

IBM Information Server version 11.3

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

The primary objective of this presentation is to help you understand the best practices to be followed for obtaining an optimal performance while transferring metadata assets between InfoSphere® Information Servers.

This presentation will help you better understand some implementation details of istool commands that will help you use the best practices for your metadata scenarios.

It will also help you understand the best practices in different scenarios. For example, how to transfer related assets, how to transfer large metadata assets and how to transfer a specific asset, like DataStage® jobs or Information Analyzer projects.

Table of contents

- Overview of InfoSphere Information Server and asset interchange
- Structure of metadata archive (.ISX) file
- Relationships between XMI files
- Impact of order of imports on performance
- Transferring related assets
- Significance of granularity of common metadata assets
- How to group assets into different metadata archive (.ISX) files

This presentation gives an introduction to InfoSphere Information Server and introduces the istool commands that provide the asset interchange capability. This presentation provides the basic education on the structure of the metadata archive and how the relationships between assets are stored in the metadata archive. It describes how the relationships between assets are reconciled when the metadata archive is imported into a target Information Server. This education will help you in understanding the best practices and use the best practices properly for your metadata assets.

This presentation also provides the reasoning behind the suggested order for importing different asset types by taking an example of Information Analyzer and database assets. It also provides the best order to be used while importing different types of assets.

This presentation describes the available options that you have while transferring related assets and it describes the pros and cons with these approaches and suggests the right approach to be used in different scenarios.

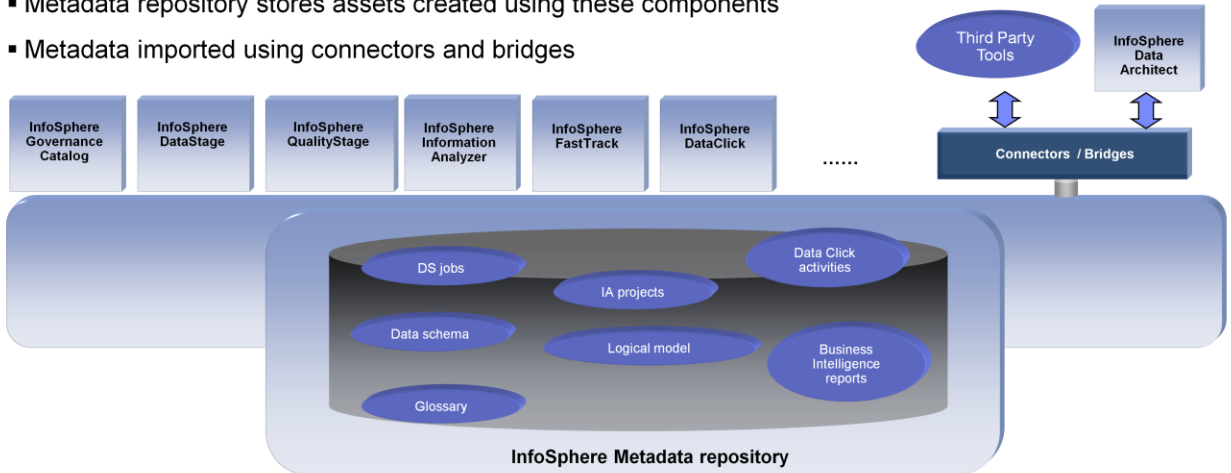
The istool command provides multiple options for exporting common metadata assets at different levels of granularity. It describes the right level of granularity to be used for obtaining the optimal performance and scalability in different scenarios.

Finally, the presentation discusses the standard approaches that are used while grouping metadata assets into multiple metadata archive files and the best practices to be used in different metadata scenarios targeted towards easier management of the

archive files.

Overview – InfoSphere Information Server

- IBM InfoSphere Information Server is a suite of components that is used to understand, cleanse, transform, and deliver information
- Metadata repository stores assets created using these components
- Metadata imported using connectors and bridges



3

Best practices for asset interchange in InfoSphere Information Server 11.3

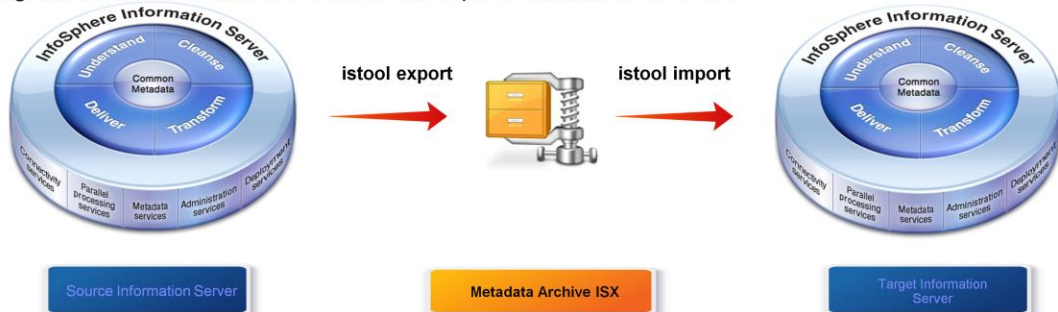
© 2014 IBM Corporation

IBM InfoSphere Information Server is a suite of components, including DataStage, Quality Stage, Information analyzer, InfoSphere Governance Catalog, Fast Track and Data Click that together provide a single, unified platform that enables companies to understand, cleanse, transform, and deliver information.

Each of the components generates metadata assets, which are stored in a shared metadata repository. Examples of these metadata assets include DataStage jobs, Information Analyzer projects, and Business Glossary categories and terms. Some metadata assets like Data schemas, Physical data model, and Business intelligence reports are imported from third-party tools or data sources like DB2®, Oracle, Cognos®, and Microstrategy and are shared across suite components.

Overview – Asset interchange

- Asset interchange
 - Allows users to move metadata assets between different installations of InfoSphere Information Server
 - Use istool commands to export and import assets
- Scenarios for using asset interchange
 - Move assets between Development, Test, and Production environments
 - Migrate assets to a newer version of InfoSphere Information Server



“Asset interchange” refers to the capability that moves metadata assets between metadata repositories of different installations of InfoSphere Information Server.

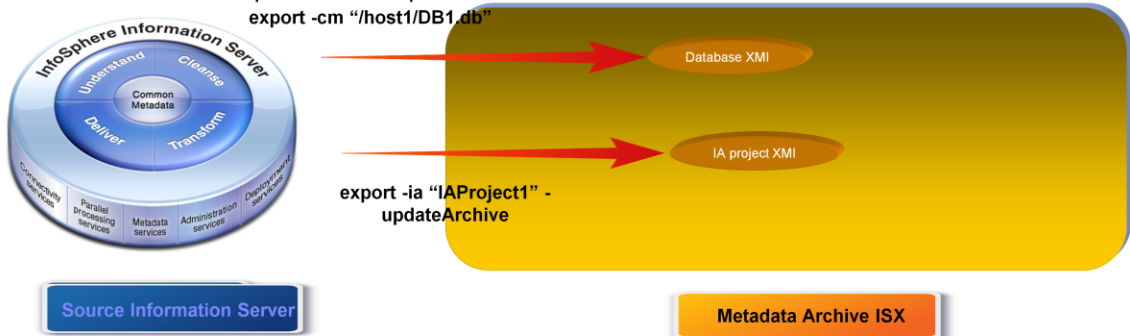
You use asset interchange capability in multiple scenarios. Some of the major scenarios are for moving the metadata assets between development, test, and production environments and migrating metadata assets to a later release of Information Server.

A set of commands called istool commands are provided for performing asset interchange. You move the assets by exporting metadata assets using the istool export command into metadata archive files. The metadata assets from metadata archives are then imported into a target Information Server installation using the istool import command.

Istool commands provide various parameters while exporting assets, such as parameters for selecting the metadata assets, such as Datastage jobs and Information Analyzer projects. It also provides various parameters while importing assets and replacing an asset if the asset already exists in the target repository.

Structure of metadata archive (ISX) file

- Assets are stored as XMI files in a metadata archive
- Values passed to parameters determine assets exported in XMI files
 - Example
 - commonmetadata (or -cm) and -informationanalyzer (or -ia) parameters determine which assets are exported into XMI files
- Each asset in XMI file is imported in a separate transaction



5

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

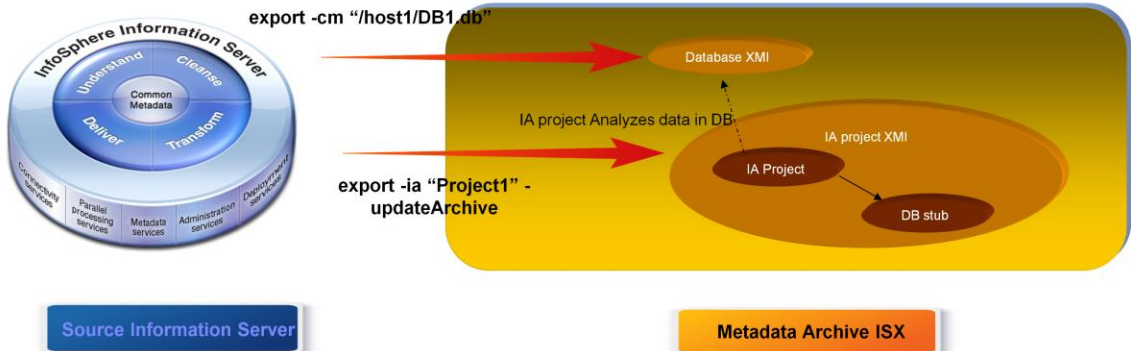
This presentation gives some insight into the structure of the metadata archive file, which is created by the istool export command. The metadata archive file is in a .zip file format that can be extracted by using a tool, such as Winzip. The metadata archive file contains several XMI files that store individual metadata assets. The component specific parameters in the istool export command determines what goes into these XMI files. For example, if you run the istool export command by giving "/host1/DB1.db" for -commonmetadata or -cm parameter, then an XMI file is created in the metadata archive that stores the metadata of the DB1 database in host1 server.

If you run the istool export command by giving "IAProject1" for -informationanalyzer or -ia parameter and give the same archive file with -updateArchive parameter, then another XMI file is created in the metadata archive that stores the Information Analyzer project called "IAProject1" .

When you run the istool import command to import the assets in the metadata archive ISX file, the assets in each of the XMI files are imported one at a time in separate transactions. This means if the import of one XMI file fails, it does not stop the import process. The import process will continue with the next XMI file in the metadata archive.

Relationships between XMI files

- Relationships between assets stored in different physical XMI files are represented by stubs (or placeholders)
- When metadata archive (.ISX) is imported onto a target InfoSphere Information Server computer, stub is used for establishing reference between assets



6

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

The diagram displayed on this slide shows how the relationships between the assets are stored in XMI files. Consider the scenario where IA project analyzes the data in a database and the IA project has reference to the database asset that indicates which tables or columns are analyzed. When IA project and database metadata are exported into separate physical XMI files, then how is the relationship from IA project to database asset stored? This is achieved by storing a stub or a placeholder for the database asset in the IA project XMI file and the IA project XMI stores the reference from IA project to the database stub.

If you import the metadata archive in a target Information Server, then the reference from IA project to the database stub is used for establishing a real reference from IA project to database asset in the target Information Server metadata repository.

Impact of order of imports on the performance – Bad order (1 of 2)

- Import an asset that has reference to other assets first
 - A stub (place holder) is created in target repository
- Later, import referenced asset
 - Reference between assets is created with help of stub
 - Stub is deleted from target repository

The order in which these dependent assets are imported into a target Information Server makes a difference to the performance of the import commands. This slide displays how the performance is impacted when the dependent assets are not imported in the right order.

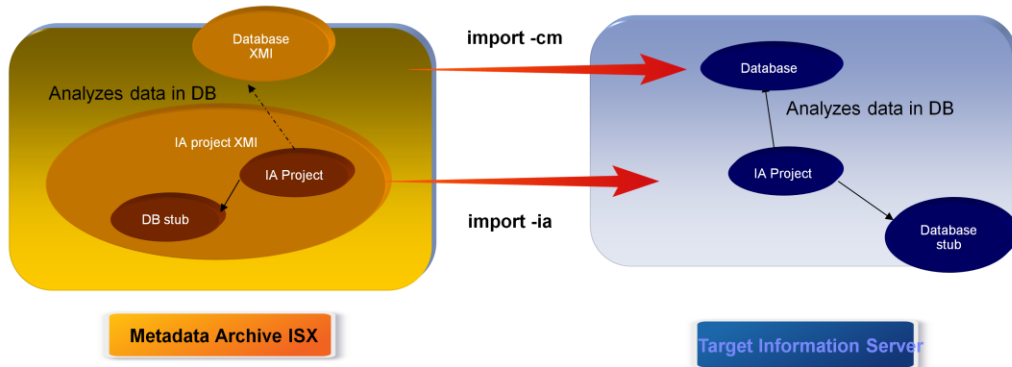
When you import the asset, such as an IA project that has reference to another asset, the stub, or place holder for the referenced asset is created in the target repository.

Later, when you import the referenced asset, such as a database, the stub in the repository is used for re-establishing the reference between these assets and the stub for the database asset is deleted from the target repository.

The creation and deletion of this temporary stub object in the target repository is a performance overhead.

Impact of order of imports on the performance – bad order (2 of 2)

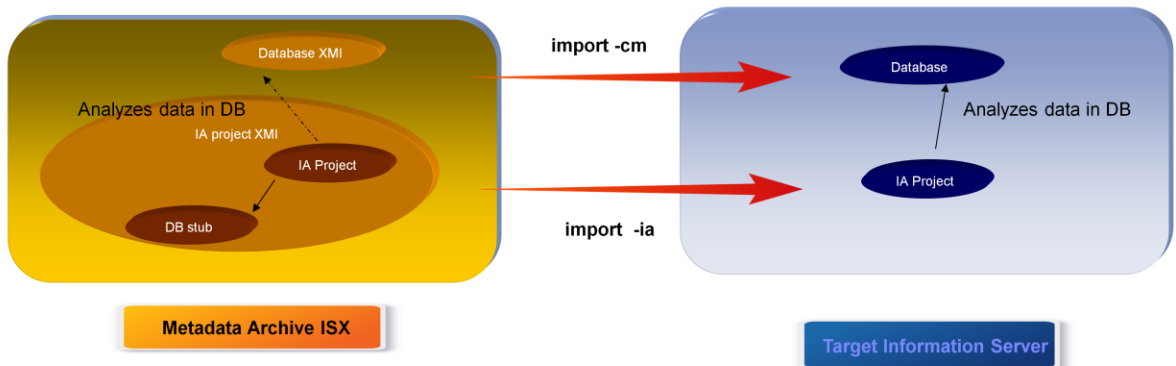
- Special cases with stubs that cause data loss in target if assets are not imported in right order
 - Example
 - Import database with contact relationship to use before importing users
 - Contact relationship is lost



There are some similar, special scenarios where stub objects cannot be created for some type of assets which causes a loss of data if the metadata assets are not imported in the correct order. For example, if you import a database with a contact assignment before importing the user, the contact assignment information is lost. You must first import users using the istool import -security command before importing any other assets that have contact assignment or steward relationship to the users.

Impact of order of imports on the performance – Preferred order

- For better performance
 - Import referenced asset first
 - Import assets that reference this asset



9

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

The overhead of the transient stub object in the target repository can be avoided by reversing the order of the import of these assets.

To get the optimal performance during import, import the referenced asset, such as the database asset, first.

Later, if you import the asset, such as an IA project that references the previously imported asset, the stub object is not created in the target repository.

Preferred order of import for the best performance (1 of 2)

- Import various asset types in this order:
 1. Users and groups using `import -security` parameter
 2. Common metadata using `import -commonmetadata` and `import -logicalmetadata` parameters
 - Database
 - Physical model
 - Business Intelligence models and reports
 - Logical data model
 3. InfoSphere DataStage and QualityStage assets using `import -datastage` and `import -qualitystage` parameters
 4. Reports using `import -report` parameter
 5. InfoSphere Information Analyzer assets using `import -informationanalyzer` parameter
 6. InfoSphere FastTrack assets using `import -fasttrack` parameter
 7. InfoSphere Information Governance Catalog extended data sources using `workbench extension source import` parameter.
 8. InfoSphere Information Governance Catalog extension mapping documents using `workbench extension mapping import` parameter

The next two slides provide the recommended order for importing different types of assets, such as security assets, common metadata assets, DataStage assets, Information Analyzer assets and FastTrack assets.

Preferred order of import for the best performance (2 of 2)

9. InfoSphere Information Governance Catalog business glossary assets using **glossary import** parameter

- In versions earlier than 11.3, these assets are created in InfoSphere Business Glossary

10. Custom attributes for InfoSphere Information Governance Catalog extended data sources and extension mappings

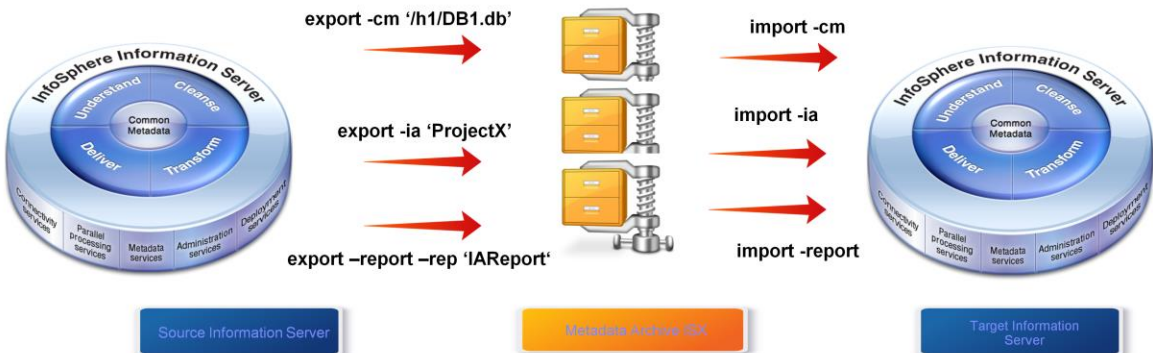
- In versions earlier than 11.3, these assets are created in InfoSphere Metadata Workbench

- Asset types not listed can be imported in any order

Asset types that are not listed on these two slides can be imported in any order without effecting the performance.

Transferring related assets (1 of 2)

- Option 1 (recommended)
 - Run multiple export commands for exporting each type of asset
 - Know related assets
 - Provide correct export sub parameters when you select each type of asset
 - Gives better control of assets being transferred
 - Eliminates possibility of transferring duplicate assets



12

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

Next, this presentation discusses transferring related assets. There are two options for transferring related assets.

The first option that is described on this slide is the recommended approach when exporting all the related assets. In this approach, you run multiple istool export commands for each type of asset with the corresponding parameter. For example, for transferring an IA project, transfer the related database assets and report assets as well. In this example, run istool export -informationanalyzer or -ia command for exporting the selected IA projects. Run istool export -commonmetadata or -cm command by giving sub parameters for selecting the database that is used by the Information Analyzer projects, and run istool export -report command by giving sub parameters for selecting the reports generated by IA project.

You can choose to export all of these assets into one metadata archive file or into separate archive files. You need to run multiple import commands for importing each of these asset types into a target Information Server. If you want to use this approach, you should know the related assets and you need to format the export command with the correct syntax for selecting the related assets.

This option is recommended because it gives you better control over what is being transferred and there is no possibility of transferring the same asset multiple times.

Transfer related assets (2 of 2)

- Option 2
 - Use sub parameters to let Asset Interchange determine related assets
 - Example: `-includeCommonMetadata`, `-includeReports`, `-includeDependent`
 - This option transfers only specific related assets
 - Example: IA project transferred with `-includeCM`
 - Only tables analyzed by IA project are transferred; causes partial schema on target computer
 - There is possibility of transferring same assets multiple times
 - Example: Transfer two IA projects that analyze same table, then tables transferred multiple times



13

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

The second option allows you to use sub parameters to let Asset Interchange determine the related assets. If you do not know the related assets, then you can use sub parameters like `"-includeDependents"`, `"-includeCommonMetadata"`, `"-includeReports"` and so on, for instructing the `istool` command to identify and export the related assets.

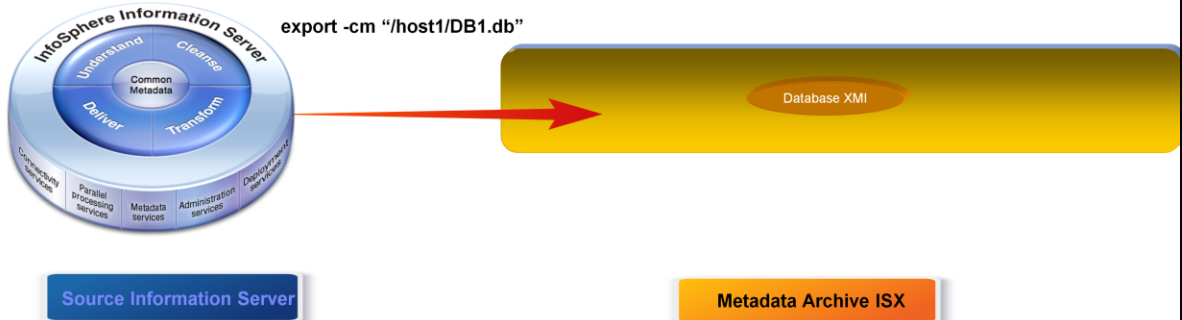
In the example from the previous slide, you can use `"-includeCommonMetadata"` and `"-includeReports"` sub parameters of `istool export -InformationAnalyzer` parameter to let the system determine the related common metadata and reports assets for the exported IA project and export these assets into the metadata archive. Here, the IA project and all the related assets are exported to a single metadata archive.

This approach is recommended only if you do not know the related assets. The `"-include"` related assets sub parameter exports only the specific assets that are used by the exported assets, which might not be the preferred behavior. In this example, the export command with `"-InformationAnalyzer"` parameter and `"-includeCommonMetadata"` sub parameter, exports only the tables analyzed by the Information Analyzer project, which results in a partial schema created in the target Information Server. You might prefer to transfer the complete schema instead of only transferring some of the tables in the schema.

There is also a possibility of transferring the same related asset multiple times. This occurs if you transfer two Information Analyzer projects that analyze the same database table, then the database table is transferred multiple times, which impacts the performance of import commands.

Significance of granularity of common metadata assets (1 of 2)

- Multiple choices of granularity provided for exporting common metadata
 - Example
 - Export entire database asset into one XMI file using `-cm "/host1/DB1.db"`
 - Provides better performance
 - Tables, columns and stored procedures stored in same XMI
 - Stubs not created
 - Not scalable
 - Very large database import in one transaction fails



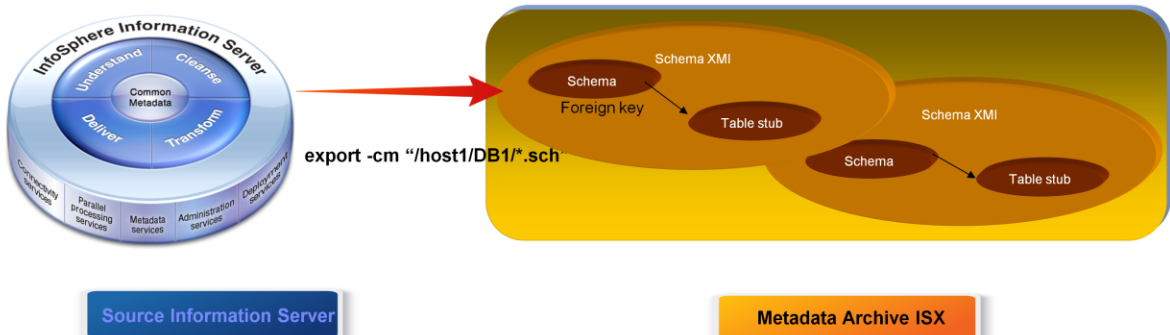
You have multiple choices of granularity for exporting common metadata assets like databases, physical models, data files, logical models, and business intelligence models and reports.

For example, when transferring database assets, you can give an identity string of the database name, for example, `-commonmetadata "/host1/DB1.db"`, so that the complete database assets that include all schemas, tables, columns, stored procedures, and so on, are exported into one XMI file. This gives better performance because the XMI contains all of the assets and there is no need for creating stubs or placeholders for the relationships between the contained assets, like foreign key relationships between database tables.

However, if the database is very large, then importing the database with all the tables and stored procedures in one transaction fails. While this approach gives better performance, it is not scalable.

Significance of granularity of common metadata assets (2 of 2)

- To improve scalability
 - Reduce granularity of asset stored in XMI file to Database schema by using “/host1/DB1/*.sch” identity string
 - Overall performance reduced
 - Stubs created for foreign key references to tables that exist in different schemas
- For large schemas
 - Reduce granularity of asset in XMI to database table by using “/host1/DB1/*/*.tbl”



15

Best practices for asset interchange in InfoSphere Information Server 11.3

© 2014 IBM Corporation

To improve the scalability while transferring large assets, you can reduce the granularity of the asset that is exported into XMI file and import it into the target Information Server as a separate transaction. In the example from the previous slide, you can reduce the granularity of the asset in the XMI file to the database schemas by giving the “/host1/DB1/*.sch” identity string. This however, might cause the creation of stub objects for the referenced tables that exist in a different schema, thus, impacting the overall performance of import commands for transferring all the schemas in a database.

If the database schema is also too large to be imported in one transaction, then you can further reduce the granularity of the asset in XMI file to database tables by using the “/host1/DB1/*/*.tbl” identity string. This further reduces the overall import performance because stubs are created for all the foreign key relationships between the tables.

The recommended best practice is to export higher granular assets when possible. If the higher granular assets are too large to be imported in one transaction, then reduce the granularity of the asset in XMI file by using the identity string of lower granular assets.

Group assets into different metadata archive files (1 of 2)

- Options for grouping assets into metadata archive files
 - Option 1
 - Export all assets from a source Information Server into one metadata archive
 - Option 2
 - Export all related assets for an IA project or DS job or a FT project into one metadata archive
 - Possibility of transferring an asset multiple times
- Metadata archive with different type of assets contained in it
 - Always run multiple import commands for each type of asset in order suggested by user guide

The `istool` command provides various choices on how you can group your assets into metadata archive files while exporting assets from a source Information Server.

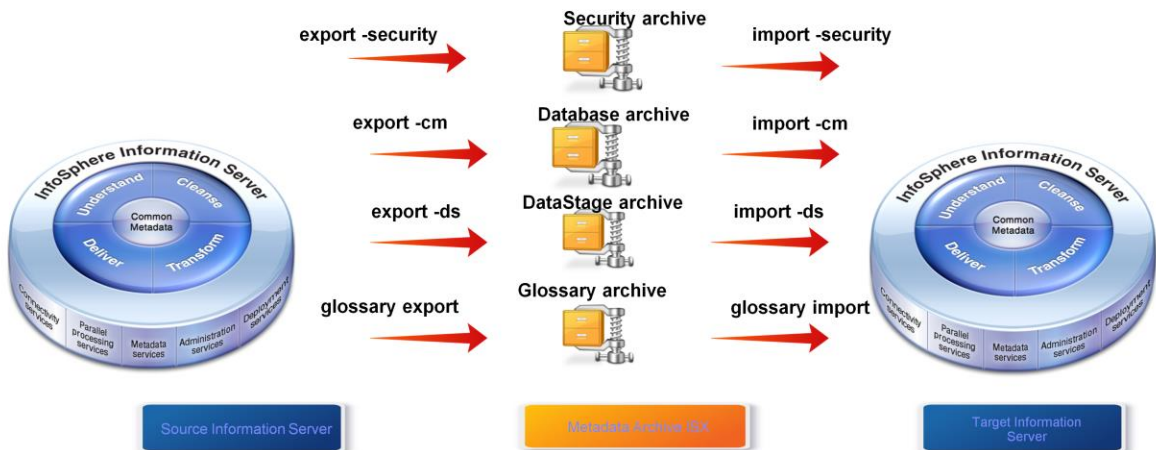
One simple approach is to export all metadata assets into one metadata archive. You can use this approach if you have a limited number of assets.

Another approach is to export all the related assets into one metadata archive file. For example, export an Information Analyzer project and all the related assets in one metadata archive and export a DataStage job and all related assets into one archive and export a Fast Track project and all its related assets into one metadata archive. This approach has potential for transferring the same asset multiple times. For example, if a database asset is used by both an Information Analyzer and DataStage job, then the database assets are included in two metadata archive files.

If you have a metadata archive with different types of assets that are contained in it, then always run multiple import commands for importing each type of asset in the order suggested by the User's Guide.

Group assets into different metadata archive files(2 of 2)

- Option 3 (recommended)
 - Export each type of metadata asset into a separate metadata archive



The recommended approach is to export each type of metadata asset into one metadata archive file. For example, create one metadata archive file with all security assets, one metadata archive with all database assets, one metadata archive with all DataStage assets, one metadata archive with all IA projects, and so on. This is the recommended approach because it is easy to manage metadata archive files and there is no scope for duplicate transfer of metadata assets.

References

- Migrating to IBM InfoSphere Information Server, Version 11.3:
 - http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.productization.iisinfsv.migrate.doc/topics/top_of_map.html
- Managing assets by using istool command line:
 - http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.iisinfsv.assetint.nav.doc/containers/istool_container_topic.html

This presentation suggested the best practices that help you in writing the right set of istool commands for exporting and importing metadata assets most efficiently.

The best practice is to use the recommended options for various aspects of asset interchange where ever possible. If you have a special scenario where the recommended option does not fit, then you can use an alternate option.

Refer to IBM Knowledge Center for a complete reference of the procedures for migrating Information Server to version 11.3 and for managing assets using istool commands.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Cognos, DataStage, DB2, DB, IA, and InfoSphere are trademarks or registered 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 other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2014. All rights reserved.