPort">9080</Param>
    </Bridge>
  </ImportExport>
</Parameters>
```

Related tasks:

“Running command-line imports” on page 87

You can import the implemented data resources, logical and physical data models, business intelligence (BI) assets, analysis information, and user information into the metadata repository.

Related reference:

“Parameters for importing metadata” on page 90

The **ImportExportManager** command includes parameters that control the behavior of the import process.

“Parameters for validating the parameter XML files” on page 92

The **ImportExportManager** command includes parameters for validating the parameter XML file and the IS parameter XML file. You must validate these files to encapsulate hidden parameters such as passwords.

Importing metadata by using the Designer client

You can import metadata about database tables, database columns, and data elements into the repository of IBM InfoSphere Information Server. You can use this metadata to create jobs in IBM InfoSphere DataStage and QualityStage Designer.

Before you begin

- The Designer client and the bridges must both be installed on the same Microsoft Windows computer.
- You must meet the software requirements for the bridge that you want to use.
- You must have one of the following roles in InfoSphere DataStage and QualityStage:

- DataStage and QualityStage Administrator role
- DataStage and QualityStage User with DataStage Developer role
- DataStage and QualityStage User with DataStage Production Manager role

Procedure

1. Open the client and connect to the server that has the metadata repository that you want to import metadata into.
2. Select **Import > Via Bridges**.
3. In the Bridge Selection window, select a bridge and click **OK**.
4. In the Parameter Selection window, specify parameters for importing metadata from the source tool. Some source tools require more input:
 - a. Type the credentials and connection information for the source tool.
 - b. Type additional parameters in the Additional Parameters pane.
The information that you provide in this pane is critical to reconcile the imported metadata with previously imported metadata from the same source. This information is also critical to the ability of IBM InfoSphere Metadata Workbench to match imported database assets to jobs and to other assets in the metadata repository.
 - c. Optional: Save the parameter settings to a file that you can use to run imports on the command line.
5. Click **OK**. The Status window opens and the bridge begins to read the data from the source tool. For some bridges, the Metadata Prefilter window opens and you can prefilter the metadata before it is retrieved from the source tool.
6. When the process is complete, click **Select All** to import all the metadata, or click **Filter** to open the Metadata Selection window and select some of the metadata to import. If you filter the metadata, you can save your filter settings to a file to use on the command line.
7. Click **OK** to start to encode the data for import to the metadata repository.
8. When the import is complete, click **Finish** in the Status window.
9. The Select Folder window opens. Select a folder to create a shared table of your imported data. Click **OK**.

What to do next

When you import tables from physical models, databases, or business intelligence tools by using the Designer client, any tables that you import become shared tables, or data collections, in the metadata repository. You can create table definitions with the shared tables and use the imported metadata in jobs.

Related reference:

“Requirements for using bridges” on page 72

Before you can import or export metadata, you must install required software programs, including Microsoft XML Core Services (MSXML) 4.0 Service Pack 2. Some bridges have additional requirements that are detailed in the list of supported bridges.

Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 13. IBM resources

Resource	Description and location
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/
My IBM	You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/
Training and certification	You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at http://www.ibm.com/software/sw-training/
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Providing feedback

The following table describes how to provide feedback to IBM about products and product documentation.

Table 14. Providing feedback to IBM

Type of feedback	Action
Product feedback	You can provide general product feedback through the Consumability Survey at www.ibm.com/software/data/info/consumability-survey

Table 14. Providing feedback to IBM (continued)

Type of feedback	Action
Documentation feedback	<p>To comment on the information center, click the Feedback link on the top right side of any topic in the information center. You can also send comments about PDF file books, the information center, or any other documentation in the following ways:</p> <ul style="list-style-type: none"><li data-bbox="933 436 1414 495">• Online reader comment form: www.ibm.com/software/data/rcf/<li data-bbox="933 499 1414 531">• E-mail: comments@us.ibm.com

Accessing product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product client interfaces, in a suite-wide information center, and in PDF file books.

The information center is installed as a common service with IBM InfoSphere Information Server. The information center contains help for most of the product interfaces, as well as complete documentation for all the product modules in the suite. You can open the information center from the installed product or from a Web browser.

Accessing the information center

You can use the following methods to open the installed information center.

- Click the **Help** link in the upper right of the client interface.

Note: From IBM InfoSphere FastTrack and IBM InfoSphere Information Server Manager, the main Help item opens a local help system. Choose **Help > Open Info Center** to open the full suite information center.

- Press the F1 key. The F1 key typically opens the topic that describes the current context of the client interface.

Note: The F1 key does not work in Web clients.

- Use a Web browser to access the installed information center even when you are not logged in to the product. Enter the following address in a Web browser: `http://host_name:port_number/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.home.doc/ic-homepage.html`. The `host_name` is the name of the services tier computer where the information center is installed, and `port_number` is the port number for InfoSphere Information Server. The default port number is 9080. For example, on a Microsoft® Windows® Server computer named `iisdocs2`, the Web address is in the following format: `http://iisdocs2:9080/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.nav.doc/dochome/iisinfsv_home.html`.

A subset of the information center is also available on the IBM Web site and periodically refreshed at `http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/index.jsp`.

Obtaining PDF and hardcopy documentation

- A subset of the PDF file books are available through the InfoSphere Information Server software installer and the distribution media. The other PDF file books are available online and can be accessed from this support document: `https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1`.
- You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at `http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss`.

Providing feedback about the documentation

You can send your comments about documentation in the following ways:

- Online reader comment form: www.ibm.com/software/data/rcf/
- E-mail: comments@us.ibm.com

Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible. The installation program installs the following product modules and components:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- IBM InfoSphere DataStage
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere QualityStage

For information about the accessibility status of IBM products, see the IBM product accessibility information at http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Notices and trademarks

This information was developed for products and services offered in the U.S.A.

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACSLink, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACSLink licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.

Index

A

analyzing imports 41
attribute 6

B

best practices for importing metadata 36
BI assets 14
 delete rules 26
 deleting 61
 finding in the metadata repository 58
 searching the metadata repository 58
 viewing details 60
BI models 14
BI reports 14
bridge parameters
 Cognos Content Manager bridge 74
bridges
 definition of 71
 functionality 71
 IBM InfoSphere Data Architect 81
 requirements 72
 supported 72
 Unicode 71
bridges, running imports and exports 73
business intelligence assets 14

C

candidate key 2
Cognos Content Manager bridge 74
command line interface
 escape characters 57
 InfoSphere Metadata Asset Manager
 command line 48, 57
 invalid characters 57
 running 48
common metadata 1
Common Metadata Administrator 32, 34
common metadata administrator role 31
common metadata importer role 31
common metadata user role 31
comparing staged imports 44
contract libraries 18
create import area from staged
 import 45
CSV file format for User Information
 MetaBroker 83
cubes 14
custom attributes 18
customer support
 contacting 97

D

data connections 18
data file field 2
data file structure 2
data item definition 2

DBM File MetaBroker 79
delete rules for metadata assets 26
design stored procedure 11
design table 11
disconnected assets 22, 25
 deleting 62
disconnected link points 69
duplicate assets 22, 24, 34
 merging 67
duplicates 22
 deleting from the metadata
 repository 63
 finding in the metadata repository 58
 merging 63

E

entity 6
entity attribute 6
entity generalization hierarchy 6
exporting metadata
 Import Export Manager 84
express import 36, 37

F

file format for User Information
 MetaBroker 83
foreign key 2

H

host computer 2

I

IBM InfoSphere Data Architect
 MetaBroker 81
identity 22
identity parameters 39
implementation relationship
 removing 65
 setting 65
implemented by 65
implemented data resources 2
 delete rules 26
 deleting 61
 finding in the metadata repository 58
 searching the metadata repository 58
 viewing details 60
implementedBy relationship 19
import area
 import parameters file 55
Import Export Manager
 command-line interface 87
Import Export Manager, exports 84
Import Export Manager, imports 85
import parameters file 55
import settings 34

ImportExportManager
 command-line import parameters 90
 IS parameter XML file 93
 validating the Parameter XML file 92
ImportExportManager command 89
importing
 DBM File MetaBroker 79
importing and exporting
 IBM InfoSphere Data Architect
 MetaBroker 81
 user information 82
 User Information MetaBroker 82
importing metadata 37
 best practices 36
 Import Export Manager 85
 import parameters file 55
 InfoSphere DataStage and
 QualityStage Designer 94
 ODBC 3.0 MetaBroker 82
InfoSphere DataStage and QualityStage
 Designer 94
InfoSphere Metadata Asset Manager
 administering 32
 allowing duplicate assets in
 imports 34
 analyzing imports 41
 comparing staged imports 44
 first steps in using the product 33
 importing assets 37
 metadata interchange server 34
 overview 29
 previewing the result of sharing an
 import 46
 reimporting metadata 42
 roles 31
 sharing imports to repository 47
 staged imports 41
 topology diagram 32
IS parameter XML
 command-line validation option 92
IS parameter XML file
 format 93

L

legal notices 103
logical data model 6
logical data model assets
 delete rules 26
 deleting 61
 finding in the metadata repository 58
 searching the metadata repository 58
 viewing details 60
logical domain 6
logical entity 6
logical model 6
logical models 81
logical relationship 6

M

- managed import 36, 37
- merge rules 68
- MetaBroker technology
 - definition of 71
 - functionality 71
- MetaBrokers
 - ODBC 3.0 MetaBroker 82
 - User Information 82
- metadata
 - importing from the command line 87, 89, 90
- metadata assets
 - delete rules 26
 - deleting 61
 - finding 58
 - managing 57
 - merging 68
 - viewing details 60
- metadata interchange agent 34
- metadata interchange server 34
- metadata repository
 - deleting disconnected assets 62
 - deleting duplicates 63
 - deleting link points 69
 - deleting metadata 61
 - finding assets 58
 - searching for assets 58
 - viewing asset details 60
- metadata repository, exporting from 84
- metadata repository, importing into 85

O

- ODBC 3.0 MetaBroker 82
- orphans 22, 25
- overview
 - InfoSphere Metadata Asset Manager 29

P

- parameter XML
 - command-line validation option 92
- physical data model assets 11
 - finding in the metadata repository 58
 - searching the metadata repository 58
- physical data resources 2
- physical domain 11
- physical models 81
- product accessibility
 - accessibility 101
- product documentation
 - accessing 99

R

- reimporting metadata 42
- relationship end 6
- roles
 - InfoSphere Metadata Asset Manager 31

S

- shared metadata 1
- sharing imports 47
- software services
 - contacting 97
- staged imports 41
- stored procedure 2
- subject area 6
- support
 - customer 97
- synonyms 81

T

- topology diagram for InfoSphere
 - Metadata Asset Manager 32
- trademarks
 - list of 103

U

- User Information MetaBroker 82
- users 83

V

- validation constraint 6



Printed in USA

SC19-3604-05



Spine information:

IBM InfoSphere Information Server

Version 8 Release 7

Guide to Managing Common Metadata

