

**Informatica
PowerCenter**

Lesson 3-Workflow Manager

Lesson Objectives

- In this module you will learn about:
 - Workflow Manager components
 - Workflow Monitor
 - Database connections
 - Tasks
 - Session Task
 - Debugger



3.1: Workflow Manager components

Workflow

- It is a set of instructions for the Integration service to perform the data transformation load
- The Integration service retrieves mapping, Workflow, and session metadata from the repository to:
 - Extract data from the source
 - Transform it
 - Load data into the target
- It combines the logic of Session Tasks, other types of Tasks and Worklets



Figure 3.1

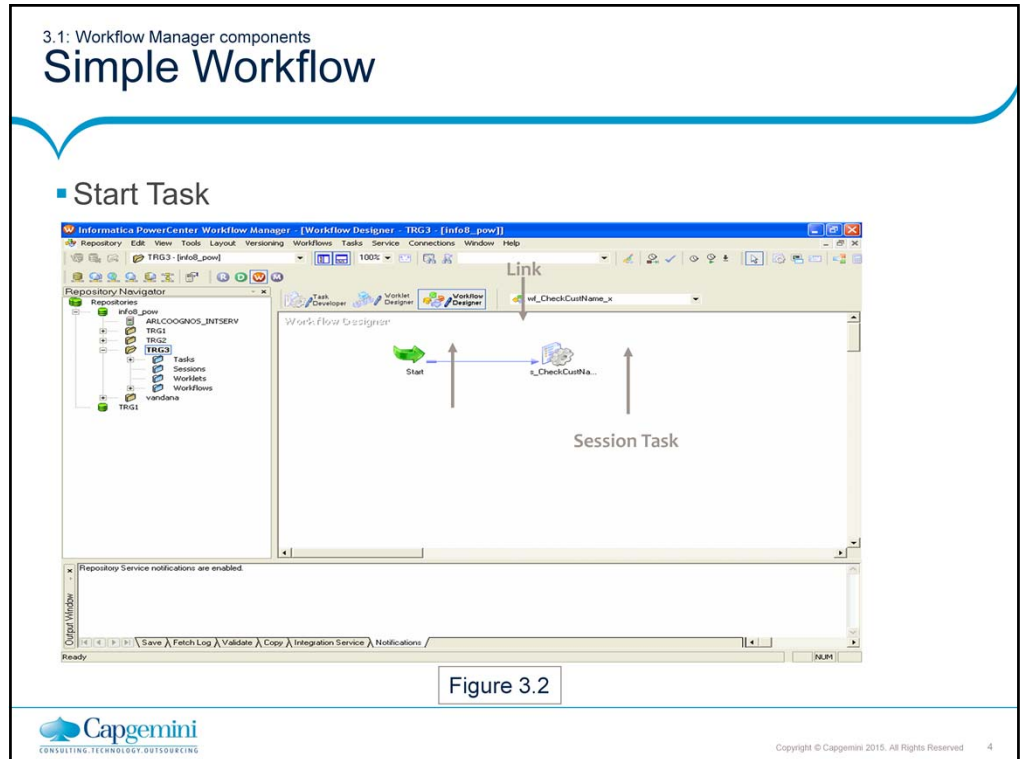


Copyright © Capgemini 2015. All Rights Reserved 3

Once a mapping is created in the Mapping Designer, the source to target dataflow specification is ready.

A *Workflow* is a set of instructions for the Integration Service to perform the data transformation and loading. When a Workflow starts, the Informatica Integration Service retrieves mapping, Workflow, and session metadata from the repository to extract data from the source, transform it, and load it into the target. It also runs the Tasks in the Workflow. The Integration Service uses Load Manager and Data Transformation Manager (DTM) processes to run the Workflow.

A *Workflow* combines the logic of Session Tasks, other types of Tasks and Worklets. A Session is a type of Task related to the movement of data through the Server. Tasks are the building blocks of a Workflow.



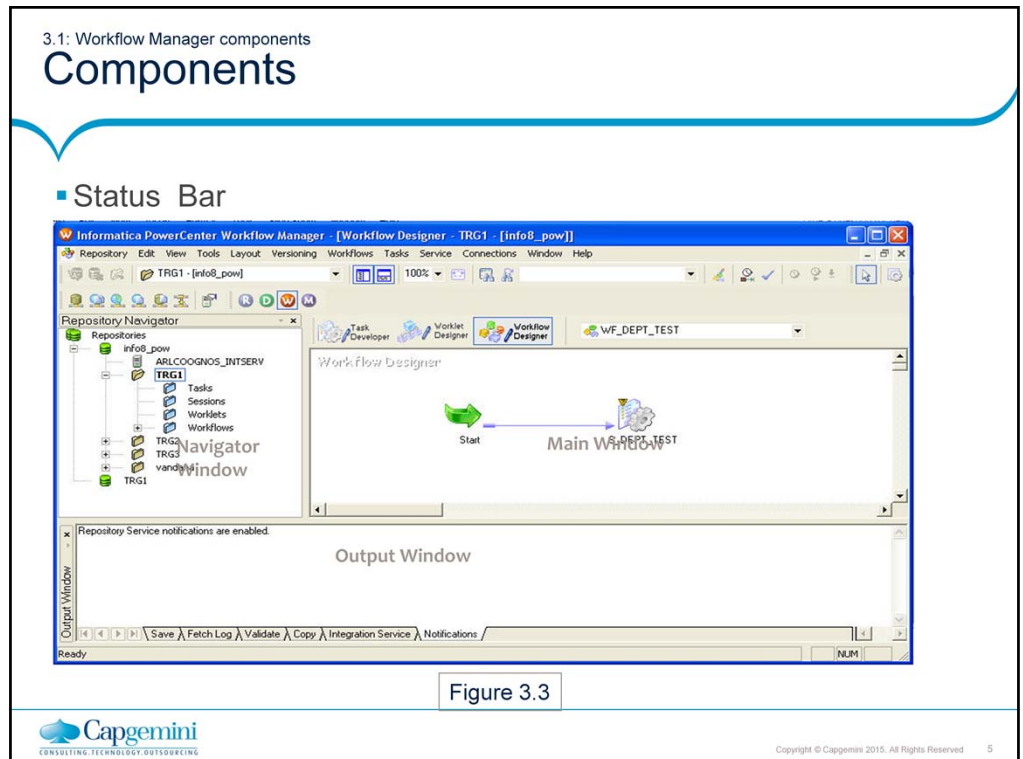
The simplest Workflow is composed of:

- Start Task
- A Link
- A Session Task (and any other Task as required for execution)

The first step to develop a Workflow is to create a new Workflow in the Workflow Designer. A Workflow must contain a Start Task. The Start Task represents the beginning of a Workflow.

When a Workflow is created, the Workflow Designer creates a Start Task and adds it to the Workflow. The Start Task cannot be deleted. The next step is to add Tasks to the Workflow. The Workflow Manager includes Tasks such as the Session Task, the Command Task, and the Email Task to design the Workflow.

Finally, the Workflow Tasks are connected with links to specify the order of execution in the Workflow.



The *Workflow Manager* tool is used to define the workflow to execute mappings built in the Designer. After a Workflow is created, it is run in the Workflow Manager and monitored in the Workflow Monitor.

The Workflow Manager provides several windows:

- The **Navigator Window** - Allows to view servers and all folders within a repository and the contents of those folders
- The **Main Window** – Allows to view all the sessions that have been created and linked to make a complete Workflow
- The **Output Window** - Contains messages from the server, such as success or failure to schedule or start sessions. The Output window contains the following tabs:
 - **Save** - Displays messages when a Workflow, a Worklet, or a Task is used. The Save tab displays a validation summary when a Workflow or a Worklet is saved
 - **Fetch Log** - Displays messages when the Workflow Manager fetches objects from the repository
 - **Validate** - Displays messages when a Workflow, a Worklet, or a Task is validated
 - **Copy** - Displays messages when repository objects are copied
 - **Server** - Displays messages from the Integration Service
 - **Notifications** - Displays messages from the Repository Service

3.1: Workflow Manager components

Development Tools

- Task Developer
 - To develop various Tasks like Session, Command etc in a Workflow
- Worklet Designer
 - To develop reusable Workflows
- Workflow Designer
 - To develop Workflows



Figure 3.4

The Workflow Manager has three development tools:

Task Developer

- Used to construct Session, Shell Command and Email Tasks
- Tasks created in Task Developer are reusable in Worklets or Workflows

Worklet Designer

- Used to construct an object which represents a set of Tasks
- Worklets are reusable in multiple Workflows

Workflow Designer

- “Maps” the execution order of Sessions, Tasks and Worklets for the Integration Service

3.2: Workflow Monitor

Introduction

- Is a tool used to monitor Workflows and Tasks
- Displays the details of Workflow runs

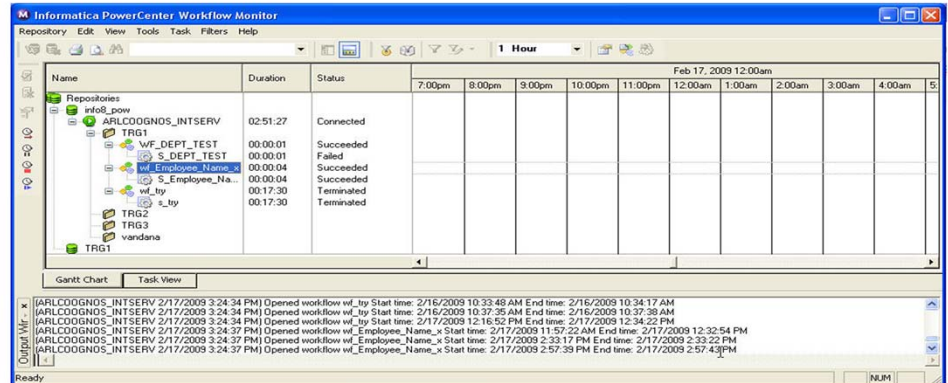
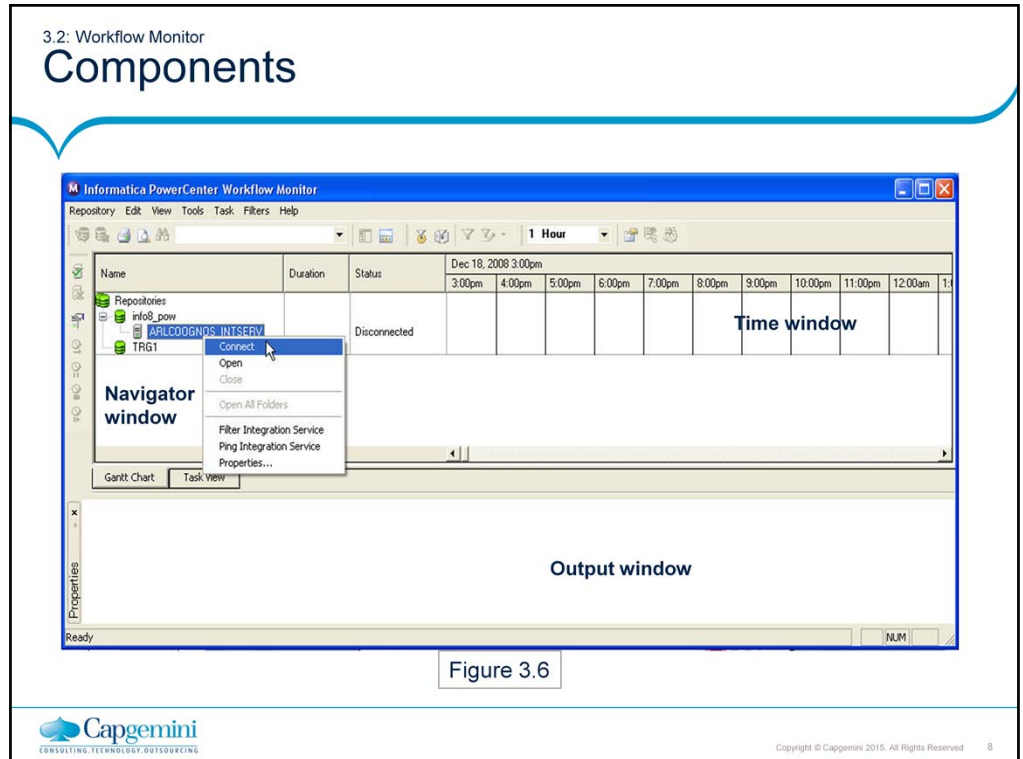


Figure 3.5

The *Workflow Monitor* is a tool used to monitor Workflows and Tasks. The details about a Workflow or a Task can be viewed in either Gantt Chart view or Task view. Workflows can be run, stopped, aborted, and resumed from the Workflow Monitor. The Workflow Monitor displays Workflows that have run at least once.

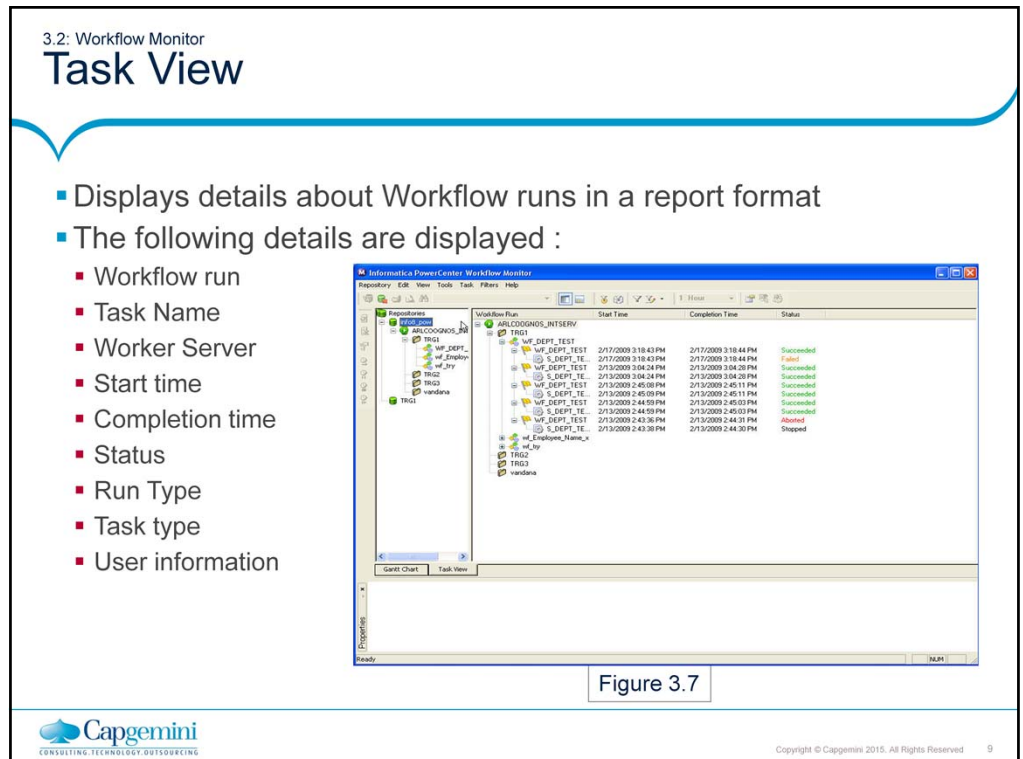


The Server can be monitored in two modes:

- **Online mode** – In this mode, the Workflow Monitor continuously receives the information from the Integration Service and the Repository Service
- **Offline mode** – Workflow Monitor displays historic information about past Workflow runs by fetching information from the Repository

The Workflow Monitor consists of the following windows:

- **Navigators window** - Displays monitored repositories, servers, and repository objects
- **Output window** - Displays messages from the Integration Service
- **Time window** - Displays progress of Workflow runs



The Task view displays details about Workflow runs in a report format. It displays the following details:

- **Task name** - The name of the task will be displayed under the workflow name
- **Folder name** - The name of the folder containing the Task
- **Status** - The status of the Task or Workflow
- **Workflow name** - The name of the Workflow
- **Worklet name** - The name of the Worklet
- **Start time** - The time that the Integration Service starts executing the Task or Workflow
- **Completion time** - The time that the Integration service finishes executing the Task or Workflow
- **Status message** - Message from the Integration Service regarding the status of the Task or Workflow
- **User name** - Name of the user who owns the Workflow or Task
- **Run type** - The method used to start the Workflow. Workflow can be manually started or scheduled to start
- **Task type** - The type of the Task

3.2: Workflow Monitor

Gantt Chart View

- Displays the details about Workflow runs in chronological format
- The following details are displayed:
 - Task name
 - Duration
 - Status
 - Connection between
 - Line 3.1.
 - objects

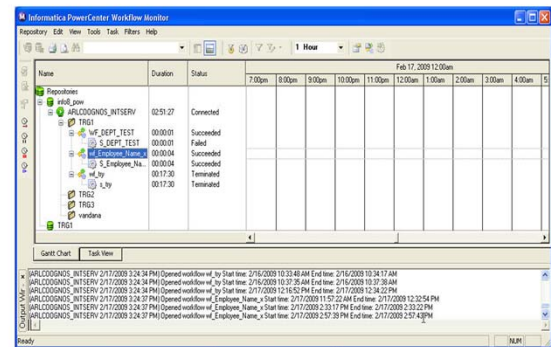


Figure 3.8

The Gantt Chart view displays details about Workflow runs in chronological format. It displays the following information:

- **Task name** - Name of the Task in the Workflow
- **Duration** - The length of time the Integration service spends executing the most recent Task or Workflow
- **Status** - The status of the most recent Task or Workflow
- **Connection between objects** - The Workflow Monitor shows links between objects in the Time window

3.3: Database Connections

Introduction

- A database connection is configured in the Workflow Manager before the Server can access a source or target database in a session
- When a session that reads from or writes to a relational database is created or modified, only configured source and target databases can be selected
- Database connections are saved in the repository



Copyright © Capgemini 2015. All Rights Reserved 11

Before running any Workflow, the first thing that needs to be done is to configure a database connection. A database connection specifies the source and target database connection information. When a session that reads from or writes to a relational database is created or modified, only configured source and target databases can be selected.

For every source and target database used in the session, a separate database connection has to be configured.

To create a database connection, one of the following privileges is required:

- Use Workflow Manager
- Super User

3.3: Database Connections

Information

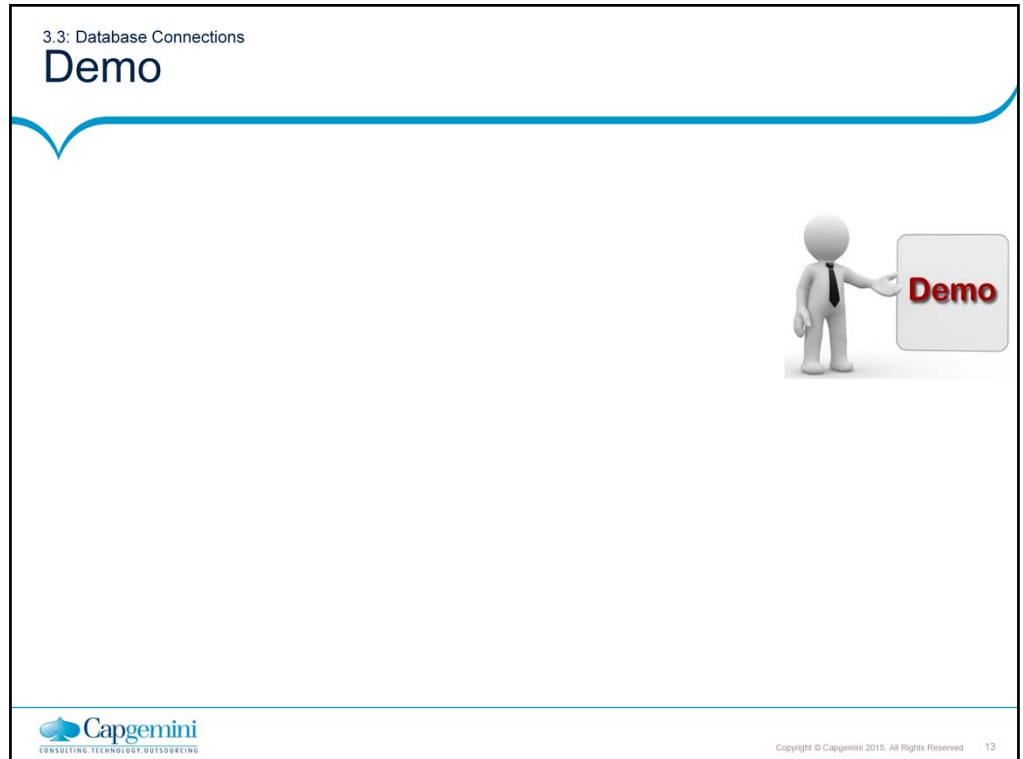
- To create a connection, the following information should be available:
 - Database name
 - Database type
 - Database username
 - Password
 - Connect string
 - Database code page

| Attribute | Value |
|----------------------------|-------------------------------------|
| Connection Environment ... | |
| Transaction Environment... | |
| Enable Parallel Mode | <input checked="" type="checkbox"/> |
| Connection Retry Period | 0 |

Figure 3.9

To create a connection, the following information should be available:

- **Database name** - Name for the connection
- **Database type** - Type of the source or target database
- **Database username** - Name of a user who has the appropriate database permissions to read from and write to the database
- **Password** - Database password
- **Connect string** - Connect string used to communicate with the repository
- **Database code page** - Code page associated with the database

**Steps to create a Relational Database Connection:**

1. In the Workflow Manager, connect to a repository.
2. Choose **Connections / Relational** menu option. A dialog box appears, listing all the registered source and target database connections.
3. Select the type of database connection to be created.
4. Click **Add**. The Connection Object Definition dialog box appears.
5. Enter the database connection name, database connection type, username, password, connect string and code page.
6. Click **OK**.

3.4: Tasks

Introduction

- Tasks are used to build Workflows and Worklets
- Tasks can be created in:
 - Task Developer
 - Workflow Designer
 - Worklet Designer
- Tasks created in the Task Developer are reusable
- Tasks created in the Workflow Designer and Worklet Designer are non-reusable by default



Copyright © Capgemini 2015. All Rights Reserved 14

Tasks are building blocks of a Workflow. The Workflow Manager contains many types of Tasks to build Workflows and Worklets. Tasks created in the Task Developer are reusable and the ones created in the Workflow Designer and Worklet Designer are non-reusable.

The Workflow Manager validates Task attributes and links. If a Task is invalid, the Workflow becomes invalid. Workflows containing invalid sessions may still be valid.

3.4: Tasks

Workflow Tasks

■ Reusable Tasks

| Task Name | Description |
|-----------|---|
| Session | Set of instructions to execute a mapping |
| Command | Specifies shell commands to run during the Workflow |
| Email | Sends email during the Workflow |

Reusable Tasks can be used in multiple Workflows in the same folder. The reusable Tasks can be seen in the Tasks node in the Navigator window.

3.4: Tasks

Workflow Tasks

■ Non-reusable Tasks

| Task Name | Description |
|-------------|---|
| Event-Raise | Used to trigger a user-defined event |
| Event-Wait | Waits for a user-defined or a pre-defined event to occur |
| Timer | Waits for a specified period of time to execute the next Task |
| Control | Stops or aborts the Workflow |
| Decision | Specifies a condition to evaluate in the Workflow. Use the Decision Task to create branches in a Workflow |

Tasks created within the Workflow are non-reusable Tasks. The Non-reusable Tasks can be seen in the Sessions node in the Navigator window.

3.5: Session Task

Introduction

- Created for each mapping to be run on the Integration Service
- Integration Service uses the instructions configured in the session to move data from sources to targets



Figure 3.10

A *Session Task* is the basic object created in the Workflow Manager that assembles the information contained in a mapping together with database connection and other information. It is the vehicle for commanding the Integration Service to extract, transform, and load data and report session status and performance characteristics. A Workflow groups two or more sessions to run sequentially or concurrently or as specified in the Workflow.

A reusable Session Task is created in the Task Developer. We can also create Session Tasks in the Workflow Designer as we develop the Workflow. Once the Session is created, the session properties can be edited at any time.

3.5: Session Task

Create Session Task

- Create a Task
- Select the type of Task and give a name to the Task
- Select the mapping to be associated with the Task



Figure 3.11

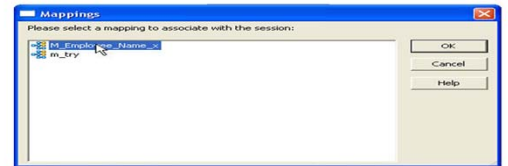
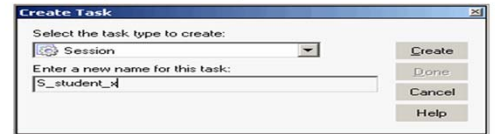


Figure 3.13



CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 18

To create a new Session Task, the following information is required:

- Mapping used for the Session Task
- Session name, which must be unique among all sessions in a given folder
- Source type
- Update strategy for writing to targets
- Target type

Note: Before creating a Session Task, the Workflow Manager has to be configured to communicate with databases and the Integration service. Appropriate permissions have to be assigned for any database, FTP, or external loader connections.

3.5: Session Task

Properties

- The Session Task properties has the following tabs

- General
- Properties
- Config Object
- Mapping
- Components
- Metadata Extension

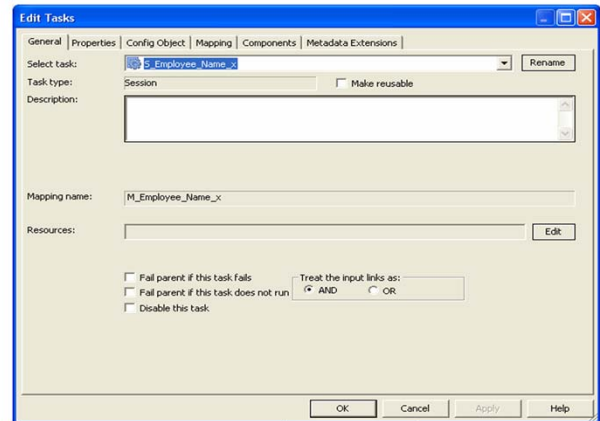
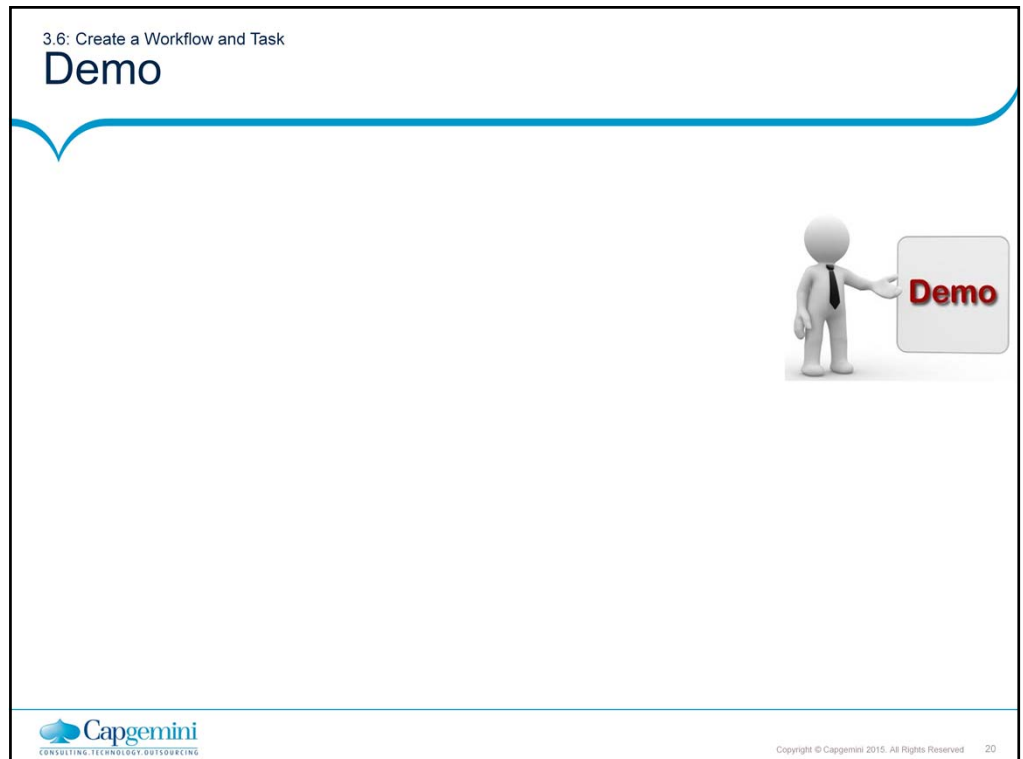


Figure 3.14

The Session Task properties has the following tabs:

- **General** - Specifies session name, mapping name, and description for the Session Task
- **Properties** - Specifies Session log information, test load settings, and performance configuration
- **Config Object** - Specifies advanced settings, log options, and error handling configuration
- **Mapping** – Specifies source, target information and Override transformation properties
- **Components** - Configure pre- or post-session shell commands and emails
- **Metadata Extension** - Associate additional non-business information with repository objects

**To create a Workflow:**

1. Open the Workflow Designer.
2. Choose **Workflows / Create** menu option.
3. Enter a name for the new Workflow.
4. Click **OK**.

To create a Task:

1. In the Workflow Designer, click the **Session Task** icon on the Tasks toolbar.
2. Choose **Tasks / Create** menu option.
3. Select **Session** task for the Task type. Enter a name for the Session Task.
4. Click **Create**. The Mappings dialog box appears.
5. Select the mapping to be used in the Session Task and click **OK**.
6. Click **Done**. The Session Task appears in the workspace.



Click in the right corner of the Expression field, enter the non-aggregate expression using one of the input ports, then click OK. To group by on a column check the group by checkbox.

Click Add and enter a name and data type for the aggregate expression port. Make the port an output port by clearing Input (I).

Click in the right corner of the Expression field to open the Expression Editor. Enter the aggregate expression, click Validate, then click OK.

Make sure the expression validates before closing the Expression Editor.

Add default values for specific ports as necessary.

Click OK.

Choose Repository | Save to save changes to the mapping.

3.7: Debugger

Information

- The Debugger is used to debug a valid mapping to trace errors and gain more insight into the data
- The mapping can be debugged
 - Before a session is run
 - After a session is run
- The Debugger can run for the following session types:
 - Existing non-reusable session
 - Existing reusable
 - Debug session instance



Copyright © Capgemini 2015. All Rights Reserved 22

You can debug a valid mapping to know about data and error conditions. To debug a mapping, the Debugger is configured and run from within the Mapping Designer.

When the Debugger is run, it pauses at breakpoints and allows viewing and editing of transformation output data.

The Debugger uses a session to run the mapping on the PowerCenter Server. When the Debugger is configured, three different debugger session types can be selected. The Debugger runs a workflow for each session type. The Debugger session types are :

- **Existing non-reusable session for the mapping.** The Debugger uses existing source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs the non-reusable session and the existing workflow.
- **Existing reusable session for the mapping.** The Debugger uses existing source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs a debug instance of the reusable session and creates and runs a debug workflow for the session.

Debugger

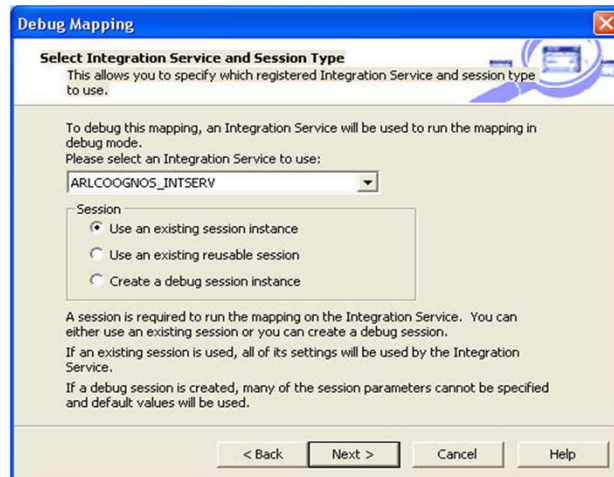


Figure 3.15

contd.

- **Debug session instance created for the mapping.** Using the Debugger Wizard, the user can configure source, target, and session configuration properties. When the Debugger is run, the PowerCenter Server runs a debug instance of the debug workflow and creates and runs a debug workflow for the session.

Debug Process

Use the following process to debug a mapping:

1. **Create breakpoints.** Create breakpoints in a mapping if PowerCenter Server is to evaluate data and error conditions.
2. **Configure the Debugger.** Use the Debugger Wizard to configure the Debugger for the mapping. Select the session type the PowerCenter Server uses when it runs Debugger. When a debug session is created, a subset of session properties within the Debugger Wizard need to be configured, such as source and target location. You can also choose to load or discard target data.
3. **Run the Debugger.** Run the Debugger from within the Mapping Designer. When the Debugger is run, the Designer connects to the PowerCenter Server. The PowerCenter Server initializes the Debugger and runs the debugging session and workflow. The PowerCenter Server reads the breakpoints and pauses the Debugger when the breakpoints evaluate to true.



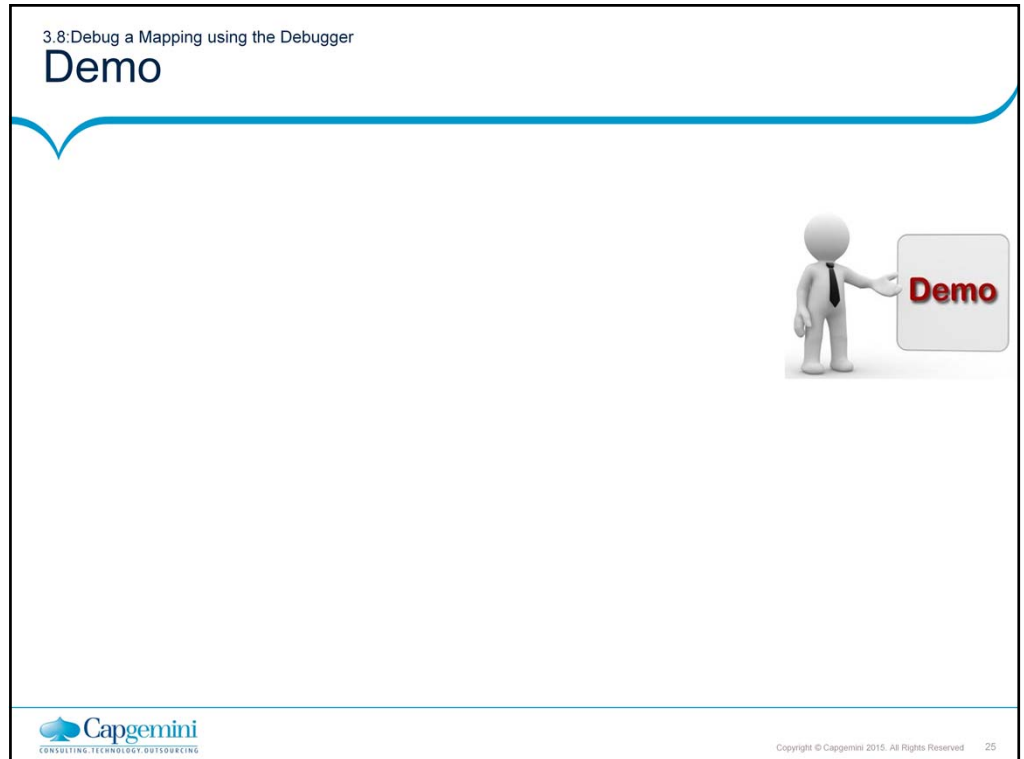
Monitor the Debugger. When the Debugger is running, target data, transformation and maplet output data, the debug log, and the session log can be monitored. When the Debugger is running, the Designer displays the following windows:

Debug log. View messages from the Debugger.

Target window. View target data.

Instance window. View transformation data.

Modify data and breakpoints. When the Debugger pauses, data can be modified to see the effect on transformations, maplets, and targets as the data moves through the pipeline. Breakpoint information can also be modified



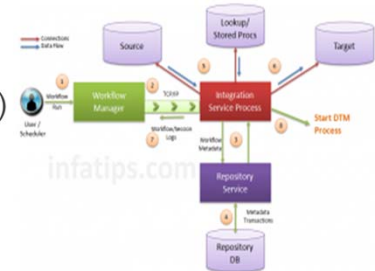
To use the Debugger

1. Select a valid mapping.
2. Select **Mappings – Debugger – Edit Breakpoints** to create breakpoints if any.
3. Select **Mappings – Debugger – Start Debugger** to invoke the Debugger.
4. Select the **server**, **session type**, **session instance** and **target table options** to configure the debugger.
5. Monitor the debugger. Observe transformation data in the instance window and target data in the target window.
6. Modify data if required.

PowerCenter Integration Service Process

Integration Service mainly consists of three process

- Integration Service Process
- Load Balancer
- Data Transformation Manager(DTM)



Integration Service Process:

- Integration Service process is responsible to carry out below tasks:
 - Workflow Schedule Management
 - Locking Workflow Before Execution
 - Reading Parameter File
 - Creating Workflow Logs
 - Running any Tasks in Workflow (e.g., Email)
 - Creating DTM processes to run Workflow Sessions



Copyright © Capgemini 2015. All Rights Reserved 25

Whenever Integration Service receives a request to execute a workflow, it spawns an Integration Service Process. In admin community it is typically called as pmserver process.

Note that Integration Service can start one or multiple Integration Processes to run and monitor Workflows.

Whenever a workflow is scheduled or changed, Workflow Manager interacts with Integration Service Process. This interaction happens over TCP/IP. Integration Service process locks the Workflow and then knocks the Repository Service to get metadata about Workflow. Repository Service intern connects to Repository Database – fetches desired information – and gives it back to Integration Service process.

Once Integration Service process receives metadata information back from Repository Service, it creates connections to Source and Target databases using ODBC or Native drivers. Integration Service process also creates connections to any lookup or stored procedure databases which are used by the Workflow. Once the data loading process is done, Integration Service process releases and closes all open database connections.

Integration Service process is also responsible to start DTM process that will actually execute Workflow Sessions. Integration Service process passes Parameter File and Session information to DTM process, that helps it to retrieve required metadata from the repository. Please refer to article [Data Transformation Manager](#) for details on DTM Process.

Load Balancer / Manager

■ While running a Workflow, the PowerCenter Server uses the Load Manager process and the Data Transformation Manager Process (DTM) to run the workflow and carry out workflow tasks. When the PowerCenter Server runs a workflow, the Load Manager performs the following tasks:

1. Locks the workflow and reads workflow properties.
2. Reads the parameter file and expands workflow variables.
3. Creates the workflow log file.
4. Runs workflow tasks.
5. Distributes sessions to worker servers.
6. Starts the DTM to run sessions.
7. Runs sessions from master servers.
8. Sends post-session email if the DTM terminates abnormally.

DTM Process

- When the PowerCenter Server runs a session, the DTM performs the following tasks:
 1. Fetches session and mapping metadata from the repository.
 2. Creates and expands session variables.
 3. Creates the session log file.
 4. Validates session code pages if data code page validation is enabled. Checks query conversions if data code page validation is disabled.
 5. Verifies connection object permissions.
 6. Runs pre-session shell commands.
 7. Runs pre-session stored procedures and SQL.
 8. Creates and runs mapping, reader, writer, and transformation threads to extract, transform, and load data.
 9. Runs post-session stored procedures and SQL.
 10. Runs post-session shell commands.
 11. Sends post-session email.



Code Page And Data Movement Modes

- The data movement mode depends on whether Informatica Server should process single byte or multi-byte character data. This mode selection can affect the enforcement of code page relationships and code page validation in the Informatica Client and Server.
 - a) Unicode – IS allows 2 bytes for each character and uses additional byte for each non-ascii character (such as Japanese characters)
 - b) ASCII – IS holds all data in a single byte.
- The IS data movement mode can be changed in the Informatica Server configuration parameters. This comes into effect once you restart the Informatica Server.
- Code Page is used to identify characters that might be in different languages.
- Incorrect selection of Code Page can lead to Junk Data or rejection of data.
- ASCII, UTF-8, UTF-16, UTF-32 are few examples for encoding/ Code pages available in Informatica.

Audit Trails Output Files

- While Informatica Load Manager and DTM threads together executes the code, it also creates multiple files for audit trails. These files help us in analyzing the code for performance enhancement and data quality.
- Informatica server log: Informatica server (on UNIX) creates a log for all status and error messages (default name: pm.server.log). It also creates an error log for error messages. These files will be created in Informatica home directory.
- Session log file: Informatica server creates session log file for each session. It writes information about session into log files such as initialization process, creation of SQL commands for reader and writer threads, errors encountered and load summary. The amount of detail in session log file depends on the tracing level that you set.
- Session detail file: This file contains load statistics for each target in mapping. Session detail includes information such as table name, number of rows written or rejected. You can view this file by double clicking on the session in monitor window.

Audit Trails Output Files

- Performance detail file: This file contains information known as session performance details which helps you where performance can be improved. To generate this file select the performance detail option in the session property sheet.
- Reject file: This file contains the rows of data that the writer does not write to targets.
- Control file: Informatica server creates control file and a target file when you run a session that uses the external loader. The control file contains the information about the target flat file such as data format and loading instructions for the external loader.
- Post session email: Post session email allows you to automatically communicate information about a session run to designated recipients. You can create two different messages. One if the session completed successfully the other if the session fails.

Audit Trails Output Files

- Indicator file: If u use the flat file as a target, you can configure the informatica server to create indicator file. For each target row, the indicator file contains a number to indicate whether the row was marked for insert,update,delete or reject.
- Output file: If session writes to a target file, the informatica server creates the target file based on file properties entered in the session property sheet.
- Cache files: When the informatica server creates memory cache it also creates cache files. For the following circumstances informatica server creates index and datacache files.
 - Aggregator transformation
 - Joiner transformation
 - Rank transformation
 - Lookup transformation



Summary

- After completing this module you now know:
 - The Workflow Manager components
 - What is a Task
 - How to create Workflow
 - How to create a Task
 - Use the Debugger



Review Question

- Question 1: The _____ is the connection between the server and the repository
- Question 2: Workflow monitor displays details about Workflow runs in two views: _____ and _____
- Question 3: _____ are reusable in multiple Workflows



Review Question

- Question 4: Session Task is a reusable Task.
 - True/False

- Question 5: The Debugger can be used for an invalid mapping.
 - True/False

