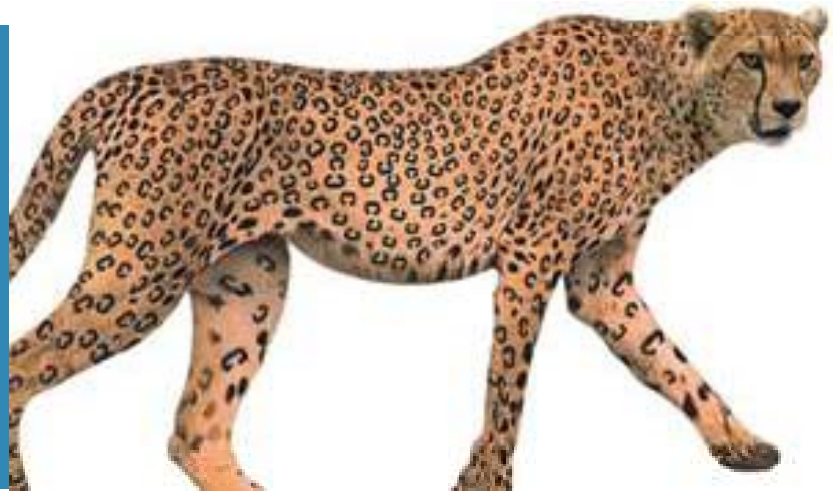
Cognizant | Academy
Passion for making a difference

Informatica 9x : Advanced Topics



Session 9: Advanced Topics

Academy

About the Author

Created By:	Cognizant Academy
Credential Information:	
Version and Date:	INFORMATICA9X/PPT/S9/0309/1.0

Cognizant Certified Official Curriculum

Cognizant | Academy
Passion for making a difference

Icons Used

Questions

Tools

Hands on Exercise

Coding Standards

Test Your Understanding

Reference

Try it Out

A Welcome Break

Contacts

Cognizant | Academy
Passion for making a difference

Advanced Topics : Overview

- **Introduction:**

This module gives a brief overview about the advanced concepts in Informatica

Advanced Topics : Objective

- **Objective:**

After completing this chapter, you will be able to

- Understand versioning
- Copy, import and export objects
- Create and use deployment group
- Work with debugger
- Understand data recovery
- Understand partitions
- Collect performance data
- Identify bottlenecks and apply various performance fine tuning techniques

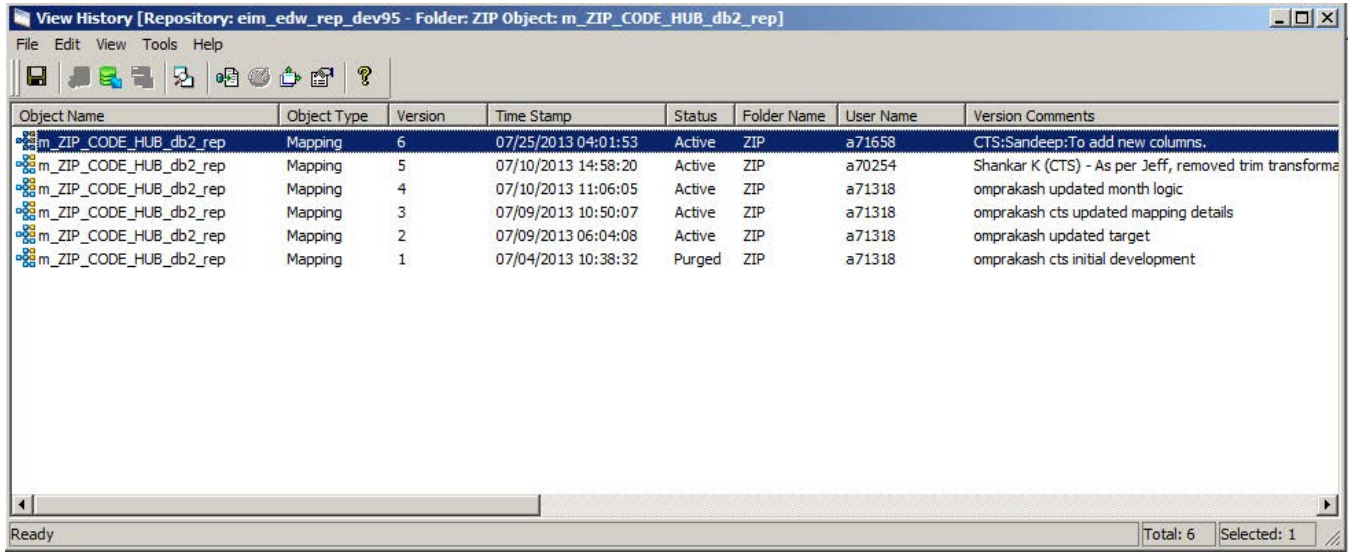
Versioning

- Versioning is a concept where you can configure the repository to store multiple versions of objects
- You can configure a repository for versioning
  - When it is created
  - When an existing repository can be upgraded to support versioned objects
- Versioning allows to
  - Store copies of previous versions of objects in development
  - Track changes to objects
  - Prepare them for deployment to a different environment
- Assigns multiple version numbers to versions of the same object
- Each time an object is checked in, the repository increments the version number by one and stores a new version of the object in the repository database

More on Versioning

- Tasks you can perform with versioning
  - » View object version properties
  - » View object history
  - » Track changes to an object
  - » Check the object version
  - » Delete or purge the object version
- By default, the navigator and workspace always display the latest version of an object
- You can view the version history of an object or create a query to search for previous versions of an object
- If you rename an object during development, different versions of the same object may have different names

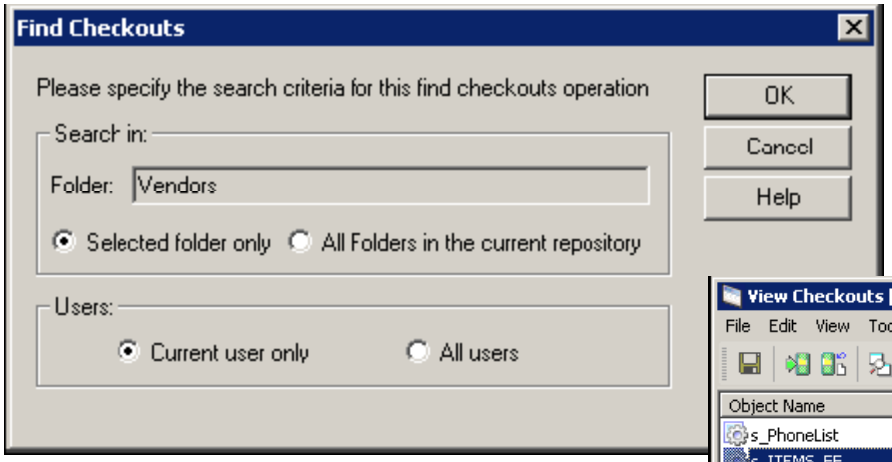
Viewing object history

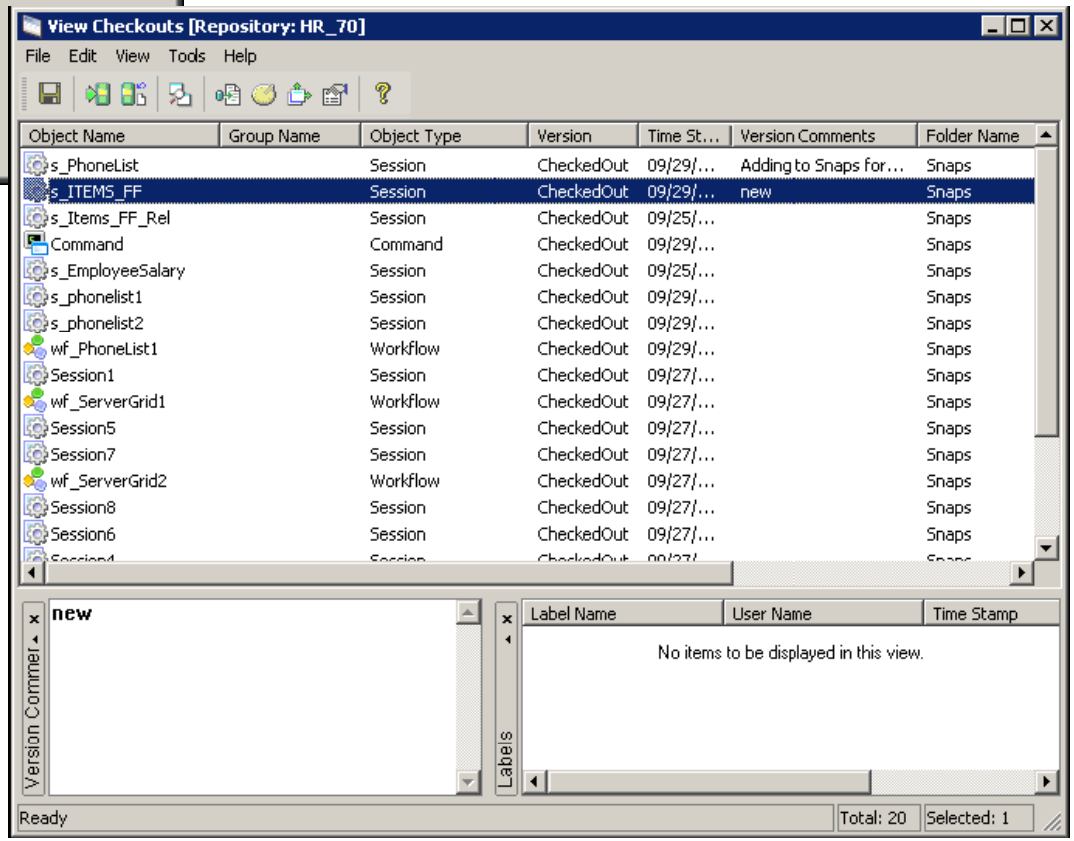


Checking Out an Object

- When you open an object in the workspace, the repository obtains a write-intent lock on the object, if no other user has checked it out
- No other repository users can edit the object while you have it checked out
- When you work with composite objects, the Repository Agent treats non-reusable objects as part of the parent object, so you cannot check in or out individual non-reusable objects
- If you disconnect from the repository without saving a checkout, the Repository Server releases the write-intent lock on the object and you lose the changes you made to the object
- The Repository Agent reverts back to last checked-in version of the object

Viewing Checked Out Objects

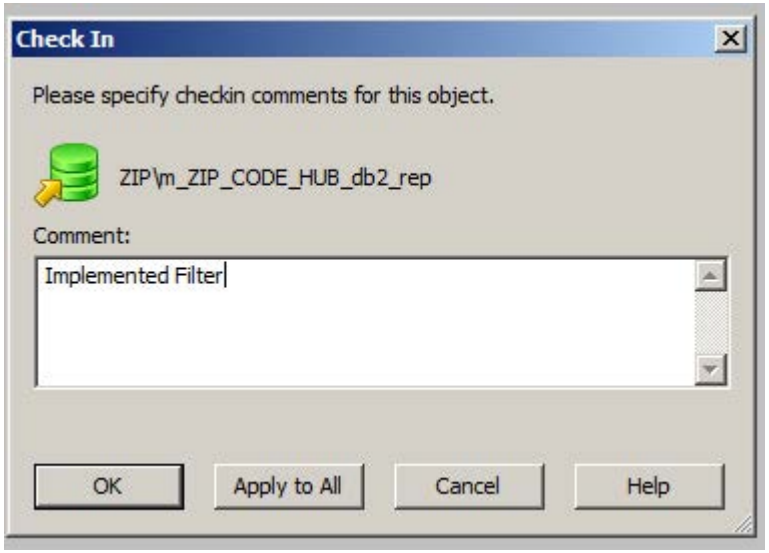




Checking in an object

- When you finish making changes to a checked out object, you can check it in and commit the changes to the repository
- While check in, the repository creates a new version of the object and assigns it a version number
- The repository increments the version number each time you check in an object
- When you work with composite objects, the Repository Agent treats non-reusable objects as part of the parent object, so that you cannot check in or out individual non-reusable objects
- The Repository Agent does not check in reusable objects when you check in a composite parent object, such as a mapping
- If you want to check in all objects used in a mapping, you must check in reusable objects separately

Checking In an Object



Copying Object

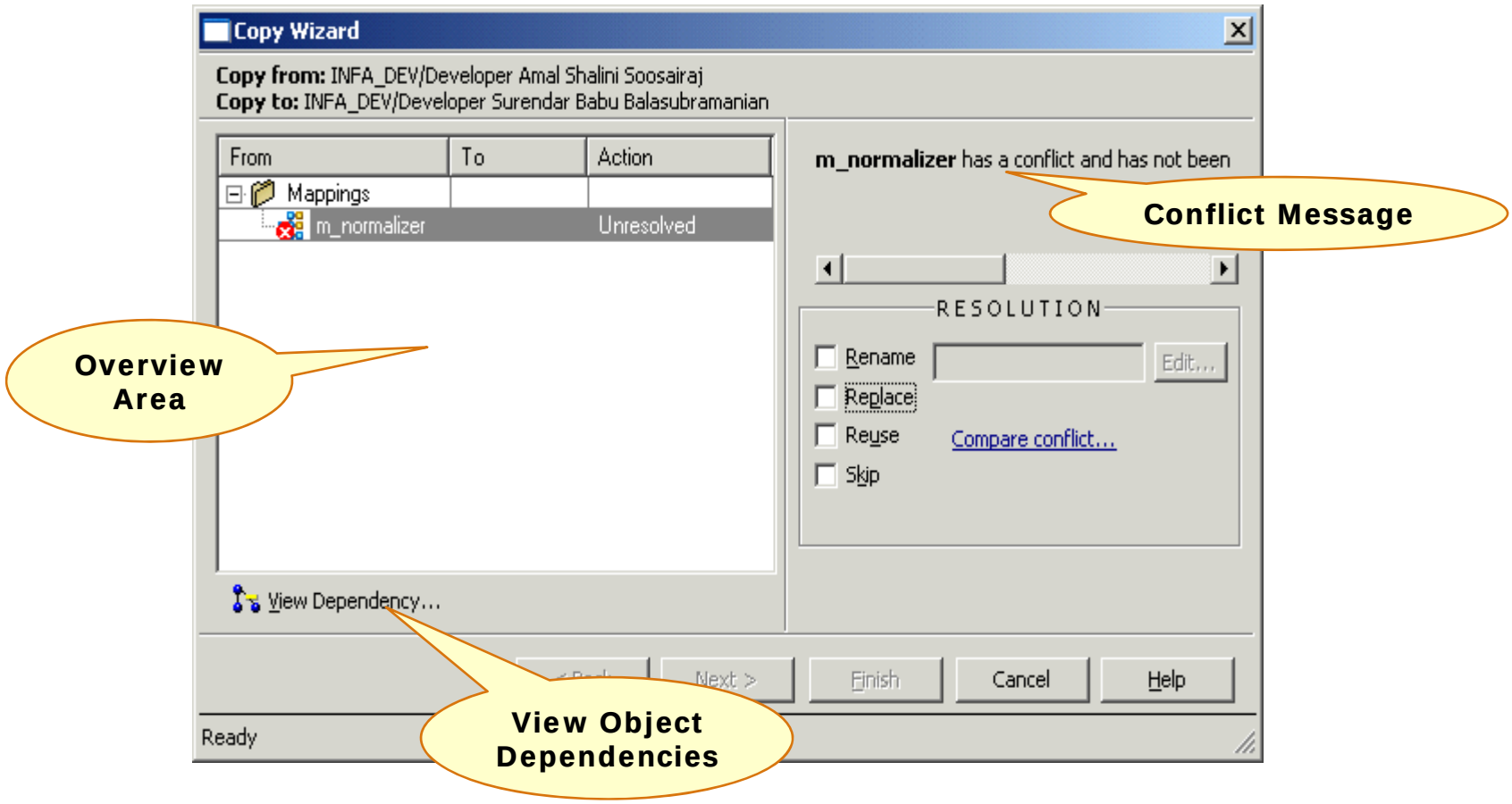
- You can copy the following repository objects
  - Workflows
  - Worklets
  - Tasks
  - Sessions
  - Mappings
  - Mapplets
  - Sources
  - Targets
  - Transformations
  - Segments of workflows or mappings

- You can copy objects
  - Within the same folder
  - To a different folder
  - To a different repository

Copying Object

- To copy an object to another folder, you must first open the target folder
- You cannot copy/paste objects from/into a shortcut to a mapping or mapplet
- You can paste the objects multiple times in any mapping or mapplet in the same folder
- The Workflow Manager, Designer, and Repository Manager provide a Copy Wizard to copy repository objects

Copying Object



Importing and Exporting Objects

- Repository objects can be exported to an XML file and then imported from the XML file
- You can use any one of the following client applications to export and import repository objects
  - » Repository Manager
  - » Designer
  - » Workflow Manager
  - » Pmrep (command line)
- Exporting and importing an object is similar to copying an object from one folder or repository to another
- When you export an object from one repository and import the object into another repository, you do not need to be connected to both repositories

Importing and Exporting Objects

- You can import objects from a valid XML file. The XML file must comply with powrmart.dtd
- You can import objects that you exported from the same repository or a different repository
- When you import an object, the PowerCenter Client performs the following tasks
  - » Validates the XML file against powrmart.dtd
  - » Parses the XML file
  - » Validates the objects in the XML file
  - » Creates the objects in the repository
- When you import using *pmrep*, you use a control file to specify the same import options in the Import Wizard

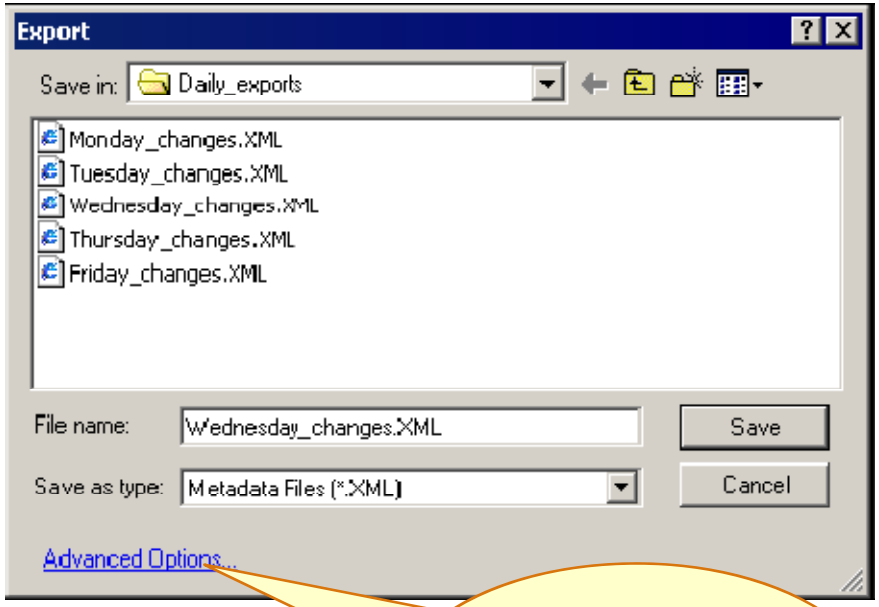
Importing and Exporting Objects

- You can export and import repository objects to accomplish the following tasks
  - » Deploy metadata into production
  - » Archive metadata
  - » Share metadata
  - » Search and replace property names in an entire repository object
  - » Copy metadata between repositories
  - » Create mappings

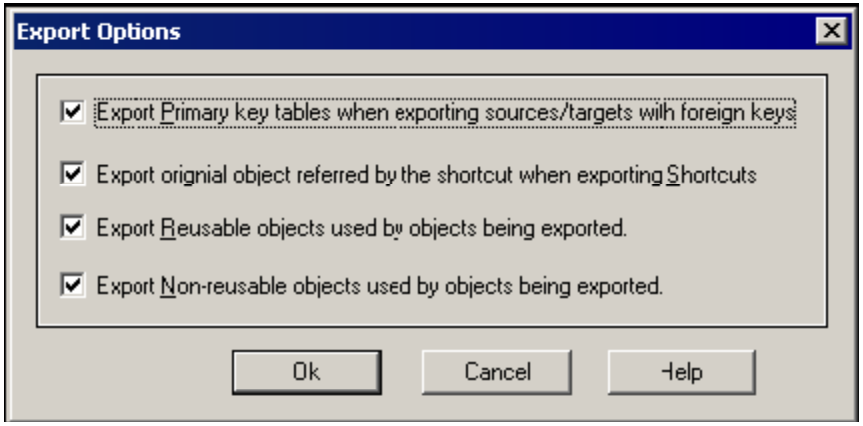
Importing and Exporting Objects

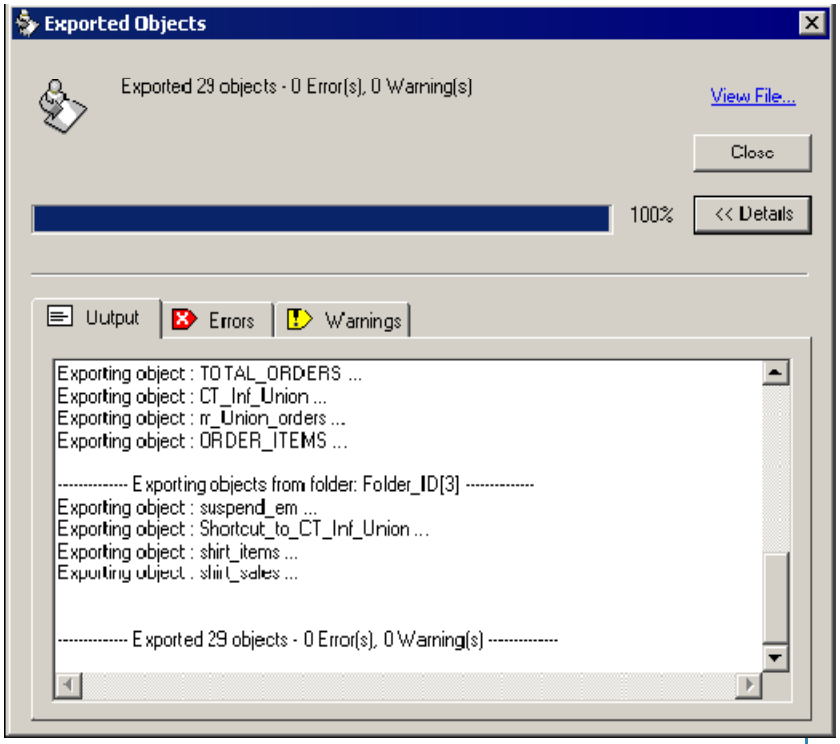
- You can choose to export and import the following types of objects
  - » Multiple object types
  - » Multiple objects
  - » Objects from multiple folders
  - » Dependent objects

Exporting Objects



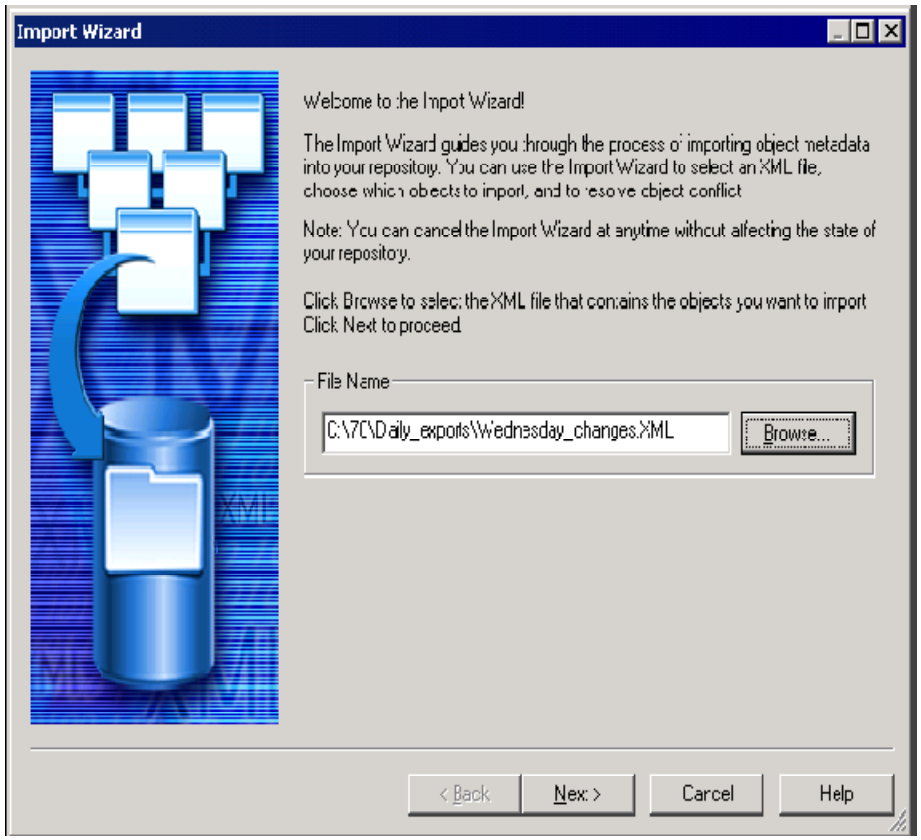
Choose which dependent objects to export.





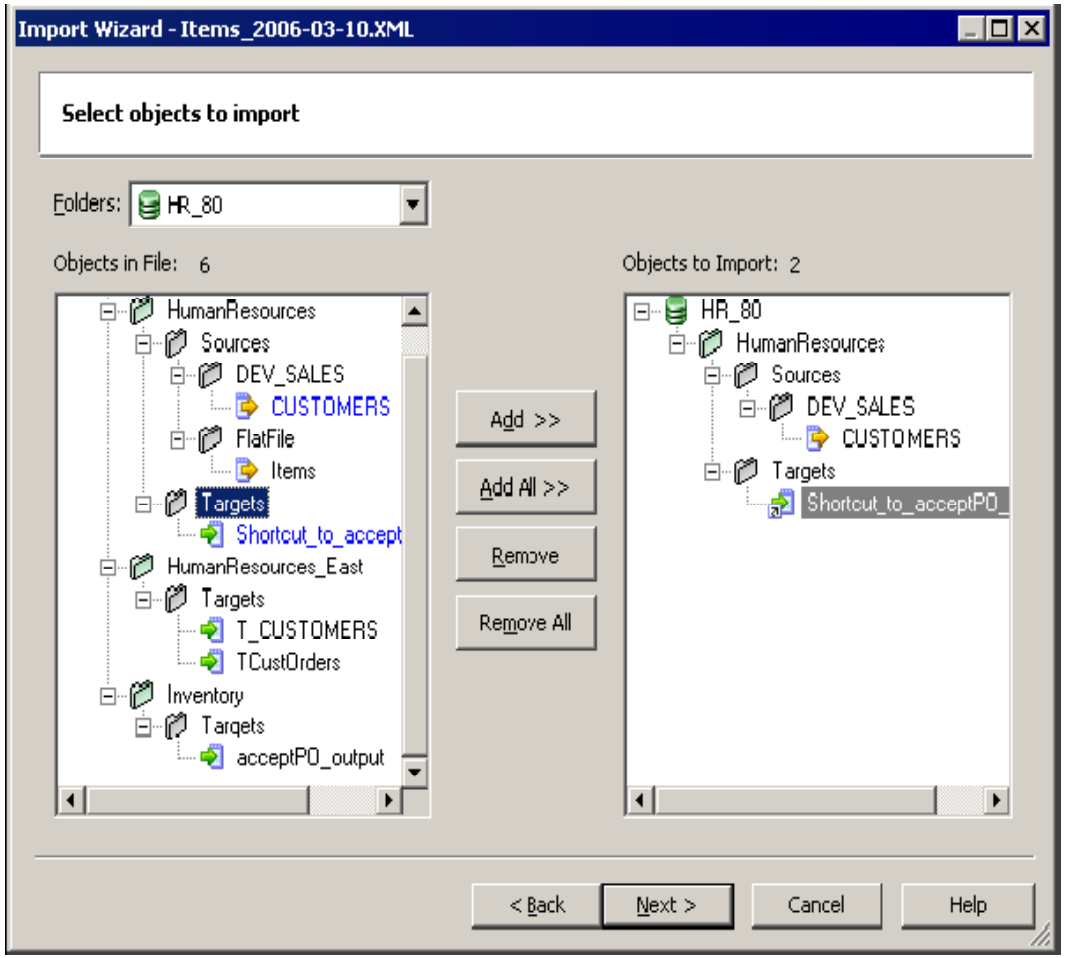
Cognizant | Academy
Passion for making a difference

Importing Objects



Cognizant | Academy
Passion for making a difference

Importing Objects



Grouping Versioned Objects

- To identify versioned objects that meet a certain criteria and group objects from the Designer, Workflow Manager, and Repository Manager, the following
  - Labels
  - Queries
  - Deployment groups

- A deployment group is a versioning object that allows you to group versioned objects for deployment to a different repository

Deployment Groups

- A deployment group is a global object that consists of versioned objects from one or more folders
- You can use a deployment group
  - To copy the versioned objects to another folder or repository
  - To copy some, but not all, of the objects in a folder
  - To copy objects from multiple folders
- You can create the following types of deployment groups
  - Static
  - Dynamic

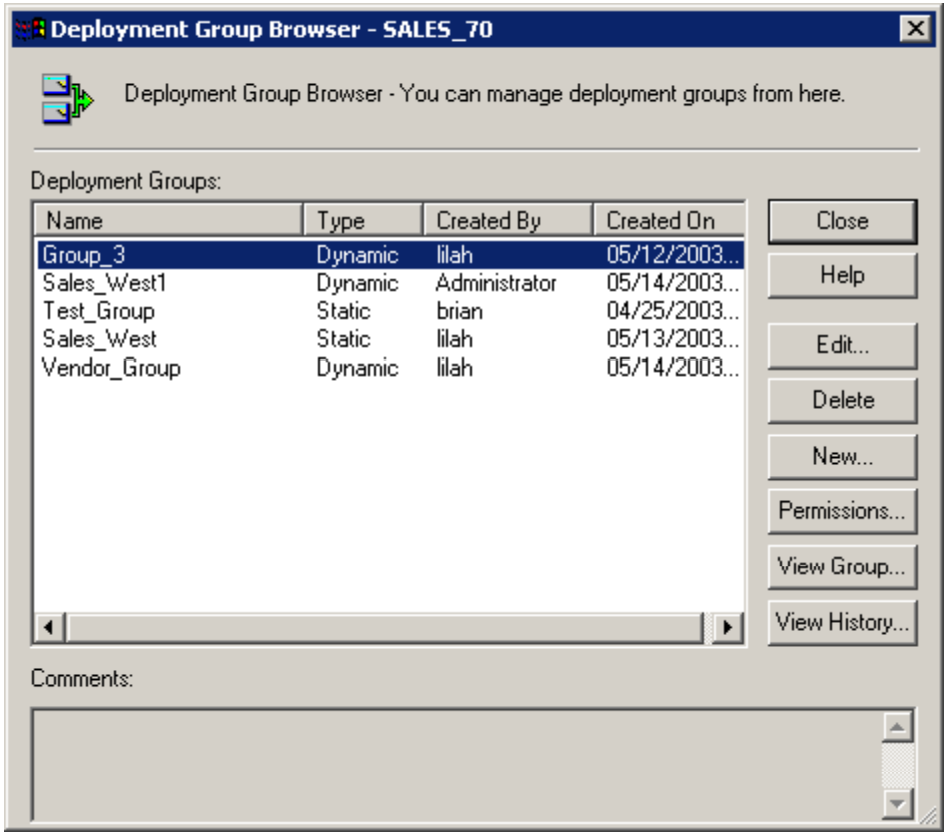
Types of Deployment Groups

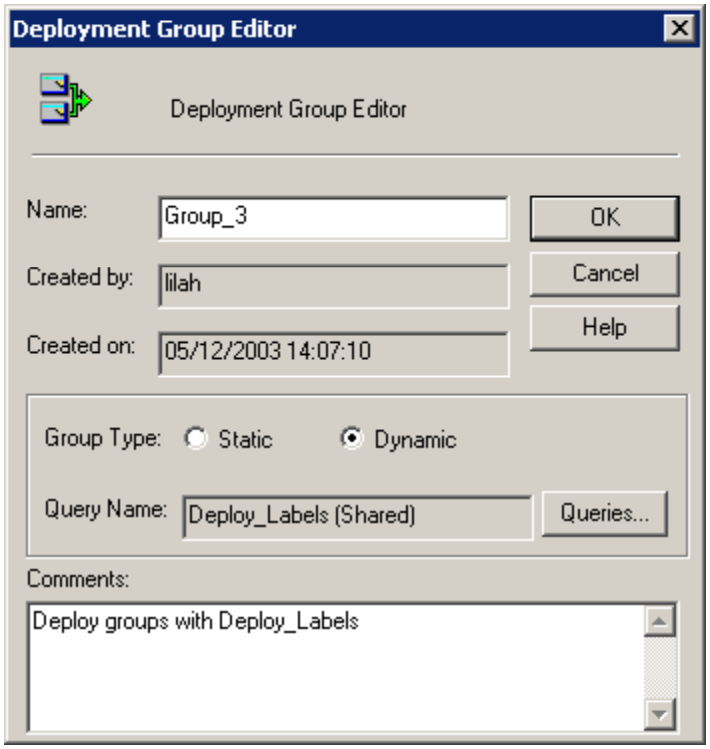
- **Static**
  - » Used when the set of deployment objects is not expected to change
- **Dynamic**
  - » Used when the set of deployment objects is expected to change frequently

Tasks on Deployment Groups

- You can complete the following tasks when you work with deployment groups
  - » Create a deployment group
  - » Edit a deployment group
  - » Configure permissions for a deployment group
  - » View the objects in a static or dynamic deployment group
  - » Add or remove objects in a static deployment group
  - » Associate a query with a dynamic deployment group
  - » View the history of a deployment group
  - » Roll back a deployment group

Deployment Group

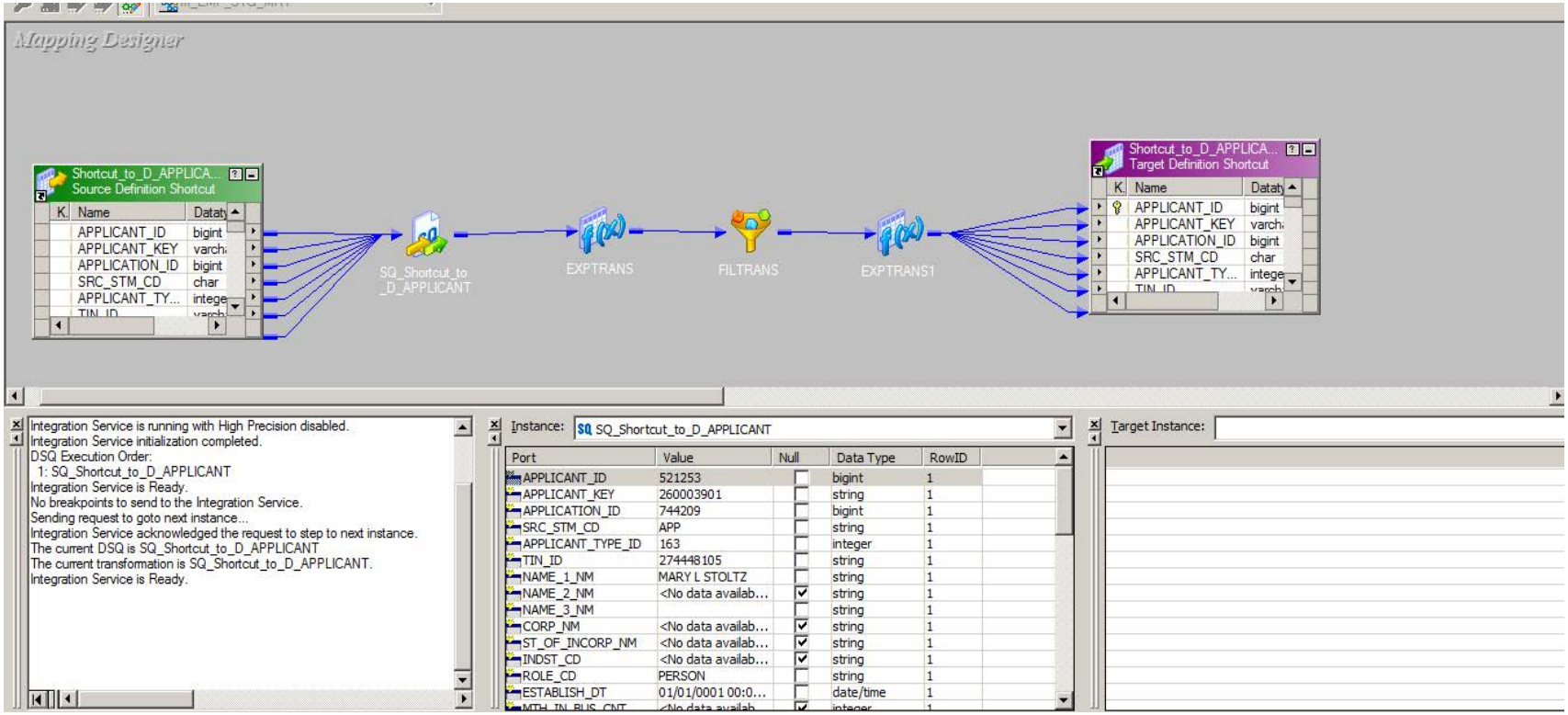




Debugger

- Debugger is an in-built tool available in Informatica PowerCenter for debugging a valid mapping to gain troubleshooting information about data and error conditions
- To debug a mapping, you configure and run the Debugger from within the Mapping Designer
- The Debugger uses a session to run the mapping on the PowerCenter
- When you run the Debugger, it pauses at breakpoints and allows you to view and edit transformation output data
- Different Debugger session types
  - » Use an existing non-reusable session for the mapping
  - » Use an existing reusable session for the mapping
  - » Create a debug session instance for the mapping

Debugger Windows



Debug Process

- Create Break Points in mapping where to evaluate the data and error conditions
- Configure the Debugger by selecting the session type
- Run the Debugger (The session is initialized by the Server, and reads the break points. It pauses the debugger when the breakpoints evaluate to true)
- Monitor the Debugger (Watch the target data, transformation output data, the debug log and the session log)
- Modify data and breakpoints to see the effect on transformations, mapplets, and targets as the data moves through the pipeline

Data Recovery

- If an error causes a session to stop unexpectedly, the usual procedure is
  - » Refer to the session logs to determine the cause of the failure.
  - » Correct the errors
  - » Complete the session

- Depending on the configuration and status of the workflow and session, you can choose one or more of the following recovery methods
  - » Recover a suspended workflow
  - » Recover a failed workflow
  - » Recover a session task

How to Perform a Data Recovery

- Configure the mapping
  - The input data should be sorted
  - Verify all targets receive data from transformations that produce repeatable data
- Configure the session
  - The session should be enabled for recovery in settings
  - The previous session run failed and the recovery information is accessible
- Configure the Workflow
  - The workflow needs to be configured as Suspend on Error, so that the errors can be corrected in between
- Configure the target database
  - Two recovery tables PM_RECOVERY and PM_TGT_RUN_ID needs to be created in the target database

Data Recovery : Limitations

- Data recovery cannot be successful under the following circumstances
  - If the number of partitions changed after the initial session failure
  - If the recovery table is empty or removed from target databases
  - Recovery cache file is empty
  - The server should be the same Operating system
  - The source data is not sorted
  - The source or target data changed after the initial session failure
  - The mapping uses a Normalizer or Sequence Generator transformation
  - The data movement mode changes after the initial session failure
  - The server code page changes after the initial session failure
  - The session writes to a relational target in bulk mode, but the session is not configured to truncate the target table

Partitioning

- A partition is a pipeline stage that executes in a single reader, transformation, or writer thread
- If you select the Partitioning option, you can increase the number of partitions. This increases the number of processing threads, which can improve session performance
- By default, the PowerCenter Server sets the number of partitions to one. You can generally define up to 64 partitions at any partition point
- Increasing the number of partitions or partition points increases the load on the server machine
- If the server machine contains ample CPU bandwidth, processing rows of data in a session concurrently can increase session performance

Partition Attributes and Types

- Partition attributes
  - Location of partition points
  - Number of partitions
  - Type of partition

- Partition types
  - Round robin, where server distributes data evenly among all partitions
  - Hash, where server applies the hash function to a partition key to group data
  - Key range, where one or more ports specified to form a compound key
  - Pass through, where all rows passed at one partition without redistributing
  - Database partitioning, where IBM DB2 is the target system

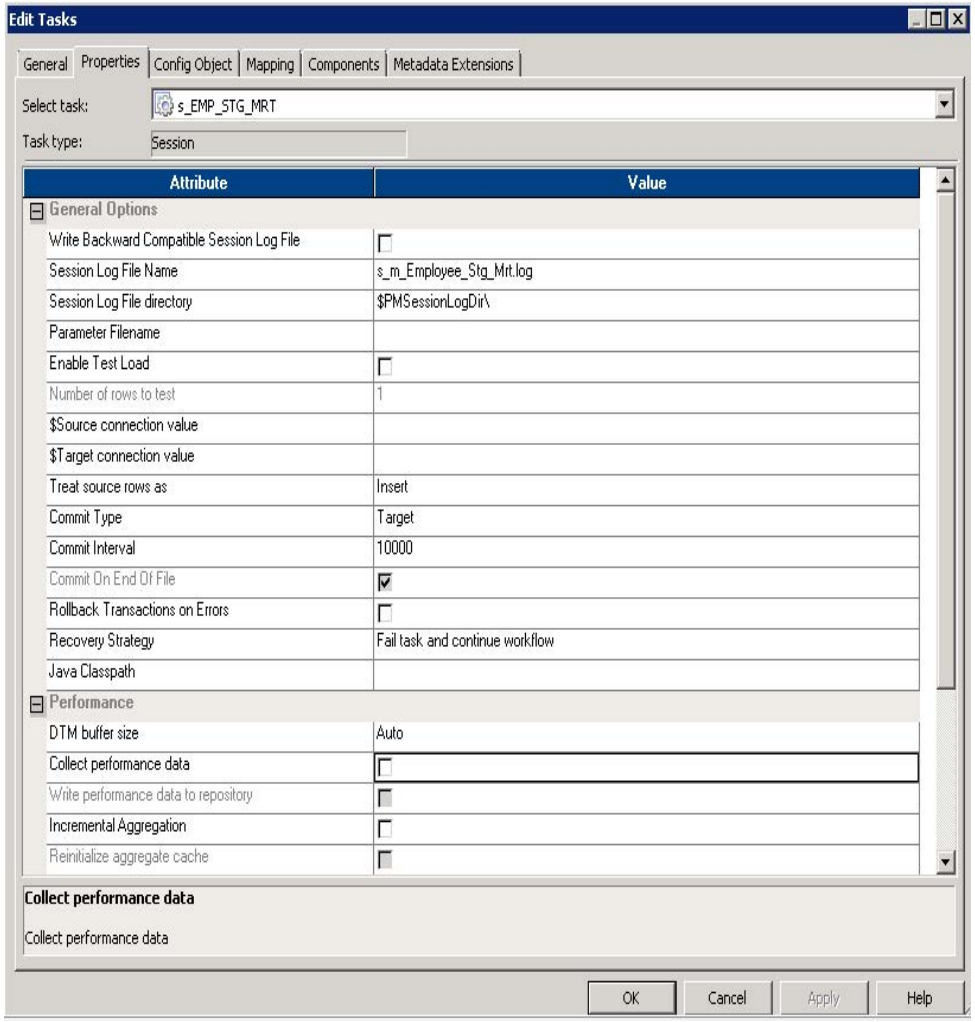
Factors Controlling Partitioning

- The partitioning information for a pipeline controls the following factors
  - The number of reader, transformation, and writer threads that the master thread creates for the pipeline
  - How the PowerCenter Server reads data from the source, including the number of connections to the source
  - How the PowerCenter Server distributes rows of data to each transformation as it processes the pipeline
  - How the PowerCenter Server writes data to the target, including the number of connections to each target in the pipeline

Performance Data Collection

- You can configure a session to collect performance details and store them in the PowerCenter repository
- you must also configure the Integration Service that runs the session to store the run-time information at the verbose level
- The Integration Service stores run-time information in the PowerCenter repository
- The Workflow Monitor displays performance details for each session that is configured to show performance details

Configuring Performance Details



- To configure performance details
  - » In the Workflow Manager, open the session properties
  - » On the Properties tab, select Collect Performance Data to view performance details while the session runs
  - » Select Write Performance Data to Repository to view performance details for previous session runs
  - » Click OK
  - » Save the changes to the repository

Bottlenecks

- Performance bottlenecks can be identified by the first step of Performance Tuning, to optimize session performance
- To tune the performance of a session, first you identify a performance bottleneck, eliminate it, and then identify the next performance bottleneck until you are satisfied with the session performance
- Performance bottlenecks can occur in the source and target databases, the mapping, the session, and the system
- You can use the test load option to run sessions when you tune session performance

Eliminating Bottlenecks

- Performance bottlenecks can be identified by the following methods
  - » Run test sessions
  - » Study performance details and thread statistics
  - » Monitor system performance

- Once you determine the location of a performance bottleneck, you can eliminate the bottleneck by following these guidelines
  - » Eliminate source and target database bottlenecks
  - » Eliminate mapping bottlenecks
  - » Eliminate session bottlenecks
  - » Eliminate system bottlenecks

Eliminating Database Bottlenecks

- Optimizing the source database
  - Optimize the query
  - Create tempdb as in-memory database
  - Use conditional filters
  - Increase database network packet size
  - Connect to Oracle databases using IPC protocol

- Optimizing the target database
  - Drop indexes and key constraints
  - Increase checkpoint intervals
  - Use bulk loading and external loading
  - Increase database network packet size
  - Optimize Oracle target databases

Eliminating Mapping/Session Bottlenecks

- Optimizing the mapping
  - Configure single-pass reading
  - Optimize data type conversions
  - Eliminate transformation errors
  - Optimize transformations
  - Optimize expressions

- Optimizing the target database
  - Increase the number of partitions
  - Reduce errors tracing
  - Remove staging areas
  - Tune session parameters

Questions?

Test Your Understanding

1. _____ is used to store copies of previous versions of objects.

2. To edit an object, you must _____ the object.

3. In the mapping designer, troubleshooting information about data and error conditions for a valid mapping can be gained from using _____.

4. Choose the correct option: To copy an object to another folder, you must open the a) Source Folder b) Target Folder

5. What are the two types of Deployment Groups?

Try it Out

1. Export/Import repository objects to/from the XML file.

2. Configure the debugger and debug an existing mapping.

3. Modify the Session Properties to collect performance details.

Tips

- More Performance Tuning Techniques are available in the attached document

Tuning Tips

Advanced Topics : Summary

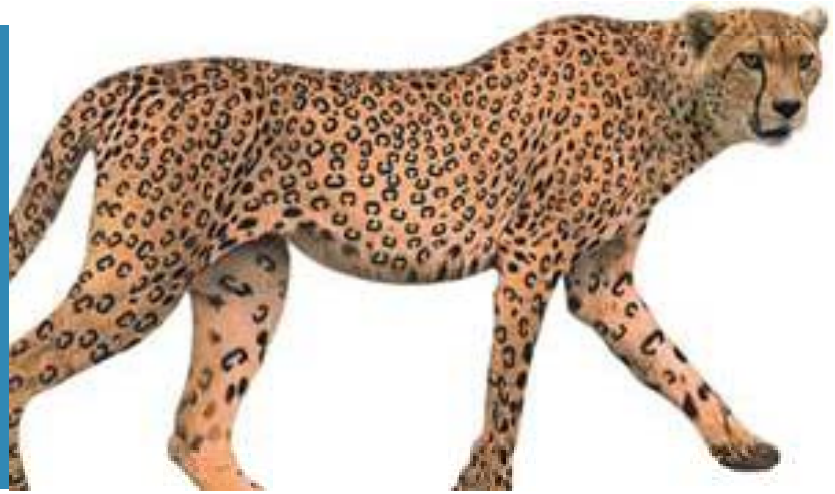
- Versioning is a concept where you can configure the repository to store multiple versions of objects
- In the PowerCenter Client, you can copy repository objects & import/export repository objects from/to XML files
- A deployment group is a global object that consists of versioned objects from one or more folders
- Debugger is an in-built tool for debugging a valid mapping to gain troubleshooting information about data and error conditions
- To tune performance Check ‘Collect Performance Data' option in the session properties of a session in Server Manager

Advanced Topics : Source

- PDF manuals that comes with Informatica Software
- www.informatica.com

Cognizant | Academy
Passion for making a difference

You have completed the Session “Advanced Topics” of Informatica 9x



Academy