

## How to Use Full Pushdown Optimization in PowerCenter

## Abstract

You can push transformation logic to the database to improve mapping performance. A common use case is full pushdown optimization where the Integration Service attempts to push all transformation logic to the target database. This article describes how to configure full pushdown optimization in PowerCenter.

## Supported Versions

- PowerCenter 9.5.1-9.6.1

## Table of Contents

Pushdown Optimization in PowerCenter Overview. . . . .	2
Full Pushdown Optimization. . . . .	2
Full Pushdown Optimization Example. . . . .	3
Configuring a Session for Full Pushdown Optimization. . . . .	3
Rules and Guidelines for Full Pushdown Optimization. . . . .	5

## Pushdown Optimization in PowerCenter Overview

To increase session performance, use pushdown optimization to push transformation logic to the source or target database. Based on the mapping and session configuration, the Integration Service executes SQL against the source or target database instead of processing the transformation logic within the Integration Service.

The amount of transformation logic you can push to the database depends on the database, transformation logic, and mapping and session configuration. The Integration Service processes all transformation logic that it cannot push to a database.

Use the **Pushdown Optimization Viewer** to preview the SQL statements and mapping logic that the Integration Service can push to the source or target database. You can also use the **Pushdown Optimization Viewer** to view the messages related to pushdown optimization.

A common use case for pushdown is to use full pushdown optimization. In full pushdown optimization, the Integration Service attempts to push all transformation logic to the target database. If the Integration Service cannot push all transformation logic to the database, it performs both source-side and target-side pushdown optimization.

Full pushdown optimization is ideal when the source and target are in the same database or when transformations such as Aggregator and Filter transformations are processed in the source database and reduce the amount of data moved. The Integration Service processes mapping logic that it cannot push to the source or target.

## Full Pushdown Optimization

A common use case for full pushdown optimization is when the source and target databases are in the same relational database management system. For example, configure full pushdown optimization to push all the transformation logic from a Teradata source to a Teradata target.

When you run a session configured for full pushdown optimization, the Integration Service analyzes the mapping from the source to the target. When the Integration Service reaches a downstream transformation that it cannot push to the target database, it generates and executes SQL statements against the source or target based on the transformation logic that it can push to the database.

## Full Pushdown Optimization Example

HypoStores Corporation needs to extract data from an Oracle database and perform batch loads into staging and target tables in a Teradata database.

As a HypoStores developer, you can create a mapping to extract the source data from Oracle into flat files and use Teradata FastLoad to load to staging tables in the Teradata database.

You can then build the transformation logic for the data and configure full pushdown optimization to load the data from the staging tables to the target tables in the Teradata database.

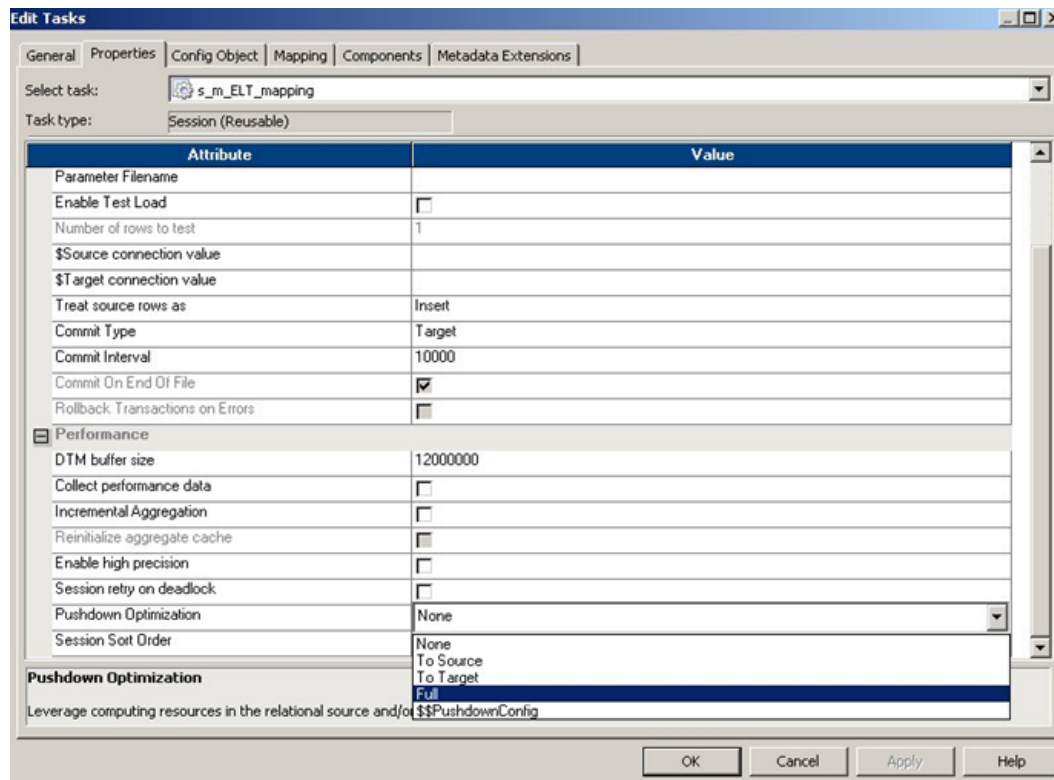
You can configure full pushdown optimization in PowerCenter to automatically generate the SQL queries and push the transformation logic from the source staging tables in the Teradata database into the target tables in the Teradata database. This helps mapping performance because PowerCenter does not have to extract data from the staging tables into memory and instead pushes the transformation logic directly to the database.

## Configuring a Session for Full Pushdown Optimization

You can configure a session for full pushdown optimization in the session properties.

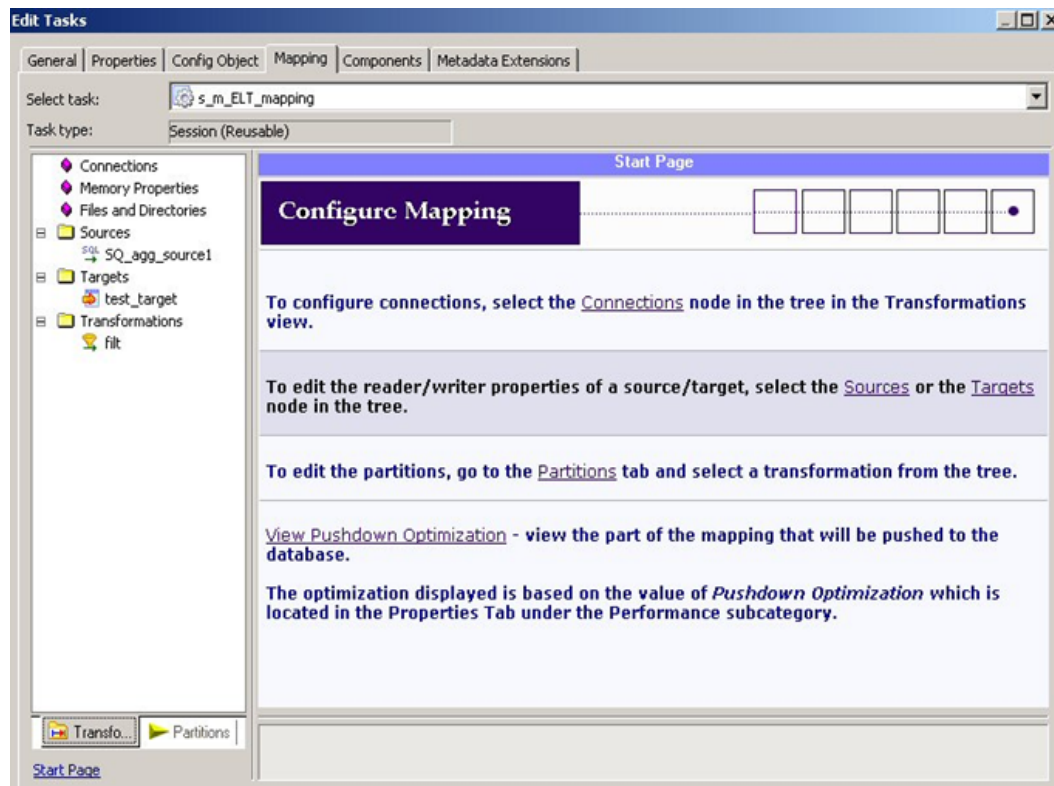
1. In the Workflow Manager, open the session properties for the session that contains the transformation logic to be pushed to the database.
2. On the **Properties** tab, select **Full**.

The following figure shows the full pushdown optimization option selected:



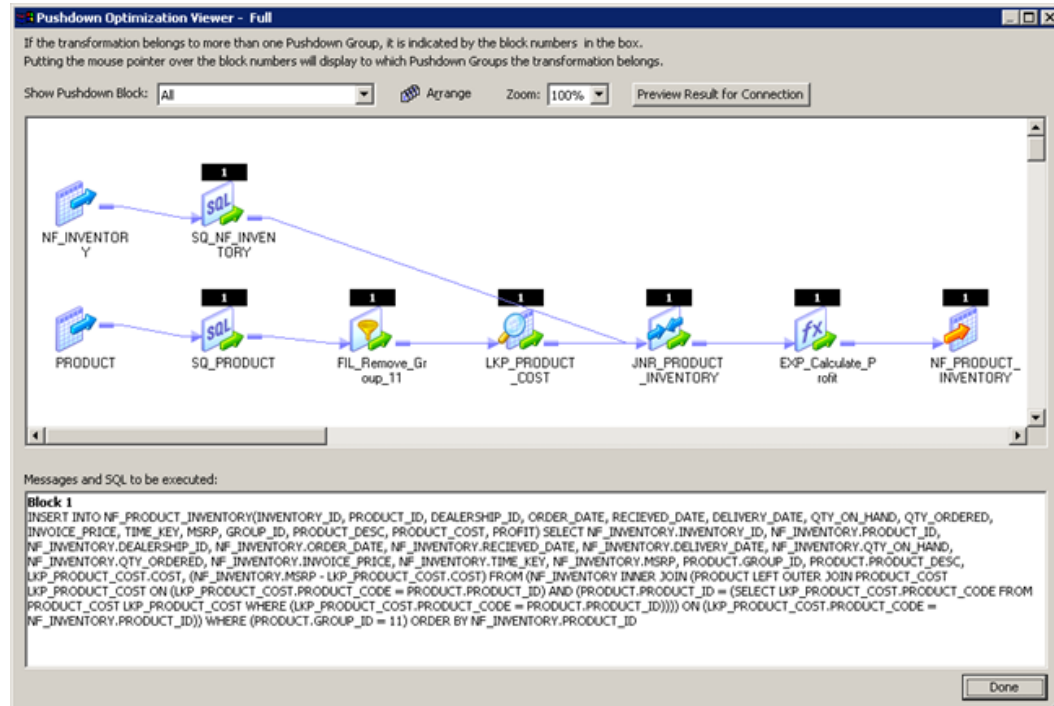
3. Click the **Mapping** tab in the session properties.

The following figure shows the **Mapping** tab:



4. Click **View Pushdown Optimization** to view the full pushdown optimization logic.

The **Pushdown Optimizer Viewer** window appears and displays the pushdown groups and the SQL that is generated to perform the transformation logic. It displays messages related to each pushdown group. It also displays numbered flags to indicate the transformations in each pushdown group. The following figure shows the full pushdown optimization logic for the mapping in the **Pushdown Optimization Viewer** window:



## Rules and Guidelines for Full Pushdown Optimization

Use the following rules and guidelines when you use pushdown optimization in PowerCenter:

- When the Integration Service pushes transformation logic to the database, it cannot track errors that occur in the database. When the Integration Service runs a session configured for full pushdown optimization and an error occurs, the database handles the errors.
- When the database handles errors, the Integration Service does not write reject rows to the reject file.
- Mappings that generate long transactions require more database resources such as locks and log space.
- The transaction is rolled back when an error is encountered because the database cannot handle partial commits.
- Pushing transformation logic to the database might generate different results than when the Integration Service runs the mapping. The database may process data differently depending on case, data formats such as Numeric conversion to Char and Date conversation to Char, null values in sort order, and precision.

## Author

Radhika Singh