IBM® InfoSphere™ Information Server

Best Practices

Asset Interchange in IBM InfoSphere Information Server

Cassio dos Santos
Software Architect, InfoSphere Information Server

Executive Summary

IBM InfoSphere Information Server is a suite of components that together provide a single unified platform that enables companies to understand, cleanse, transform, and deliver information. Each of the components has a set of assets that are stored in the shared metadata repository.

Different kinds of metadata assets are created by the various InfoSphere Information Server components and shared through the metadata repository. Examples include IBM InfoSphere DataStage® jobs, IBM InfoSphere Information Analyzer projects, and IBM InfoSphere Business Glossary categories. Some metadata assets are imported from third-party tools and shared across multiple tools, such as physical data resources (for example, tables) or business intelligence reports..

"Asset interchange" refers to the functionality that moves metadata assets between different metadata repositories of InfoSphere Information Server, across environments such as development, test, and production.

The objective of this document is to communicate best practices for a more effective use of the InfoSphere Information Server asset interchange tools. It is assumed that the reader is familiar with the tools that are used to perform asset export and import, in particular the `istool` command-line tool and its export and import commands and parameters that are available for the various components that support asset interchange.

For more information on istool commands for asset interchange, see the public information center documentation for InfoSphere Information Server, Version 8.5: http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r5/com.ibm.swg.im.iis.iisinfsv.assetint.nav.doc/containers/istool_container_topic.html, or see the *IBM InfoSphere Information Server Administration Guide.*

Understand dependencies between assets

You can use the asset interchange tools to move individual assets or large groups of assets, depending on your needs. In any case, it is important to determine which assets the assets that you need to transfer depend on, because you might also need to transfer these dependent assets.

Different options provided by the different components allow the transfer of a dependent asset either at the same time or independently of the assets that depend on it. When a dependent asset is not transferred at the same time as an asset that depends on it, a link to it may be transferred so that the relationship can be reestablished in the target system where both assets are later imported.

An asset can depend on other assets from the same component or other components. For instance, an InfoSphere DataStage parallel job can depend on other InfoSphere DataStage assets in the same project, such as routines or shared containers, and it can also depend on assets from another component, such as a database table from the common metadata component.

To determine the right subset of interdependent assets to transfer, it is important to understand the dependencies among different kinds of assets. Sometimes the order in which interdependent assets are transferred can also help minimize the required total time to perform the transfer.

Each component exposes a number of options that can be used to include assets from other components at export time. Those options are summarized in the table below.

Component	Export option to include dependent/referred asset	Export option to include link to dependent asset
InfoSphere DataStage	`-includedependent`	Links to dependent assets are automatically exported

InfoSphere Information Analyzer	`-includeCommonMetadata`
	
	`-includeReports`
		Links to dependent assets are automatically exported

IBM InfoSphere FastTrack	`-includeReports`
	`-includeCommonMetadata`
	`-includeDataStageAssets`
	`-includeDependent`
		Links to dependent assets are automatically exported

InfoSphere Business Glossary	Only links to assigned assets can be exported along with business terms, not the assigned assets themselves	`-includeassignedassets` `-includestewardship`

Note that when you export InfoSphere Business Glossary assets, the related assigned assets and stewards are never exported along with the associated business terms and categories, but the user has the option to export the links to them. With the other tools, the links to related and dependent assets are by default always exported and the user has the option to export the actual assets instead of only the links to them.

Some tools, including InfoSphere DataStage, offer a general `-includedependent` option to include all the dependencies for the assets that are being exported (other assets from the same tool or assets that are owned by other components). Other tools offer a general option (InfoSphere FastTrack offers `-includeDependent`) as well as component specific options (InfoSphere FastTrack offers `-includeDataStageAssets`, `-includeReports`, and `-includeCommonMetadata`).

When dependent assets are not explicitly included in the export, links to them are exported so that if they exist in the target system the relationship between the assets can be reestablished. We say that the link is resolved at import time in that case. If a dependent or related asset is not found in the target system at import time, the link information is stored so that the relationship can be reestablished whenever the referred asset is imported at a later time. The links can be then resolved (unless the stored links are eventually purged from the system as part of a maintenance clean-up procedure).

Import dependencies first

If a set of assets S1 depends on another set of assets S2, importing S2 prior to importing S1 can help to reduce the overall import time. This is so because the cost of storing the unresolved links is avoided.

The main cases that should be taken into account include:

- Import the database tables before importing the InfoSphere DataStage jobs that depend on them.
- Import the corresponding users and groups before importing the InfoSphere Business Glossary assets that are exported with the `-includestewardship` option.

- Import the corresponding users and groups before importing the common metadata assets that are exported with the `-includeContactAssignment` option.

- Import users and groups before transferring InfoSphere Information Analyzer or InfoSphere FastTrack projects, if you are transferring project roles.

- Import physical data resources (common metadata assets) ***and verify the associated data connection*** before transferring the InfoSphere Information Analyzer projects that depend on them.

Export assets from different components separately

Assets from different components can be exported together to a given archive in different ways.

First, you can use a single export command to export assets from different components.

For instance: `istool export … -datastage '…' -fasttrack '…' -ia '…'`

Second, when you export assets from a given component, you can use specific export options to include assets from other components.

For instance:
```
istool export …-fasttrack 'FTProject1.ftp
                -includeReports
                -includeCommonMetadata
                -includeDataStageAssets'
```

Finally, you can add assets to an existing archive by using the `-updatearchive` option.

Whether to group assets from multiple components in a single archive file or to use separate archive files to group assets per component depends on the case at hand. In general exporting assets from different components separately and keeping them in different archive files gives more flexibility and can produce better overall performance for both export and import operations.

First, it allows assets from different components to be archived and transferred independently from dependencies to or from other assets across different systems.

Second, it allows assets to be more easily imported in an optimal order based on inter-asset dependencies.

Finally, it allows archive sizes and contents to be better controlled.

Using separate archives for different major units of work for a given component is a variation of the same best practice. For instance, more flexibility is achieved by using a

separate archive per project (InfoSphere DataStage, InfoSphere FastTrack, InfoSphere Information Analyzer), per database (common metadata) or per root category (InfoSphere Business Glossary).

The above guidelines may be more or less relevant depending on the volume of data that is being transferred and on or the context of the data transfer (for example, a regularly scheduled dev-2-prod or a one-of-a-kind dev-2-dev data transfer), among other factors. For instance, the reports associated with a given project can be transferred along with the project by using the appropriate `-includeReports` options, or by using the export options specific to the Report component (`-reportName`, `-ownedByProduct`). Which alternative is the best depends on the case at hand.

When you export reports with InfoSphere Information Analyzer assets, the best option is to export each project and its associated reports into separate archives, so that the import can be better managed in case of conflicts.

Export sub-assets along with their parent asset

If only a small subset of the sub-assets of a given parent asset needs to be exported, then exporting the sub-assets only without the parent asset is a better option. For instance, if only a small subset of all the tables of a large schema is needed by a given DataStage project, you should export the tables separately from the project.

On the other hand, if you need to export all or most of the tables of a given schema (for example, /host1/db1/schema1/*.tbl), then exporting the entire schema and all its contained tables instead (for example, /host1/db1/schema1.sch) is more efficient.

This is so because you avoid the cost of exporting and storing the link between each table and the containing schema so that the association can be later reestablished at import time when you export the schema with all or most of its tables rather than just several of them. Note that the same savings are not obtained for links between assets across different components, because links are exported as separate entries in the same archive (for example, a job and a database table), while tables and their containing schemas are exported into the same file in a given archive.

Import assets from different components separately

This is the import counterpart of the “export assets from different components separately” best practice.

Each component provides different options to control how assets are to be merged with or to replace identical assets that are found in the target repository at import time. Because different components apply different rules and provide different options, it is easier to understand and control the overall final result of an import by isolating imports per component and using separate istool commands for each of them.

Another way to describe this best practice is "understand merge and replace actions for each type of asset." While for some components the `-replace` import option is truly optional, other components make it mandatory because they don't allow matching assets to be merged. Other components in turn offer options to resolve name conflicts at import time (replace/rename/ignore).

Also, the order in which dependent assets are imported can significantly affect performance. Even if a set of assets and all its dependents are exported into a single archive instead of separate archives per component as recommended above, for best import performance you should run separate commands to import each type of metadata that was exported to a single archive file. You should run the imports in the order that gives the best performance. Start by importing assets that are depended upon and then import the assets that depend on the assets already imported.

When, for instance, an InfoSphere FastTrack project is exported along with the common metadata, reports and InfoSphere DataStage jobs that it depends on, it is more efficient to first import the common metadata (by using the import `-cm` option), then the DS jobs (by using the import `-ds` option), then the report (by using the `-report` and `-replace` options), and finally the InfoSphere FastTrack project itself (by using the `-ft` option). Tech note 1446771 provides more details on how to optimize the export and import of large numbers of IBM InfoSphere FastTrack assets and their dependent assets.

Finally, you should import assets from different components separately, so that a different user having the required user role for each component can perform the corresponding import, and a system administrator is not required to import the entire set of interrelated assets in a single import execution.

Refer to the user documentation for a detailed account of the merge and replace actions for each component and type of asset.

Use the preview option

The `istool -preview` option can be used to preview either the export or import operation without actually executing the import and export commands. This option is supported for all the components except InfoSphere Business Glossary.

The preview option can be used as a query command which, given search input parameters in the form of asset identity strings that contain wildcard characters, can give the list of assets that match the search criteria.

For export, the preview option can be used to refine the input parameters to validate that all the assets that should be exported are exported, and that assets that are not needed are not exported, without incurring the cost of actually exporting them. It can also help to identify the dependent assets that could be exported using separate export commands to improve perfromance, when used in conjunction with dependency inclusion export options.

For import, the preview option can be used to ensure that undesired changes to the repository are not performed.

Script asset interchange

As considered above, breaking up the transfer of a large set of interrelated assets from multiple components into multiple steps can improve performance and reduce the complexity of maintaining the required dependencies and corresponding command-line options.

Using the `istool` command to export and/or import assets from multiple components with multiple dependencies across them may require careful planning and experimentation.

You can script a sequence of exports or imports either by using the `-script` option of the `istool` command to read multiple commands from a text file, or by issuing multiple `istool` commands from a shell script.

Avoid concurrent imports

While concurrent data transfers can be performed by running parallel `istool` commands, it is recommended that such commands be executed in a sequence, in particular the import commands, to avoid undesired contention (database locks, server memory) and consequent failures or inefficiencies (for example, deadlocks, excessive garbage collections). This is particularly true when dependencies exist across assets being imported through different processes running in parallel.

Clean-up unresolved links

As described above, when assets are exported along with links to dependent or related assets that don't exist in the target system the links are stored to allow the relationship to be reestablished when the target asset is later imported. Those stored links are invisible to the end user and are used only by the import routine to reconnect the assets when the target of the link is imported.

Unresolved links may accumulate over time and it may be desirable to perform some maintenance clean-up periodically to remove links that are no longer needed, usually based on obsolescence (for example, any unresolved link older than a given number of days can be considered obsolete and eligible for removal).

Besides reclaiming database space, cleaning up unresolved links can improve overall performance for asset imports.

A utility is provided to clean up unresolved links based on different criteria. For more details please refer to Tech note 7019784.

Use component-specific options and UIs

Some component specific export and/or import options are available and should be used whenever appropriate.

For example, when you import Business Glossary assets, you can use the option –mappingfile to map some attributes in the XMI file to other values. This can be useful, for instance, to map host names that are used in the identities of some physical data resources and to allow links from business terms to assigned assets to be resolved when there are some differences in the host names of the assigned resources between the source and the target systems.

As another example, when you export physical data resources to be used by other tools such as InfoSphere InformationAnalyzer, you should include the corresponding data connection assets by using the option –includeDataConnection. Test the connection in the target system before you importing the assets that depend on the physical data resources.

Some components also offer dedicated asset interchange user interfaces that can be easier to use than the istool command-line interface. For instance:

- InfoSphere Business Glossary Administrator tool for transfering InfoSphere Business Glossary assets. (When you import very large sets of glossary assets, use the asset interchange command-line instead to improve performance.)

- InfoSphere Information Server Manager for transfering InfoSphere DataStage and QualityStage assets.

  InfoSphere DataStage also offers asset interchange legacy tools that use the DSX file format to export and import assets in InfoSphere DataStage Designer and some command-line tools. Do not combine the DSX and ISX formats when you transfer interrelated data. The data might not reconcile consistently in the target repository.

Use istool console mode

When you use the command line interface run asset interchange, use the istool command prompt instead of running in the operating system command prompt. This avoid unexpected results that can occur due to the way some operating system command prompts interpret some special characters such as the tilde (~).

Tune the system configuration

Asset interchange involving very large volumes of data can be very resource intensive (memory, CPU) and may require special or advanced tuning of the system configuration to eliminate some performance bottlenecks and to scale appropriately (for example, server Java heap size, database transaction timeout, disk configuration, database configuration).

Summary of Best Practices

- Understand dependencies between assets.
- Import dependencies first.
- Export assets from different components separately.
- Export sub-assets along with their parent asset.
- Import assets from different components separately.
- Use the preview option.
- Script asset interchange.
- Avoid concurrent imports.
- Clean up unresolved links.
- Use component-specific options and UIs.
- Use istool console mode.
- Tune the system configuration.

Conclusion

The ideas discussed in this paper must be adapted to the case at hand. The occasional transfer of a small volume of assets might not require too much planning or consideration. If you use the best practices in this document for recurring or large asset transfers across different environments, you can realize significant cost savings and reduce the complexity of maintaining cross-component dependencies across systems.

Further reading

- Tech note 1446771:http://www.ibm.com/support/docview.wss?uid=swg21446771
- Tech note 7019784:http://www.ibm.com/support/docview.wss?uid=swg27019784
- IBM InfoSphere Information Server Information Center: **http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r5/topic/com.ibm.swg.im.iis.iisinfsv.assetint.doc/topics/assetinterchange.html**
- "Managing assets by using the command line" in *IBM InfoSphere Information Server Administration Guide*.

Contributors

The authors would like to recognize the following individuals for their feedback on this paper and their contributions to this topic:

[Walter H. Crockett]
[Information Developer, InfoSphere Information Server]

[Prasad S Madugundu]
[Software Designer, InfoSphere Information Server]

[Zhi Li]
[Software Developer, InfoSphere Information Server]

[Anu Tumkur]
[Software Developer, InfoSphere Information Server]

[Robin M. Noble]
[Software Developer, InfoSphere Information Server]

[Carlene P Nakagawa]
[Software Test Specialist, InfoSphere Information Server]

[Benny Halberstadt]

[Software Architect, InfoSphere Information Server]

Notices

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

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.

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.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

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.

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 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

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

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