



Informatica Power Center 9.x

Introduction

At the end of this course you will -

- Understand how to use all major PowerCenter components
- Be able to build basic ETL Mappings and Maplets
- Understand the different Transformations and their basic attributes in PowerCenter
- Be able to create, run and monitor Workflows
- Be able to troubleshoot common development problems

This section will include -

- Concepts of ETL
- PowerCenter Architecture
- Connectivity between PowerCenter components

Extract, Transform, and Load

Operational Systems



- Transaction level data
- Optimized for Transaction Response Time
- Current
- Normalized or De-Normalized data

Extract

Decision Support



- Aggregated data
- Historical

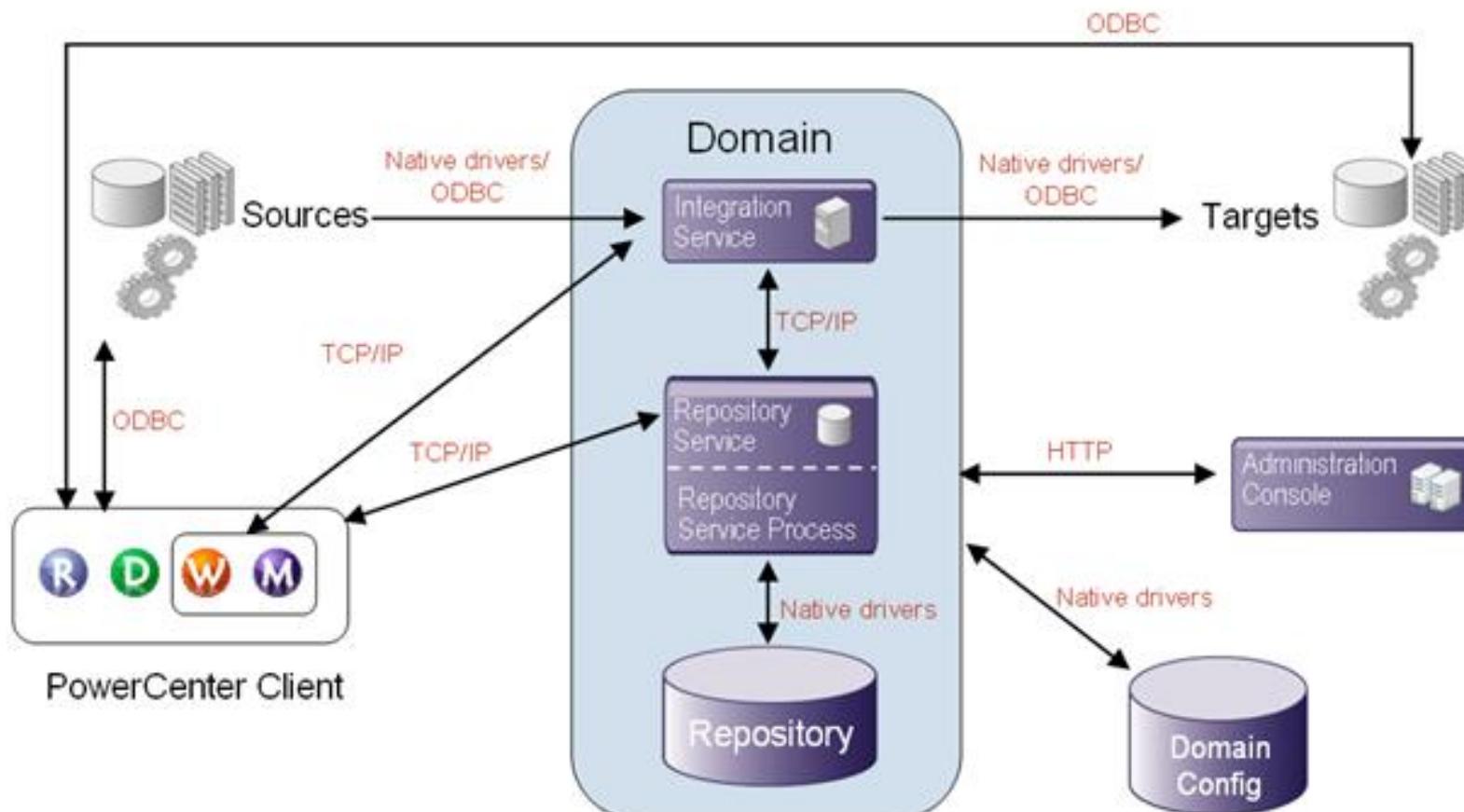
Cleanse Data
Apply Business Rules
Aggregate Data
Consolidate Data
De-normalize

Transform

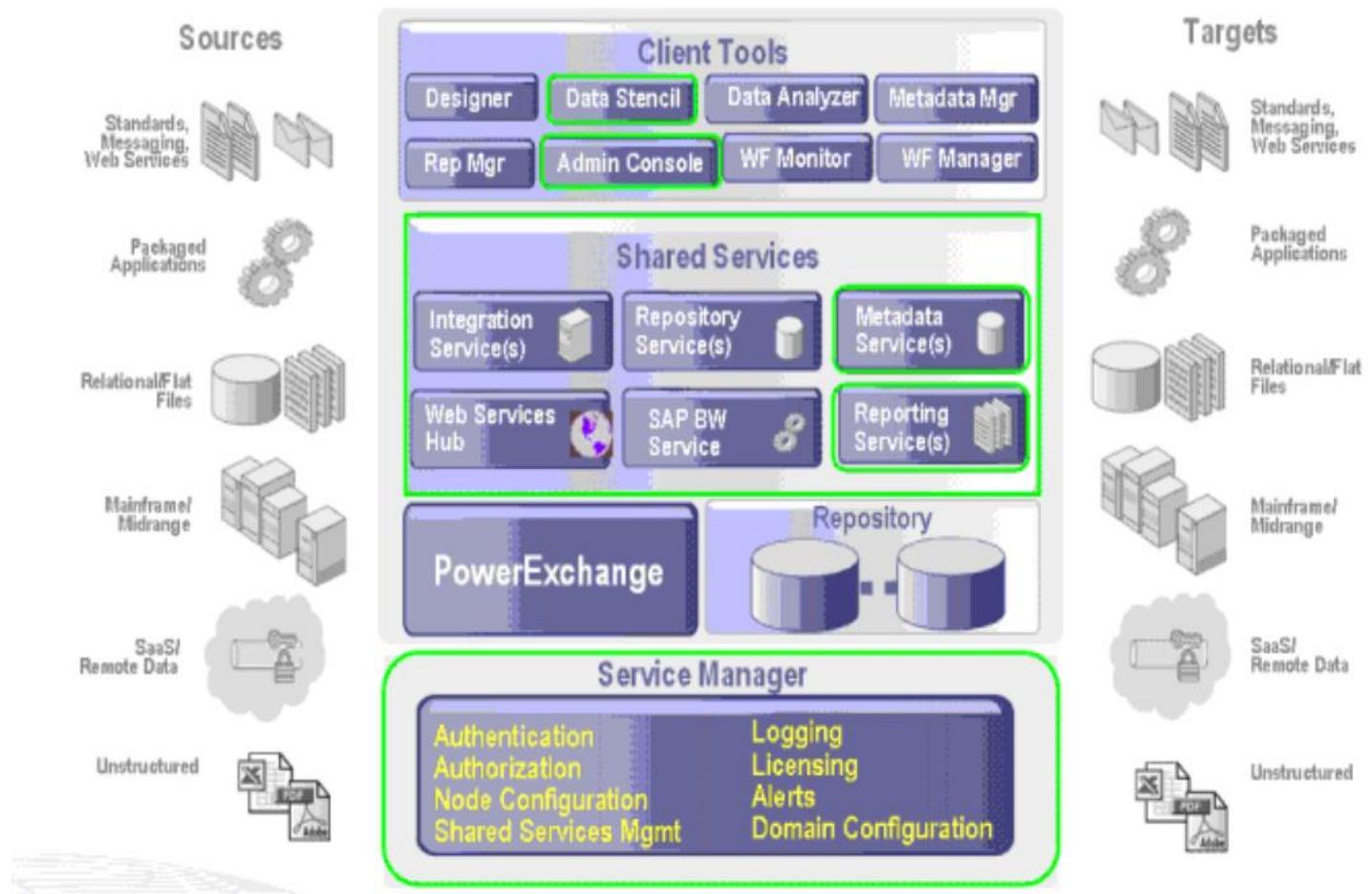


Load

PowerCenter Architecture



PowerCenter Components



PowerCenter Components

- PowerCenter Repository
 - Repository Service
 - Integration Service
 - PowerCenter Client
 - Administration Console
 - Repository Manager
 - Designer
 - Workflow Manager
 - Workflow Monitor
 - External Components
 - Sources
 - Targets
- 
- Application Services*

This section includes -

- The purpose of the Repository and Repository Service
- Admin Console
- The Repository Manager
- Security and privileges
- Object sharing, searching and locking

PowerCenter Repository

- It is a relational database managed by the Repository Service
- Stores metadata about the objects (mappings, transformations etc.) in database tables called as Repository Content
- The Repository database can be in Oracle, IBM DB2 UDB, MS SQL Server or Sybase ASE
- To create a repository service one must have full privileges in the Administrator Console and also in the domain
- Integration Service uses repository objects for performing the ETL

Repository Service

- A Repository Service process is a multi-threaded process that fetches, inserts and updates metadata in the repository
- Manages connections to the Repository from client applications and Integration Service
- Maintains object consistency by controlling object locking
- Each Repository Service manages a single repository database. However multiple repositories can be connected and managed using repository domain
- It can run on multiple machines or nodes in the domain. Each instance is called a Repository Service process

Administration Console

A web-based interface used to administer the PowerCenter domain.

Following tasks can be performed:

- Manage the domain
- Shutdown and restart domain and Nodes
- Manage objects within a domain
- Create and Manage Folders, Grid, Integration Service, Node, Repository Service, Web Service and Licenses

Administration Console

- Enable/ Disable various services like the Integration Services, Repository Services etc.
- Upgrade Repositories and Integration Services
- View log events for the domain and the services
- View locks
- Add and Manage Users and their profile
- Monitor User Activity
- Manage Application Services

Repository Manager



Use Repository manager to navigate through multiple folders and repositories.

Perform following tasks:

- Add/Edit Repository Connections
- Implement Repository Security (By changing the password only)
- Perform folder functions (Create , Edit , Delete ,Compare)
- Compare Repository Objects
- Manage Workflow/Session Log Entries
- View Dependencies
- Exchange Metadata with other BI tools

We will walk through -

- Design Process
- PowerCenter Designer Interface
- Mapping Components

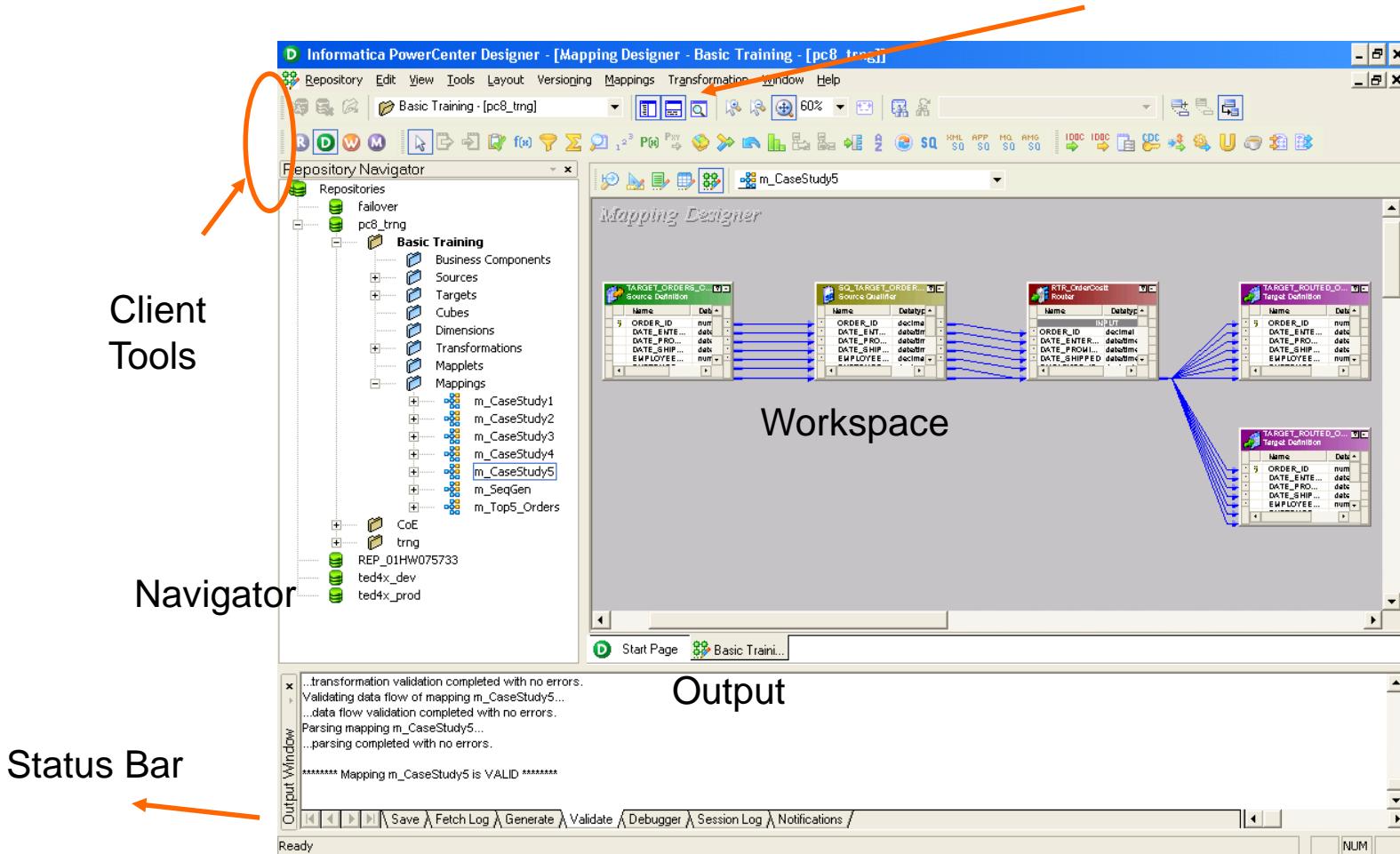
Design Process

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

Designer- Interface



Overview Window

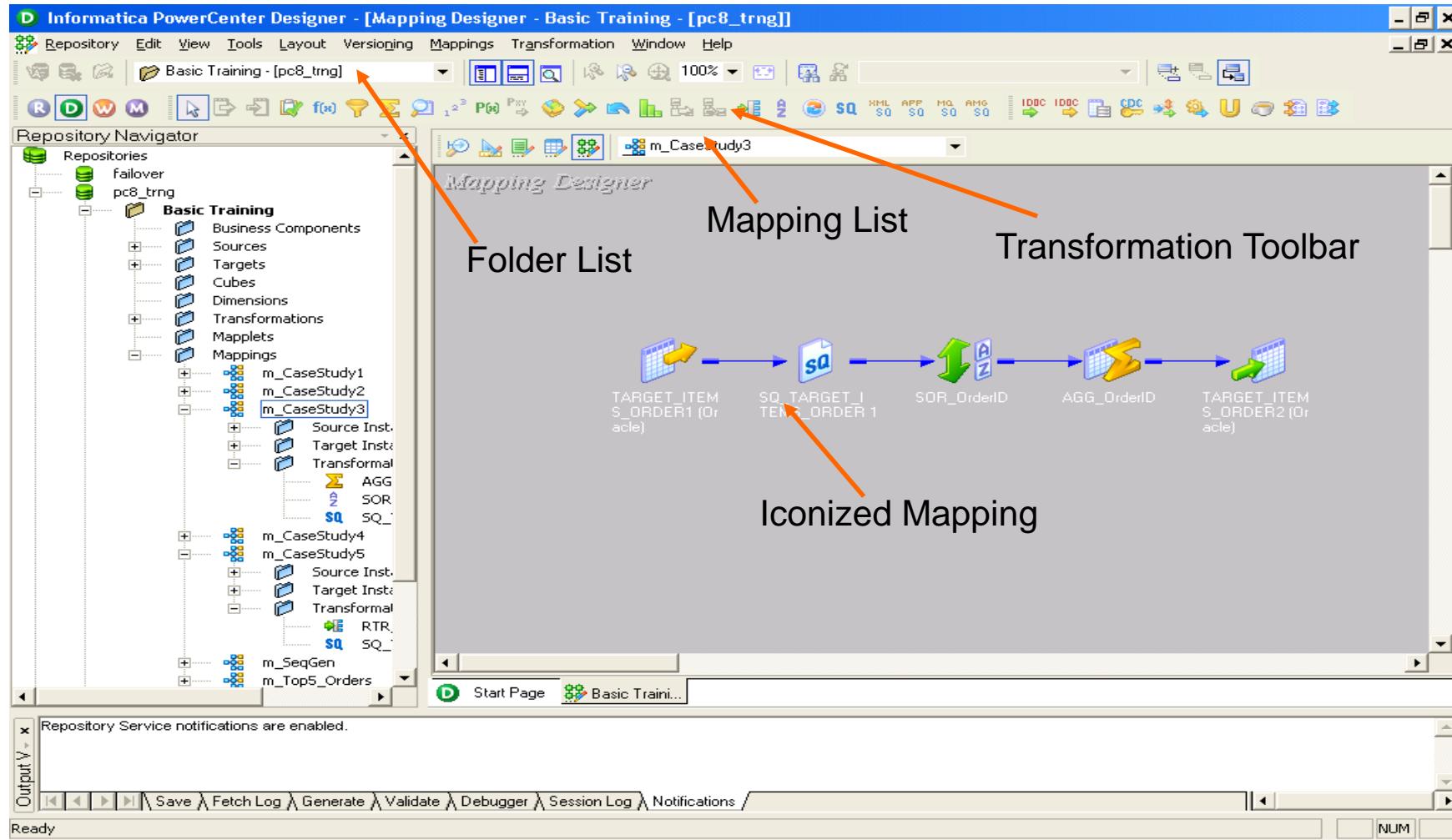




Mapping Components

- Each PowerCenter mapping consists of one or more of the following mandatory components
 - Sources
 - Transformations
 - Targets
- The components are arranged sequentially to form a valid data flow from Sources → Transformations → Targets

Designer- Interface



Designer- Source Analyzer



Diagram illustrating the Source Analyzer in Informatica PowerCenter Designer, showing dependencies between tables.

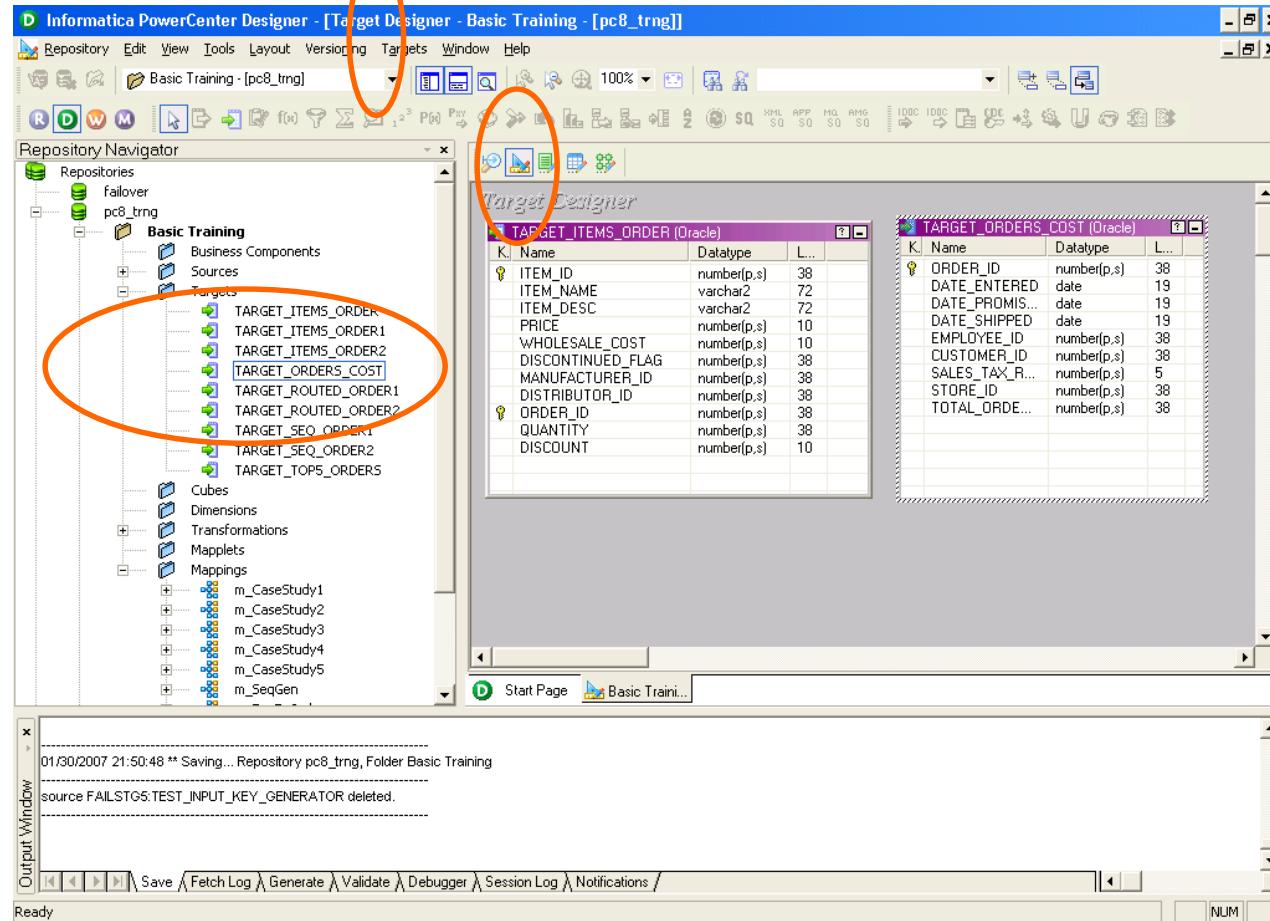
The screenshot shows the PowerCenter Designer interface with the following components:

- Repository Navigator:** Displays the repository structure. A red circle highlights the "Basic Training" folder under "Sources". Inside "Basic Training", a red circle highlights the "TUTORIAL_SOURCE" folder, which contains tables: CUSTOMERS, DISTRIBUTORS, EMPLOYEES, ITEMS, JOBS, MANUFACTURERS, ORDERS, ORDER_ITEMS, and STORES.
- Source Analyzer:** Two tables are displayed: "CUSTOMERS (Oracle)" and "ORDERS (Oracle)".
- Relationship:** A blue arrow labeled "Foreign Key" points from the "CUSTOMERS" table to the "ORDERS" table, indicating a relationship between the two.
- Output Window:** Shows log messages:

```
01/30/2007 21:50:48 ** Saving... Repository pc8_trng, Folder Basic Training
source FAILSTG5:TEST_INPUT_KEY_GENERATOR deleted.
```

It Shows the Dependencies of the tables also

Designer- Target Designer



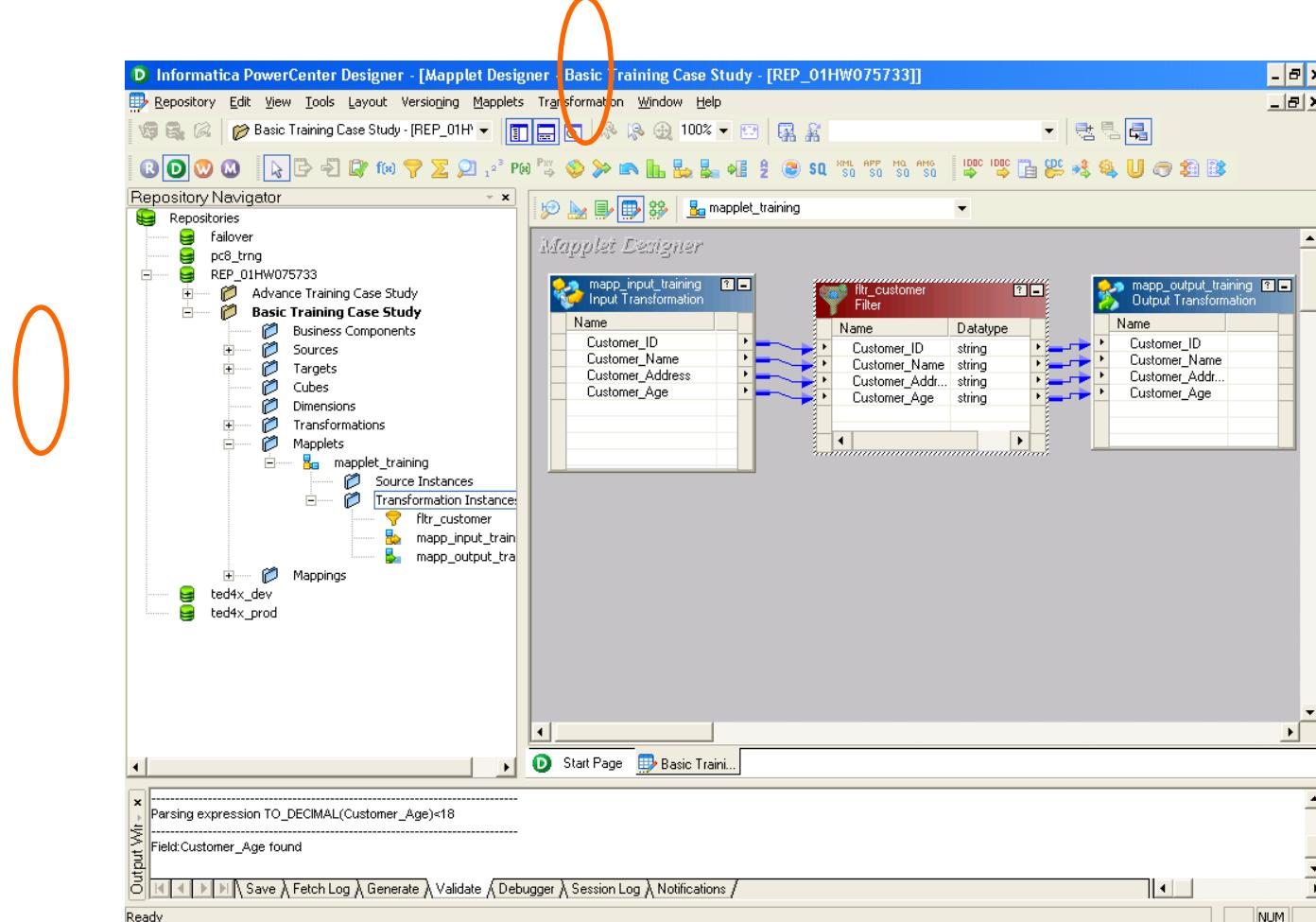
Transformation Developer



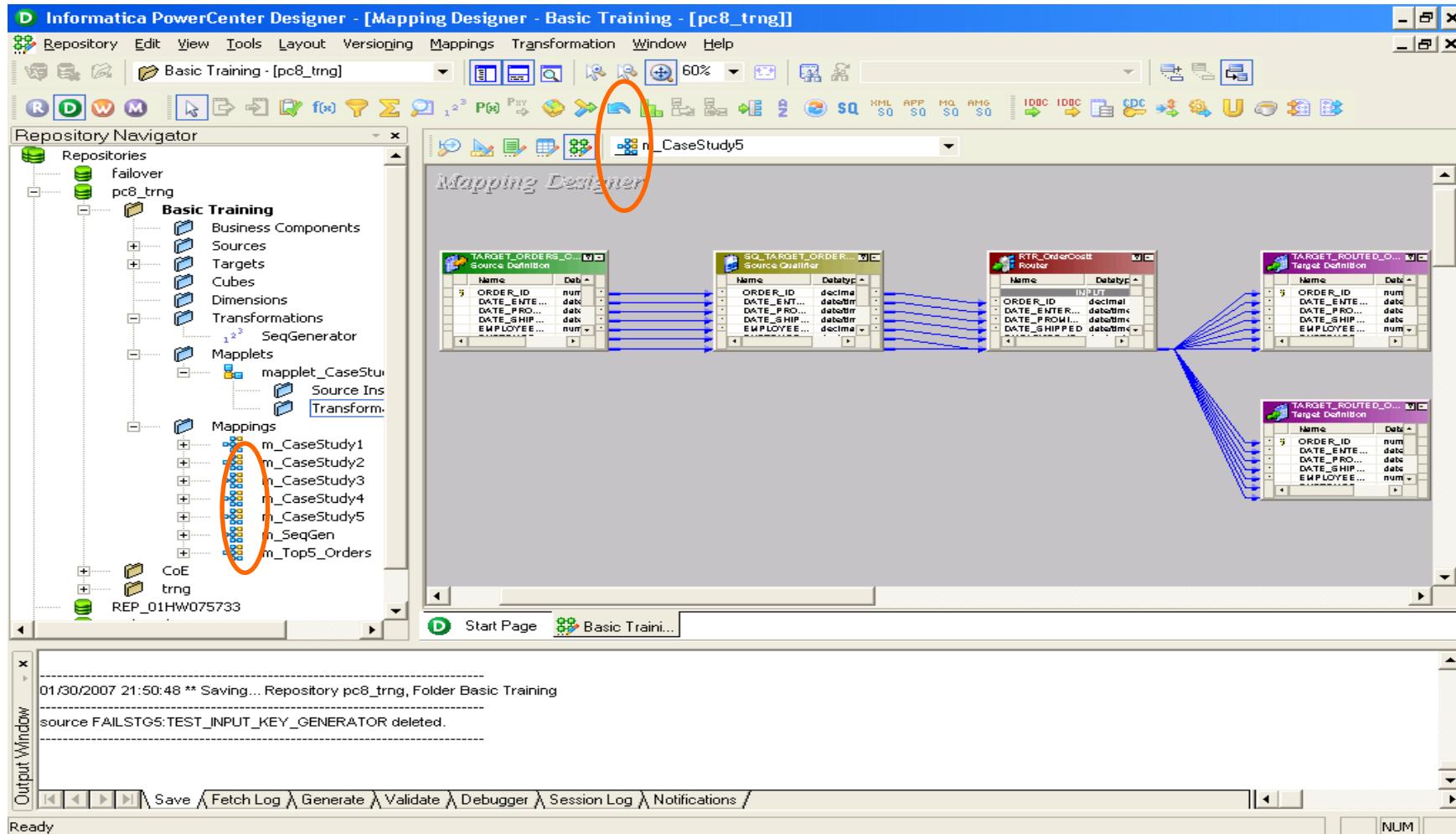
01/30/2007 21:50:48 ** Saving... Repository pc8_trng, Folder Basic Training
source FAILSTGS:TEST_INPUT_KEY_GENERATOR deleted.

Transformation Developer is used only for creating reusable transformations

Mapplet Designer



Designer- Mapping Designer



This section introduces to -

- Different Source Types
- Creation of ODBC Connections
- Creation of Source Definitions
- Source Definition properties
- Data Preview option

Source Analyzer



Navigation Window

Analyzer Window

Informatica PowerCenter Designer - [Source Analyzer - Basic Training - [pc8_trng]]

Repository Edit View Tools Layout Versioning Sources Window Help

Basic Training - [pc8_trng]

Source Analyzer

CUSTOMERS (Oracle)

Name	Datatype	Length/Precision
CUSTOMER_ID	number(p,s)	38
COMPANY	varchar2	50
FIRST_NAME	varchar2	30
LAST_NAME	varchar2	30
ADDRESS1	varchar2	72
ADDRESS2	varchar2	72
CITY	varchar2	30
STATE	varchar2	2
POSTAL_CODE	varchar2	10
PHONE	varchar2	30
EMAIL	varchar2	30

ORDERS (Oracle)

Name	Datatype	Length/Precision
ORDER_ID	number(p,s)	3...
DATE_ENT...	date	1...
DATE_PRO...	date	1...
DATE_SHI...	date	1...
EMPLOYEE...	number(p,s)	3...
CUSTOMER...	number(p,s)	3...
SALES_TA...	number(p,s)	5
STORE_ID	number(p,s)	3...

MANUFACTURERS (Oracle)

Name	Datatype	Length/Precision
MANUFACTURER_ID	number(p,s)	38
MANUFACTURER_NAME	varchar2	48

Start Page Basic Train...

Repository Service notifications are enabled.

01/31/2007 13:07:56-[REP_55102] Failed to connect to repository service [pc8_trng].

01/31/2007 14:35:59-[REP_55102] Failed to connect to repository service [pc8_trng].

01/31/2007 14:41:35-[REP_55102] Failed to connect to repository service [pc8_trng].

Output Window

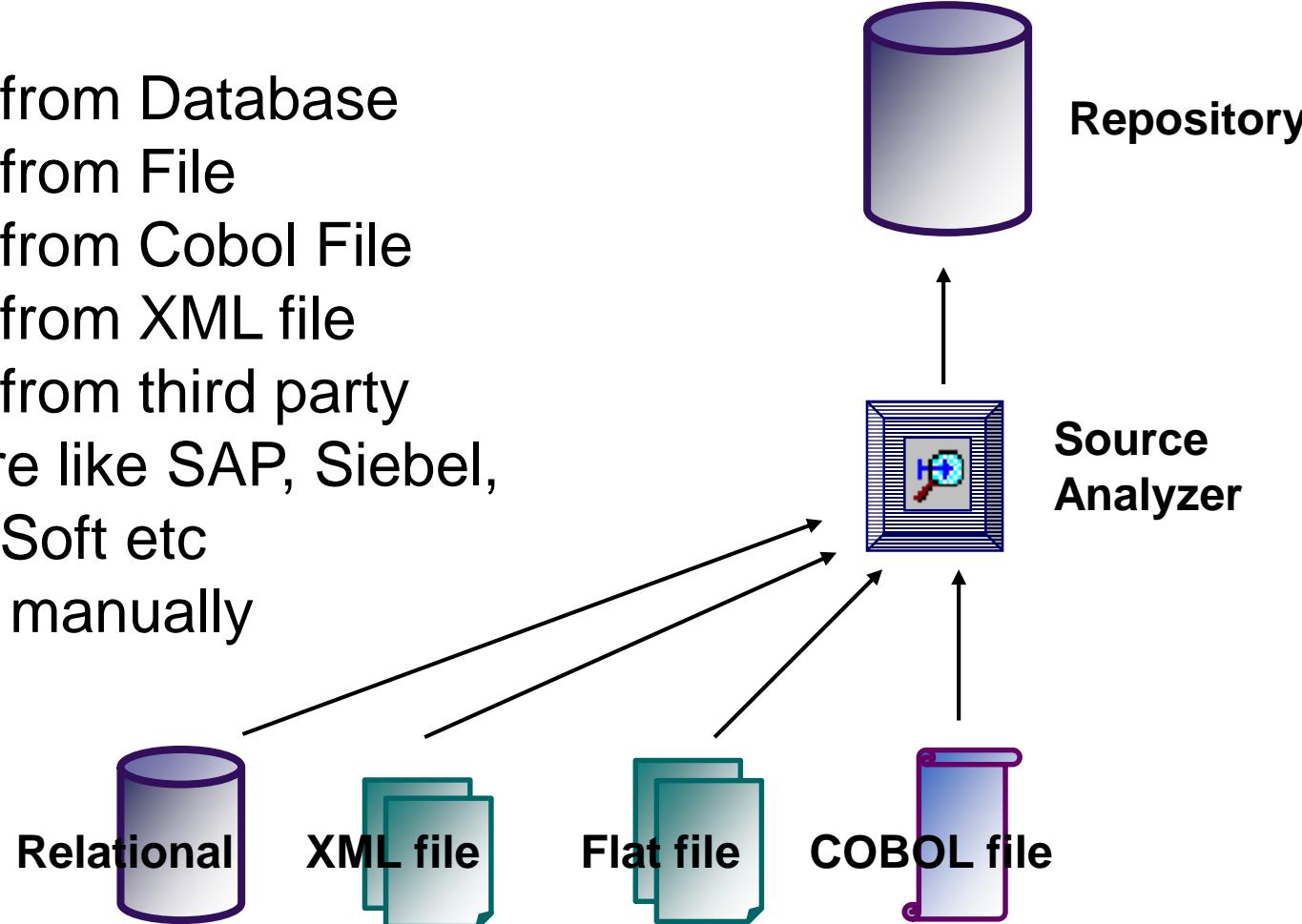
Save Fetch Log Generate Validate Debugger Session Log Notifications

Ready

Methods of Analyzing Sources

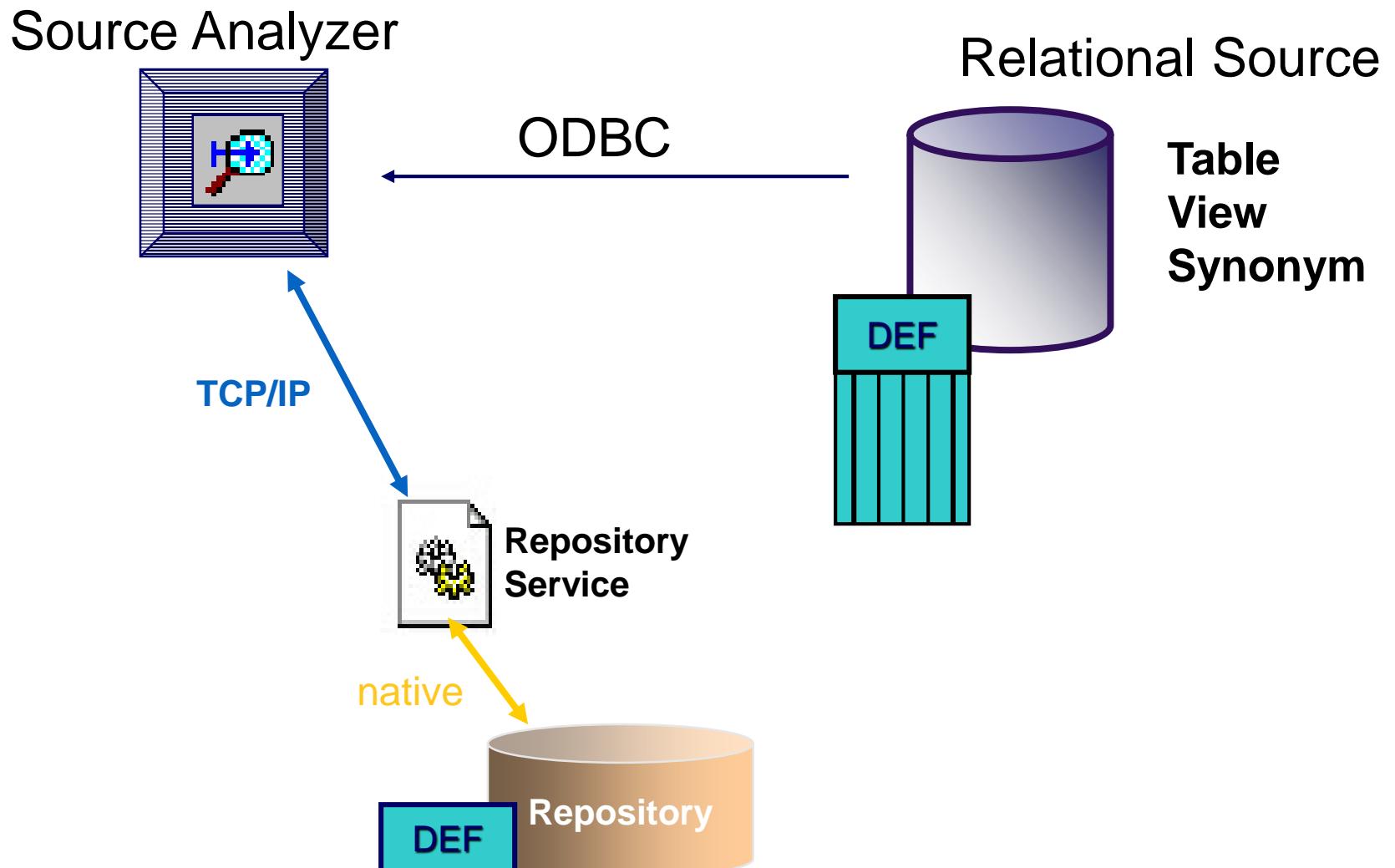


- Import from Database
- Import from File
- Import from Cobol File
- Import from XML file
- Import from third party software like SAP, Siebel, PeopleSoft etc
- Create manually





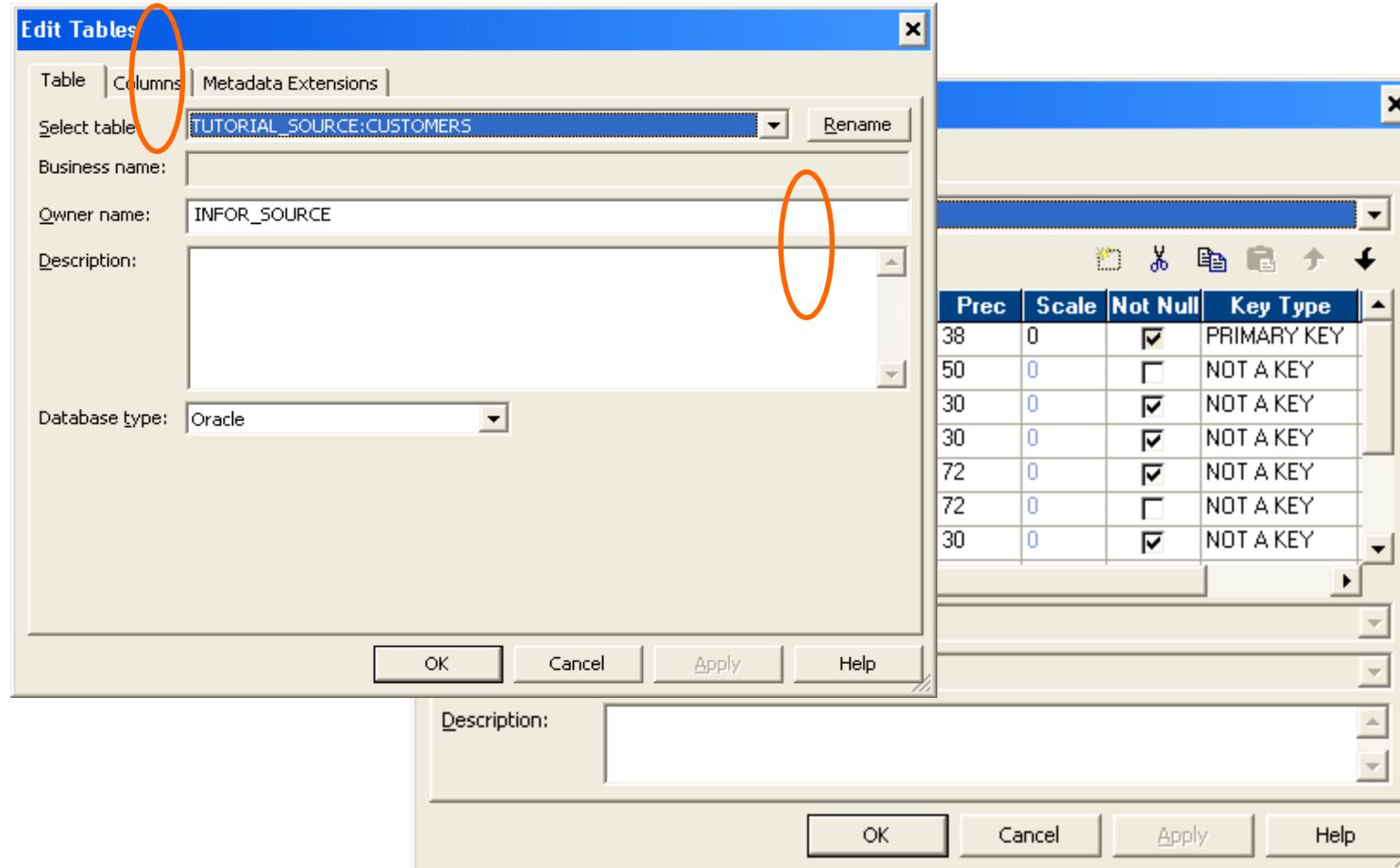
Analyzing Relational Sources



Analyzing Relational Sources



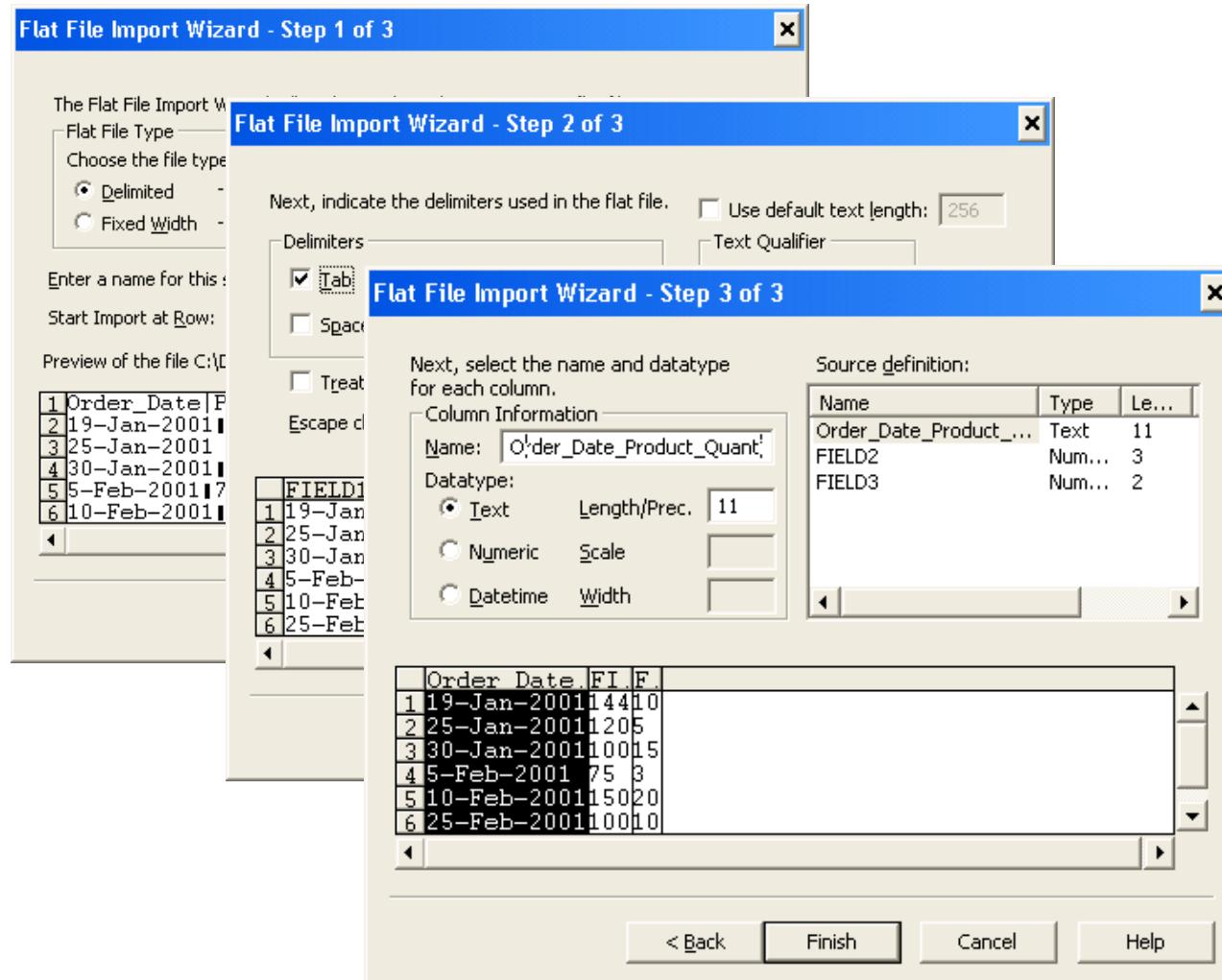
Editing Source Definition Properties





Flat File Wizard

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



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

Target Designer



Informatica PowerCenter Designer - [Target Designer - Basic Training - [pc8_trng]]

Repository Edit View Tools Layout Versioning Targets Window Help

R D W M

Repository Navigator

- Repositories
 - Failover
 - pc8_trng
 - Basic Training**
 - Business Components
 - Sources
 - Targets
 - TARGET_ITEMS_ORDER**
 - TARGET_ITEMS_ORDER1**
 - TARGET_ITEMS_ORDER2**
 - TARGET_ORDERS_COST** (highlighted)
 - TARGET_ROUTED_ORDER1**
 - TARGET_ROUTED_ORDER2**
 - TARGET_SEQ_ORDER1**
 - TARGET_SEQ_ORDER2**
 - TARGET_TOP5_ORDERS**
 - Cubes
 - Dimensions
 - Transformations
 - Mapplets
 - Mappings
 - m_CaseStudy1**
 - m_CaseStudy2**
 - m_CaseStudy3**
 - m_CaseStudy4**
 - m_CaseStudy5**
 - m_SeqGen**

K.	Name	Datatype	L...
1	ITEM_ID	number(p,s)	38
2	ITEM_NAME	varchar2	72
3	ITEM_DESC	varchar2	72
4	PRICE	number(p,s)	10
5	WHOLESALE_COST	number(p,s)	10
6	DISCONTINUED_FLAG	number(p,s)	38
7	MANUFACTURER_ID	number(p,s)	38
8	DISTRIBUTOR_ID	number(p,s)	38
9	ORDER_ID	number(p,s)	38
10	QUANTITY	number(p,s)	38
11	DISCOUNT	number(p,s)	10

K.	Name	Datatype	L...
1	ORDER_ID	number(p,s)	38
2	DATE_ENTERED	date	19
3	DATE_PROMISED	date	19
4	DATE_SHIPPED	date	19
5	EMPLOYEE_ID	number(p,s)	38
6	CUSTOMER_ID	number(p,s)	38
7	SALES_TAX_R...	number(p,s)	5
8	STORE_ID	number(p,s)	38
9	TOTAL_ORDER...	number(p,s)	38

Output Window

```
01/30/2007 21:50:48 ** Saving... Repository pc8_trng, Folder Basic Training
source FAILSTG5.TEST_INPUT_KEY_GENERATOR deleted.
```

Save Fetch Log Generate Validate Debugger Session Log Notifications

Ready

Creating Target Definitions

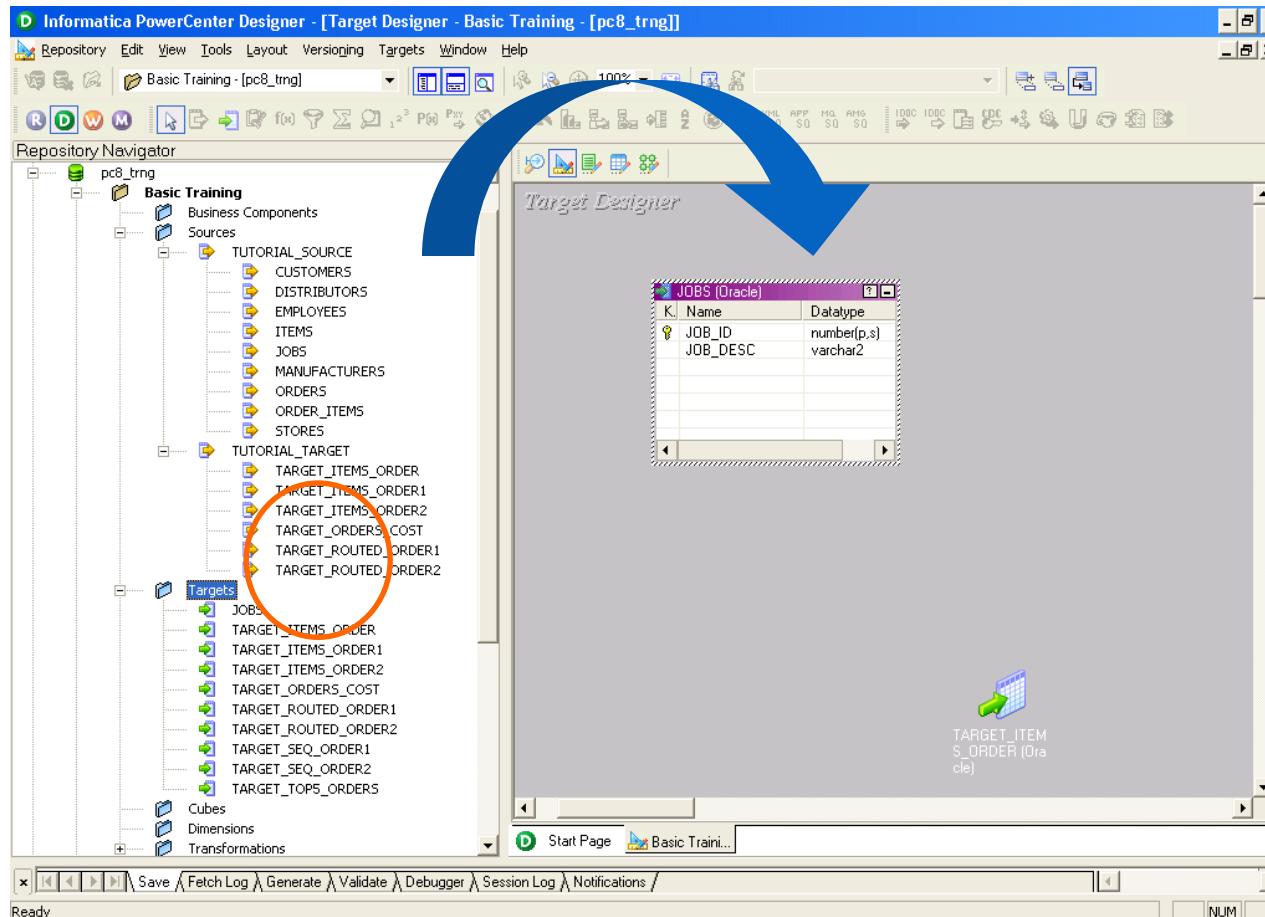
Methods of creating Target Definitions

- Import from Database
- Import from an XML file
- Import from third party software like SAP, Siebel etc.
- Manual Creation
- Automatic Creation



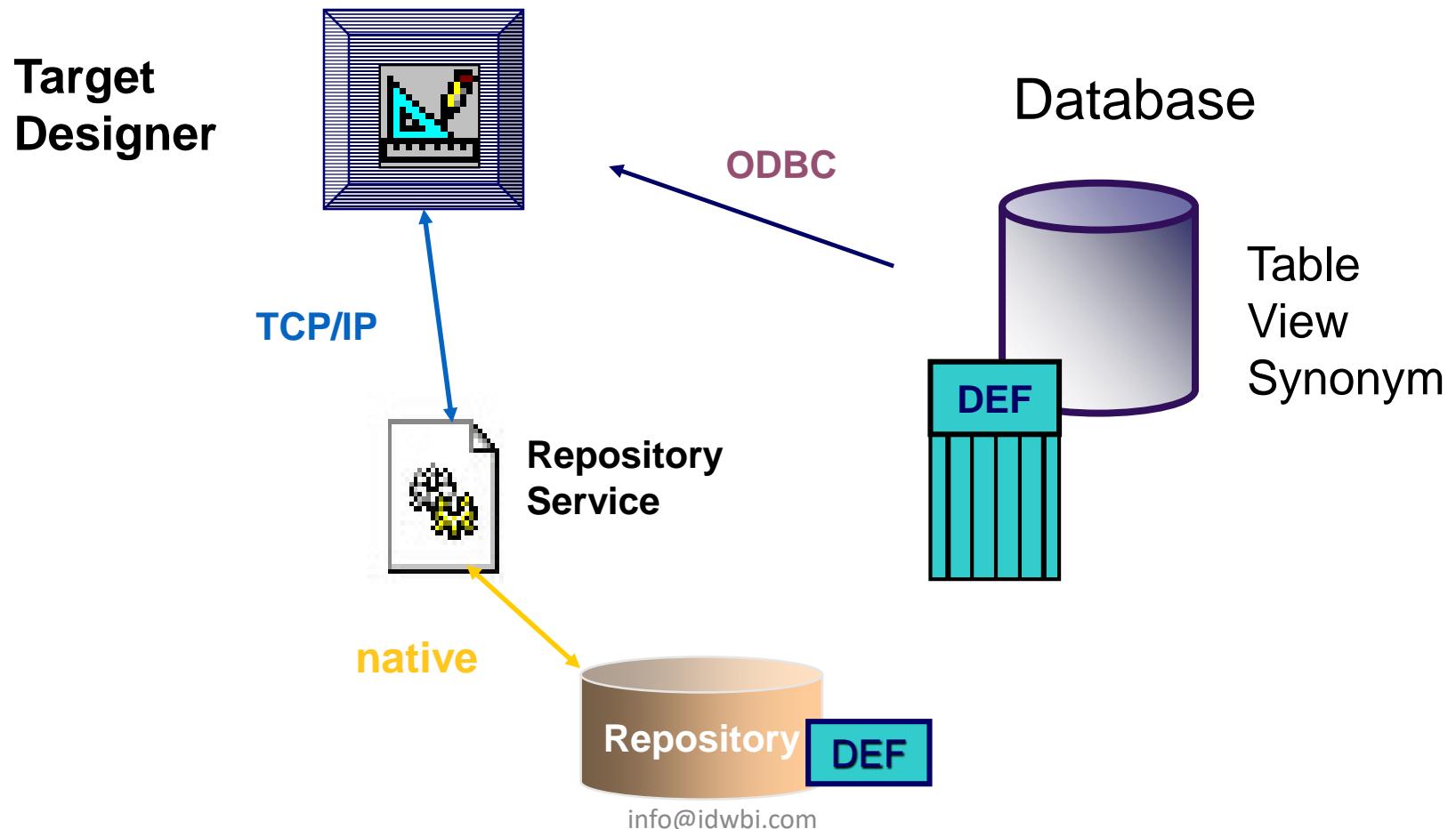
Automatic Target Creation

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

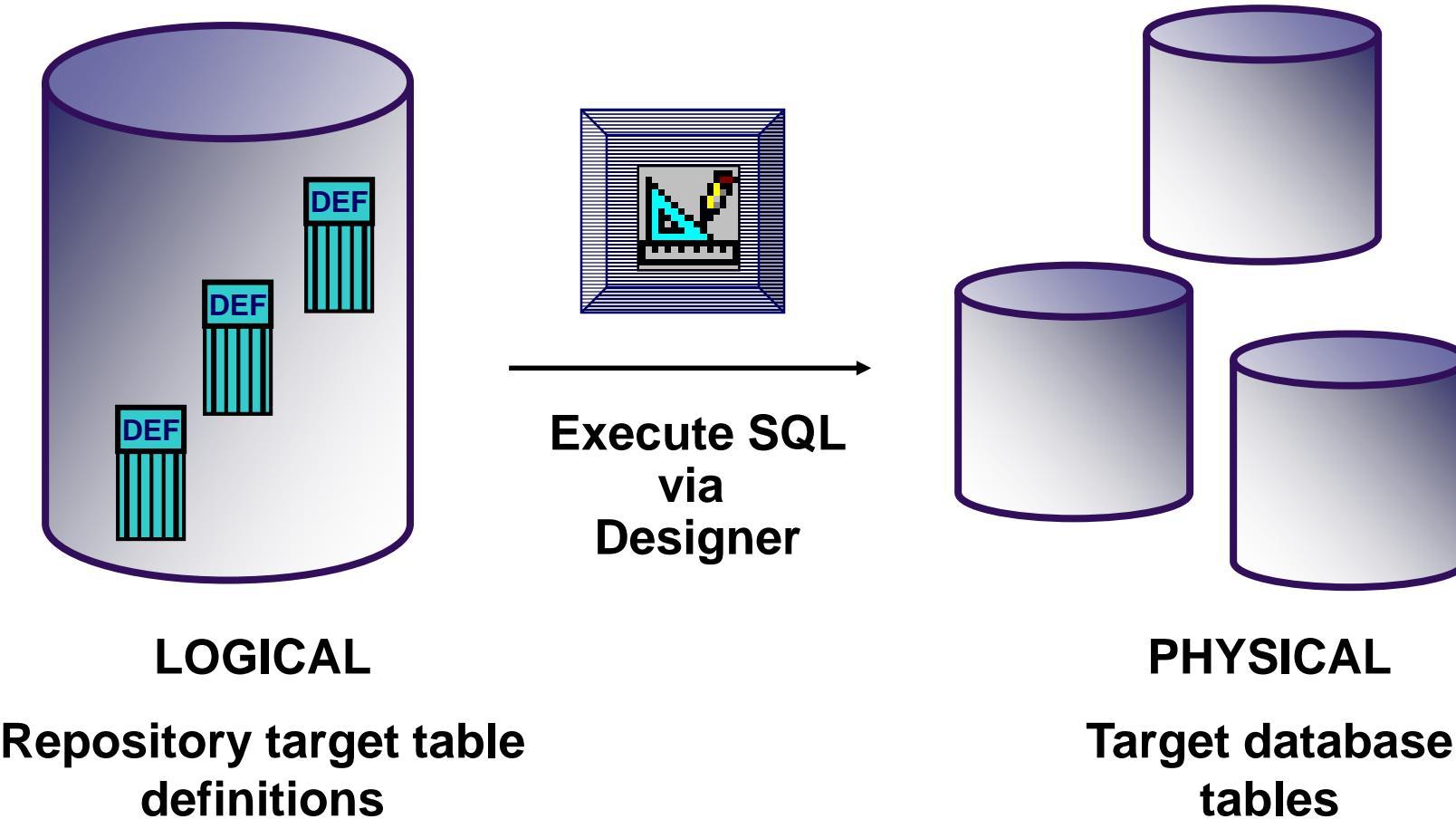


Import Definition from Database

Can “*Reverse engineer*” existing object definitions from a database system catalog or data dictionary



Creating Physical Tables





Transformation Concepts

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

- Transformation types
- Data Flow Rules
- Transformation Views
- PowerCenter Functions
- Expression Editor and Expression validation
- Port Types
- PowerCenter data types and Datatype Conversion
- Connection and Mapping Validation
- PowerCenter Basic Transformations – Source Qualifier, Filter, Joiner, Expression

Types of Transformations

- Active/Passive
 - Active : Changes the numbers of rows as data passes through it
 - Passive: Passes all the rows through it
- Connected/Unconnected
 - Connected : Connected to other transformation through connectors
 - Unconnected : Not connected to any transformation. Called within a transformation



Transformation Types

PowerCenter provides 24 objects for data transformation

- Aggregator: performs aggregate calculations
- Application Source Qualifier: reads Application object sources as ERP
- Custom: Calls a procedure in shared library or DLL
- Expression: performs row-level calculations
- External Procedure (TX): calls compiled code for each row
- Filter: drops rows conditionally
- Mapplet Input: Defines mapplet input rows. Available in Mapplet designer
- Java: Executes java code
- Joiner: joins heterogeneous sources
- Lookup: looks up values and passes them to other objects
- Normalizer: reads data from VSAM and normalized sources
- Mapplet Output: Defines mapplet output rows. Available in Mapplet designer

Transformation Types

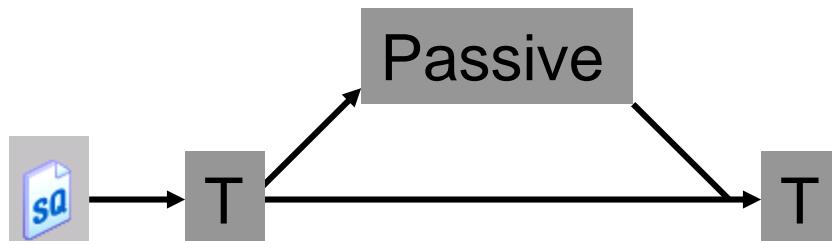
- Rank: limits records to the top or bottom of a range
- Router: splits rows conditionally
- Sequence Generator: generates unique ID values
- Sorter: sorts data
- Source Qualifier: reads data from Flat File and Relational Sources
- Stored Procedure: calls a database stored procedure
- Transaction Control: Defines Commit and Rollback transactions
- Union: Merges data from different databases
- Update Strategy: tags rows for insert, update, delete, reject
- XML Generator: Reads data from one or more Input ports and outputs XML through single output port
- XML Parser: Reads XML from one or more Input ports and outputs data through single output port
- XML Source Qualifier: reads XML data



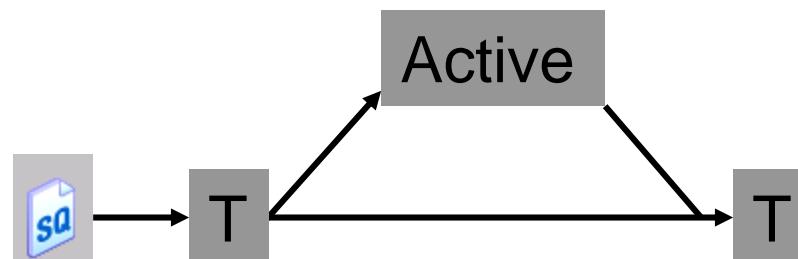
Data Flow Rules

- Each Source Qualifier starts a single data stream (a dataflow)
- Transformations can send rows to more than one transformation (split one data flow into multiple pipelines)
- Two or more data flows can meet together -- if (and only if) they originate from a common active transformation
 - Cannot add an active transformation into the mix

ALLOWED



DISALLOWED



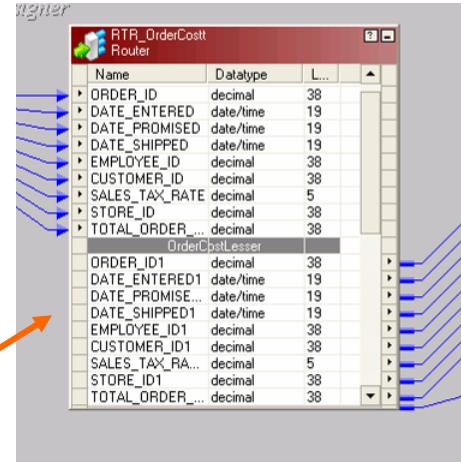
Example holds true with Normalizer in lieu of Source Qualifier. Exceptions are:
Maplet Input and Joiner transformations

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 and properties; allows editing



Port Name	Datatype	Prec	Scale	IN	OUT
1 ORDER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2 DATE_ENTERED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 DATE_PROMISED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 DATE_SHIPPED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 EMPLOYEE_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6 CUSTOMER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7 SALES_TAX_RATE	decimal	5	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8 STORE_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9 TOTAL_ORDER_COST	decimal	38	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>



Edit Mode

Allows users with folder “write” permissions to change or create transformation ports and properties

Define port level handling

Define transformation level properties

Enter comments

Make reusable

Switch between transformations

	Port Name	Datatype	Prec	Scale	I	O
1		INPUT				
2	ORDER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	DATE_ENTERED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	DATE_PROMISED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	DATE_SHIPPED	date/time	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	EMPLOYEE_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	CUSTOMER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	SALES_TAX_RATE	decimal	5	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	STORE_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	TOTAL_ORDER_COST	decimal	38	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Default value:

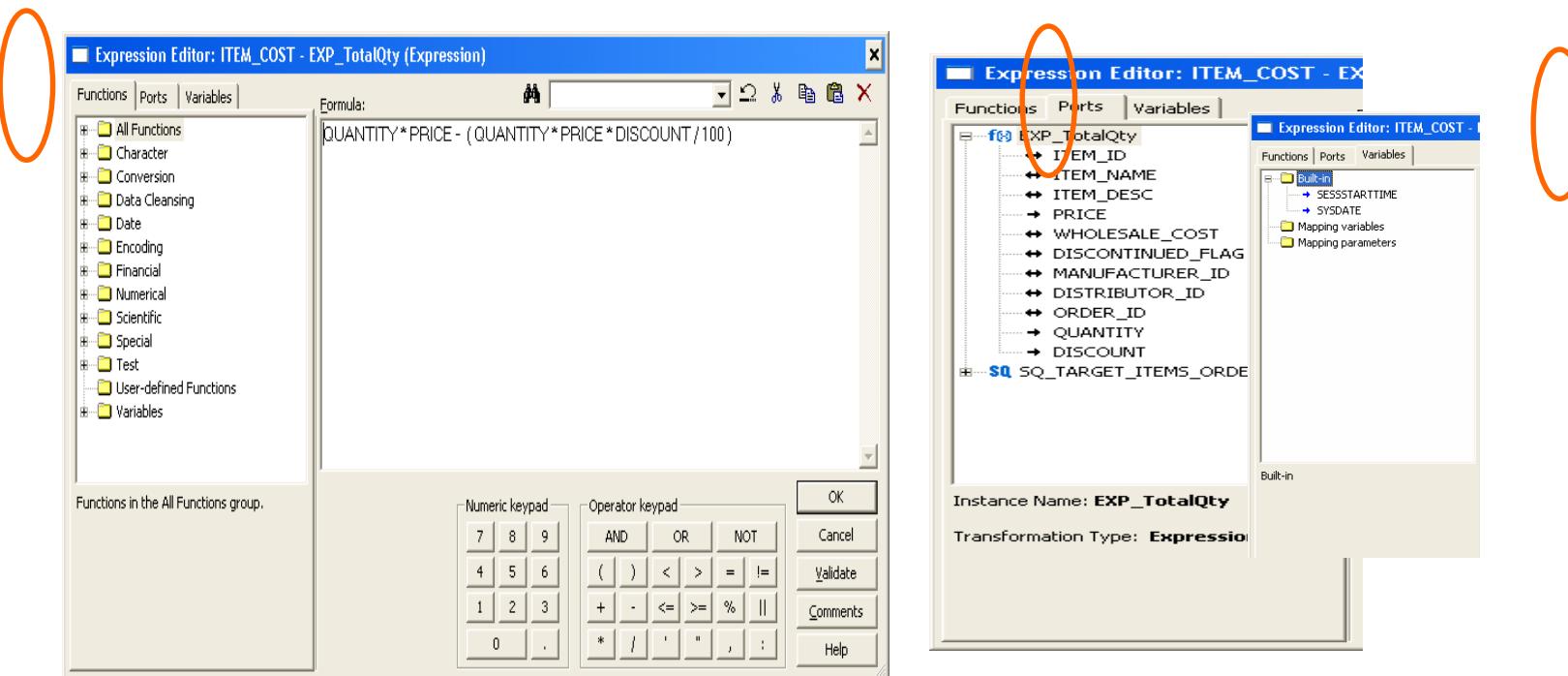
Description:

OK Cancel Apply Help

Expression Editor



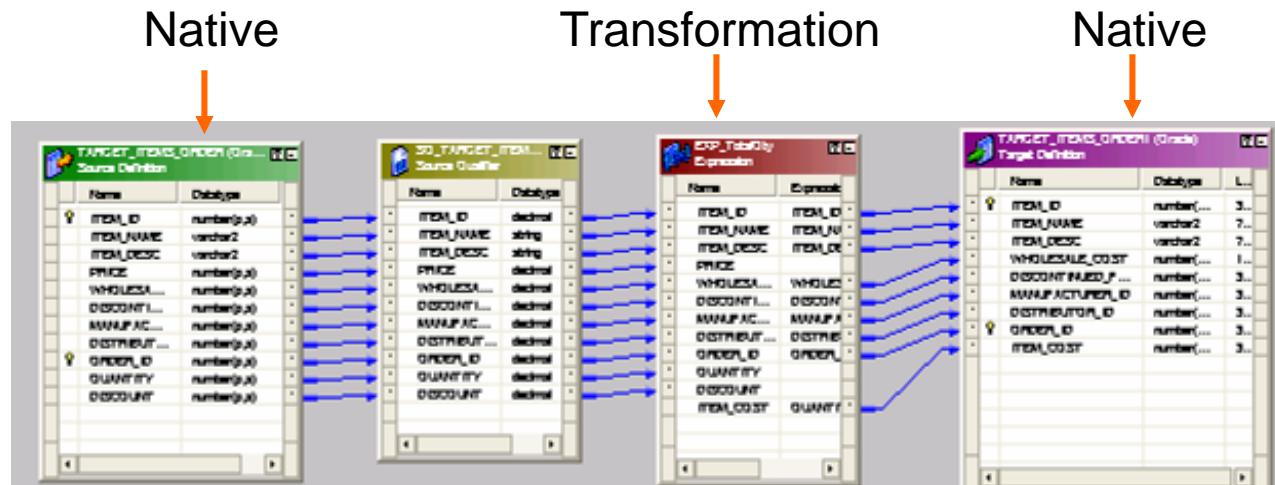
- An expression formula is a calculation or conditional statement
- Used in Expression, Aggregator, Rank, Filter, Router, Update Strategy
- Performs calculation based on ports, functions, operators, variables, literals, constants and return values from other transformations



PowerCenter Data Types



NATIVE DATATYPES	TRANSFORMATION DATATYPES
Specific to the source and target database types	PowerCenter internal datatypes based on ANSI SQL-92
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)



Datatype Conversions

	Integer, Small Int	Decimal	Double, Real	String , Text	Date/ Time	Binary
Integer, Small Integer	x	x	x	x		
Decimal	x	x	x	x		
Double , Real	x	x	x	x		
String , Text	x	x	x	x	x	
Date/Time				x	x	
Binary						x

- All numeric data can be converted to all other numeric datatypes, e.g. - integer, double, and decimal
- All numeric data can be converted to string, and vice versa
- Date can be converted only to date and string, and vice versa
- Raw (binary) can only be linked to raw
- Other conversions not listed above are not supported
- These conversions are implicit; no function is necessary



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

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

For backwards compatibility only - use || instead



- TO_CHAR (numeric)
- TO_DATE
- TO_DECIMAL
- TO_FLOAT
- TO_INTEGER
- TO_NUMBER

- ADD_TO_DATE
- DATE_COMPARE
- DATE_DIFF
- GET_DATE_PART
- LAST_DAY
- ROUND (date)
- SET_DATE_PART
- TO_CHAR (date)
- TRUNC (date)

Conversion Functions

- Used to convert datatypes

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

PowerCenter Functions



ERROR
ABORT
DECODE
IIF

Special Functions

- Used to handle specific conditions within a session; search for certain values; test conditional statements
- IIF(Condition,True,False)**

ISNULL
IS_DATE
IS_NUMBER
IS_SPACES

Test Functions

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

SOUNDEX
METAPHONE

Encoding Functions

- Used to encode string values



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 transformation attributes
 - e.g. - filter condition, lookup condition, SQL Query
- Parse default values
- Check spelling, correct number of arguments in functions, other syntactical errors



Types of Ports

- Four basic types of ports are there
 - Input
 - Output
 - Input/Output
 - Variable
- Apart from these Look-up & Return ports are also there that are specific to the Lookup transformation



Variable and Output 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	I	O	V	Expression
DATE_ENTERED	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MONTH	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	GET_DATE_PART(DATE_ENTERED, 'MM')
QUANTITY	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
PRICE1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DISCOUNT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Q1_SALES	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(QUANTITY * PRICE1 - DISCOUNT, MONTH = 1 OR MONTH = 2 OR MONTH = 3)
Q2_SALES	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(QUANTITY * PRICE1 - DISCOUNT, MONTH = 4 OR MONTH = 5 OR MONTH = 6)...
Q3_SALES	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(QUANTITY * PRICE1 - DISCOUNT, MONTH = 7 OR MONTH = 8 OR MONTH = 9)...
Q4_SALES	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(QUANTITY * PRICE1 - DISCOUNT, MONTH = 10 OR MONTH = 11 OR MONTH = ...)



Connection Validation

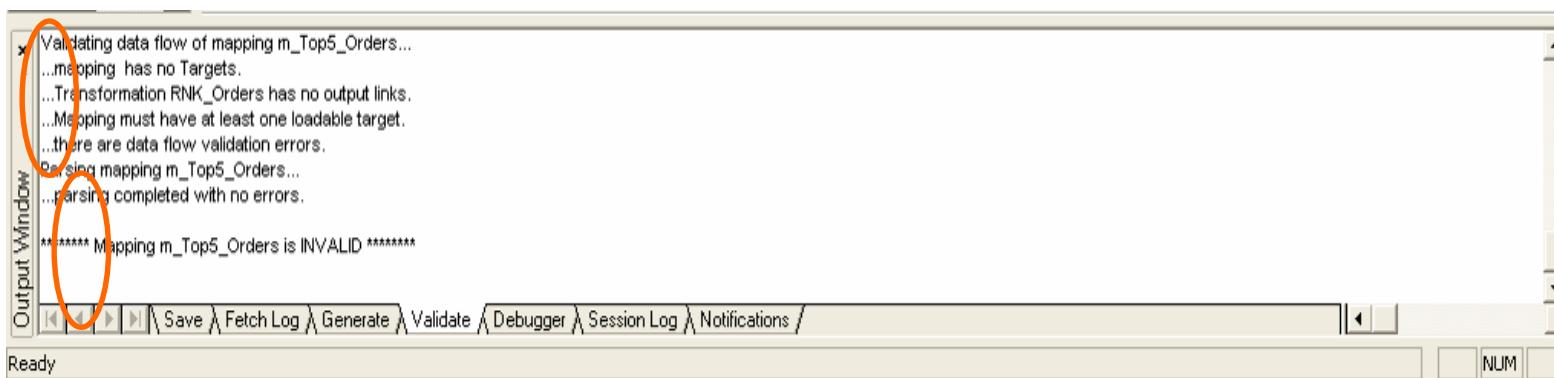
Examples of invalid connections in a Mapping:

- 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
- Connecting more than one active transformation to another transformation (invalid dataflow)



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 will always display reason for invalidity



Source Qualifier Transformation



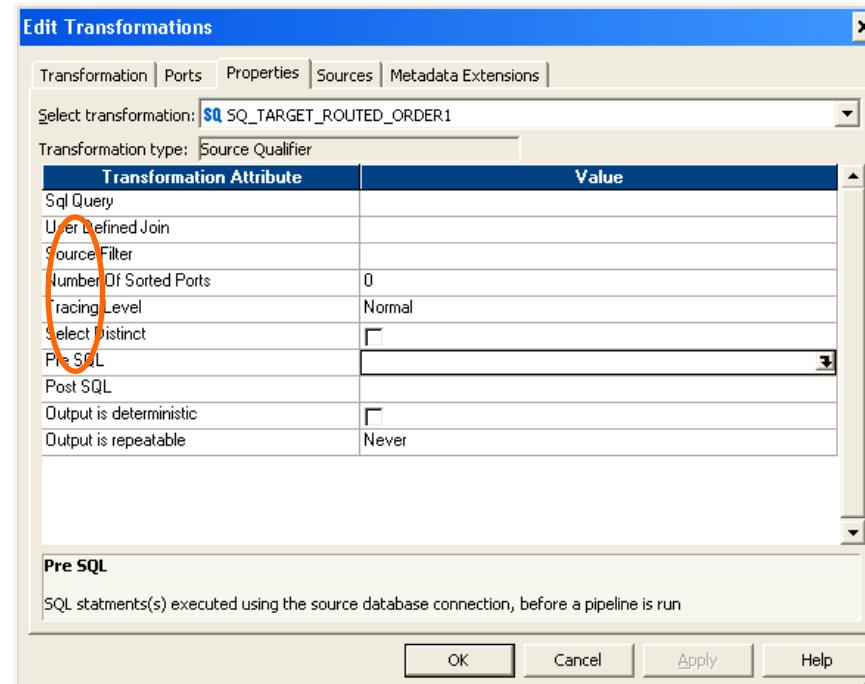
- Reads data from the sources
- Active & Connected Transformation
- Applicable only to relational and flat file sources
- Maps database/file specific datatypes to PowerCenter Native datatypes.
 - Eg. Number(24) becomes decimal(24)
- Determines how the source database binds data when the Integration Service reads it
- If mismatch between the source definition and source qualifier datatypes then mapping is invalid
- All ports by default are Input/Output ports

Source Qualifier Transformation



Used as

- Joiner for homogenous tables using a where clause
- Filter using a where clause
- Sorter
- Select distinct values



Pre-SQL and Post-SQL Rules



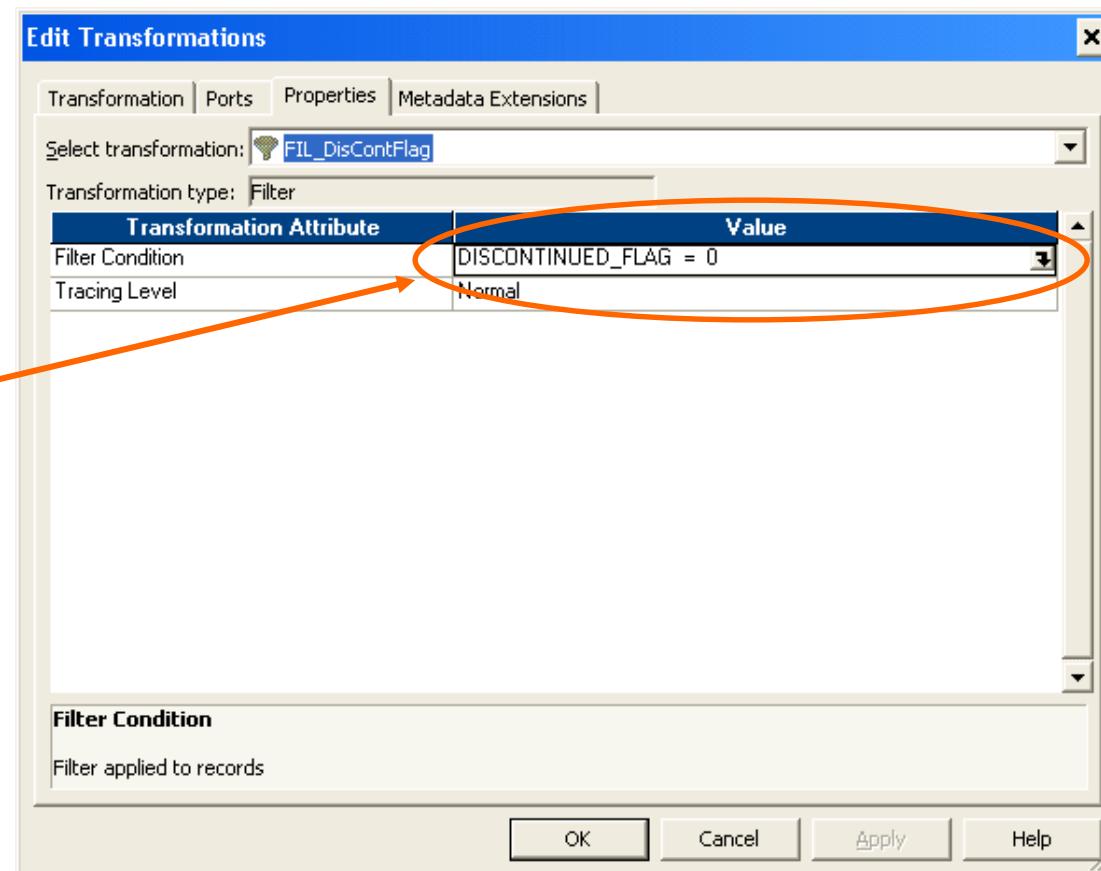
- Can use any command that is valid for the database type; no nested comments
- Can use Mapping Parameters and Variables in SQL executed against the source
- Use a semi-colon (;) to separate multiple statements
- Informatica Server ignores semi-colons within single quotes, double quotes or within /* ... */
- To use a semi-colon outside of quotes or comments, 'escape' it with a back slash (\)
- Workflow Manager does not validate the SQL

Filter Transformation



Drops rows conditionally

- Active Transformation
- Connected
- Ports
 - All input / output
- Specify a Filter condition
- Usage
 - Filter rows from flat file sources
 - Single pass source(s) into multiple targets

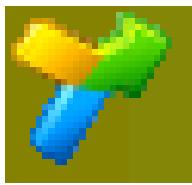


Filter Transformation – Tips



- Boolean condition is always faster as compared to complex conditions
- Use filter transformation early in the mapping
- Source qualifier filters rows from relational sources but filter transformation is source independent
- Always validate a condition

Joiner Transformation



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

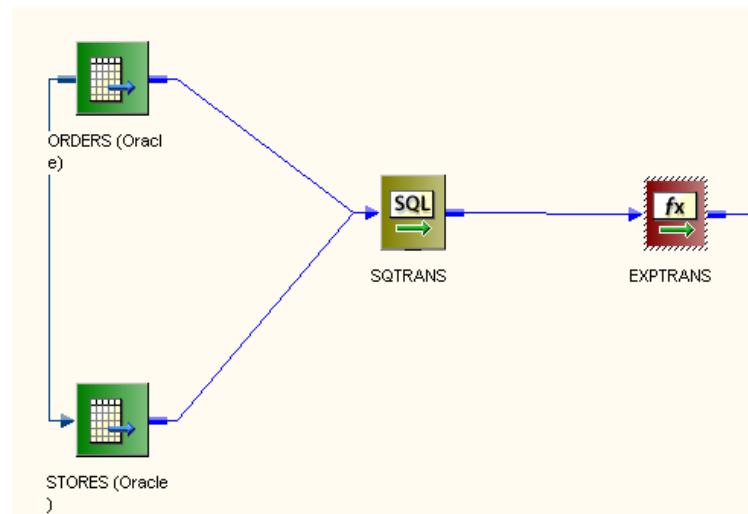
- When to use a Joiner Transformation
- Homogeneous Joins
- Heterogeneous Joins
- Joiner properties
- Joiner Conditions
- Nested joins

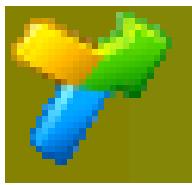


Homogeneous Joins

Joins that can be performed with a SQL SELECT statement:

- Source Qualifier contains a SQL join
- Tables on same database server (or are synonyms)
- Database server does the join “work”
- Multiple homogenous tables can be joined

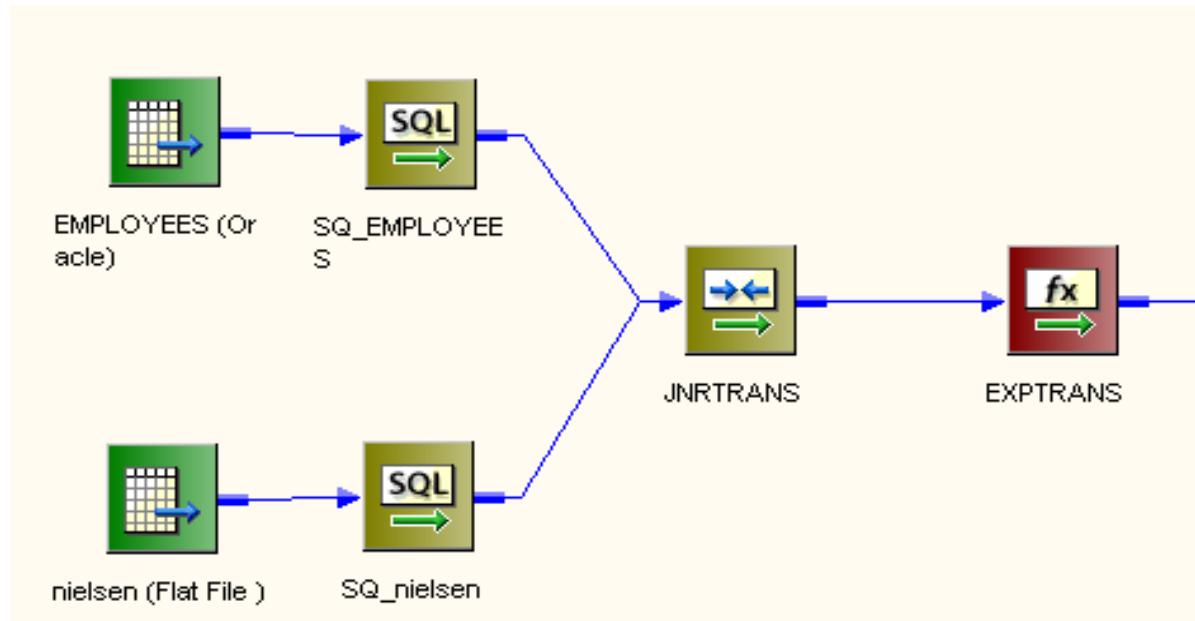




Heterogeneous Joins

Joins that cannot be done with a SQL statement:

- An Oracle table and a Sybase table
- Two Informix tables on different database servers
- Two flat files
- A flat file and a database table

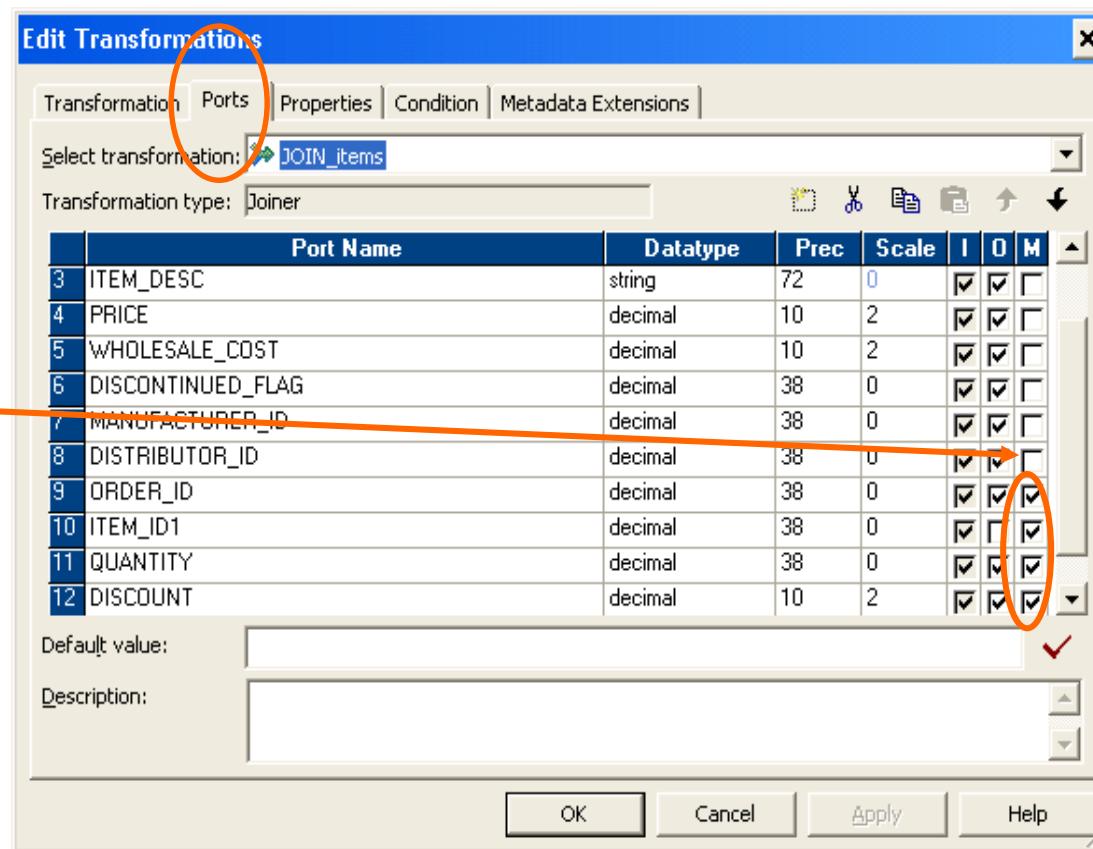


Joiner Transformation



Performs heterogeneous joins on records from two tables on same or different databases or flat file sources

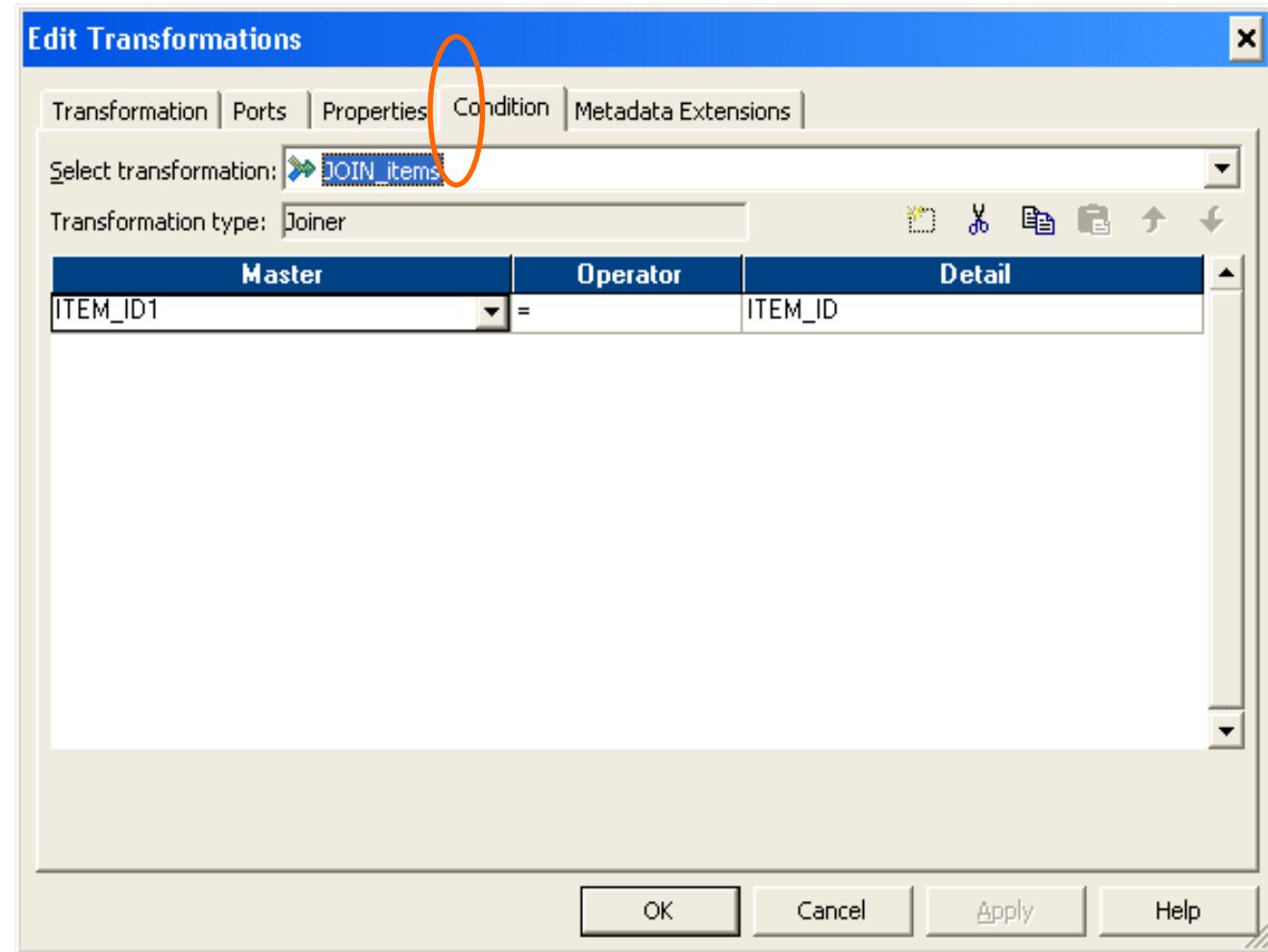
- Active Transformation
- Connected
- Ports
 - All input or input / output
 - “M” denotes port comes from master source
- Specify the Join condition
- Usage
 - Join two flat files
 - Join two tables from different databases
 - Join a flat file with a relational table



Joiner Conditions



Multiple
join
conditions
are supported





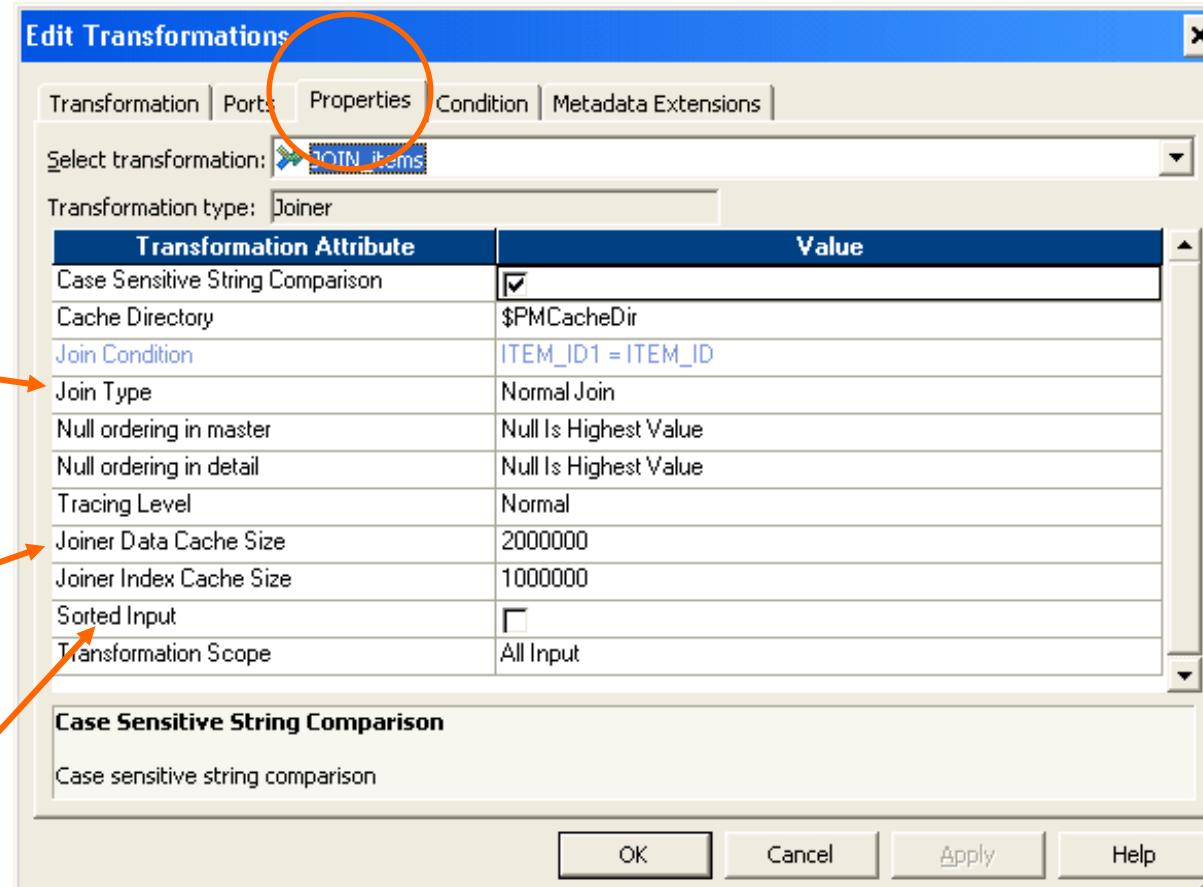
Joiner Properties

Join types:

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

Set Joiner Cache

Joiner can accept sorted data



Sorted Input for Joiner



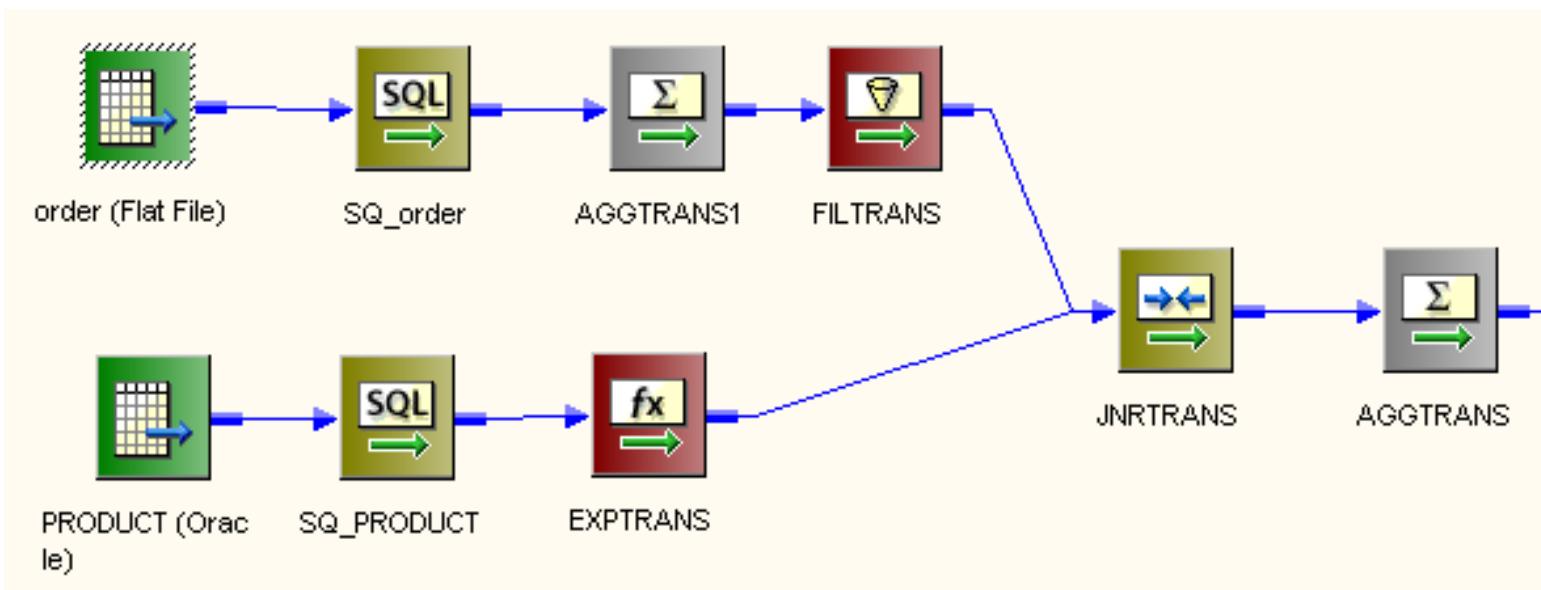
- Using sorted input improves session performance minimizing the disk input and output
- The pre-requisites for using the sorted input are
 - Database sort order must be same as the session sort order
 - Sort order must be configured by the use of sorted sources (flat files/relational tables) or sorter transformation
 - The flow of sorted data must be maintained by avoiding the use of transformations like Rank, Custom, Normalizer etc. which alter the sort order
 - Enable the sorted input option is properties tab
 - The order of the ports used in joining condition must match the order of the ports at the sort origin
 - When joining the Joiner output with another pipeline make sure that the data from the first joiner is sorted



Mid-Mapping Join - Tips

The Joiner does not accept input in the following situations:

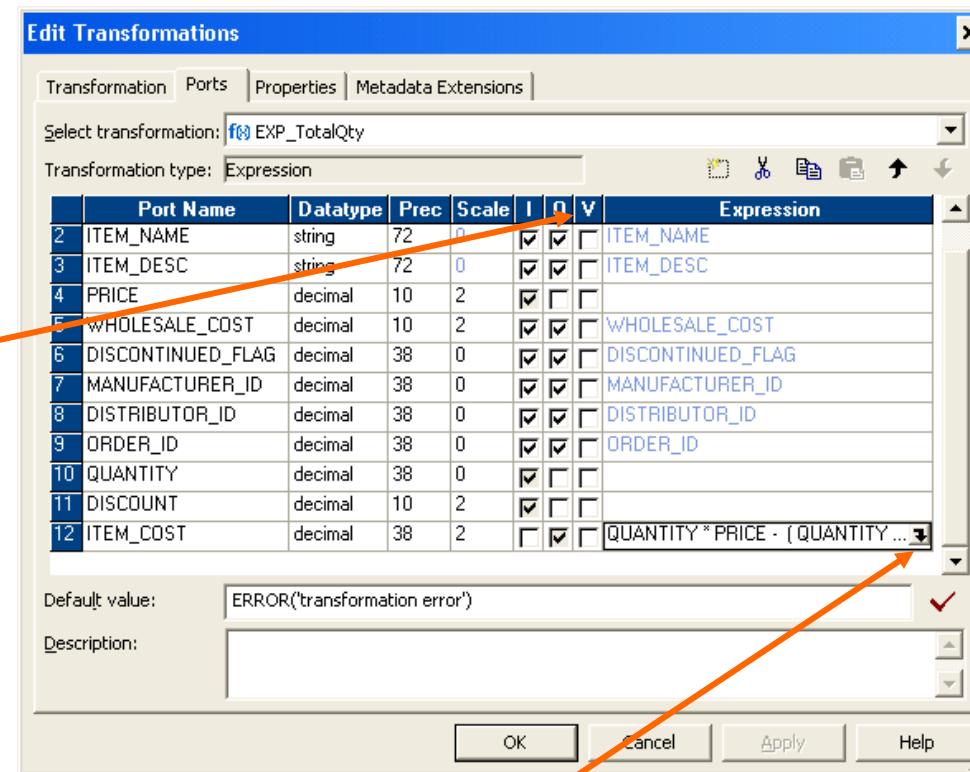
- Both input pipelines begin with the same Source Qualifier
- Both input pipelines begin with the same Normalizer
- Both input pipelines begin with the same Joiner
- Either input pipeline contains an Update Strategy



Expression Transformation



- Passive Transformation
- Connected
- Ports
 - Mixed
 - Variables allowed
- Create expression in an output or variable port
- Usage
 - Perform majority of data manipulation



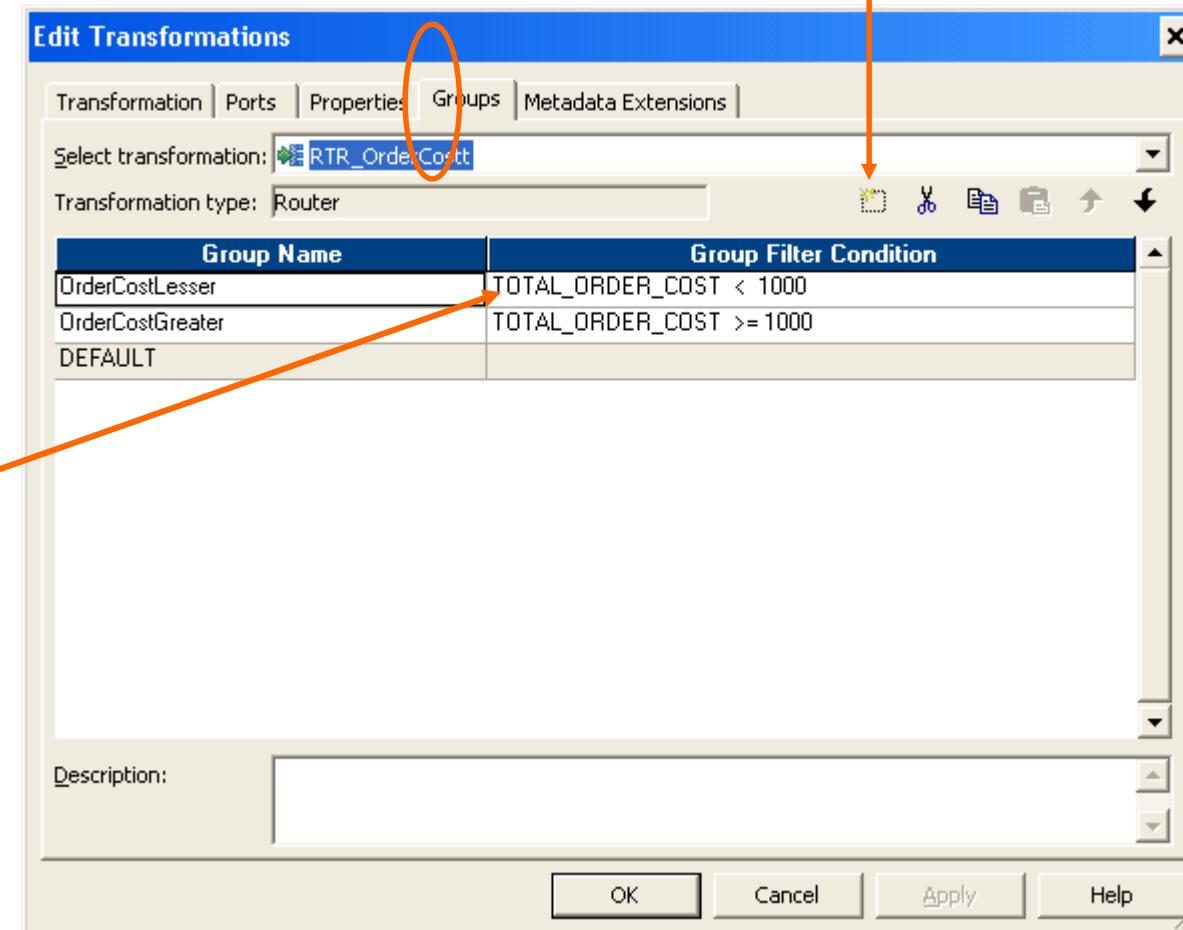
Click here to invoke the
Expression Editor

Router Transformation



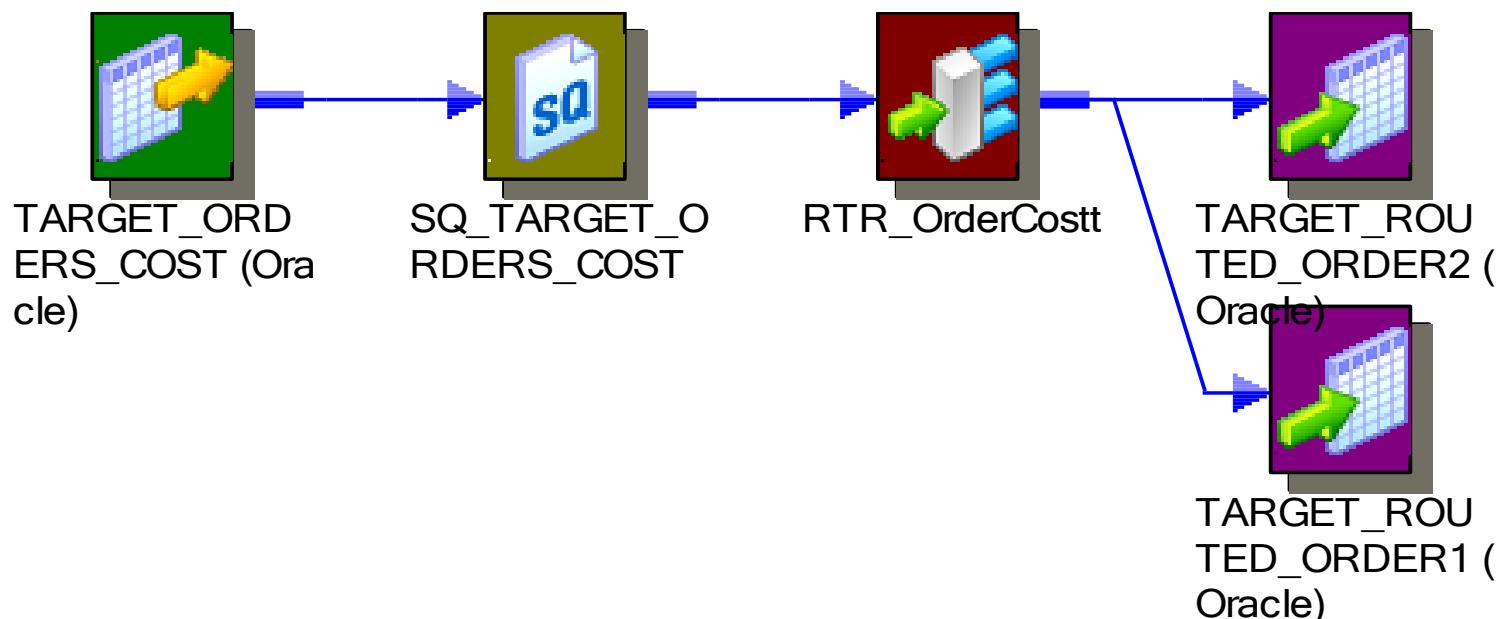
Multiple filters in single transformation

- Active Transformation
- Connected
- Ports
 - All input/output
- Specify filter conditions for each Group
- Usage
 - Link source data in one pass to multiple filter conditions





Router Transformation in a Mapping



Comparison – Filter and Router



Filter	Router
Tests rows for only one condition	Tests rows for one or more condition
Drops the rows which don't meet the filter condition	Routes the rows not meeting the filter condition to default group

- In case of multiple filter transformation the Integration service processes rows for each transformation but in case of router the incoming rows are processed only once.

Sorter Transformation



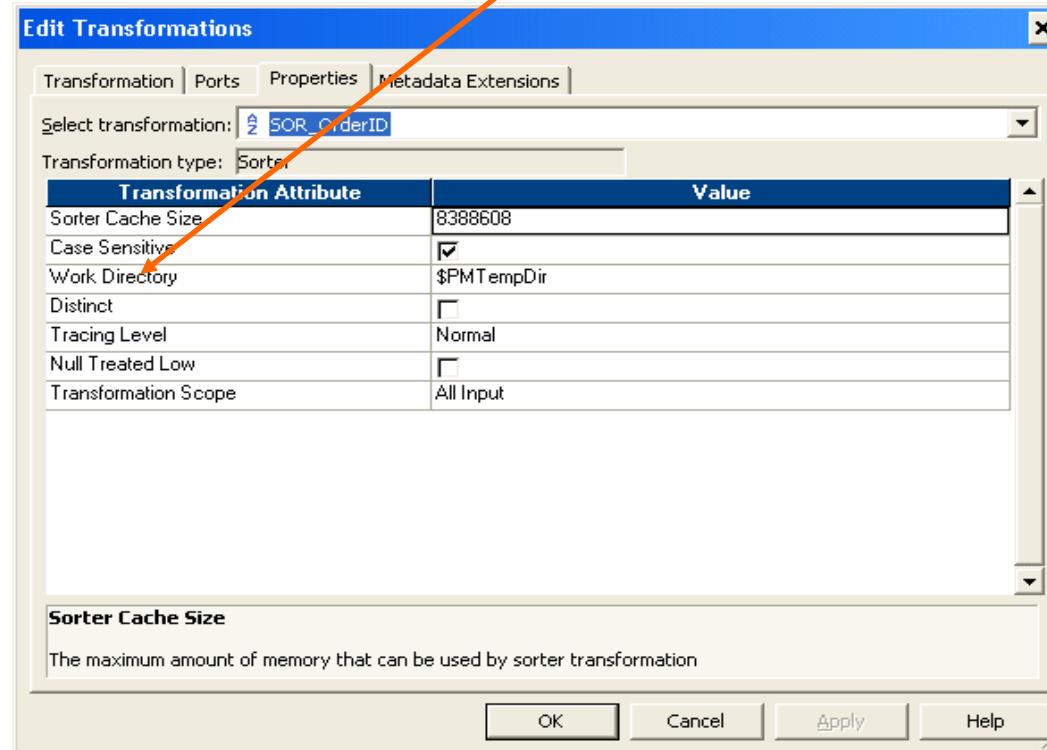
Sorts the data, selects distinct

- Active transformation
- Is always connected
- Can sort data from relational tables or flat files both in ascending or descending order
- Only Input/Output/Key ports are there
- Sort takes place on the Integration Service machine
- Multiple sort keys are supported. The Integration Service sorts each port sequentially
- The Sorter transformation is often more efficient than a sort performed on a database with an ORDER BY clause

Sorter Transformation



- Discard duplicate rows by selecting 'Distinct' option



- Acts as an active transformation with distinct option else as passive

Aggregator Transformation



Performs aggregate calculations

- Active Transformation
- Connected
- Ports
 - Mixed
 - Variables allowed
 - Group By allowed
- Create expressions in output or variable ports
- Usage
 - Standard aggregations

Edit Transformations

Transformation Ports Properties Metadata Extensions

Select transformation: AGG_OrderID

Transformation type: Aggregator

	Port Name	Datatype	Prec	Scale	I	O	V	Expression	GroupBy
2	ITEM_NAME	string	72	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ITEM_NAME	<input type="checkbox"/>
3	ITEM_DESC	string	72	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ITEM_DESC	<input type="checkbox"/>
4	WHOLESALE_COST	decimal	10	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	WHOLESALE_COST	<input type="checkbox"/>
5	DISCONTINUED_FLAG	decimal	38	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DISCONTINUED_FLAG	<input type="checkbox"/>
6	MANUFACTURER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANUFACTURER_ID	<input type="checkbox"/>
7	DISTRIBUTOR_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DISTRIBUTOR_ID	<input type="checkbox"/>
8	ORDER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ORDER_ID	<input checked="" type="checkbox"/>
9	ITEM_COST	decimal	38	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
10	TOTAL_ORDER_COST	decimal	38	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(ITEM_COST)	<input type="checkbox"/>

Default value:

Description:

OK Cancel Apply Help

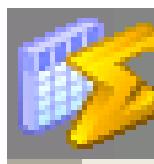


PowerCenter Aggregate Functions

Aggregate Functions

AVG
COUNT
FIRST
LAST
MAX
MEDIAN
MIN
PERCENTILE
STDDEV
SUM
VARIANCE

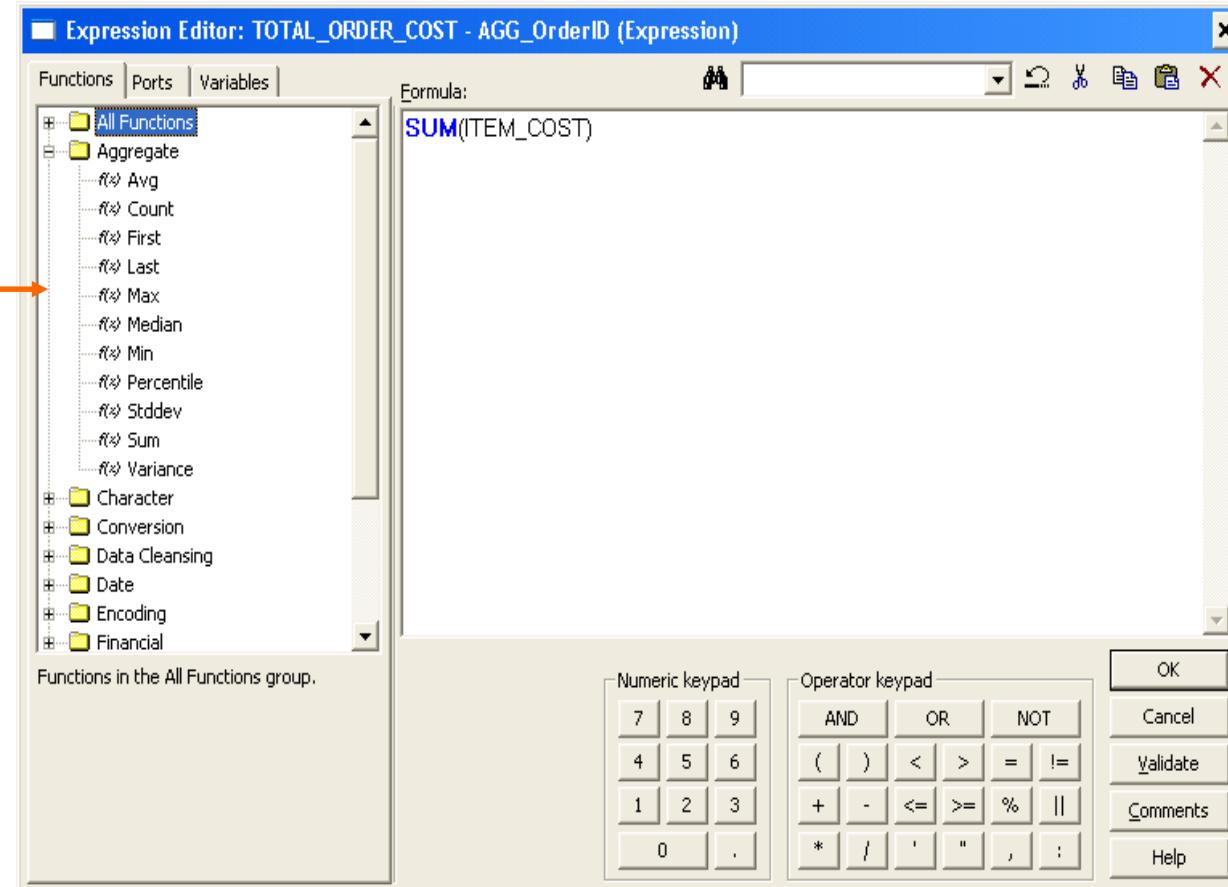
- Return summary values for non-null data in selected ports
- Used only in Aggregator transformations
- Used in output ports only
- Calculate a single value (and row) for all records in a group
- Only one aggregate function can be nested within an aggregate function
- Conditional statements can be used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



Aggregate Expressions

Aggregate
functions are
supported
only in the
Aggregator
Transformation

Conditional
Aggregate
expressions are
supported



Conditional SUM format: $\text{SUM}(\text{value}, \text{condition})$

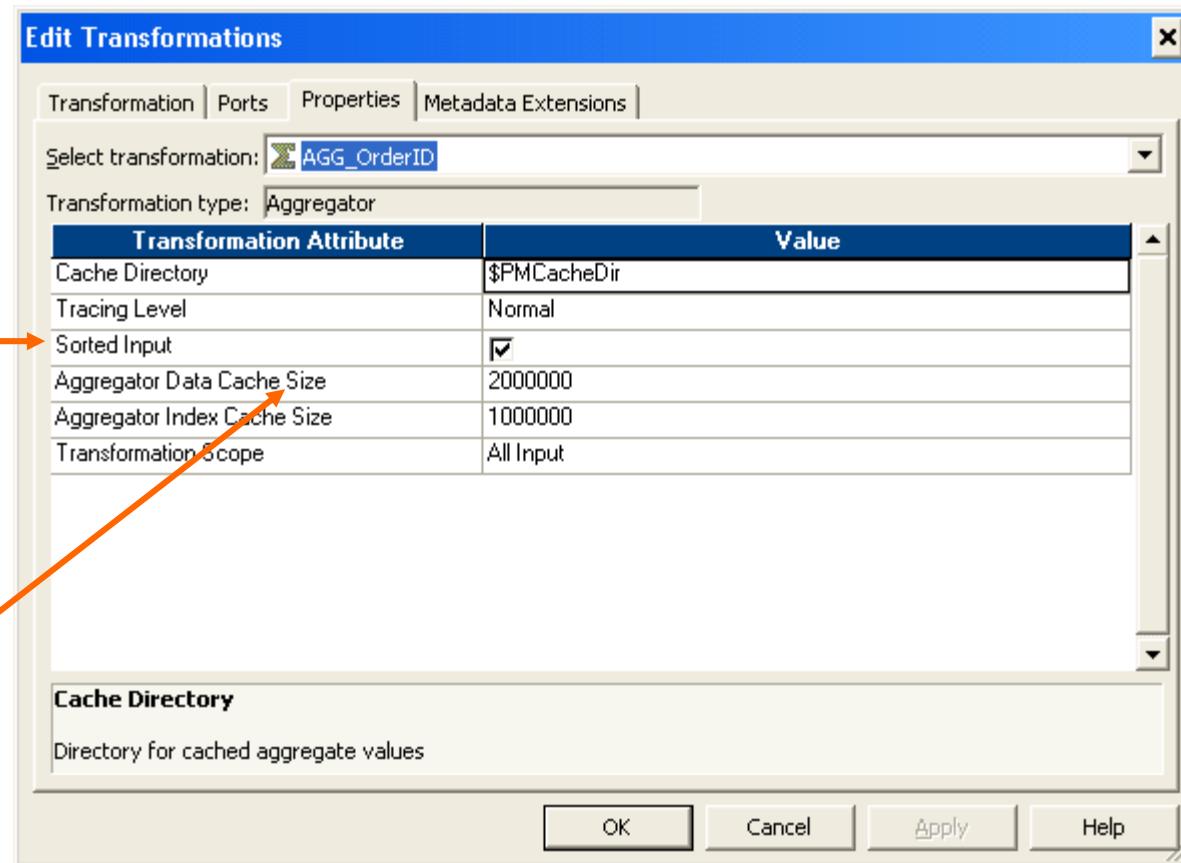


Aggregator Properties

Sorted Input Property

Instructs the Aggregator to expect the data to be sorted

Set Aggregator cache sizes (on Integration Service machine)





Why Sorted Input?

- Aggregator works efficiently with sorted input data
 - Sorted data can be aggregated more efficiently, decreasing total processing time
- The Integration Service 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 “Group By” ports
- Performance gain will depend upon varying factors

Lookup Transformation



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

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

Lookup Transformation



- Get Related Value
- Get Multiple Values
- Perform Calculation
- Update Slowly Changing Dimension tables

Lookup Transformation

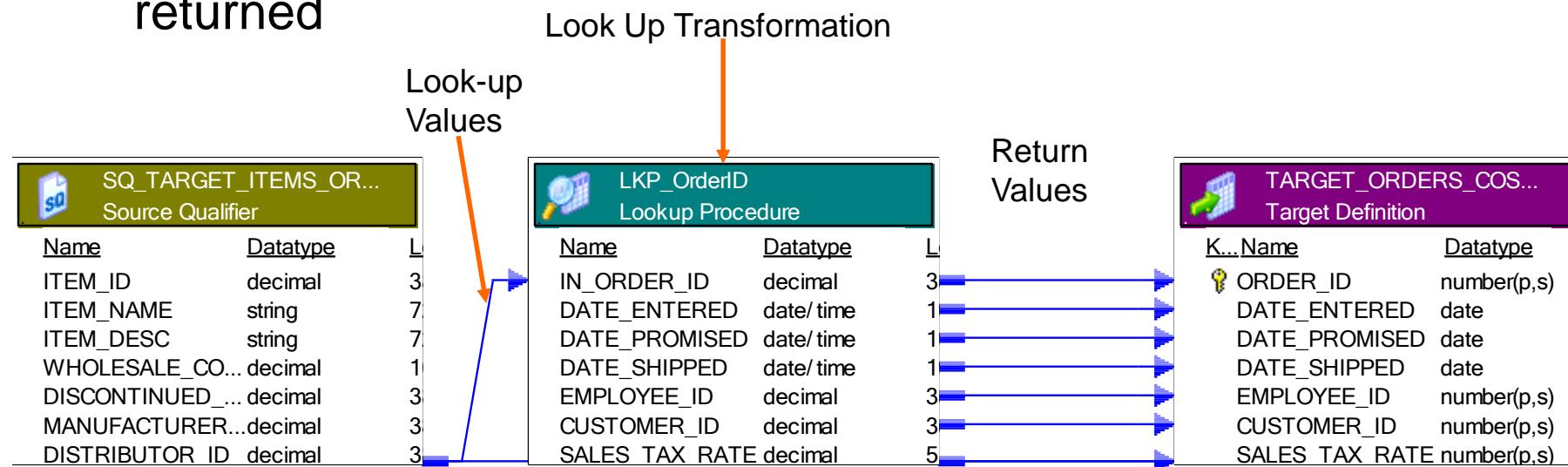


- Different types of configuration possible :-
 - Flat File or Relational
 - Pipeline Lookup
 - Connected or Unconnected lookup
 - Cached or Un-cached Lookup



How a Lookup Transformation Works

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



Lookup Transformation



Looks up values in a database table or flat files and provides data to downstream transformation in a Mapping

- Passive Transformation
- Connected / Unconnected
- Ports
 - Mixed
 - “L” denotes Lookup port
 - “R” denotes port used as a return value (unconnected Lookup only)
- Specify the Lookup Condition
- Usage
 - Get related values
 - Verify if records exists or if data has changed

Port Name	Datatype	Prec	S...	I	O	L	R	Associated Port	I..	I..
1 ORDER_ID	decimal	38	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2 IN_ORDER_ID	decimal	38	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
3 DATE_ENTERED	date/time	19	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
4 DATE_PROMISED	date/time	19	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
5 DATE_SHIPPED	date/time	19	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
6 EMPLOYEE_ID	decimal	38	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
7 CUSTOMER_ID	decimal	38	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
8 SALES_TAX_RATE	decimal	5	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>
9 STORE_ID	decimal	38	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	- N/A -	<input type="checkbox"/>	<input type="checkbox"/>

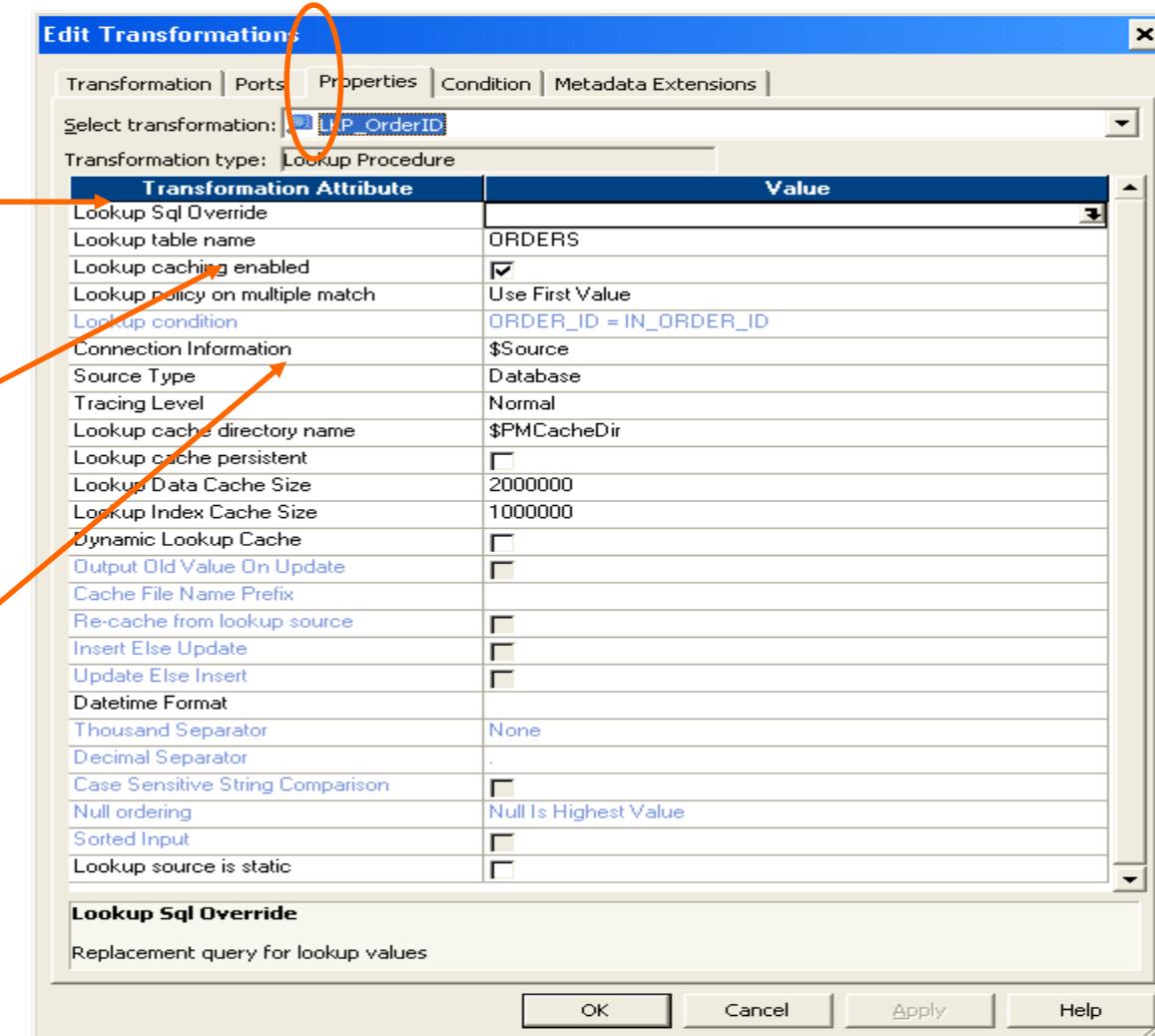


Lookup Properties

Override
Lookup SQL
option

Toggle
caching

Native
Database
Connection
Object name



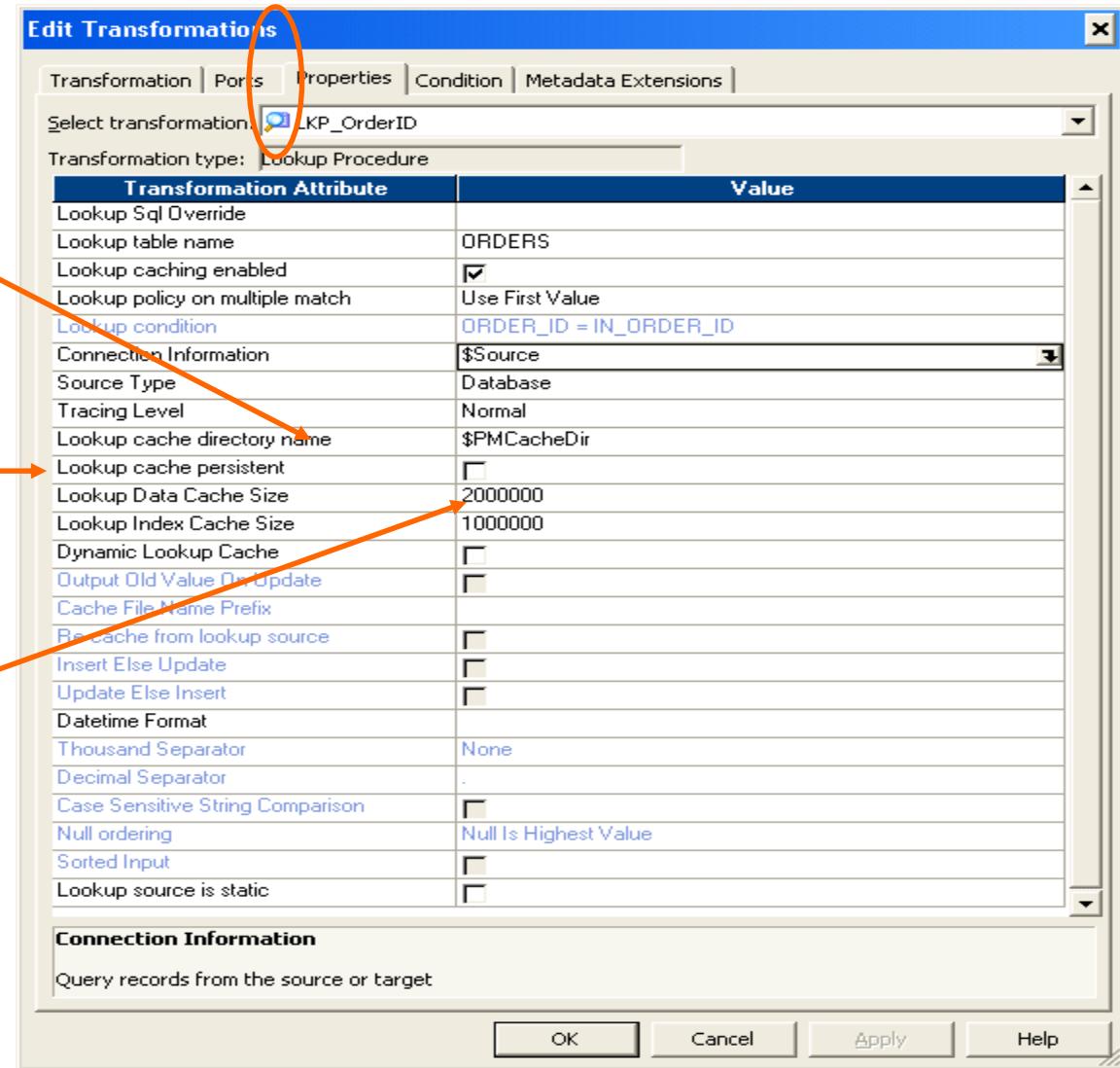


Additional Lookup Properties

Set cache directory

Make cache persistent

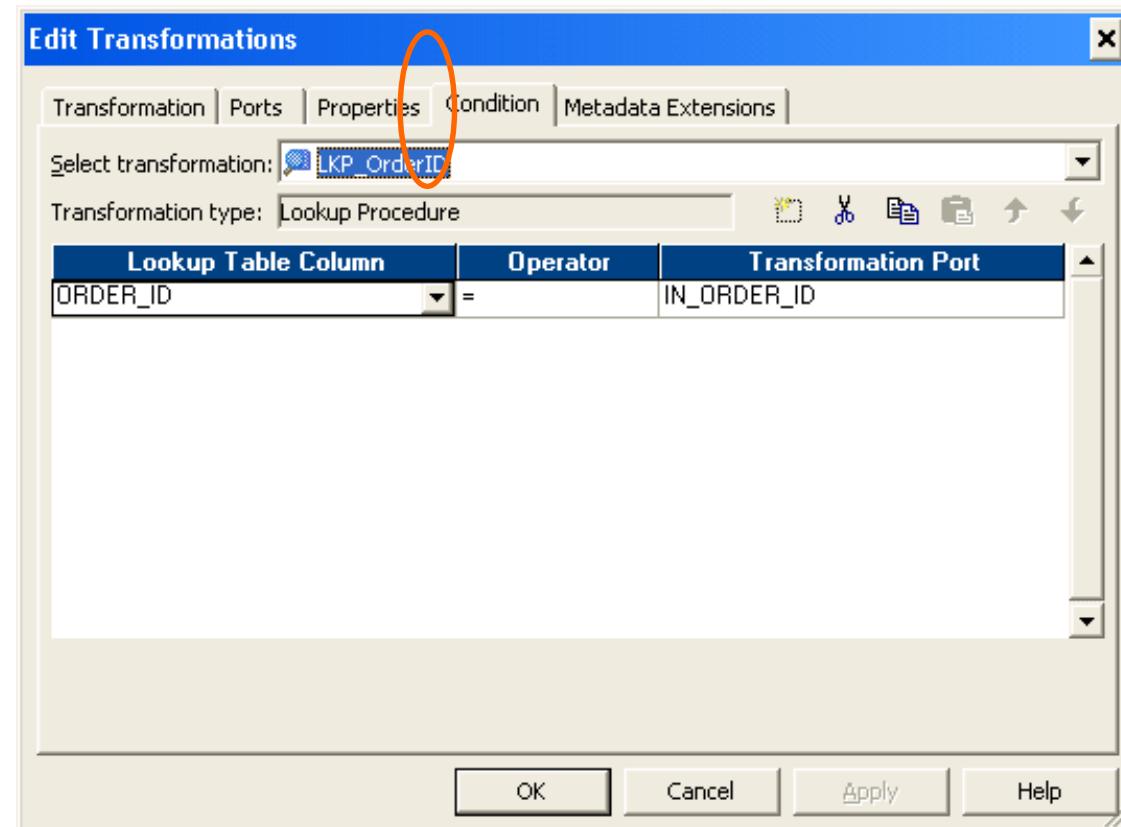
Set Lookup cache sizes



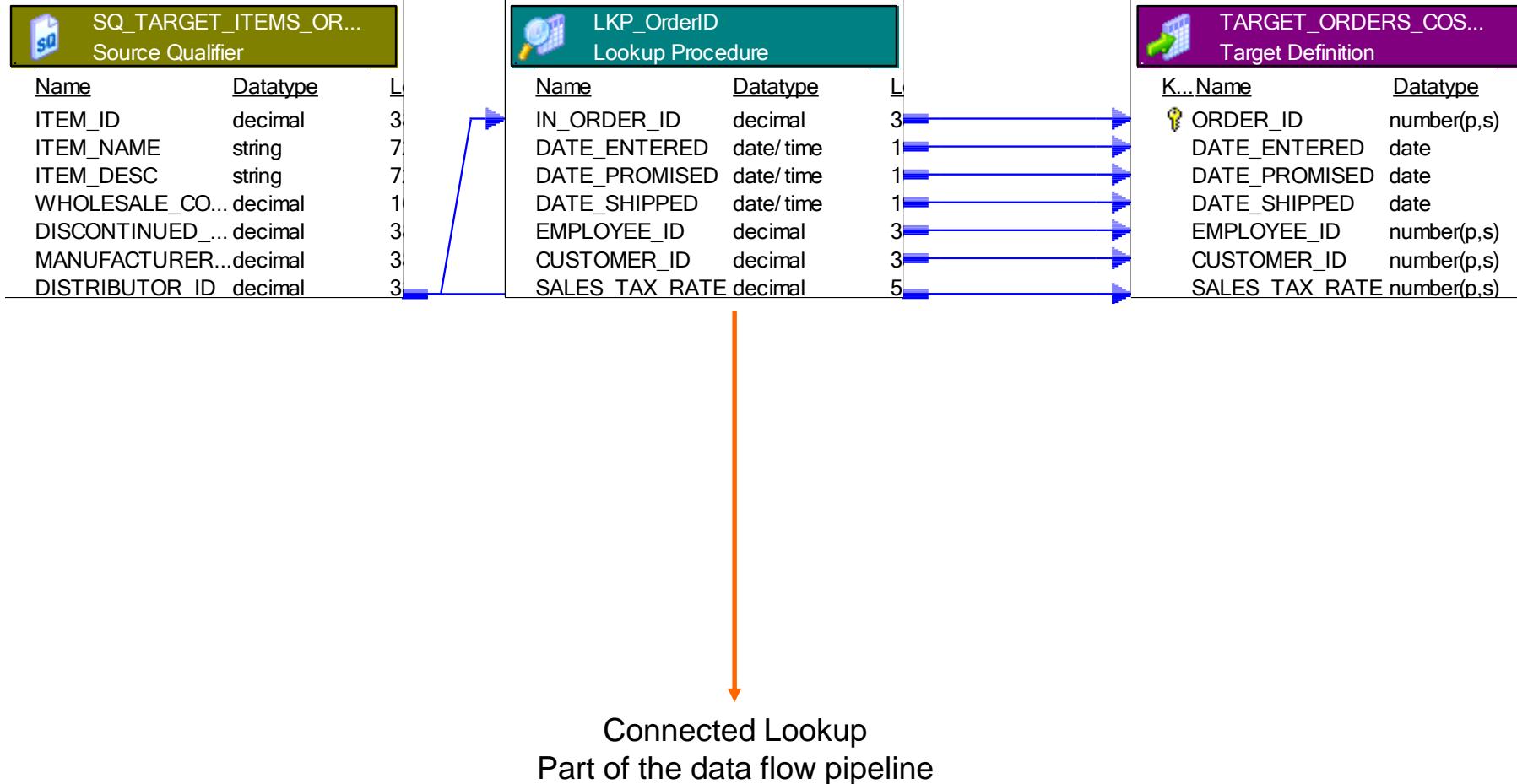


Lookup Conditions

Multiple conditions are supported



Connected Lookup

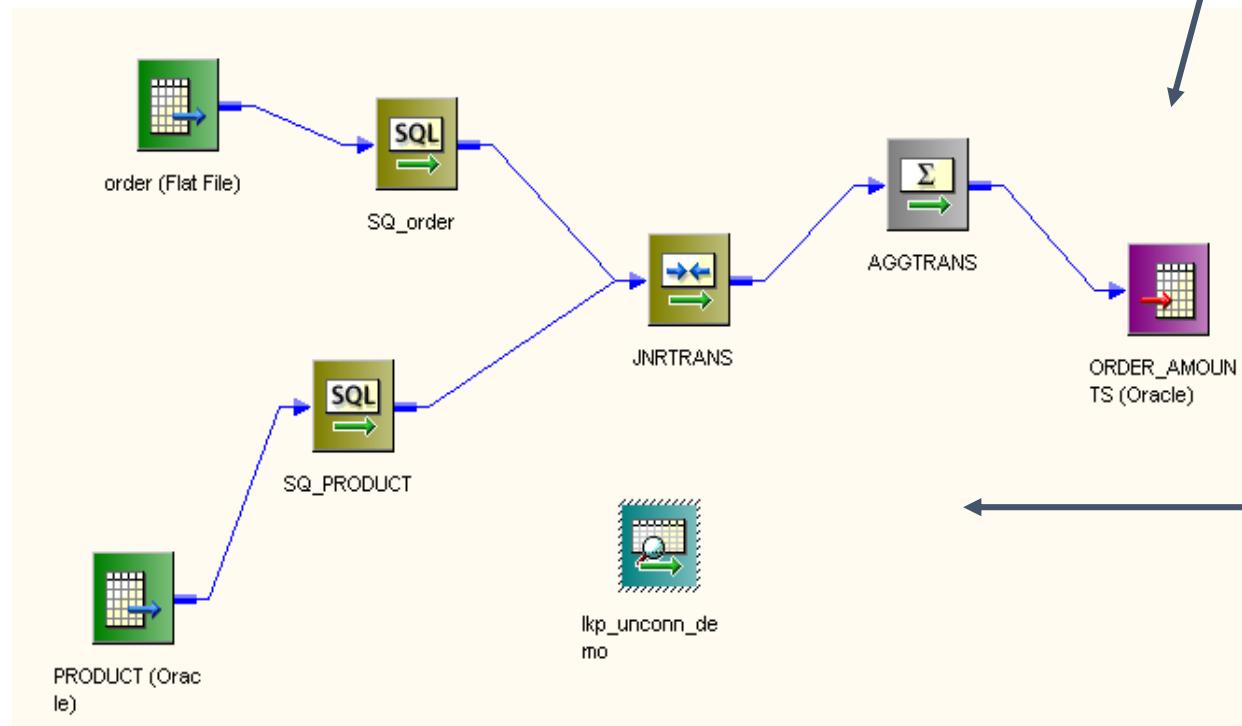




Unconnected Lookup

- Will be physically “unconnected” from other transformations
 - There can be NO data flow arrows leading to or from an unconnected Lookup

Lookup function can be set within any transformation that supports expressions



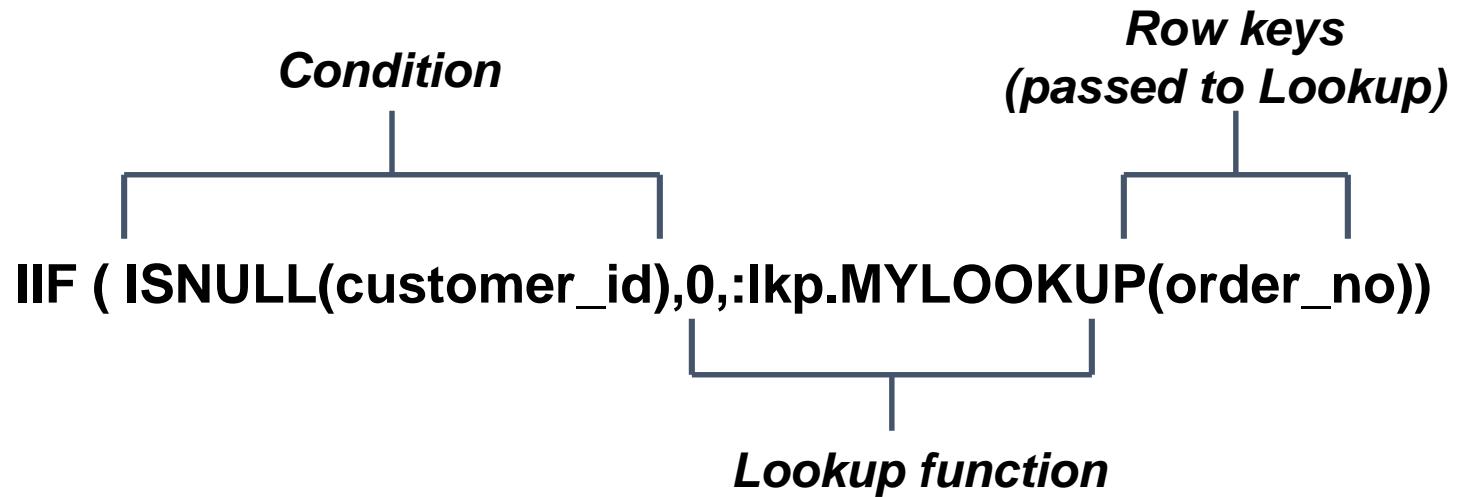
Lookup data is called from the point in the Mapping that needs it

Conditional Lookup Technique



Two requirements:

- Must be Unconnected (or “function mode”) Lookup
- Lookup function used within a conditional statement



- Conditional statement is evaluated for each row
- Lookup function is called only under the pre-defined condition

Conditional 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), 0,:lkp.MYLOOKUP (sku_numb))



Condition

(true for 2 percent of all rows)

Lookup

(called only when condition is true)

Net savings = 490,000 lookups



Unconnected Lookup - Return Port

- The port designated as 'R' is the return port for the unconnected lookup
- There can be only one return port
- The look-up (L) / Output (O) port can be assigned as the Return (R) port
- The Unconnected Lookup can be called in any other transformation's expression editor using the expression
`:LKP.Lookup_Transformation(argument1, argument2,..)`



Connected vs. 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

To Cache or not to Cache?



Caching can significantly impact performance

- Cached
 - Lookup table data is cached locally on the machine
 - 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 lookup or large cache memory is available for Integration Service

Additional Lookup Cache Options



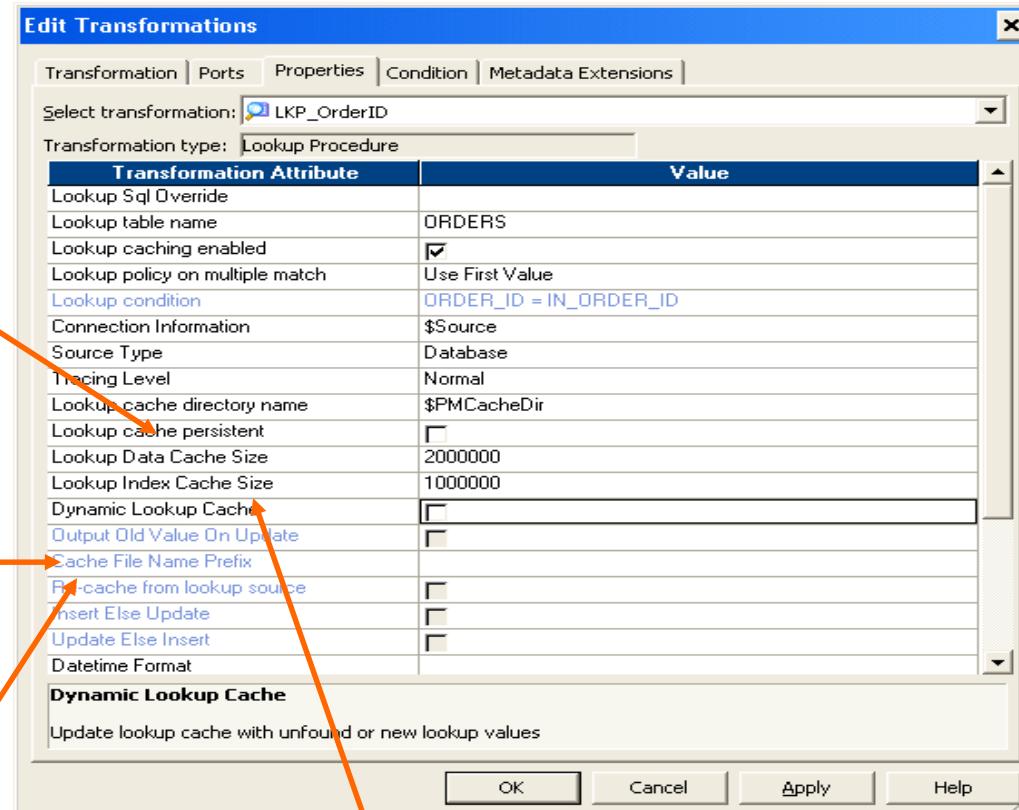
Make cache persistent

Cache File Name Prefix

- Reuse cache by name for another similar business purpose

Recache from Source

- Overrides other settings and Lookup data is refreshed



Dynamic Lookup Cache

- Allows a row to know about the handling of a previous row

Persistent Caches



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

Update Strategy Transformation

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

- Update Strategy functionality
- Update Strategy expressions
- Refresh strategies
- Smart aggregation

Target Refresh Strategies

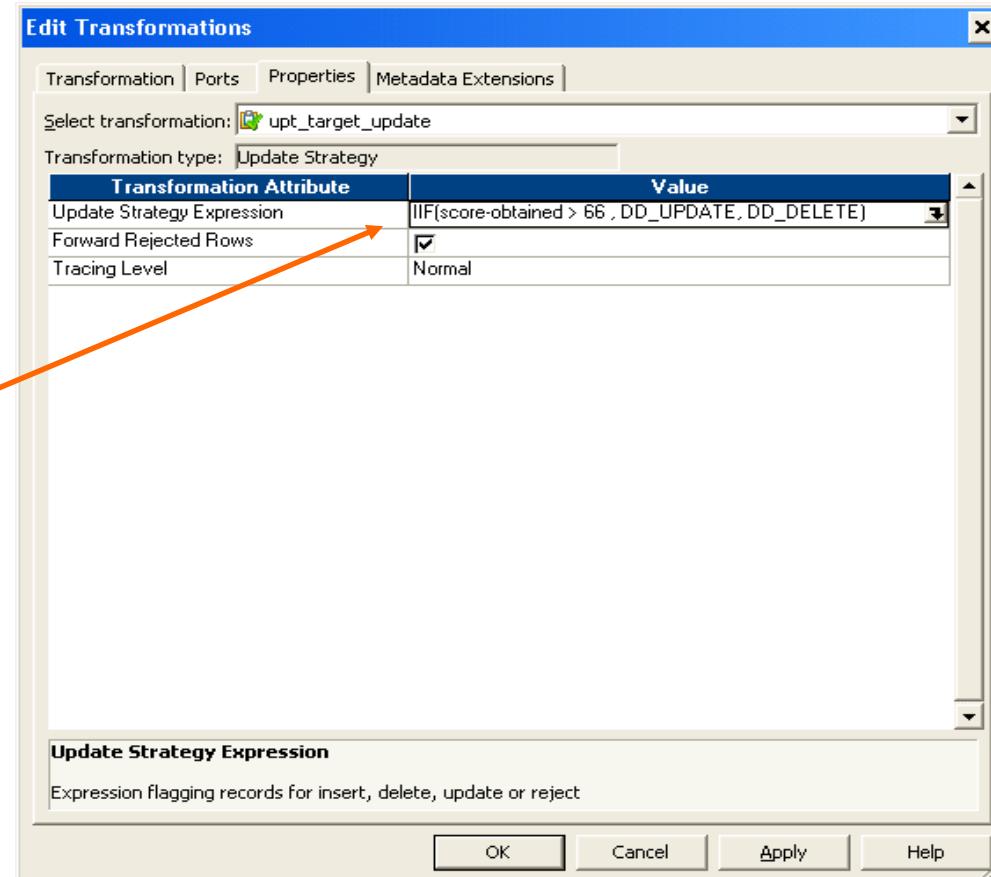
- Single snapshot: Target truncated, new records inserted
- Sequential snapshot: new records inserted
- Incremental: Only new records are inserted. Records already present in the target are ignored
- Incremental with Update: Only new records are inserted. Records already present in the target are updated

Update Strategy Transformation

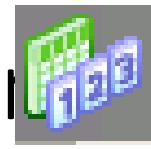


Used to specify how each individual row will be used to update target tables (insert, update, delete, reject)

- Active Transformation
- Connected
- Ports
 - All input / output
- Specify the Update Strategy Expression
- Usage
 - Updating Slowly Changing Dimensions
 - IIF or DECODE logic determines how to handle the record

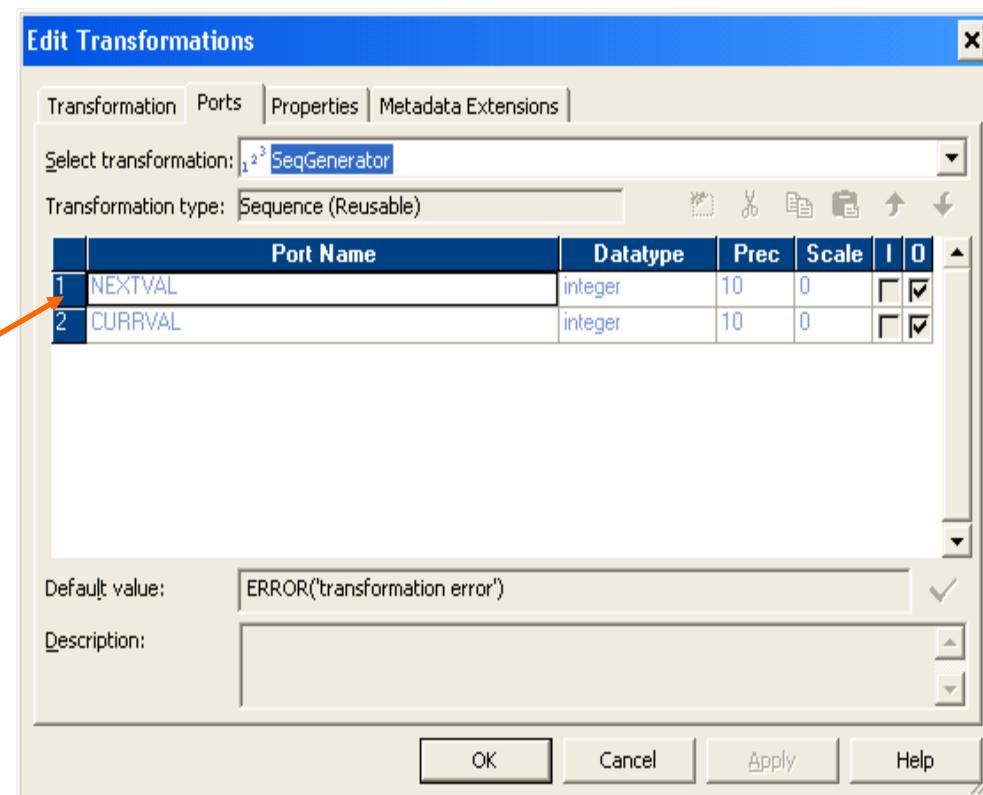


Sequence Generator Transformation

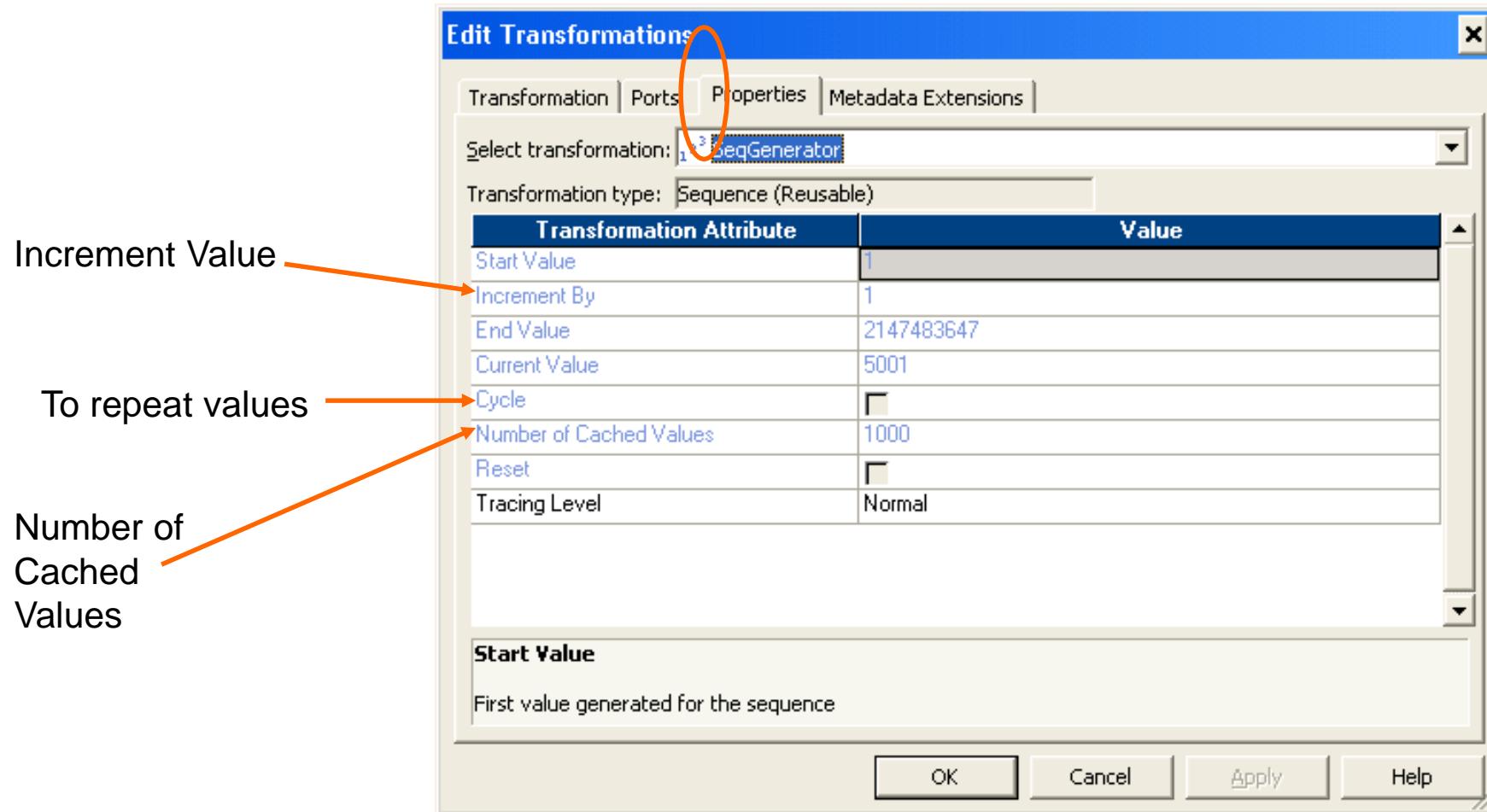


Generates unique keys for any port on a row

- Passive Transformation
- Connected
- Ports
 - Two predefined output ports,
 - NEXTVAL
 - CURRVAL
 - No input ports allowed
- Usage
 - Generate sequence numbers
 - Shareable across mappings



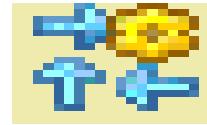
Sequence Generator Properties



This section will include -

- Integration Service Concepts
- The Workflow Manager GUI interface
- Setting up Server Connections
 - Relational
 - FTP
 - External Loader
 - Application
- Task Developer
- Creating and configuring Tasks
- Creating and Configuring Wokflows
- Workflow Schedules

Integration Service



- Application service that runs data integration sessions and workflows
- To access it one must have permissions on the service in the domain
- Is managed through Administrator Console
- A repository must be assigned to it
- A code page must be assigned to the Integration Service process which should be compatible with the repository service code page

Workflow Manager Interface



**Task
Tool Bar**

Workflow Designer

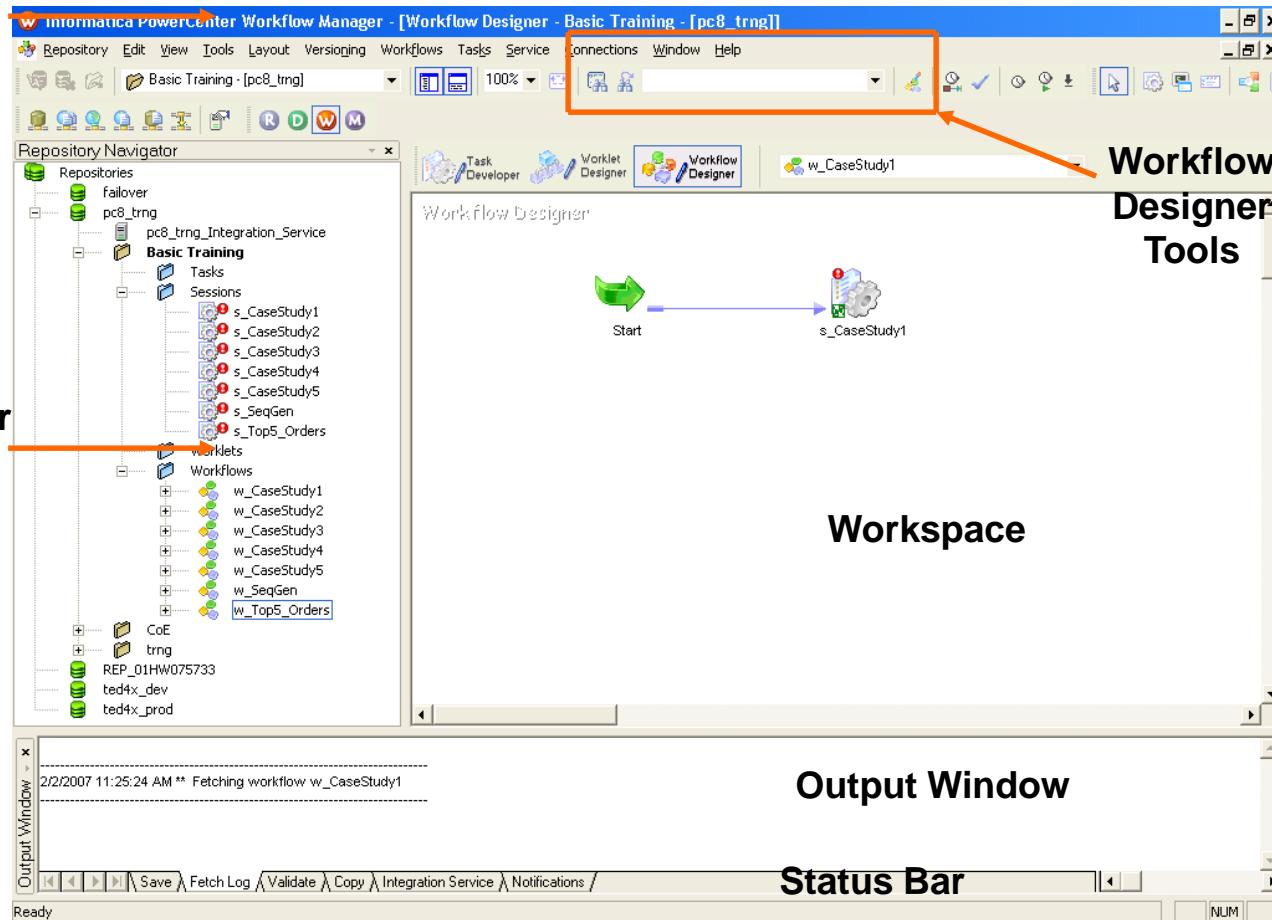
**Workflow
Designer
Tools**

**Navigator
Window**

Workspace

Output Window

Status Bar



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

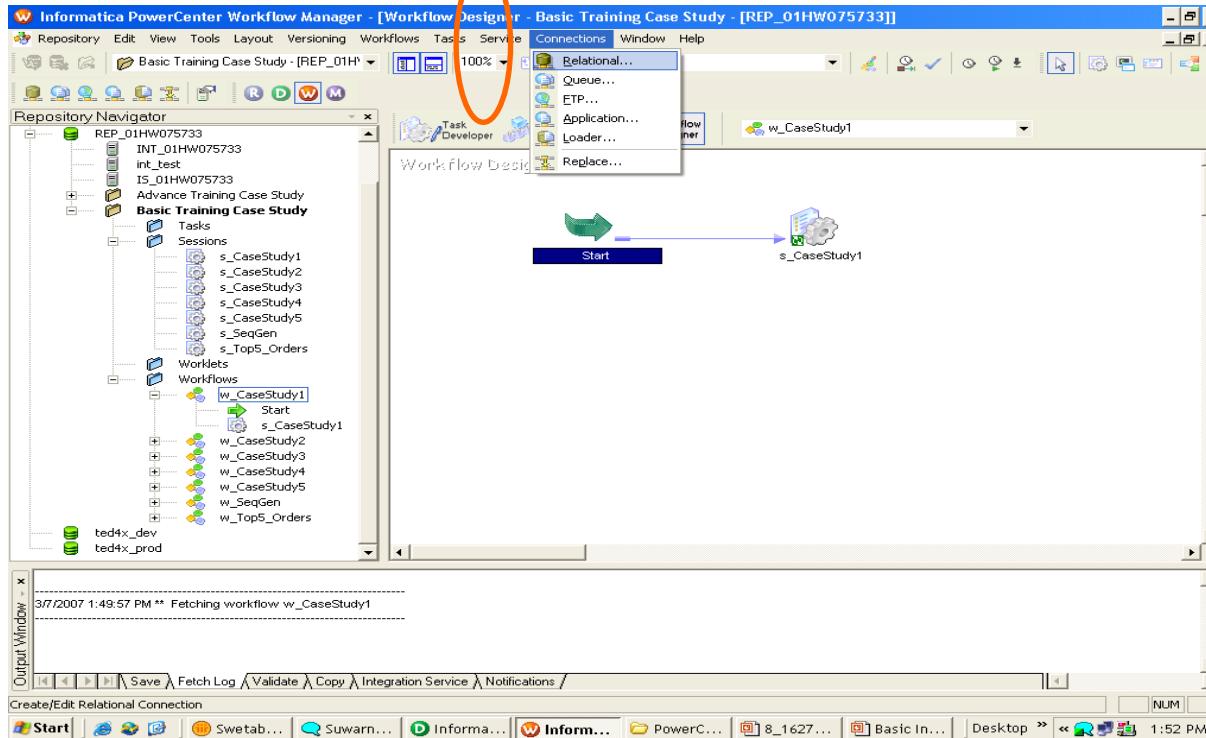


- Creates objects that contain a set of tasks
- Worklet objects are reusable

Source & Target Connections



- Configure Source & Target data access connections
 - Used in Session Tasks



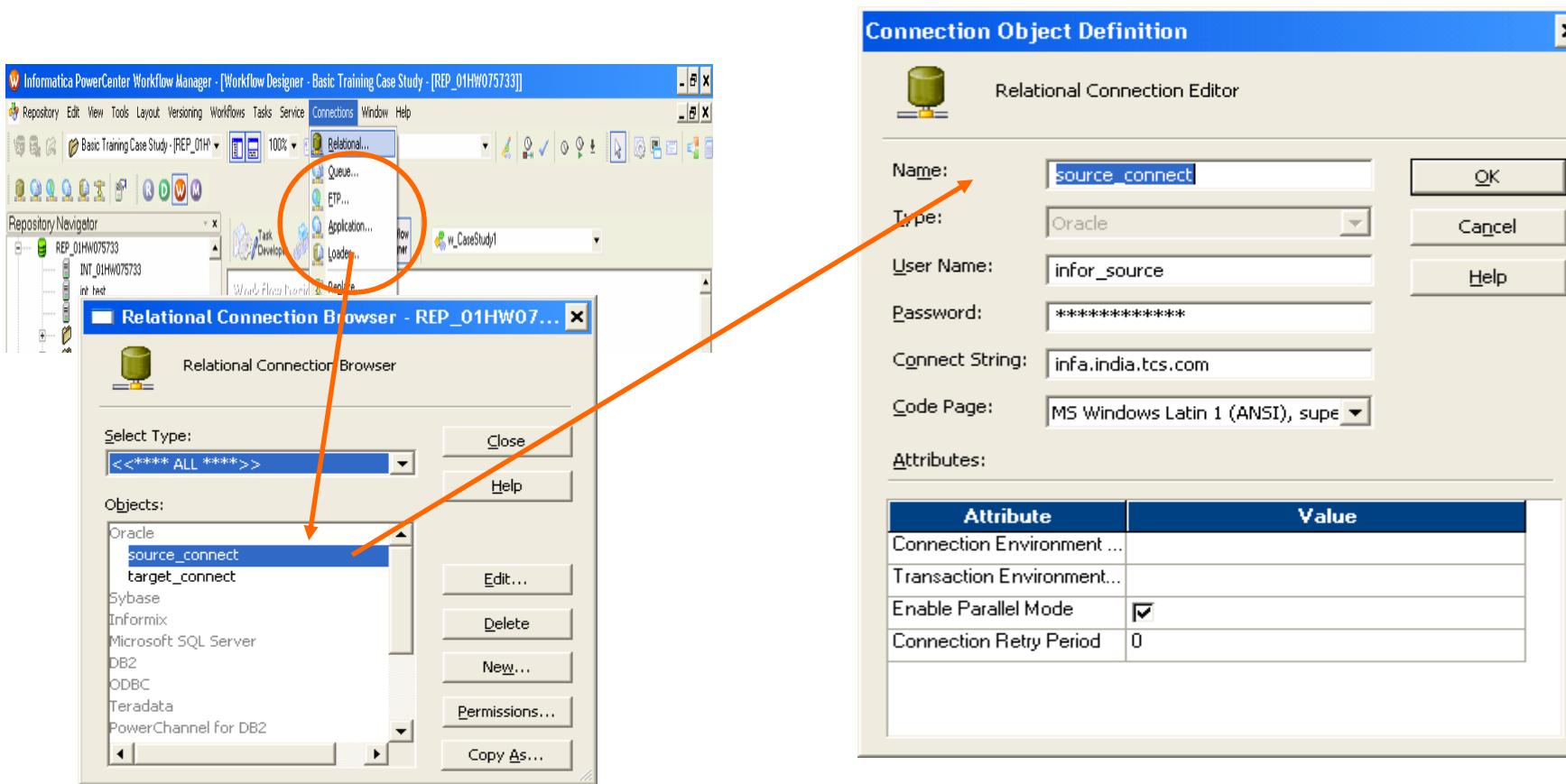
Configure:

- Relational
- MQ Series
- FTP
- Application
- Loader

Relational Connections (Native)



- Create a relational (database) connection
 - Instructions to the Integration Service to locate relational tables
 - Used in Session Tasks



Relational Connection Properties

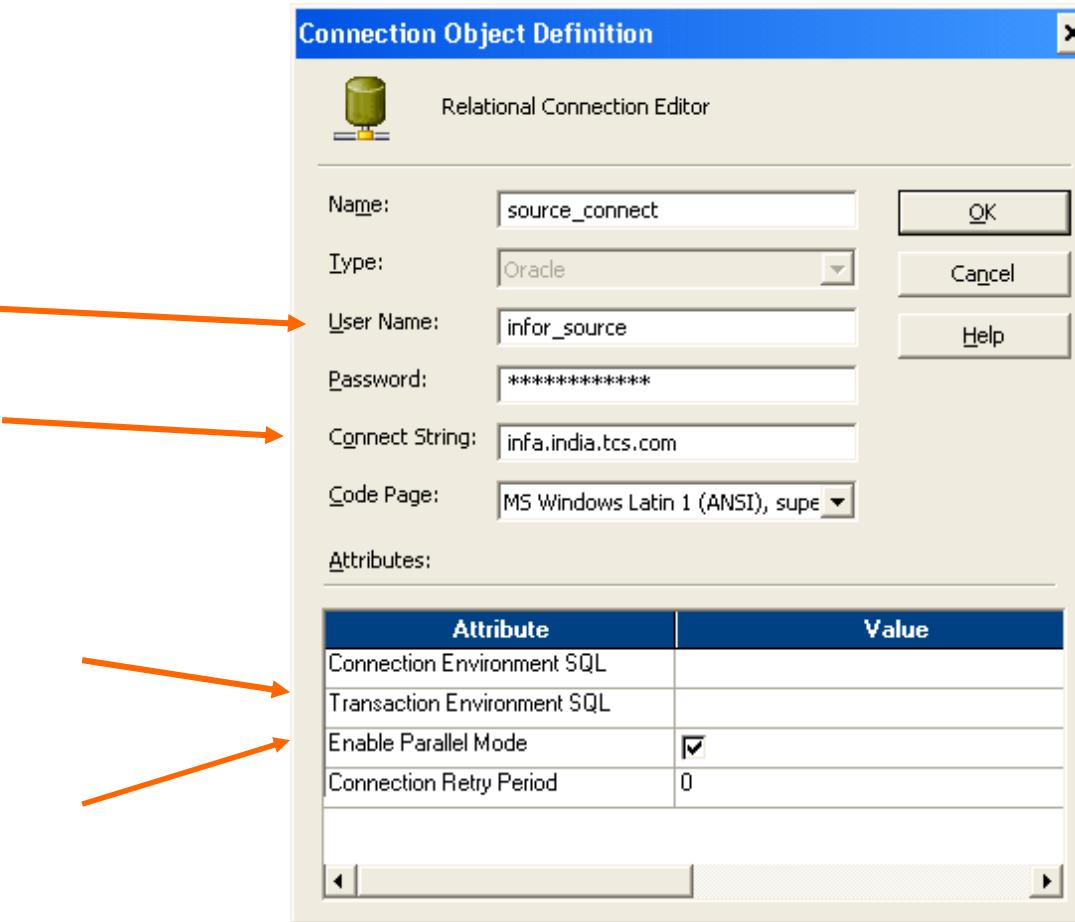
- Define native relational (database) connection

User Name/Password

Database connectivity information

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

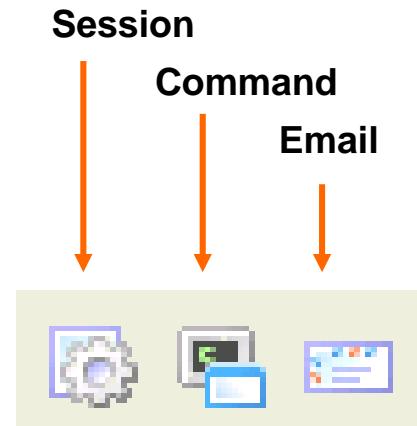
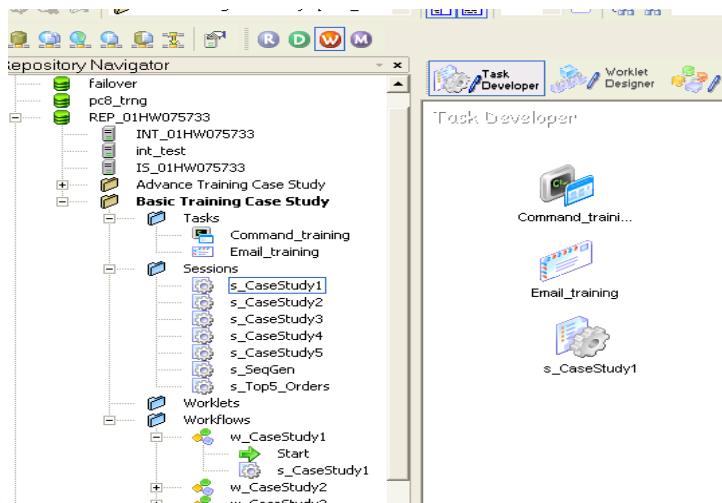
Optional Environment SQL (executed before initiation of each transaction)



Task Developer



- Create basic Reusable “building blocks” – to use in any Workflow
- Reusable Tasks
 - Session - Set of instructions to execute Mapping logic
 - Command - Specify OS shell / script command(s) to run during the Workflow
 - Email - Send email at any point in the Workflow



Session Tasks



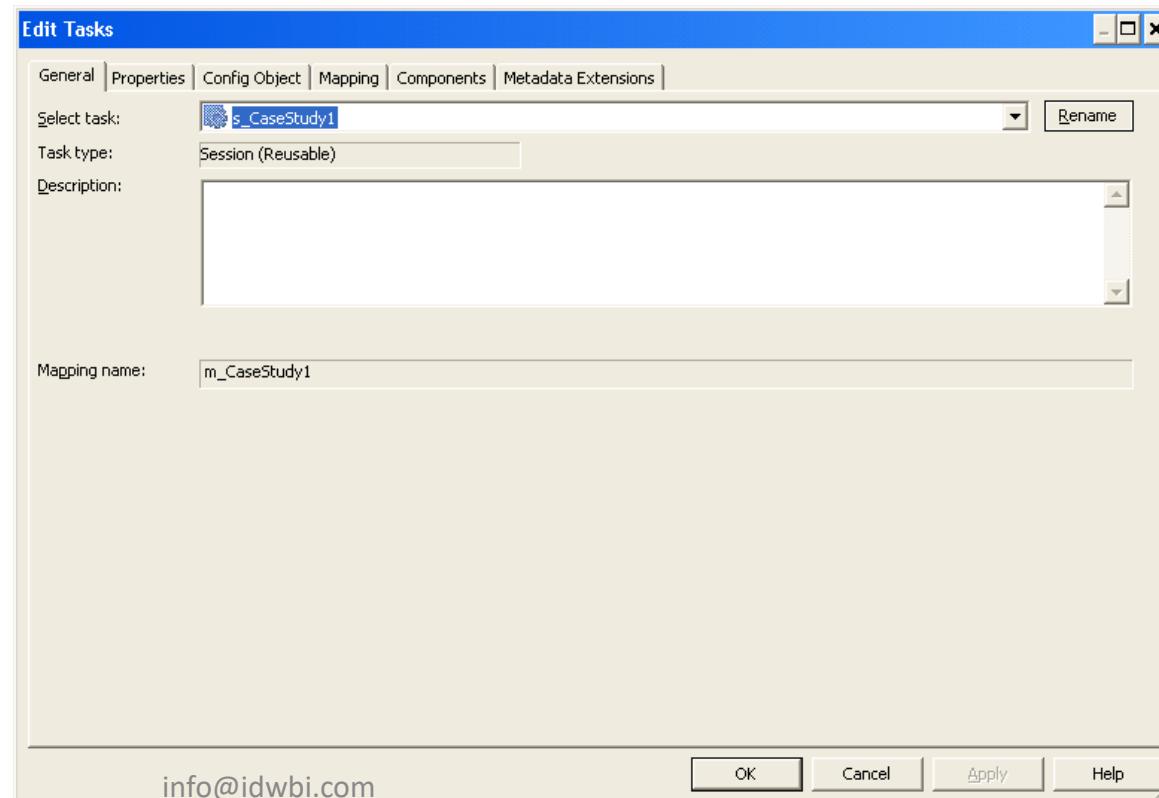
After this section, you will be familiar with:

- How to create and configure Session Tasks
- Session Task properties
- Transformation property overrides
- Reusable vs. non-reusable Sessions
- Session partitions

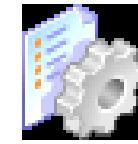
Session Task



- Integration Service instructs to runs 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)



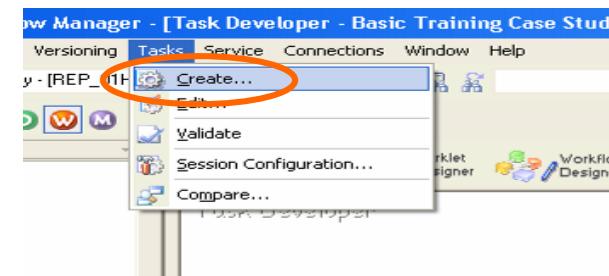
Session Task



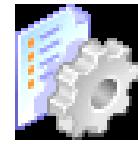
- Created to execute the logic of a mapping (one mapping only)
- Session Tasks can be created in the Task Developer (reusable) or Workflow Developer (Workflow-specific)
- Steps to create a Session Task
 - Select the Session button from the Task Toolbar or
 - Select menu Tasks -> Create



Session Task Bar Icon



Session Task - Sources



Edit Tasks

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

Select task: **s_CaseStudy1**

Task type: Session (Reusable)

Sources

Instance	Readers
sq SQ_ITEMS	Relational Reader
sq SQ_ORDER_ITEMS	Relational Reader

Connections

Type	Value
sq SQ_ITEMS - DB Connection	Relational source_connect
sq SQ_ORDER_ITEMS - DB Connection	Relational source_connect

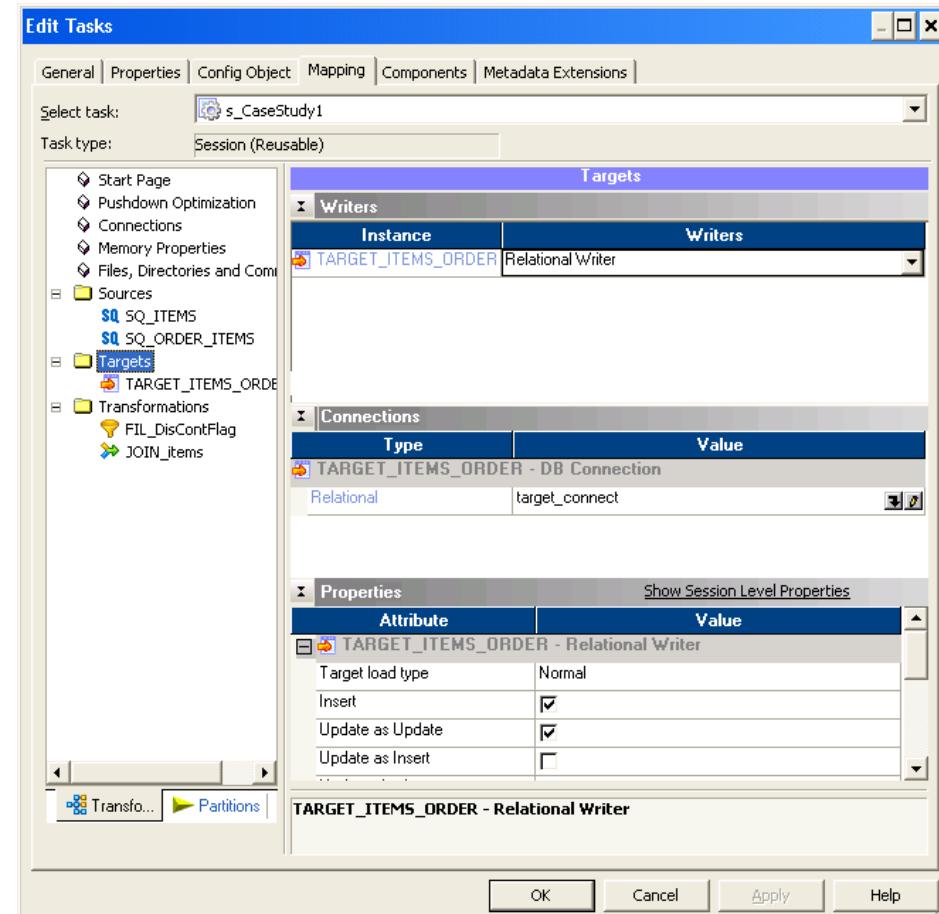
Properties

Attribute	Value
ITEMS - Source	Owner Name
sq SQ_ITEMS - Source Qualifier	User Defined Join
	Number Of Sorted Ports 0

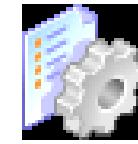
ITEMS - Source

OK Cancel Apply Help

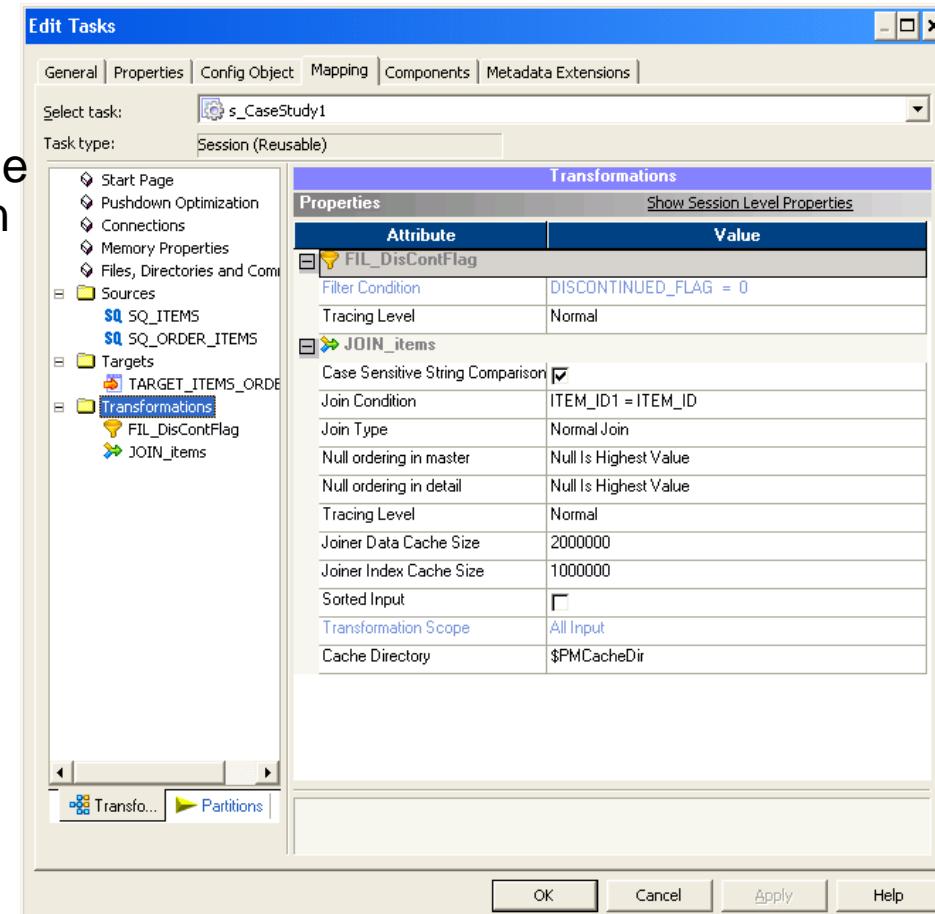
Session Task - Targets



Session Task - Transformations



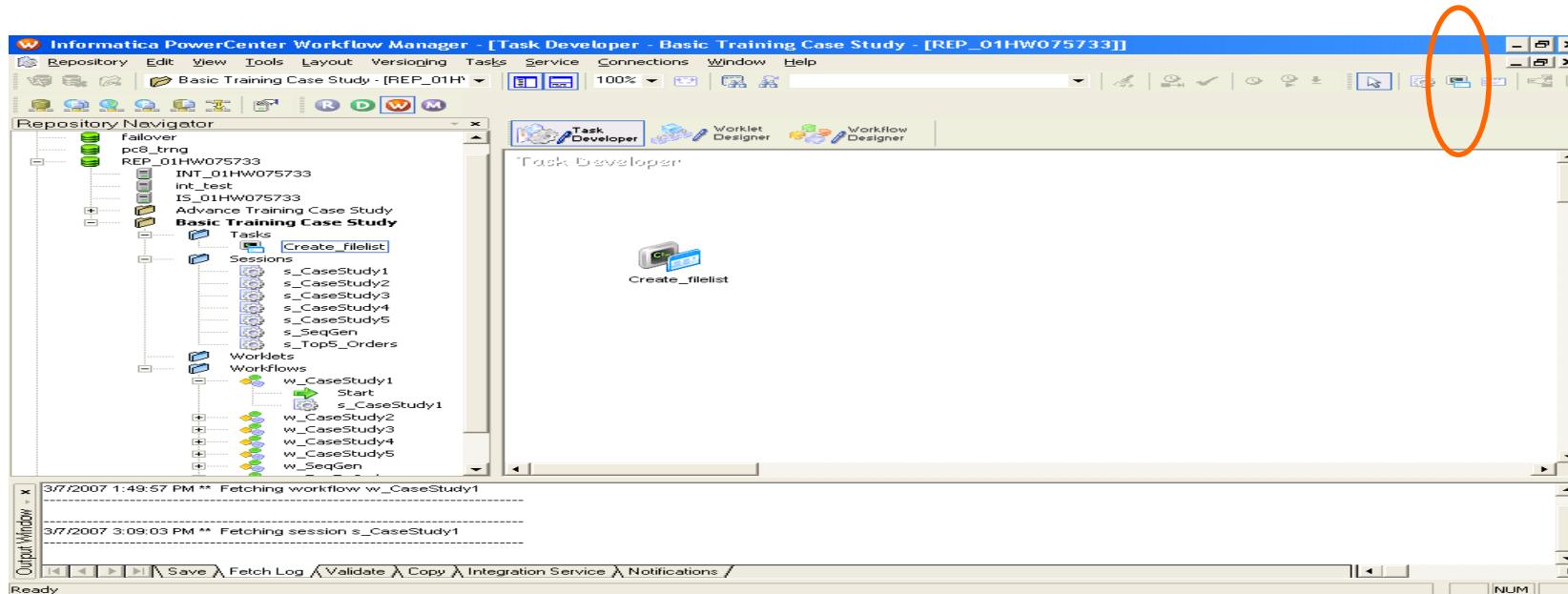
- Allows overrides of some transformation properties
- Does not change the properties in the Mapping





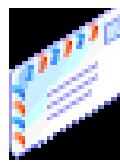
Command Task

- 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 configured in the Task Developer, the Command Task is reusable (optional)

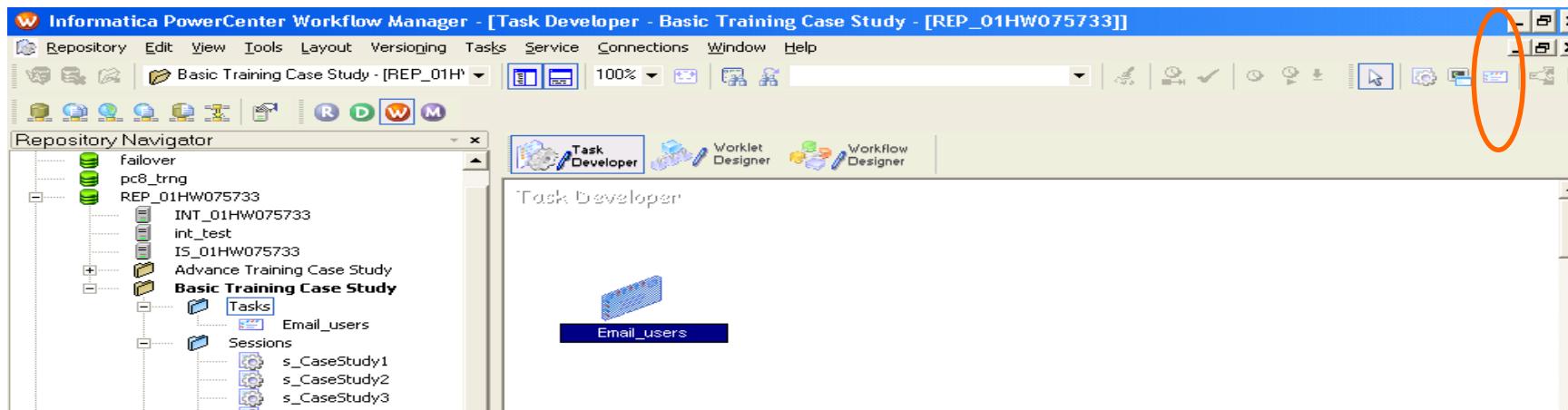


Commands can also be referenced in a Session through the Session “Components” tab as Pre- or Post-Session commands

Email Task



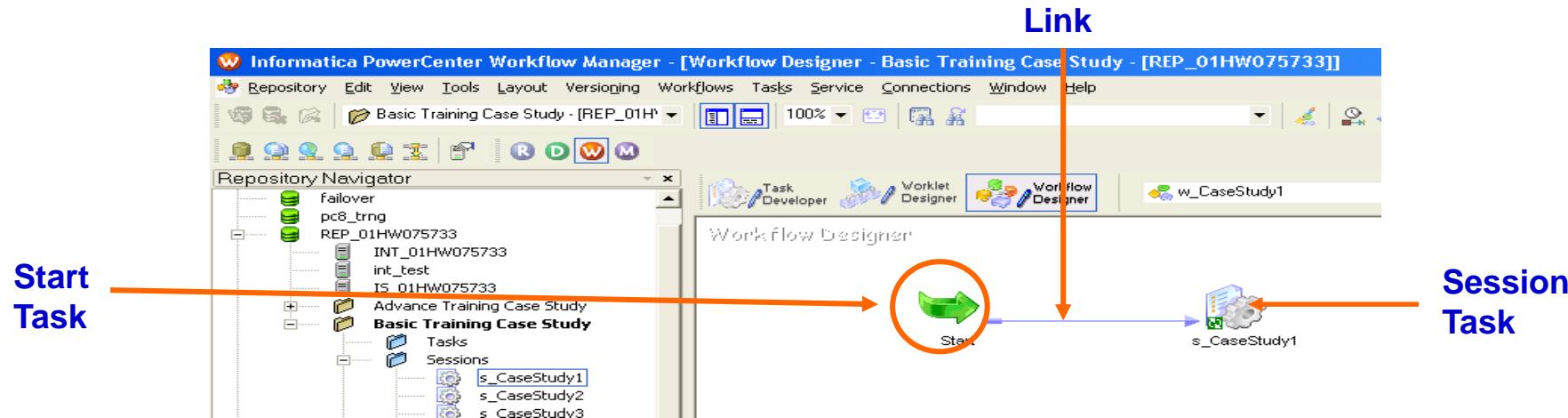
- Sends email during a workflow
- Becomes a component of a Workflow (or Worklet)
- If configured in the Task Developer, the Email Task is reusable (optional)
- Email can be also sent by using post-session email option and suspension email options of the session. (Non-reusable)



Workflow Structure



- A Workflow is set of instructions for the Integration Service 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

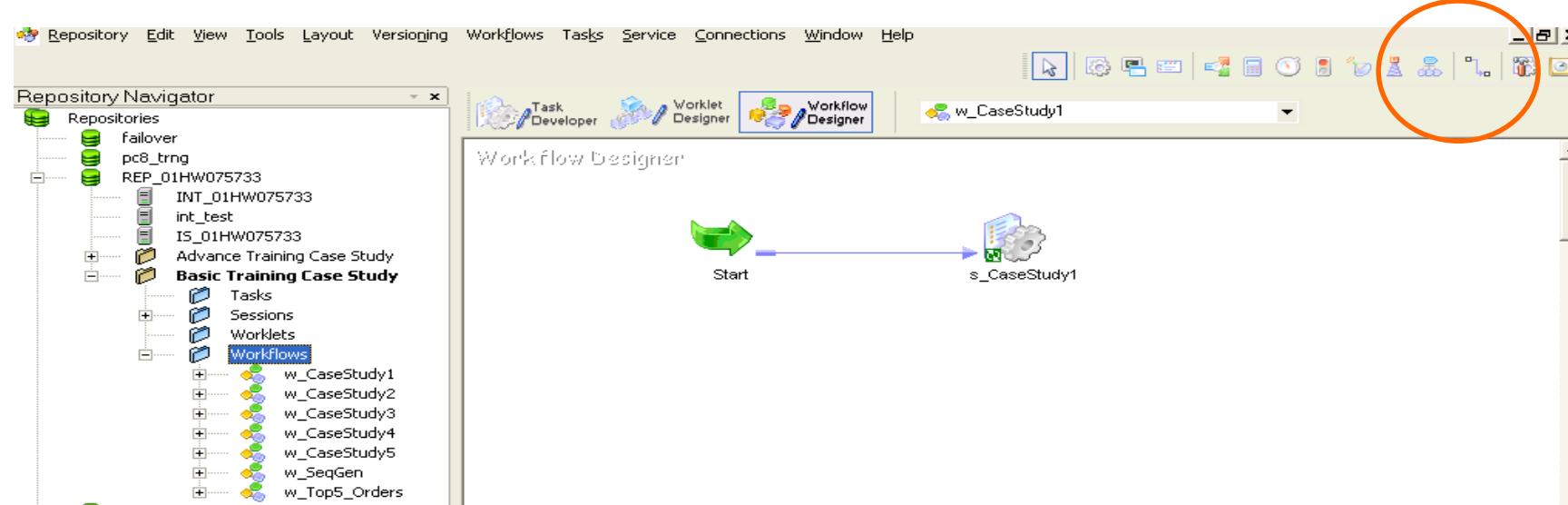


Additional Workflow Components



Two additional components are Worklets and Links

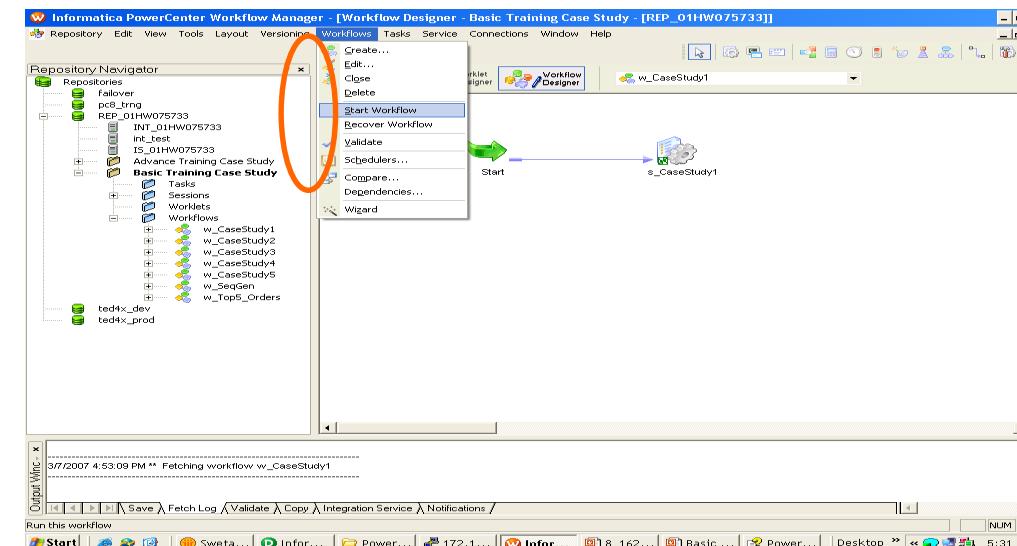
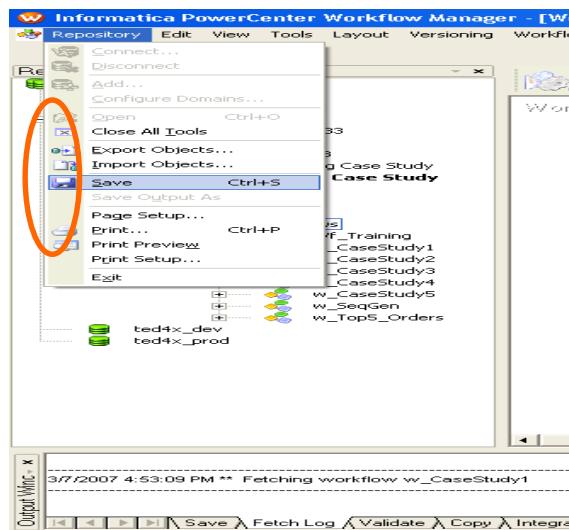
- Worklets are objects that contain a series of Tasks
- Links are required to connect objects in a Workflow



Building Workflow Component



- Add Sessions and other Tasks to the Workflow
- Connect all Workflow components with Links
- Save the Workflow
- Assign the workflow to the integration Service
- Start the Workflow



Sessions in a Workflow can be independently executed

Workflow Properties



Create Workflow - Wf_Training

General Properties Scheduler Variables Events Metadata Extensions

Set the properties of the workflow/worklet in the grid below:

Attribute	Value
Parameter Filename	
Write Backward Compatible Workflow Log File	<input type="checkbox"/>
Workflow Log File Name	Wf_Training.log
Workflow Log File Directory	\$PMWorkflowLogDir\
Save Workflow log by	By timestamp
Save workflow log for these runs	0
Enable HA recovery	<input type="checkbox"/>
Automatically recover terminated tasks	<input checked="" type="checkbox"/>
Maximum automatic recovery attempts	5

Automatically recover terminated tasks

Recover tasks that terminate unexpectedly in the workflow if they can be recovered.

OK Cancel Help

Customize Workflow Properties

Create Workflow - Wf_Training

Workflow log displays

General Properties Scheduler Variables Events Metadata Extensions

A workflow can have a reusable or a non-reusable scheduler attached to it. Choose " reusable" to browse through a list of reusable schedulers and select or edit from the list. Select " non-reusable" to create a new scheduler that cannot be reused:

Non Reusable Reusable

Scheduler: Scheduler

Description:

Summary: Property Value

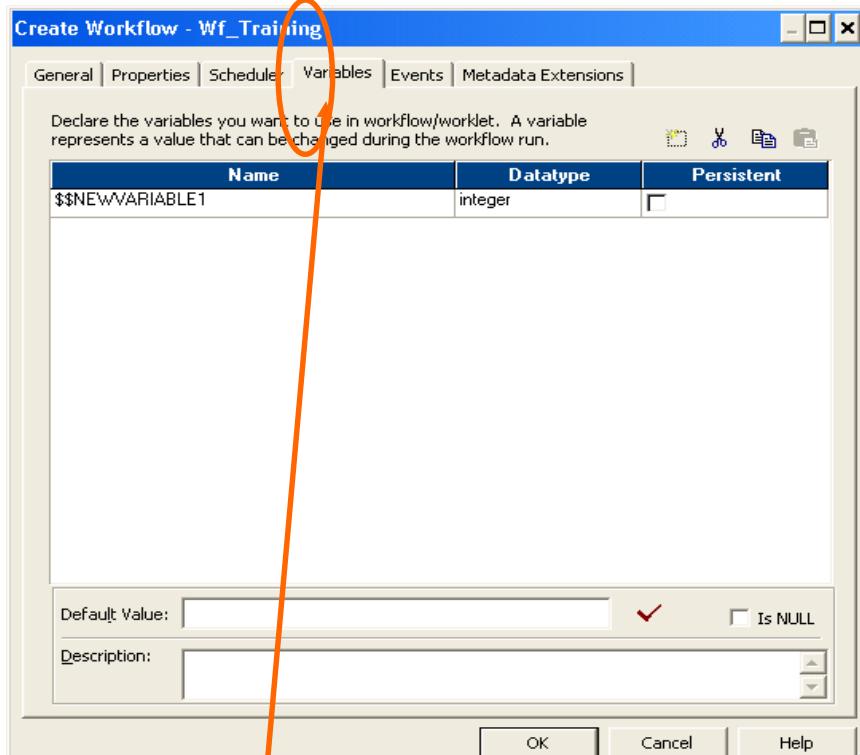
Run Options	Run on demand
-------------	---------------

OK Cancel Help

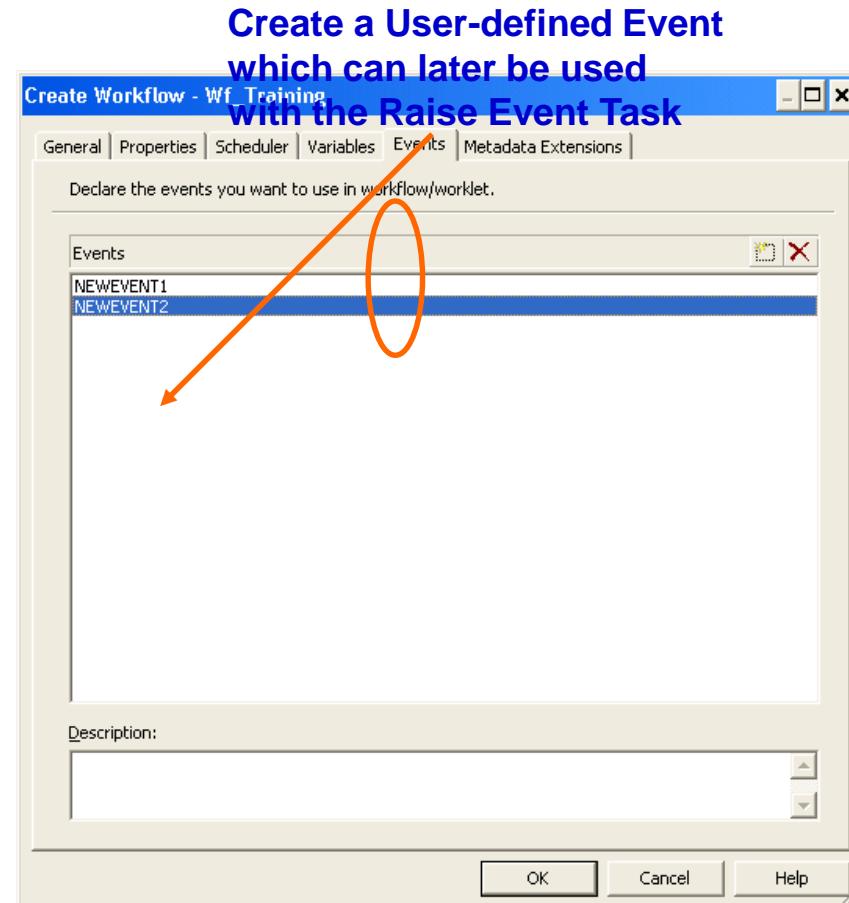
Select a Workflow Schedule (optional)

May be reusable or non-reusable

Workflows Properties



Define Workflow Variables that can be used in later Task objects
(example: Decision Task)

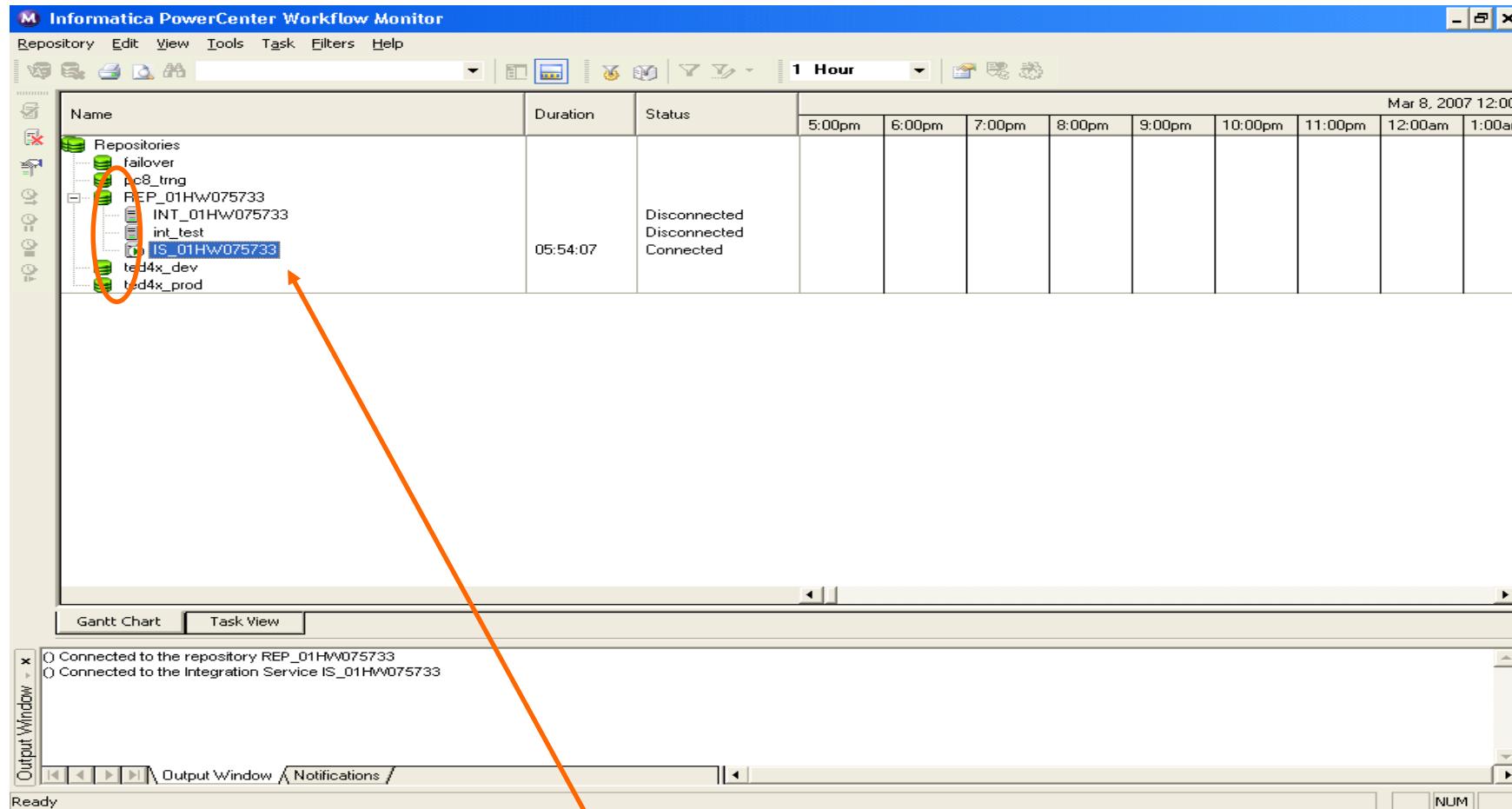


This section details -

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



Workflow Monitor Interface



Available Integration Services



Monitoring Workflows

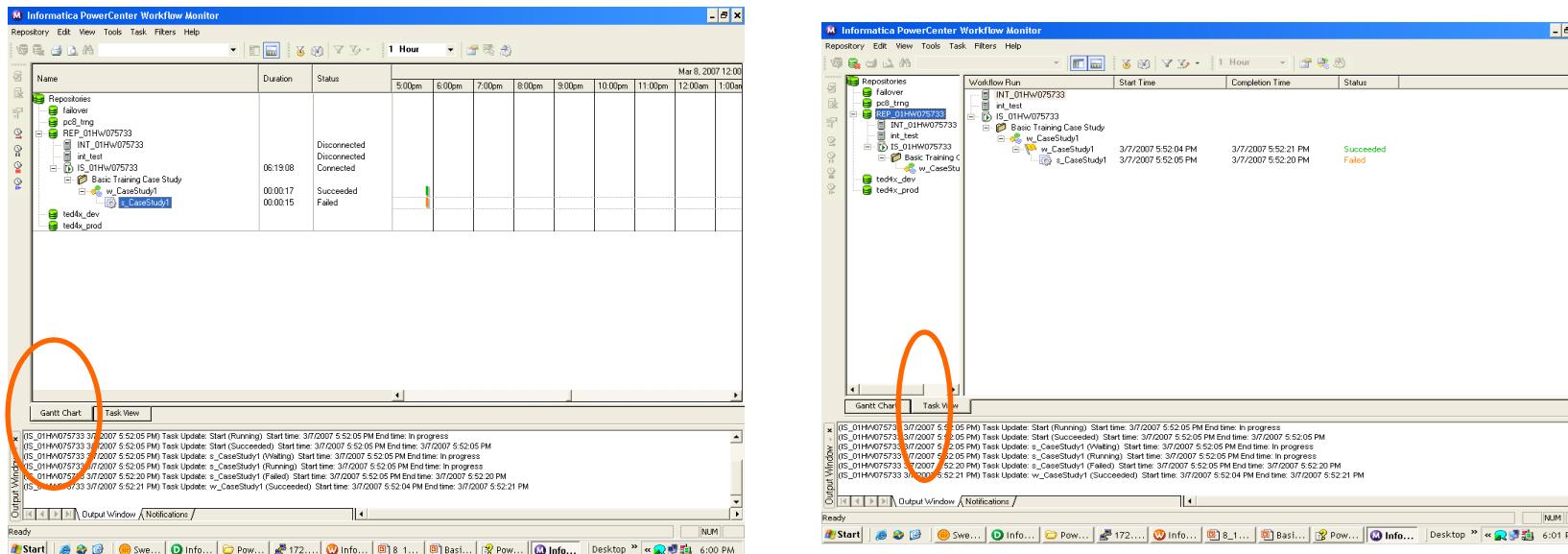
- Perform operations in the Workflow Monitor
 -  Restart -- restart a Task, Workflow or Worklet
 -  Stop -- stop a Task, Workflow, or Worklet
 -  Abort -- abort a Task, Workflow, or Worklet
 -  Recover -- recovers a suspended Workflow after a failed Task is corrected from the point of failure
- View Session and Workflow logs
- Abort has a 60 second timeout
 - If the Integration Service 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

Monitor Workflows



- The Workflow Monitor is the tool for monitoring Workflows and Tasks
- Review details about a Workflow or Task in two views
 - Gantt Chart view
 - Task view



Monitoring Workflows



Task View

Workflow Start Time Completion Time Status

Status Bar

Workflow Run	Start Time	Completion Time	Status
Basic Training Case ...	3/7/2007 5:52:04 PM	3/7/2007 5:52:21 PM	Succeeded
w_CaseStudy1	3/7/2007 5:52:05 PM	3/7/2007 5:52:05 PM	Succeeded
int_test	3/7/2007 5:52:05 PM	3/7/2007 5:52:20 PM	Failed
IS_01HW075733			
Basic Training Case Study			
w_CaseStudy1			
w_SeqGen	3/7/2007 6:09:48 PM	3/7/2007 6:09:55 PM	Succeeded
s_Case...	3/7/2007 6:09:48 PM	3/7/2007 6:09:48 PM	Succeeded
s_SeqGen	3/7/2007 6:09:48 PM	3/7/2007 6:09:55 PM	Failed
ted4x_dev			
ted4x_prod			

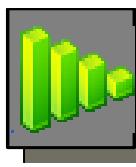
Output Window

```
IS_01HW075733 3/7/2007 6:09:48 PM Task Update: Start (Running) Start time: 3/7/2007 6:09:48 PM End time: In progress
IS_01HW075733 3/7/2007 6:09:48 PM Task Update: Start (Succeeded) Start time: 3/7/2007 6:09:48 PM End time: 3/7/2007 6:09:48 PM
IS_01HW075733 3/7/2007 6:09:48 PM Task Update: s_SeqGen (Waiting) Start time: 3/7/2007 6:09:48 PM End time: In progress
IS_01HW075733 3/7/2007 6:09:48 PM Task Update: s_SeqGen (Running) Start time: 3/7/2007 6:09:48 PM End time: In progress
IS_01HW075733 3/7/2007 6:09:55 PM Task Update: s_SeqGen (Failed) Start time: 3/7/2007 6:09:48 PM End time: 3/7/2007 6:09:55 PM
IS_01HW075733 3/7/2007 6:09:55 PM Task Update: w_SeqGen (Succeeded) Start time: 3/7/2007 6:09:48 PM End time: 3/7/2007 6:09:55 PM
```


This section introduces to -

- Rank
- Normalizer
- Stored Procedure

Rank Transformation



- Active
- Connected
- Selects the top and bottom rank of the data
- Different from MAX,MIN functions as we can choose a set of top or bottom values
- You can designate only one Rank port in a Rank transformation, i.e. Rank is decided based on one column only.
- The Integration Service uses the Rank Index port to store the ranking position for each row.
- The Designer creates a RANKINDEX port for each Rank transformation.

Normalizer Transformation



- Active
- Connected
- Used to organize data to reduce redundancy primarily with the COBOL sources
- A single long record with repeated data is converted into separate records.

Stored Procedure



- Passive
- Connected/ Unconnected
- Used to run the Stored Procedures already present in the database
- A valid relational connection should be there for the Stored Procedure transformation to connect to the database and run the stored procedure

This section discusses -

- Parameters and Variables
- Transformations
- Mapplets
- Tasks

Parameters and Variables

- System Variables
- Creating Parameters and Variables
- Features and advantages
- Establishing values for Parameters and Variables

System Variables

SYSDATE

- Provides current datetime on the Integration Service machine
 - Not a static value

\$\$\$SessStartTime

- Returns the system date value as a *string when a session is initialized*.
Uses system clock on machine hosting Integration Service
 - format of the string is database type dependent
 - Used in SQL override
 - Has a constant value

SESSSTARTTIME

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

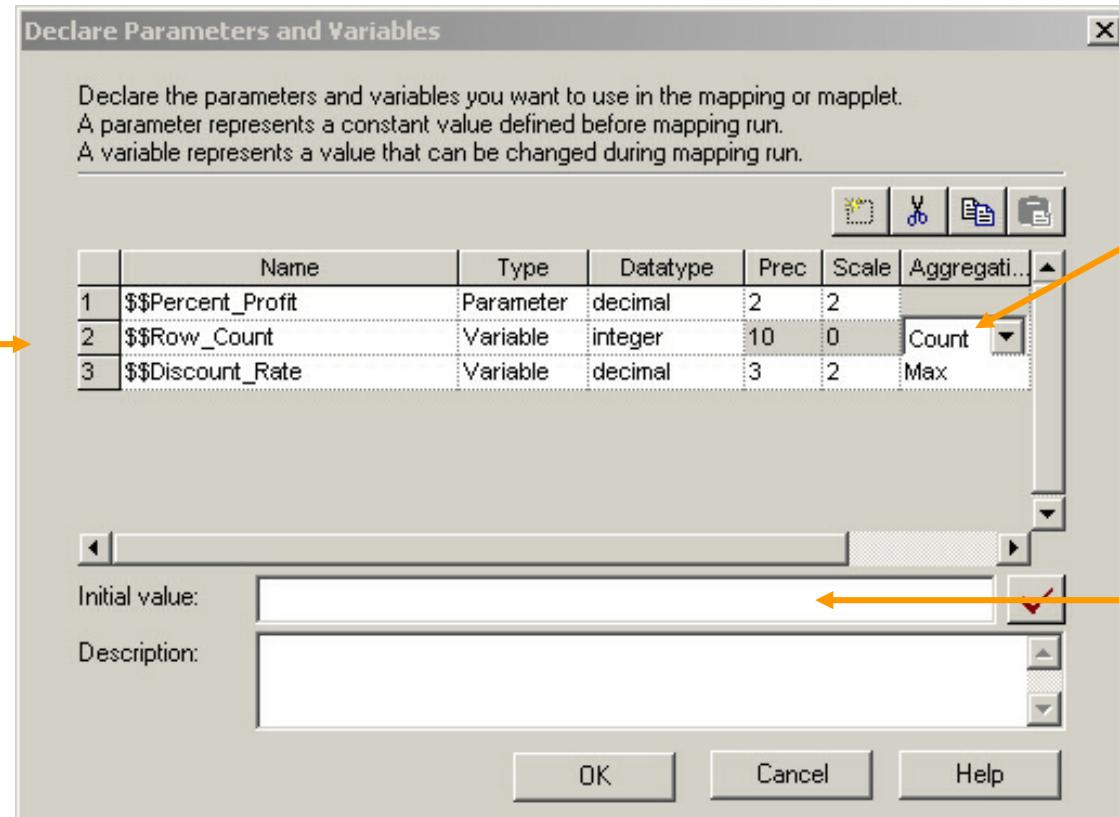
Mapping Parameters and Variables

- 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

Mapping Parameters and Variables

Sample declarations

User-defined names



Set the appropriate aggregation type

Set optional Initial Value

Declare Variables and Parameters in the Designer Mappings menu

Transformation Developer

- Transformations used in multiple mappings are called Reusable Transformations
 - Two ways of building reusable transformations
 - Using the Transformation developer
 - Making the transformation reusable by checking the reusable option in the mapping designer
 - Changes made to the reusable transformation are inherited by all the instances (Validate in all the mappings that use the instances)
 - Most transformations can be made non-reusable /reusable.
- ***External Procedure transformation can be created as a reusable transformation only

Mapplet Developer

- When a group of transformation are to be reused in multiple mappings then we develop mapplets
- Input and/ Output can be defined for the mapplet
- Editing the mapplet changes the instances of the mapplet used

Reusable Tasks

- Tasks can be created in
 - Task Developer (Reusable)
 - Workflow Designer (Non-reusable)
- Tasks can be made reusable my checking the 'Make Reusable' checkbox in the general tab of sessions
- Following tasks can be made reusable:
 - Session
 - Email
 - Command
- When a group of tasks are to be reused then use a worklet (in worklet designer)

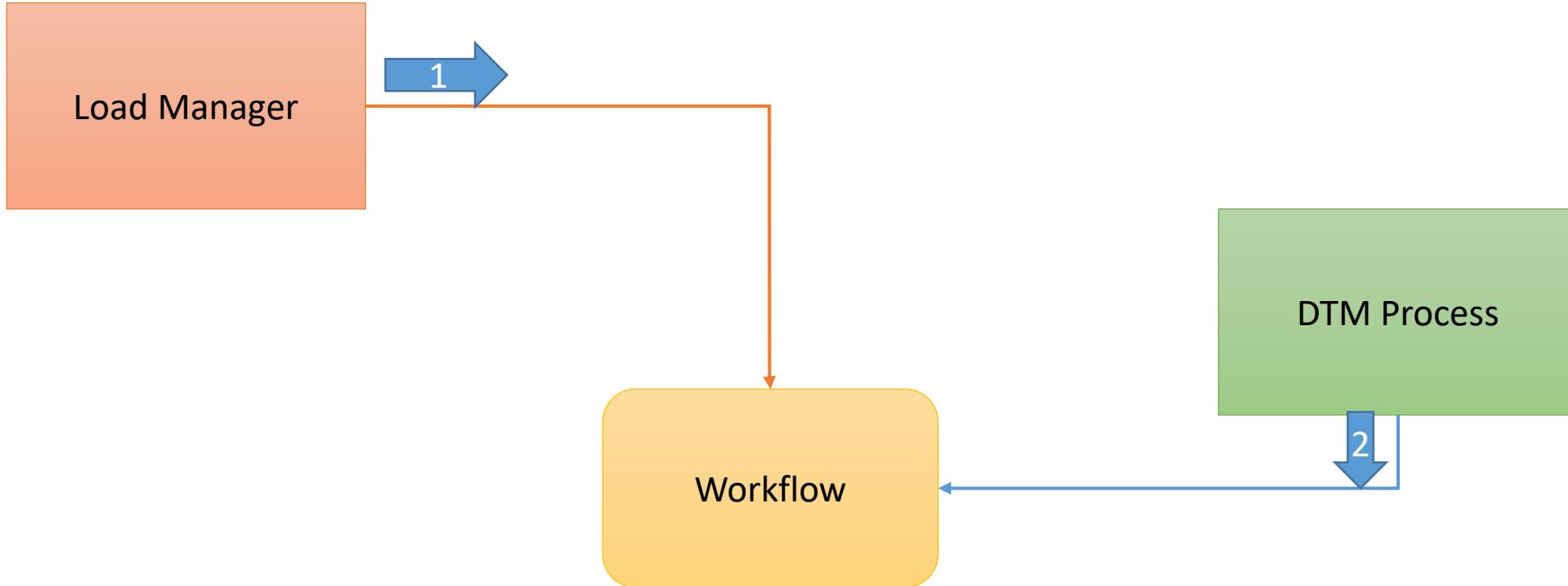
Informatica version 8.6 vs 9.x

Aspect	9x
Architecture	No Change
Client Tools	IDQ – Developer, Analyst
Lookup Transformations	Cache update feature
	Multiple rows return of look up making it an Active transformation
SQL transformation	Introduction of Query mode (DML – Active) & Script Mode (DDL – Passive)
XML Transformation	XML error can be passed as an output to a target without session failure.
Resilience	DB deadlock resilience. Auto recovery during dead lock.
Monitoring	Job monitoring can be done directly from Admin Console
Deployments	Deployment group creation feature is available

Informatica version 9x vs 9.5

Aspect	9x
Architecture	No Change
Client Tools	IDQ – Developer, Analyst
Lookup Transformations	Cache update feature
	Multiple rows return of look up making it an Active transformation
SQL transformation	Introduction of Query mode (DML – Active) & Script Mode (DDL – Passive)
XML Transformation	XML error can be passed as an output to a target without session failure.
Resilience	DB deadlock resilience. Auto recovery during dead lock.
Monitoring	Job monitoring can be done directly from Admin Console
Deployments	Deployment group creation feature is available

Workflow Execution Process



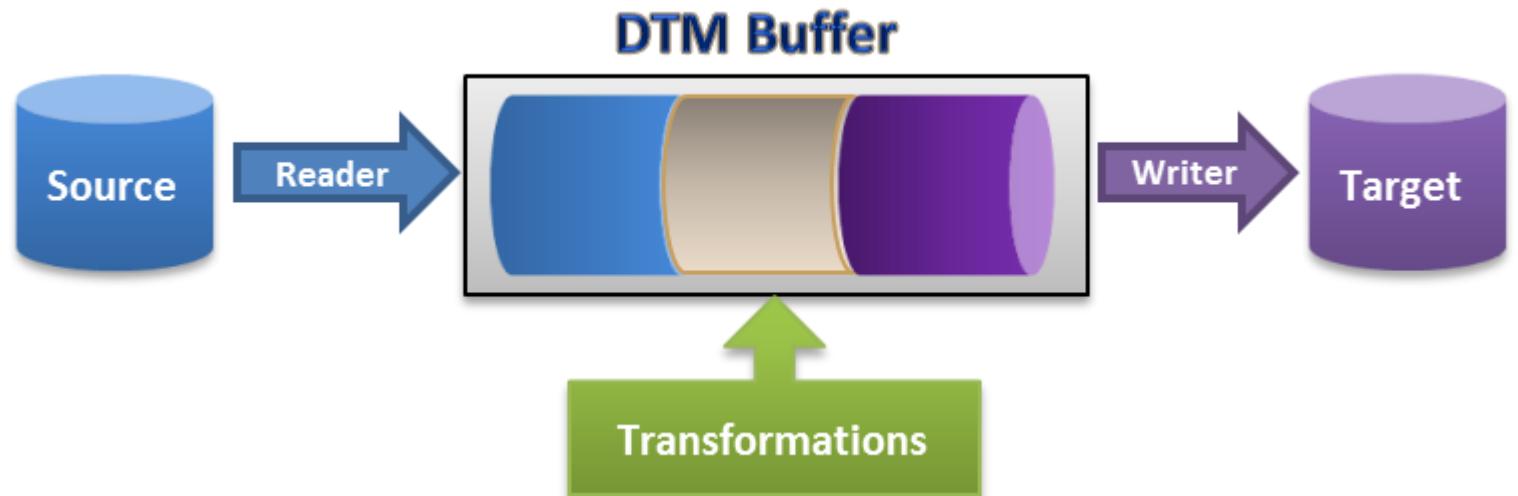
Load Manager Process Steps

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

DTM Process Steps

- Fetches session and mapping metadata from the repository.
- Creates and expands session variables.
- Creates the session log file.
- Validates session code pages if data code page validation is enabled. Checks query conversions if data code page validation is disabled.
- Verifies connection object permissions.
- Runs pre-session shell commands.
- Runs pre-session stored procedures and SQL.
- Creates and runs mapping, reader, writer, and transformation threads to extract, transform, and load data.
- Runs post-session stored procedures and SQL.
- Runs post-session shell commands.
- Sends post-session email.

- Master Thread
- Mapping Thread
- Pre and Post Session Thread
- Reader Thread
- Writer Thread
- Transformation Thread



Lookup cache – What is it?

- Lookup transformations can be configured to use cache files.
- The Integration Service builds the cache in memory when the first row is processed. If the memory is inadequate, the data is paged into a cache file.
- If you use a flat file lookup, the Integration Service always caches the lookup rows.
- By default, the cache files are created under \$PMCacheDir.
- Cache if the number (and size) of records in the Lookup table is small relative to the number of mapping rows requiring the lookup.

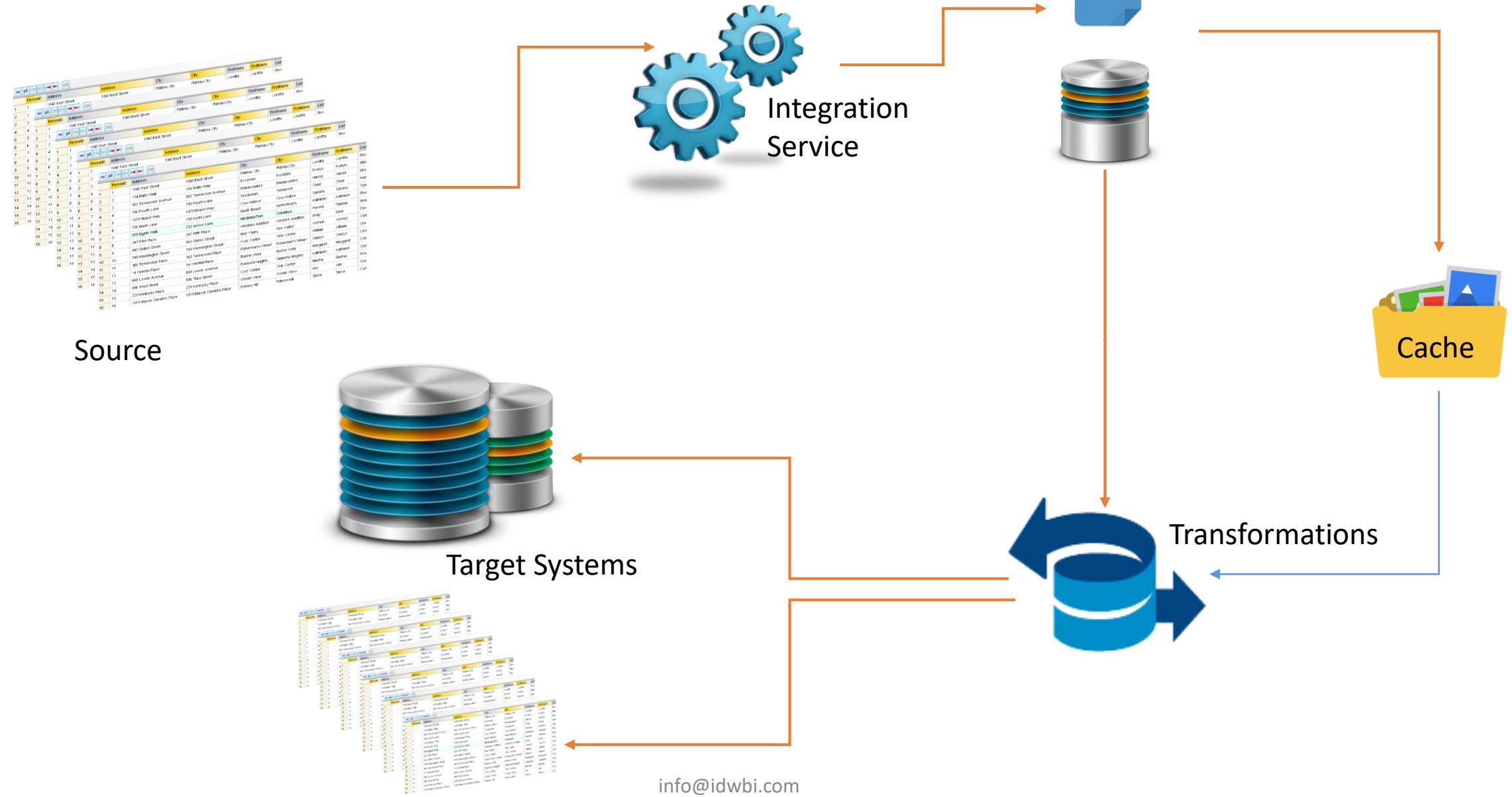
Types of Lookup Cache

- Static Cache
- Dynamic cache
- Persistent Cache
- Re-cache from Source
- Shared Cache
 - Un-named Cache
 - Named Cache

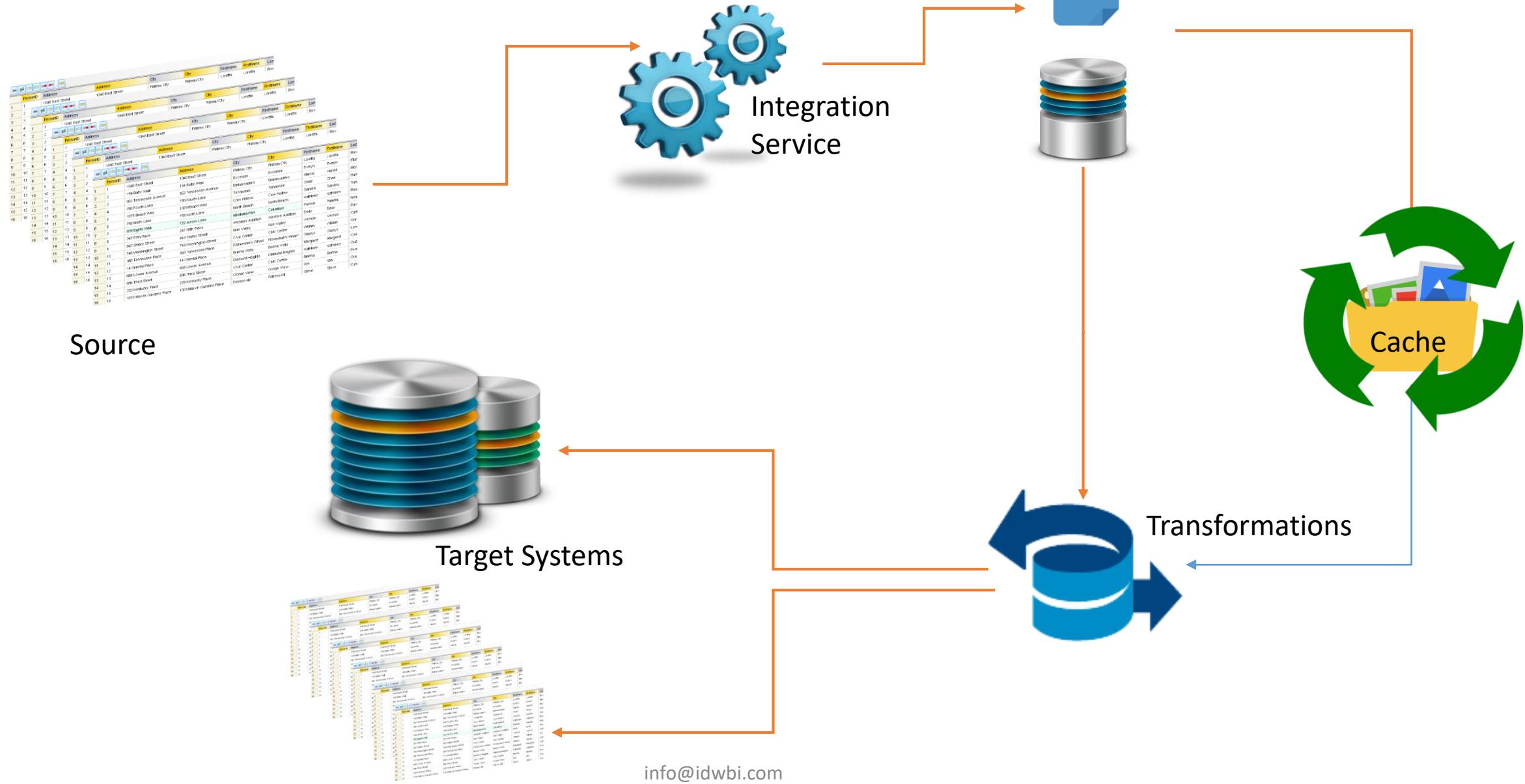
Lookup cache: Static

- This is the default type of cache.
- Cache is built when the first lookup row is processed.
- For each row that passes the transformation, the cache is queried for specified condition.
- If a match is available, the proper value is returned.
- If a match is not available either default value (for connected lookups only) or NULL is returned.
- If multiple matches are found, rows are returned based on the option specified in “Lookup policy on multiple match” in the lookup properties.

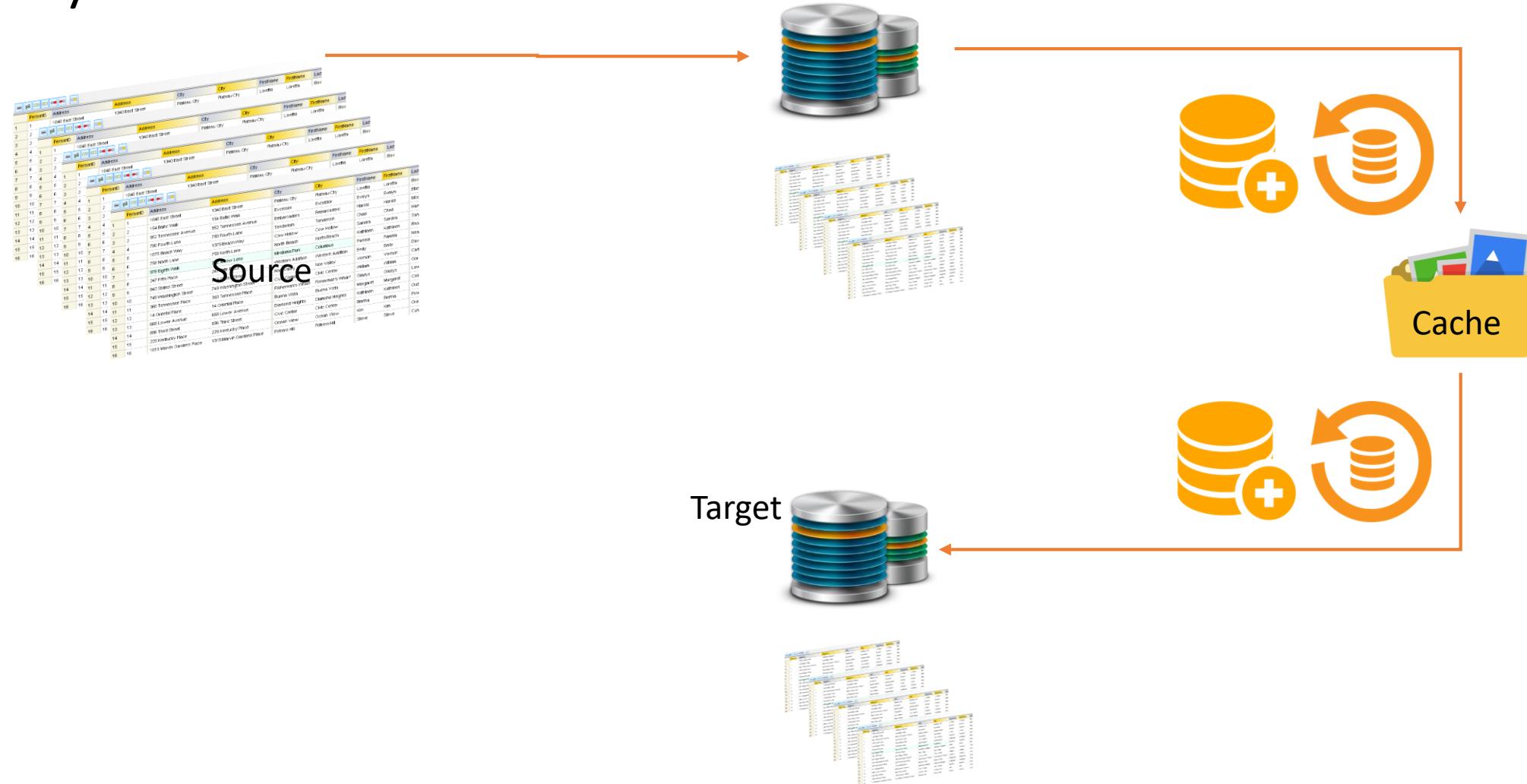
Static Cache



Persistent Cache



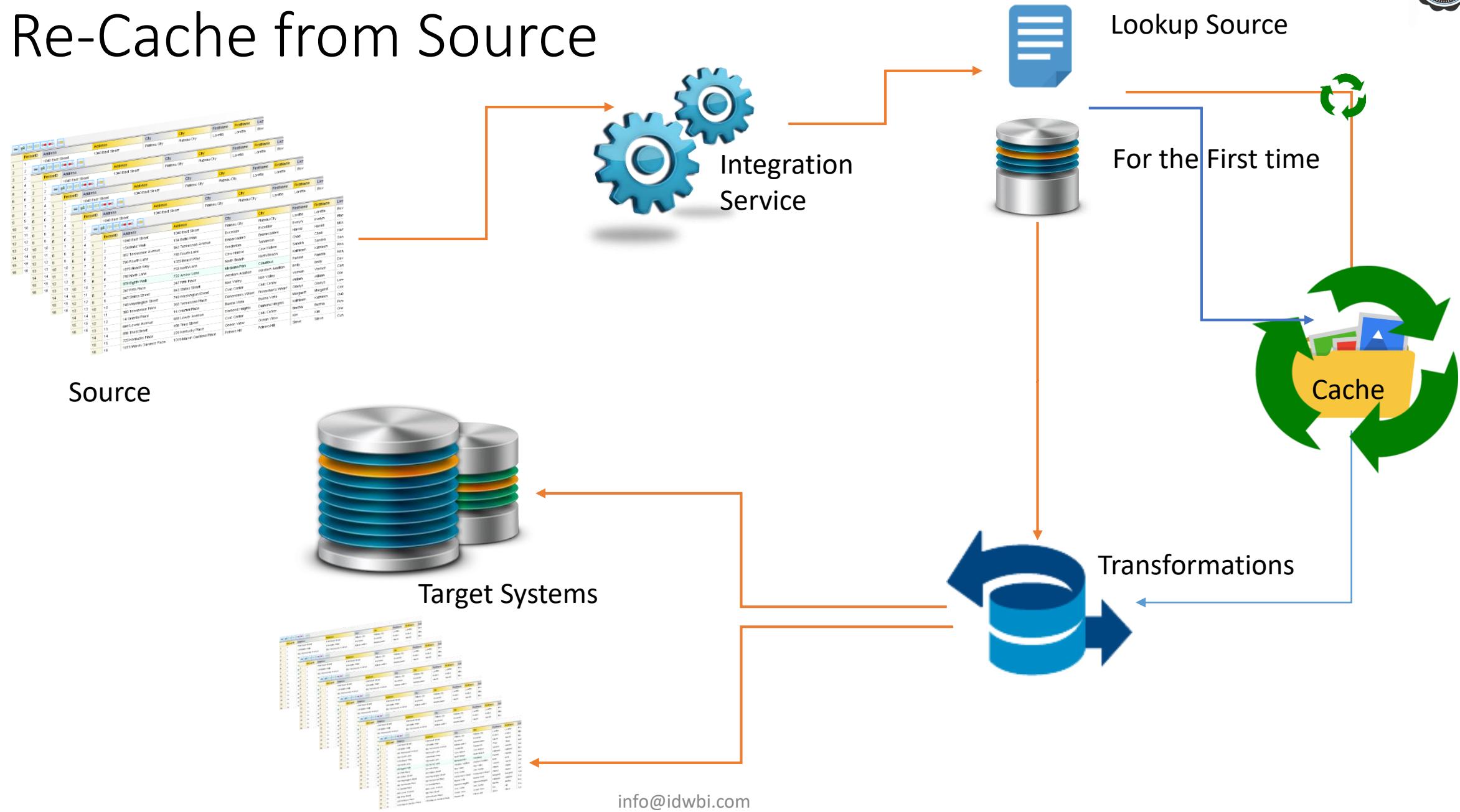
Dynamic Cache



Lookup cache: Dynamic

- The cache file is constantly updated by the following actions
- Insert - Inserts the row into the cache if it is not present and you specified to insert rows. You can configure to insert rows into cache based on input ports or generated sequence IDs.
- Update – updates the row in cache if the row is already present and an update is specified in the properties
- No change:
 - Row does not exist in cache, but you have specified to only insert new rows
 - Row does not exist in cache, but you have specified update existing rows only
 - Row exists in the cache, but based on the lookup conditions nothing changes

Re-Cache from Source



Shared Cache – Unnamed Cache

- When two Lookup transformations share an unnamed cache, the Integration Service saves the cache for a Lookup transformation and uses it for subsequent Lookup transformations that have the same lookup cache structure.
- For example, if you have two instances of the same reusable Lookup transformation in one mapping and you use the same output ports for both instances, the Lookup transformations share the lookup cache by default
- Shared transformations must use the same ports in the lookup condition. The conditions can use different operators, but the ports must be the same.

Shared Cache – Named Cache

- You can also share the cache between multiple Lookup transformations by using a persistent lookup cache and naming the cache files.
- When the Integration Service processes the first Lookup transformation, it searches the cache directory for cache files with the same file name prefix.
- If the Integration Service finds the cache files and you do not specify to recache from source, the Integration Service uses the saved cache files.
- If the Integration Service does not find the cache files or if you specify to recache from source, the Integration Service builds the lookup cache us.
- The Integration Service saves the cache files to disk after it processes each target load order.

Shared Cache – Named Cache

- The Integration Service fails the session if you configure subsequent Lookup transformations to re-cache from source, but not the first one in the same target load order group.
- If the cache structures do not match, the Integration Service fails the session.
- The Integration Service processes multiple sessions simultaneously when the Lookup transformations only need to read the cache files.
- The Integration Service fails the session if one session updates a cache file while another session attempts to read or update the cache file.
 - For example, Lookup transformations update the cache file if they are configured to use a dynamic cache or re-cache from source.