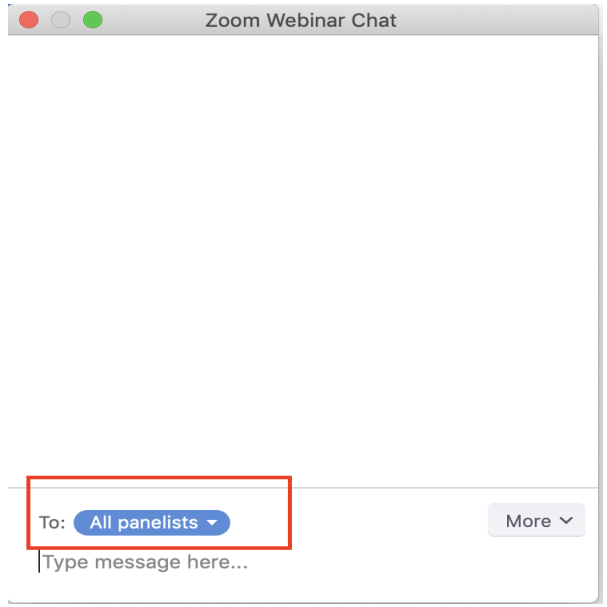
CDI – Performance Tuning

Ragavendra Kinnera
Principal Customer Success Technologist

Informatica™

Housekeeping Tips



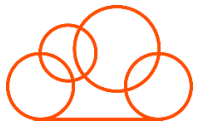
- Todays Webinar is scheduled to last 1 hour including Q&A
- All dial-in participants will be muted to enable the speakers to present without interruption
- Questions can be submitted to “All Panelists" via the Chat option and we will respond at the end of the presentation
- The webinar is being recorded and will be available to view on our INFASupport YouTube channel. The link will be emailed as well.
- Please take time to complete the post-webinar survey and provide your feedback and suggestions for upcoming topics.

Informatica

Success Portal

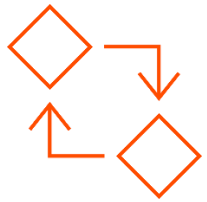
https://success.informatica.com

Learn. Adopt. Succeed.



Bootstrap product trial experience

Enriched Onboarding experience



Product Learning Paths and weekly Expert sessions

Informatica Concierge with Chatbot integrations

Tailored training and content recommendations

Informatica

Safe Harbor

The information being provided today is for informational purposes only. The development, release and timing of any Informatica product or functionality described today remain at the sole discretion of Informatica and should not be relied upon in making a purchasing decision. Statements made today are based on currently available information, which is subject to change. Such statements should not be relied upon as a representation, warranty or commitment to deliver specific products or functionality in the future.

Agenda

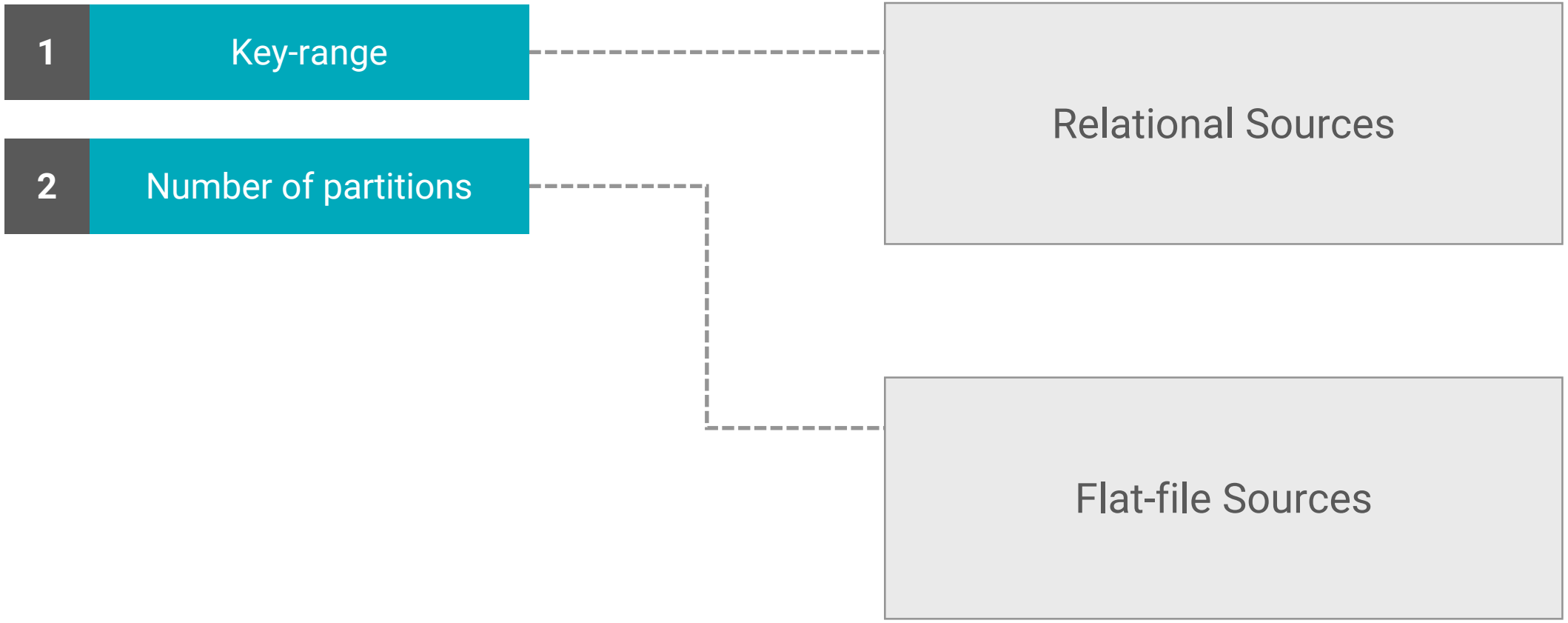
- Partitions
  - Types of Partitions
  - Rules and Guidelines
- Pushdown Optimization
  - Types of Pushdown Optimization
- Data Transformation Manager (DTM) process and its configuration

Partitions

Partitions Overview

- Partitions enable you to optimize performance for mapping tasks
  - Reduces execution time of the task by processing partitions of data concurrently
- Enable partition while configuring the Source transformation
  - When you configure partitions in the Source transformation, partitioning occurs throughout the mapping

Types of Partitioning

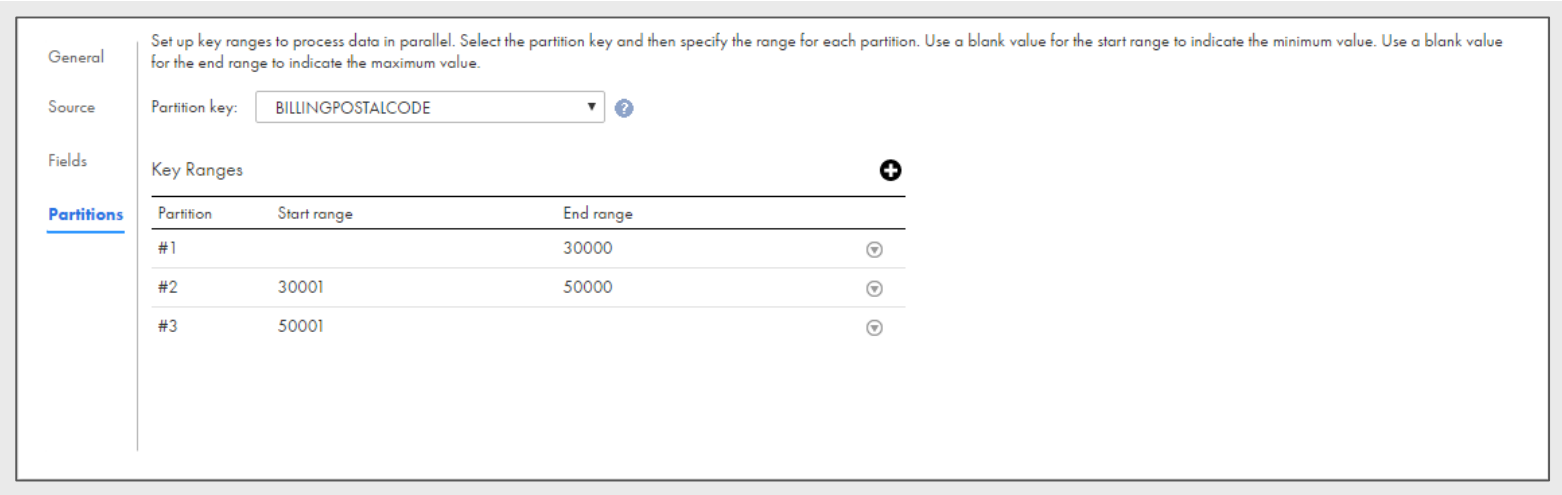


Key Range Partitioning

- Use key range partitioning method for a mapping with a relational source
- Mapping task distributes rows of data based on a field that you define as a partition key
- Specify one field in the source as the partition key
- Define a range of values for the partition key
- Key ranges can be of the following datatypes:
  - String
  - Number
  - Date/time (MM/DD/YYYY HH24:MI:SS)
- For a mapping with multiple sources, use the same number of key ranges for each source

Key Range Partitioning – Example

- Partition the source data into three partitions based on postal codes
- Specify the key ranges as follows:
  - First partition: Minimum value to 30000
  - Second partition: 30001 to 50000
  - Third partition: 50001 to maximum value
- On the Partitions tab for the Source transformation, select the BILLINGPOSTALCODE field for the partition key

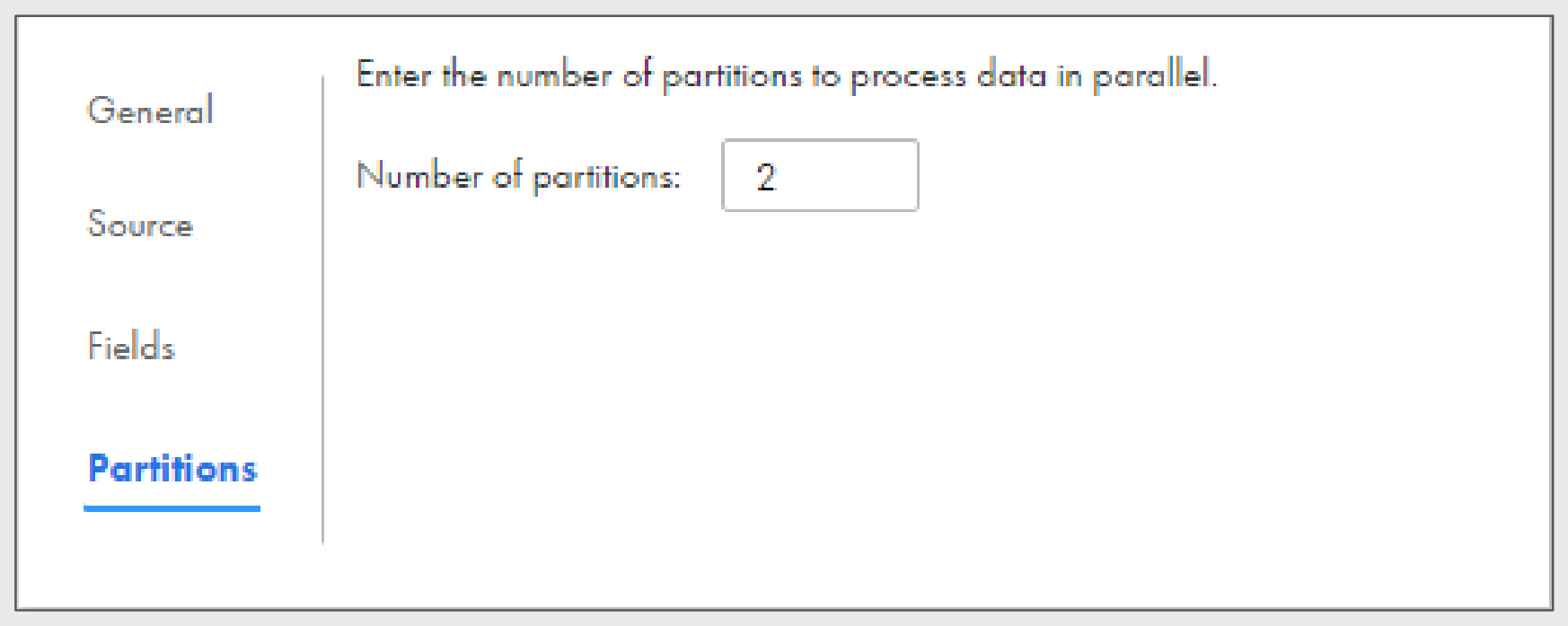


Number of Partitions

- Specify the number of partitions for a source type that does not allow key range partitioning
- Can also use this method when the mapping includes a transformation that does not support key range partitioning
- Can specify up to 64 partitions
- Consider the number of records you want to pass in the mapping to determine an appropriate number of partitions
- For a mapping with multiple sources, specify same number of partitions for each source

Number of Partitions – Example

- Mapping task uses 1GB flat file source
- Specify two partitions in the Source transformation
- On the Partitions tab of the Source transformation, enter the number of partitions



Partitioning Restrictions

- You cannot partition a mapping in the following situations:

When the mapping uses a parameterized source or source query

When the mapping includes Web Services or Hierarchy Parser transformation

When the mapping includes multiple sources that use custom relationships or advanced relationships

Informatica

Partitioning Rules and Guidelines

- For Flat File partitioning, session performance is optimal with large source files
- Set up caching in the Sequence Generator transformation
- The sequence numbers that the Normalizer and Sequence Generator transformations generate may not be sequential
- The Sorter transformation sorts data in each partition separately
- Place a Sorter transformation before any Joiner or Aggregator transformation
- You cannot use parameters for key range values

Pushdown Optimization

Pushdown Optimization Overview

- Use pushdown optimization to push transformation logic to source databases or target databases for execution
- Task converts the transformation logic into a SQL query
- The amount of transformation logic that you can push to the database depends on the database, transformation logic, and task configuration

Pushdown Optimization Types

Source pushdown optimization

The task analyzes the mapping from source to target or until it reaches the transformation logic that it cannot push to the source database.

The task generates and executes a Select statement based on the transformation logic for each transformation that it can push to the database.

The task reads the results of the SQL query and processes the remaining transformations.

Pushdown Optimization Types

Target pushdown optimization

The task analyzes the mapping from target to source or until it reaches the transformation logic that it cannot push to the target database.

The task generates an Insert, Delete, or Update statement based on the transformation logic for each transformation that it can push to the target database.

The task processes the transformation logic up to the point where it can push the transformation logic to the database. The task then executes the generated SQL on the target database.

Pushdown Optimization Types

Full pushdown optimization

The task analyzes the mapping from source to target or until it reaches the transformation logic that it cannot push to the target database.

The task generates and executes SQL statements against the source or target, based on the transformation logic that it can push to the database.

You can use full pushdown optimization when the source and target databases are in the same relational database management system.

Cross-schema Pushdown Optimization

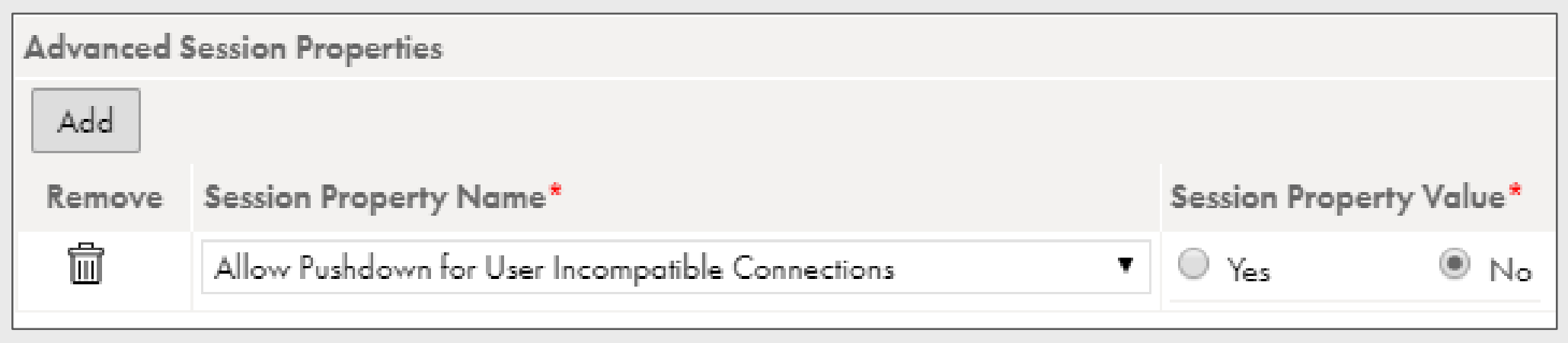
- Enable cross-schema pushdown optimization for tasks that use source or target objects associated with different schemas within the same database
- To use cross-schema pushdown optimization, create a connection for each schema
- Cross-schema pushdown optimization is enabled by default

Pushdown Optimization User-defined Parameters

- Use a user-defined parameter to perform pushdown optimization based on the parameter value defined in the parameter file
- Use a pushdown optimization user-defined parameter when you want to perform different pushdown optimization options
- Example:
  - use source or target pushdown optimization during the peak hours of the day
  - use full pushdown optimization from midnight until 2 a.m.

Pushdown Optimization – Connections

- When you run a pushdown optimization session that involves multiple database connection objects, IICS selects only one connection as the active connection
- IICS uses the active connection to execute the pushdown SQL query
- When the source and target reside in separate databases, enable session property 'Allow Pushdown for User Incompatible Connections'



Pushdown Optimization – Error Handling

- Some functionalities available in Data Integration Service are not available in Database processing
- If an error occurs in a pushdown optimization session, the database handles the error
- You cannot use IICS error handling features for pushdown optimization session failures
- For failed pushdown optimization sessions, IICS cannot perform incremental recovery

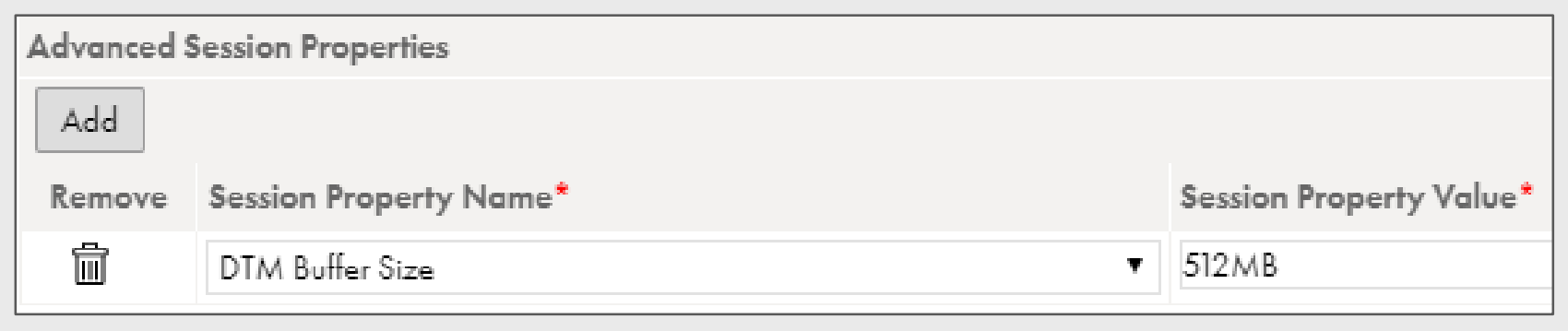
Data Transformation Manager Performance Properties

DTM Process Overview

- A DTM process is associated with the session run
- Creates and manages threads that carry out the session tasks
- Allocates process memory for the session and divides it into buffers
- Configure DTM session properties in mapping tasks

DTM Buffer Size Configuration

- DTM buffer size specifies the amount of memory that is allocated to the task from the DTM process
- When you select the DTM buffer size advanced session property in a mapping task, you must specify the session property value as either ‘Auto’ or a numeric value
- When you select ‘Auto’ the task uses automatic memory settings
- You can also provide a numeric value for the session property



DTM Buffer Size Configuration

- When a task contains large amounts of character data, increase the DTM buffer size to 24 MB
- When a task contains ‘n’ partitions, increase the DTM buffer size to at least ‘n’ times the value for the task with one partition
- When a source contains a large binary object with a precision larger than the allocated DTM buffer size, increase the DTM buffer size so that the task does not fail.

Informatica

Thank You

Informatica™