INFORMATICA®

Informatica PowerCenter 7.1 Basics

Education Services

Version PC7B-20040608

© Informatica Corporation, 2003 - 2004. All rights reserved.

Introduction

INFORMATICA

Course Objectives

By the end of this course you will:

- Understand how to use the major PowerCenter components for development
- Be able to build basic ETL mappings and mapplets
- Be able to create, run and monitor workflows
- Understand available options for loading target data
- Be able to troubleshoot most problems

About Informatica

- Founded in 1993
- Leader in enterprise solution products
- Headquarters in Redwood City, CA
- Public company since April 1999 (INFA)
- 2000+ customers, including over 80% of Fortune 100
- Strategic partnerships with IBM, HP, Accenture, SAP, and many others
- Worldwide distributorship

Informatica Resources

- www.informatica.com – provides information (under Services) on:
  - Professional Services
  - Education Services
  - Technical Support
- my.informatica.com – sign up to access:
  - Product documentation (under Products, documentation downloads)
  - Velocity Methodology (under Services)
  - Knowledgebase
  - Webzine
- devnet.informatica.com – sign up for Informatica Developers Network

Informatica Professional Certification

Informatica offers three distinct Certification titles:

INFORMATICA ADMINISTRATOR

- **Exam A: Architecture and Administration**
- **Exam C: Advanced Administration**

INFORMATICA DESIGNER

- **Exam A: Architecture and Administration**
- **Exam B: Mapping Design**
- **Exam D: Advanced Mapping Design**

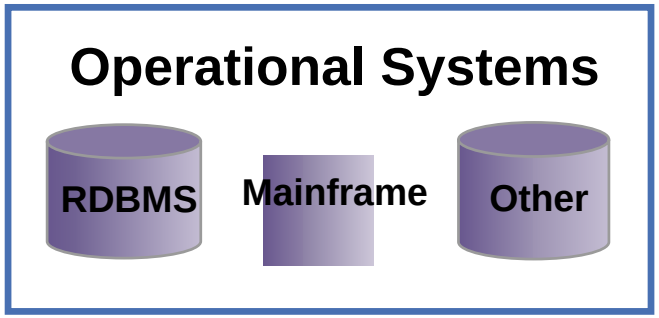
INFORMATICA CONSULTANT

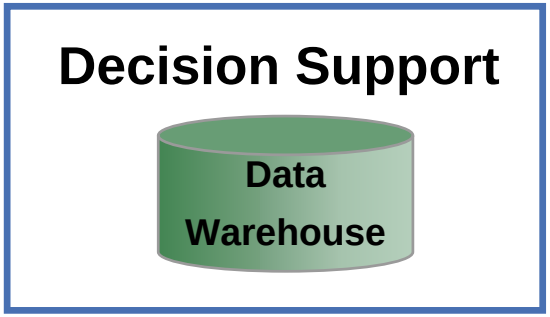
- **Exams A, B, C, D plus**
- **Exam E: Enablement Technologies**

For more information and to register to take an exam:

http://www.informatica.com/services/Education+Services/Professional+Certification/

Extract, Transform and Load





- Transaction level data
- Optimized for transaction response time
- Current
- Normalized or De-normalized data

- Aggregate data
- Cleanse data
- Consolidate data
- Apply business rules
- De-normalize data

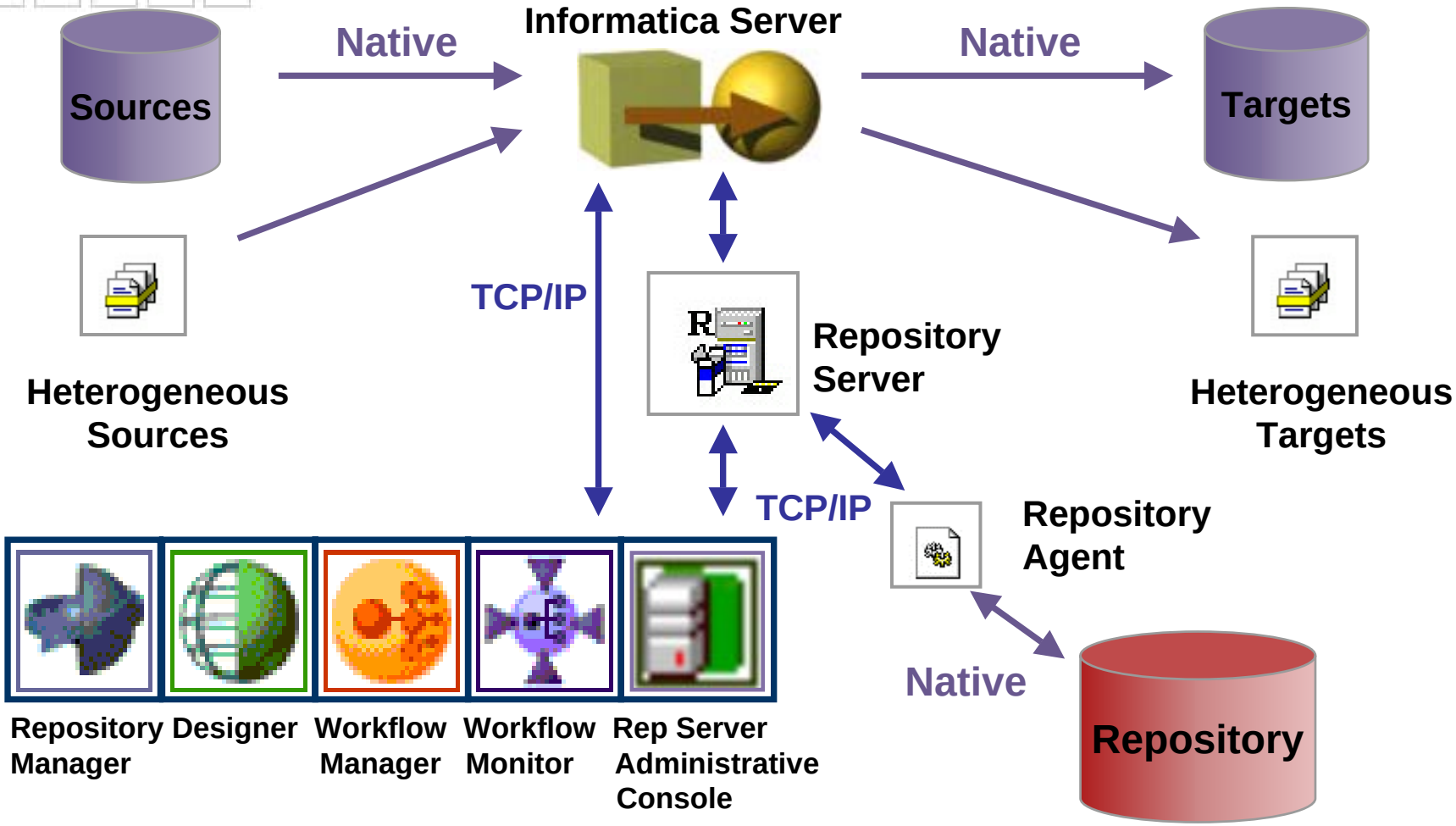
Transform

- Aggregated data
- Historical data

Extract

Load

PowerCenter 7 Architecture



Not Shown: Client ODBC Connections for Source and Target metadata

Connectivity Products for PowerCenter

These allow companies to directly source from and integrate with a variety of transactional applications and real-time services

- **PowerExchange (for mainframe, AS/400 and complex flat files)**
- **PowerConnects for:**

Transactional Applications
- Essbase
- PeopleSoft
- SAP R/3
- SAP BW
- SAS
- Siebel

Real-time Services
- JMS
- MSMQ
- MQSeries
- SAP IDOCs
- TIBCO
- WebMethods
- Web Services

- **PowerConnect SDK (available on the Informatica Developer Network)**

PowerCenter 7.1 Options

Data Profiling	Profile wizards, rules definitions, profile results tables, and standard reports
Data Cleansing	Name and address cleansing functionality, including directories for US and certain international countries
Server Grid	Server group management, automatic workflow distributio across multiple heterogeneous servers
Real-Time/WebServices	ZL Engine, always-on non-stop sessions, JMS connectivity, and real-time Web Services provider
Partitioning	Data smart parallelism, pipeline and data parallelism, partitioning
Team-Based Development	Version control, deployment groups, configuration management, automatic promotion
PowerCenter	Server engine, metadata repository, unlimited designers, workflow scheduler, all APIs and SDKs, unlimited XML an flat file sourcing and targeting, object export to XML file, LDAP authentication, role-based object-level security, metadata reporter, centralized monitoring

Watch for short virtual classroom courses on these options and XML!

INFORMATICA

Design and Execution Process

1. Create Source definition(s)
2. Create Target definition(s)
3. Create a Mapping
4. Create a Session Task
5. Create a Workflow with Task components
6. Run the Workflow and verify the results

Demonstration

INFORMATICA

Source Object Definitions

INFORMATICA

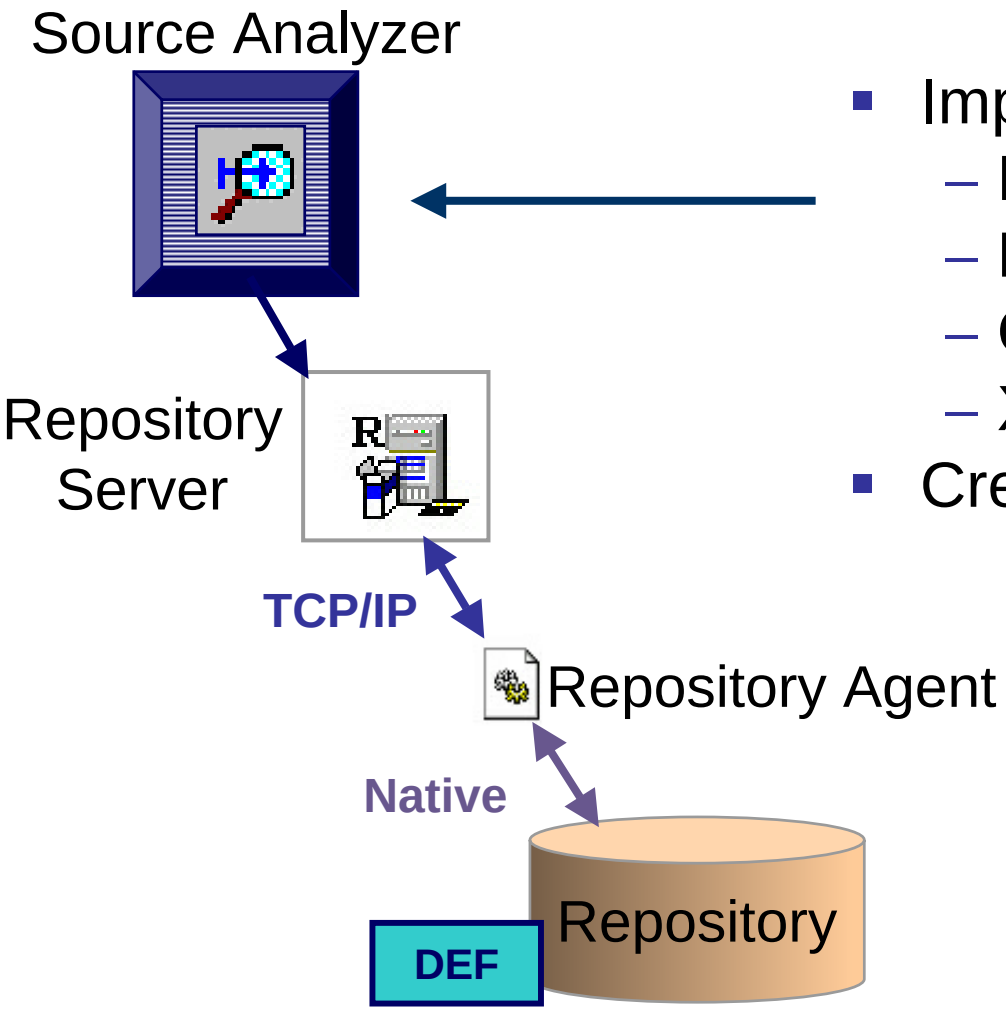
Source Object Definitions

By the end of this section you will:

- Be familiar with the Designer interface
- Be familiar with Source Types
- Be able to create Source Definitions
- Understand Source Definition properties
- Be able to use the Data Preview option

INFORMATICA

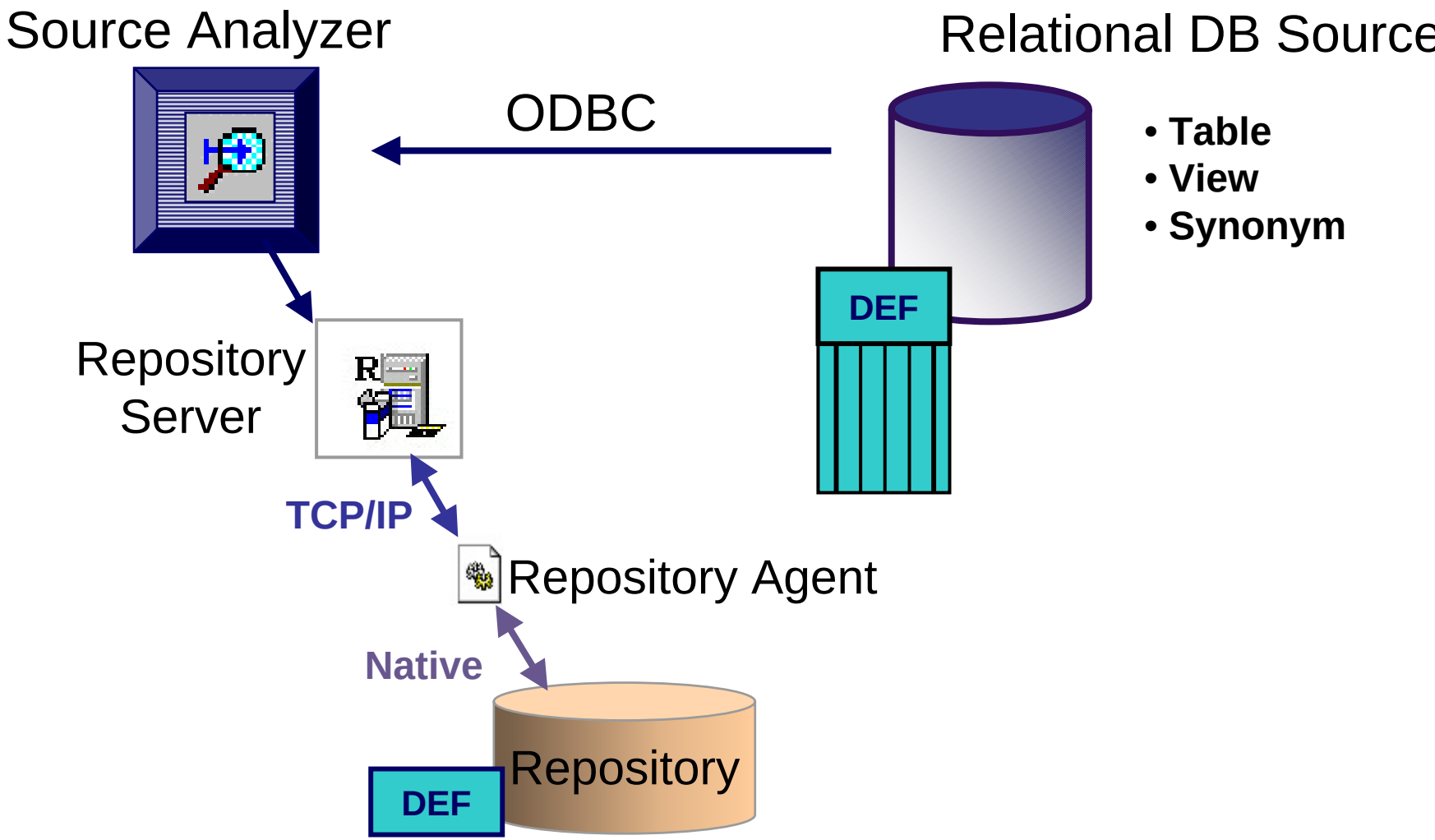
Methods of Analyzing Sources



- Import from:
  - Relational database
  - Flat file
  - COBOL file
  - XML object
- Create manually

INFORMATICA

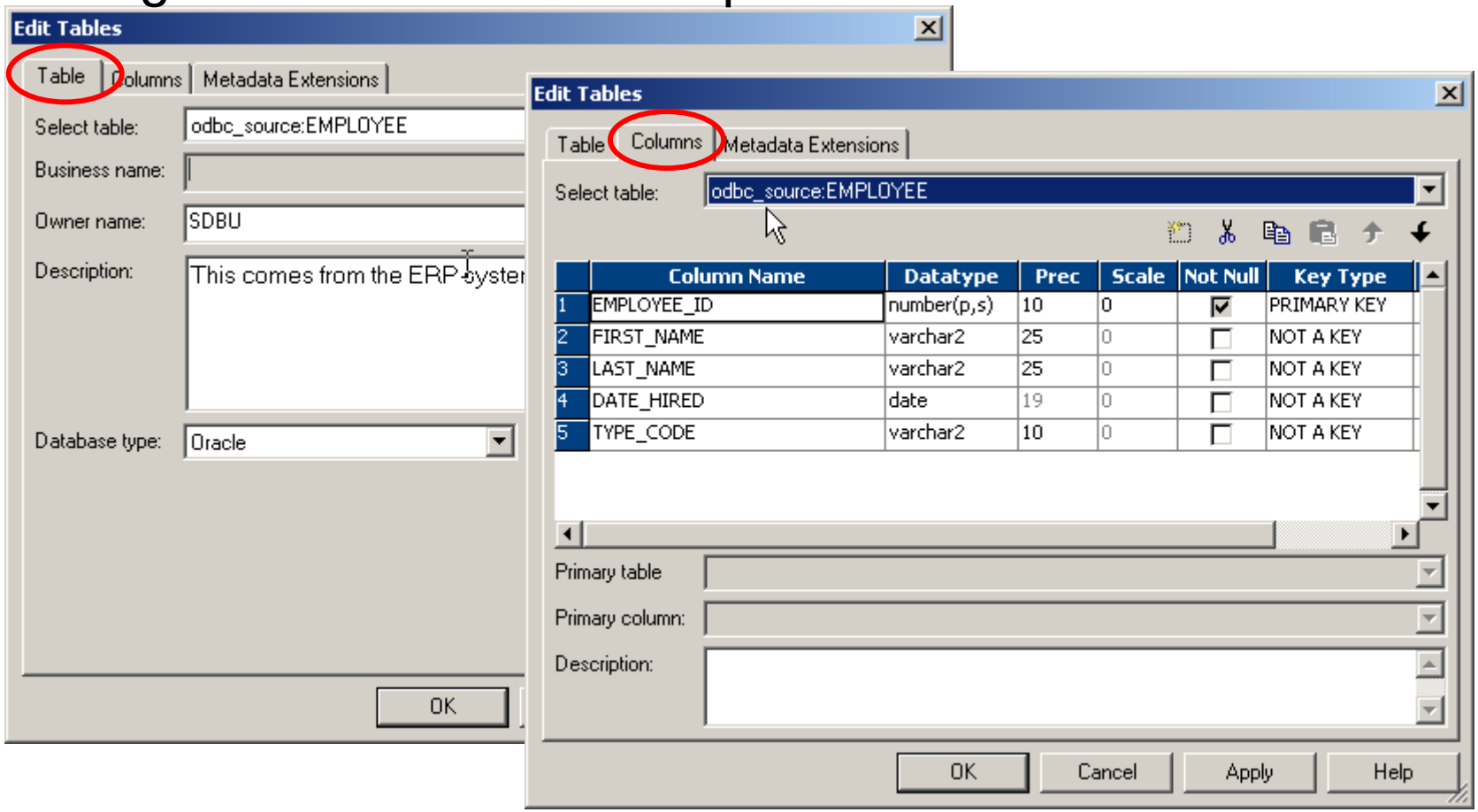
Analyzing Relational Database Sources



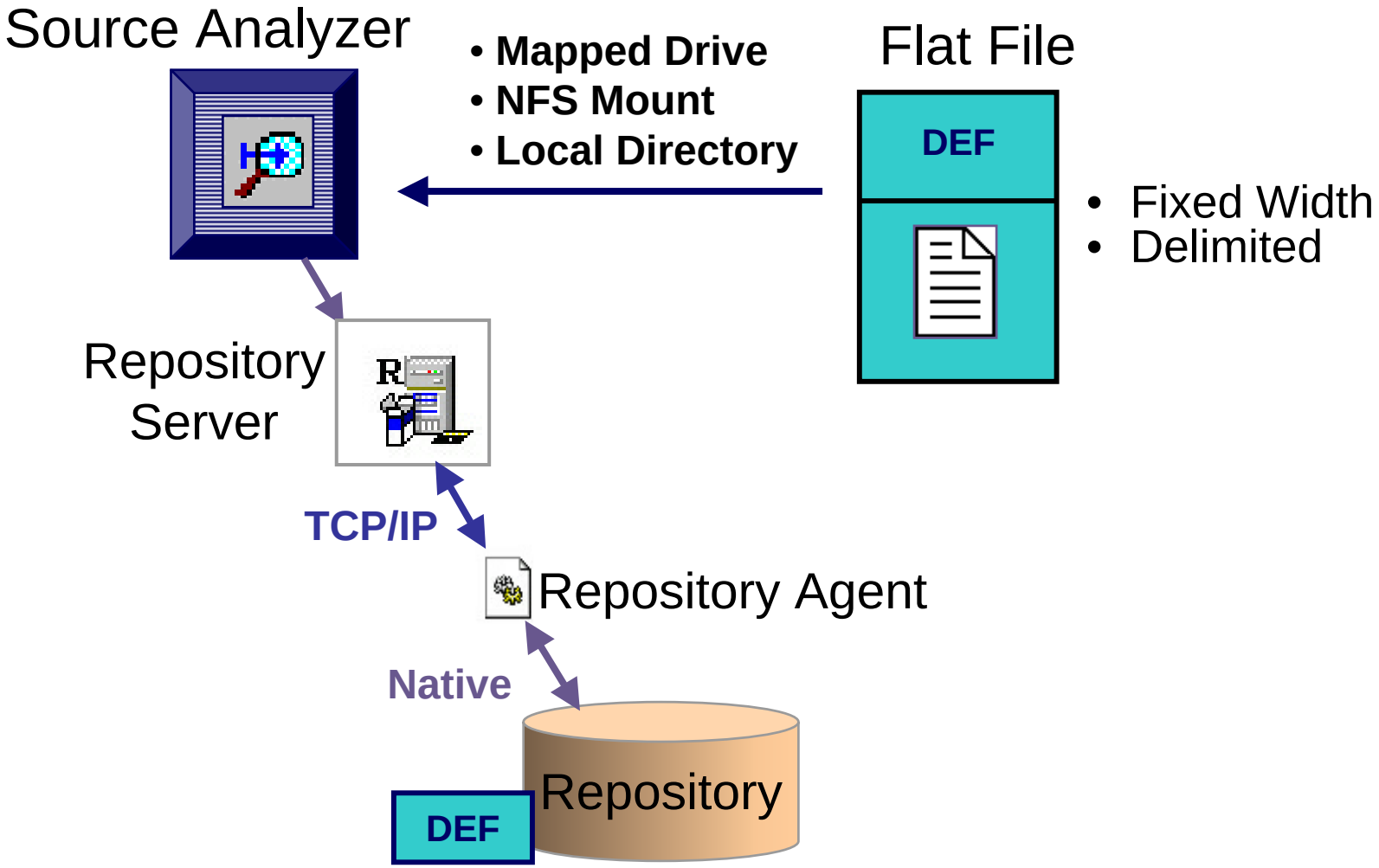
INFORMATICA

Analyzing Relational Database Sources

Editing Source Definition Properties



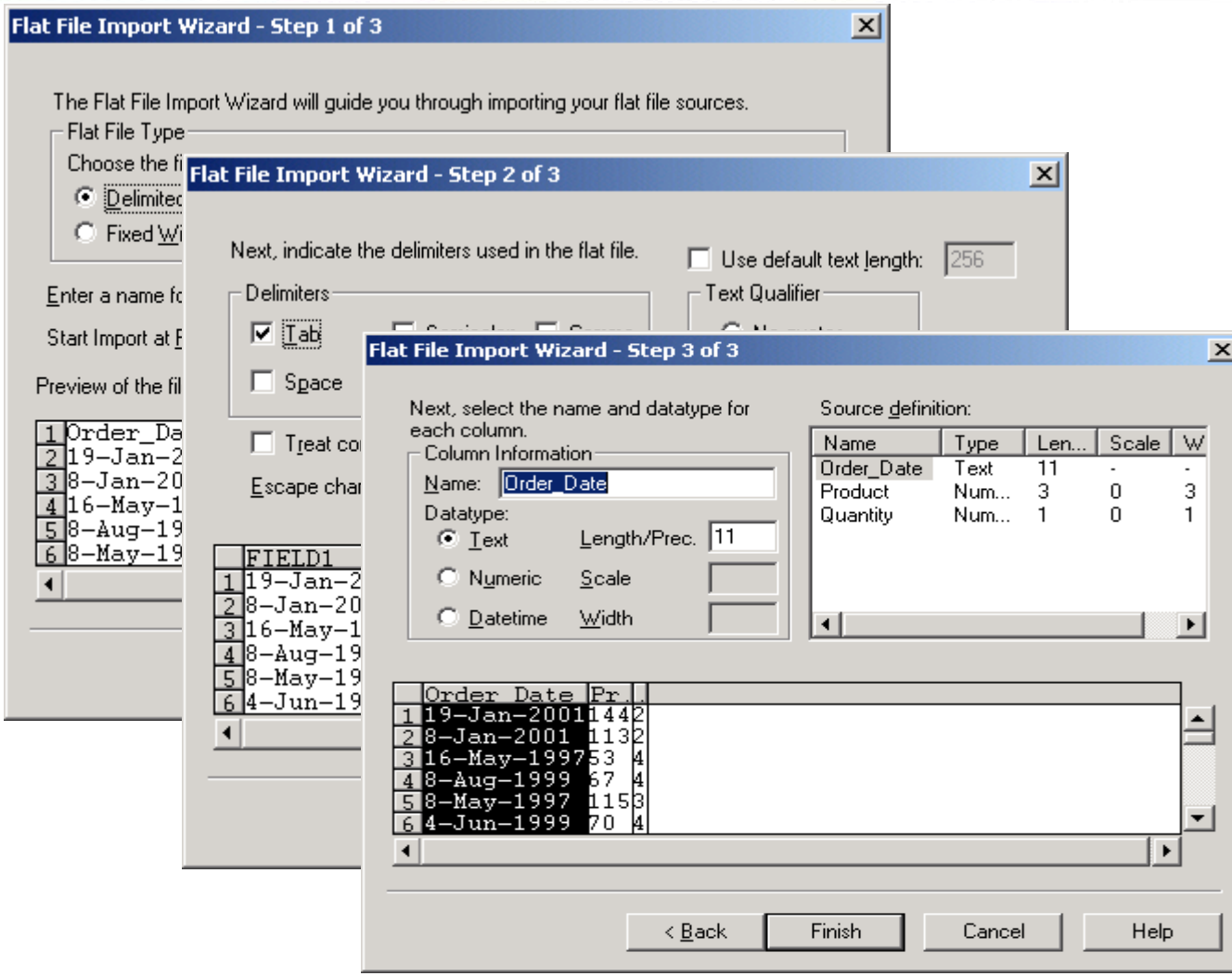
Analyzing Flat File Sources



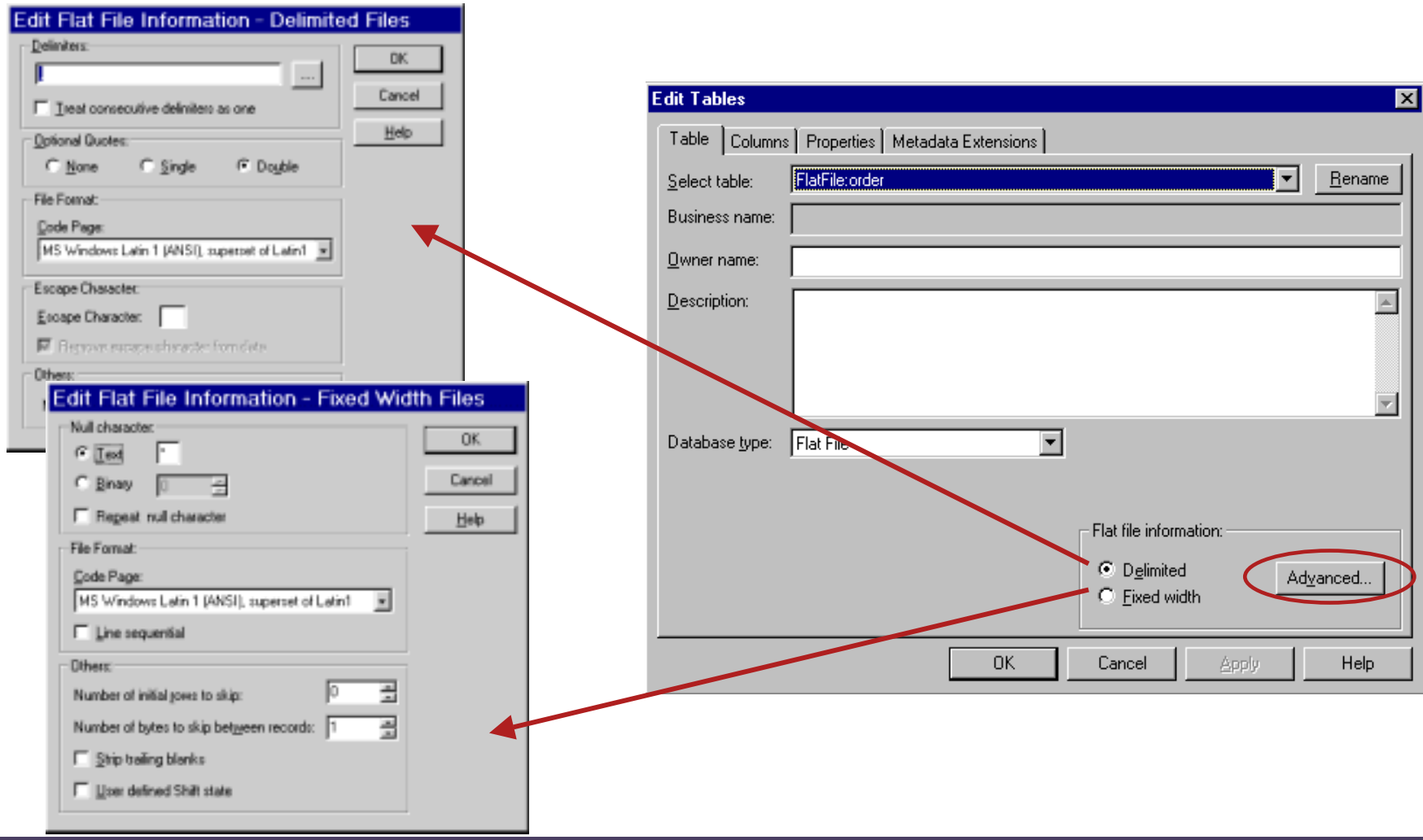
INFORMATICA

Flat File Wizard

- Three-step wizard
- Columns can be renamed within wizard
- Text, Numeric and Datetime datatypes are supported
- Wizard 'guesses' datatype

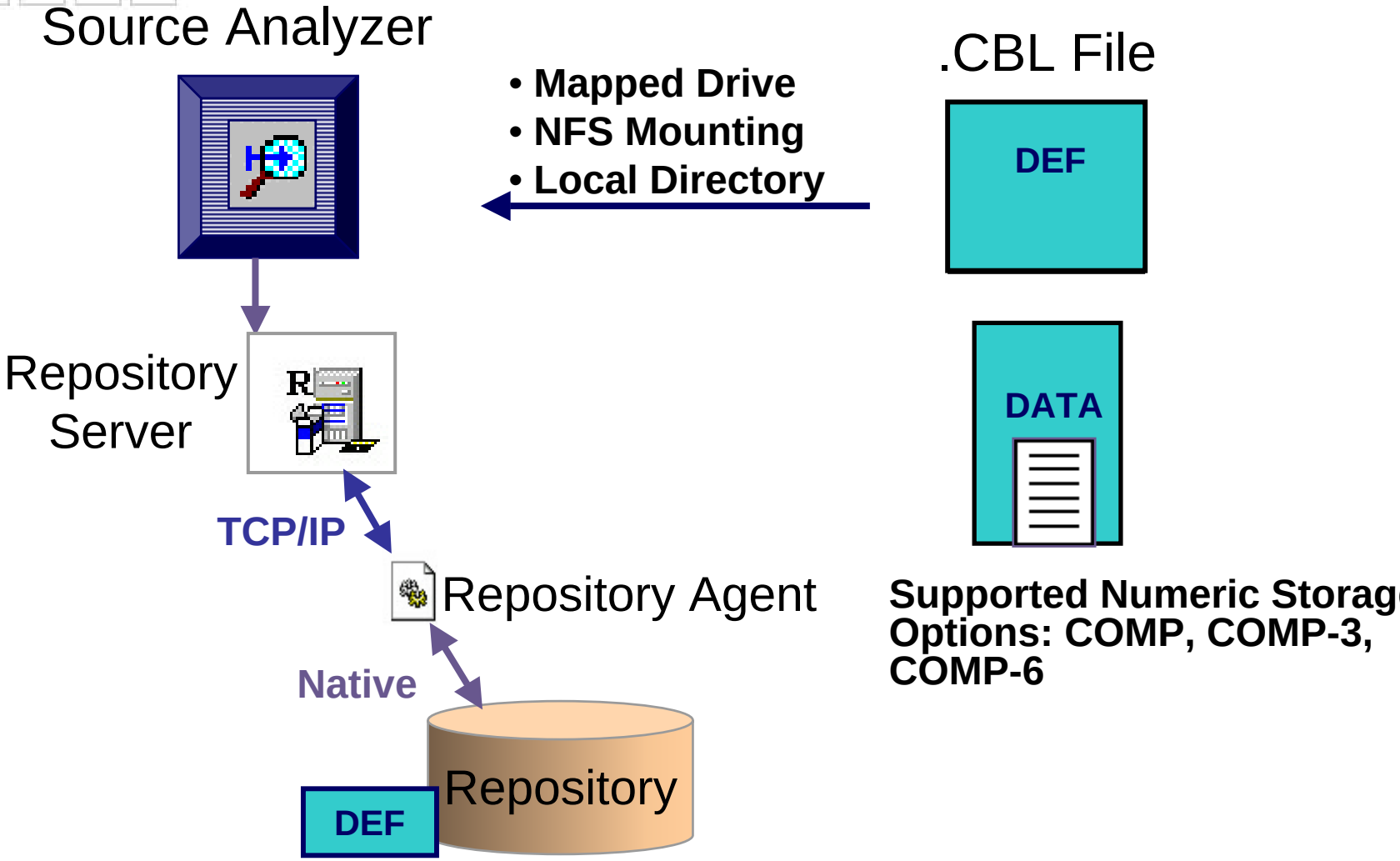


Flat File Source Properties



INFORMATICA

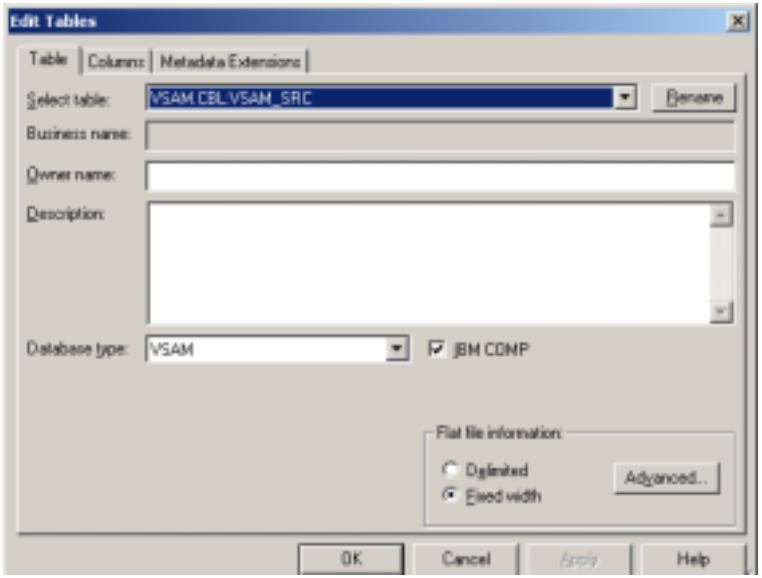
Analyzing COBOL (VSAM) Sources



Cobol (VSAM) Source Properties

VSAM_SRC (VSAM)

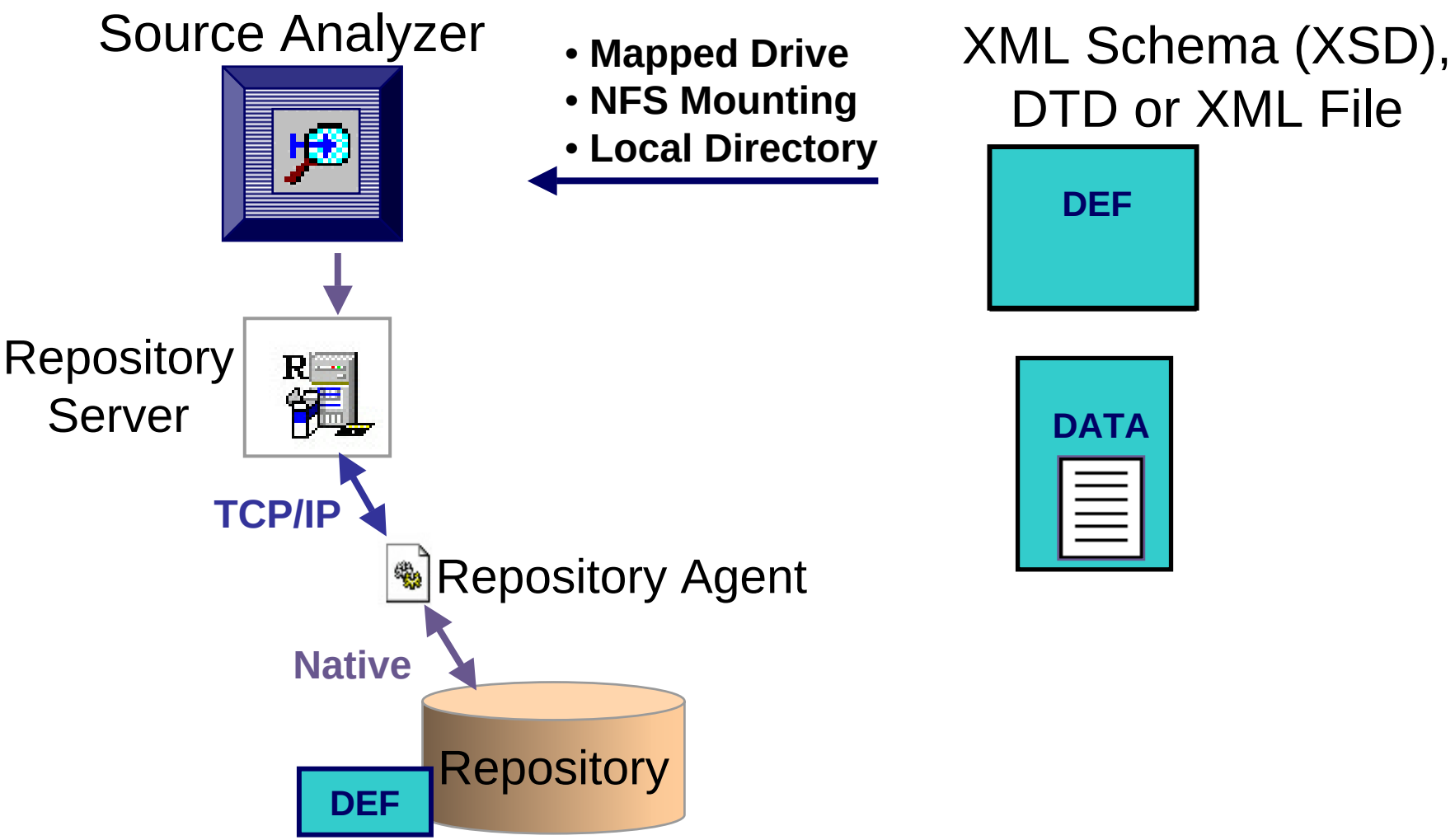
Name	Datatype	Length/Precision	Occurs
ACCOUNT1		0	0
ACCOUNT	string	10	0
OWNER	string	10	0
DEPT	number	8	0
LOCATION	string	10	0
REVIEWLIST	string	0	2
REVIEWERID	number	7	0
REVIEWERNAME	string	10	0



	POffs	PLen	Column Name	Level	Occurs	Datatype	Prec	Scale	Picture	Usage	Key Type	S	T	I	R	Redefines	Business Name
	0	72	ACCOUNT1	1	0							☐	☐	☐	☐		
	0	10	ACCOUNT	2	0	string	10	0	X(10)	DISPLAY	NOT A KEY	☐	☐	☐	☐		
	10	10	OWNER	2	0	string	10	0	X(10)	DISPLAY	NOT A KEY	☐	☐	☐	☐		
	20	8	DEPT	2	0	number	8	0	9(8)	DISPLAY	NOT A KEY	☐	☑	☐	☐		
	28	10	LOCATION	2	0	string	10	0	X(10)	DISPLAY	NOT A KEY	☐	☐	☐	☐		
	38	17	REVIEWLIST	2	2							☐	☐	☐	☐		
	38	7	REVIEWERID	3	0	number	7	0	9(7)	DISPLAY	NOT A KEY	☐	☑	☐	☐		
	45	10	REVIEWERN...	3	0	string	10	0	X(10)	DISPLAY	NOT A KEY	☐	☐	☐	☐		

INFORMATICA

Analyzing XML Sources



INFORMATICA

Data Previewer

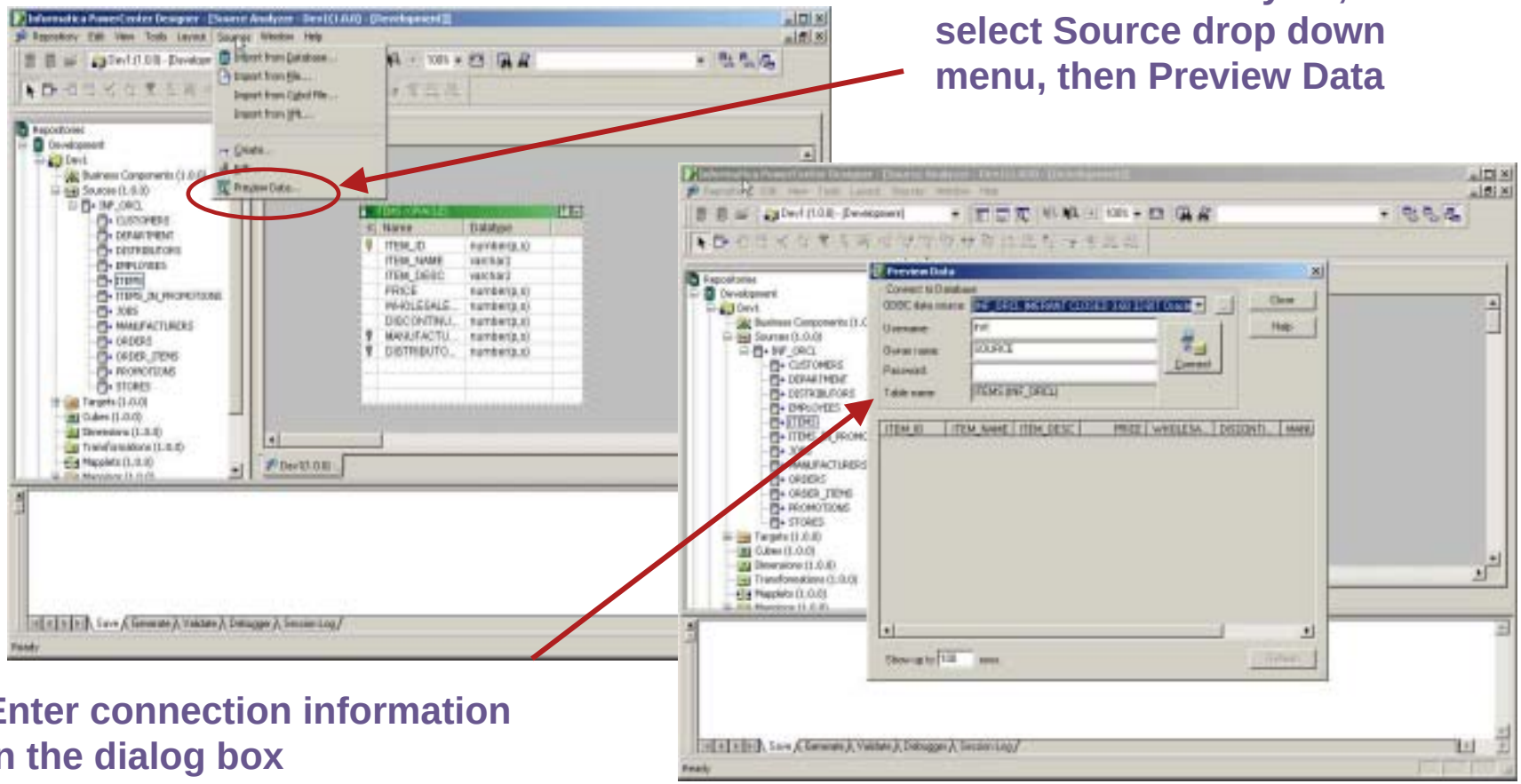
- Preview data in
  - Relational database sources
  - Flat file sources
  - Relational database targets
  - Flat file targets
- Data Preview Option is available in
  - Source Analyzer
  - Warehouse Designer
  - Mapping Designer
  - Mapplet Designer

INFORMATICA

Using Data Previewer in Source Analyzer

Data Preview Example

From Source Analyzer, select Source drop down menu, then Preview Data



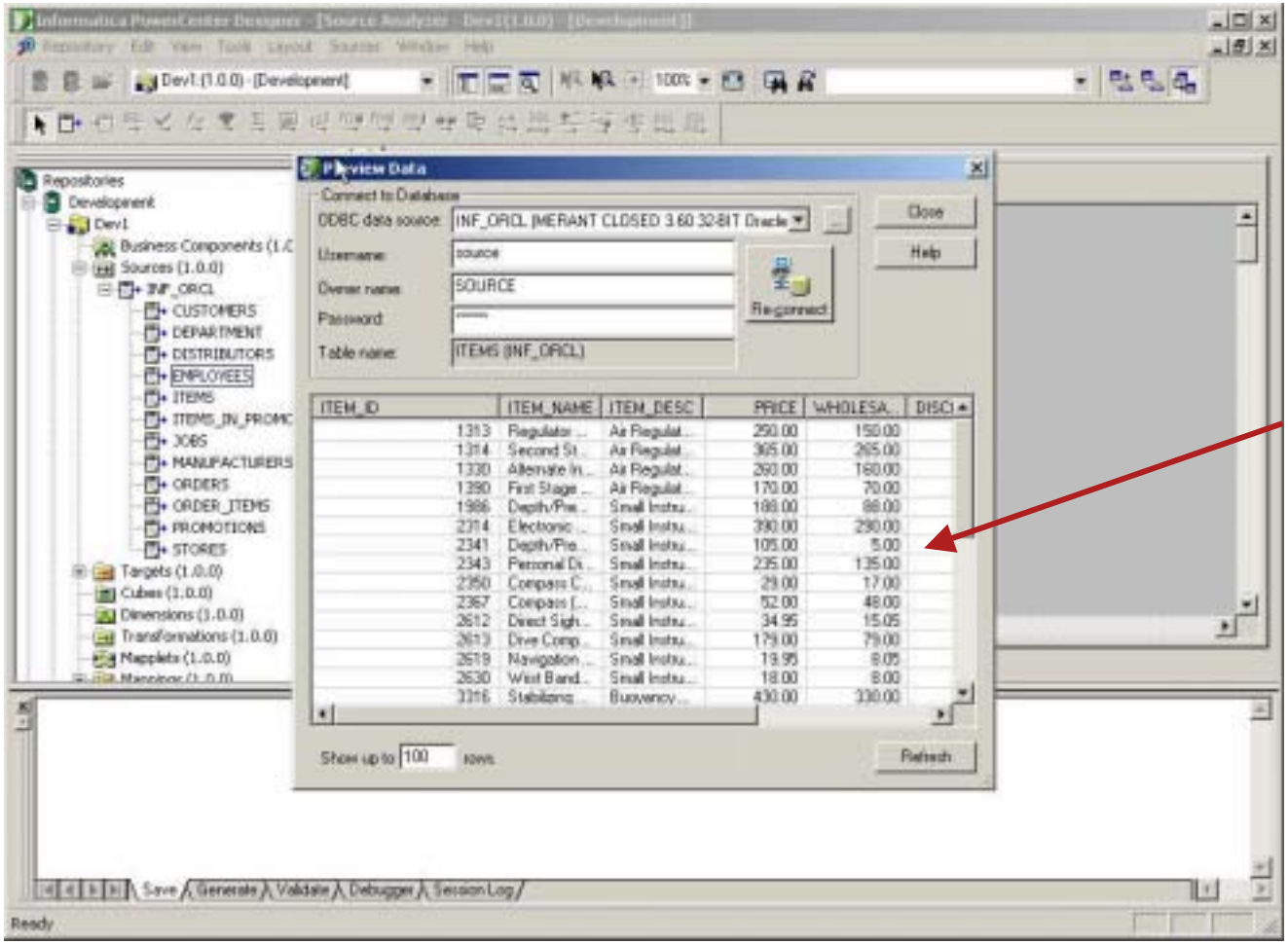
Enter connection information in the dialog box

A right mouse click on the object can also be used to preview data

INFORMATICA

Using Data Previewer in Source Analyzer

Data Preview Results



Data Display

View up to 500 rows

INFORMATICA

Metadata Extensions

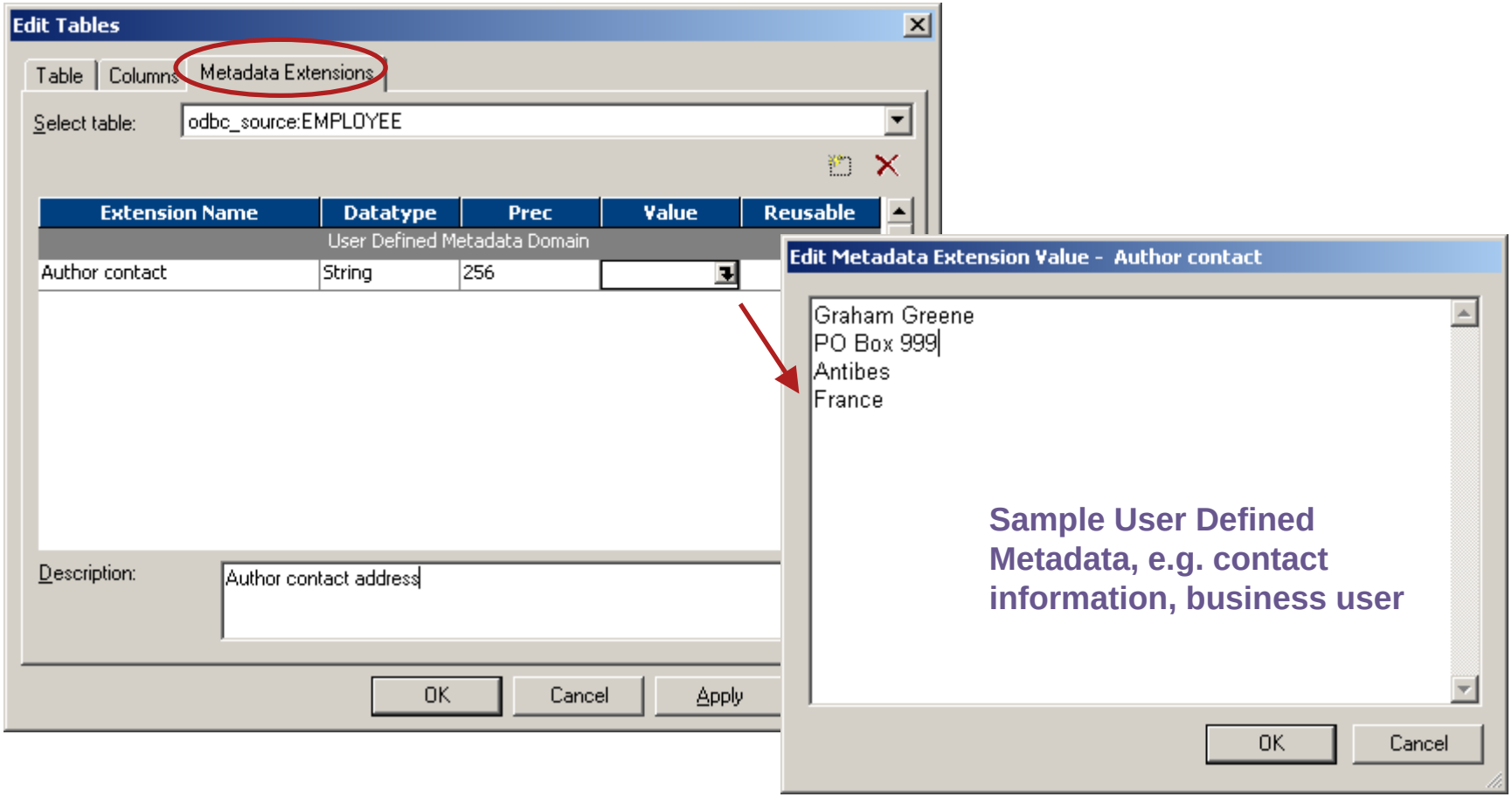
- Allows developers and partners to extend the metadata stored in the Repository
- Metadata extensions can be:
  - User-defined – PowerCenter users can define and create their own metadata
  - Vendor-defined – Third-party application vendor-created metadata lists
    - For example, applications such as Ariba or PowerConnect for Siebel can add information such as contacts, version, etc.

Metadata Extensions

- Can be reusable or non-reusable
- Can promote non-reusable metadata extensions to reusable; this is not reversible
- Reusable metadata extensions are associated with all repository objects of that object type
- A non-reusable metadata extensions is associated with a single repository object
- Administrator or Super User privileges are required for managing reusable metadata extensions

INFORMATICA

Example – Metadata Extension for a Source



Target Object Definitions

INFORMATICA

Target Object Definitions

By the end of this section you will:

- Be familiar with Target Definition types
- Know the supported methods of creating Target Definitions
- Understand individual Target Definition properties

Creating Target Definitions

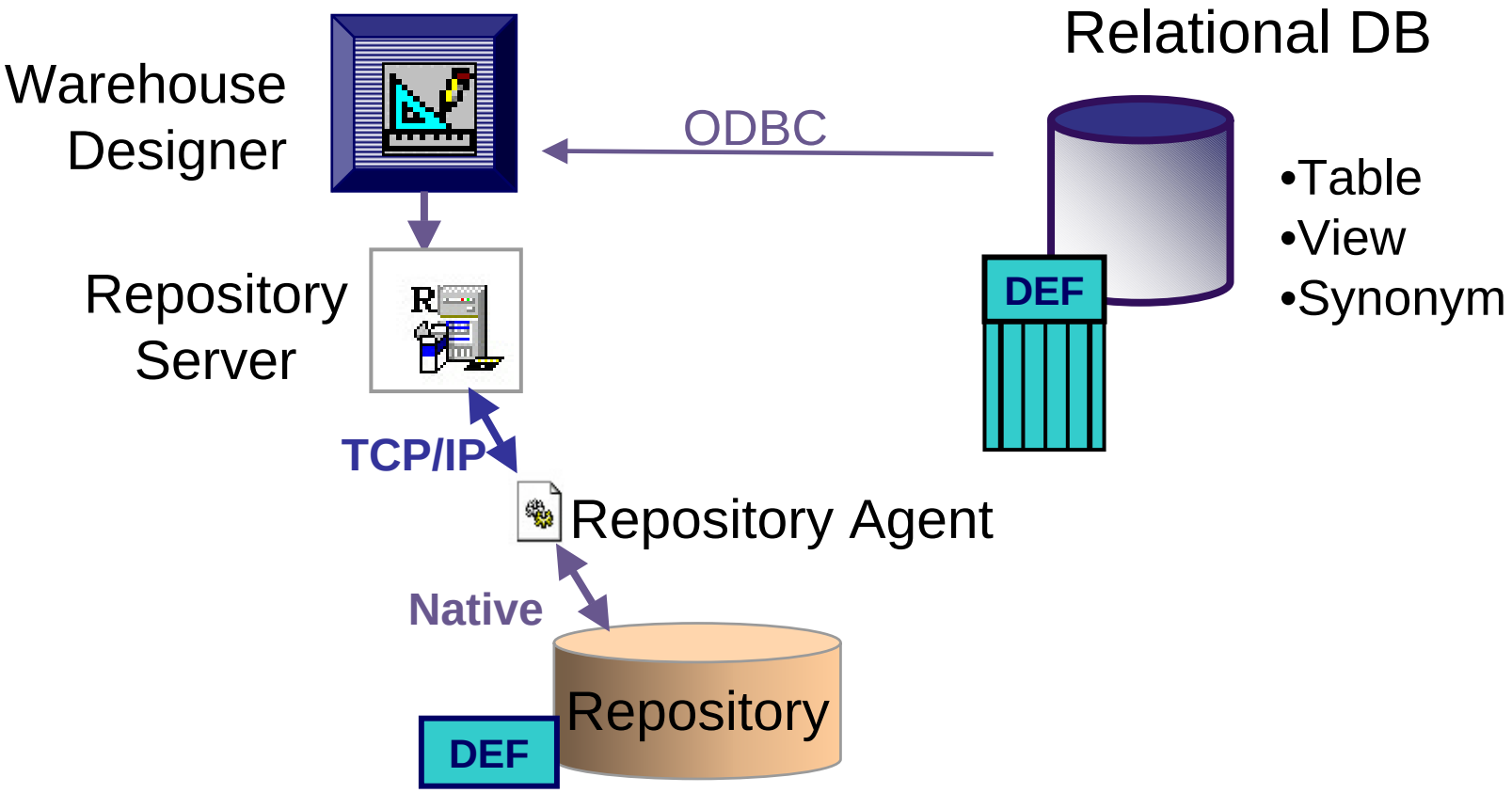
Methods of creating Target Definitions

- Import from relational database
- Import from XML object
- Create automatically from a source definition
- Create manually (flat file or relational database)

INFORMATICA

Import Definition from Relational Database

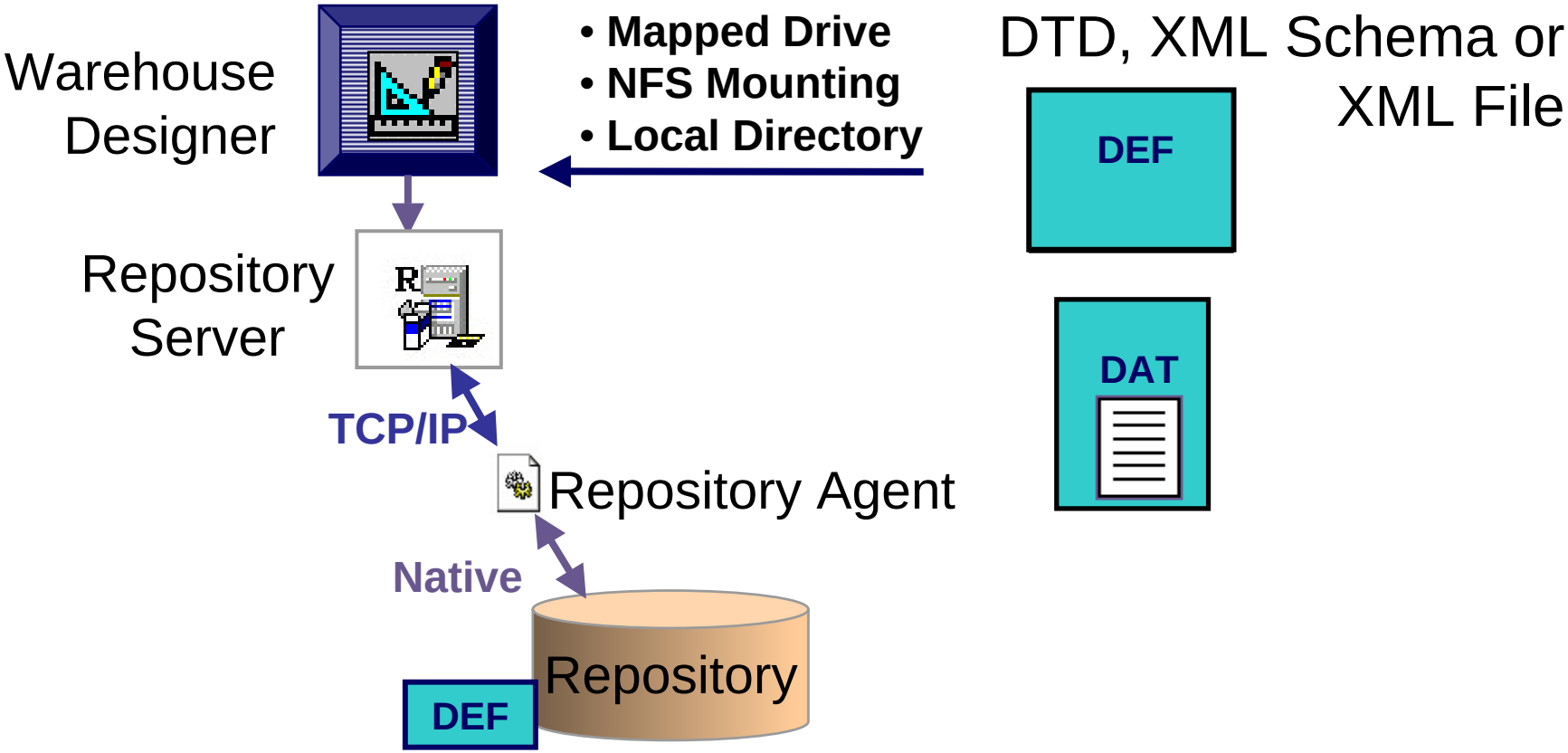
Can infer existing object definitions from a database system catalog or data dictionary



INFORMATICA

Import Definition from XML Object

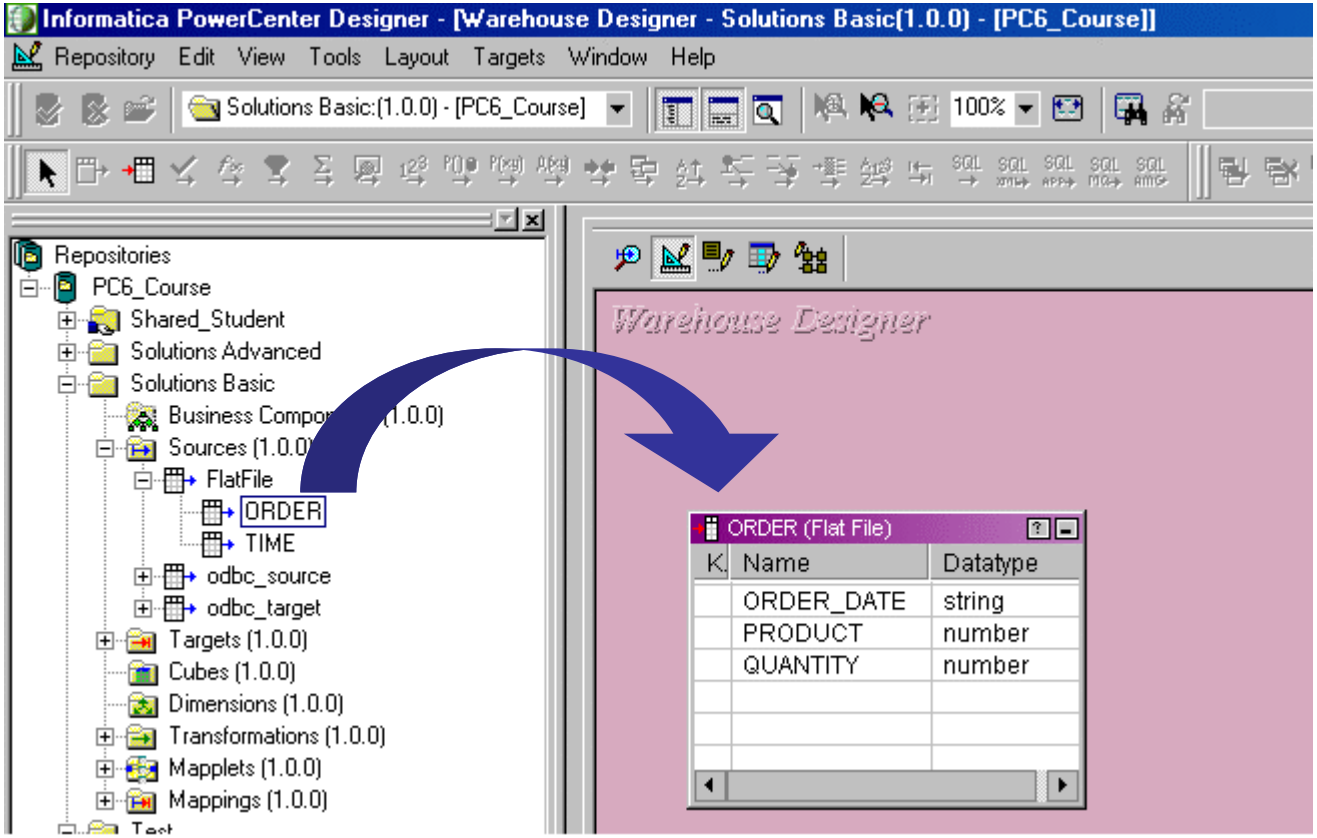
Can infer existing object definitions from a database system catalog or data dictionary



INFORMATICA

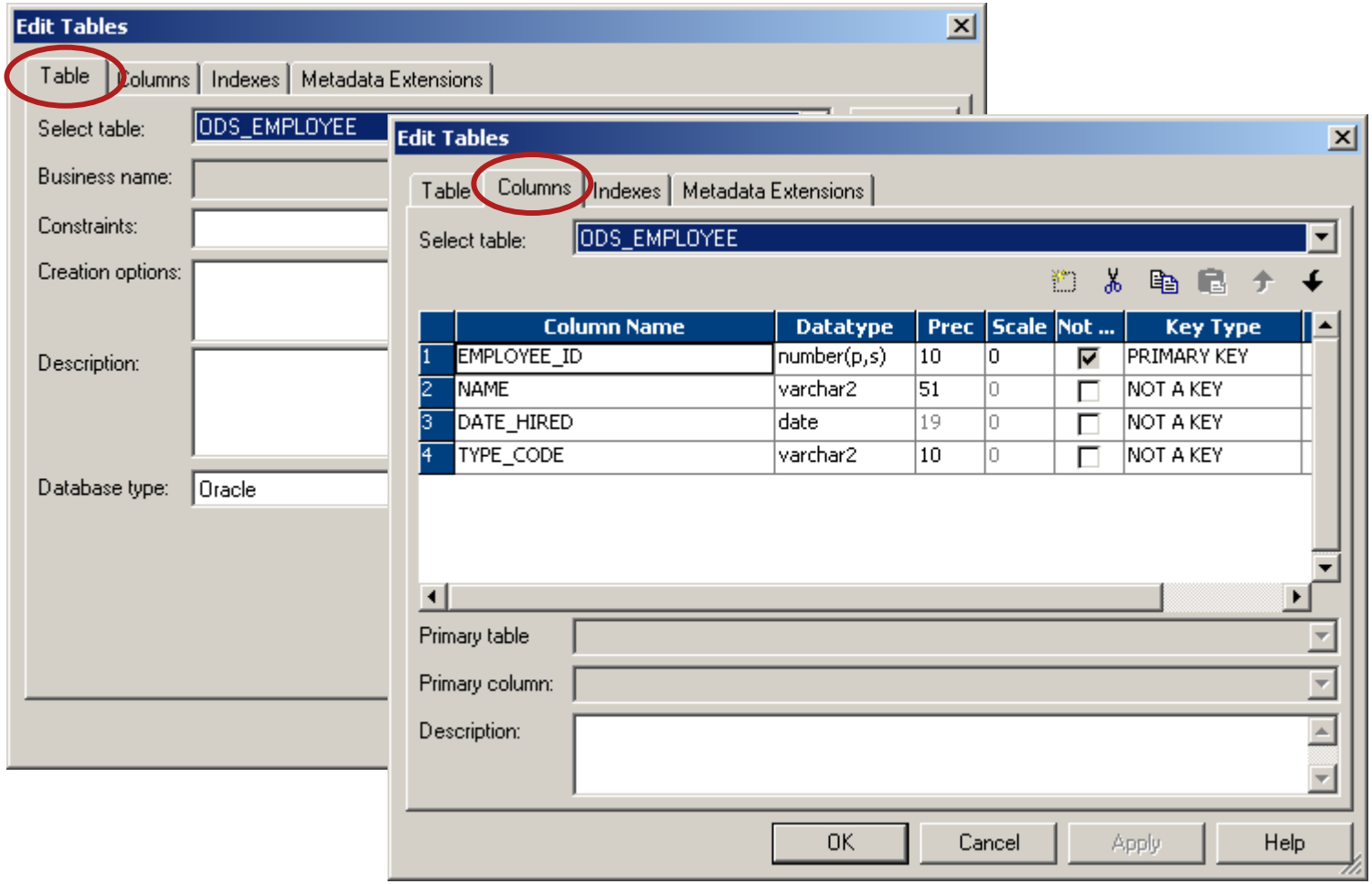
Creating Target Automatically from Source

Drag-and-drop a Source Definition into the Warehouse Designer Workspace



INFORMATICA

Target Definition Properties



INFORMATICA

Lab 1 – Define Sources and Targets

INFORMATICA

Mappings

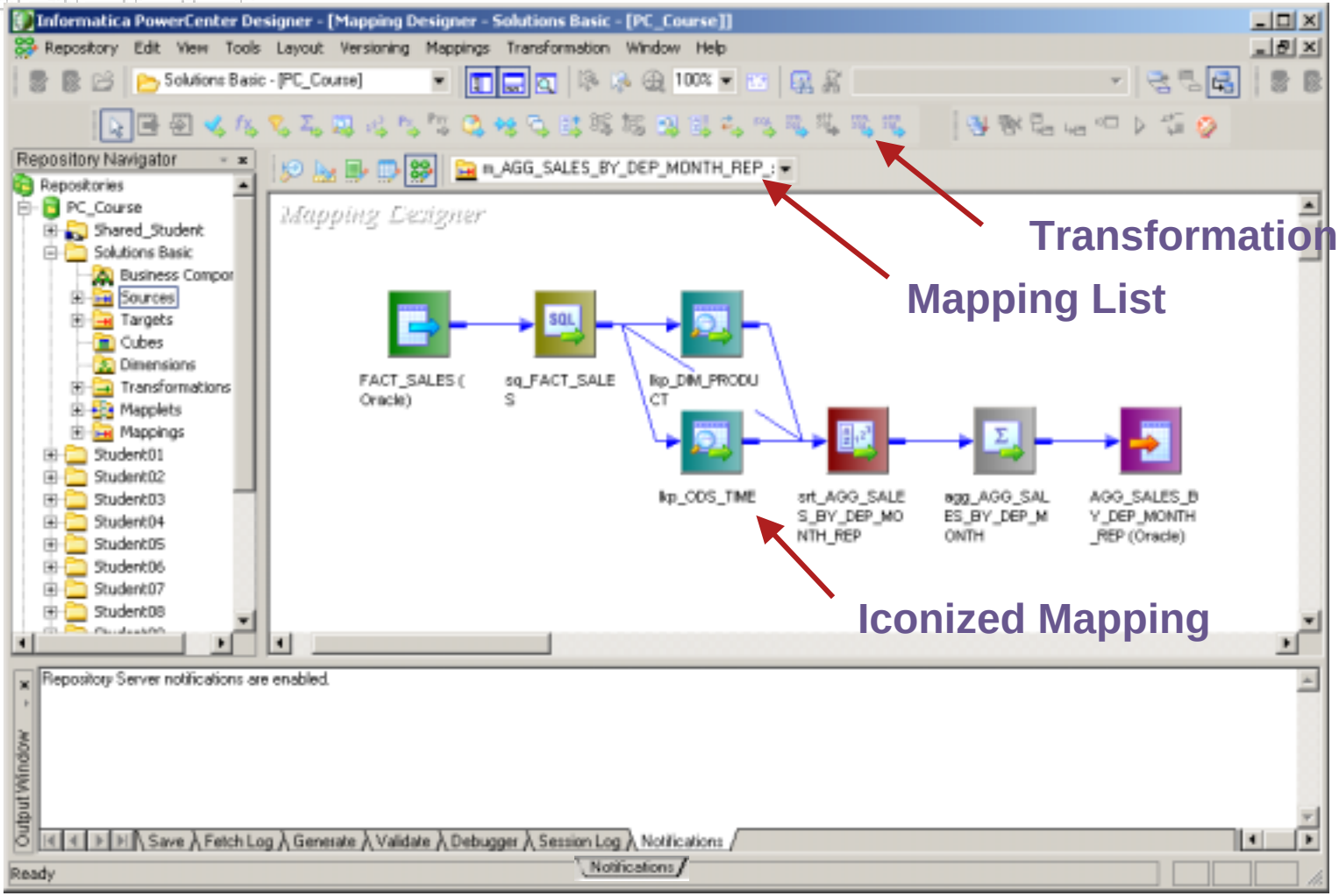
INFORMATICA

Mappings

By the end of this section you will be familiar with:

- The Mapping Designer interface
- Transformation objects and views
- Source Qualifier transformation
- The Expression transformation
- Mapping validation

Mapping Designer



INFORMATICA

Transformations Objects Used in This Class

- Source Qualifier: reads data from flat file & relational sources
- Expression: performs row-level calculations
- Filter: drops rows conditionally
- Sorter: sorts data
- Aggregator: performs aggregate calculations
- Joiner: joins heterogeneous sources
- Lookup: looks up values and passes them to other objects
- Update Strategy: tags rows for insert, update, delete, reject
- Router: splits rows conditionally
- Sequence Generator: generates unique ID values

Other Transformation Objects

- Normalizer: normalizes records from relational or VSAM sources
- Rank: filters the top or bottom range of records
- Union: merges data from multiple pipelines into one pipeline
- Transaction Control: allows user-defined commits
- Stored Procedure: calls a database stored procedure
- External Procedure : calls compiled code for each row
- Custom: calls compiled code for multiple rows
- Midstream XML Parser: reads XML from database table or message queue
- Midstream XML Generator: writes XML to database table or message queu

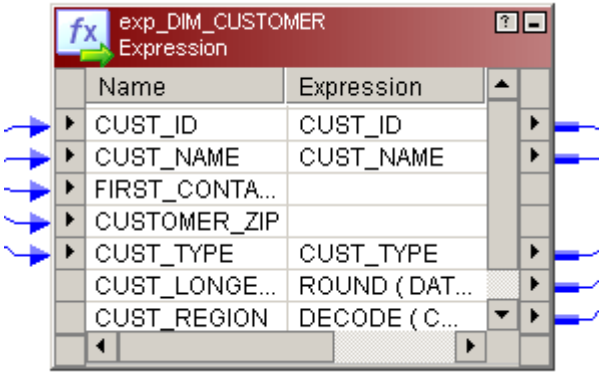
More Source Qualifiers: read from XML, message queues and applications

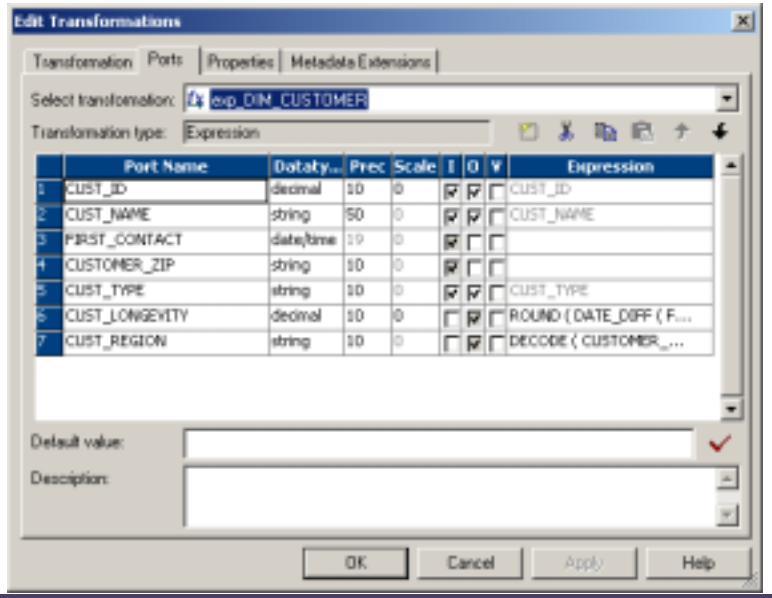
INFORMATICA

Transformation Views

A transformation has three views:

- Iconized – shows the transformation in relation to the rest of the mapping
- Normal – shows the flow of data through the transformation
- Edit – shows transformation ports (= table columns) and properties; allows editing





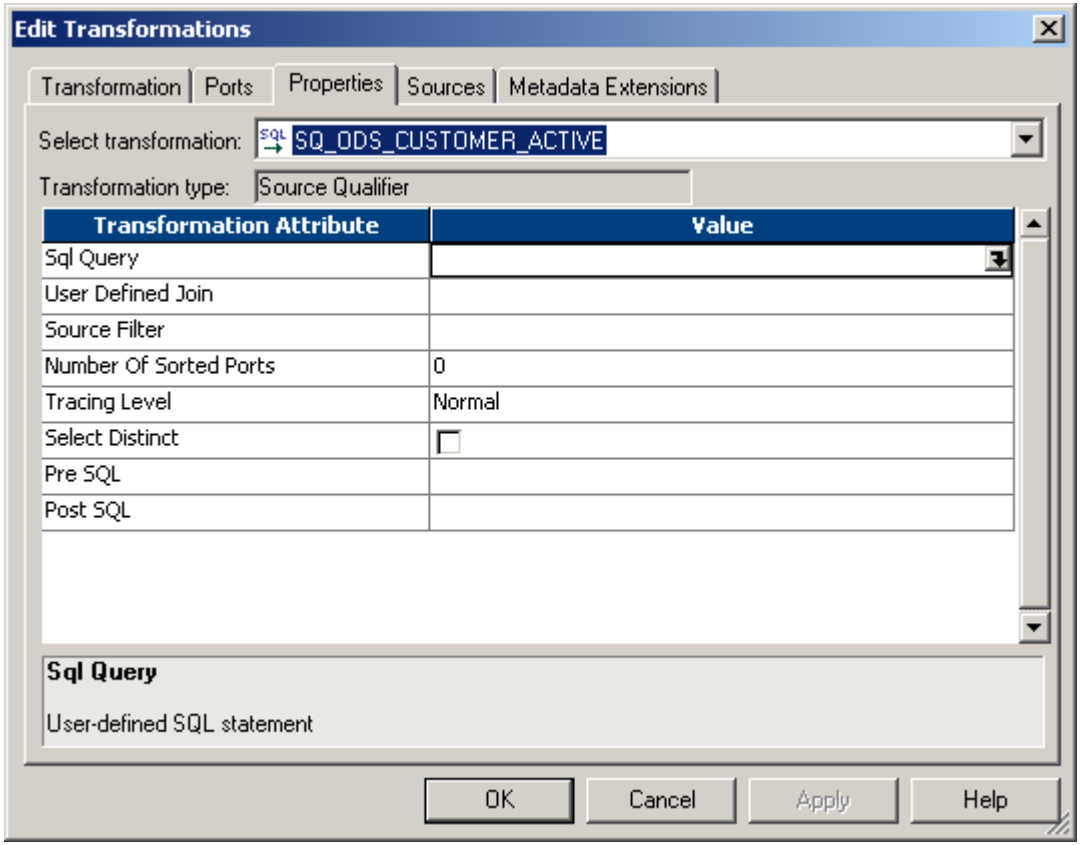
INFORMATICA

Source Qualifier Transformation

Represents the source record set queried by the Server. Mandatory in Mappings using relational or flat file sources

Ports
- All input/output

Usage
- Convert datatypes
- For relational sources:
  Modify SQL statement
  User Defined Join
  Source Filter
  Sorted ports
  Select DISTINCT
  Pre/Post SQL



INFORMATICA

Source Qualifier Properties

- User can modify SQL SELECT statement (DB sources)
- Source Qualifier can join homogenous tables
- User can modify WHERE clause
- User can modify join statement
- User can specify ORDER BY (manually or automatically)
- Pre- and post-SQL can be provided
- SQL properties do not apply to flat file sources

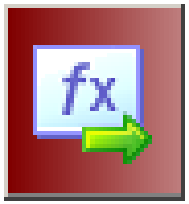
INFORMATICA

Pre-SQL and Post-SQL Rules

- Can use any command that is valid for the database type; no nested comments
- Can use Mapping Parameters and Variables in SQL executed against the source
- Use a semi-colon (;) to separate multiple statements
- Informatica Server ignores semi-colons within single quotes, double quotes or within /* ...*/
- To use a semi-colon outside of quotes or comments, ‘escape’ it with a back slash (\)

INFORMATICA

Expression Transformation



Perform calculations using non-aggregate functions (row level)

Ports
- Mixed
- Variables allowed

Create expression in an output or variable port

Usage
- Perform majority of data manipulation

Edit Transformations

Transformation | Ports | Properties | Metadata Extensions

Select transformation: exp_DIM_CUSTOMER

Transformation type: Expression

	Port Name	Dataty...	Prec	Scale	I	O	V	Expression
1	CUST_ID	decimal	10	0	☑	☑	☐	CUST_ID
2	CUST_NAME	string	50	0	☑	☑	☐	CUST_NAME
3	FIRST_CONTACT	date/time	19	0	☑	☐	☐	
4	CUSTOMER_ZIP	string	10	0	☑	☐	☐	
5	CUST_TYPE	string	10	0	☑	☑	☐	CUST_TYPE
6	CUST_LONGEVITY	decimal	10	0	☐	☑	☐	ROUND (DATE_DIFF(FIR
7	CUST_REGION	string	10	0	☐	☑	☐	DECODE (CUSTOMER ...

Default value: ERROR('transformation error')

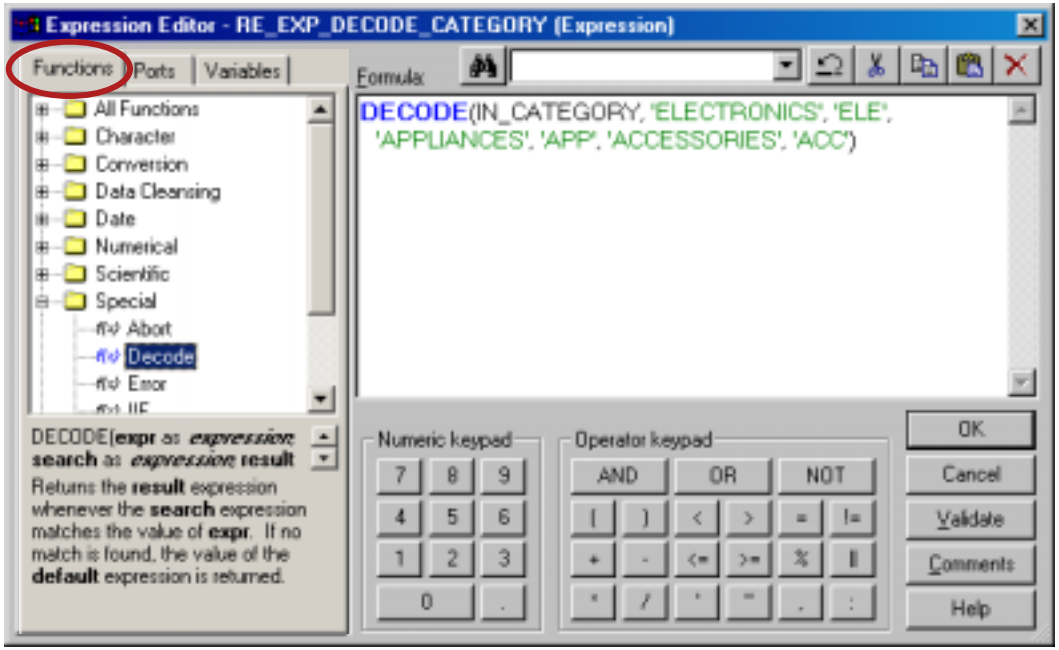
Description:

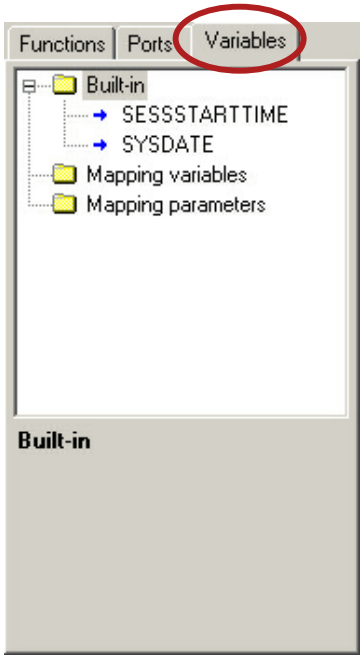
Click here to invoke the Expression Editor

OK | Cancel | Apply | Help

Expression Editor

- **An expression formula is a calculation or conditional statement for a specific port in a transformation**
- **Performs calculation based on ports, functions, operators, variables, constants and return values from other transformations**





INFORMATICA

Expression Validation

The Validate or ‘OK’ button in the Expression Editor will:

- Parse the current expression
  - Remote port searching (resolves references to ports in other transformations)
- Parse default values
- Check spelling, correct number of arguments in functions, other syntactical errors

INFORMATICA

Informatica Functions – Character

ASCII
CHR
CHRCODE
CONCAT
INITCAP
INSTR
LENGTH
LOWER
LPAD
LTRIM
REPLACECHR
REPLACESTR
RPAD
RTRIM
SUBSTR
UPPER

Character Functions

- Used to manipulate character data
- CHRCODE returns the numeric value (ASCII or Unicode) of the first character of the string passed to this function
- CONCAT is for backward compatibility only. Use || instead

Informatica Functions – Conversion

TO_CHAR (numeric)
TO_DATE
TO_DECIMAL
TO_FLOAT
TO_INTEGER

Conversion Functions

- Used to convert datatypes

Informatica Functions – Data Cleansing

INSTR
IS_DATE
IS_NUMBER
IS_SPACES
ISNULL
LTRIM
METAPHONE
REPLACECHR
REPLACESTR
RTRIM
SOUNDEX
SUBSTR
TO_CHAR
TO_DATE
TO_DECIMAL
TO_FLOAT
TO_INTEGER

- Used to process data during data cleansing
- METAPHONE and SOUNDEX create indexes based on English pronunciation (2 different standards)

INFORMATICA

Informatica Functions – Date

ADD_TO_DATE
DATE_COMPARE
DATE_DIFF
GET_DATE_PART
LAST_DAY
ROUND (Date)
SET_DATE_PART
TO_CHAR (Date)
TRUNC (Date)

Date Functions

- Used to round, truncate, or compare dates; extract one part of a date; or perform arithmetic on a date
- To pass a string to a date function, first use the TO_DATE function to convert it to an date/time datatype

INFORMATICA

Informatica Functions – Numerical and Scientific

ABS
CEIL
CUME
EXP
FLOOR
LN
LOG
MOD
MOVINGAVG
MOVINGSUM
POWER
ROUND
SIGN
SQRT
TRUNC

Numerical Functions

- Used to perform mathematical operations on numeric data

Scientific Functions

- Used to calculate geometric values of numeric data

COS
COSH
SIN
SINH
TAN
TANH

INFORMATICA

Informatica Functions – Special and Test

Special Functions

- Used to handle specific conditions within a session; search for certain values; test conditional statements

ABORT
DECODE
ERROR
IIF
LOOKUP

IIF(Condition,True,False)

Test Functions

IS_DATE
IS_NUMBER
IS_SPACES
ISNULL

- Used to test if a lookup result is null
- Used to validate data

INFORMATICA

Variable Ports

- Use to simplify complex expressions
  - e.g. create and store a depreciation formula to be referenced more than once
- Use in another variable port or an output port expression
- Local to the transformation (a variable port cannot also be an input or output port)
- Available in the Expression, Aggregator and Rank transformations

	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	DATE_ENTERED	date/time	19	0	☑	☐	☐	
2	MONTH	integer	10	0	☐	☐	☑	GET_DATE_PART(DATE_ENTERED,'MM')
3	QUANTITY	integer	10	0	☑	☐	☐	
4	PRICE1	decimal	10	2	☑	☐	☐	
5	DISCOUNT	decimal	10	2	☑	☐	☐	
6	Q1_SALES	decimal	10	2	☐	☑	☐	SUM(QUANTITY * PRICE1 - DISCOUNT,MONTH = 1 OR MONTH = 2 OR MONTH = 3)
7	Q2_SALES1	decimal	10	2	☐	☑	☐	SUM(QUANTITY * PRICE1 - DISCOUNT,MONTH = 4 OR MONTH = 5 OR MONTH = 6)
8	Q3_SALES2	decimal	10	2	☐	☑	☐	SUM(QUANTITY * PRICE1 - DISCOUNT,MONTH = 7 OR MONTH = 8 OR MONTH = 9)
9	Q4_SALES3	decimal	10	2	☐	☑	☐	SUM(QUANTITY * PRICE1 - DISCOUNT,MONTH = 10 OR MONTH = 11 OR MONTH = 12)

INFORMATICA

Variable Ports (cont'd)

- Use for temporary storage
- Variable Ports can remember values across rows; useful for comparing values
- Variables are initialized (numeric to 0, string to "") when the Mapping logic is processed
- Variables Ports are <u>not</u> visible in Normal view, only in Edit view

Edit Transformations

Transformation | Ports | Properties | Metadata Extensions

Select transformation: exp_COUNT_CUST_STATES

Transformation type: Expression

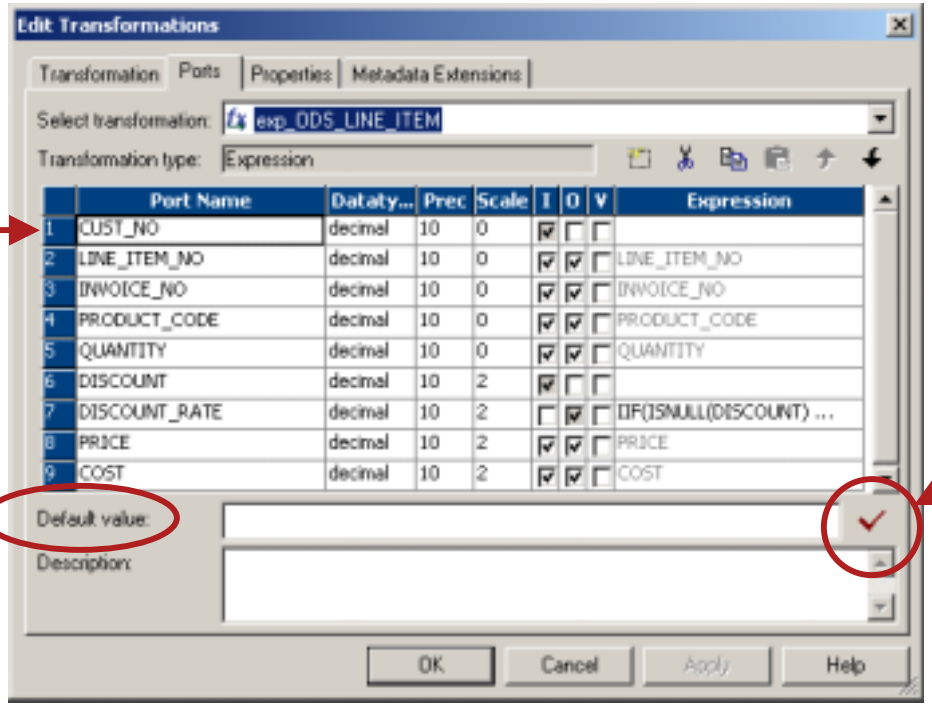
	Port Name	Datatype	Prec	Scale	I	O	V	Expression
1	STATE	string	2	0	☑	☑	☐	STATE
2	STATE_COUNTER	integer	10	0	☐	☐	☑	IIF(PREVIOUS_STATE = STATE, STATE_COUNTER + 1, 1)
3	PREVIOUS_STATE	string	2	0	☐	☐	☑	STATE
4	STATE_COUNTER_OUT	integer	10	0	☐	☑	☐	STATE_COUNTER_OUT

Default Values – Two Usages

- For input and I/O ports, default values are used to *replace* null values
- For output ports, default values are used to handle transformation calculation errors (not-null handling)

Selected port

Default value for the selected port



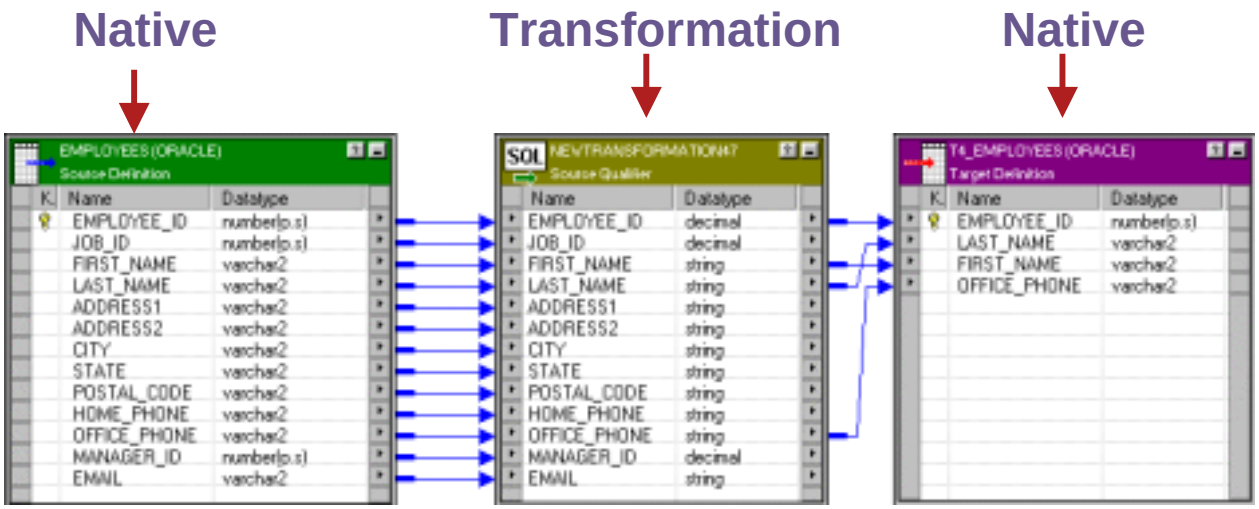
Validate the default value expression

ISNULL function is not required

INFORMATICA

Informatica Datatypes

NATIVE DATATYPES	TRANSFORMATION DATATYPES
Specific to the source and target database types	PowerCenter internal datatypes based on UCS-2
Display in source and target tables within Mapping Designer	Display in transformations within Mapping Designer



Transformation datatypes allow *mix and match* of source and target database types
When connecting ports, native and transformation datatypes must be compatible (or must be explicitly converted)

INFORMATICA

Datatype Conversions within PowerCenter

- Data can be converted from one datatype to another by:
  - Passing data between ports with different datatypes
  - Passing data from an expression to a port
  - Using transformation functions
  - Using transformation arithmetic operators
- Only conversions supported are:
  - Numeric datatypes ↔ Other numeric datatypes
  - Numeric datatypes ↔ String
  - Date/Time ↔ Date or String
- For further information, see the PowerCenter Client Help > Index > port-to-port data conversion

INFORMATICA

Mapping Validation

INFORMATICA

Connection Validation

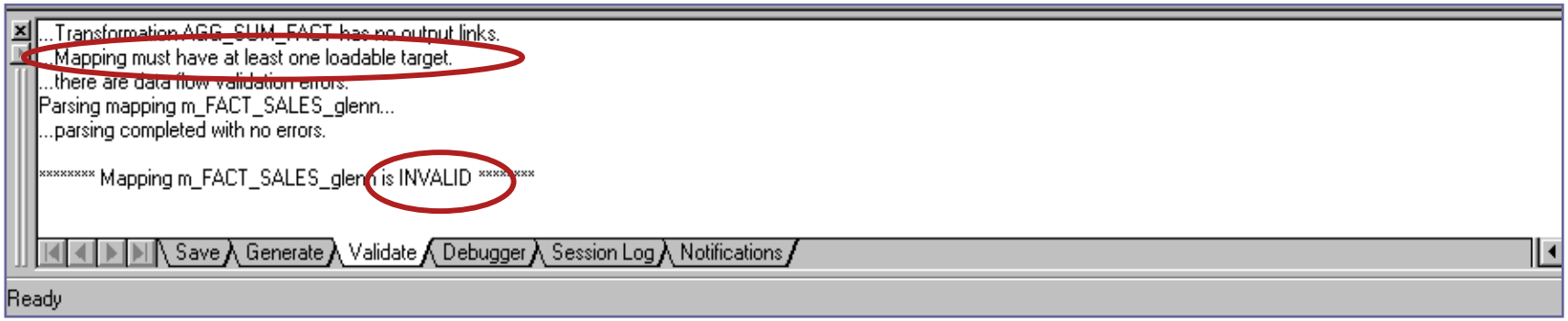
Examples of invalid connections in a Mapping:

- Connecting ports with incompatible datatypes
- Connecting output ports to a Source
- Connecting a Source to anything but a Source Qualifier or Normalizer transformation
- Connecting an output port to an output port or an input port to another input port

INFORMATICA

Mapping Validation

- Mappings must:
  - Be valid for a Session to run
  - Be end-to-end complete and contain valid expressions
  - Pass all data flow rules
- Mappings are always validated when saved; can be validated without being saved
- Output Window displays reason for invalidity



INFORMATICA

Lab 2 – Create a Mapping

INFORMATICA

Workflows

INFORMATICA

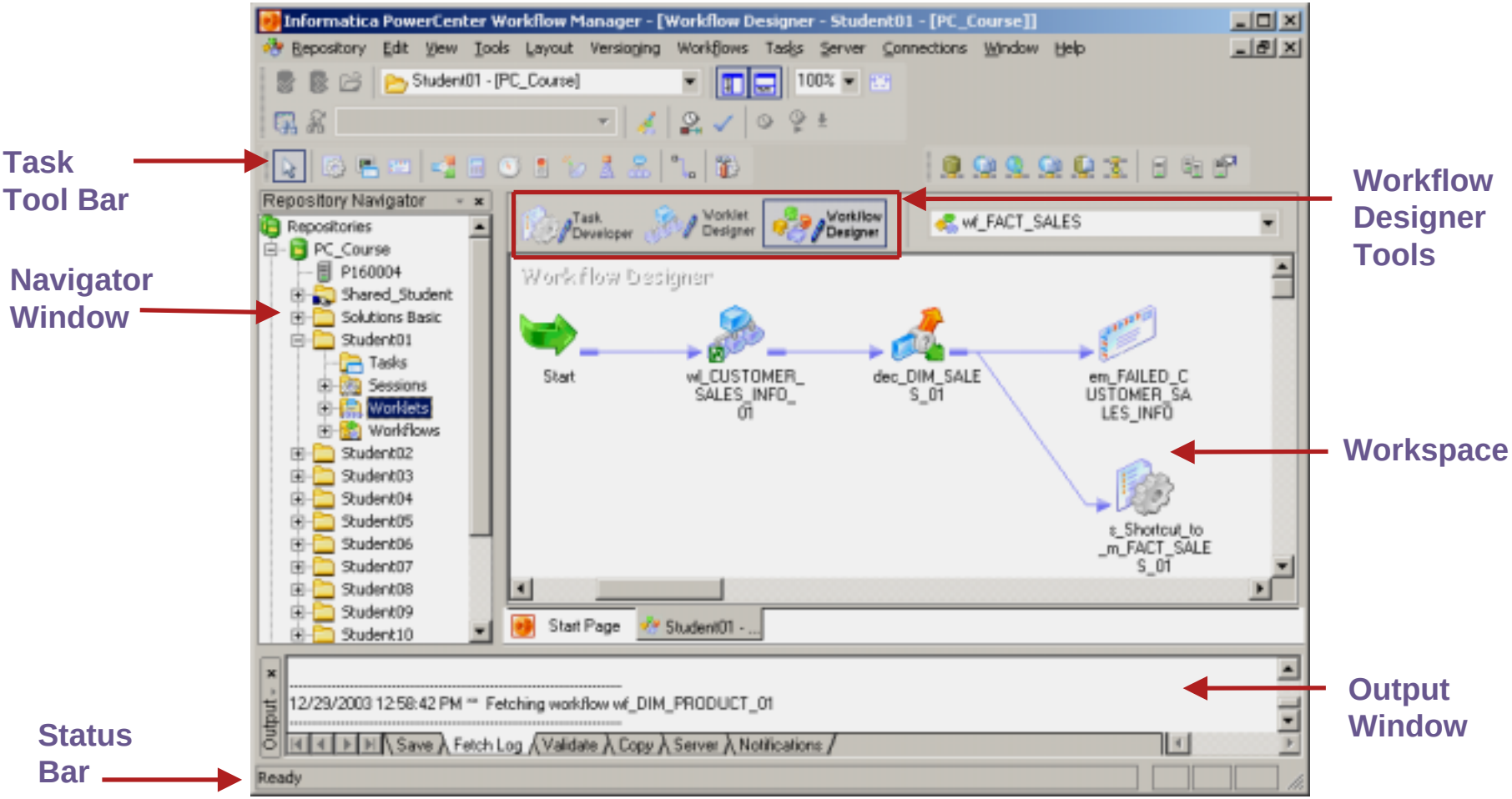
Workflows

By the end of this section, you will be familiar with:

- The Workflow Manager GUI interface
- Creating and configuring Workflows
  - Workflow properties
  - Workflow components
  - Workflow tasks

INFORMATICA

Workflow Manager Interface



Workflow Manager Tools

- **Workflow Designer**

  - Maps the execution order and dependencies of Sessions, Tasks and Worklets, for the Informatica Server

- **Task Developer**

  - Create Session, Shell Command and Email tasks
  - Tasks created in the Task Developer are reusable

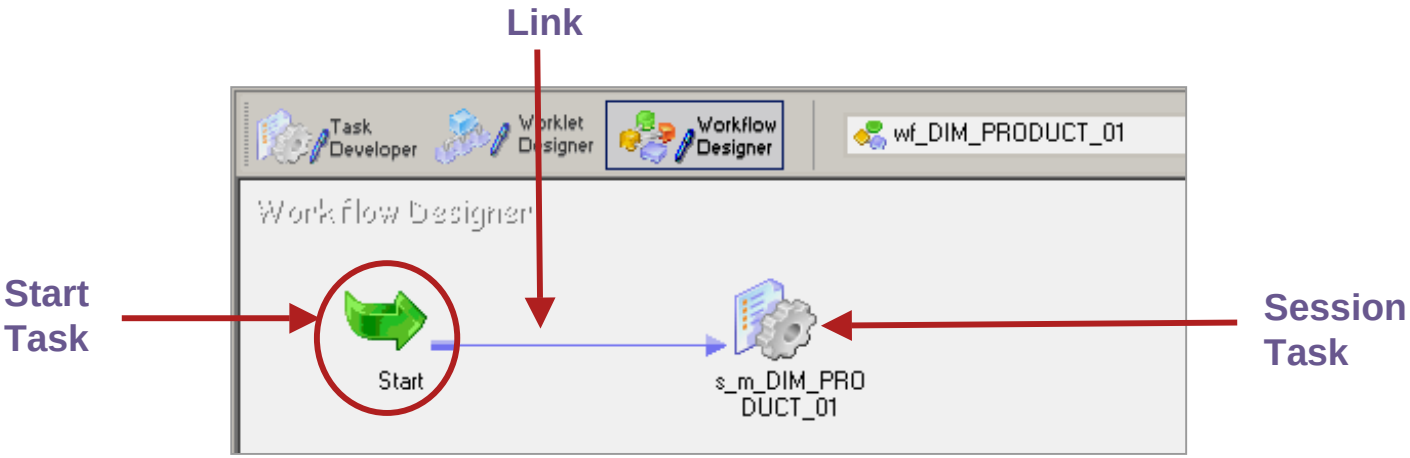
- **Worklet Designer**

  - Creates objects that represent a set of tasks
  - Worklet objects are reusable

INFORMATICA

Workflow Structure

- A *Workflow* is set of instructions for the Informatica Server to perform data transformation and load
- Combines the logic of Session Tasks, other types of Tasks and Worklets
- The simplest Workflow is composed of a Start Task, a Link and one other Task



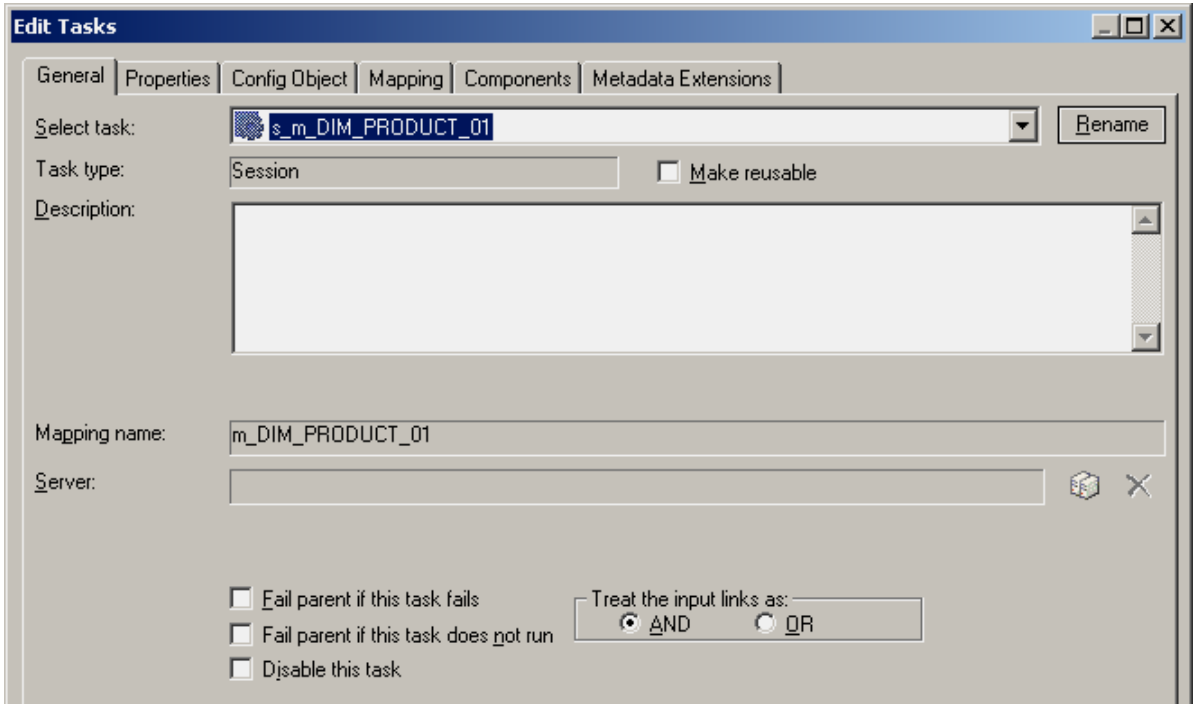
INFORMATICA

Session Task

- Server instructions to run the logic of ONE specific mapping

  e.g. source and target data location specifications, memory allocation, optional Mapping overrides, scheduling, processing and load instructions

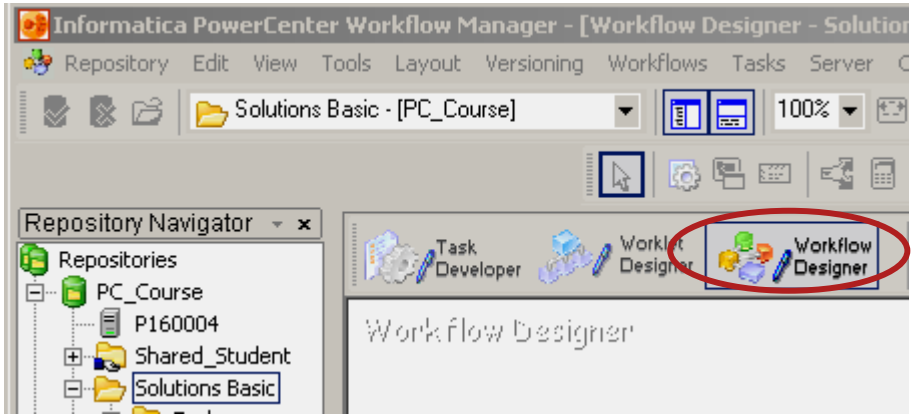
- Becomes a component of a Workflow (or Worklet)
- If configured in the Task Developer, the Session Task is reusable (optional)



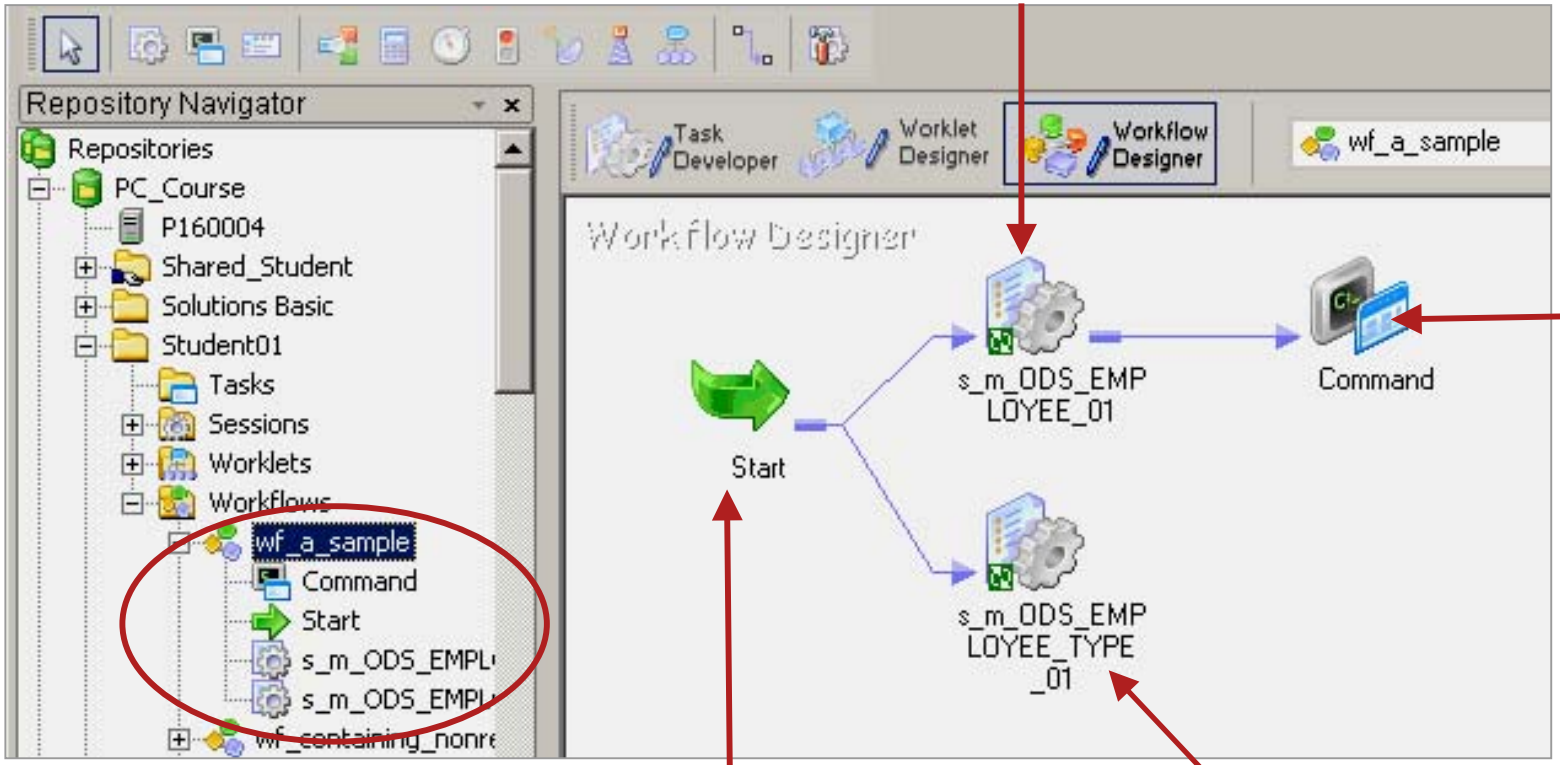
INFORMATICA

Additional Workflow Tasks

- **Eight additional Tasks are available in the Workflow Designer (covered later)**
  - Command
  - Email
  - Decision
  - Assignment
  - Timer
  - Control
  - Event Wait
  - Event Raise



Sample Workflow



Session 1

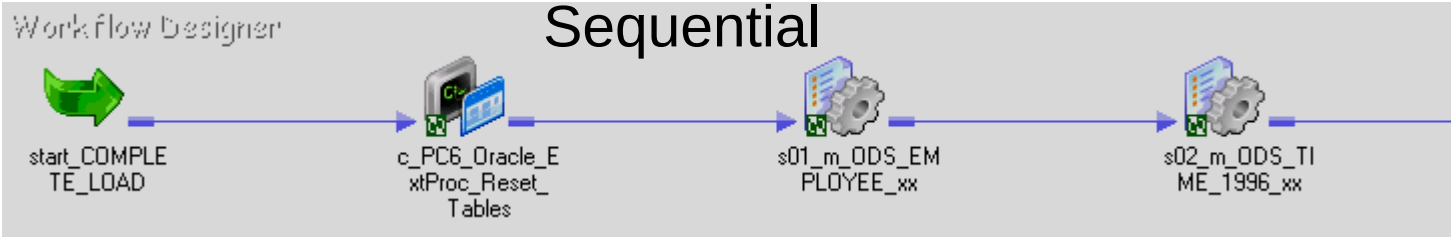
Command Task

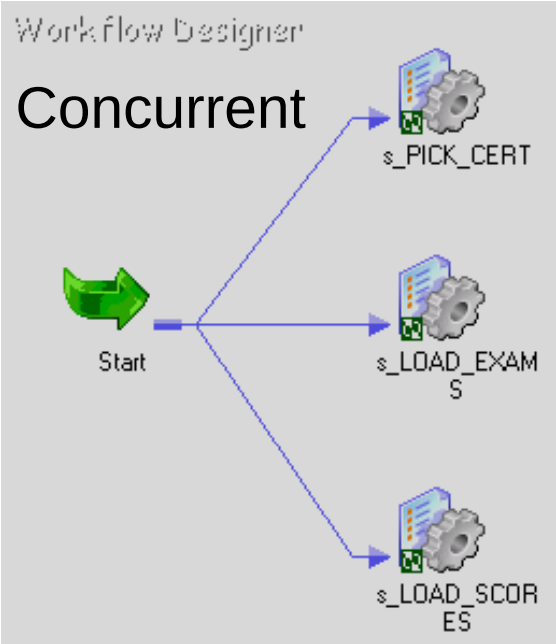
Start Task
(required)

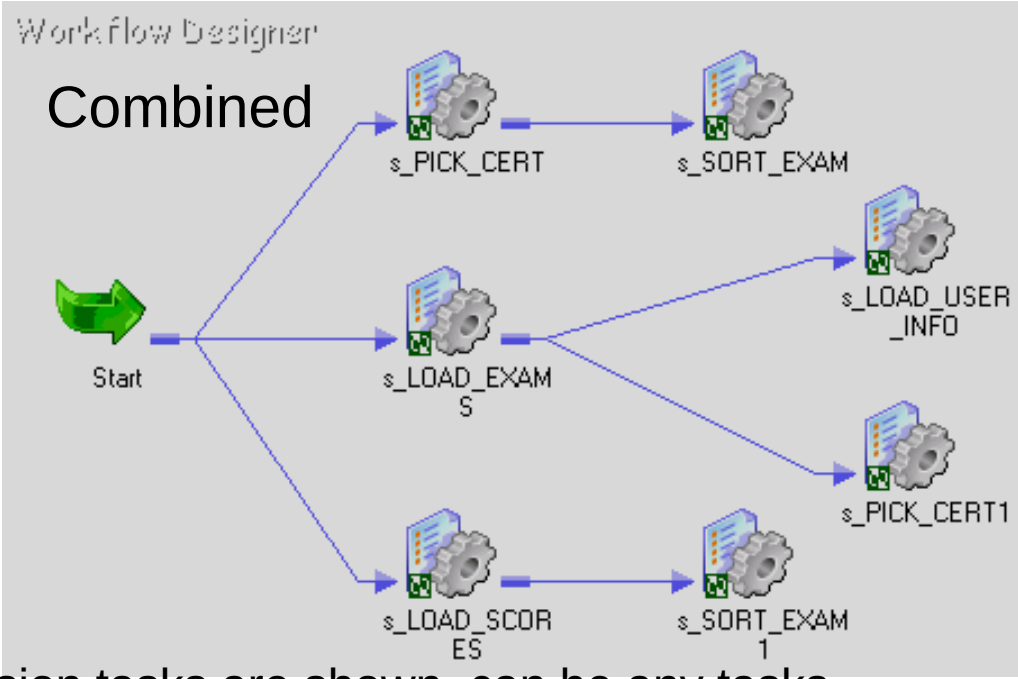
Session 2

INFORMATICA

Sequential and Concurrent Workflows



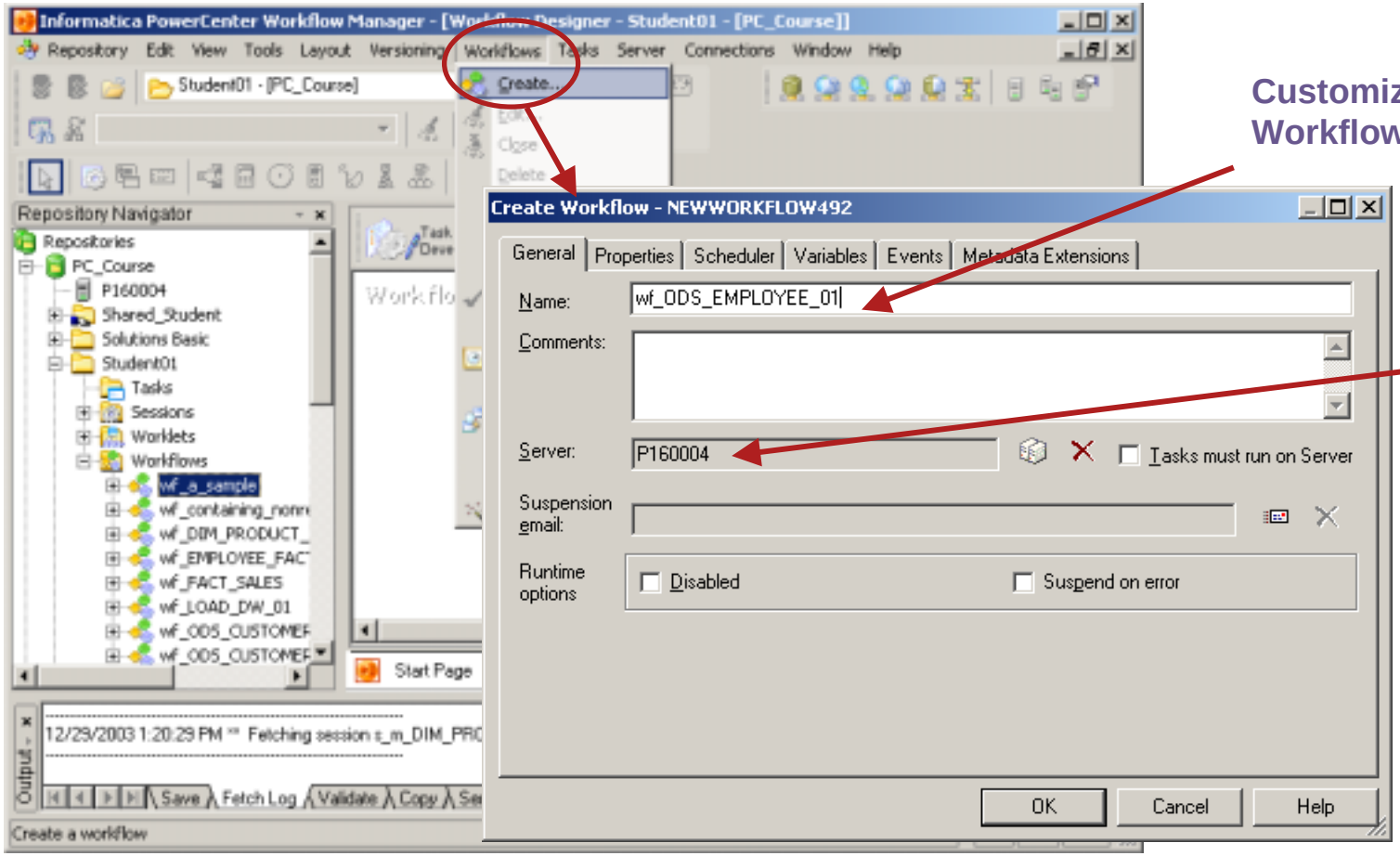




Note: Although only session tasks are shown, can be any tasks

INFORMATICA

Creating a Workflow

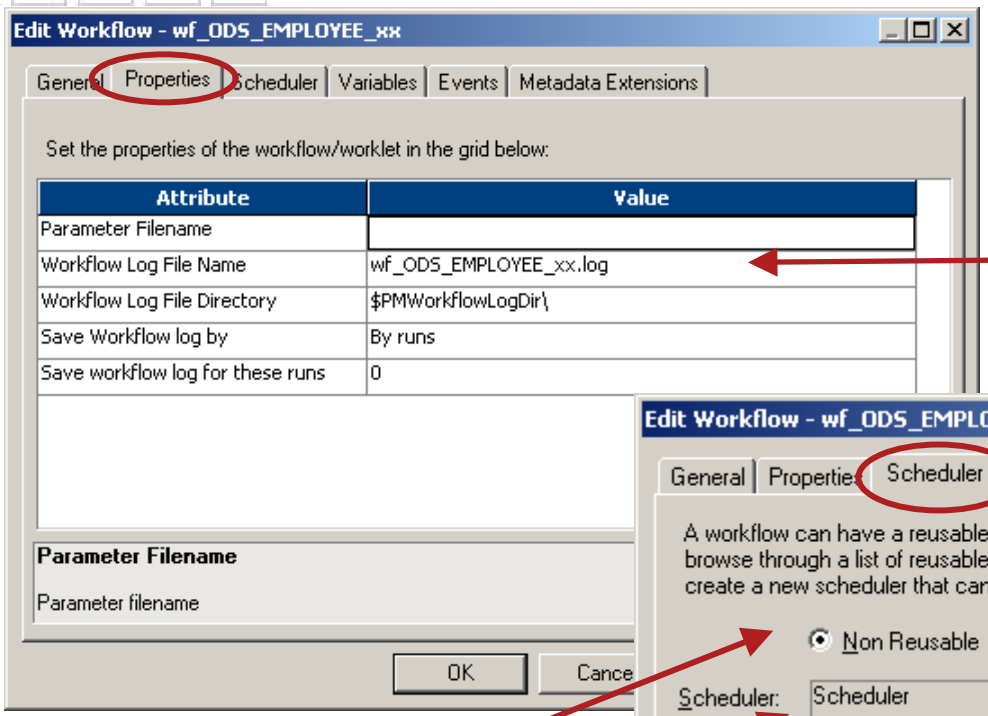


Customize Workflow name

Select a Server

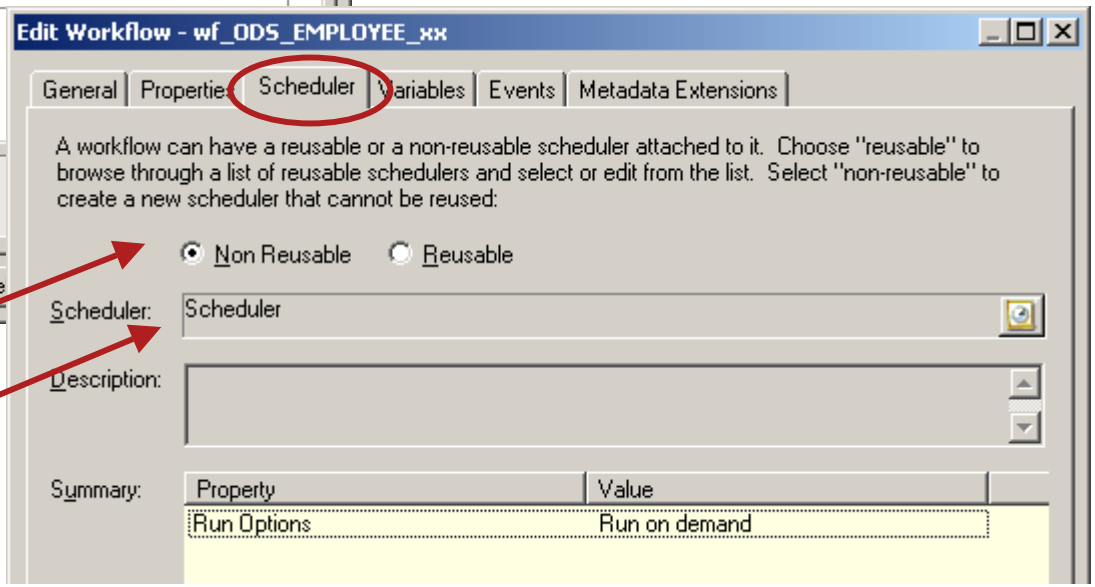
INFORMATICA

Workflow Properties



Customize Workflow Properties

Workflow log displays

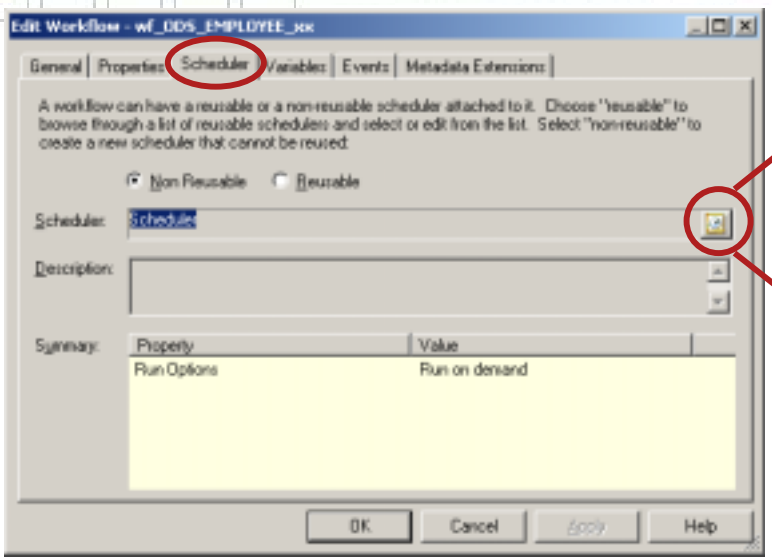


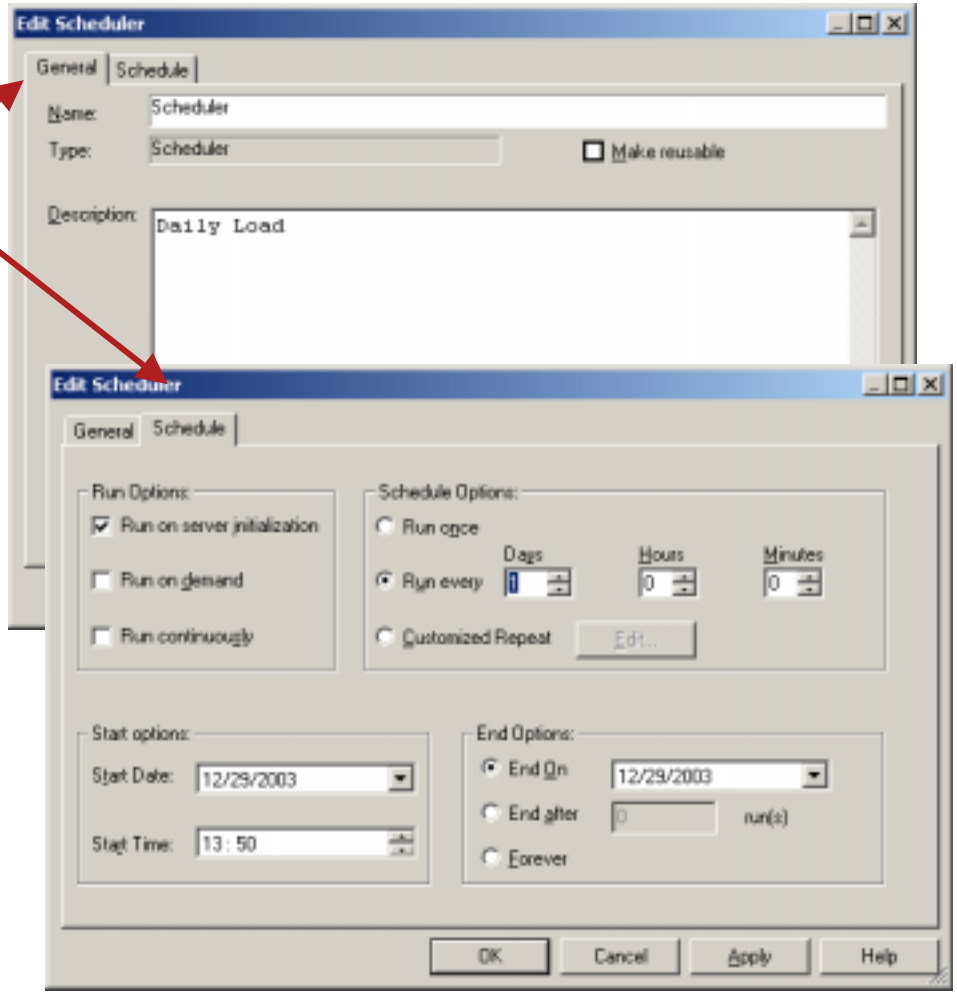
May be reusable or non-reusable

Select a Workflow Schedule (optional)

INFORMATICA

Workflow Scheduler

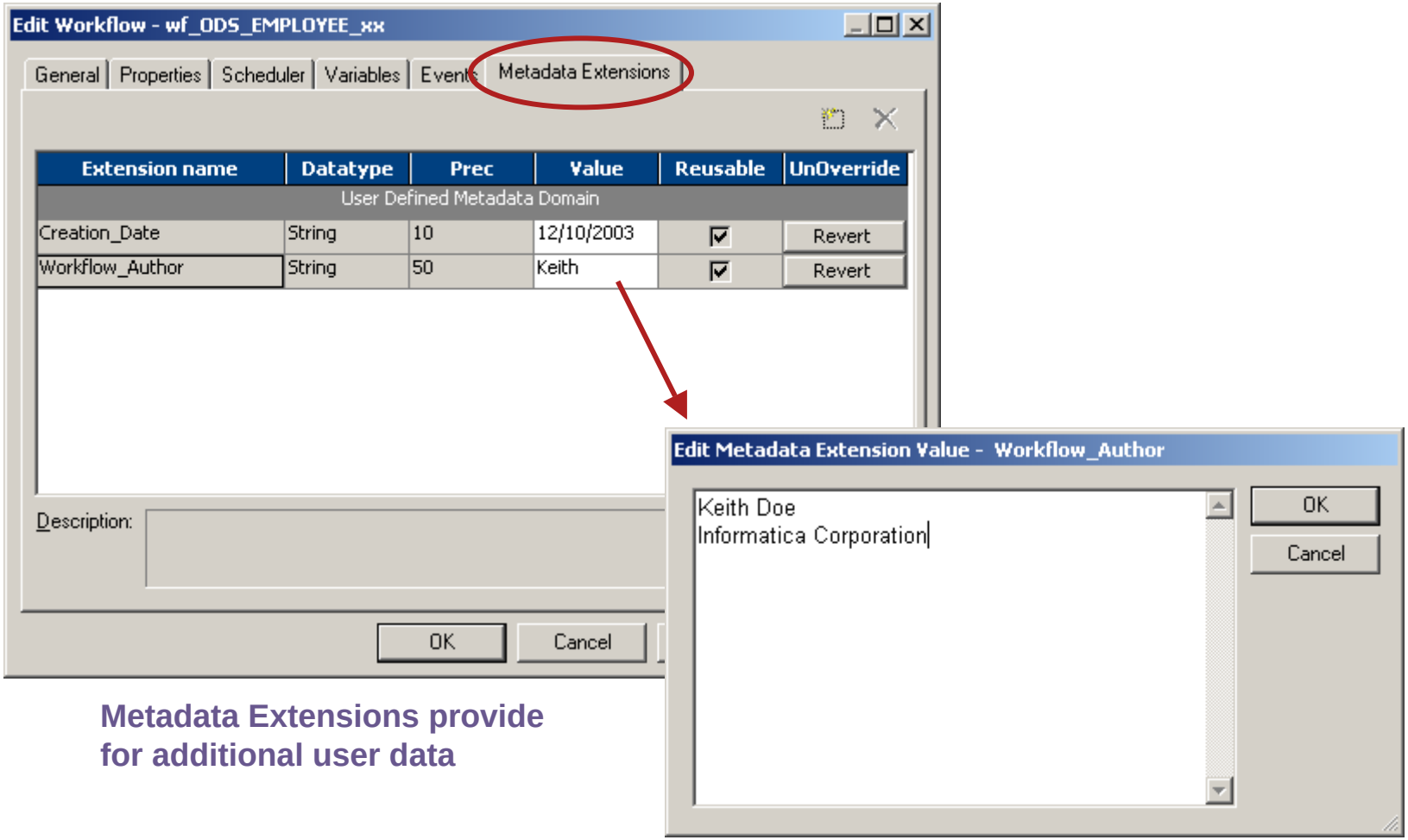




Set and customize workflow-specific schedule

INFORMATICA

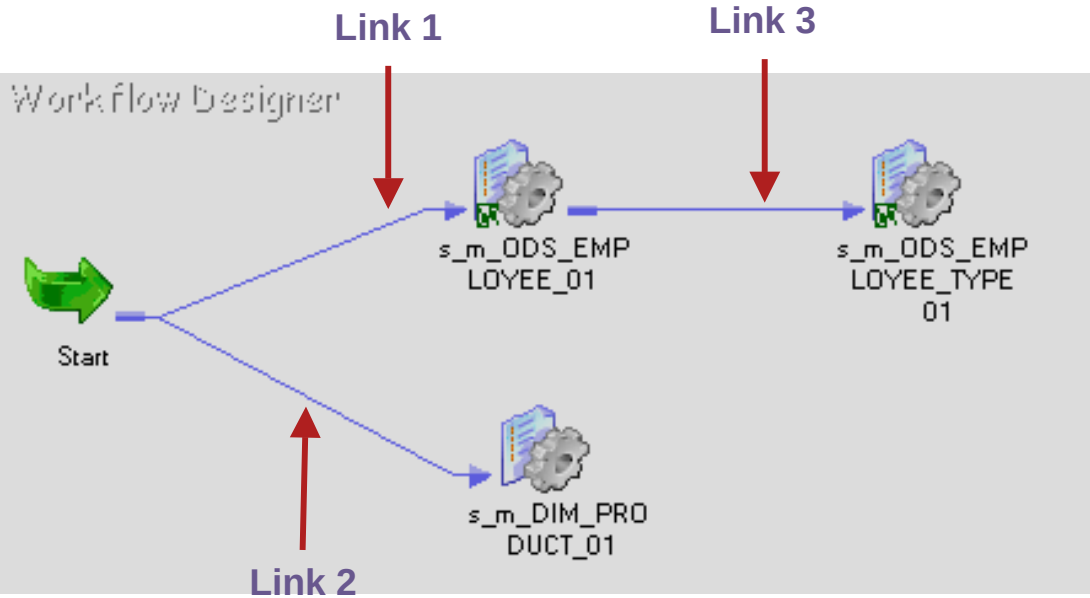
Workflow Metadata Extensions



Metadata Extensions provide for additional user data

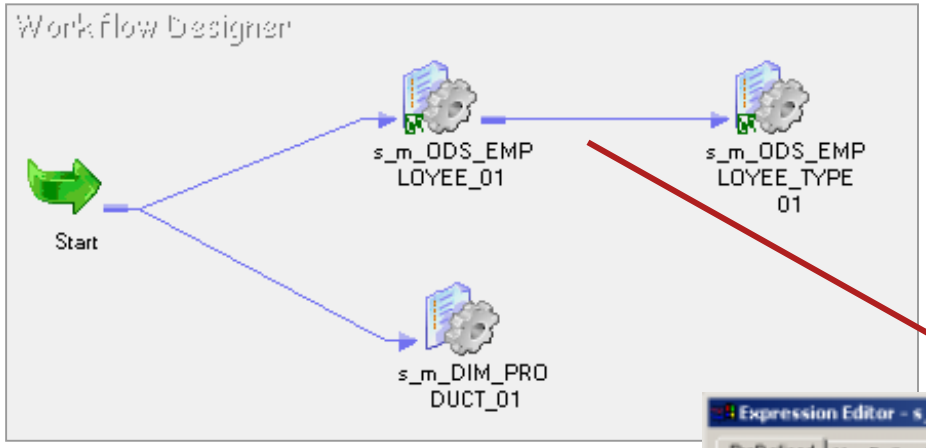
Workflow Links

- Required to connect Workflow Tasks
- Can be used to create branches in a Workflow
- All links are executed – unless a link condition is used which makes a link false

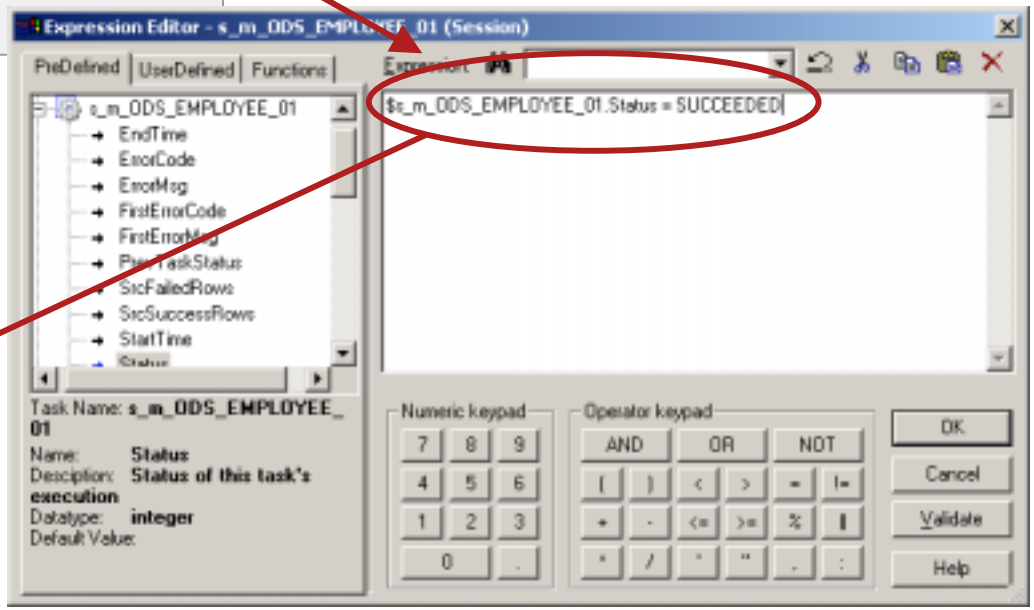


INFORMATICA

Conditional Links



Optional link condition

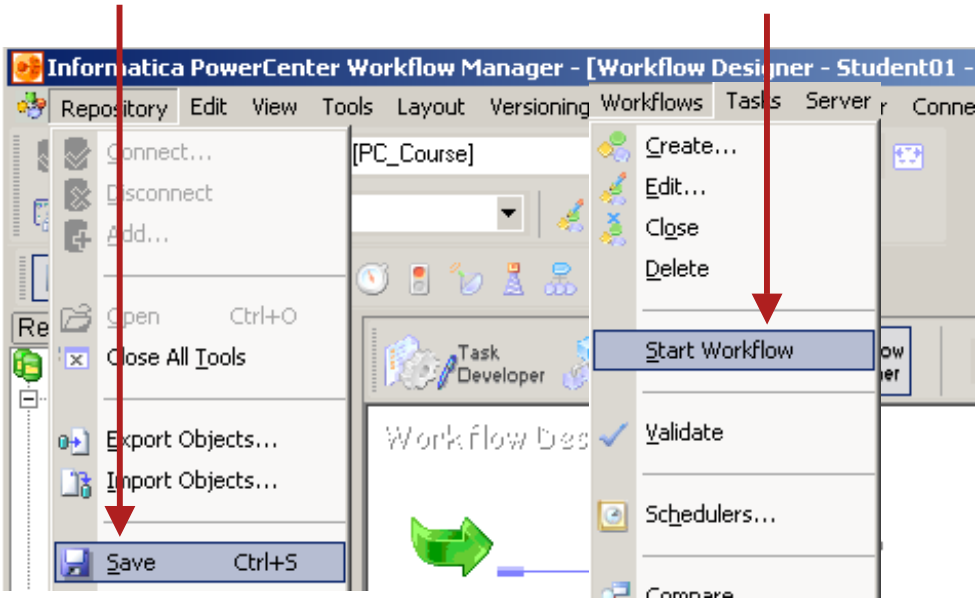


'$*taskname*.STATUS'
is a pre-defined workflow variable

INFORMATICA

Workflow Summary

1. Add Sessions and other Tasks to the Workflow
2. Connect all Workflow components with Links
3. Save the Workflow
4. Start the Workflow



Sessions in a Workflow can be executed independently

Session Tasks

INFORMATICA

Session Tasks

After this section, you will be familiar with:

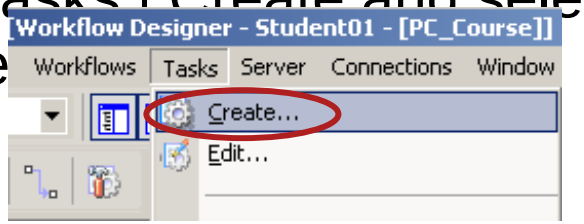
- **How to create and configure Session Tasks**
- **Session Task source and target properties**

INFORMATICA

Creating a Session Task

- **Created to execute the logic of a mapping (one mapping only)**
- **Session Tasks can be created in the Task Developer (reusable) or Workflow Developer (Workflow-specific)**
- **To create a Session Task**
  - Select the Session button from the Task Toolbar

  - **Or** Select menu Tasks | Create and select Session from the drop-down me



INFORMATICA

Session Task Tabs

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

INFORMATICA

Session Task – Mapping Sources

lapping Tab

ession Task

elect source
nstance

et connection

et properties

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_ODS_EMPLOYEE_01

Task type: Session (Reusable)

- Connections
- Memory Properties
- Files and Directories
- Sources
  - sq_EMPLOYEE
- Targets
  - ODS_EMPLOYEE
- Transformations
  - exp_ODS_EMPLOYEE

Transf... | Partitions

Start Page

m_ODS_EMPLOYEE_01.sq_EMPLOYEE

Readers

Instance	Readers
sq_EMPLOYEE	Relational Reader

Connections

Type	Value
sq_EMPLOYEE - DB Connection	
Relational	native_source

Properties — Show Session Level Properties

Attribute	Value
sq_EMPLOYEE - Source Qualifier	
User Defined Join	
Number Of Sorted Ports	0
Tracing Level	Normal

OK | Cancel | Apply | Help

INFORMATICA

Session Task – Mapping Targets

lapping Tab

ession Task

elect target
nstance

et connection

et properties

ote: Heterogeneous
argets are supported

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_ODS_EMPLOYEE_01

Task type: Session (Reusable)

- Connections
- Memory Properties
- Files and Directories
- Sources
  - sq_EMPLOYEE
- Targets
  - ODS_EMPLOYEE
- Transformations
  - exp_ODS_EMPLOYEE

Transf... | Partitions

Start Page

m_ODS_EMPLOYEE_01.ODS_EMPLOYEE

Writers

Instance	Writers
ODS_EMPLOYEE	Relational Writer

Connections

Type	Value
Relational	native_target_01

Properties — Show Session Level Properties

Attribute	Value
ODS_EMPLOYEE - Relational Writer	
Target load type	Normal
Insert	☑
Update as Update	☑

OK | Cancel | Apply | Help

INFORMATICA

Monitoring Workflows

INFORMATICA

Monitoring Workflows

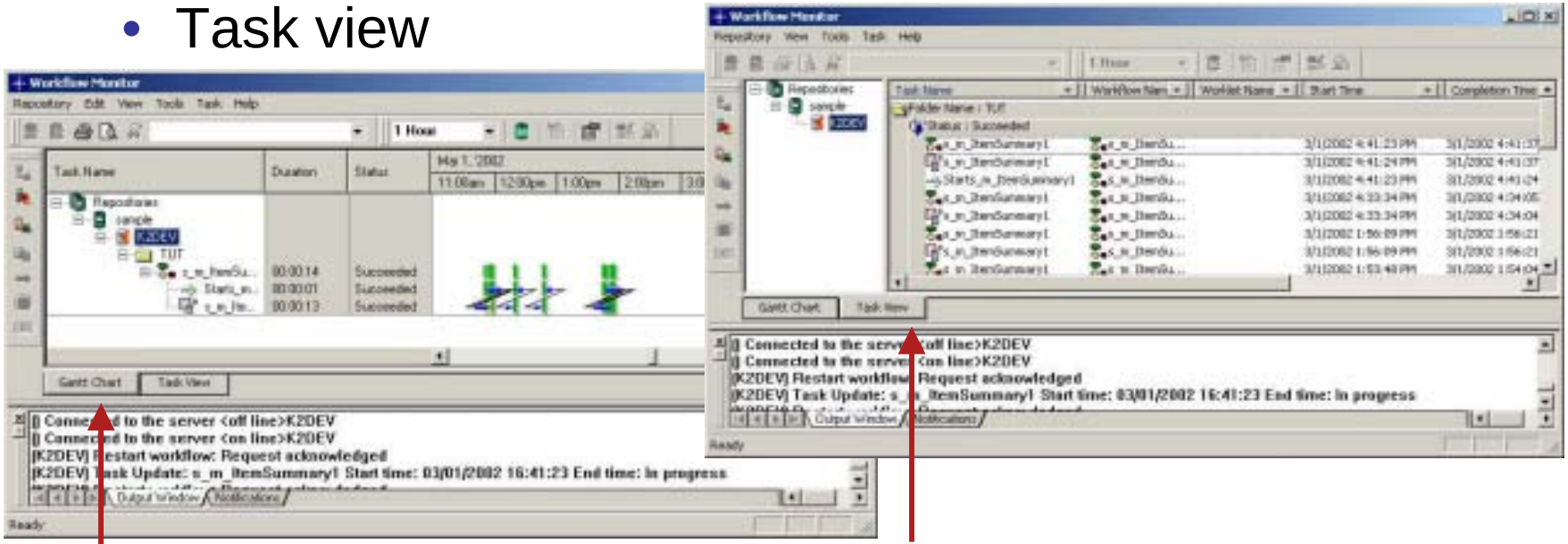
By the end of this section you will be familiar with:

- The Workflow Monitor GUI interface
- Monitoring views
- Server monitoring modes
- Filtering displayed items
- Actions initiated from the Workflow Monitor
- Truncating Monitor Logs

INFORMATICA

Workflow Monitor

- **The Workflow Monitor is the tool for monitoring Workflows and Tasks**
- **Choose between two views:**
  - Gantt chart
  - Task view



Gantt Chart view

Task view

INFORMATICA

Monitoring Current and Past Workflows

- **The Workflow Monitor displays only workflows that have been run**
- **Choose between two modes:**
  - Online

    Displays real-time information from the Informatica Server and the Repository Server about current workflow runs
  - Offline

    Displays historic information from the Repository about past workflow runs

Refresh rate adjustment not required; in online mode, screen is automatically refreshed

INFORMATICA

Monitoring Operations

- **Perform operations in the Workflow Monitor**
  - Stop, Abort, or Restart a Task, Workflow or Worklet
  - Resume a suspended Workflow after a failed Task is corrected
  - Reschedule or Unschedule a Workflow
- **View Session and Workflow logs**
- **Abort has a 60 second timeout**
  - If the Server has not completed processing and committing data during the timeout period, the threads and processes associated with the Session are killed

Stopping a Session Task means the Server stops reading data

Monitoring in Task View

Task | Server | Workflow | Worklet | Start Time | Completion Time

Informatica PowerCenter Workflow Monitor

Repository View Tools Task Help

1 Hour

Repositories
PC_Course
P160004

Task Name	Worker Server Name	Workflow Name	Worklet Name	Start Time	Completion Time
Folder Name : Student01					
Status : Succeeded					
wf_ODS_ORDER_AMOUNT_01		wf_ODS_ORDER_AMOUNT_01		1/23/2004 4:19:19 PM	1/23/2004 4:19:25 PM
s_m_ODS_ORDER_AMOUNT_01	P160004	wf_ODS_ORDER_AMOUNT_01		1/23/2004 4:19:19 PM	1/23/2004 4:19:25 PM
Status : Running					
ew_EMPLOYEE_LOAD_COMPLETE	P160004	wf_EMPLOYEE_FACT_SALES...		1/23/2004 4:20:26 PM	In progress
wf_EMPLOYEE_FACT_SALES_L...		ES...		1/23/2004 4:20:26 PM	In progress

Restart Workflow from Task
Restart Task
Recover Workflow from Task
Recover Task
Stop
Abort
Go to Gantt Chart
Get Session Log
Properties...

Gantt Chart | Task View

Output Window

() Connected to the server <on line> P160004
(P160004 1/23/2004 4:19:22 PM) Task Update: wf_ODS_ORDER_AMOUNT_01 (Running) S... me: In progress
(P160004 1/23/2004 4:19:22 PM) Task Update: s_m_ODS_ORDER_AMOUNT_01 (Running) ... time: In progress
(P160004 1/23/2004 4:19:22 PM) Task Update: Start (Succeeded) Start time: 1/23/2004 4:19: ... M
(P160004 1/23/2004 4:19:25 PM) Task Update: s_m_ODS_ORDER_AMOUNT_01 (Succeede... nd time: 1/23/2004 4:19:25 PM
(P160004 1/23/2004 4:19:25 PM) Task Update: wf_ODS_ORDER_AMOUNT_01 (Succeeded) ... d time: 1/23/2004 4:19:25 PM
(P160004 1/23/2004 4:20:26 PM) Task Update: wf_EMPLOYEE_FACT_SALES_LOAD_01 (Ru... PM End time: In progress
(P160004 1/23/2004 4:20:26 PM) Task Update: ew_EMPLOYEE_LOAD_COMPLETE (Running) Start time: 1/23/2004 4:20:26 PM End time: In progress

Output Window | Notifications

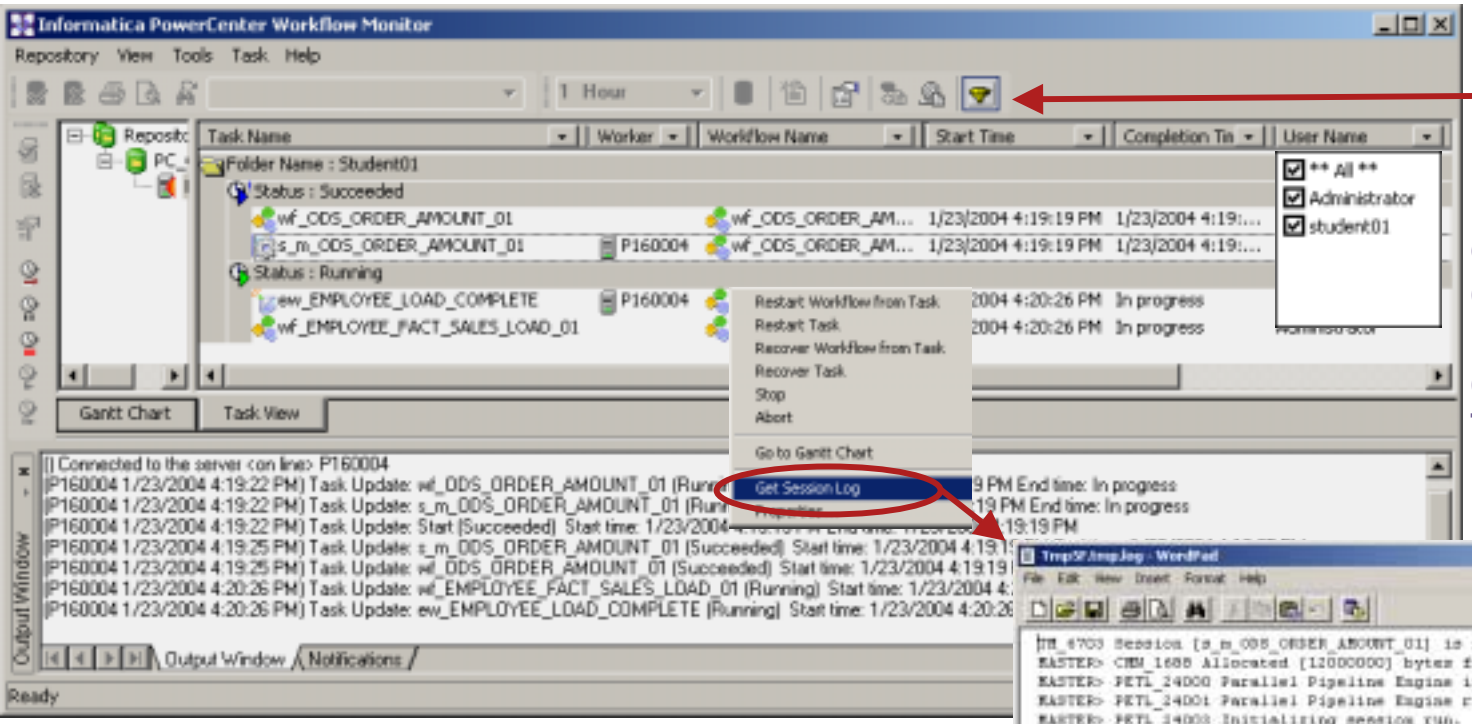
Ready

Status Bar

Start, Stop, Abort, Resume
Tasks,Workflows and Worklets

INFORMATICA

Filtering in Task View

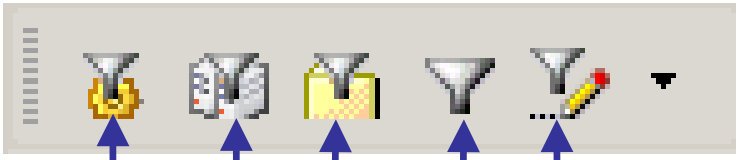


Monitoring filters can be set using drop down menus. Minimizes items displayed in Task View

Right-click on Session to retrieve the Session Log (from the Server to the local PC Client)

INFORMATICA

Filter Toolbar



Select type of tasks to filter

Select servers to filter

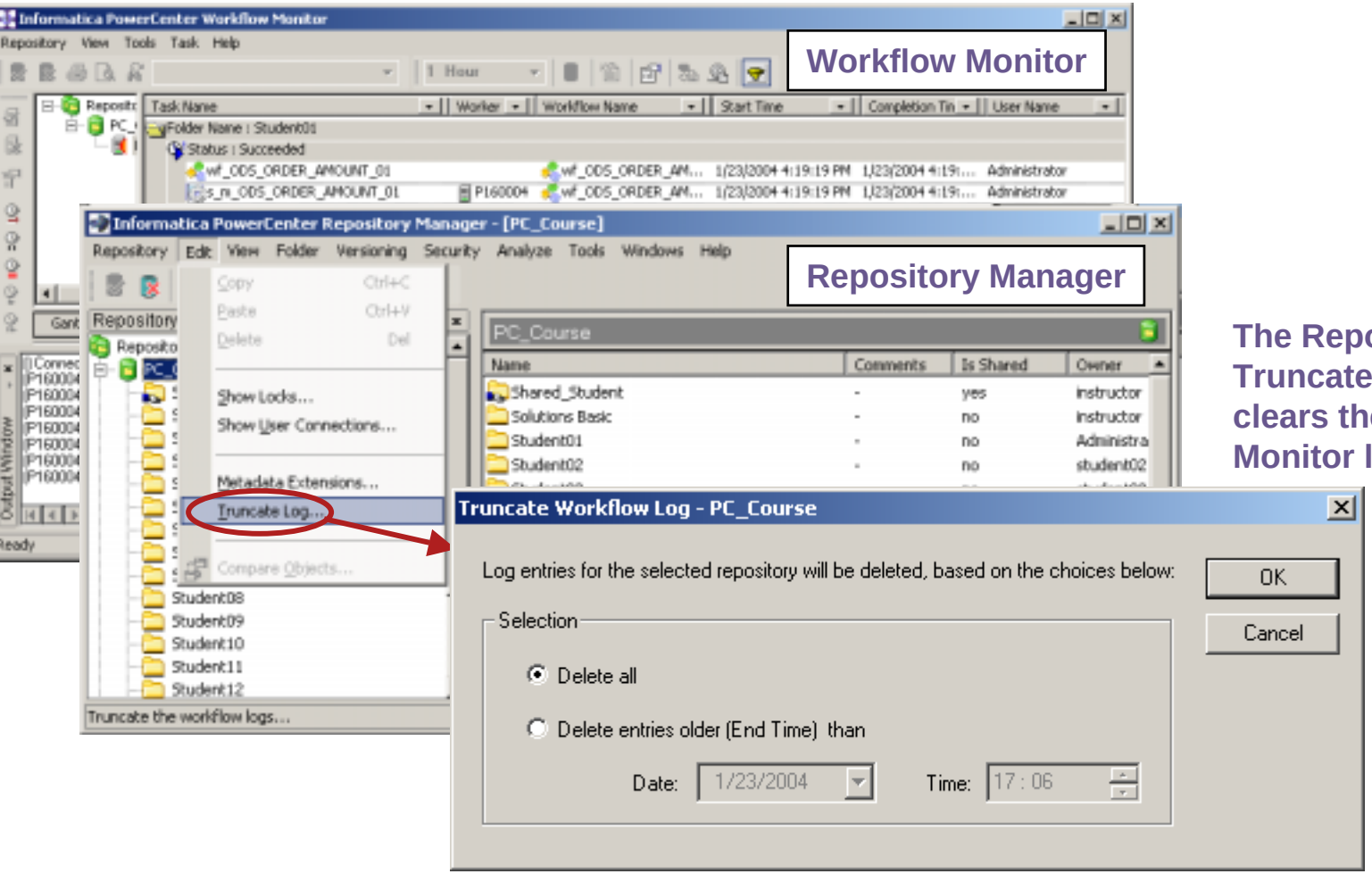
View all folders or folders owned only by current user

Filter tasks by specified criteria

Display recent runs

INFORMATICA

Truncating Workflow Monitor Logs



The Repository Manager Truncate Log option clears the Workflow Monitor logs

INFORMATICA

Lab 3 – Create and Run a Workflow

INFORMATICA

Lab 4 – Features and Techniques I

Debugger

INFORMATICA

Debugger

By the end of this section you will be familiar with:

- Creating a Debug Session
- Debugger windows and indicators
- Debugger functionality and options
- Viewing data with the Debugger
- Setting and using Breakpoints
- Tips for using the Debugger

INFORMATICA

Debugger Features

- Wizard driven tool that runs a test session
- View source / target data
- View transformation data
- Set break points and evaluate expressions
- Initialize variables
- Manually change variable values
- Data can be loaded or discarded
- Debug environment can be saved for later use

INFORMATICA

Debugger Interface

Flashing yellow SQL indicator

Debugger Mod indicator

Solid yellow arrow Curren Transformatio indicator

Output Window – Debugger Log

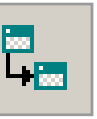
Transformation Instance Data window

Target Instance window

INFORMATICA

Debugger Tips

- Server must be running before starting a Debug Session
- When the Debugger is started, a spinning icon displays. Spinning stops when the Debugger Server is ready
- The flashing yellow/green arrow points to the current active Source Qualifier. The solid yellow arrow points to the current Transformation instance



Next Instance – proceeds a single step at a time; one row moves from transformation to transformation

Step to Instance – examines one transformation at a time, following successive rows through the same transformation

INFORMATICA

Lab 5 – The Debugger

INFORMATICA

Filter Transformation

INFORMATICA

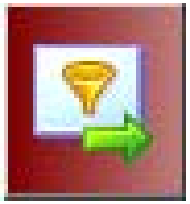
Filter Transformation

By the end of this section you will be familiar with:

- Filter functionality
- Filter properties

INFORMATICA

Filter Transformation

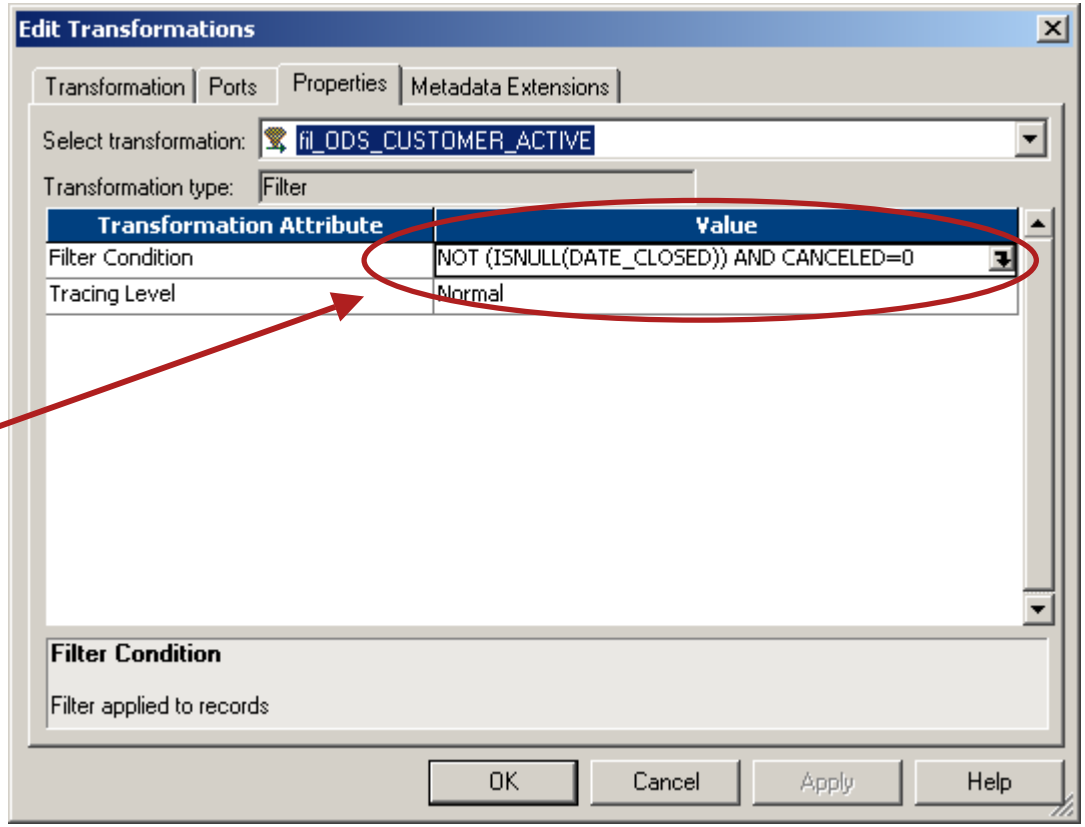


Drops rows conditionally

Ports
- All input / output

Specify a Filter condition

Usage
- Filter rows from input flow



INFORMATICA

Lab 6 – Flat File Wizard and Filter Transformation

INFORMATICA

Sorter Transformation

INFORMATICA

Sorter Transformation

By the end of this section you will be familiar with:

- Sorter functionality
- Sorter properties

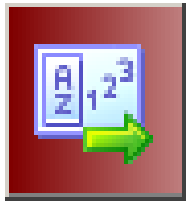
INFORMATICA

Sorter Transformation

- Can sort data from relational tables or flat files
- Sort takes place on the Informatica Server machine
- Multiple sort keys are supported
- The Sorter transformation is often more efficient than a sort performed on a database with an ORDER BY clause

INFORMATICA

Sorter Transformation



Sorts data from any source, at any point in a data flow

Ports

- Input/Output
- Define one or more sort keys
- Define sort order for each key

Example of Usage

- Sort data before Aggregator to improve performance

Sort Keys

Edit Transformations

Transformation | Ports | Properties | Metadata Extensions

Select transformation: srt_AGG_SALES_BY_DEP_MONTH_REP

Transformation type: Sorter

	Port Name	Datatype	Prec	Scale	I	O	K...	Direction
1	PRODUCT_DEPARTMENT	string	25	0	☑	☑	☑	Ascending
2	MONTH	string	3	0	☑	☑	☑	Ascending
3	REP_ID	decimal	10	0	☑	☑	☑	Ascending
4	PRICE	decimal	10	2	☑	☑	☐	- N/A -
5	QUANTITY	decimal	10	0	☑	☑	☐	- N/A -
6	PROFIT	decimal	10	2	☑	☑	☐	- N/A -
7	COST	decimal	10	2	☑	☑	☐	- N/A -

Default value:

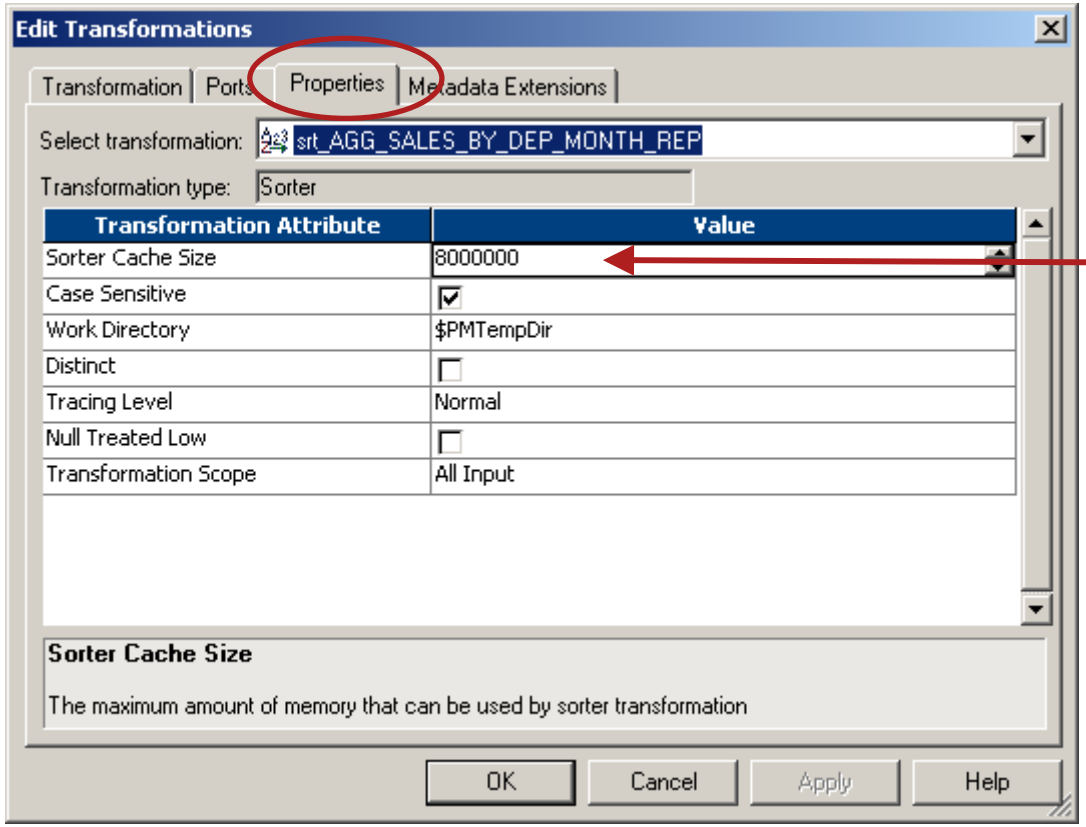
Description:

Sort Order

OK | Cancel | Apply | Help

INFORMATICA

Sorter Properties



- Cache size can be adjusted. **Default is 8 Mb**
- **Server uses twice the cache listed**
- **Ensure sufficient memory is available on the Informatica Server (else Session Task will fail)**

INFORMATICA

Aggregator Transformation

INFORMATICA

Aggregator Transformation

By the end of this section you will be familiar with:

- Basic Aggregator functionality
- Creating subtotals with the Aggregator
- Aggregator expressions
- Aggregator properties
- Using sorted data

INFORMATICA

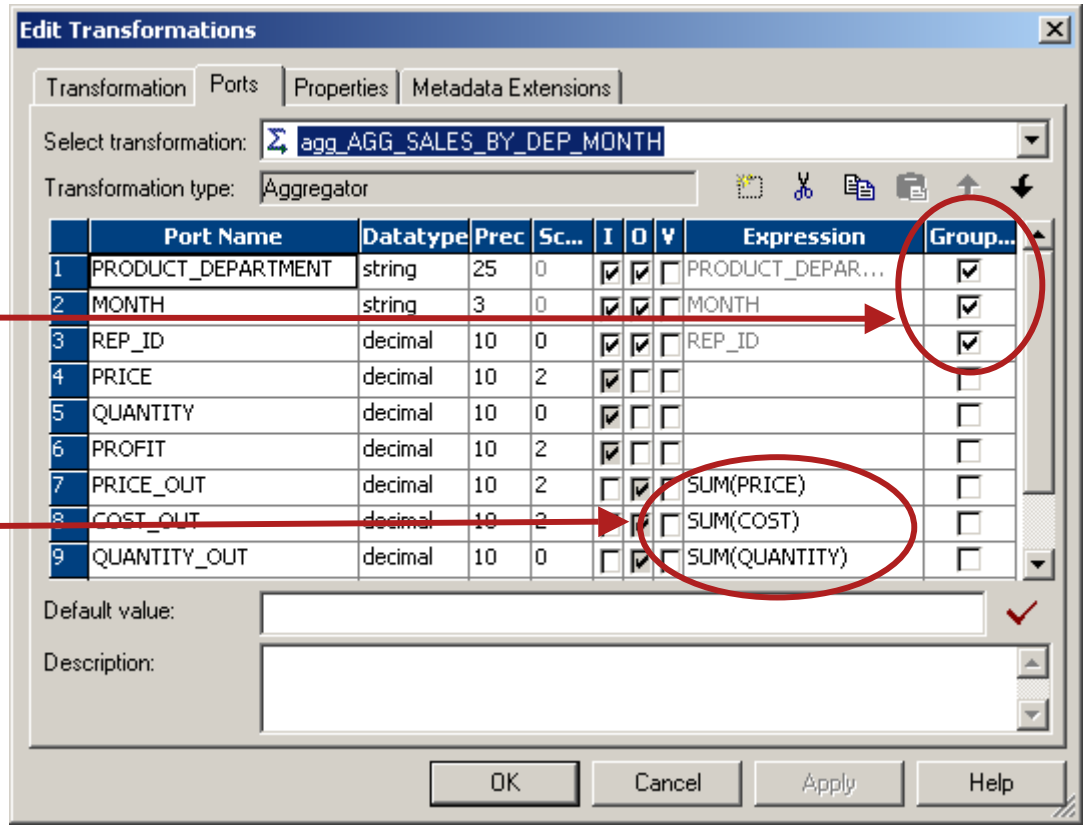
Aggregator Transformation

Performs aggregate calculations

Ports
- Mixed
- Variables allowed
- Group By allowed

Create expressions in output ports

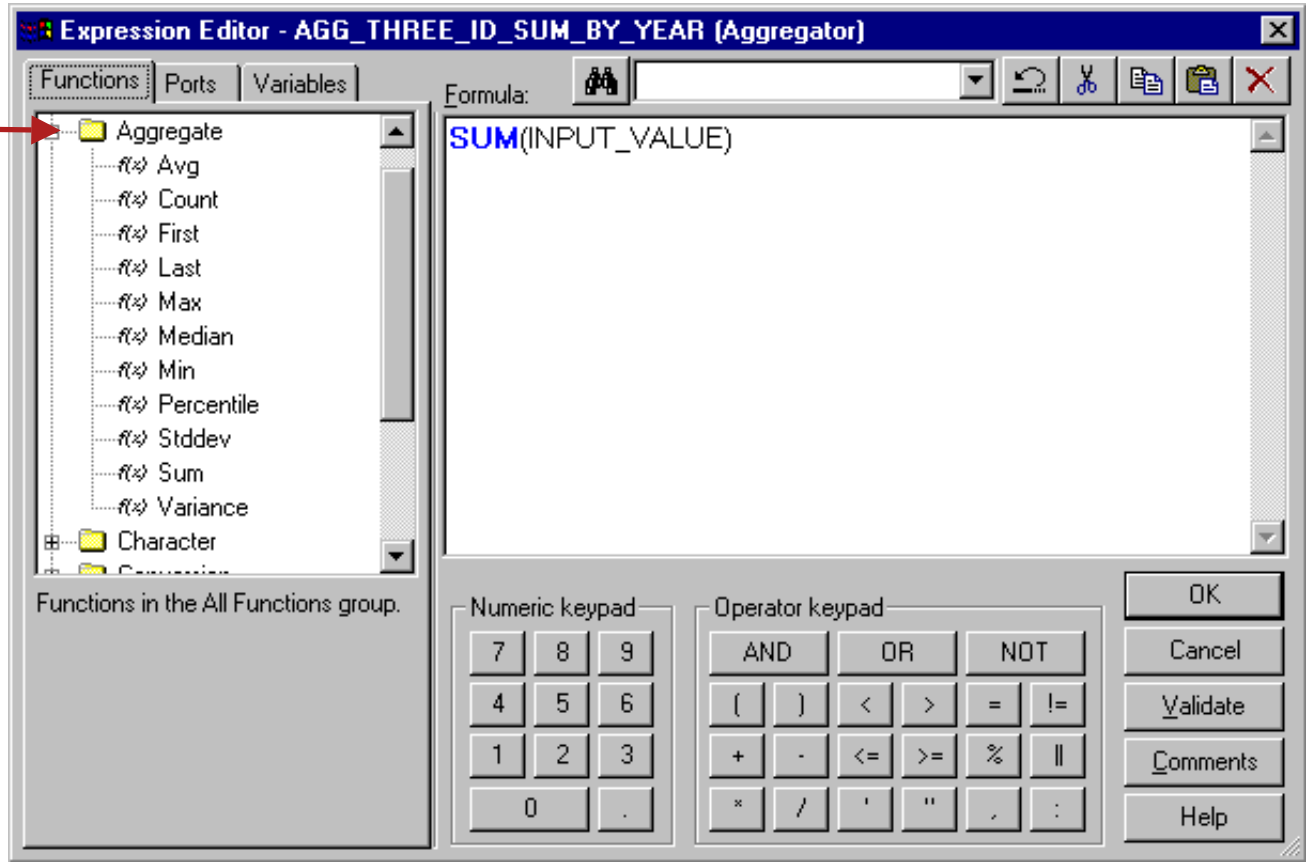
Usage
- Standard aggregations



INFORMATICA

Aggregate Expressions

Aggregate functions are supported only in the Aggregator Transformation



Conditional Aggregate expressions are supported: Conditional SUM format: SUM(value, condition)

INFORMATICA

Aggregator Functions

AVG
COUNT
FIRST
LAST
MAX
MEDIAN
MIN
PERCENTILE
STDDEV
SUM
VARIANCE

- Return summary values for non-null data in selected ports
- Use only in Aggregator transformations
- Use in output ports only
- Calculate a single value (and row) for all records in a group
- Only one aggregate function can be nested within an aggregate function
- Conditional statements can be used with these functions

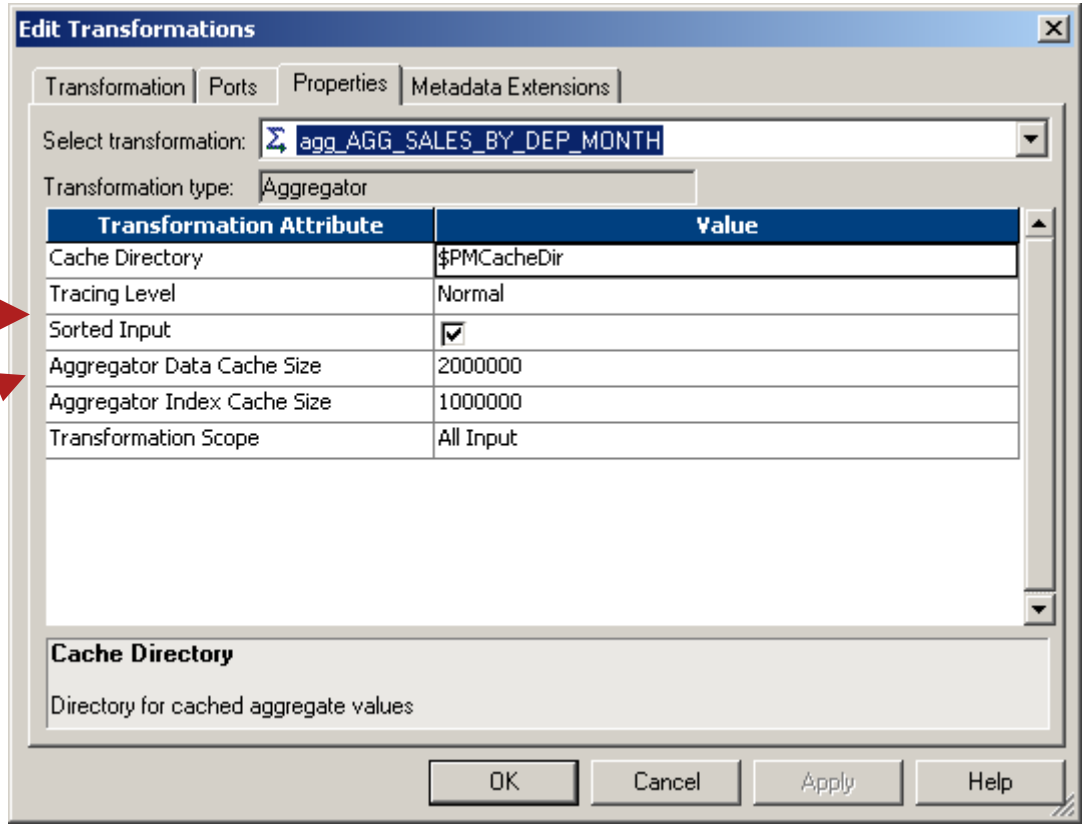
INFORMATICA

Aggregator Properties

Sorted Input Property

Instructs the Aggregator to expect the data to be sorted

Set Aggregator cache sizes for Informatica Server machine



INFORMATICA

Sorted Data

- The Aggregator can handle sorted or unsorted data

  Sorted data can be aggregated more efficiently, decreasing total processing time

- The Server will cache data from each group and release the cached data – *upon reaching the first record of the next group*

- Data *must* be sorted according to the order of the Aggregator's Group By ports

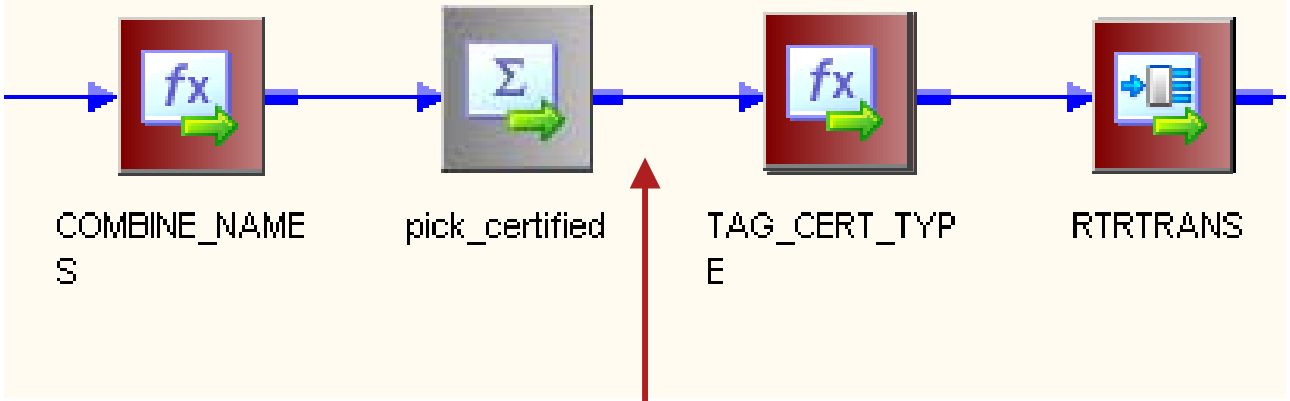
- Performance gain will depend upon varying factors

INFORMATICA

Aggregating Unsorted Data

Unsorted data

Group By:
- store
- department
- date



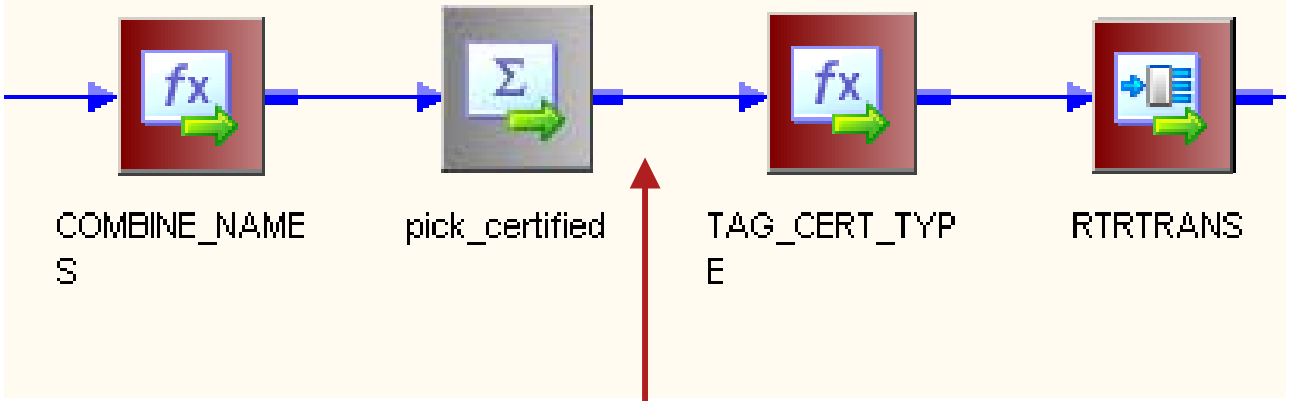
No rows are released from Aggregator until all rows are aggregated

INFORMATICA

Aggregating Sorted Data

Data sorted by:
- store
- department
- date

Group By:
- store
- department
- date



Each separate group (one row) is released as soon as the last row in the group is aggregated

INFORMATICA

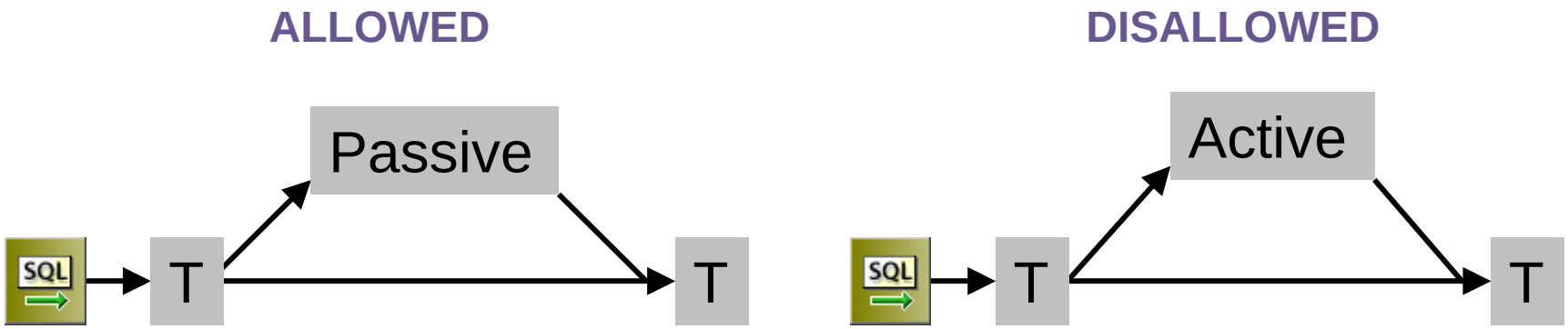
Data Flow Rules – Terminology

- **Passive transformation**
  - Operates on one row of data at a time **AND**
  - Cannot change the number of rows on the data flow
  - Example: Expression transformation
- **Active transformation**
  - Can operate on groups of data rows **AND/OR**
  - Can change the number of rows on the data flow
  - Examples: Aggregator, Filter, Source Qualifier

INFORMATICA

Data Flow Rules

- Each Source Qualifier starts a single data stream (data flow)
- Transformations can send rows to more than one transformation (split one data flow into multiple pipelines)
- Two or more data flows can meet only if they originate from a common active transformation



Example holds true with Normalizer instead of Source Qualifier.
Exceptions are: Mapplet Input and sorted Joiner transformations

INFORMATICA

Joiner Transformation

INFORMATICA

Joiner Transformation

By the end of this section you will be familiar with:

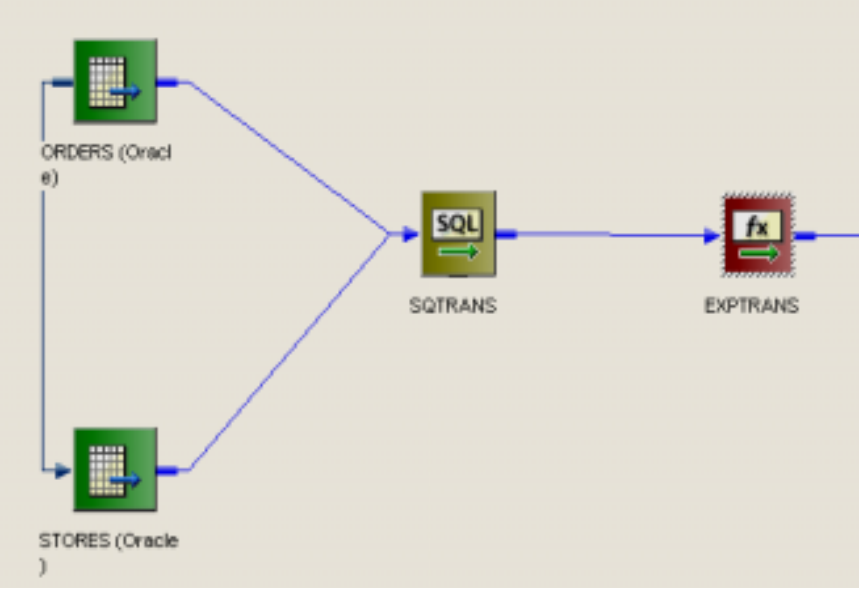
- When to use a Joiner transformation
- Homogeneous joins
- Heterogeneous joins
- Joiner properties
- Joiner conditions
- Nested joins

INFORMATICA

Homogeneous Joins

Joins can be performed within a Source Qualifier (using a SQL Query) when:

- The source tables are on the same database server *and*
- The database server performs the join

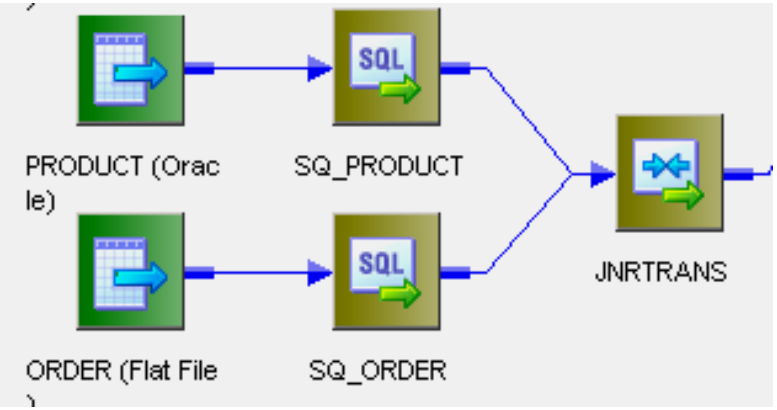


INFORMATICA

Heterogeneous Joins

Joins *cannot* be performed within a Source Qualifier when

- The source tables or on different database servers
- The sources are heterogeneous e.g.
  - **An Oracle table and a DB2 table**
  - **Two flat files**
  - **A flat file and a database table**



INFORMATICA

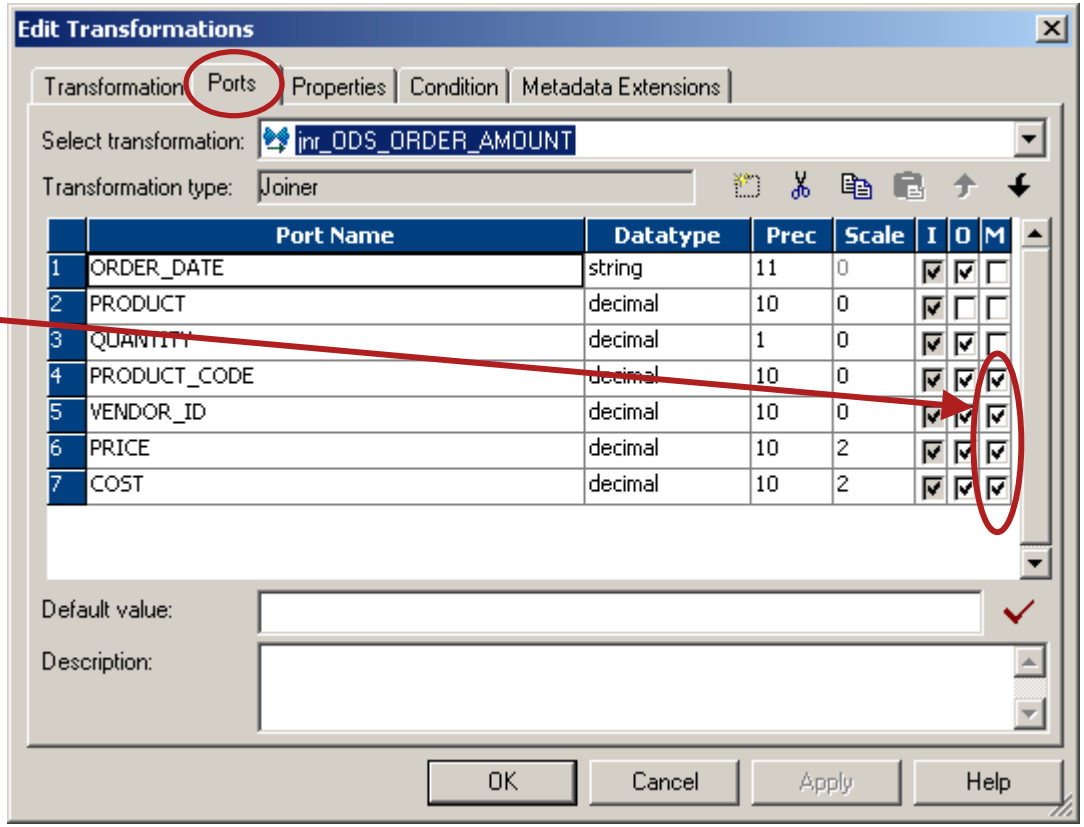
Joiner Transformation

Performs heterogeneous joins on different data flows

Active Transformation

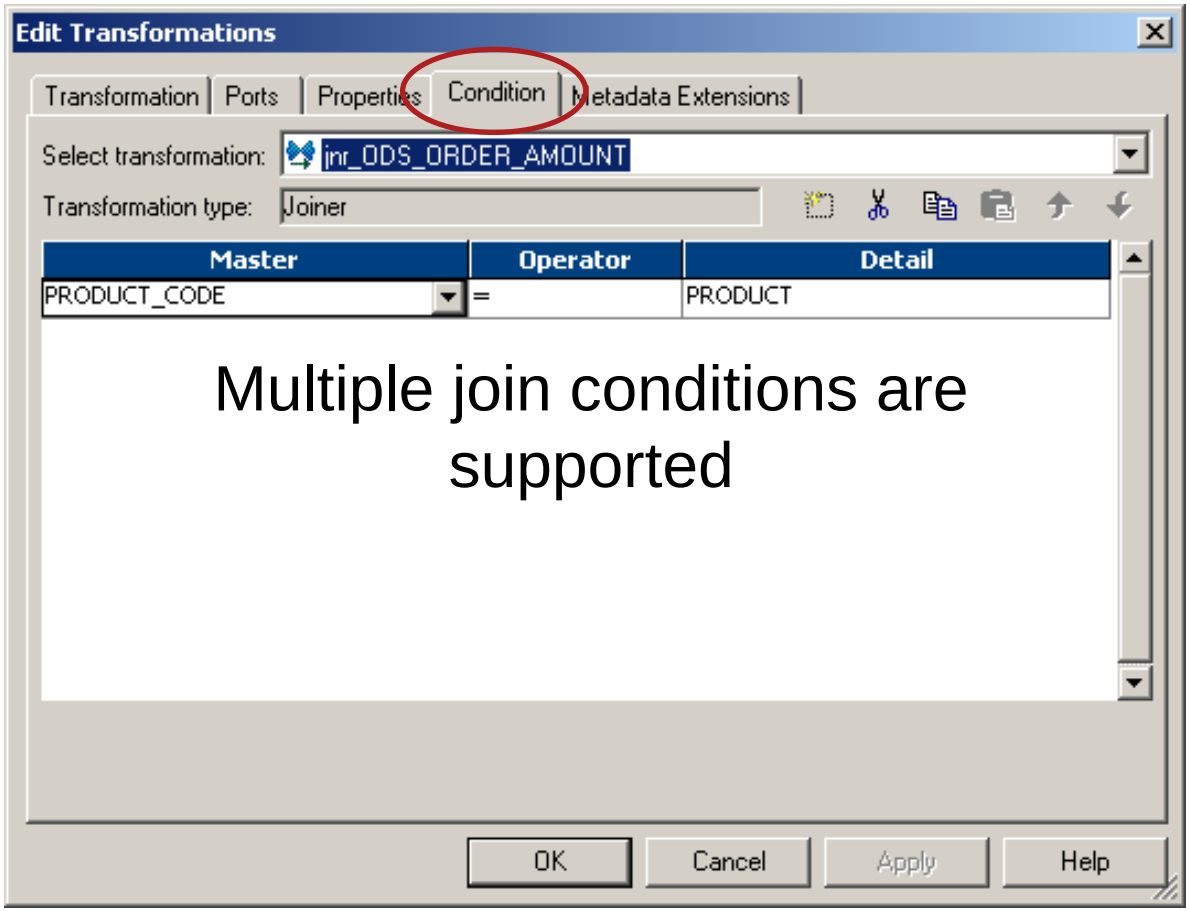
Ports
- All input or input / output
- "M" denotes port comes from master source

Examples
- Join two flat files
- Join two tables from different databases
- Join a flat file with a relational table



INFORMATICA

Joiner Conditions

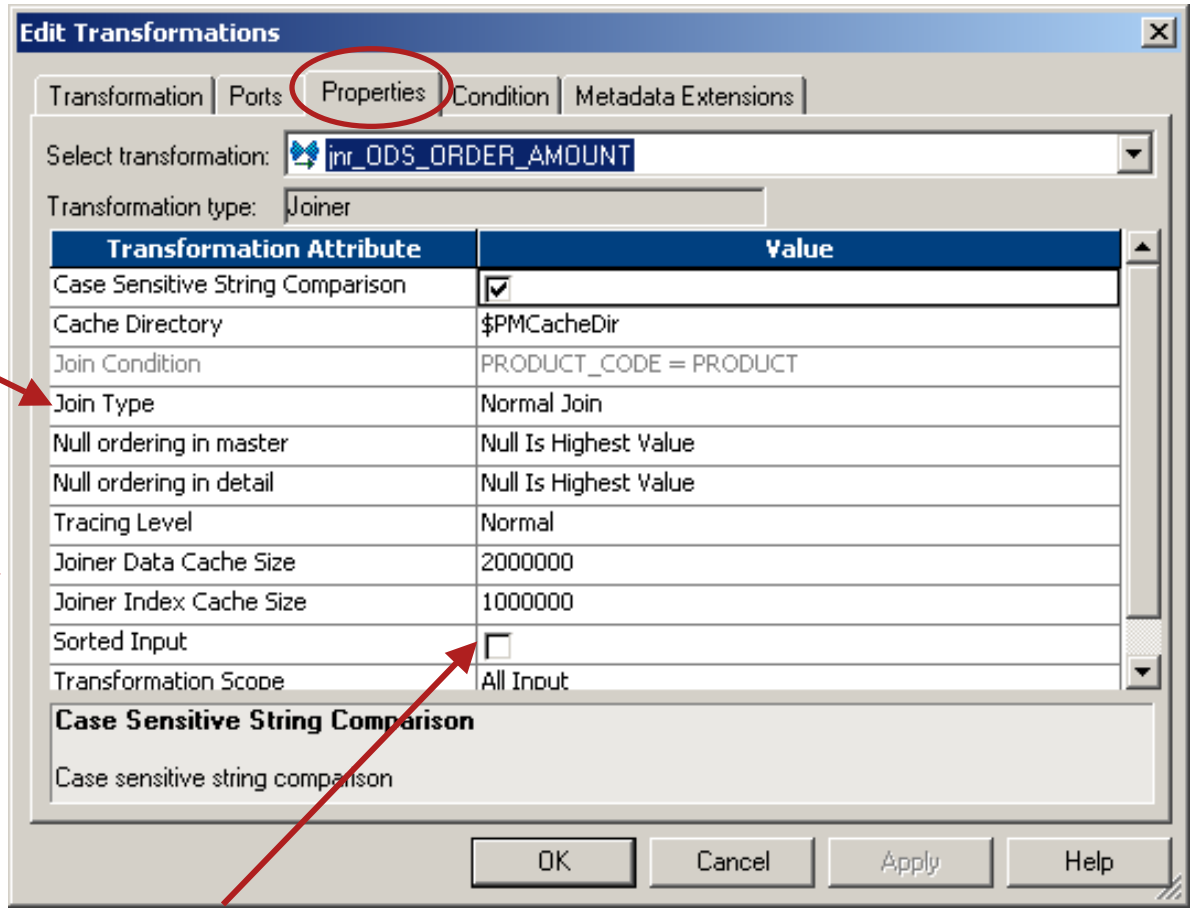


Joiner Properties

Join types:

- Normal (inner)
- Master outer
- Detail outer
- Full outer

Set Joiner Caches

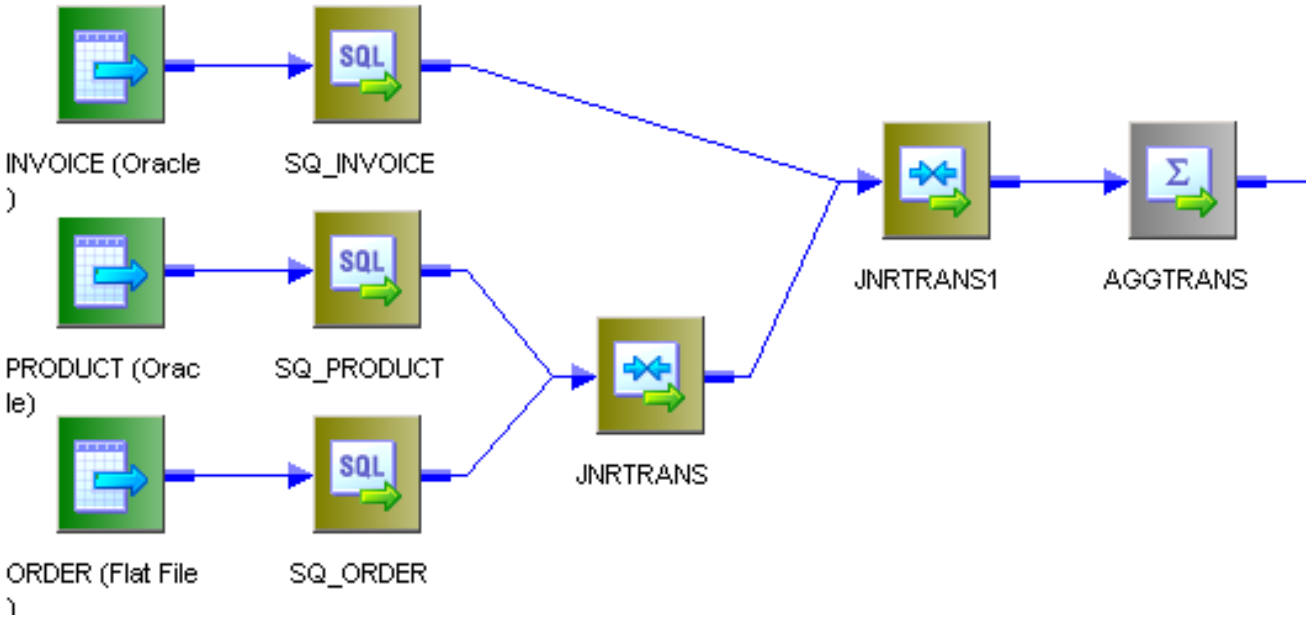


Joiner can accept sorted data (configure the join condition to use the sort origin ports)

INFORMATICA

Nested Joins

Used to join three or more heterogeneous sources



Mid-Mapping Join (Unsorted)

- The unsorted Joiner does not accept input in the following situations:
  - Both input pipelines begin with the same Source Qualifier
  - Both input pipelines begin with the same Joiner
- The sorted Joiner does not have these restrictions.

INFORMATICA

Lab 7 – Heterogeneous Join, Aggregator, and Sorter

INFORMATICA

Lookup Transformation

INFORMATICA

Lookup Transformation

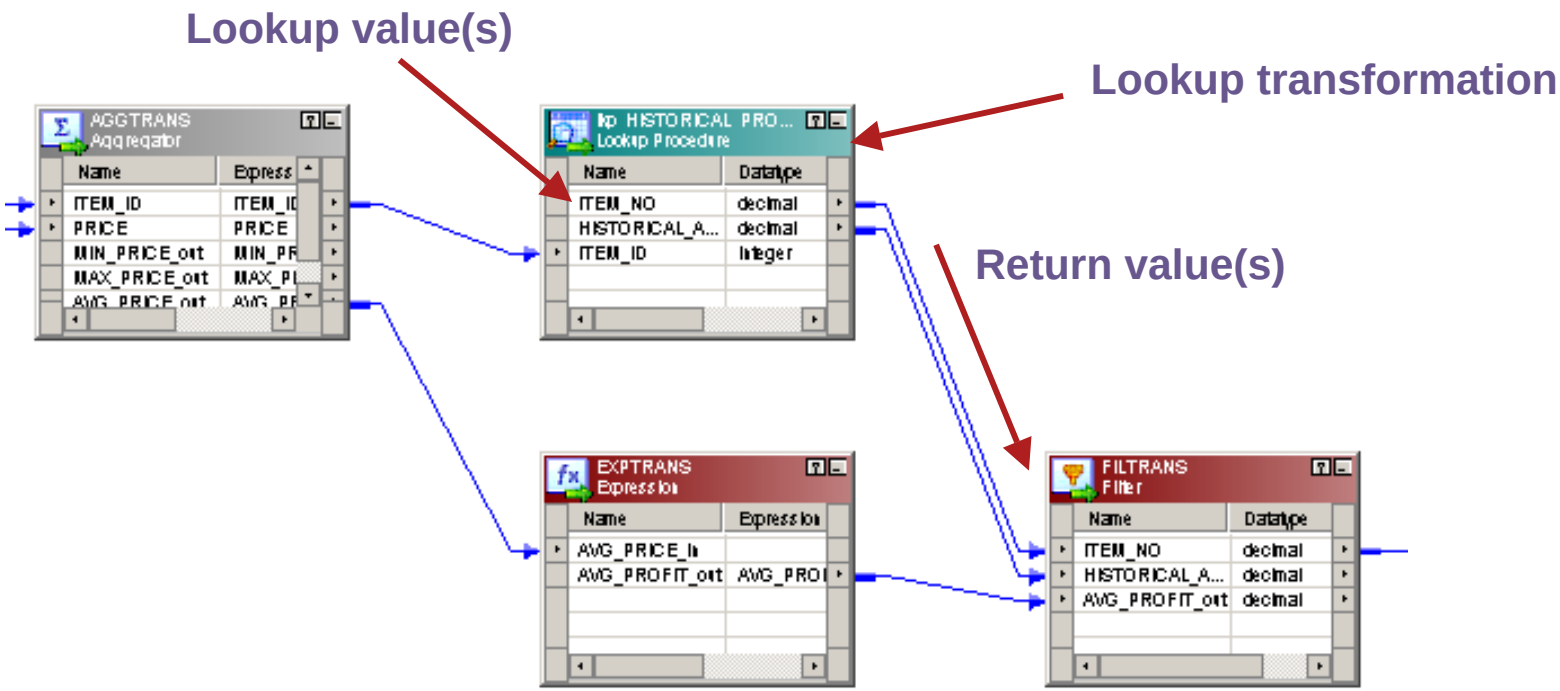
By the end of this section you will be familiar with:

- Lookup principles
- Lookup properties
- Lookup conditions
- Lookup techniques
- Caching considerations
- Persistent caches

INFORMATICA

How a Lookup Transformation Works

- For each mapping row, one or more port values are looked up in a database table or flat file
- If a match is found, one or more table values are returned to the mapping. If no match is found, NULL is returned



INFORMATICA

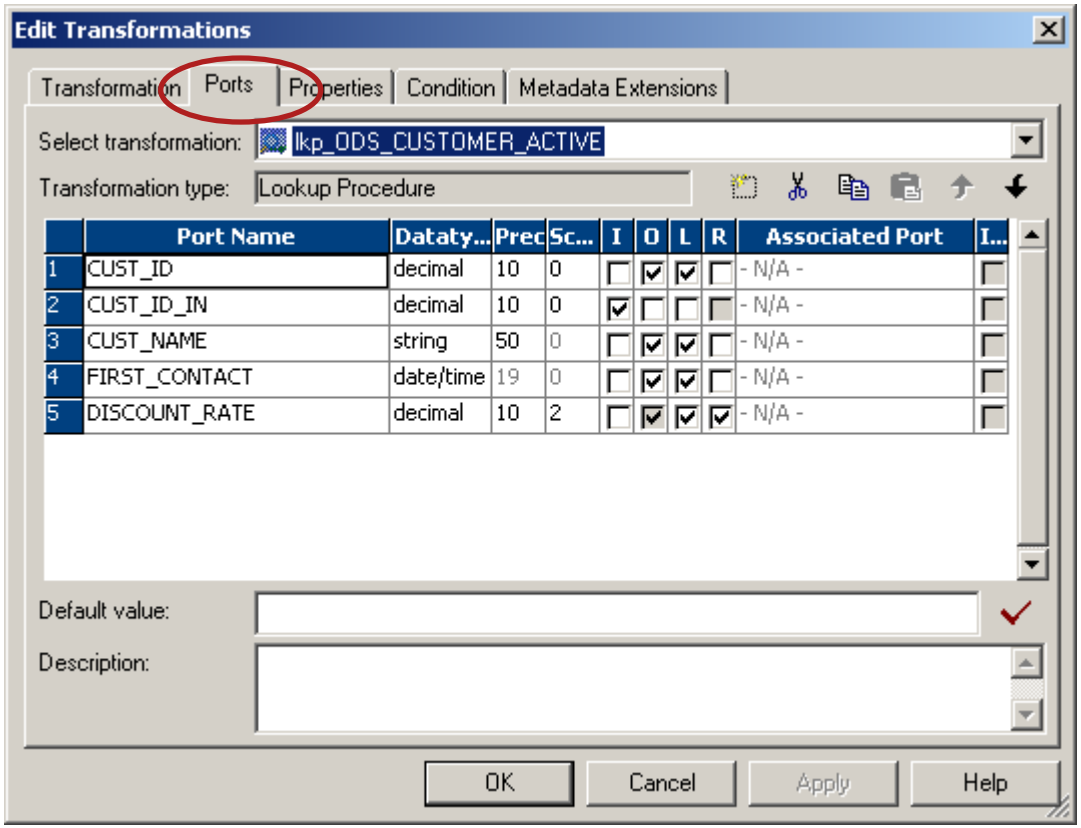
Lookup Transformation

Looks up values in a database table or flat file and provides data to other components in a mapping

Ports
- Mixed
- “L” denotes Lookup port
- “R” denotes port used as a return value (unconnected Lookup only – see later)

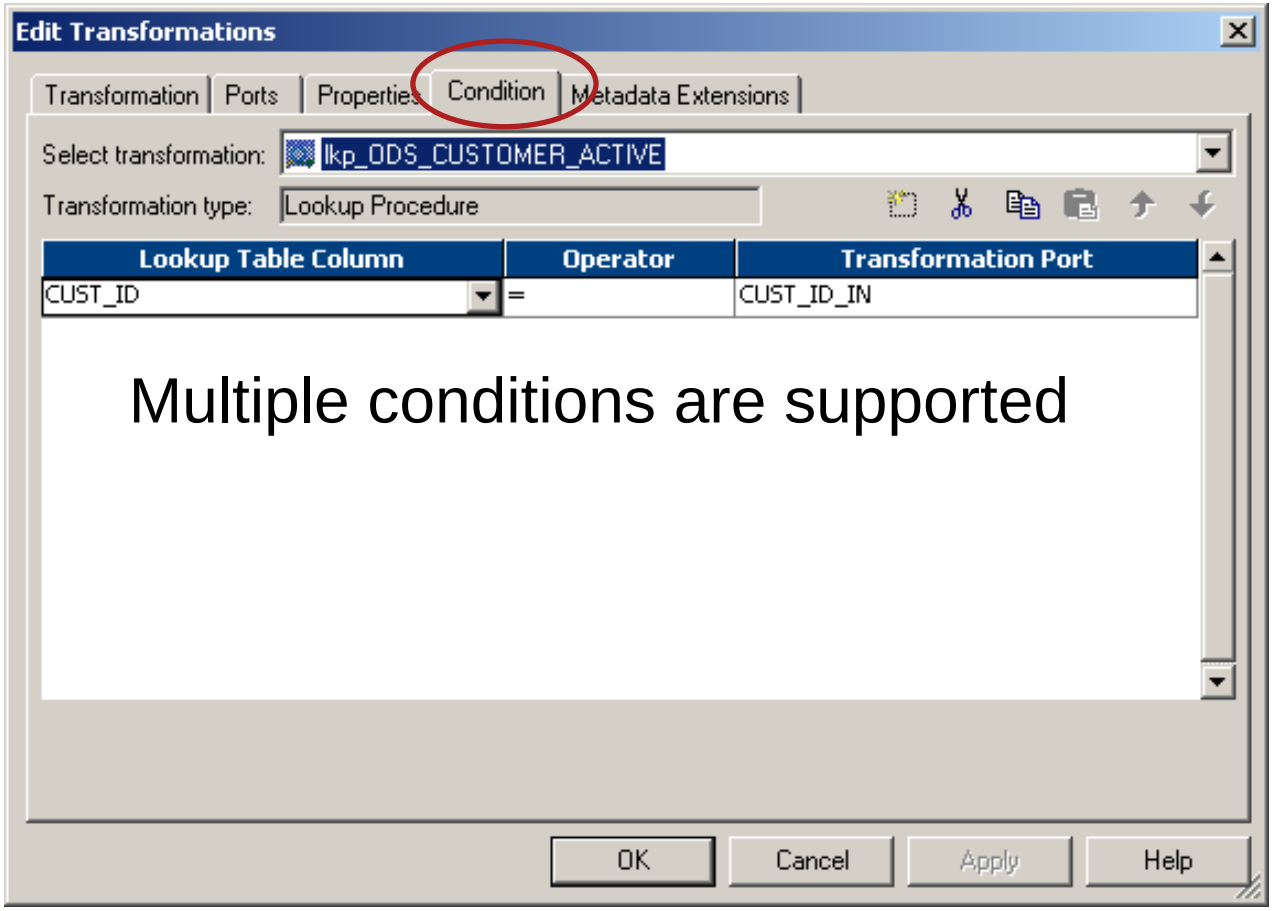
Specify the Lookup Condition

Usage
- Get related values
- Verify if records exists or if data has changed



INFORMATICA

Lookup Conditions



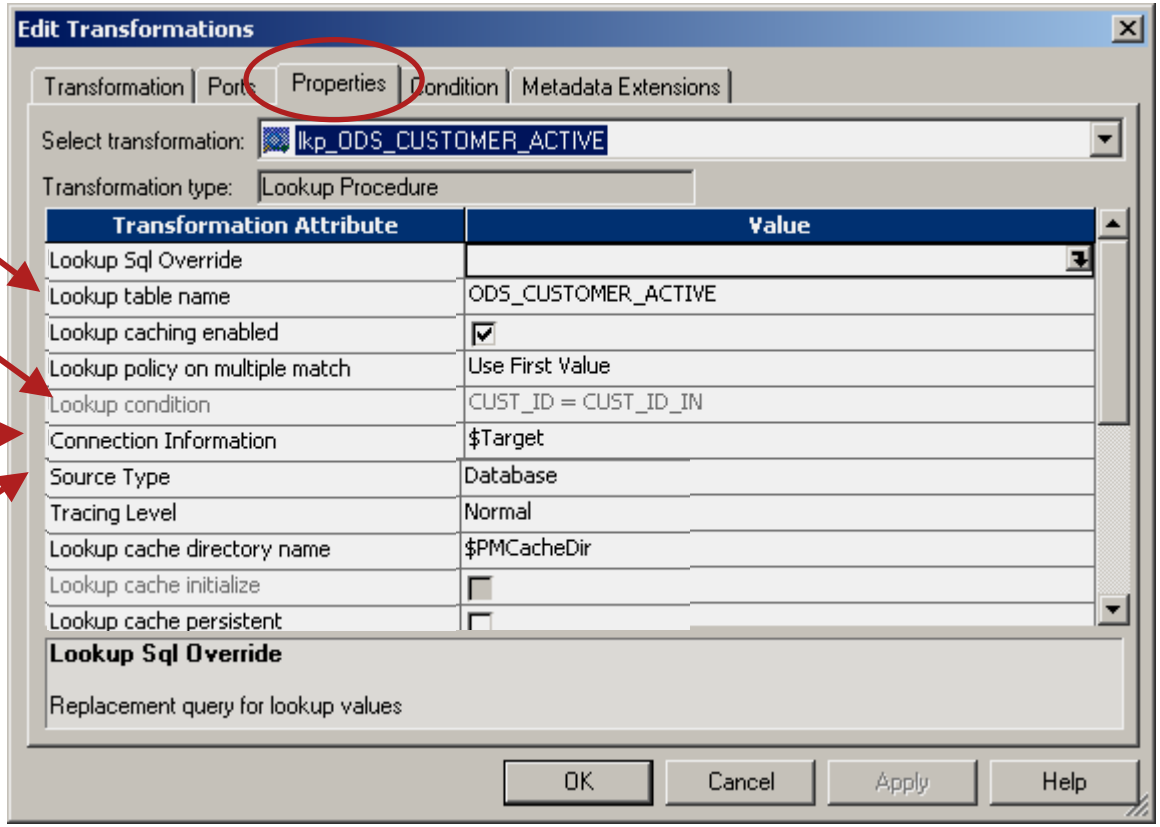
Lookup Properties

Lookup table name

Lookup condition

Native database connection object name

Source type: Database or Flat File

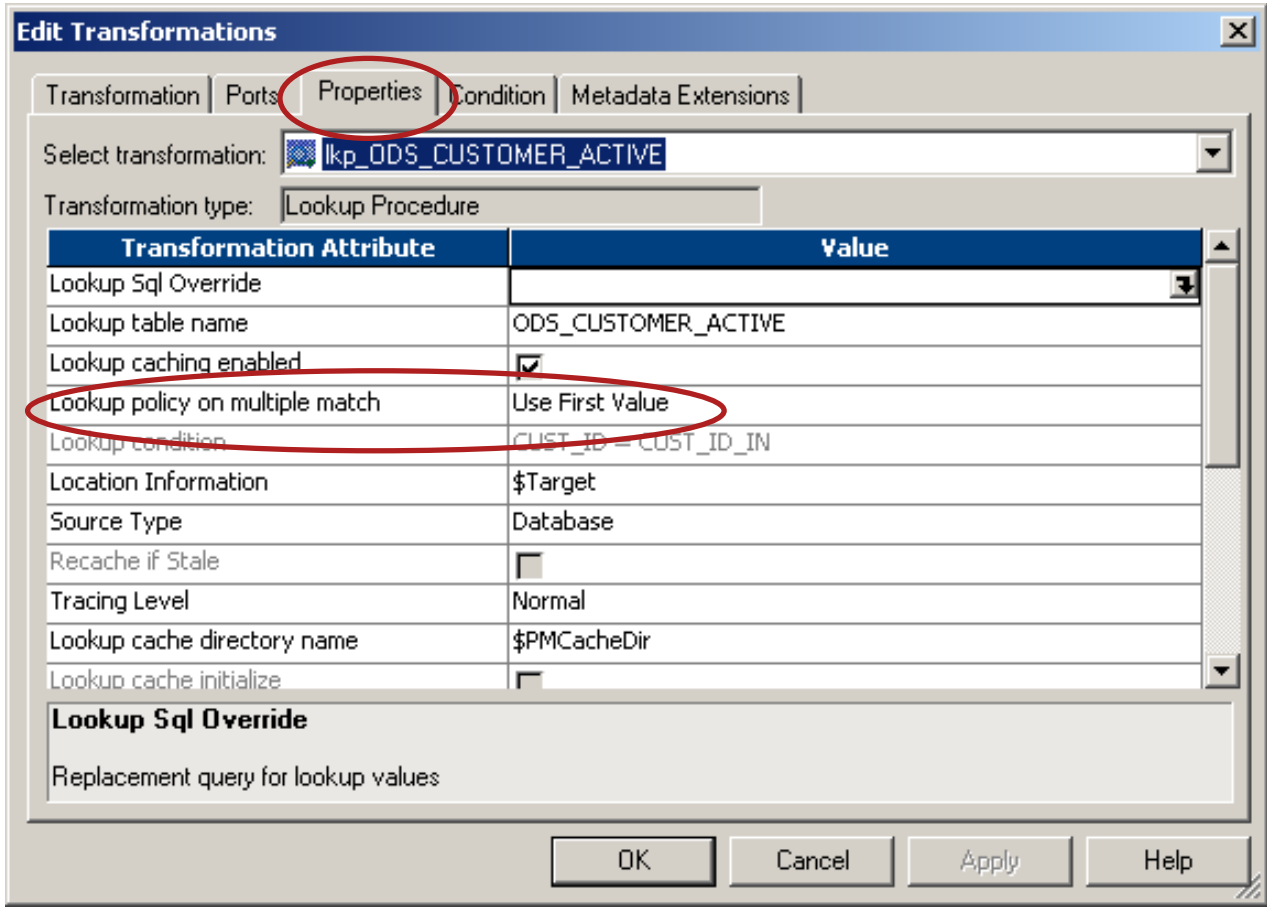


INFORMATICA

Lookup Properties cont'd

Policy on multiple match:

- Use first value
- Use last value
- Report error



INFORMATICA

Lookup Caching

Caching can significantly impact performance

- Cached
  - Lookup table data is cached locally on the Server
  - Mapping rows are looked up against the cache
  - Only one SQL SELECT is needed
- Uncached
  - Each Mapping row needs one SQL SELECT
- Rule Of Thumb: Cache if the number (and size) of records in the Lookup table is small relative to the number of mapping rows requiring the lookup

INFORMATICA

Persistent Caches

- By default, Lookup caches are not persistent; when the session completes, the cache is erased
- Cache can be made persistent with the Lookup properties
- When Session completes, the persistent cache is stored on the server hard disk
- The next time Session runs, cached data is loaded fully or partially into RAM and reused
- A named persistent cache may be shared by different sessions
- Can improve performance, but “stale” data may pose a problem

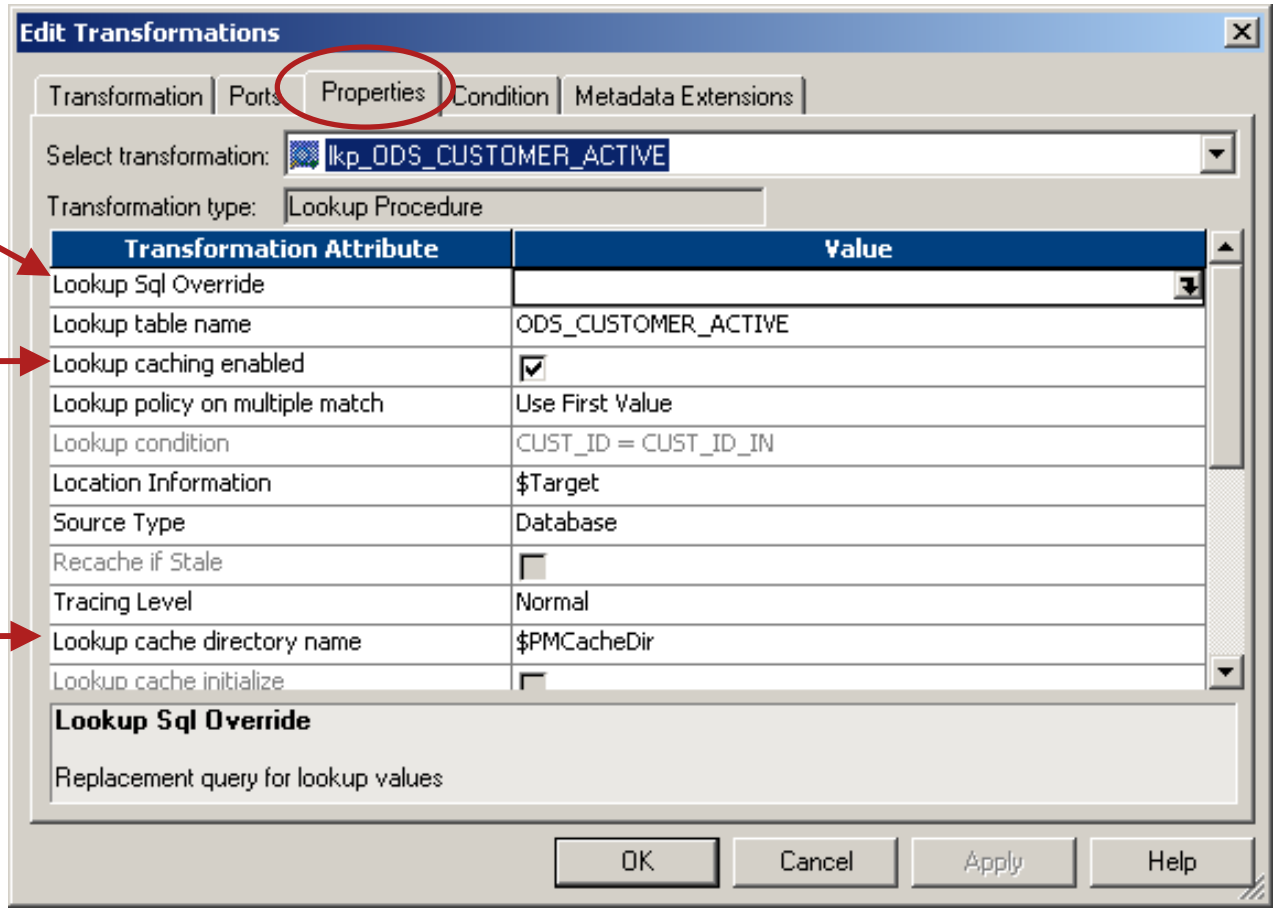
INFORMATICA

Lookup Caching Properties

Override Lookup SQL option

Toggle caching

Cache directory



INFORMATICA

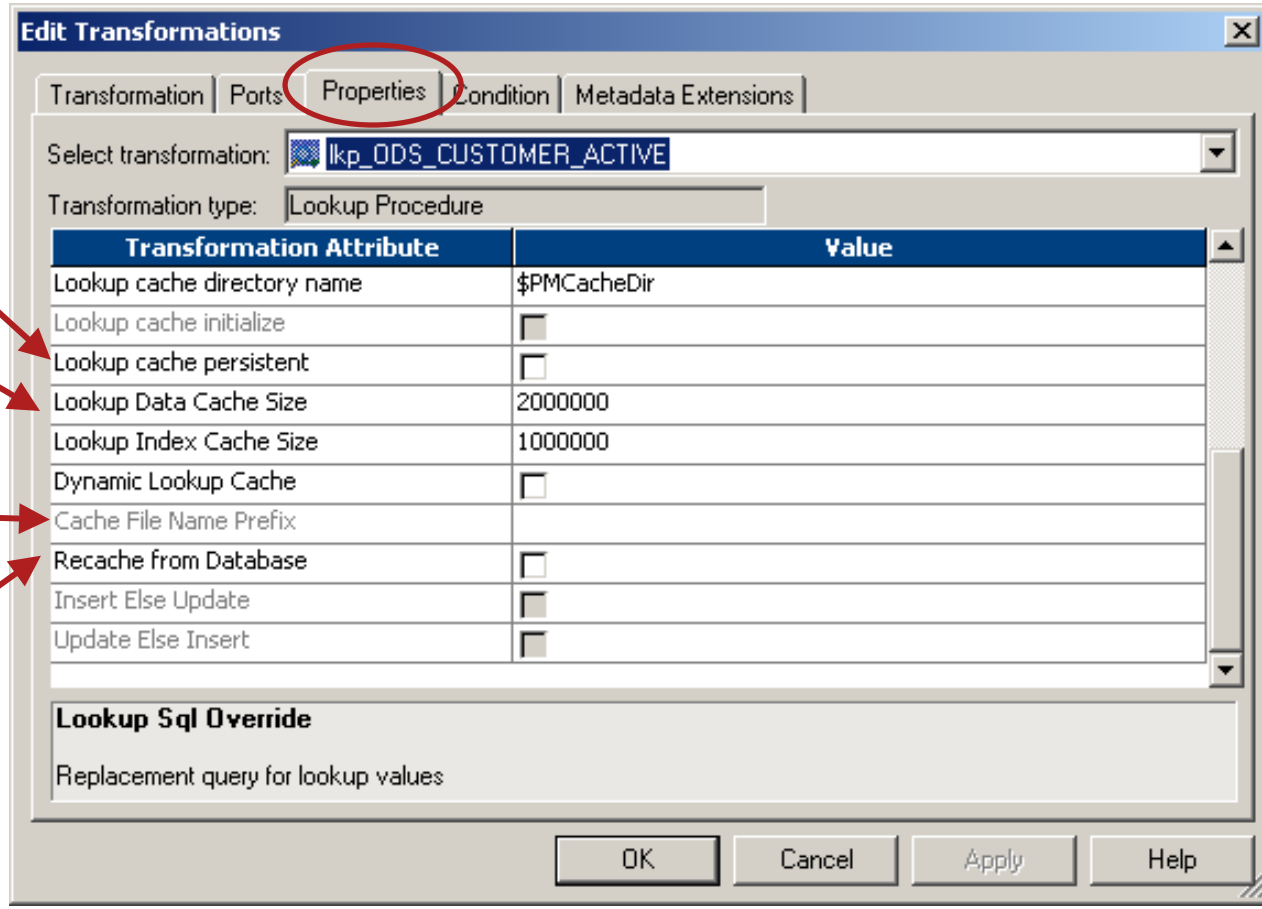
Lookup Caching Properties (cont'd)

Make cache persistent

Set Lookup cache sizes

Set prefix for persistent cache file name

Reload persistent cache



INFORMATICA

Lab 8 – Basic Lookup

INFORMATICA

Target Options

INFORMATICA

Target Options

By the end of this section you will be familiar with:

- Default target load type
- Target properties
- Update override
- Constraint-based loading

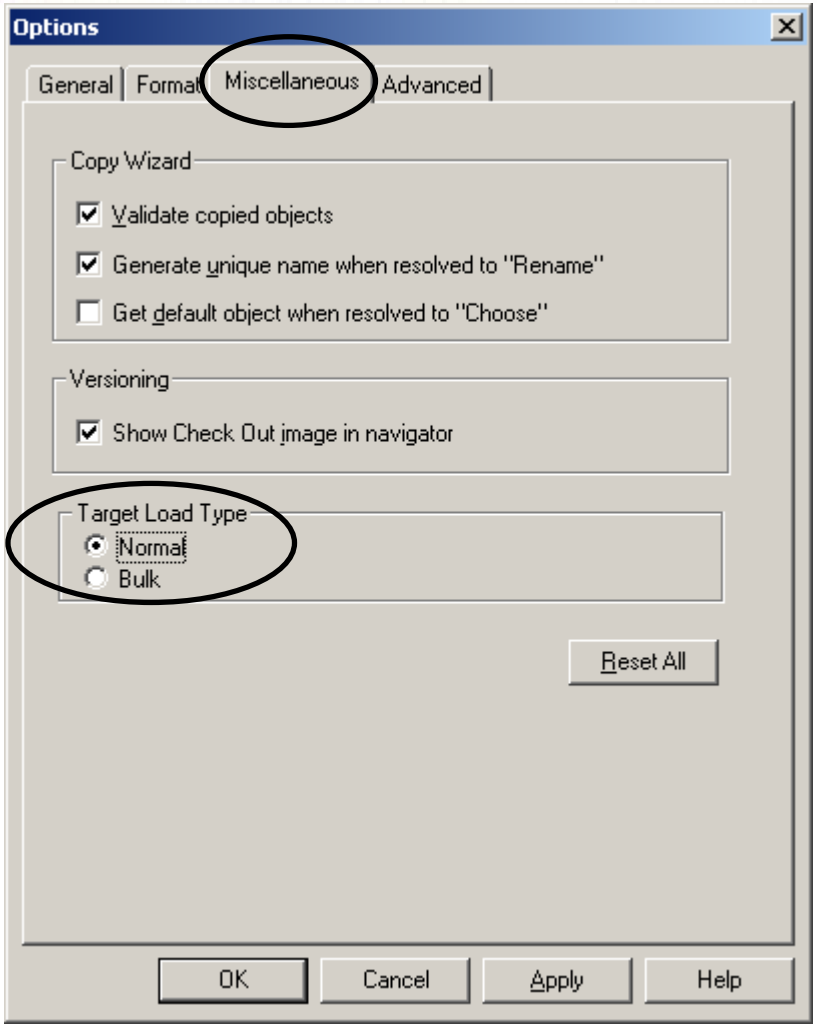
INFORMATICA

Setting Default Target Load Type

Set Target Load Type default in Workflow Manager
Tools => Options

- Normal (usual in development)
- Bulk (usual in production)

Can override in individual target properties.



INFORMATICA

Target Properties

Edit Tasks: Mappings Tab

Session Task

Select target instance

Target load type

Row loading operations

Error handling

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_ODS_CUSTOMER_ACTIVE_01

Task type: Session

Connections
Memory Properties
Files and Directories
Sources
SQ_INVOICE
Targets
ODS_CUSTOMER_A
Transformations
lk_ODS_CUSTOMER_
lkp_CUSTOMER

Transf... | Partitions

Start Page

Targets

Writers

Instance	Writers
ODS_CUSTOMER_ACTIVE	Relational Writer

Connections

Properties — Show Session Level Properties

Attribute	Value
Target load type	Normal
Insert	☑
Update as Update	☑
Update as Insert	☐
Update else Insert	☐
Delete	☑
Truncate target table option	☑
Reject file directory	$PMBadFileDir\
Reject filename	ods_customer_active1.bad

OK | Cancel | Apply | Help

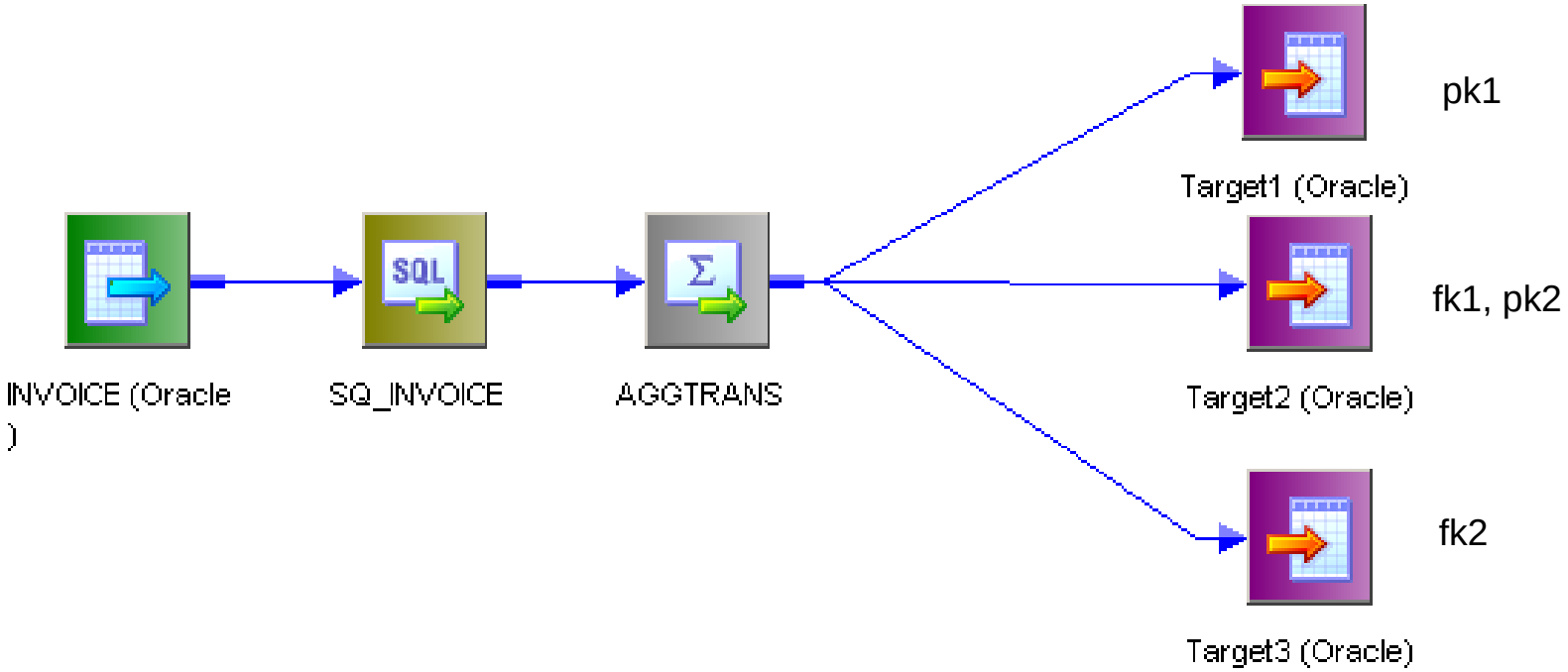
INFORMATICA

WHERE Clause for Update and Delete

- PowerCenter uses the primary keys defined in the Warehouse Designer to determine the appropriate SQL WHERE clause for updates and deletes
- Update SQL
  - UPDATE <target> SET <col> = <value>
    WHERE <primary key> = <pkvalue>
  - The only columns updated are those which have values linked to them
  - All other columns in the target are unchanged
  - The WHERE clause can be overridden via Update Override
- Delete SQL
  - DELETE from <target> WHERE <primary key> = <pkvalue>
- SQL statement used will appear in the Session log file

INFORMATICA

Constraint-based Loading



To maintain referential integrity, primary keys must be loaded before their corresponding foreign keys: in order Target1, Target2, Target 3

INFORMATICA

Setting Constraint-based Loading

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_ODS_EMPLOYEE_TYPE_01

Task type: Session (Reusable)

Config Name: default_session_config

Attribute	Value	UnOverride
Advanced		
Constraint based load ordering	☑	Revert
Cache LOOKUP() function	☑	
Default buffer block size	64000	
Line Sequential buffer length	1024	
Log Options		
Save session log by	Session runs	
Save session log for these runs	0	
Error handling		
Stop on errors	0	
Override tracing	None	
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	

Constraint based load ordering

Constraint based load ordering

OK | Cancel | Apply | Help

INFORMATICA

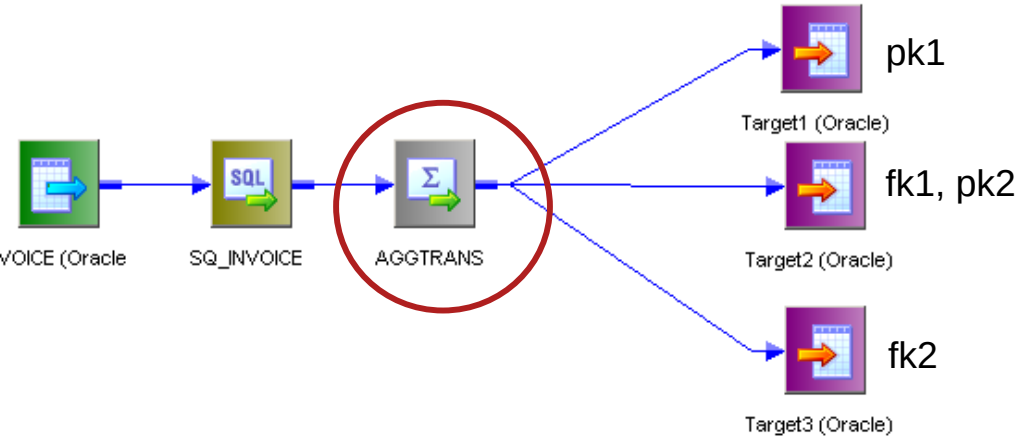
Constraint-based Loading – Terminology

- Active transformation
  - Can operate on groups of data rows and/or can change the number of rows on the data flow
  - Examples: Source Qualifier, Aggregator, Joiner, Sorter, Filter
- Active source
  - Active transformation that generates rows
  - Cannot match an output row with a distinct input row
  - Examples: Source Qualifier, Aggregator, Joiner, Sorter
  - (The Filter is NOT an active source)
- Active group
  - Group of targets in a mapping being fed by the same active source

INFORMATICA

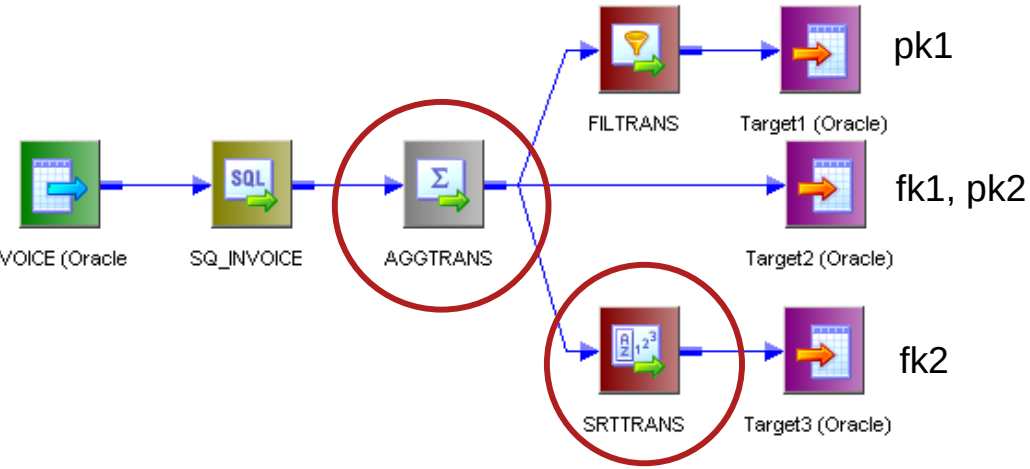
Constraint-Based Loading – Restrictions

Cannot have two active groups



Example 1

With only *one* Active source, rows for Targets1, 2, and 3 will be loaded properly and maintain referential integrity



Example 2

With *two* Active sources, it is not possible to control whether rows for Target3 will be loaded before or after those for Target2

INFORMATICA

Lab 9 – Deleting Rows

INFORMATICA

Update Strategy Transformation

INFORMATICA

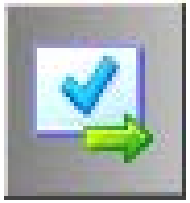
Update Strategy Transformation

By the end of this section you will be familiar with:

- Update Strategy functionality
- Update Strategy expressions

INFORMATICA

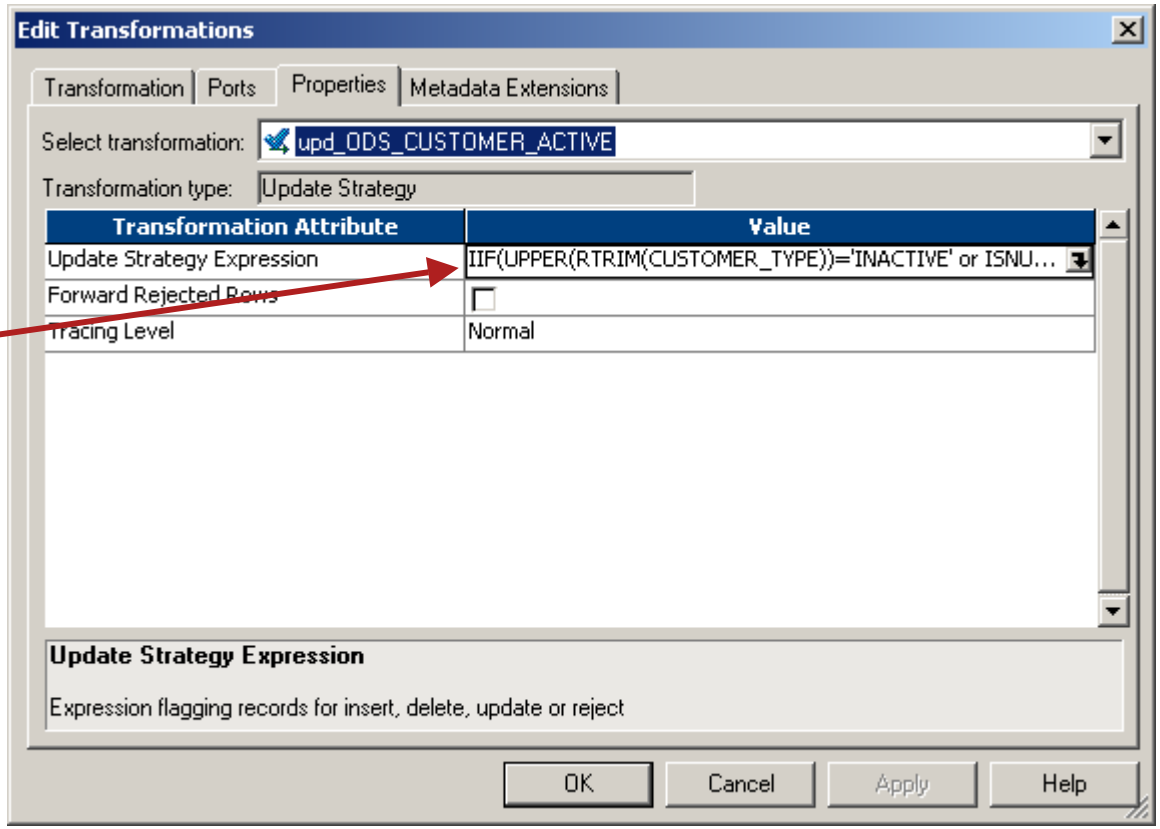
Update Strategy Transformation



Used to specify how each individual row will be used to update target tables (insert, update, delete, reject)

Ports
- All input / output
- Specify the Update Strategy Expression – IIF or DECODE logic determines how to handle the record

Example
- Updating Slowly Changing Dimensions



INFORMATICA

Update Strategy Expressions

IIF (score > 69, DD_INSERT, DD_DELETE)

- Expression is evaluated for each row
- Rows are “tagged” according to the logic of the expression
- Appropriate SQL (DML) is submitted to the target database: insert, delete or update
- DD_REJECT means the row will not have SQL written for it. Target will not “see” that row
- “Rejected” rows may be forwarded through Mapping

INFORMATICA

Lab 10 – Data Driven Operations

INFORMATICA

Lab 11 – Incremental Update

INFORMATICA

Lab 12 – Features and Techniques II

INFORMATICA

Router Transformation

INFORMATICA

Router Transformation

By the end of this section you will be familiar with:

- Router functionality
- Router filtering groups
- How to apply a Router in a Mapping

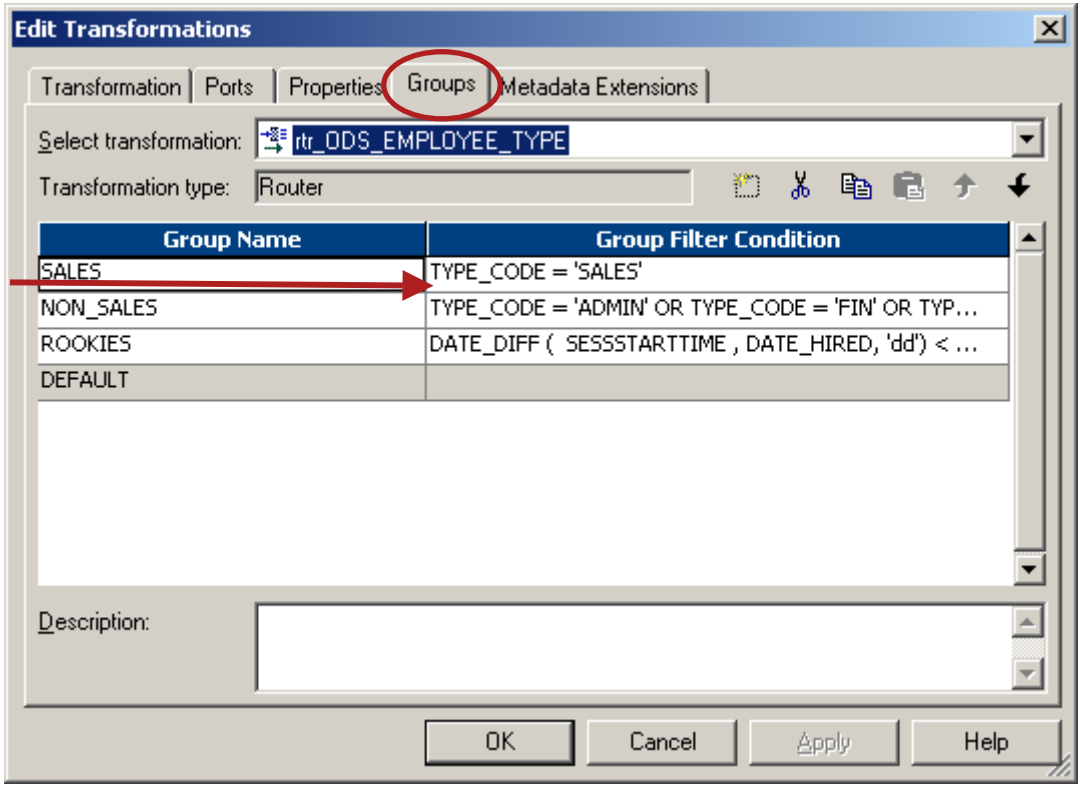
INFORMATICA

Router Transformation

Rows sent to multiple filter conditions

Ports
- All input/output
- Specify filter conditions for each Group

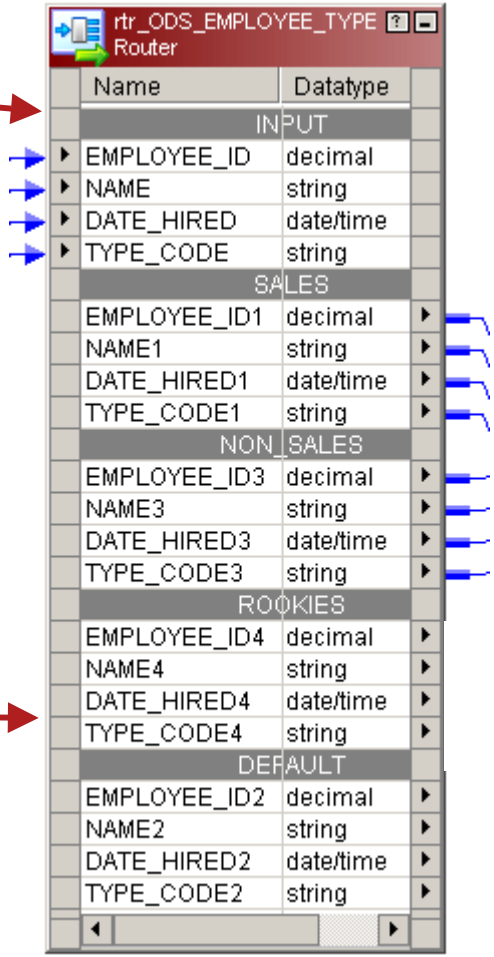
Usage
- Link source data in one pass to multiple filter conditions



INFORMATICA

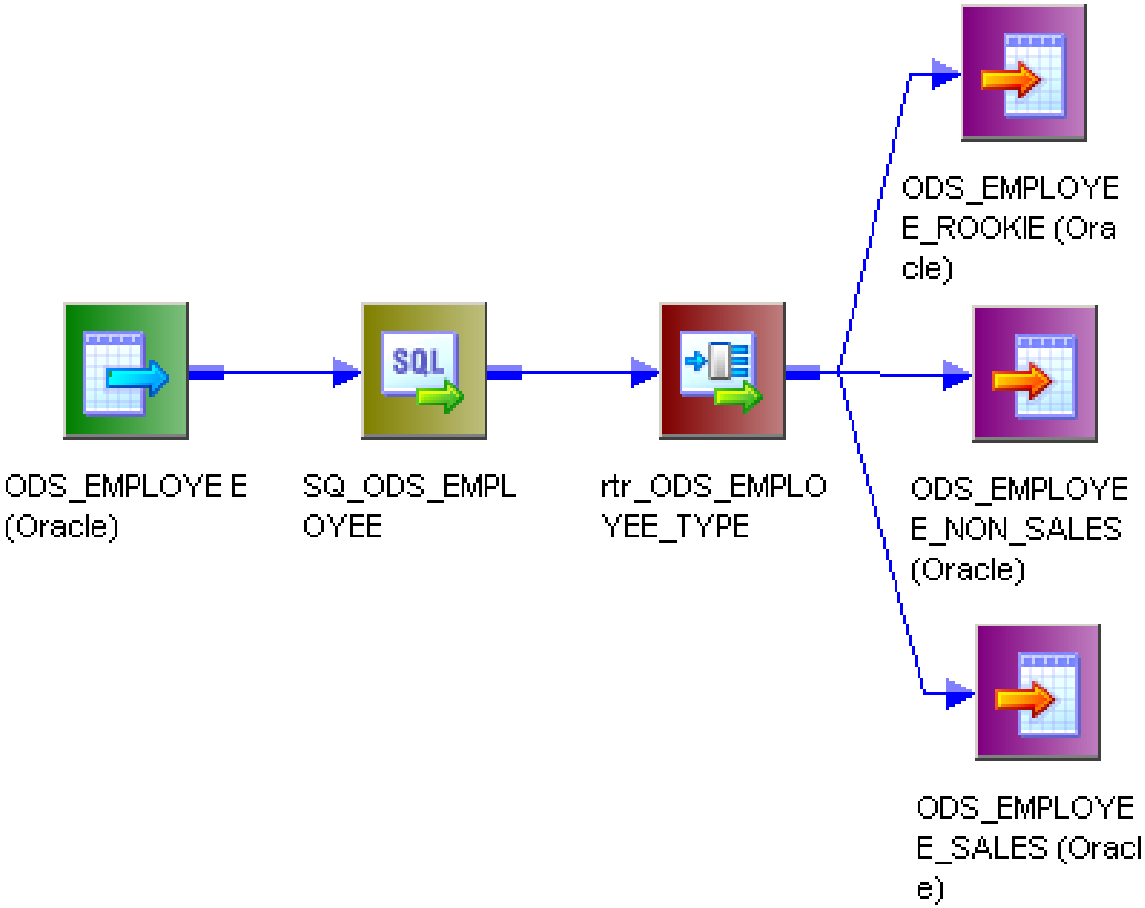
Router Groups

- Input group (always one)
- User-defined groups
- Each group has one condition
- ALL group conditions are evaluated for EACH row
- One row can pass multiple conditions
- Unlinked Group outputs are ignored
- Default group (always one) *can* capture rows that fail all Group conditions



INFORMATICA

Router Transformation in a Mapping



INFORMATICA

Lab 13 – Router

INFORMATICA

Sequence Generator Transformation

INFORMATICA

Sequence Generator Transformation

By the end of this section you will be familiar with:

- Sequence Generator functionality
- Sequence Generator properties

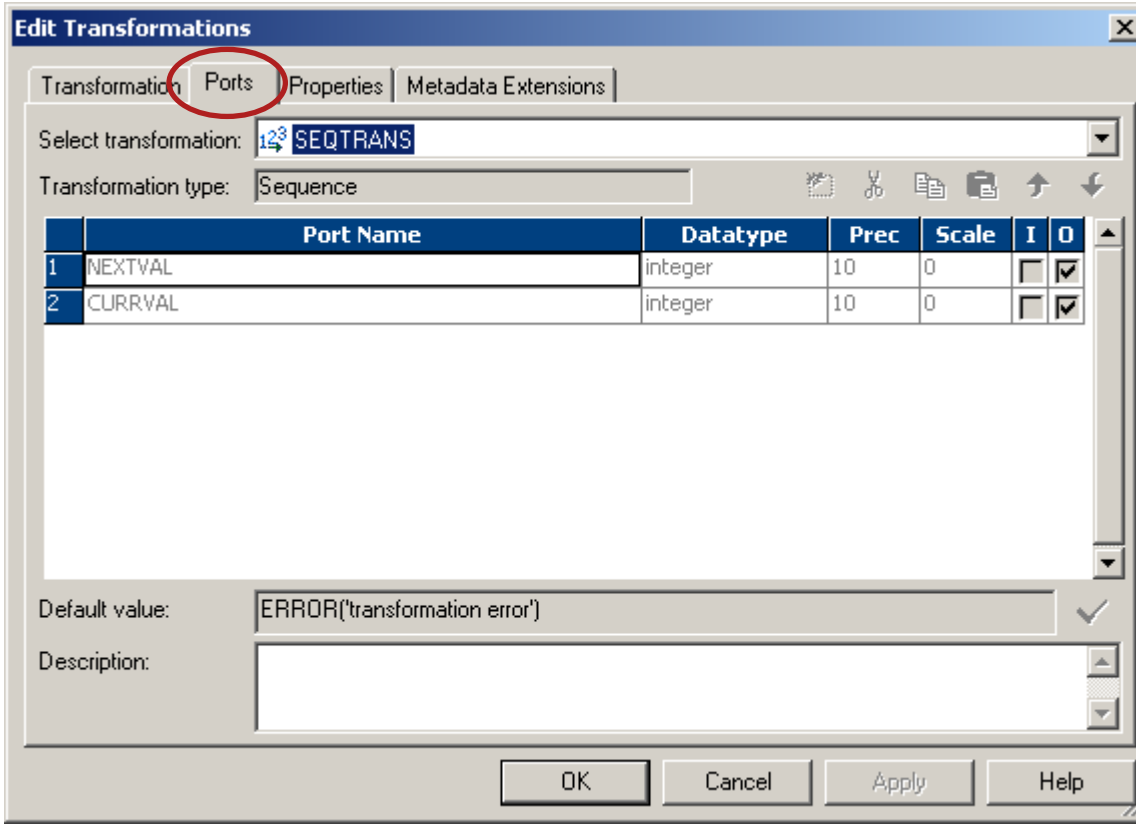
INFORMATICA

Sequence Generator Transformation

Generates unique keys for any port on a row

Ports
- Two predefined output ports, NEXTVAL and CURRVAL
- No input ports allowed

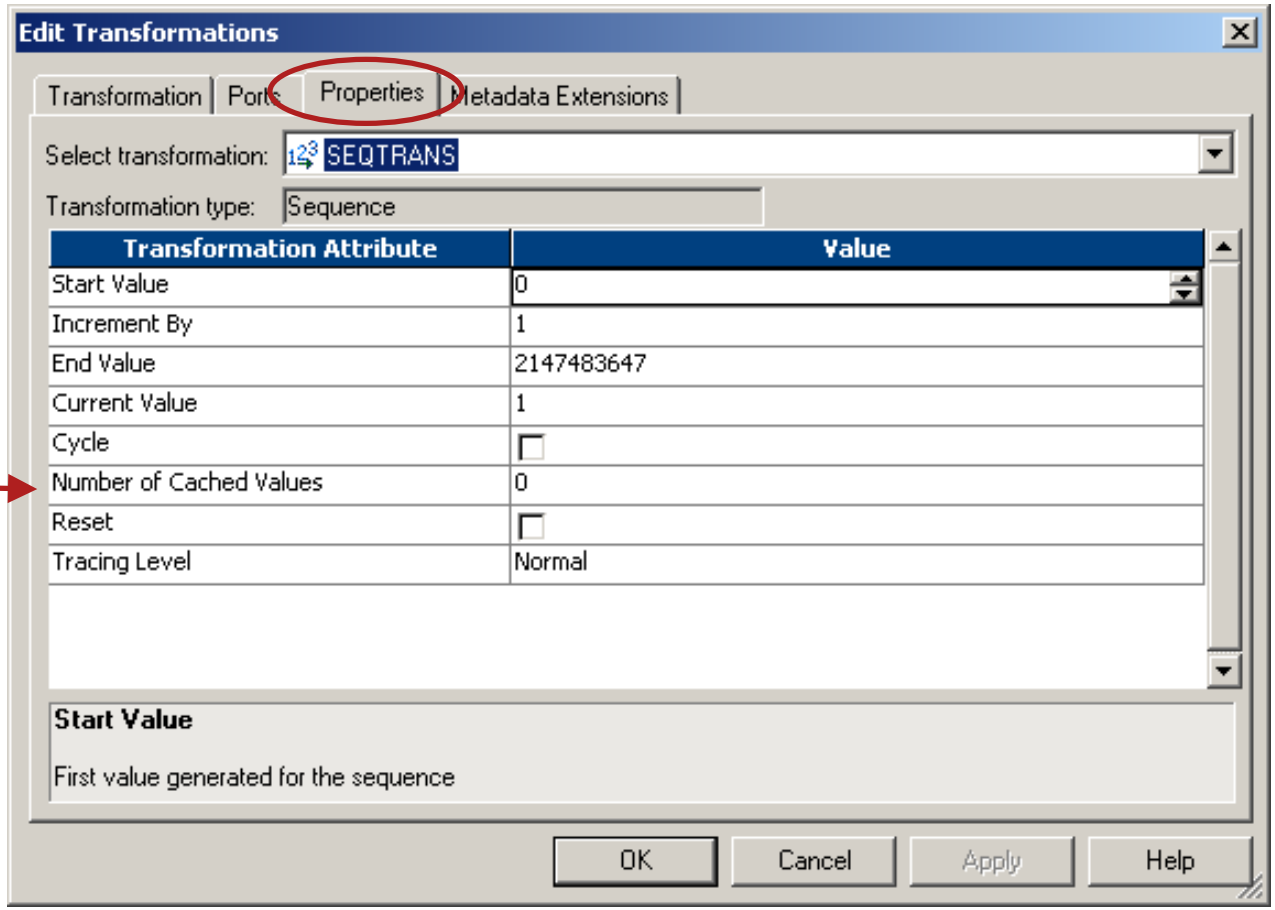
Usage
- Generate sequence numbers
- Shareable across mappings



INFORMATICA

Sequence Generator Properties

Number of cached values



INFORMATICA

Parameters and Variables

INFORMATICA

Parameters and Variables

By the end of this section you will understand:

- System variables
- Mapping parameters and variables
- Parameter files

INFORMATICA

System Variables

SYSDATE

- Provides current datetime on the Informatica Server machine
  - Not a static value

SESSSTARTTIME

- Returns the system date value on the Informatica Server
  - Used with any function that accepts transformation date/time datatypes
  - Not to be used in a SQL override
  - Has a constant value

$$$SessStartTime

- Returns the system date value *as a string* Uses system clock on machine hosting Informatica Server
  - format of the string is database type dependent
  - Used in SQL override
  - Has a constant value

INFORMATICA

Mapping Parameters and Variables

- Apply to all transformations within one Mapping
- Represent declared values
- Variables can change in value during run-time
- Parameters remain constant during run-time
- Provide increased development flexibility
- Defined in Mapping menu
- Format is $$*VariableName* or $$*ParameterName*

INFORMATICA

Mapping Parameters and Variables

Sample declarations

Set datatype

User-defined names

Set aggregation type

Declare Parameters and Variables

Declare the parameters and variables you want to use in the mapping or mapplet.
A parameter represents a constant value defined before mapping run.
A variable represents a value that can be changed during mapping run.

Name	Type	Datatype	Prec	Scale	Aggrega...
$$MIN_RATE	Parameter	decimal	10	2	
$$NEWVARIABLE1	Variable	string	10	0	Max
$$MAX_RATE	Parameter	decimal	10	2	

Initial value:

Description:

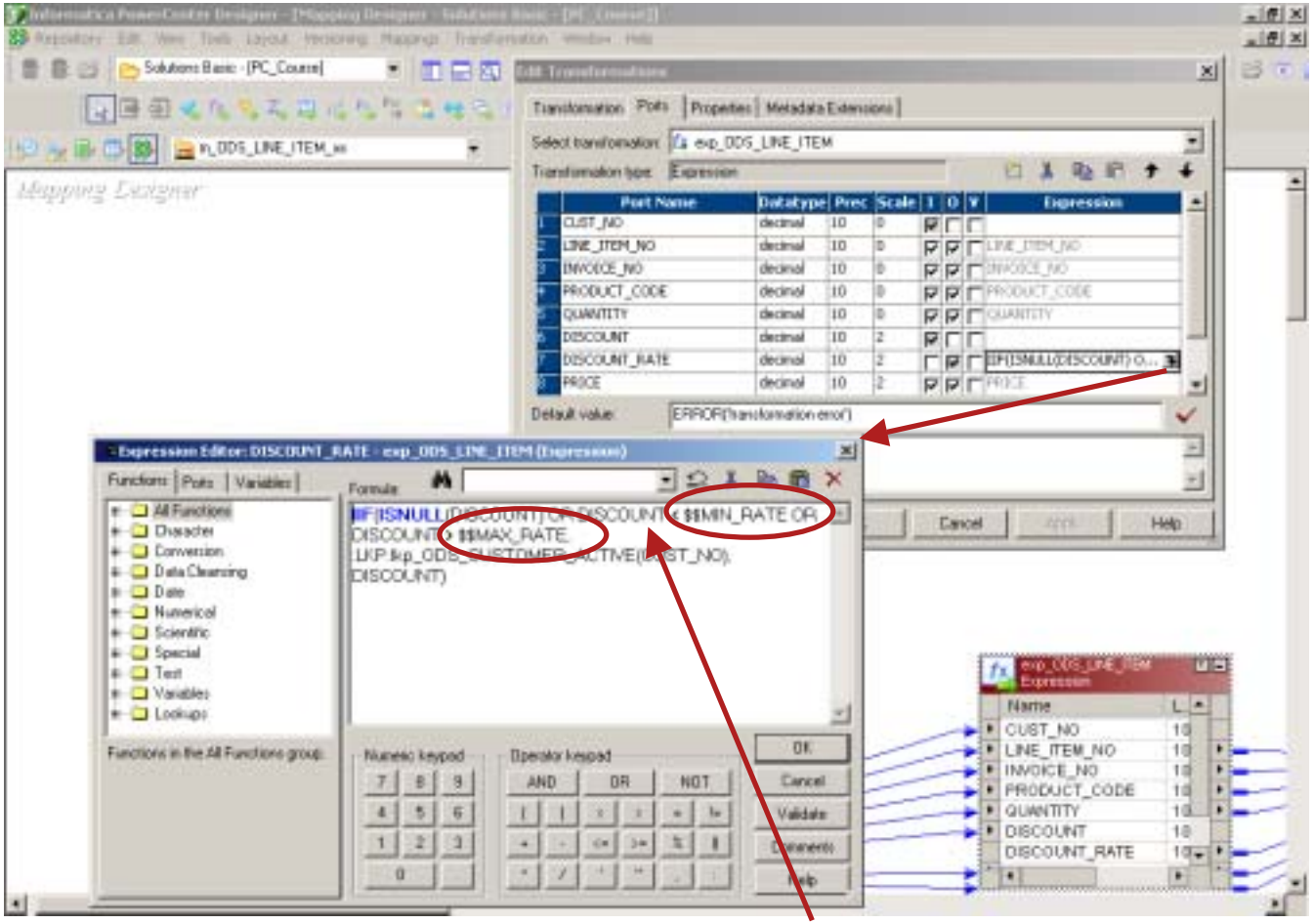
OK Cancel Help

Set optional initial value

Declare Variables and Parameters in the Designer Mappings/Mapplets menu

INFORMATICA

Mapping Parameters and Variables



Apply parameters or variables in formula

INFORMATICA

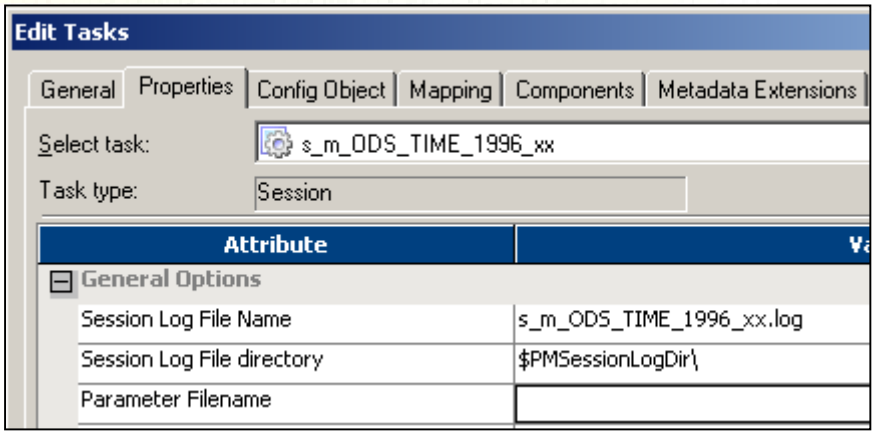
Functions to Set Mapping Variables

- **SETMAXVARIABLE($$*Variable,value*)**
  Sets the specified variable to the higher of the current value or the specified value
- **SETMINVARIABLE($$*Variable,value*)**
  Sets the specified variable to the lower of of the current value or the specified value
- **SETVARIABLE($$*Variable,value*)**
  Sets the specified variable to the specified value
- **SETCOUNTVARIABLE($$*Variable*)**
  Increases or decreases the specified variable by the number of rows leaving the function (+1 for each inserted row, -1 for each deleted row, no change for updated or rejected rows)

INFORMATICA

Parameter Files

You can specify a parameter file for a session in the session editor



Parameter file contains folder.session name and initializes each parameter and variable for that session. For example:

```
[Production.s_MonthlyCalculations]
$$State=MA
$$Time=10/1/2000 00:00:00
$InputFile1=sales.txt
$DBConnection_target=sales
$PMSessionLogFile=D:/session logs/firstrun.txt
```


INFORMATICA

Priorities for Initializing Parameters & Variables

1. Parameter file
2. Repository value
3. Declared initial value
4. Default value

INFORMATICA

Unconnected Lookups

INFORMATICA

Unconnected Lookups

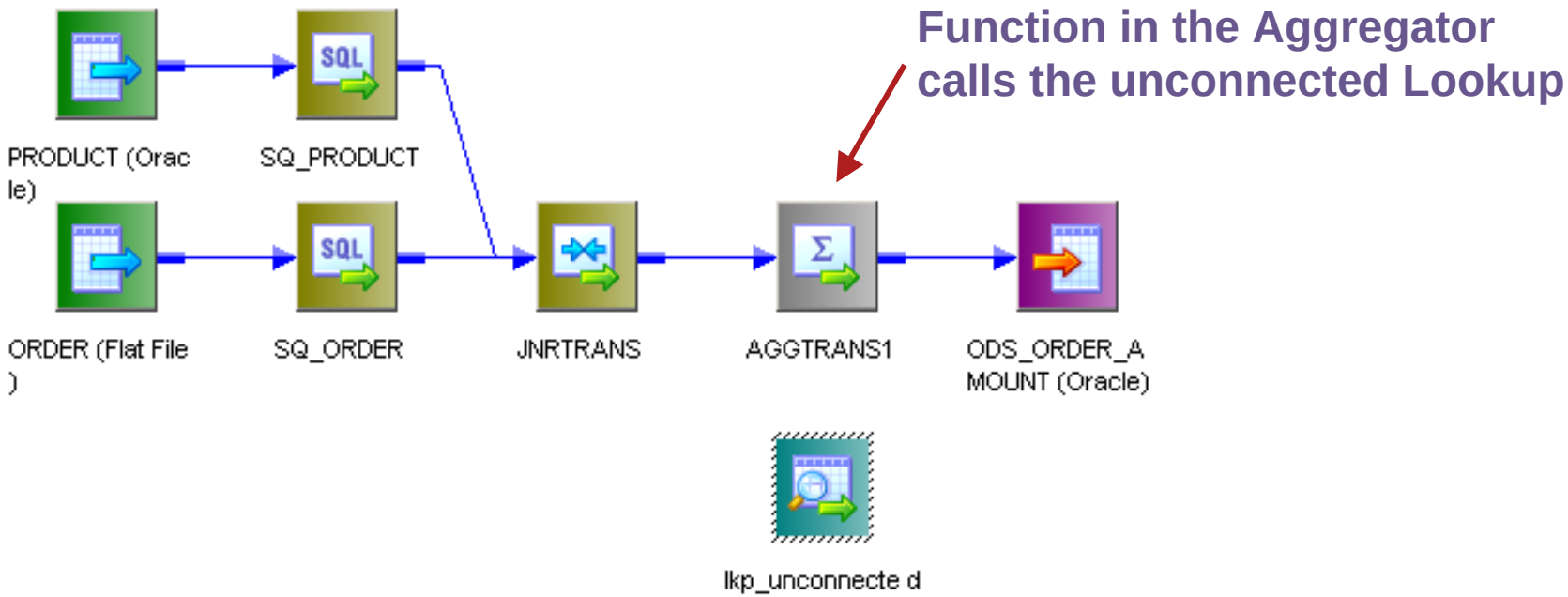
By the end of this section you will know:

- Unconnected Lookup technique
- Unconnected Lookup functionality
- Difference from Connected Lookup

INFORMATICA

Unconnected Lookup

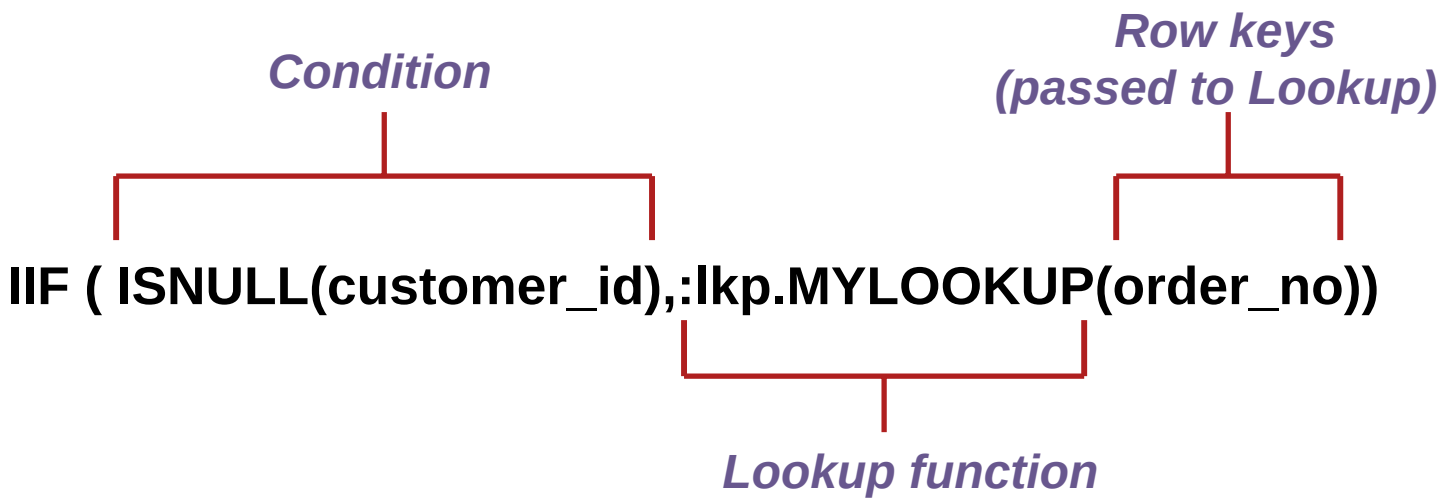
- Physically unconnected from other transformations – NO data flow arrows leading to or from an unconnected Lookup
- Lookup data is called from the point in the Mapping that needs it
- Lookup function can be set within any transformation that supports expressions



INFORMATICA

Unconnected Lookup Technique

- Use lookup lookup function within a conditional statement



- Condition is evaluated for each row but Lookup function is called only if condition satisfied

INFORMATICA

Unconnected Lookup Advantage

- Data lookup is performed *only* for those rows which require it. Substantial performance can be gained

EXAMPLE: A Mapping will process 500,000 rows. For two percent of those rows (10,000) the item_id value is NULL. Item_ID can be derived from the SKU_NUMB.

IIF (ISNULL(item_id), :lkp.MYLOOKUP (sku_numb))

Condition
***(true for* 2 percent of all rows)**

Lookup
(called only when condition is true)

Net savings = 490,000 lookups

INFORMATICA

Unconnected Lookup Functionality

One Lookup port value may be returned for each Lookup

Edit Transformations

Transformation | Ports | Properties | Condition | Metadata Extensions

Select transformation: lkp_ODS_CUSTOMER_ACTIVE

Transformation type: Lookup Procedure

	Port Name	Dataty...	Prec	Sc...	I	O	L	R	Associated Port	I...
1	CUST_ID_in	decimal	10	0	☑	☐	☐	☐	- N/A -	☐
2	CUST_ID	decimal	10	0	☐	☑	☑	☐	- N/A -	☐
3	CUST_NAME	string	50	0	☐	☐	☑	☐	- N/A -	☐
4	FIRST_CONTACT	date/time	19	0	☐	☐	☑	☐	- N/A -	☐
5	CUSTOMER_STATE	string	2	0	☐	☐	☑	☐	- N/A -	☐
6	CUSTOMER_ZIP	string	10	0	☐	☐	☑	☐	- N/A -	☐
7	CUSTOMER_TYPE	string	10	0	☐	☐	☑	☐	- N/A -	☐
8	DISCOUNT_RATE	decimal	10	2	☐	☑	☑	☑	- N/A -	☐

Default value:

Description:

OK | Cancel | Apply | Help

Must check a Return port in the Ports tab, else fails at runtime

INFORMATICA

Connected versus Unconnected Lookups

CONNECTED LOOKUP	UNCONNECTED LOOKUP
Part of the mapping data flow	Separate from the mapping data flow
Returns multiple values (by linking output ports to another transformation)	Returns one value - by checking the Return (R) port option for the output port that provides the return value
Executed for every record passing through the transformation	Only executed when the lookup function is called
More visible, shows where the lookup values are used	Less visible, as the lookup is called from an expression within another transformation
Default values are used	Default values are ignored

INFORMATICA

Lab 14 – Straight Load

Lab 15 – Conditional Lookup

INFORMATICA

Heterogeneous Targets

INFORMATICA

Heterogeneous Targets

By the end of this section you will be familiar with:

- Heterogeneous target types
- Heterogeneous target limitations
- Target conversions

INFORMATICA

Definition: Heterogeneous Targets

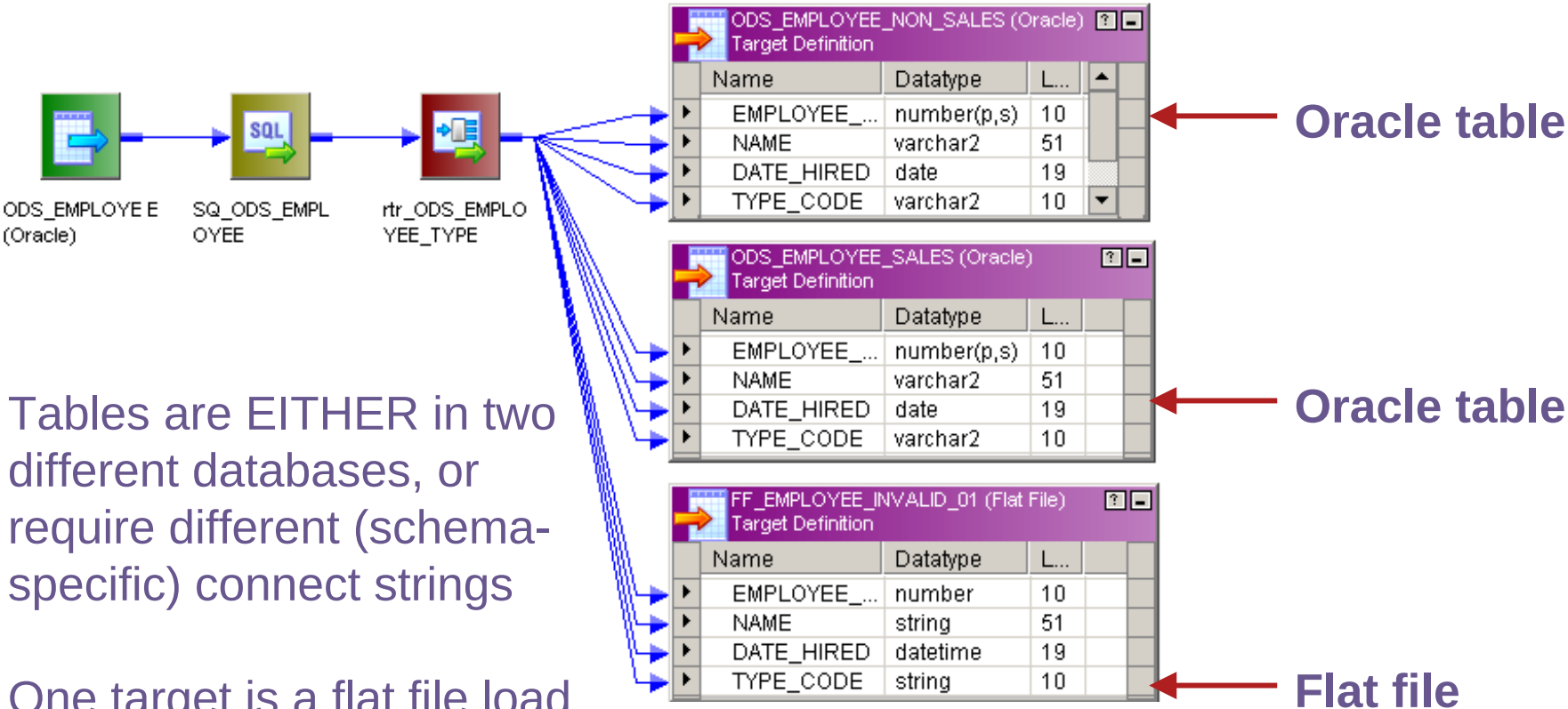
Supported target definition types:

- Relational database
- Flat file
- XML
- SAP BW, PeopleSoft, etc. (via PowerConnects)

Heterogeneous targets are targets within a single Session Task that have different types *or* have different database connections

INFORMATICA

Step One: Identify Different Target Types



Tables are EITHER in two different databases, or require different (schema-specific) connect strings

One target is a flat file load

INFORMATICA

Step Two: Different Database Connections

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_ODS_EMPLOYEE_INVALID_TYPES_01

Task type: Session

- Connections
- Memory Properties
- Files and Directories
- Sources
  - SQ_ODS_EMPLOYEE
- Targets
  - ODS_EMPLOYEE_NON_SALES
  - ODS_EMPLOYEE_SALES
  - FF_EMPLOYEE_INVALID_01
- Transformations
  - rtr_ODS_EMPLOYEE_TYPE

Transformations | Partitions

Start Page

Targets

Writers

Instance	Writers
ODS_EMPLOYEE_N...	Relational Writer
ODS_EMPLOYEE_S...	Relational Writer
FF_EMPLOYEE_INV...	File Writer

Connections

Type	Value
ODS_EMPLOYEE_NON_SALES - DB Connection	
Relational	native_target_01
ODS_EMPLOYEE_SALES - DB Connection	
Relational	native_target_02
FF_EMPLOYEE_INVALID_01 - Connection	
None	

Properties

< Please pick a connection >

OK | Cancel | Apply | Help

The two database connections are different

Flat file requires separate location information

INFORMATICA

Target Type Override (Conversion)

Example: Mapping has SQL Server target definitions. Session Task can be set to load Oracle tables instead, using an Oracle database connection.

Only the following overrides are supported:

- Relational target to flat file target
- Relational target to any other relational database type
- SAP BW target to a flat file target

CAUTION: If target definition datatypes are not compatible with datatypes in newly selected database type, modify the target definition

INFORMATICA

Lab 16 – Heterogeneous Targets

INFORMATICA

Mapplets

INFORMATICA

Mapplets

By the end of this section you will be familiar with:

- Mapplet Designer
- Mapplet advantages
- Mapplet types
- Mapplet rules
- Active and Passive Mapplets
- Mapplet Parameters and Variables

INFORMATICA

Mapplet Designer

Informatica PowerCenter Designer - [Mapplet Designer - Solutions Basic - [PC7_Course]]

Repository Edit View Tools Layout Versioning Mapplets Transformation Window Help

Solutions Basic - [PC7_Course]

100%

mplt_PROFIT_PERCENT

Mapplet Designer

ODS_LINE_ITEM (Oracle)

ODS_INVOICE (Oracle)

sq_ODS_INVOICE_ODS_LINE_ITEM

agg_PROFIT_PERCENT

out_PROFIT_PERCENT

Ready

Mapplet Designer Tool

Mapplet Input and Output Transformation Icons

Mapplet Output Transformation

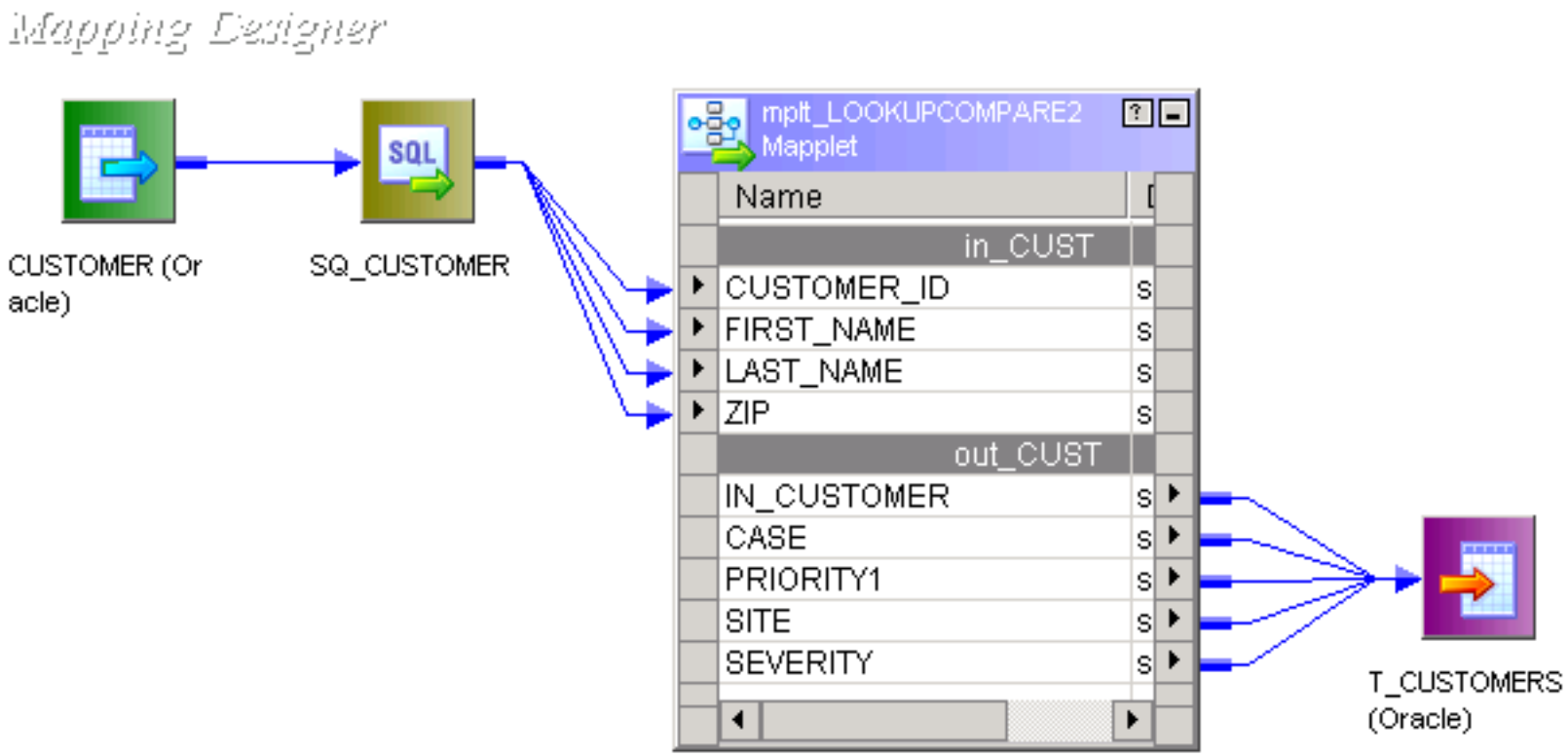
INFORMATICA

Mapplet Advantages

- Useful for repetitive tasks / logic
- Represents a set of transformations
- Mapplets are reusable
- Use an ‘instance’ of a Mapplet in a Mapping
- Changes to a Mapplet are inherited by all instances
- Server expands the Mapplet at runtime

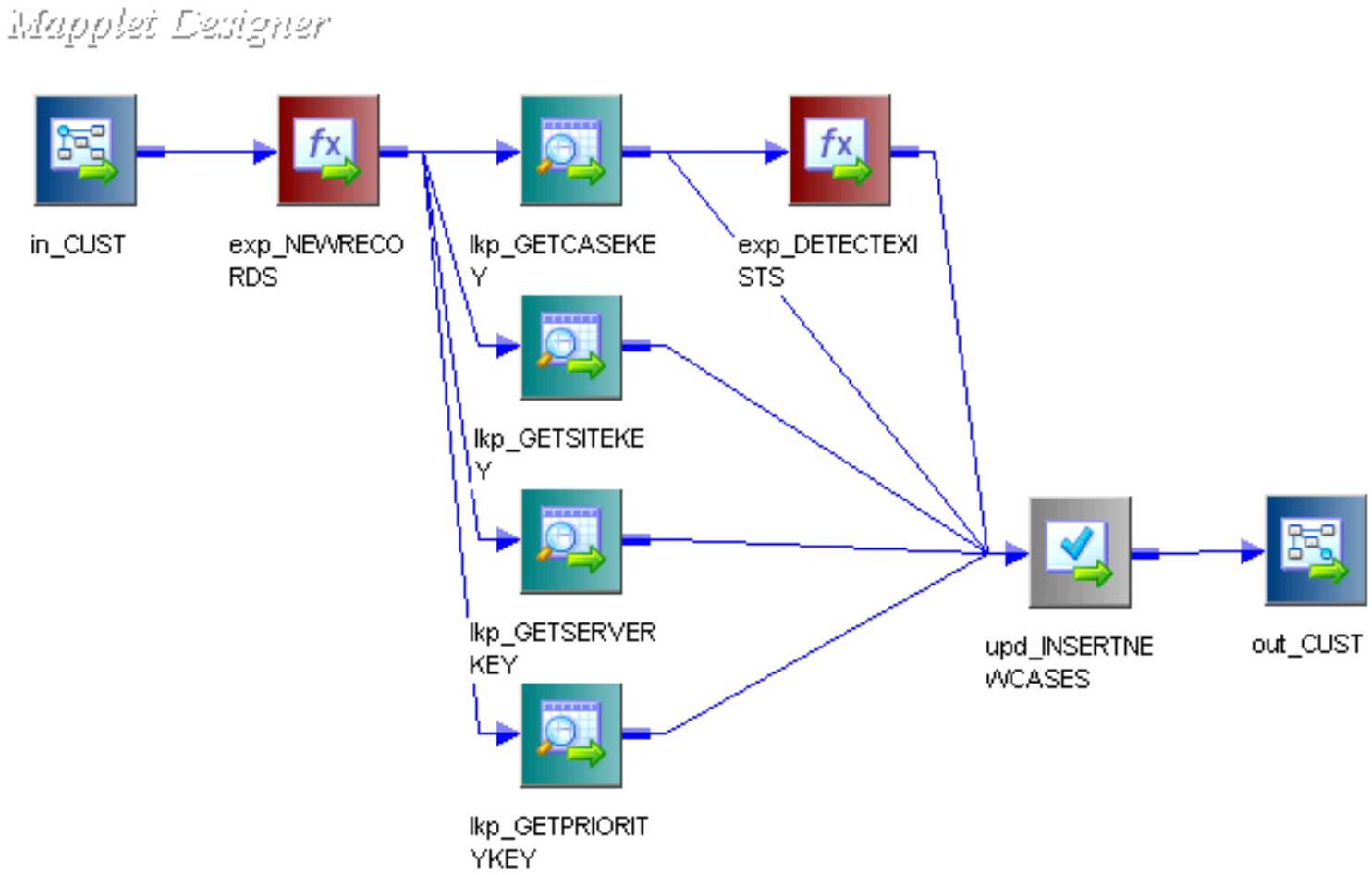
INFORMATICA

A Mapplet Used in a Mapping



INFORMATICA

The “Detail” Inside the Mapplet



INFORMATICA

Unsupported Transformations

Use any transformation in a Mapplet *except:*

- XML Source definitions
- COBOL Source definitions
- Normalizer
- Pre- and Post-Session stored procedures
- Target definitions
- Other Mapplets

INFORMATICA

Mapplet Source Options

- Internal Sources
  - One or more Source definitions / Source Qualifiers *within* the Mapplet
- External Sources
  - Mapplet contains a Mapplet Input transformation
  - Receives data from the Mapping it is used in
- Mixed Sources
  - Mapplet contains one or more of either of a Mapplet Input transformation AND one or more Source Qualifiers
  - Receives data from the Mapping it is used in, AND from the Mapplet

INFORMATICA

Mapplet Input Transformation

Use for data sources outside a Mapplet

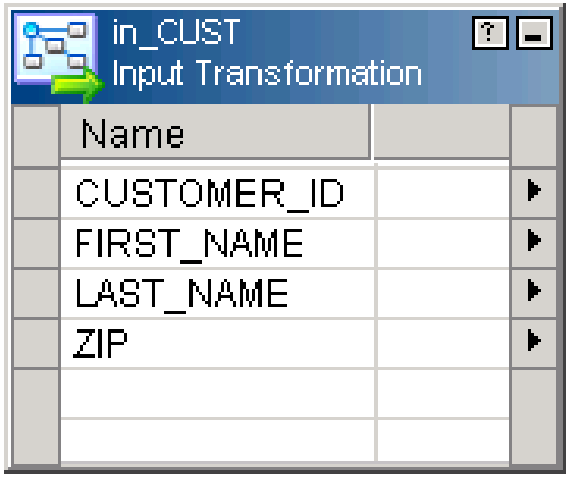
Passive Transformation
Connected

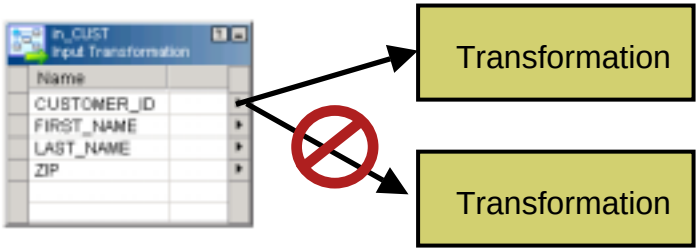
Ports

- Output ports only

Usage

- Only those ports connected from an Input transformation to another transformation will display in the resulting Mapplet





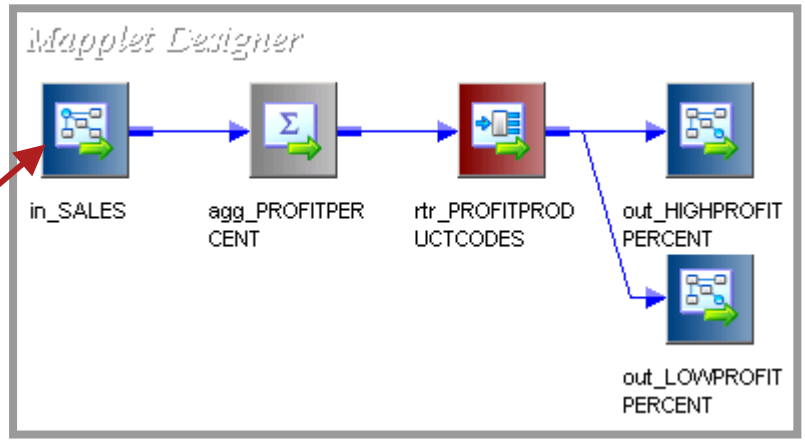
- **Connecting the same port to more than one transformation is disallowed**
- **Pass to an Expression transformation first**

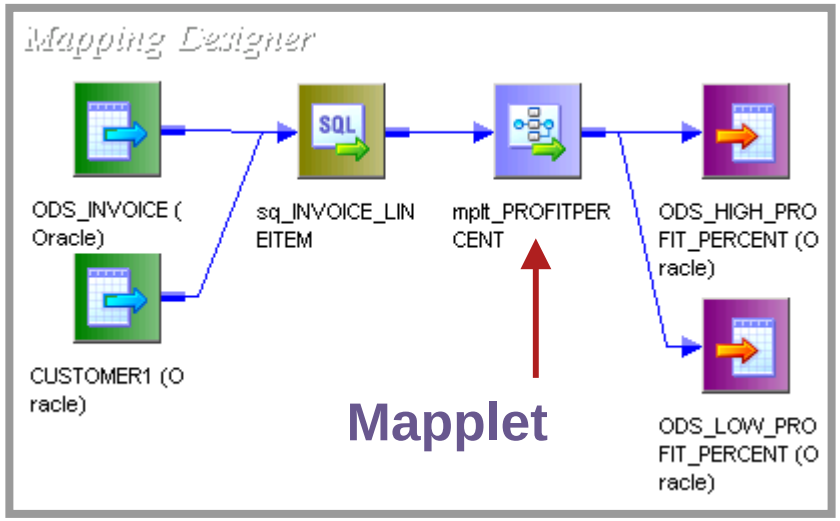
INFORMATICA

Data Source *Outside* a Mapplet

Source data is defined OUTSIDE the Mapplet logic

Mapplet Input Transformation





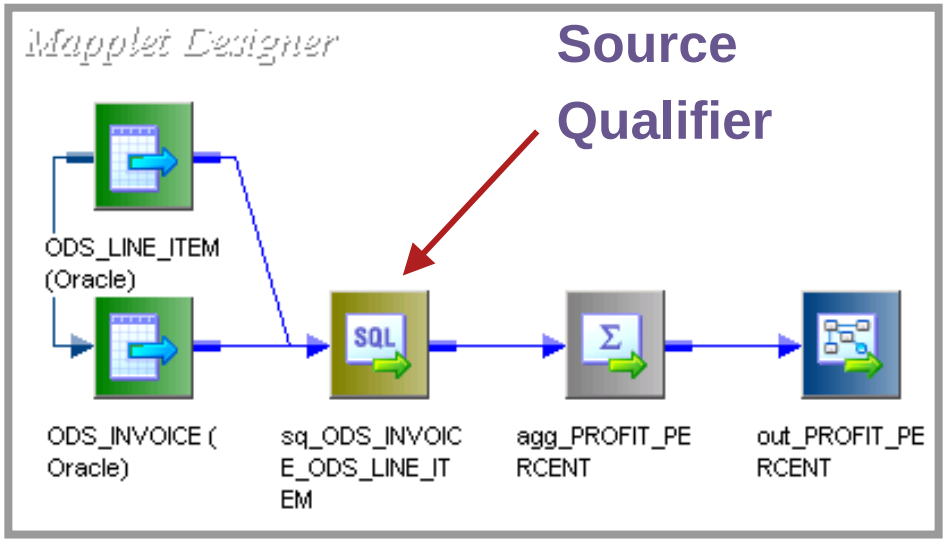
- Resulting Mapplet HAS input ports
- When used in a Mapping, the Mapplet may occur at any point in mid-flow

INFORMATICA

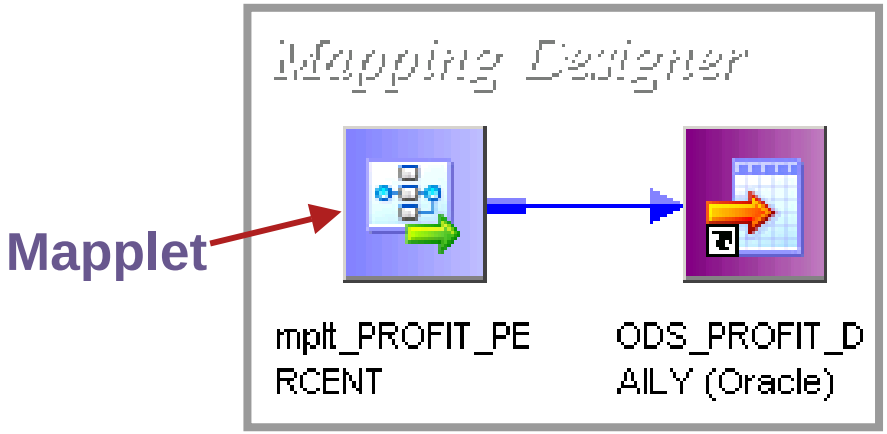
Data Source *Inside* a Mapplet

Source data is defined WITHIN the Mapplet logic

- No Input transformation is required (or allowed)
- Use a Source Qualifier instead



- Resulting Mapplet has *no* input ports
- When used in a Mapping, the Mapplet is the first object in the data flow



INFORMATICA

Mapplet Output Transformation

Use to contain the results of a Mapplet pipeline. Multiple Output transformations are allowed.

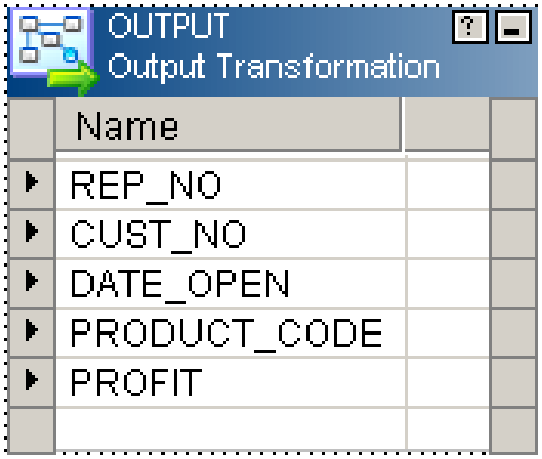
Passive Transformation
Connected

Ports

- Input ports only

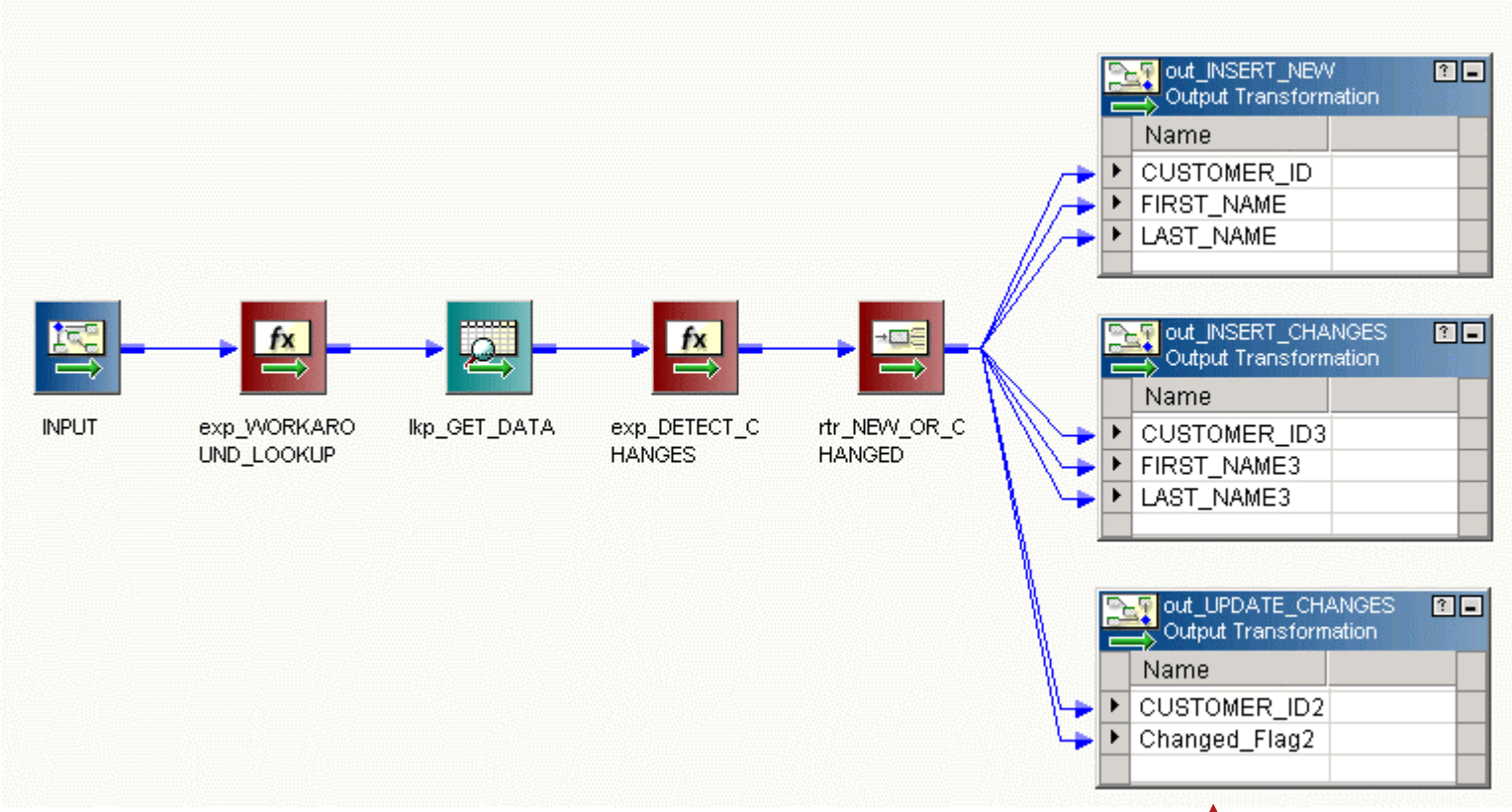
Usage

- Only those ports connected to an Output transformation (from another transformation) will display in the resulting Mapplet
- One (or more) Mapplet Output transformations are required in every Mapplet



INFORMATICA

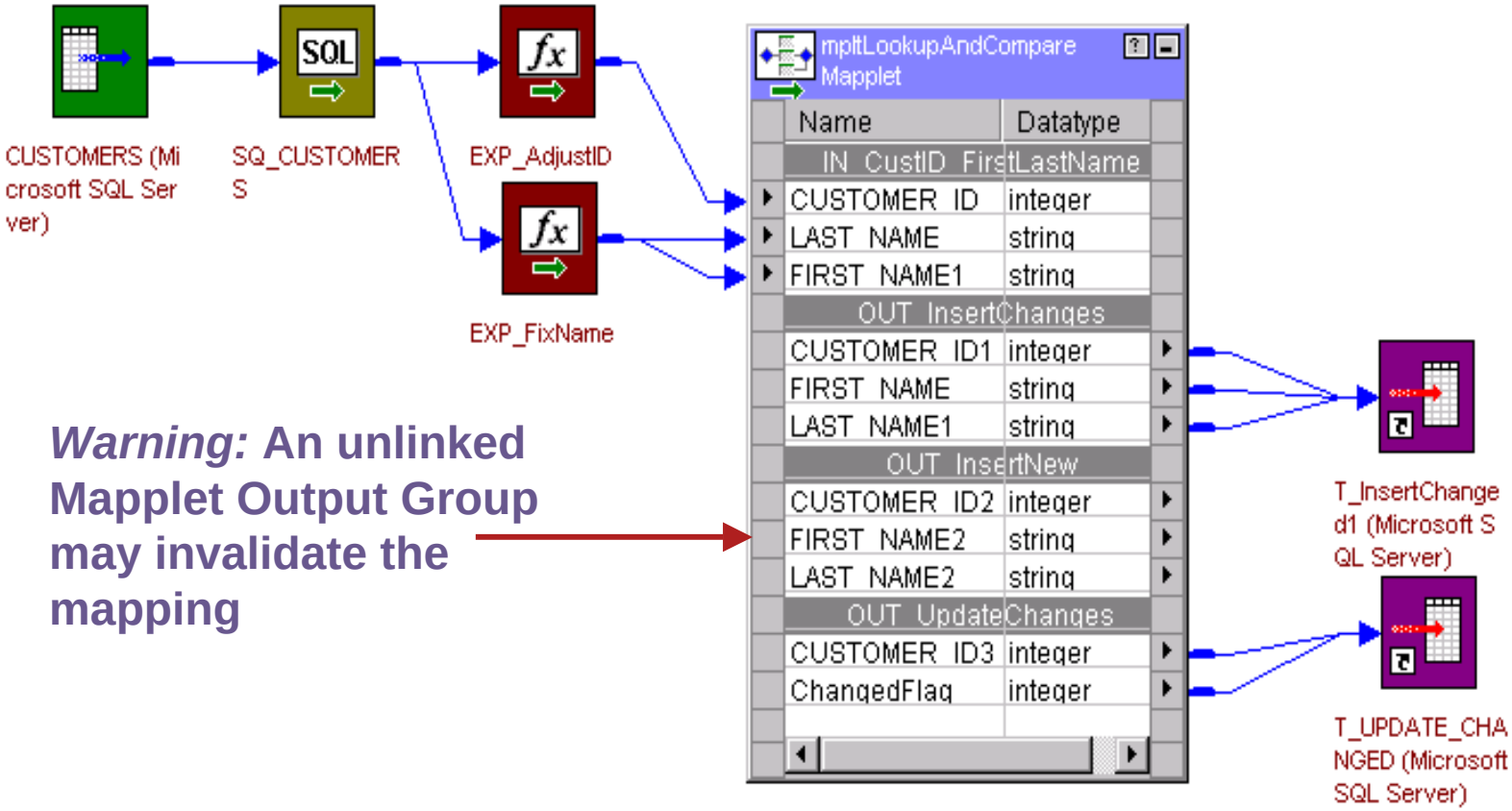
Mapplet with Multiple Output Groups



Can output to multiple instances of the same target table

INFORMATICA

Unmapped Mapplet Output Groups



Warning: **An unlinked Mapplet Output Group may invalidate the mapping**

INFORMATICA

Active and Passive Mapplets

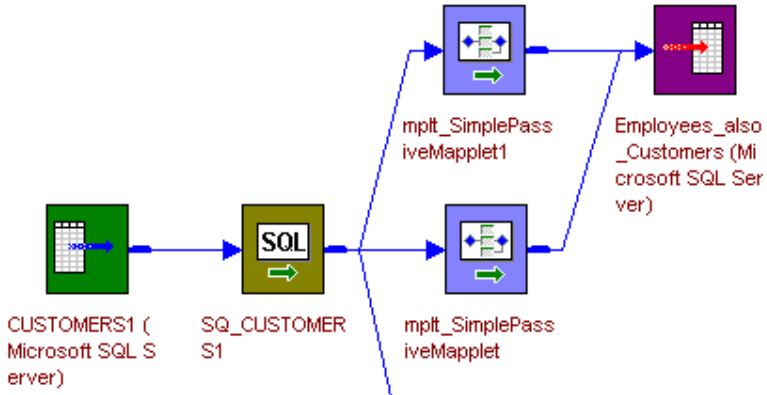
- Passive Mapplets contain only passive transformations
- Active Mapplets contain one or more active transformations

CAUTION: Changing a passive Mapplet into an active Mapplet may invalidate Mappings which use that Mapplet – so do an impact analysis in Repository Manager first

INFORMATICA

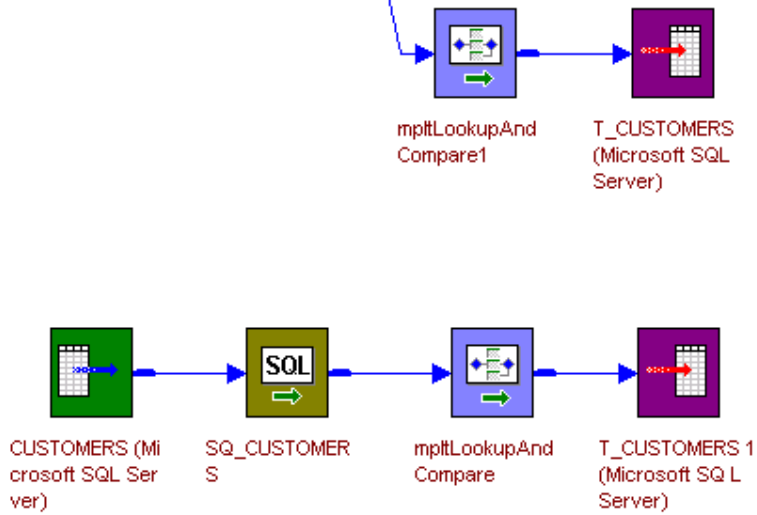
Using Active and Passive Mapplets

Passive



Multiple Passive Mapplets <u>can</u> populate the same target instance

Active



Multiple Active Mapplets or Active and Passive Mapplets <u>cannot</u> populate the same target instance

INFORMATICA

Mapplet Parameters and Variables

- Same idea as mapping parameters and variables
- Defined under the
  Mapplets | Parameters and Variables
  menu option
- A parameter or variable defined in a mapplet is not visible in any parent mapping
- A parameter or variable defined in a mapping is not visible in any child mapplet

INFORMATICA

Lab 17 – Mapplets

INFORMATICA

Reusable Transformations

INFORMATICA

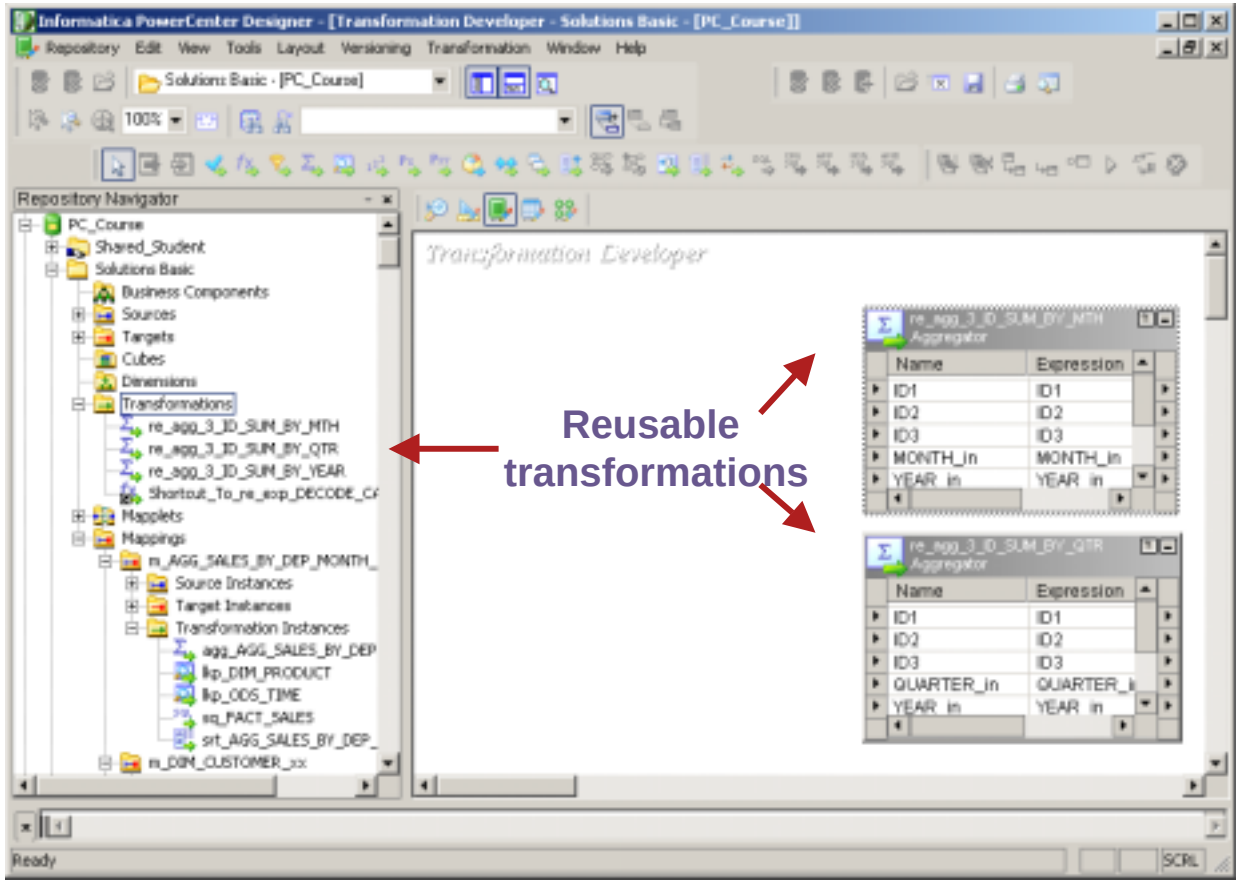
Reusable Transformations

By the end of this section you will be familiar with:

- Transformation Developer
- Reusable transformation rules
- Promoting transformations to reusable
- Copying reusable transformations

INFORMATICA

Transformation Developer



Make a transformation reusable from the outset,

or

test it in a mapping first

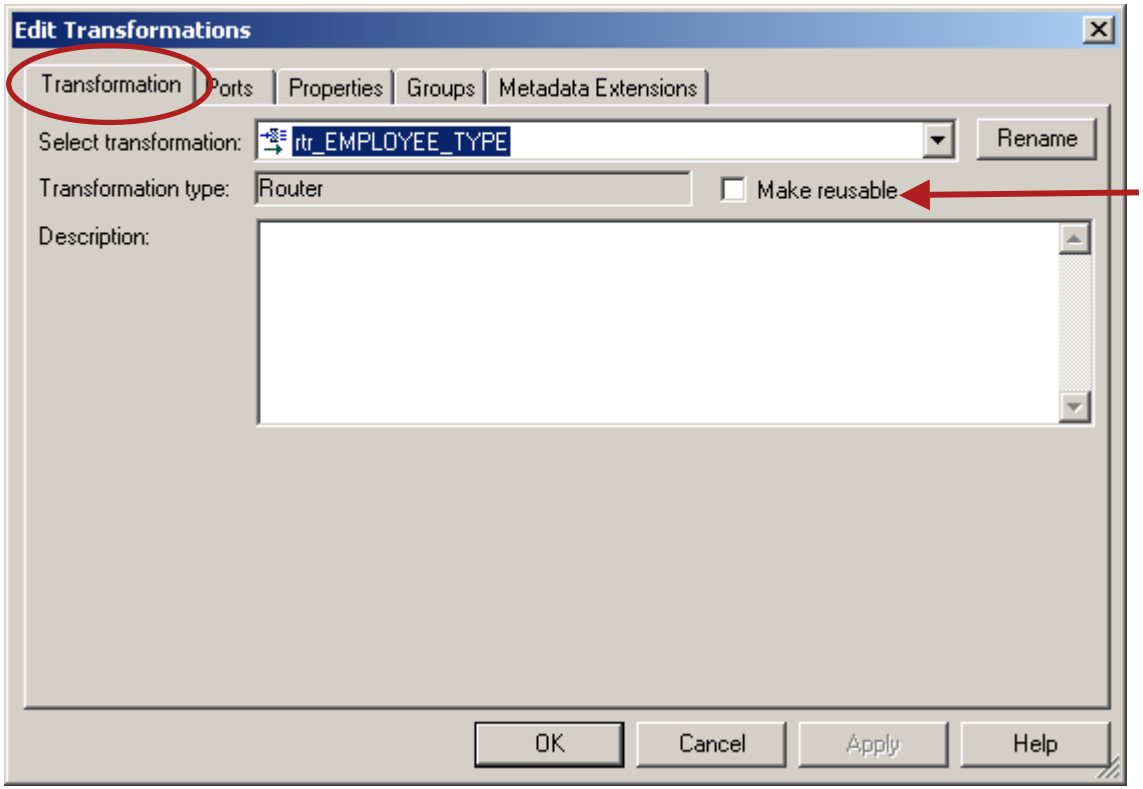
INFORMATICA

Reusable Transformations

- Define once, reuse many times
- Reusable Transformations
  - Can be a copy or a shortcut
  - Edit Ports only in Transformation Developer
  - Can edit Properties in the mapping
  - Instances dynamically inherit changes
  - Caution: changing reusable transformations can invalidate mappings
- Transformations that ***cannot*** be made reusable
  - Source Qualifier
  - ERP Source Qualifier
  - Normalizer (used to read a COBOL data source)

INFORMATICA

Promoting a Transformation to Reusable



Check the Make reusable box (irreversible)

INFORMATICA

Copying Reusable Transformations

This copy action must be done within the same folder

1. **Hold down Ctrl key and drag a Reusable transformation from the Navigator window into a mapping (Mapping Designer tool)**
2. **A message appears in the status bar:**

   Make a non-reusable copy of this transformation and add it to this mapping.

3. **Drop the transformation into the mapping**
4. **Save the changes to the Repository**

INFORMATICA

Lab 18 – Reusable Transformations

INFORMATICA

Session-Level Error Logging

INFORMATICA

Error Logging Objectives

By the end of this section, you will be familiar with:

- **Setting error logging options**
- **How data rejects and transformation errors are handled with logging on and off**
- **How to log errors to a flat file or relational table**
- **When and how to use source row logging**

INFORMATICA

Error Types

- **Transformation error**
  - Data row has only passed partway through the mapping transformation logic
  - An error occurs within a transformation
- **Data reject**
  - Data row is fully transformed according to the mapping logic
  - Due to a data issue, it cannot be written to the target
  - A data reject can be forced by an Update Strategy

INFORMATICA

Error Logging Off/On

Error Type	**Logging OFF** (Default)	**Logging ON**
Transformation errors	Written to session log then discarded	Appended to flat file or relational tables. Only fatal errors written to session log.
Data rejects	Appended to reject file (one .bad file per target)	Not written to reject file

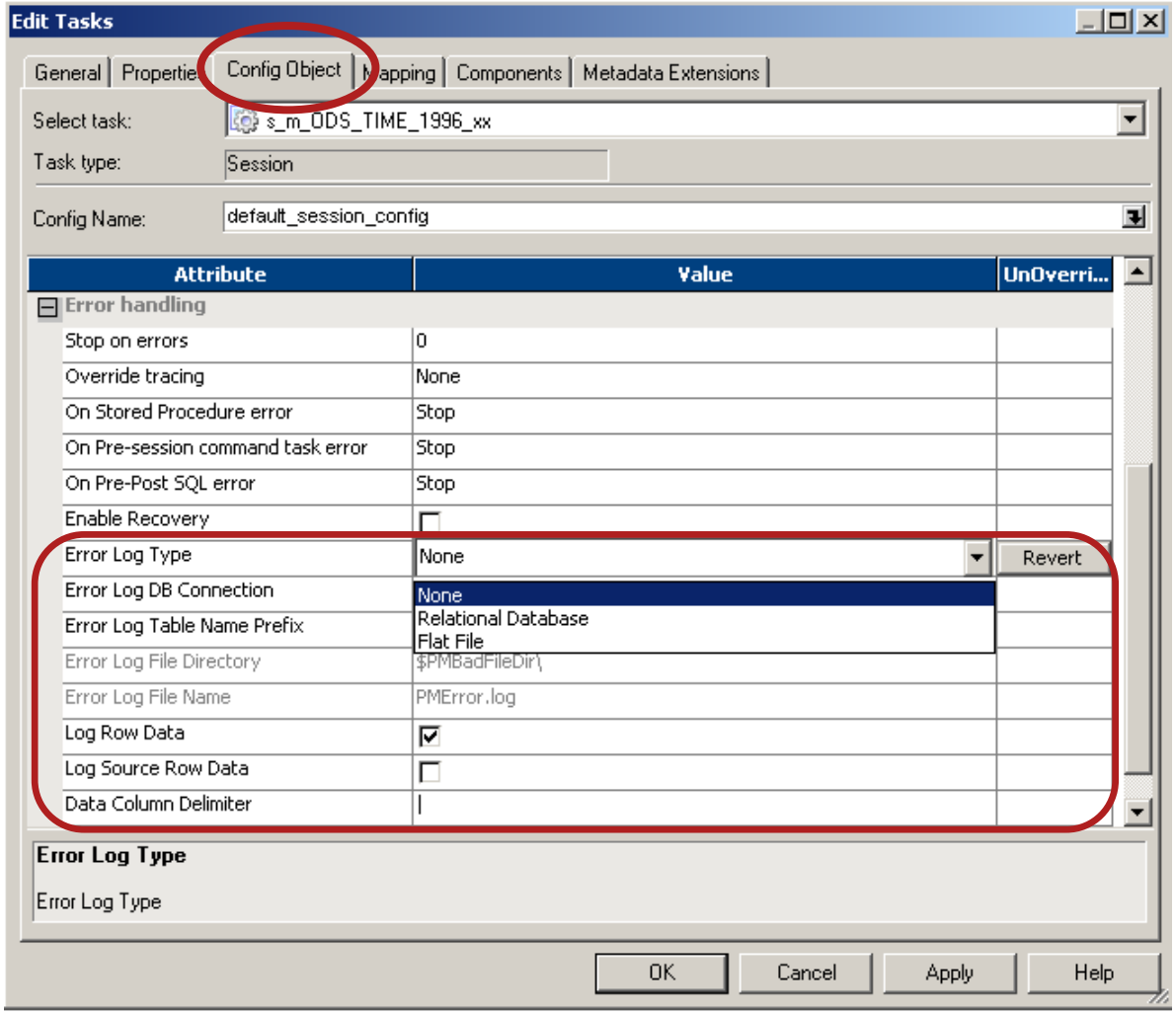
INFORMATICA

Setting Error Log Options

In Session task

Error Log Type

Log Row Data
Log Source Row Data

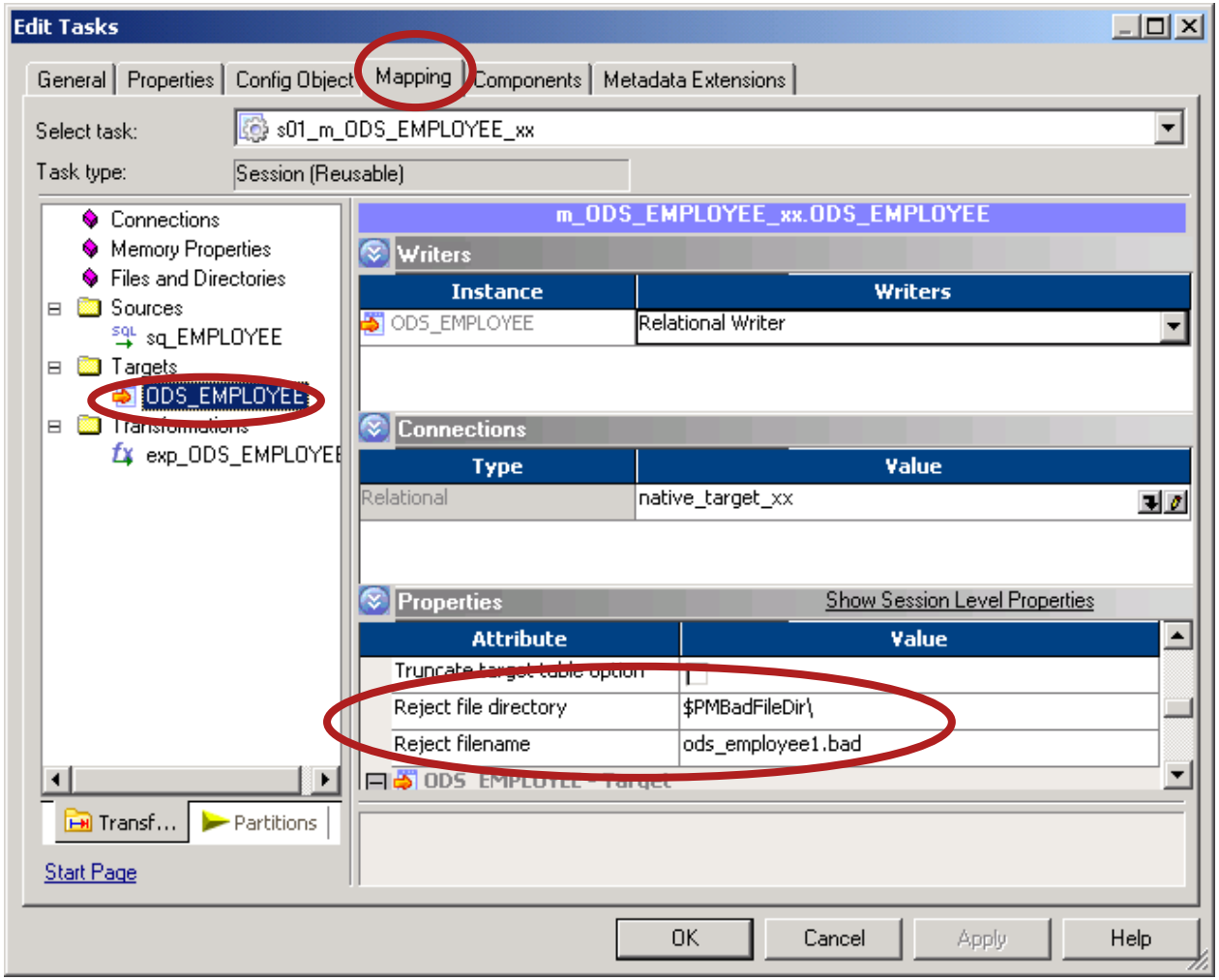


INFORMATICA

Error Logging Off – Specifying Reject Files

n Session task

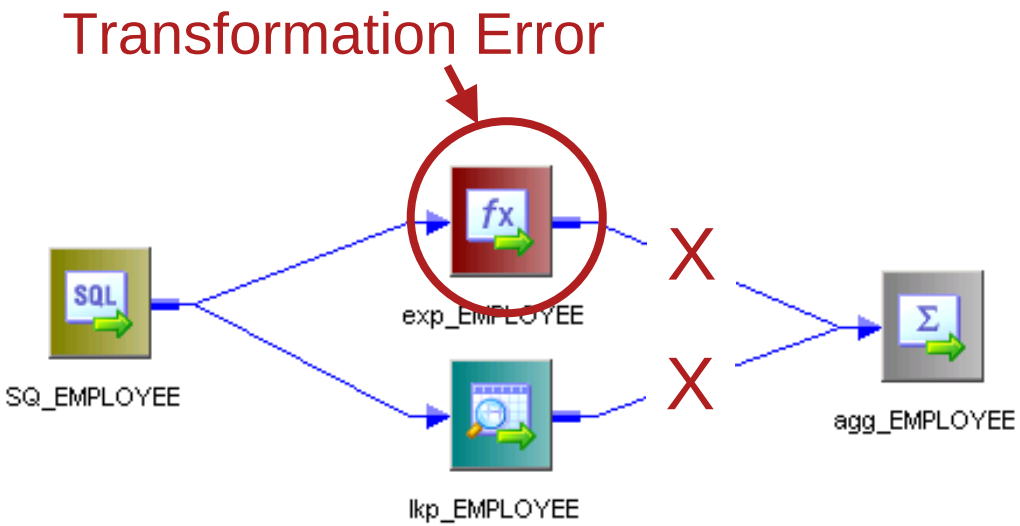
1 file per target



INFORMATICA

Error Logging Off – Transformation Errors

- Details and data are written to session log
- Data row is discarded
- If data flows concatenated, corresponding rows in parallel flow are also discarded



INFORMATICA

Error Logging Off – Data Rejects

Conditions causing data to be rejected include:

- Target database constraint violations, out-of-space errors, log space errors, null values not accepted
- Data-driven records, containing value ‘3’ or DD_REJECT (the reject has been forced by an Update Strategy)
- Target table properties ‘reject truncated/overflowed rows’

Sample reject file

First column:
0=INSERT →
1=UPDATE →
2=DELETE →
3=REJECT →

```
0,D,1313,D,Regulator System,D,Air Regulators,D,250.00,D,150.00,D
1,D,1314,D,Second Stage Regulator,D,Air Regulators,D,365.00,D,265.00,D
2,D,1390,D,First Stage Regulator,D,Air Regulators,D,170.00,D,70.00,D
3,D,2341,D,Depth/Pressure Gauge,D,Small Instruments,D,105.00,D,5.00,D
```

Indicator describes preceding column value
D=Data, O=Overflow, N=Null or T=Truncated

INFORMATICA

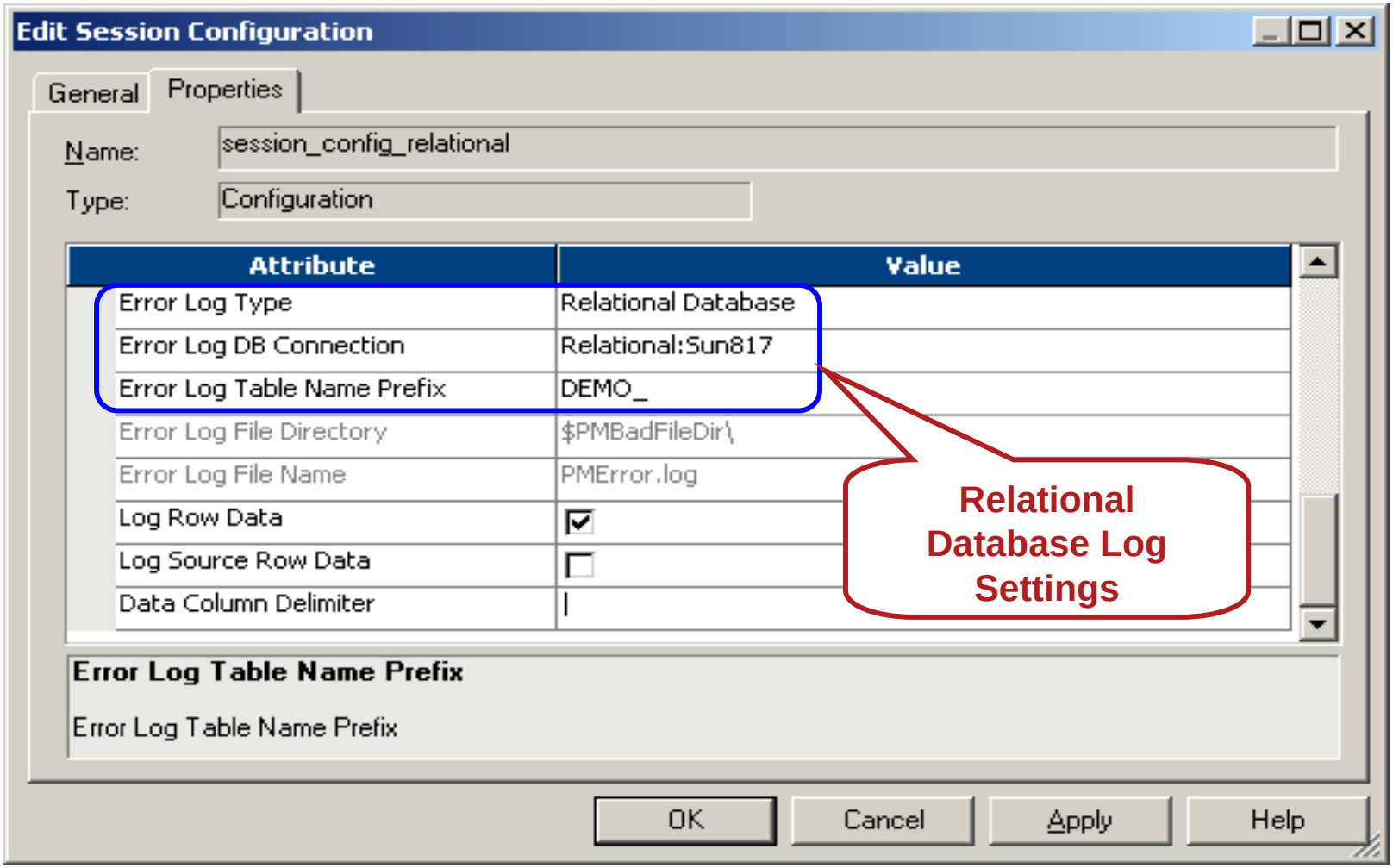
Log Row Data

Logs:

- Session metadata
- Reader, transformation, writer and user-defined errors
- For errors on input, logs row data for I and I/O ports
- For errors on output, logs row data for I/O and O ports

INFORMATICA

Logging Errors to a Relational Database 1

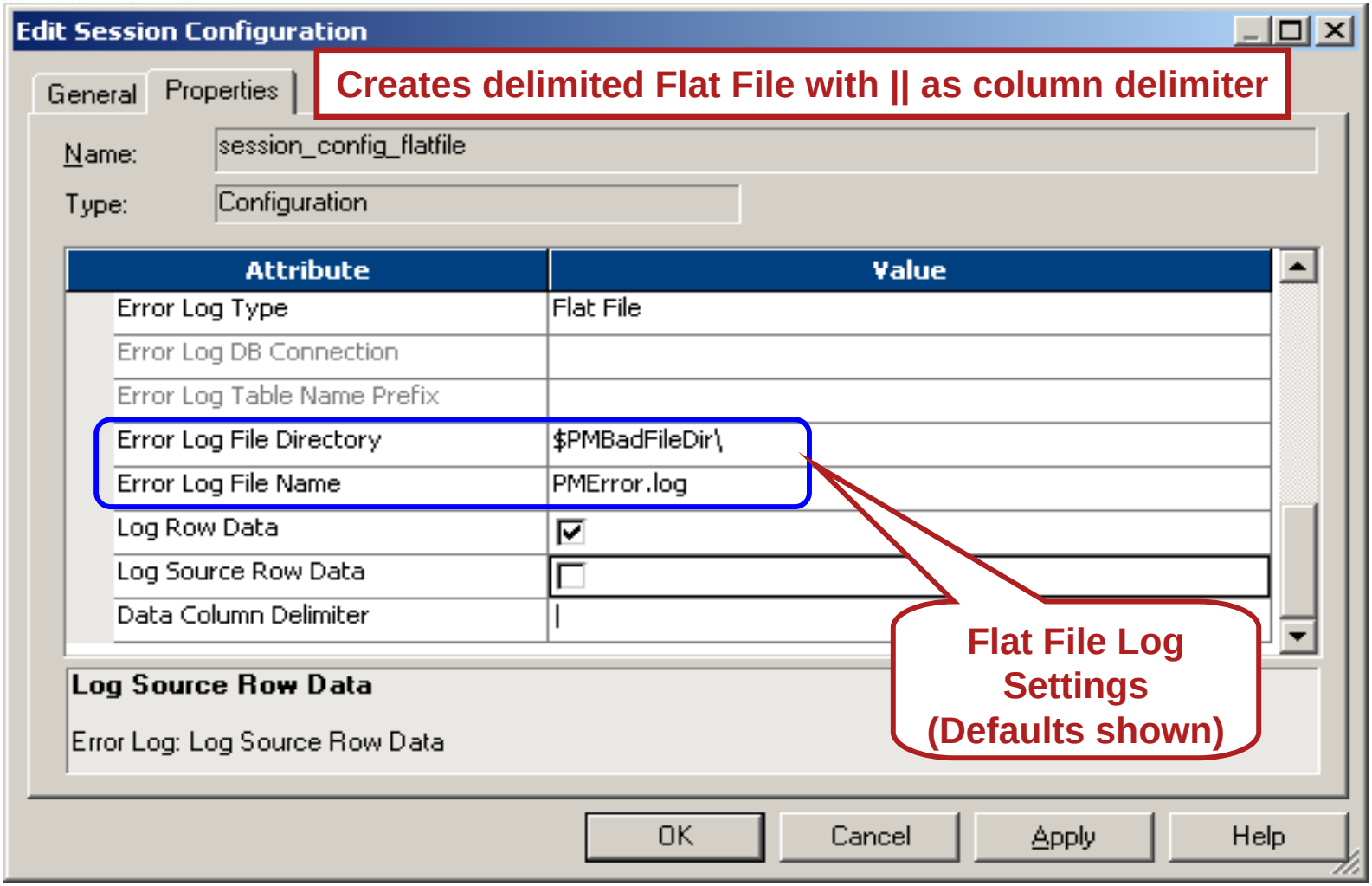


Logging Errors to a Relational Database 2

- **PMERR_SESS**: Stores metadata about the session run such as workflow name, session name, repository name etc
- **PMERR_MSG:** Error messages for a row of data are logged in this table
- **PMERR_TRANS:** Metadata about the transformation such as transformation group name, source name, port names with datatypes are logged in this table
- **PMERR_DATA:** The row data of the error row as well as the source row data is logged here. The row data is in a string format such as [indicator1: data1 | indicator2: data2]

INFORMATICA

Error Logging to a Flat File 1



INFORMATICA

Logging Errors to a Flat File 2

- **Format: Session metadata followed by de-normalized error information**
- **Sample session metadata**

```
*****************************************************************
Repository GID: 510e6f02-8733-11d7-9db7-00e01823c14d
Repository: RowErrorLogging
Folder: ErrorLogging
Workflow: w_unitTests
Session: s_customers
Mapping: m_customers
Workflow Run ID: 6079
Worklet Run ID: 0
Session Instance ID: 806
Session Start Time: 10/19/2003 11:24:16
Session Start Time (UTC): 1066587856
*****************************************************************
```

- **Row data format**

Transformation || Transformation Mapplet Name || Transformation Group || Partition Index || Transformation Row ID || Error Sequence || Error Timestamp || Error UTC Time || Error Code || Error Message || Error Type || Transformation Data || Source Mapplet Name || Source Name || Source Row ID || Source Row Type || Source Data

INFORMATICA

Log Source Row Data 1

- Separate checkbox in session task
- Logs the source row associated with the error row
- Logs metadata about source, e.g. Source Qualifier, source row id, and source row type

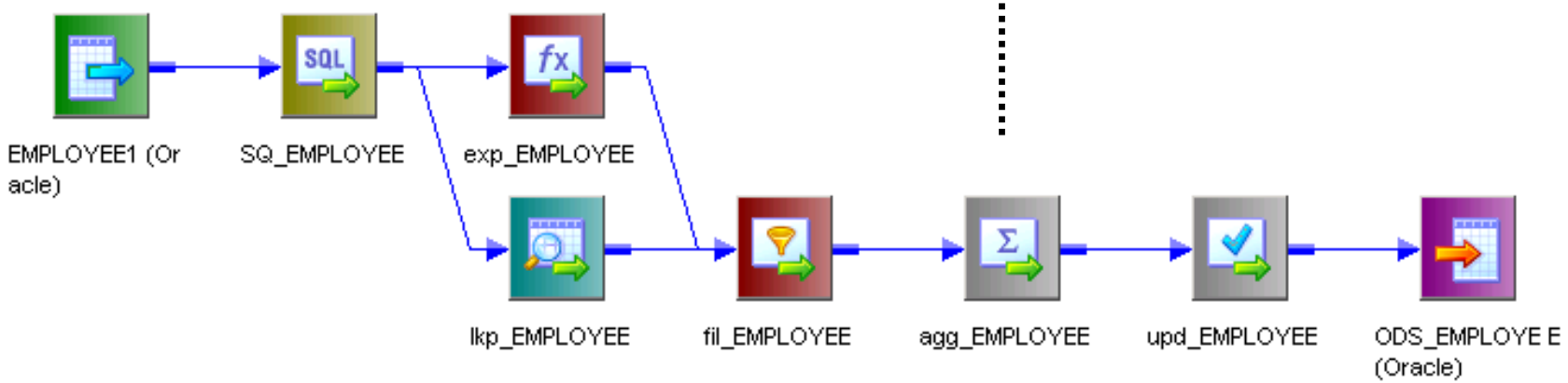
INFORMATICA

Log Source Row Data 2

Source row logging is not available downstream of an Aggregator, Joiner, Sorter (where output rows are not uniquely correlated with input rows)

Source row logging available

Source row logging not available



INFORMATICA

Workflow Configuration

INFORMATICA

Workflow Configuration Objectives

By the end of this section, you will be able to create:

- **Workflow Server Connections**
- **Reusable Schedules**
- **Reusable Session Configurations**

INFORMATICA

Workflow Configuration

- Workflow Server Connections
- Reusable Workflow Schedules
- Reusable Session Configurations

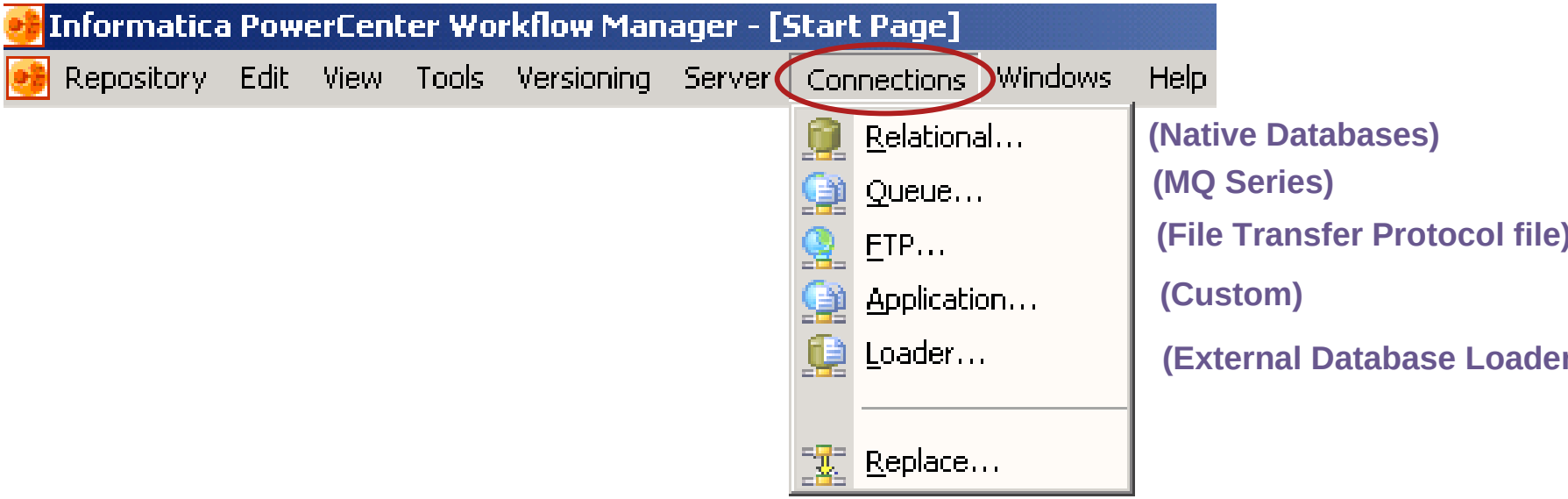
INFORMATICA

Workflow Server Connections

INFORMATICA

Workflow Server Connections

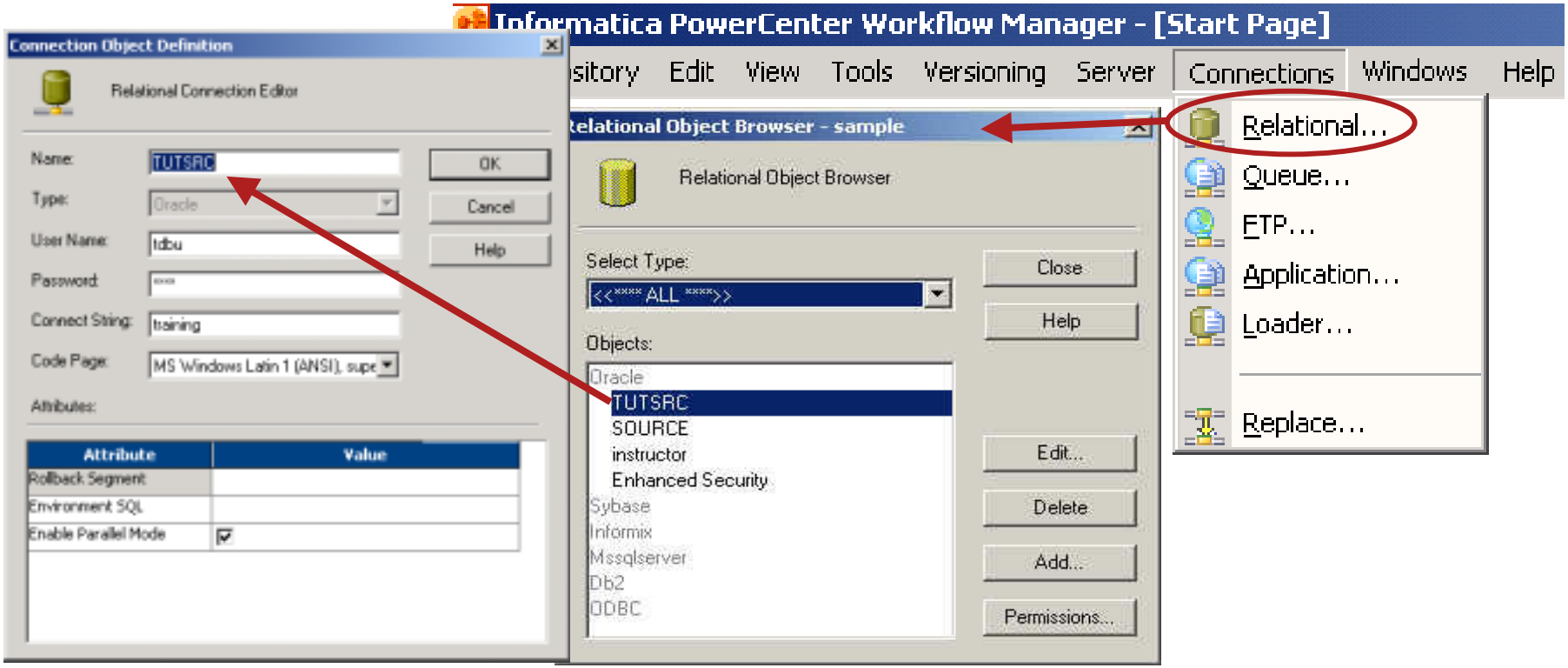
- Configure Server data access connections in the Workflow Manager
- Used in Session Tasks



INFORMATICA

Relational Connections (Native)

- Create a relational [database] connection
  - Instructions to the Server to locate relational tables
  - Used in Session Tasks



INFORMATICA

Relational Connection Properties

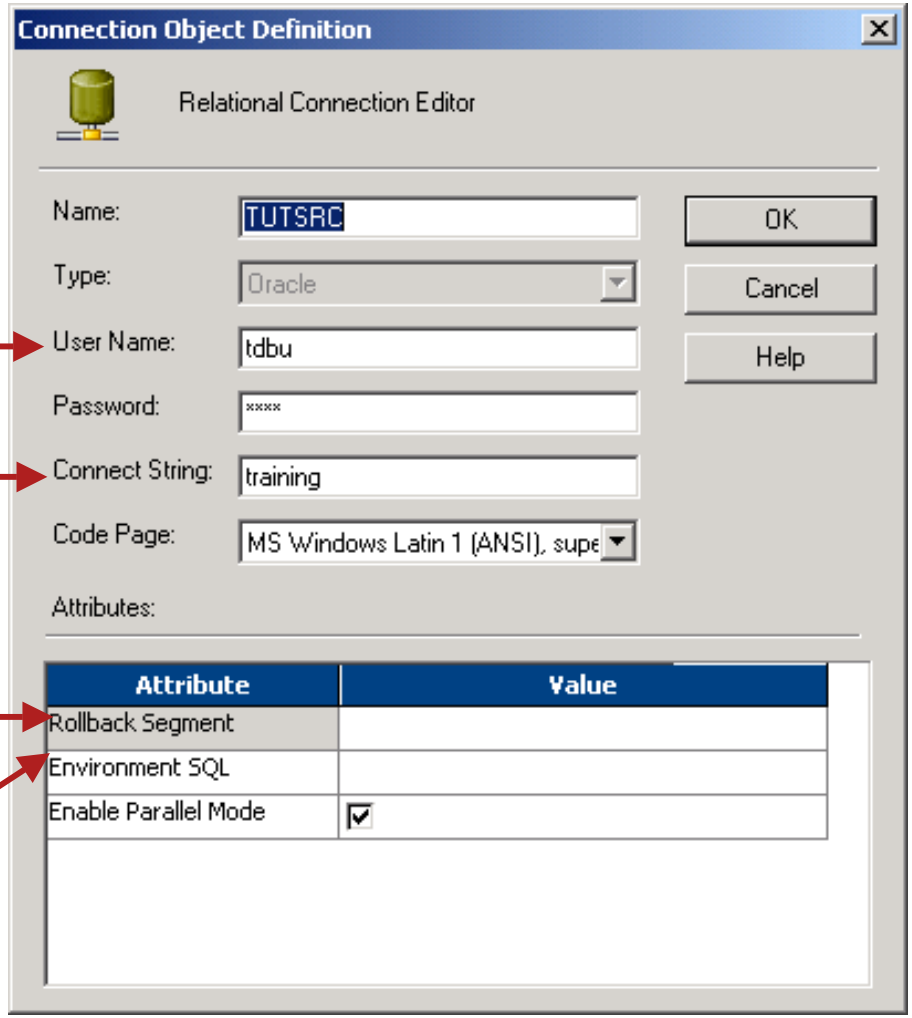
Define native relational database connection

User Name/Password

Database connectivity information

Rollback Segment assignment (optional)

Optional Environment SQL (executed with each use of database connection)

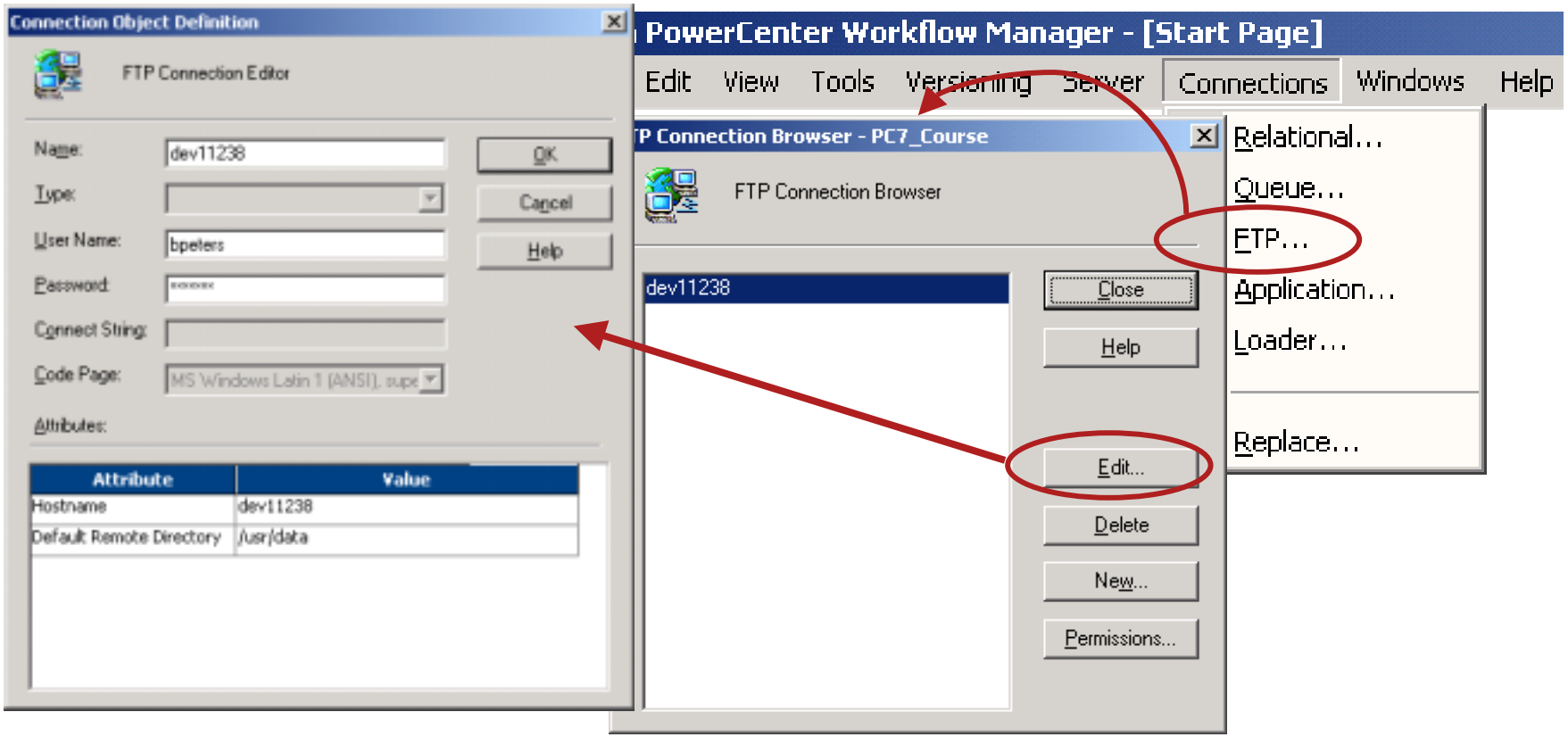


INFORMATICA

FTP Connection

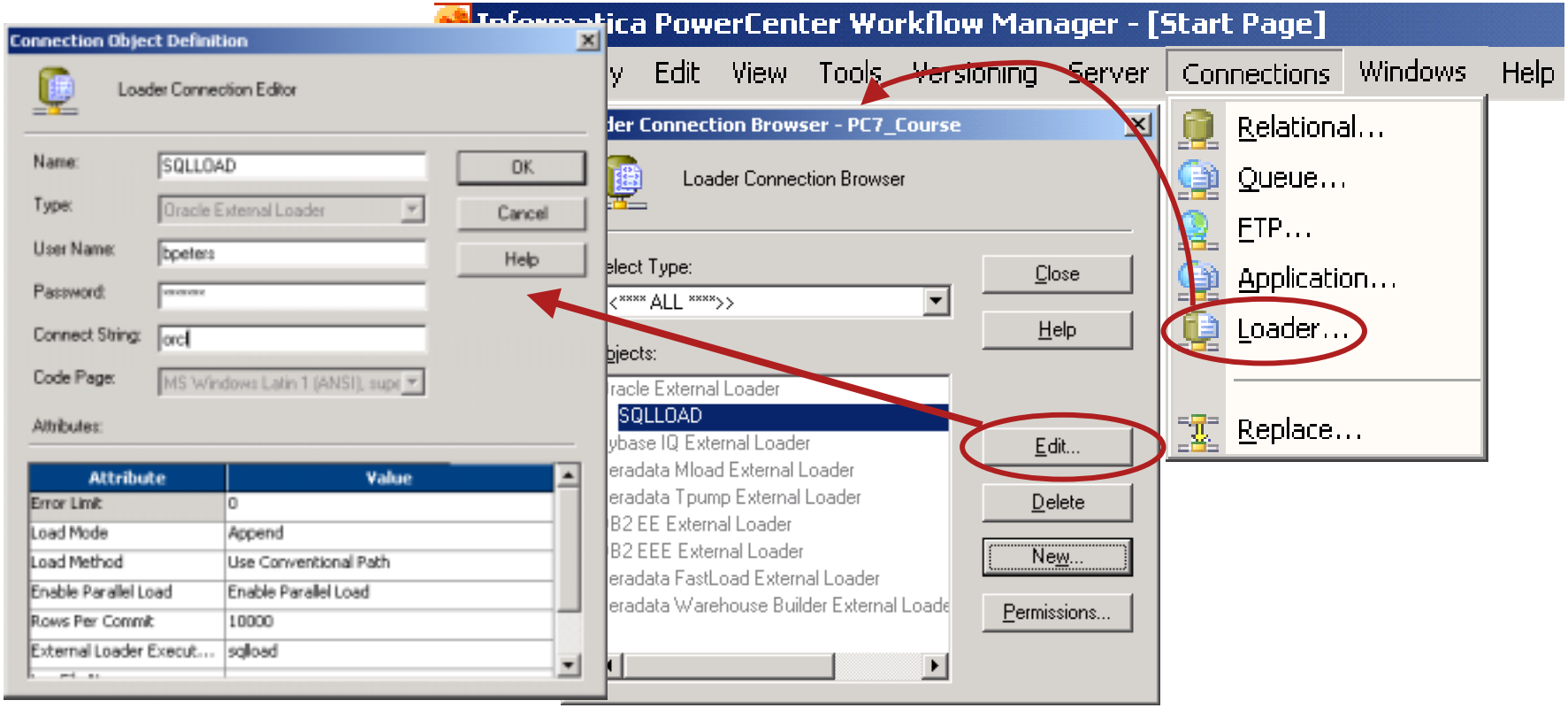
Create an FTP connection

- Instructions to the Server to ftp flat files
- Used in Session Tasks



External Loader Connection

- Create an External Loader connection
  - Instructs the Server to invoke an external database loader
  - Used in Session Tasks



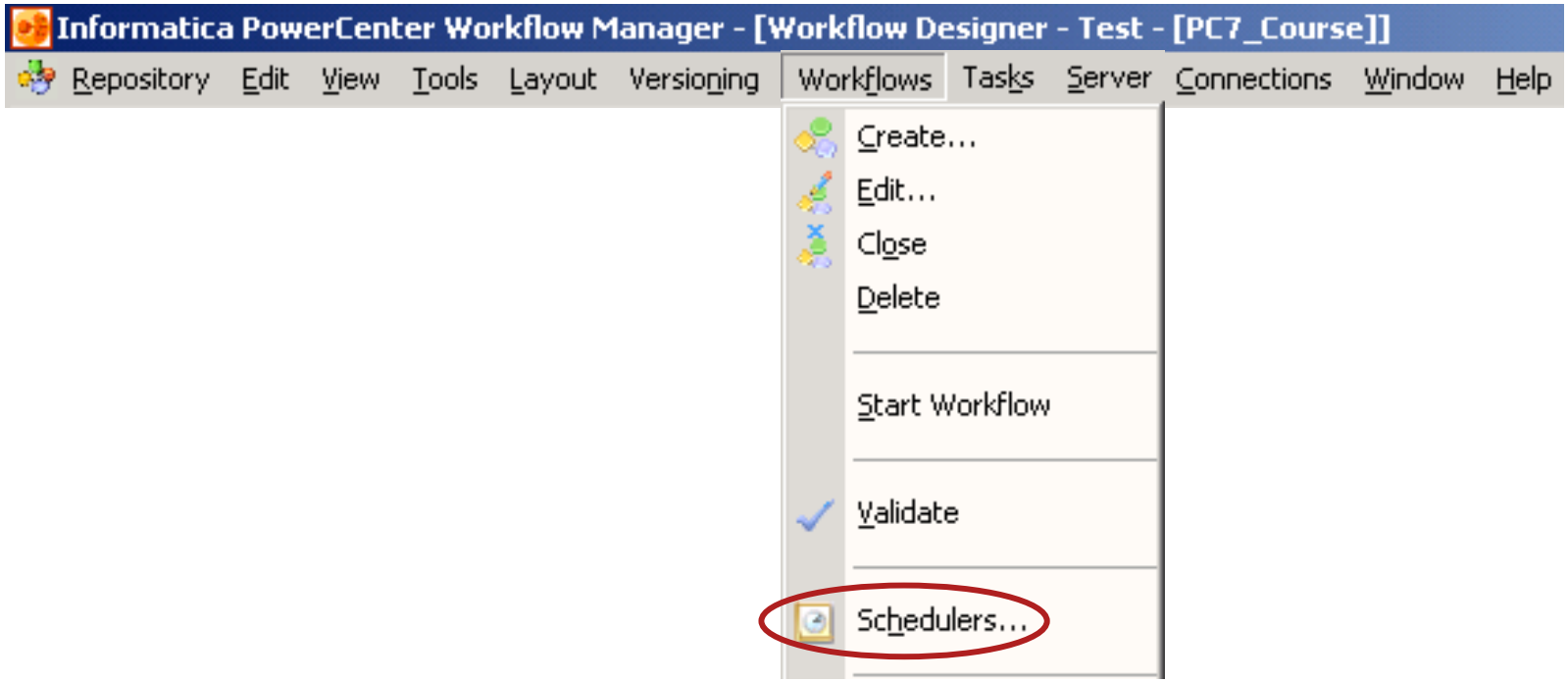
INFORMATICA

Reusable Workflow Schedules

INFORMATICA

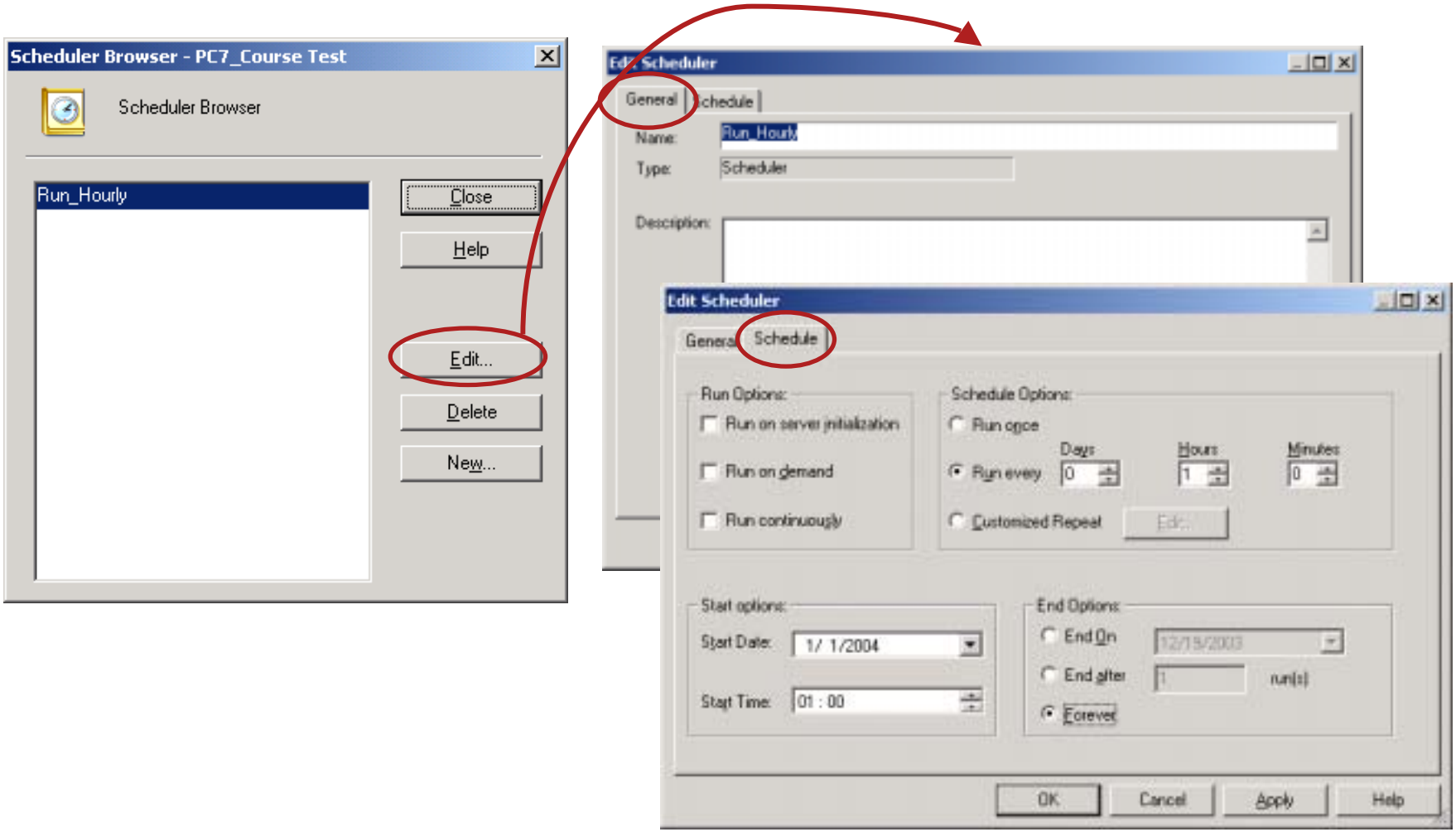
Reusable Workflow Schedules

- **Set up reusable schedules to associate with multiple Workflows**
  - Defined at folder level
  - Must have the Workflow Designer tool open



INFORMATICA

Reusable Workflow Schedules



INFORMATICA

Reusable Session Configurations

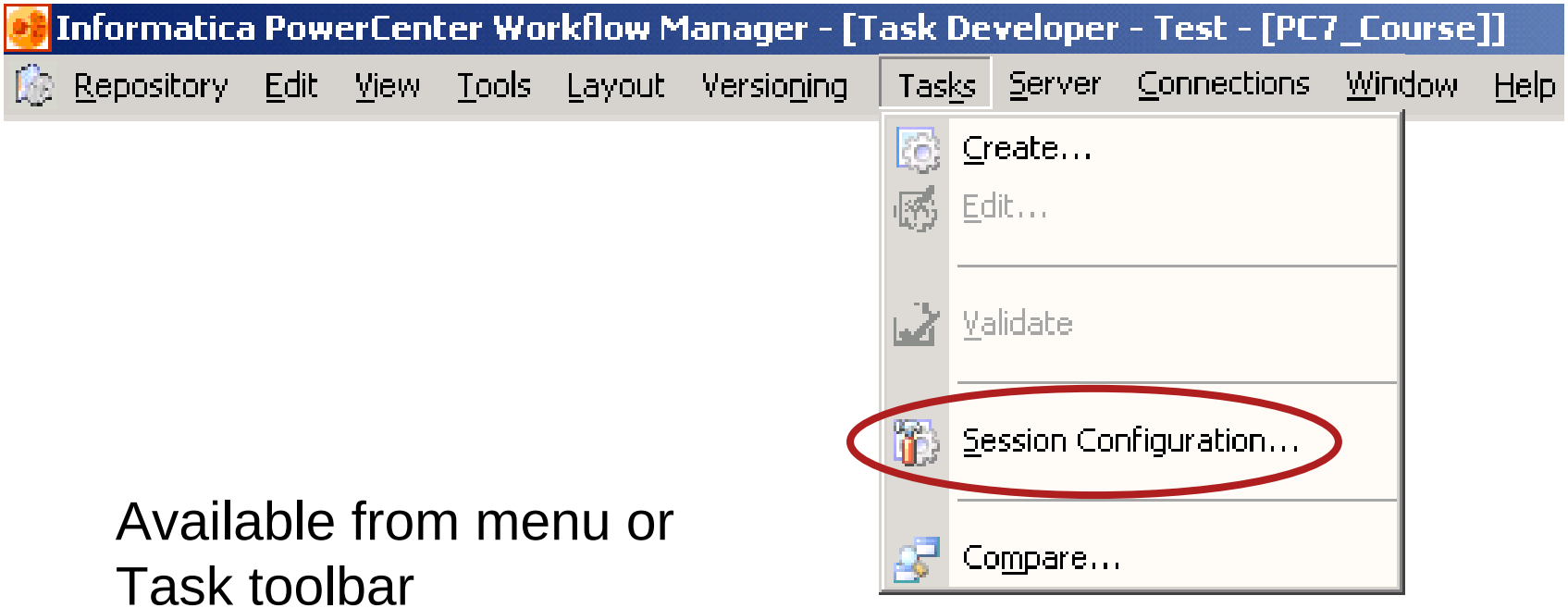
INFORMATICA

Session Configuration

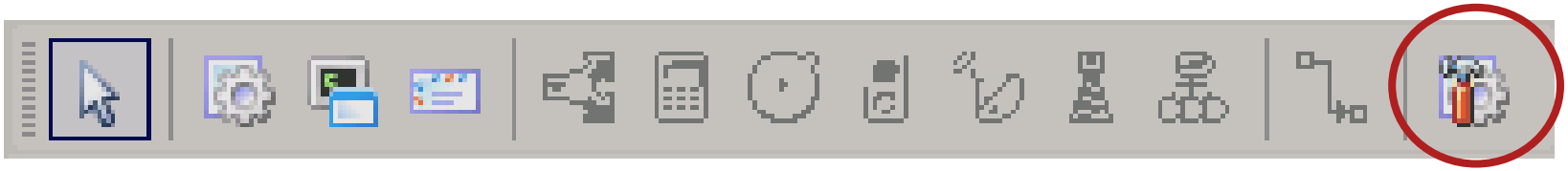
- Define properties to be reusable across different sessions
- Defined at folder level
- Must have one of these tools open in order to access

INFORMATICA

Session Configuration (cont'd)

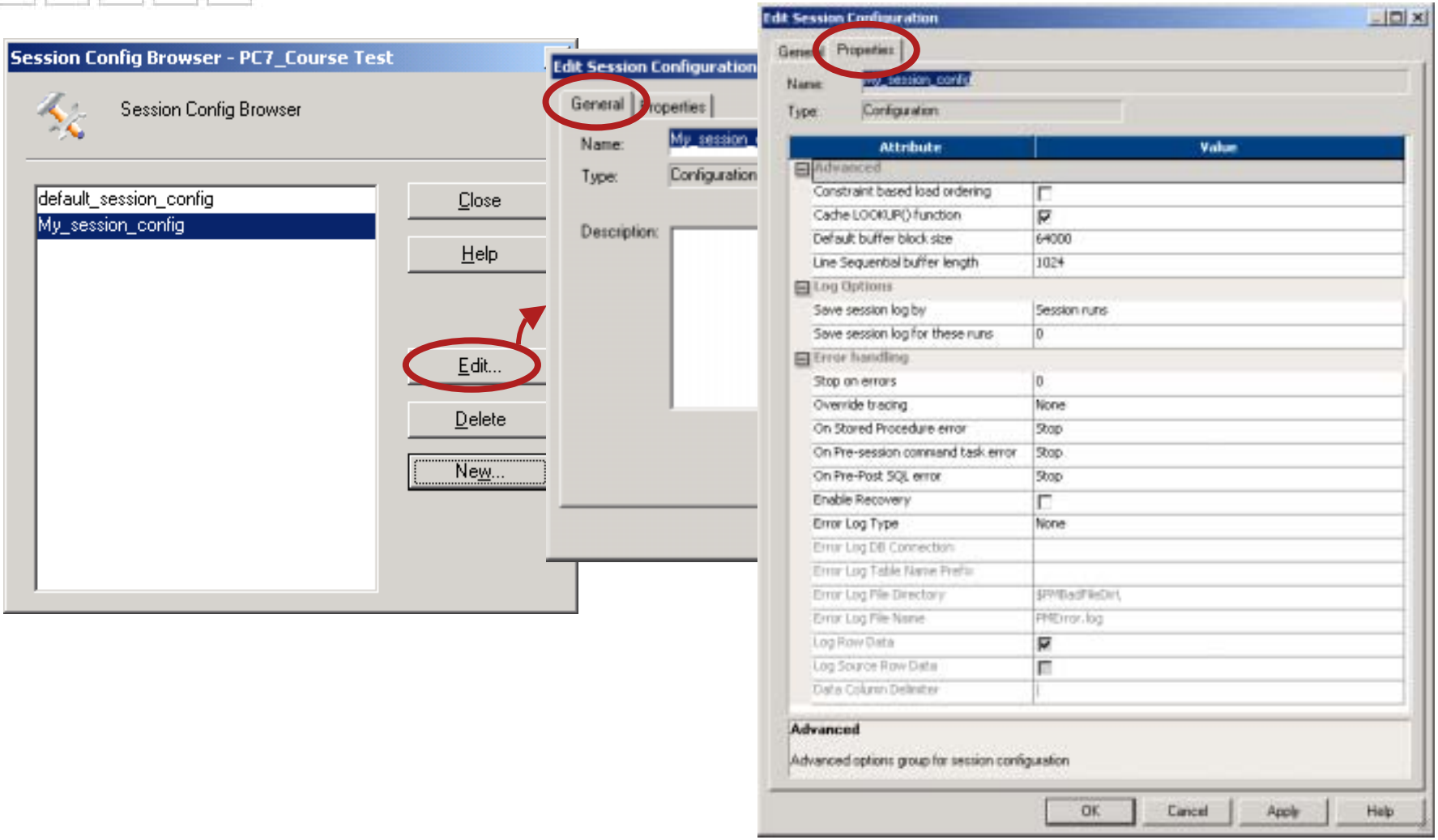


Available from menu or Task toolbar



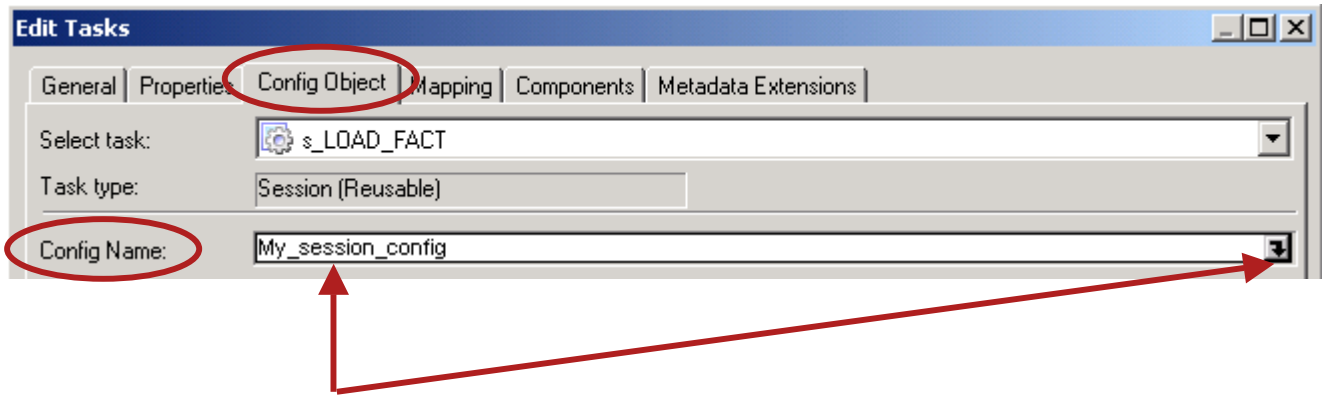
INFORMATICA

Session Configuration (cont'd)



INFORMATICA

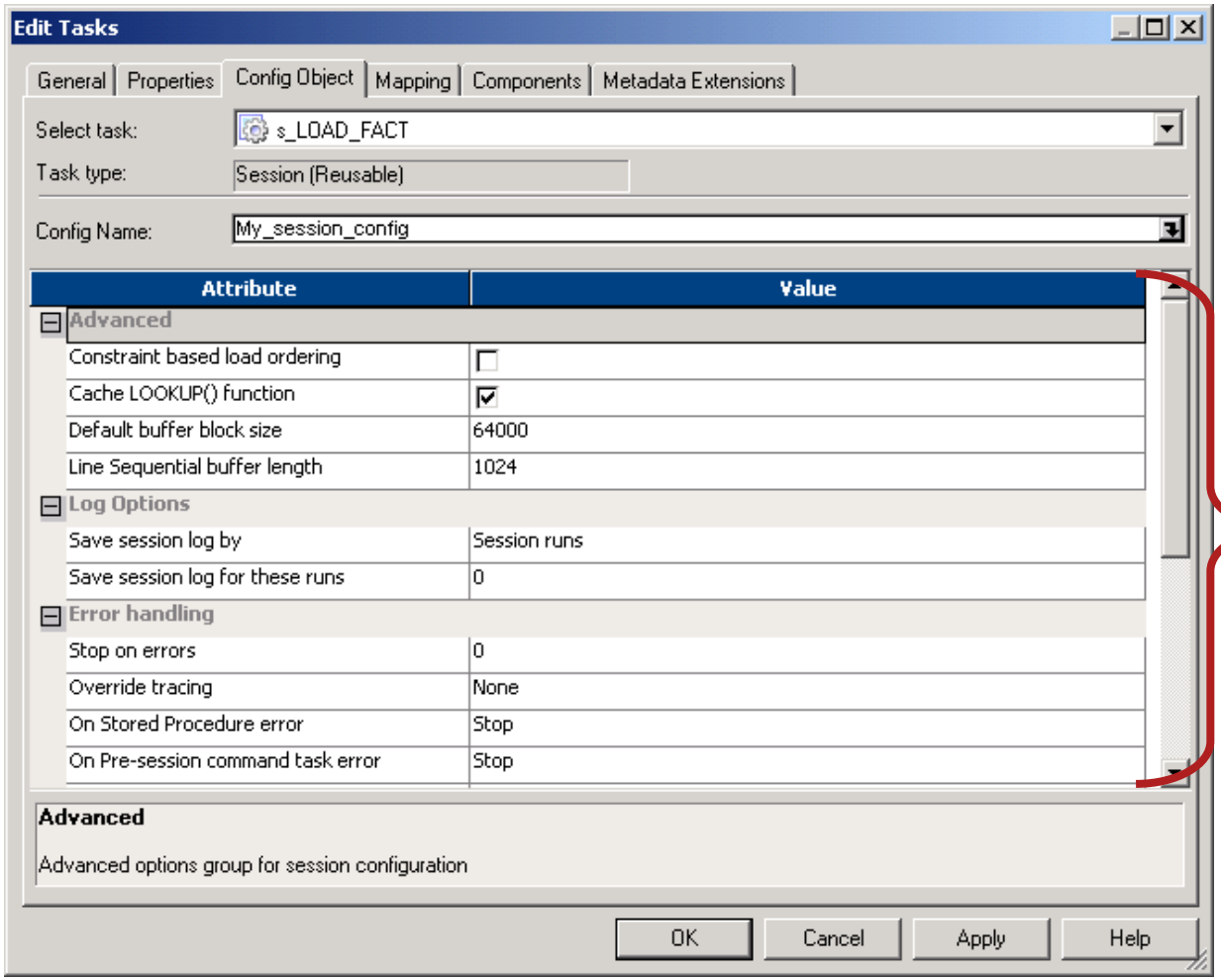
Session Task – Config Object



Within Session task properties, choose desired configuration

INFORMATICA

Session Task – Config Object Attributes



Attributes may be overridden within the Session task

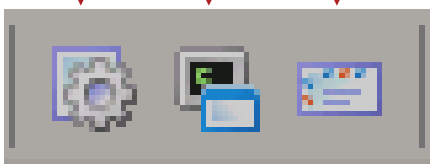
INFORMATICA

Reusable Tasks

INFORMATICA

Reusable Tasks

- **Three types of reusable Tasks**



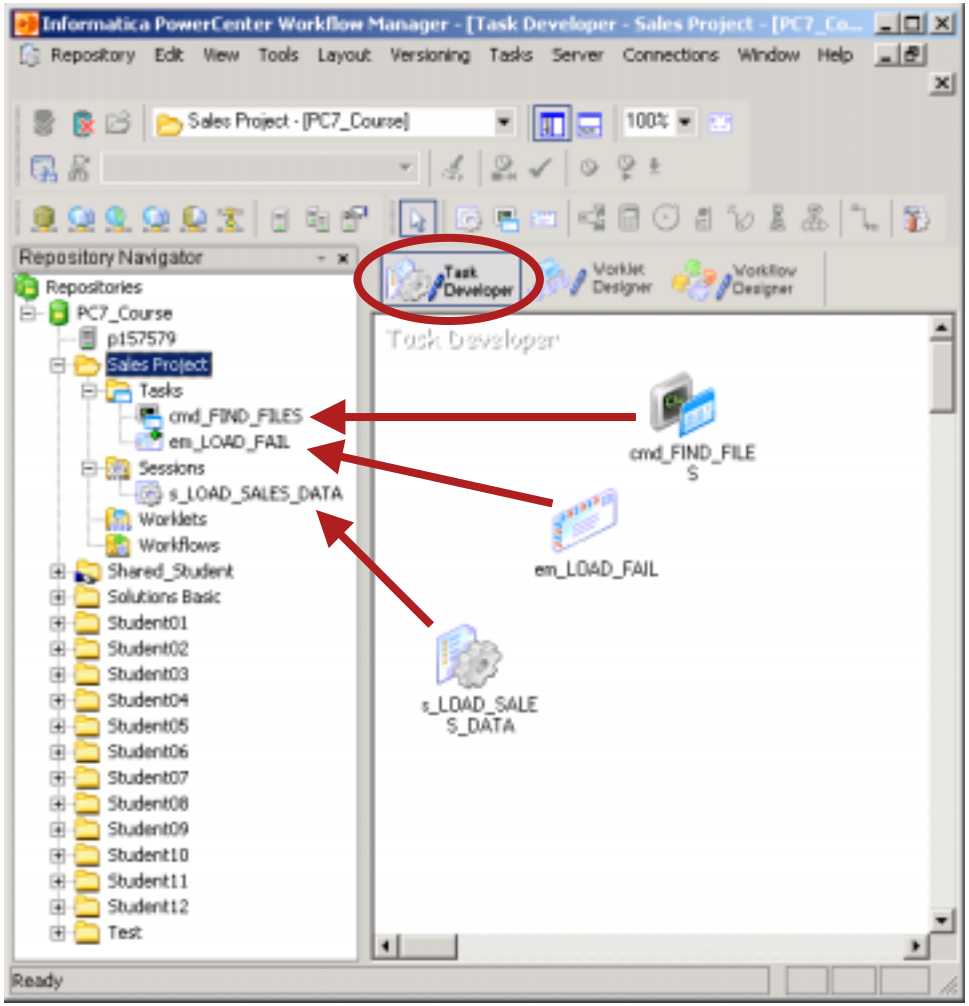
Session – Set of instructions to execute a specific Mapping

Command – Specific shell commands to run during any Workflow

Email – Sends email during the Workflow

INFORMATICA

Reusable Tasks

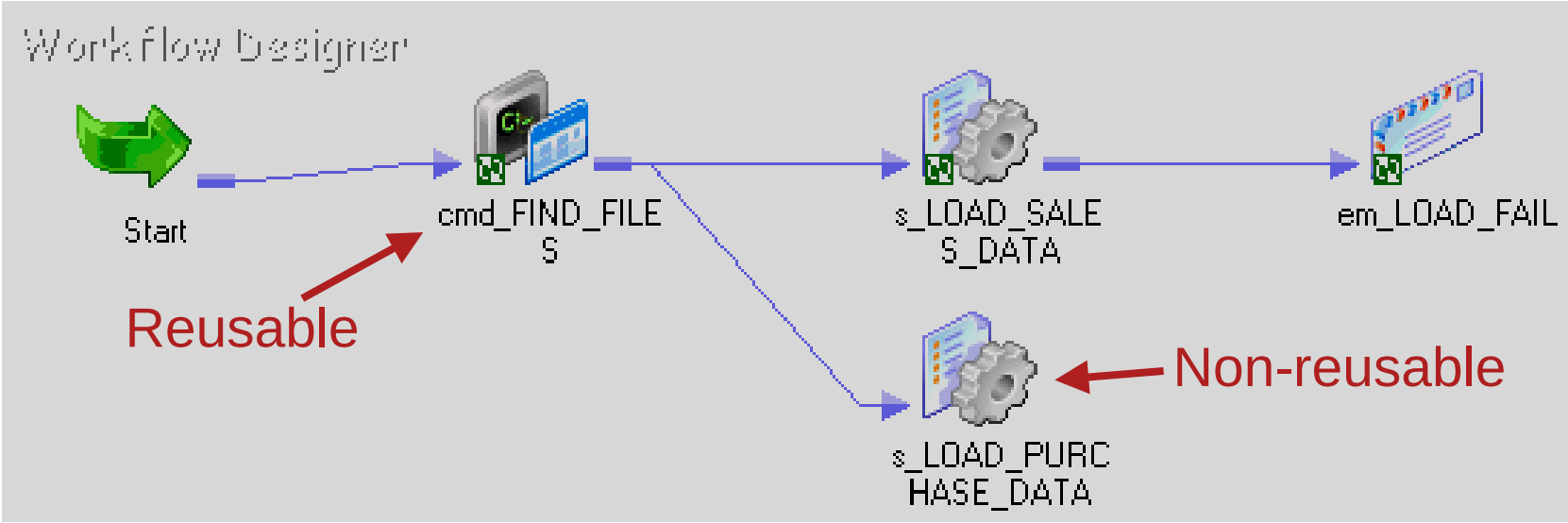


- Use the *Task Developer* to create reusable tasks
- These tasks will then appear in the Navigator and can be dragged and dropped into any workflow

INFORMATICA

Reusable Tasks in a Workflow

- **In a workflow, a reusable task is represented with the symbol**



INFORMATICA

Command Task

- **Specify one or more Unix shell or DOS commands to run during the Workflow**
  - Runs in the Informatica Server (UNIX or Windows) environment
- **Shell command status (successful completion or failure) is held in the pre-defined variable $command_task_name.STATUS**
- **Each Command Task shell command can execute before the Session begins or after the Informatica Server executes a Session**

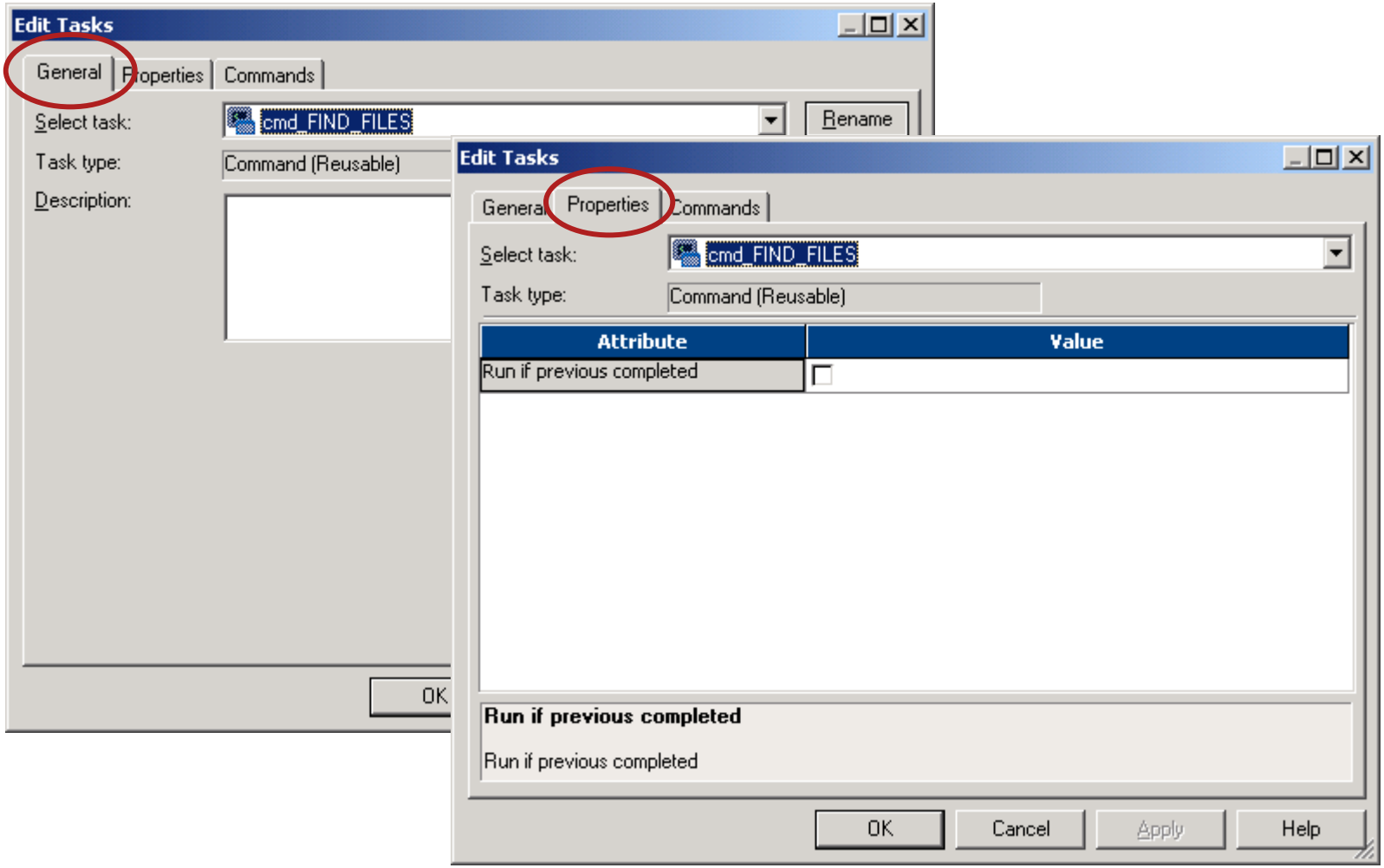
INFORMATICA

Command Task

- Specify one (or more) Unix shell or DOS (NT, Win2000) commands to run at a specific point in the workflow
- Becomes a component of a workflow (or worklet)
- If created in the Task Developer, the Command task is reusable
- If created in the Workflow Designer, the Command task is not reusable
- Commands can also be invoked under the *Components* tab of a Session task to run pre- or post-session

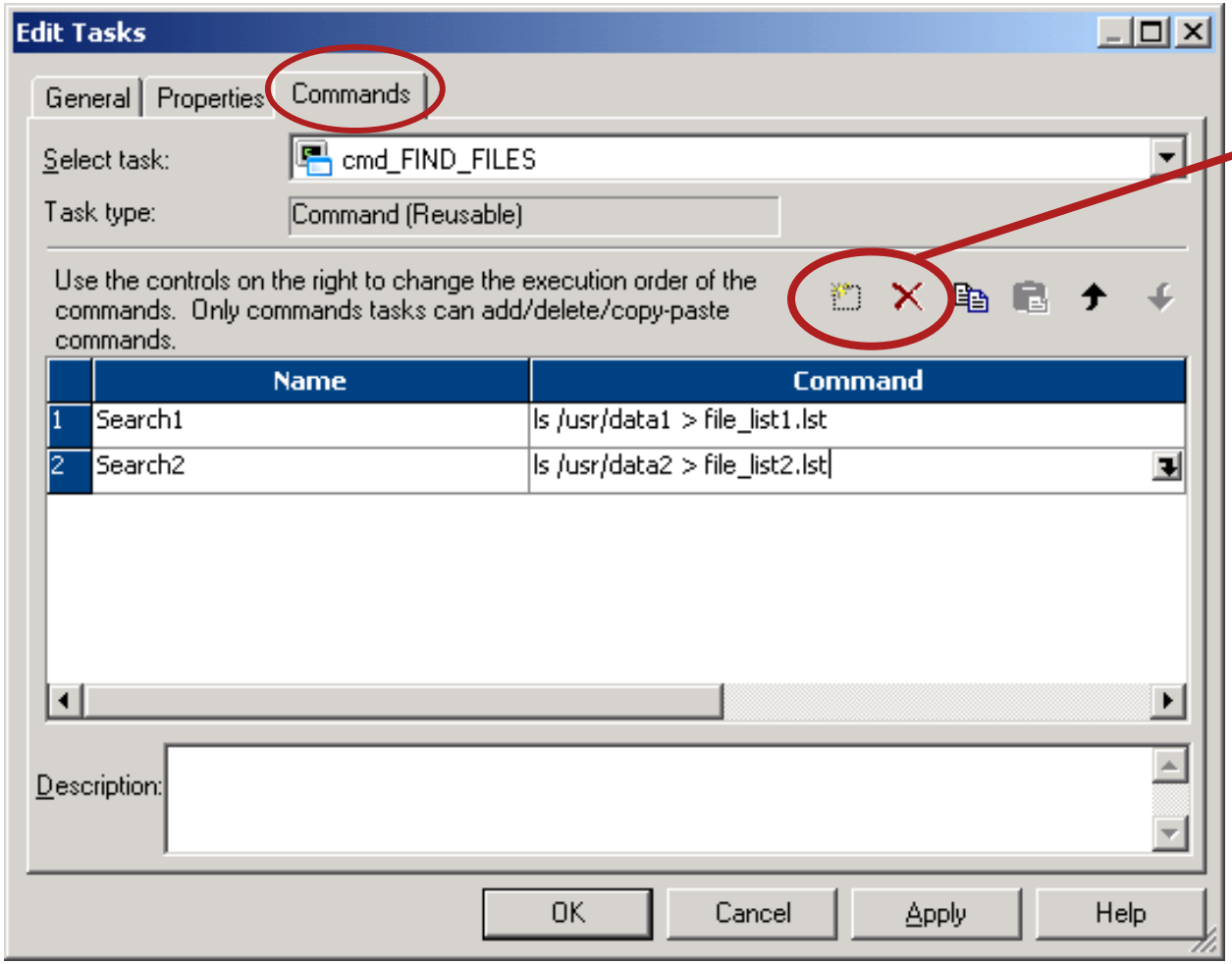
INFORMATICA

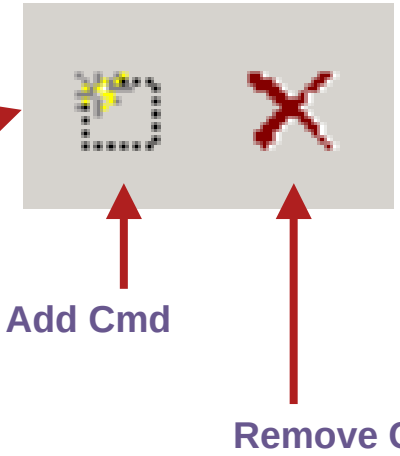
Command Task (cont'd)



INFORMATICA

Command Task (cont'd)





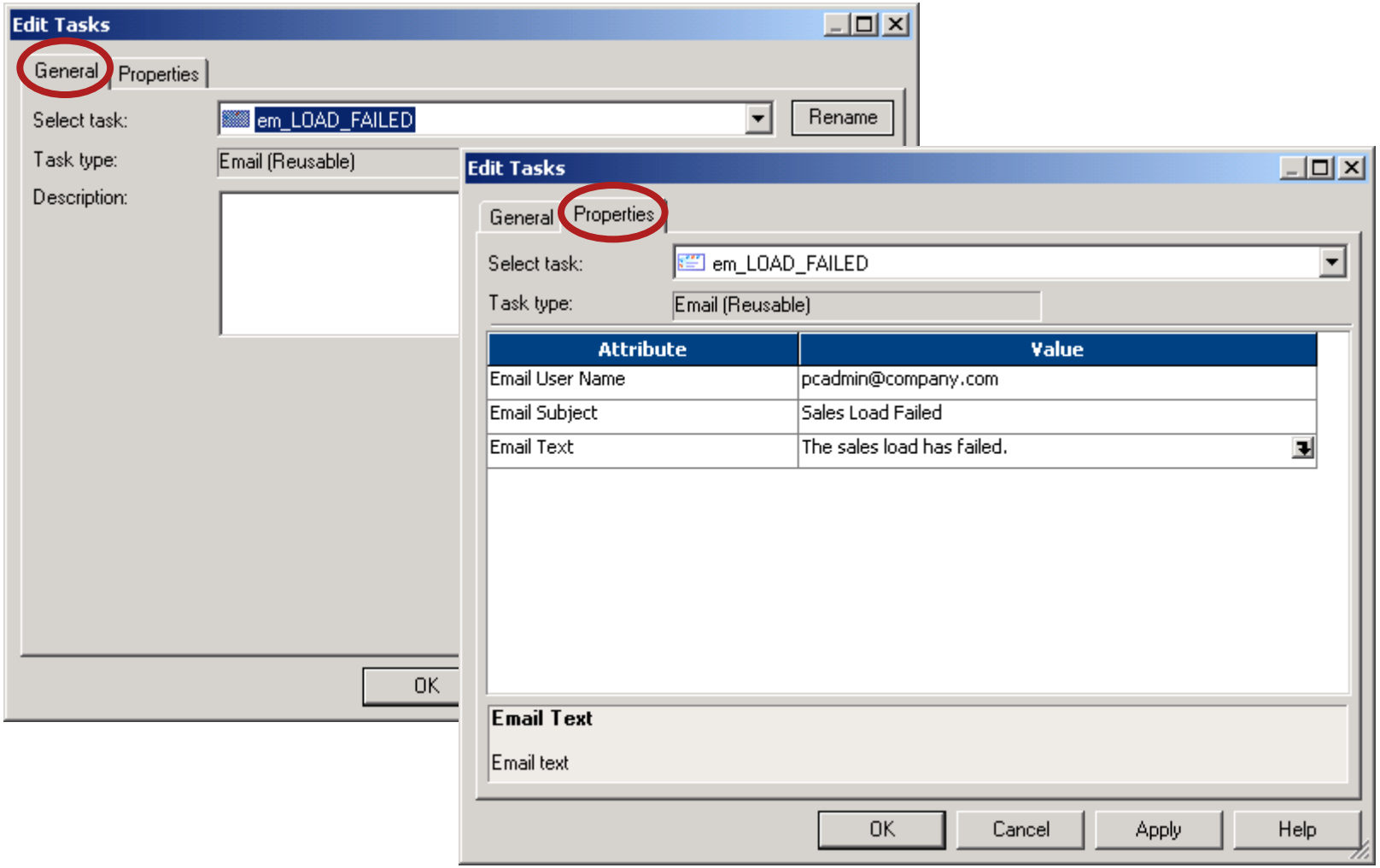
INFORMATICA

Email Task

- Configure to have the Informatica Server to send email at any point in the Workflow
- Becomes a component in a Workflow (or Worklet)
- If configured in the Task Developer, the Email Task is reusable (optional)
- Emails can also be invoked under the *Components* tab of a Session task to run pre- or post-session

INFORMATICA

Email Task (cont'd)



INFORMATICA

Lab 19 – Sequential Workflow and Error Logging

INFORMATICA

Lab 20 – Command Task

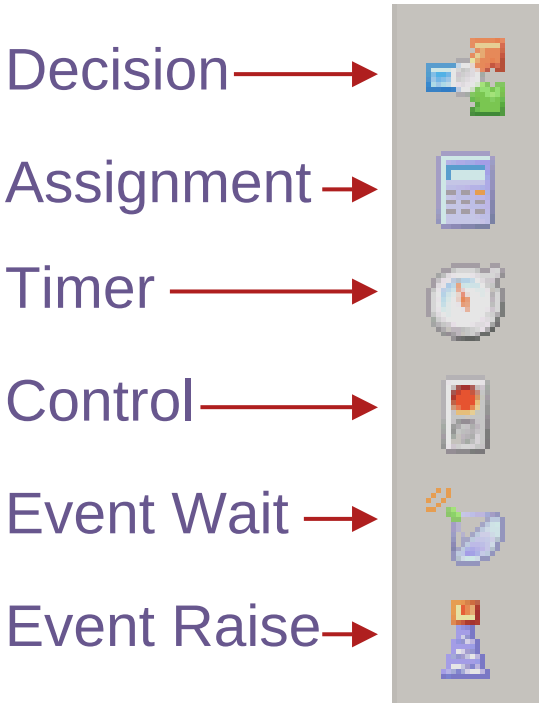
INFORMATICA

Non-Reusable Tasks

INFORMATICA

Non-Reusable Tasks

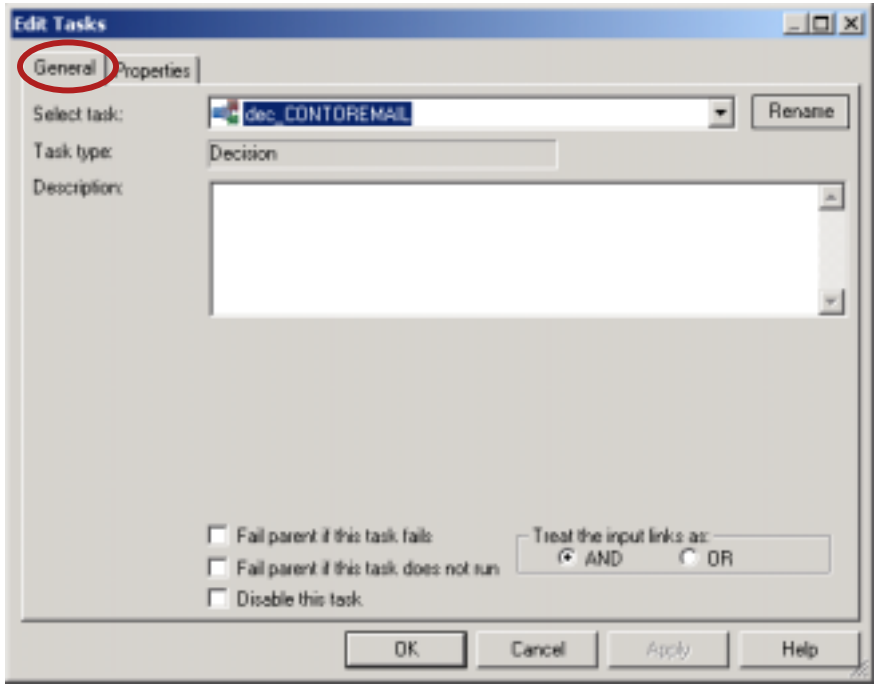
- **Six additional Tasks are available in the Workflow Designer**

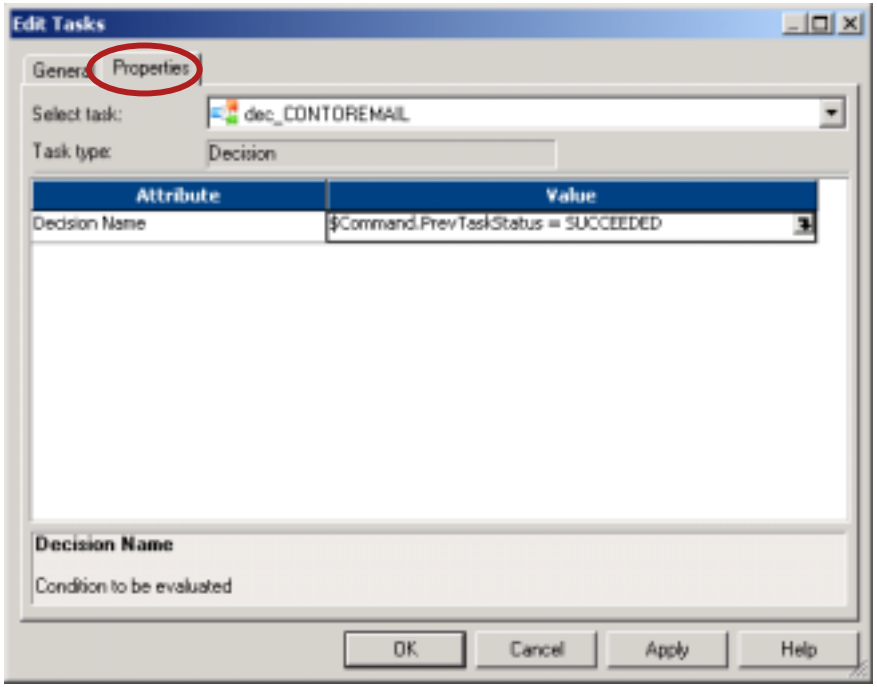


INFORMATICA

Decision Task

- **Specifies a condition to be evaluated in the Workflow**
- **Use the Decision Task in branches of a Workflow**
- **Use link conditions downstream to control execution flow by testing the Decision result**

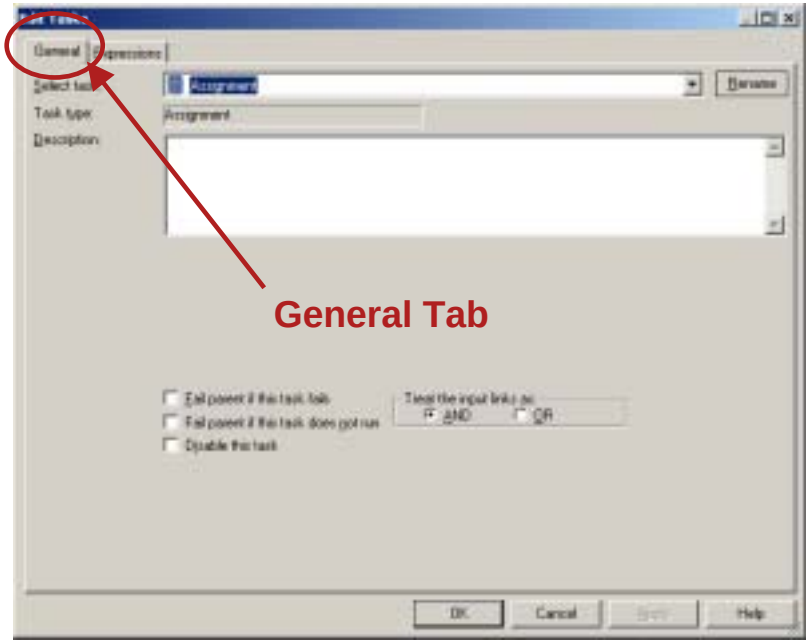


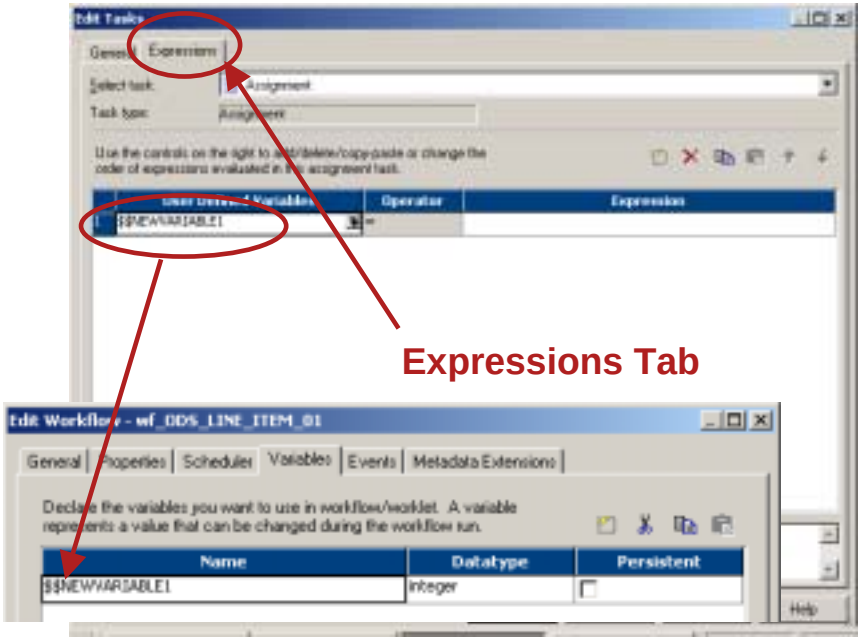


INFORMATICA

Assignment Task

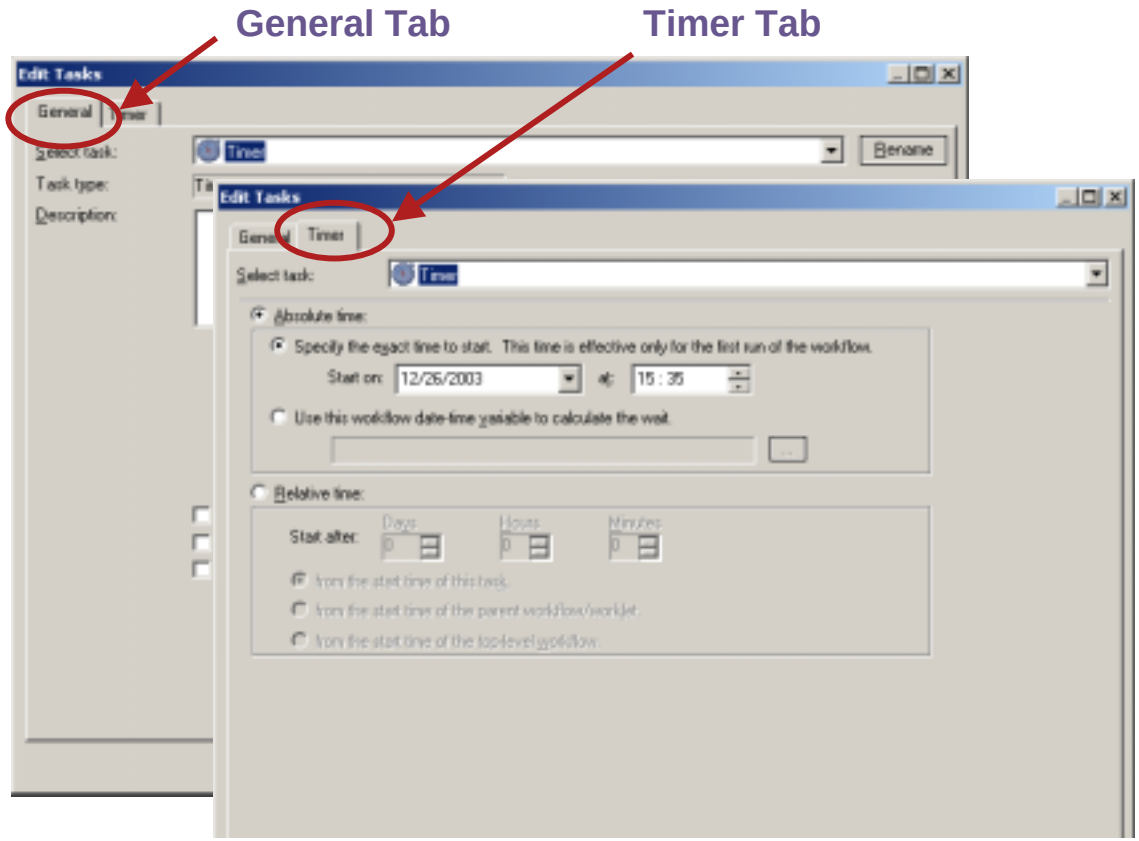
- **Assigns a value to a Workflow Variable**
- **Variables are defined in the Workflow object**





Timer Task

- **Waits for a specified period of time to execute the next Task**



- Absolute Time
- Datetime Variable
- Relative Time

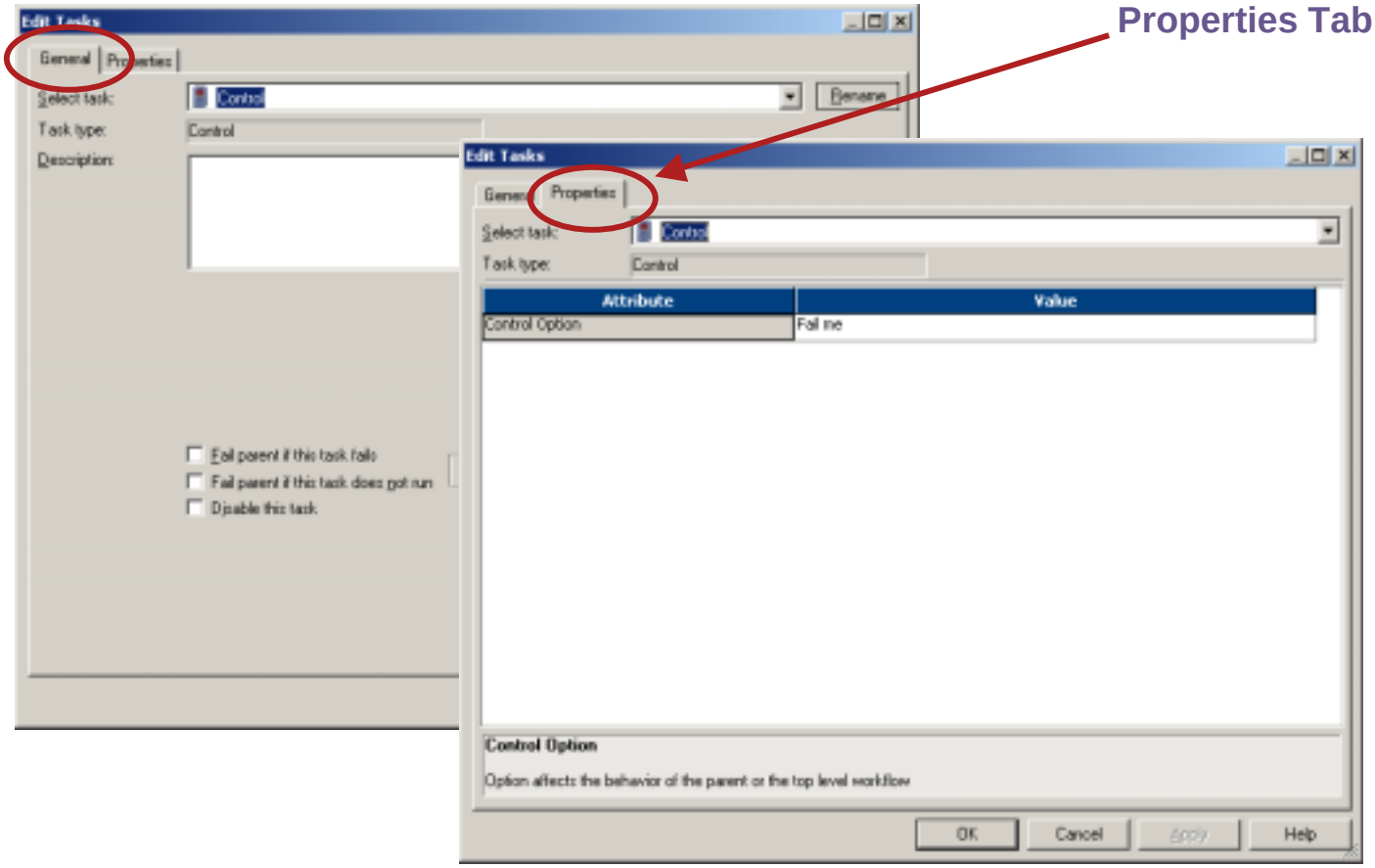
INFORMATICA

Control Task

- **Stop or ABORT the Workflow**

General Tab

Properties Tab



INFORMATICA

Event Wait Task

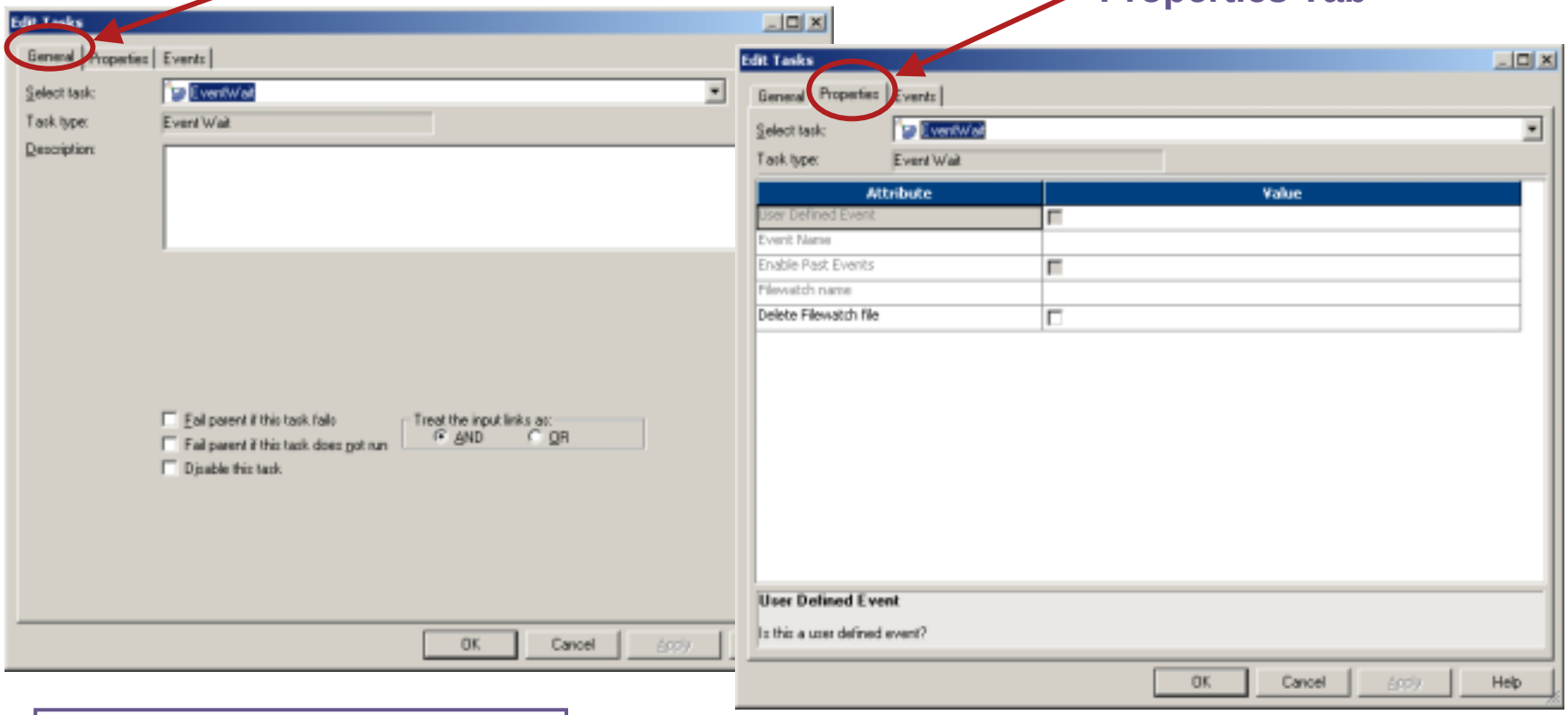
- Waits for a user-defined or a pre-defined event to occur
- Once the event occurs, the Informatica Server completes the rest of the Workflow
- Used with the Event Raise Task
- Events can be a file watch (indicator file) or user-defined
- User-defined events are defined in the Workflow itself

INFORMATICA

Event Wait Task (cont'd)

General Tab

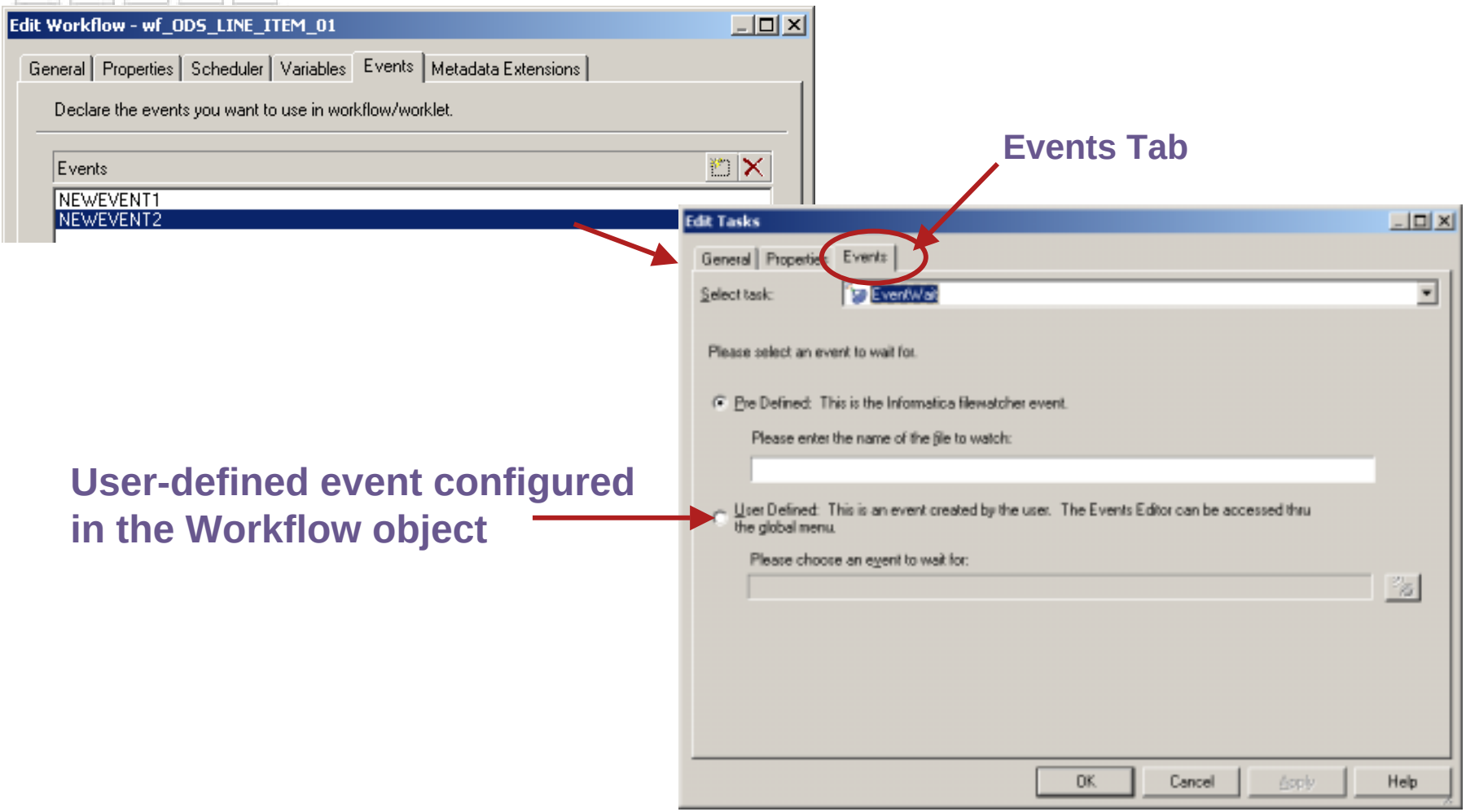
Properties Tab



Used with the Event Raised Task

INFORMATICA

Event Wait Task (cont'd)

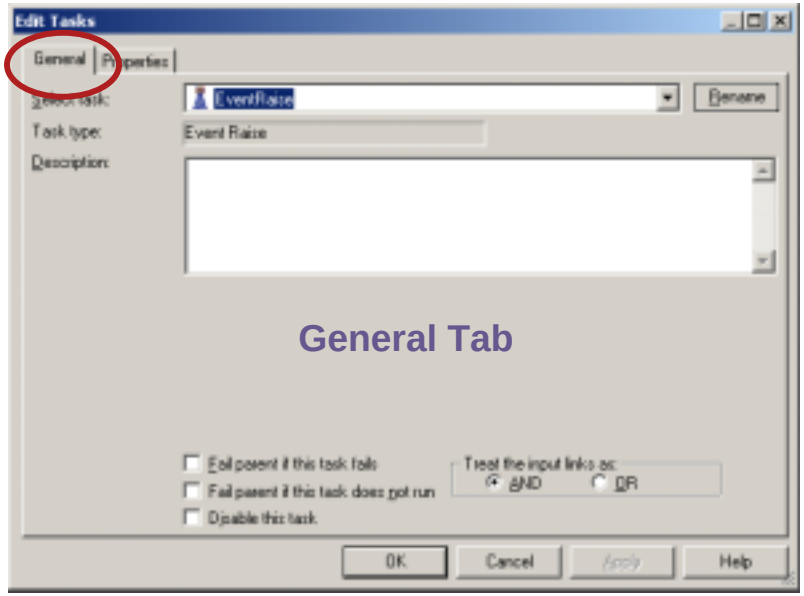


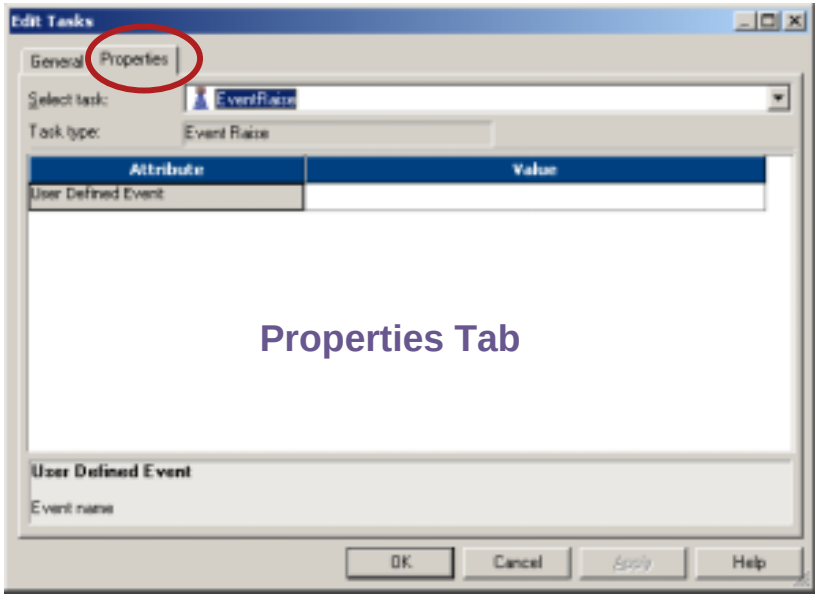
User-defined event configured in the Workflow object

INFORMATICA

Event Raise Task

- Represents the location of a user-defined event
- The Event Raise Task triggers the user-defined event when the Informatica Server executes the Event Raise Task





Used with the Event Wait Task

INFORMATICA

Worklets

INFORMATICA

Worklets

- An object representing a set or grouping of Tasks
- Can contain any Task available in the Workflow Manager
- Worklets expand and execute inside a Workflow
- A Workflow which contains a Worklet is called the "parent Workflow"
- Worklets CAN be nested
- Reusable Worklets – create in the Worklet Designer
- Non-reusable Worklets – create in the Workflow Designer

INFORMATICA

Re-usable Worklet

- In the Worklet Designer, select Worklets | Create

Informatica PowerCenter Workflow Manager - [Worklet Designer - Student01 - [PC_Course]]

Repository Edit View Tools Layout Versioning Worklets Tasks Server Connections Window Help

Repository Navigator

Worklets
- wl_combined_sessions
- wl_CUSTOMER_SALES_INFO_01

Workflows

Student02
Student03
Student04
Student05
Student06
Student07
Student08
Student09
Student10

Worklets Node

Task Developer | Worklet Designer | Workflow Designer

wl_CUSTOMER_SALES_INFO_01

Worklet Designer

Start
s_Shortcut_to_m_DIM_CUSTOMER_01
s_Shortcut_to_m_DIM_INVOICE_01
s_Shortcut_to_m_DIM_PRODUCT_xx
s_Shortcut_to_m_DIM_REP_01
s_Shortcut_to_m_DIM_TIME_01
s_Shortcut_to_m_DIM_VENDOR_01

Tasks in a Worklet

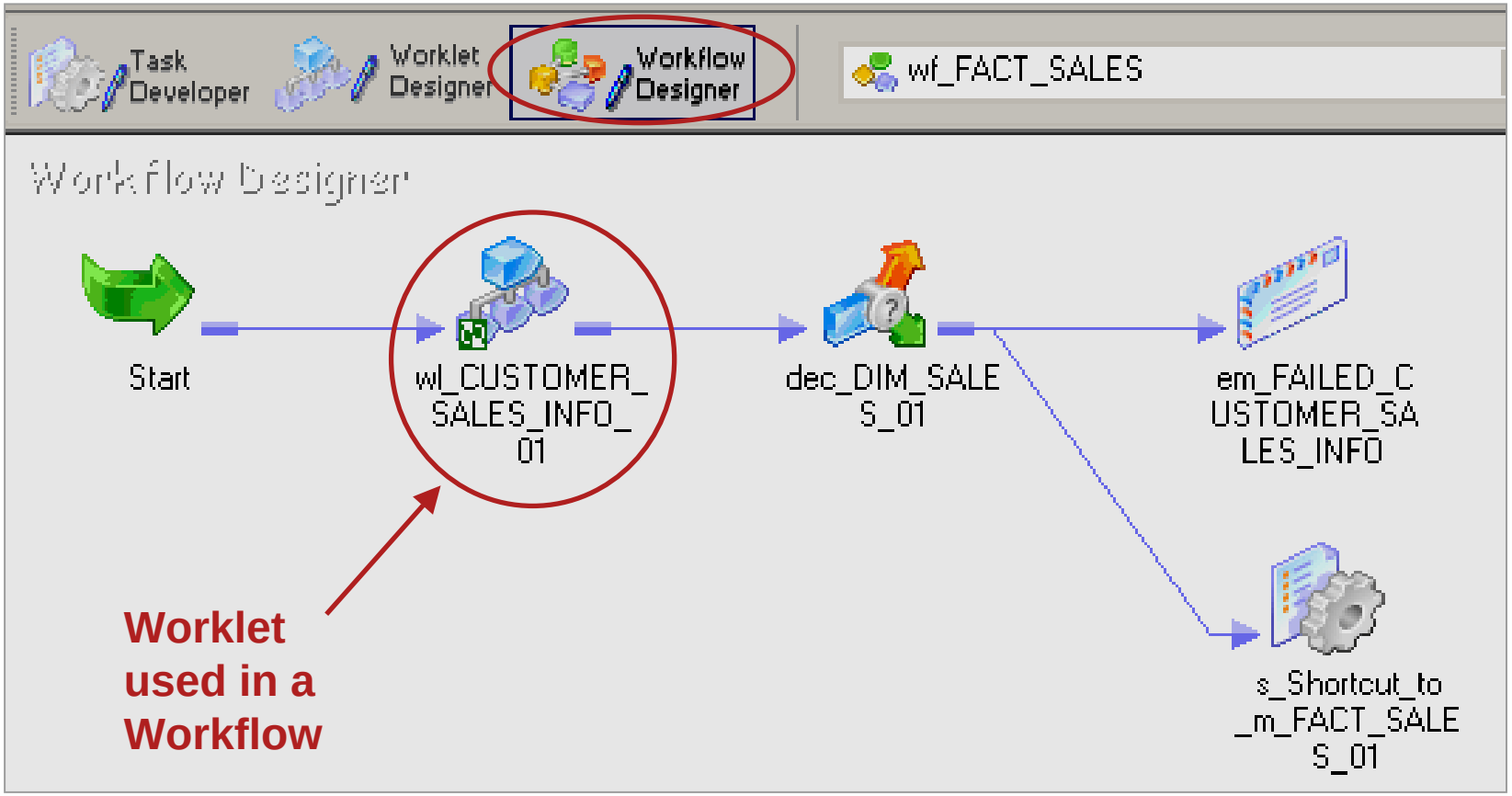
Start Page | Student01 - ...

Save | Fetch Log | Validate | Copy | Server | Notifications

Ready

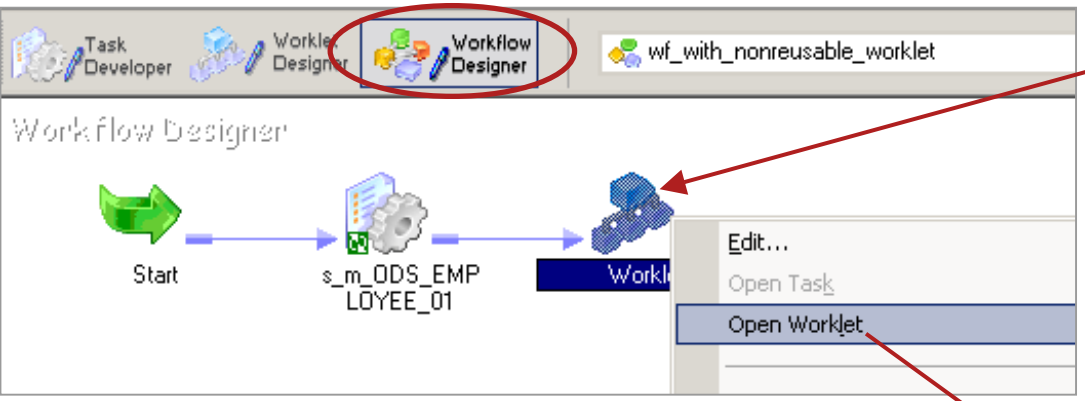
INFORMATICA

Using a Reusable Worklet in a Workflow



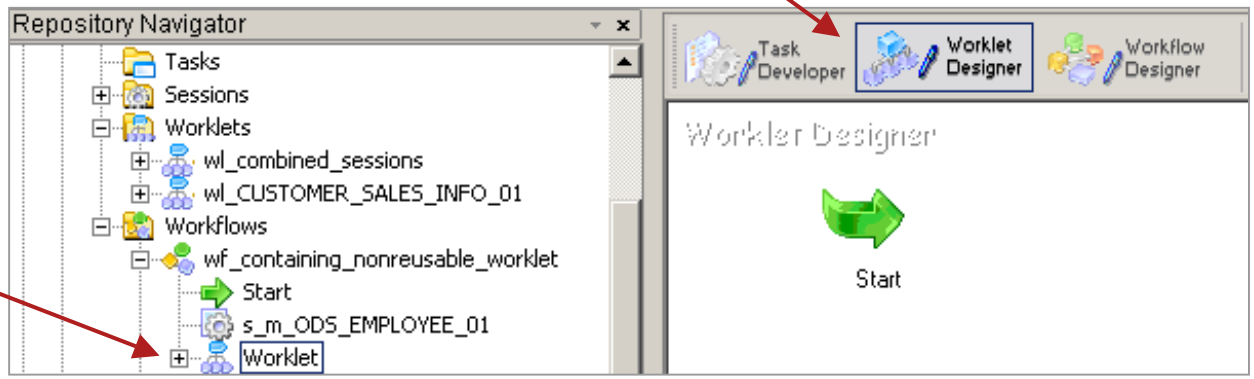
INFORMATICA

Non-Reusable Worklet



1. Create worklet task in Workflow Designer
2. Right-click on new worklet and select Open Worklet
3. Workspace switches to **Worklet Designer**

NOTE: Worklet shows only under Workflows node



INFORMATICA

Lab 21 – Reusable Worklet and Decision Task

INFORMATICA

Lab 22 – Event Wait with Pre-Defined Event

INFORMATICA

Lab 23 – User-Defined Event, Event Raise, and Event Wait

INFORMATICA

INFORMATICA®

Turning integration into insight.™