

Integration Approach of SAP BW 3.x and 7.0 with Data Transfer Process



Applies to:

SAP BW 3.x & SAP BI Net Weaver 2004s. For more information, visit the [Business Intelligence homepage](#).

Summary

The objective of this article is to share and explore the various options in the Data Transfer Process and what needs to be done, when integrating it with BW 3.x data flow.

Author: Vikram Srivastava

Company: Infosys Technologies Limited

Created on: 17 February 2010

Author Bio



Vikram Srivastava is working as Technology Analyst with Infosys Technologies Limited. He has got rich experience on various BW Implementation/Support Projects in both SAP BW 3.5 and SAP BW 7.0.

Table of Contents

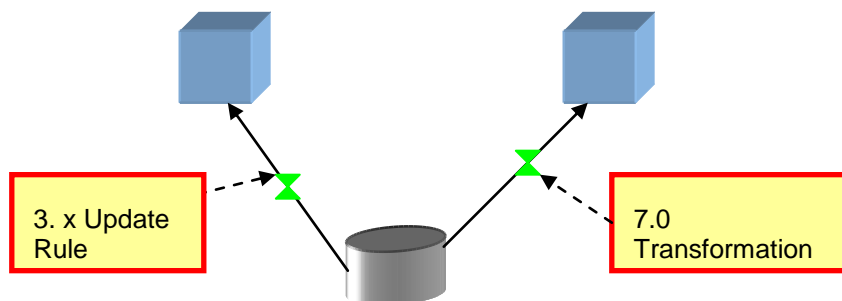
Introduction to DTP	3
Loading data from an existing 3.5 flow to 7.0 objects	3
Error Handling using DTP	4
Options in error handling	4
How to find the error Stack Table?	5
Temporary data storage using DTP	5
Extraction Modes	7
Full	7
Delta	8
Processing Mode	9
Parameters with Delta Mode	10
Significance of Semantic Groups	11
Performance Optimization using DTP	11
Handling Duplicate Data Records	14
Related Content	16
Disclaimer and Liability Notice	17

Introduction to DTP

The data transfer process (DTP) is the process to transfer the data within the BW data warehouse. It is used to update data from one data target to another data target. It also enables error handling of records.

Loading data from an existing 3.5 flow to 7.0 objects

While loading the data to an existing 3.5 flow e.g., DSO to Cube (Delta load), after the completion of the data load the data mart status is set.

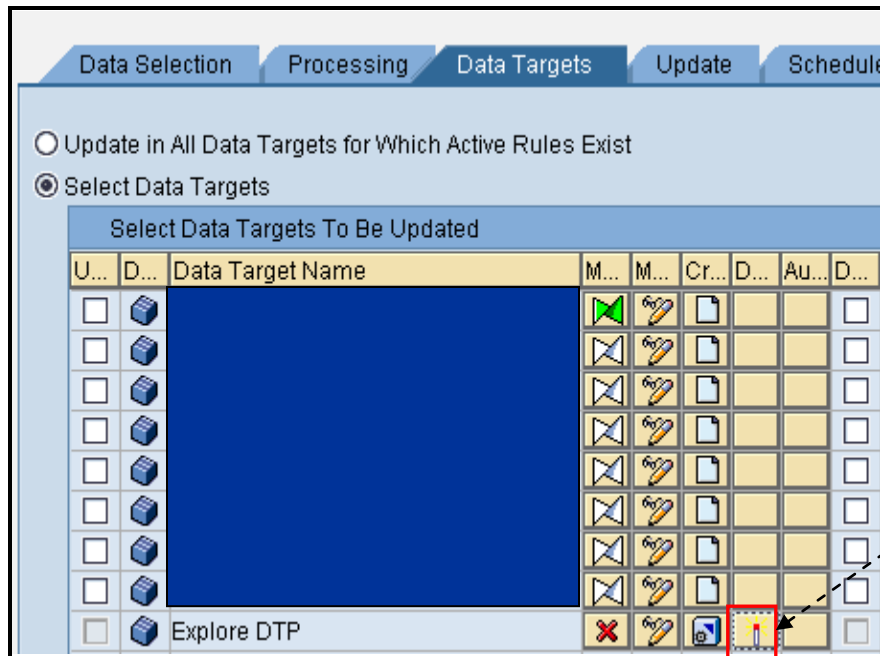


Hence if we create a new cube which needs to get delta from this DSO using BW7 development of DTP's and transformation, it is not able to recognize the deltas.

Contents		Requests		Reconstruction			
Requests from DataStore Object: B03_05_02_03114 Ad card redemptions(ZBB)							
Requ...	R...	ID of ...	Re...	Loa...	Request Number	D...	Log...
38620		38621			DTPR		

This symbol shows that this request is already loaded to target and is not delta enabled for any other target.

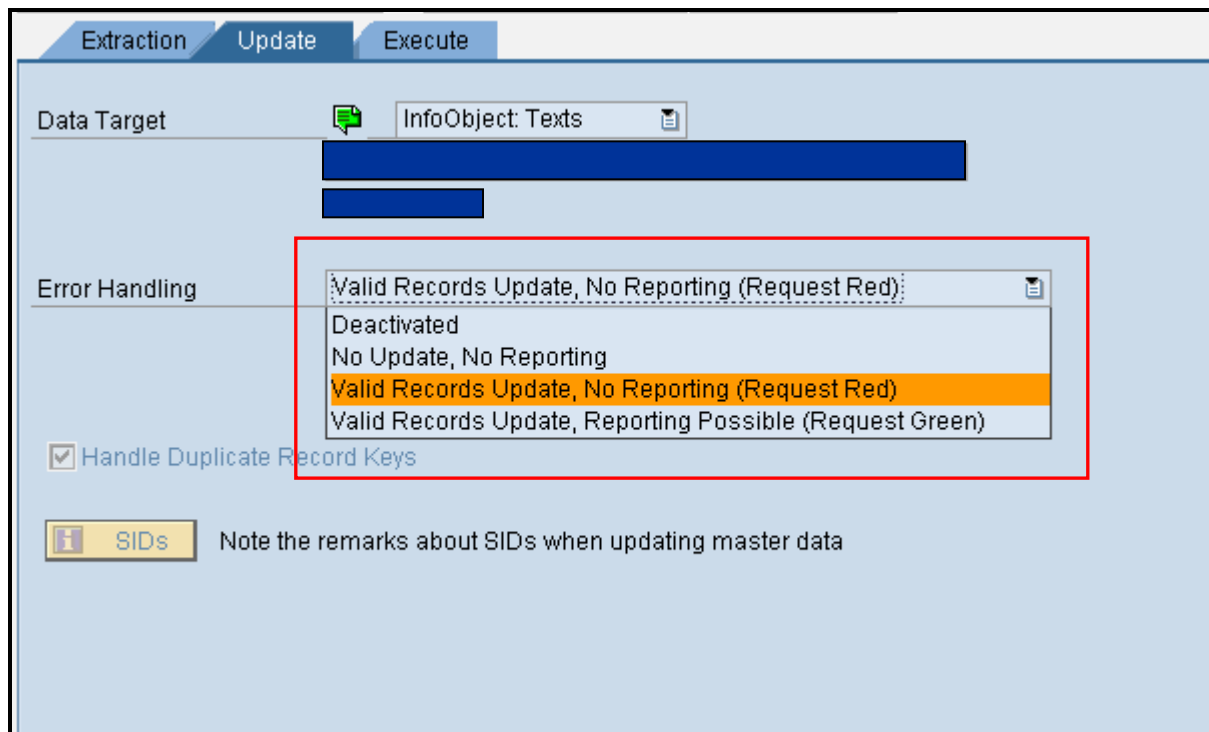
To handle this situation we have a setting in the source DSO info package (Info package of 3.x data flow which is loading data from ODS to Cube) to show that the DTP load is active. After making this setting the cube will be able to detect the delta in the base DSO even after data mart status being set.



This setting is very useful, when we plan to integrate 2 flows one of BW 3.x and BW 7.0.

Error Handling using DTP

Any error records are updated in error stack. Error DTP is then used to update the data to the subsequent data targets.



Options in error handling

- Deactivated

Using this option error stack is not enabled at all. Hence for any failed records no data is written to the error stack. Thus if the data load fails, all the data needs to be reloaded again.

- No update, no reporting

If there is erroneous/incorrect record and we have this option enabled in the DTP, the load stops there with no data written to the error stack. Also this request will not be available for reporting. Correction would mean reloading the entire data again.

- Valid Records Update, No reporting (Request Red)

Using this option all correct data is loaded to the cubes and incorrect data to the error stack. The data will not be available for reporting until the erroneous records are updated and QM status is manually set to green. The erroneous records can be updated using the error DTP.

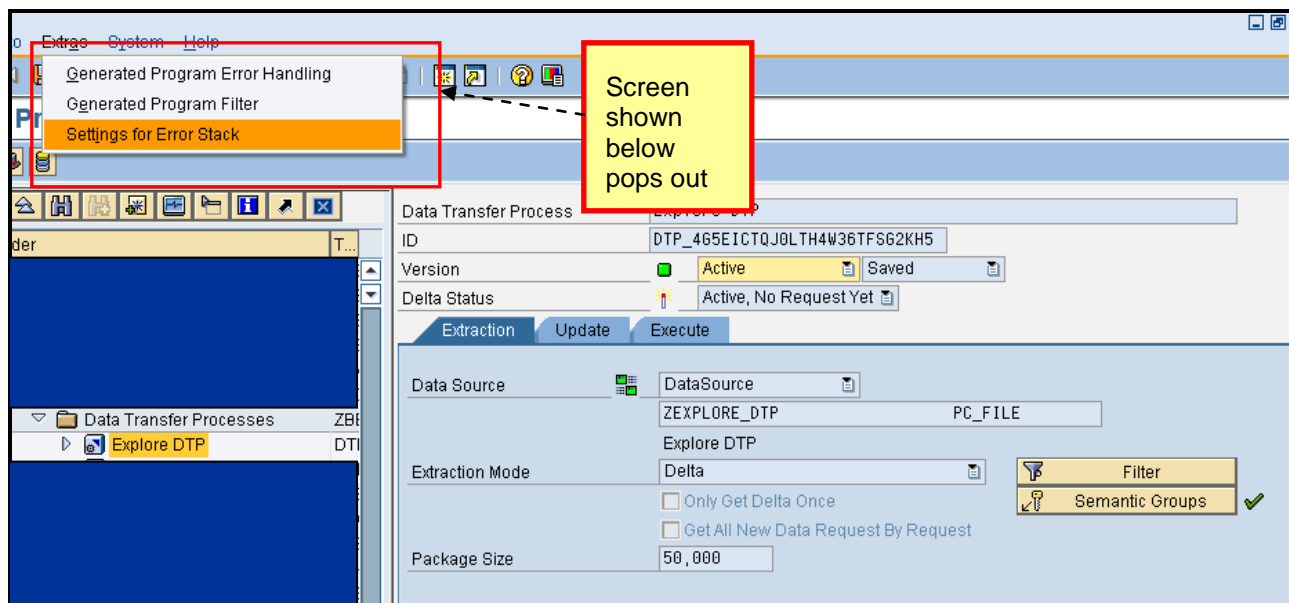
- Valid Records Updated, Reporting Possible (Request Green)

Using this option all correct data is loaded to the cubes and incorrect data to the error stack. The data will be available for reporting and process chains continue with the next steps. The erroneous records can be updated using the error DTP.

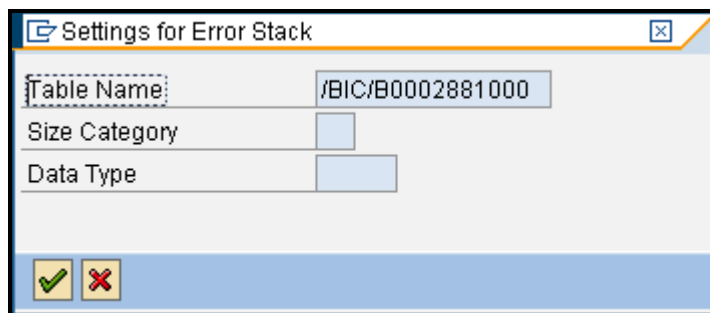
How to find the error Stack Table?

To locate the error stack table navigate as shown below:

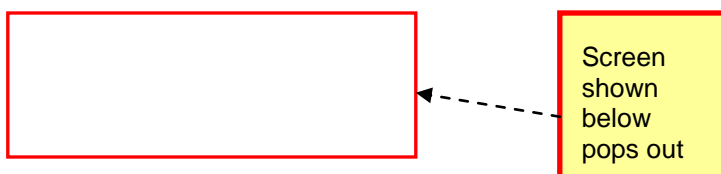
Extras -> Settings for Error Stack



Here the table name gives the name of the table where the error records reside.



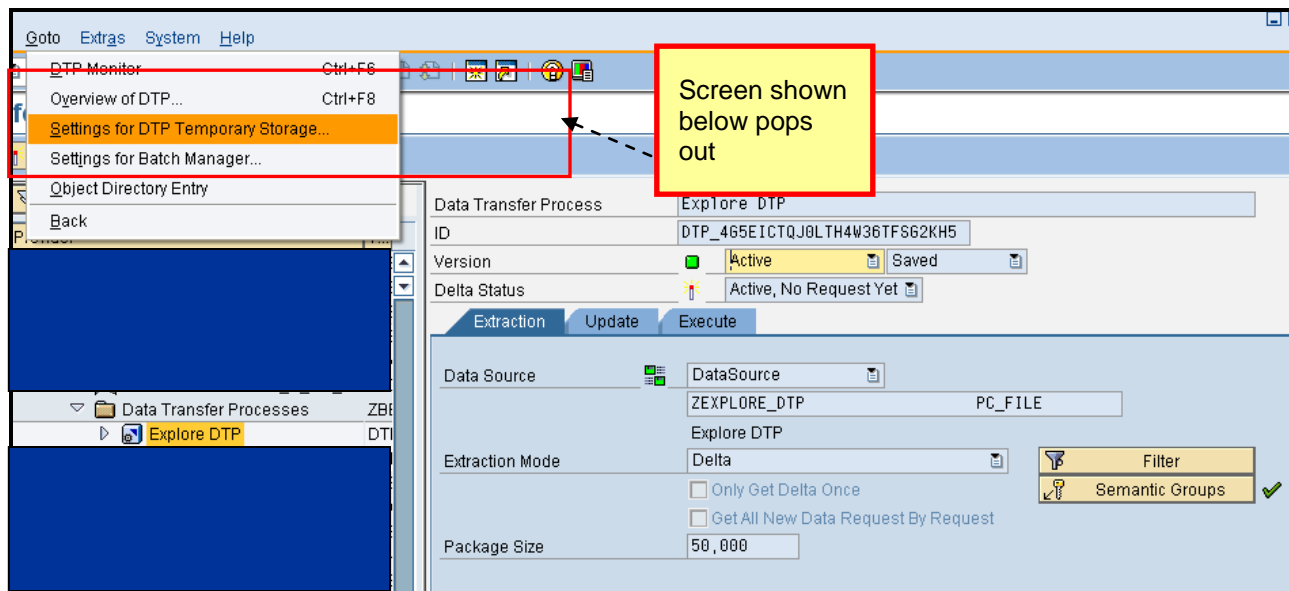
Temporary data storage using DTP



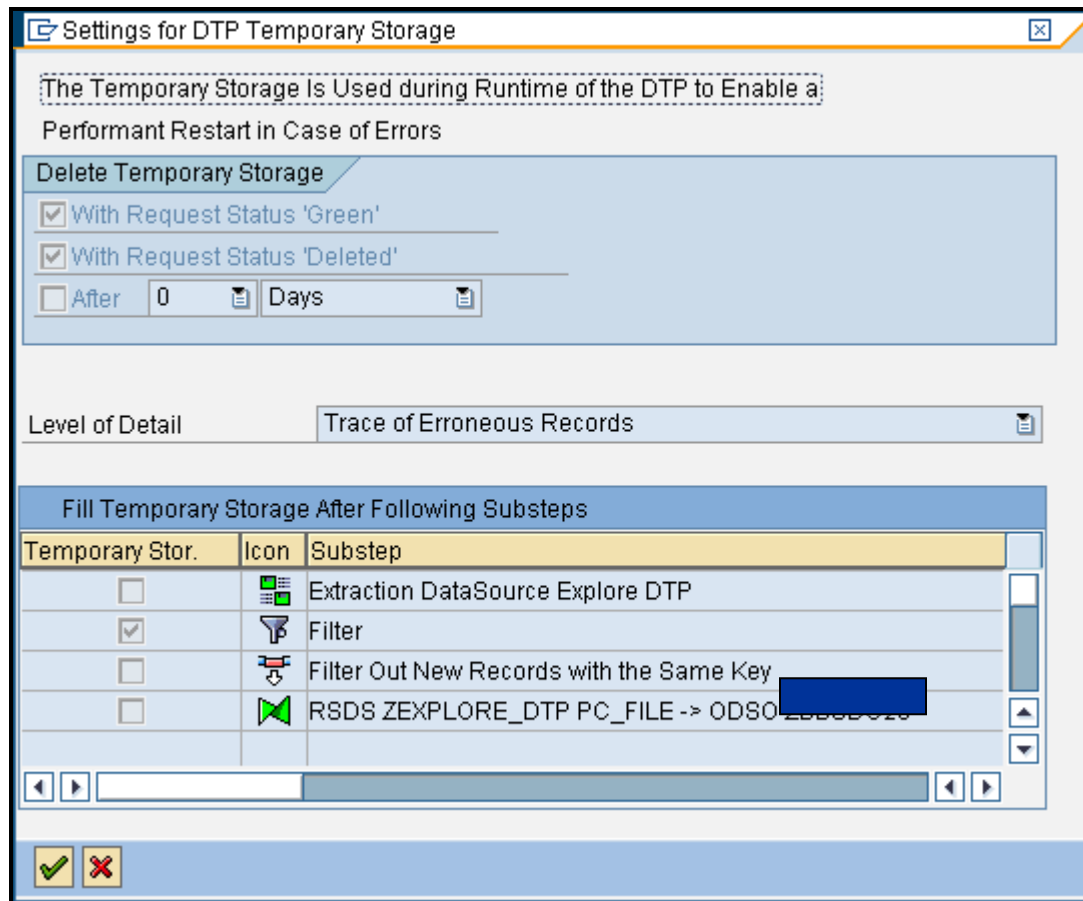
In DTP we can define if we want to store the data temporarily at any stage during the process of data load. It could be before extraction, after transformation etc. It helps in easier analysis of the data. So, it also becomes easier to start the failed load process after this.

To locate this setting, we navigate as shown below:

Goto -> Settings for DTP Temporary Storage



Here we can view the temporary storage settings.



Extraction Modes

The data can be loaded to the data target via two extraction modes.

- Full -> Extraction of data that is requested
- Delta Mode - > Extraction of data new data

Each of these is explained below.

Full

It behaves in the same way as info package does. PSA should be deleted every time before using this option.

The screenshot shows the SAP Data Transfer Process (DTP) configuration window. The 'Extraction' tab is active. The 'Data Source' is 'DataSource' with 'ZEXPLORE_DTP' and 'PC_FILE'. The 'Extraction Mode' is set to 'Full', which is highlighted with a red rectangle. Other fields include 'Package Size' set to '50,000' and a checkbox for 'Do Not Extract from PSA but Access Data Source (for Small Amounts of Data)'. Buttons for 'Filter' and 'Semantic Groups' are visible on the right.

The following sources can have only Full load option.

1. Info Object
2. Info Set
3. Direct Update DSO

Delta

No initialization is required to have a DTP work with delta mechanism. When the DTP is executed with this option for the first time, it brings all the data from the source and also sets the target in such way that it is initialized.

The screenshot shows the SAP Data Transfer Process (DTP) configuration interface. The 'Execute' tab is selected. The 'Data Source' is 'DataSource' with 'ZEXPLORE_DTP' and 'PC_FILE'. The 'Extraction Mode' is set to 'Delta' (highlighted with a red box). Below it are checkboxes for 'Only Get Delta Once' and 'Get All New Data Request By Request'. The 'Package Size' is '50,000'. On the right, there are buttons for 'Filter' and 'Semantic Groups' with a green checkmark.

In the scenarios, where we want to load only the new data, the option “**No data transfer; delta status in source: fetched as processing mode**” in the execute tab of the DTP works in the same way as the init in the info package.

Data Transfer Process	Explore DTP		
ID	DTP_4G5EICTQJ0LTH4W36TFS62KH5		
Version		Active	Saved
Delta Status		Active, No Request Yet	
<div>Extraction</div> <div>Update</div> <div>Execute</div>			
Technical Request Status	Request status is set to 'green' if warnings occur		
Overall Status of Request	Set Overall Status Automatically		
Processing Mode	<div>Serial Extraction, Immediate Parallel Processing</div> <div>Execute</div>		
Program Flow	<div>Serial Extraction, Immediate Parallel Processing</div> <div>Serially in the Dialog Process (for Debugging)</div> <div>No Data Transfer; Delta Status in Source: Fetched</div>		
<div>Explore DTP</div> <div>Start Main Background Process</div> <div>Data Package Loop</div> <div>Extraction DataSource Explore DTP</div> <div>Filter</div> <div>Start Parallel Background Process</div> <div>Filter Out New Records with the Same Key</div> <div>RSDS ZEXPLORE_DTP_BO_FILE - ODSO ZBBER</div> <div>Update to DataStore Object ZBBER000</div>	<div>Change Breakpoints</div> <div>Change Breakpoints</div> <div>Change Breakpoints</div> <div>Change Breakpoints</div> <div>Change Breakpoints</div>		

Processing Mode

These modes detail the steps that are carried out during DTP execution (e.g. Extraction, transformation, transfer etc). Processing mode also depends on the type of source.

The various types of processing modes are shown below:

1. Serial extraction, immediate parallel processing (asynchronous processing)

This option is most used in background processing when used in process chains. It processes the data packages in parallel.

2. Serial in dialog process (for debugging) (synchronous processing)

This option is used if we want to execute the DTP in dialog process and this is primarily used for debugging.

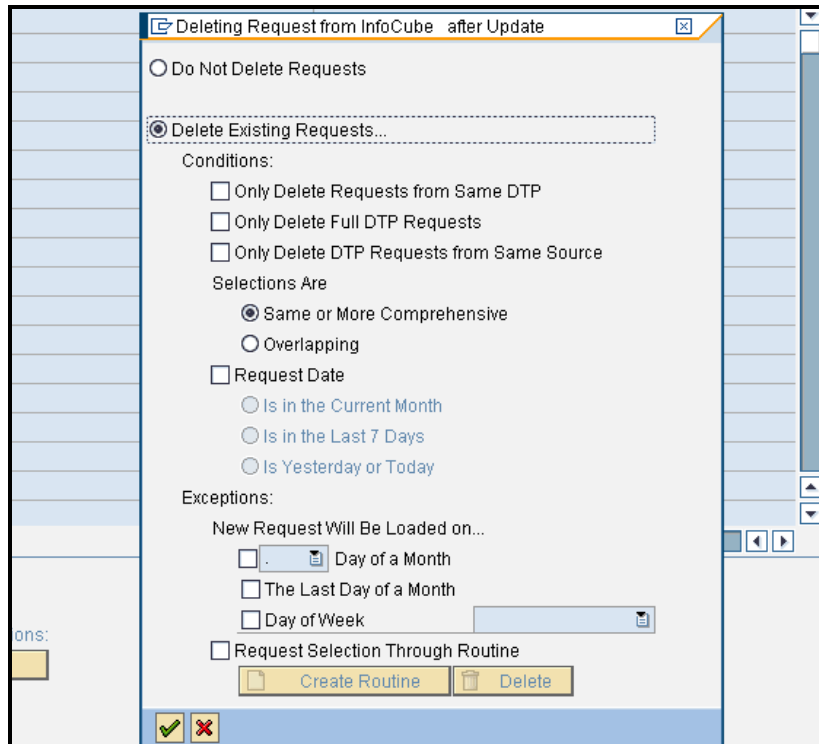
3. No data transfer; delta status in source: fetched

This option behaves exactly in the same way as explained above.

Parameters with Delta Mode

1. **Only Get Delta Once:** This is best used in the scenario where we want to use the snapshot data. E.g. Budget data, if we want to overwrite the existing budgets in the cube, this setting will enable us to delete the existing request from the cube and overwrite with the new request with the same selections.

If we use this setting in the process chains, we need to use the delete overlapping request option along with this.



2. **Get data Request by Request:** If we use this option a scenario of backlog might be created. If 5 requests of the data have been loaded to the PSA, when DTP is executed with the same setting, it just brings the oldest request. This is not advisable to be used in the process chains as it does not load the current data.
3. **Retrieve until No More New Data:** This option is added to the DTP's to enable the DTP's to get all the requests in the source until all the requests are loaded. E.g. if we have 3 requests in the source, disabling this setting will force only one request to get transferred to the target, whereas by enabling this setting we will be able to see 3 separate requests in the target. This option becomes very useful when the data package cannot handle too many records.

Extraction | Update | Execute

Data Source: DataSource
ZEXPLORE_DTP PC_FILE
Explore DTP

Extraction Mode: Delta

☐ Only Get Delta Once
☒ Get All New Data Request By Request
☒ Retrieve Until No More New Data

Package Size: 50,000

Filter
Semantic Groups ✓

Significance of Semantic Groups

This defines the key fields of the data package when read from the source. The same keys applies to the error stack as well e.g. while doing the parallel processing of the time dependant master data, DTP may

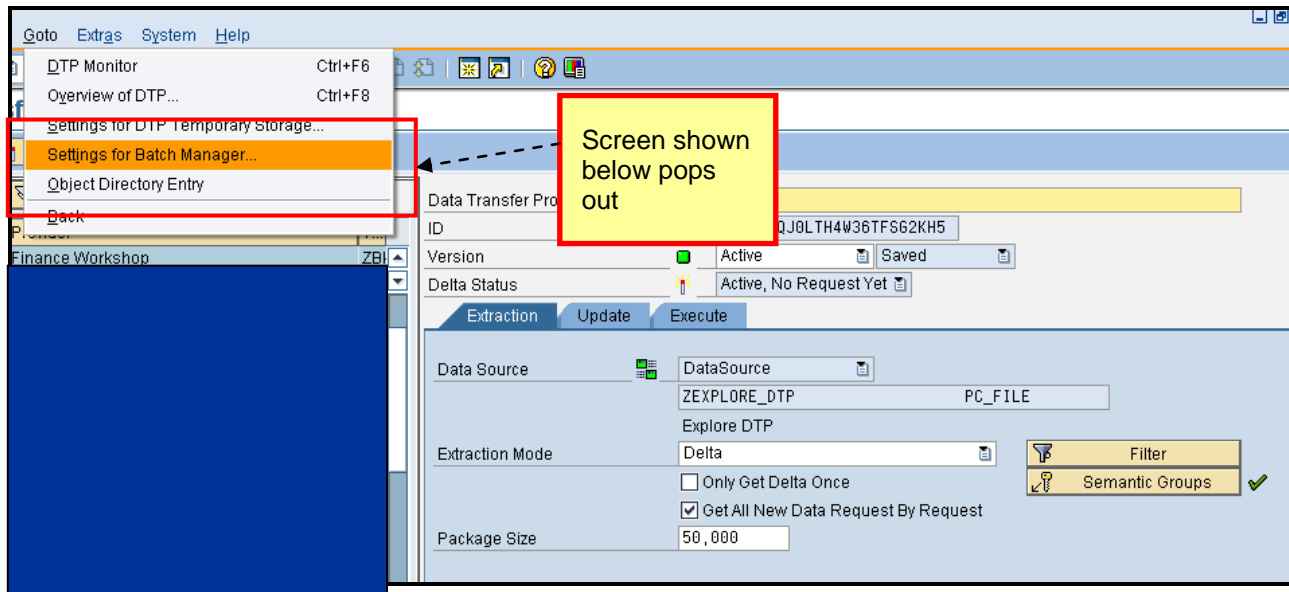
Key Field	Field Name	Long Description
<input type="checkbox"/>	CALDAY	Calendar Day
<input type="checkbox"/>	MATERIAL	Article
<input type="checkbox"/>	PLANT	Plant

contain different settings from the DataSources.

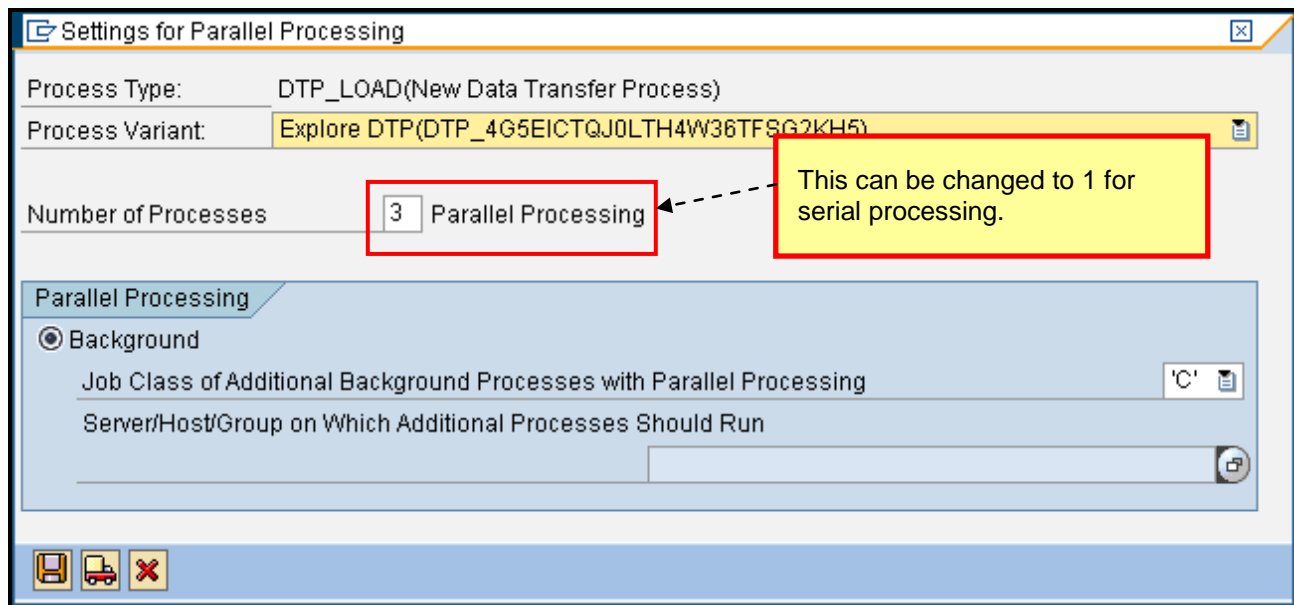
Performance Optimization using DTP

1. Number of parallel process

We can define the number of processes to be used in the DTP by navigating as shown below:



If the number of processes is equal to 1, is serial processing, else parallel.



2. Data Package Size

The default size of the data package is 50000. This can be changed depending on the size of the records.

3. Do not load too large DTP Requests at one go.

This can be achieved by enabling the setting "Get all new data request by request" as detailed above. This facilitates in reduction of the size of each DTP, hence improving the performance.

Extraction Update Execute

Data Source: DataSource
ZEXPLORE_DTP PC_FILE

Explore DTP

Extraction Mode: Delta

☐ Only Get Delta Once

☒ Get All New Data Request By Request

☒ Retrieve Until No More New Data

Package Size: 50,000

Filter

Semantic Groups ✓

4. If we are planning to use DTP filters while loading the data from data source, small data packages can be avoided.

In such scenarios it is best to redefine the semantic keys.

5. If DSO is the source, extraction of data before doing first delta or while full extraction from the active data table is not advisable.

Full load can be best read from active data table, owing to less number of records.

Extraction Update Execute

Data Source: DataStore Object

Extraction Mode: Delta

☐ Only Get Delta Once

☐ Get All New Data Request By Request

Package Size: 50,000

Key Date for Master Data

☐ Currency conversion

Extraction From

☐ Active Table (Full Extract.Only)

☒ DataStore Change Log

Filter

Semantic Groups

6. If we are planning to use Info Cubes as the source to load data, its best to use the option use extraction from aggregates.

If this option is enabled, and if the required data exists in the aggregates, it reads the data from the aggregates before reading it from F table or E table directly.

Extraction Update Execute

Data Source: InfoCube

Extraction Mode: Full

Package Size: 50,000

Key Date for Master Data:

☐ Currency conversion

☒ Use Aggregates

Filter

Semantic Groups

Handling Duplicate Data Records

This functionality is used to handle the duplicate data records which loading the data to the attributes or texts.

Extraction Update Execute

Data Target: InfoObject: Texts

Error Handling: Valid Records Update, No Reporting (Request Red)

Maximum Number of Errors per Pack: 100

Creating Error DTPs

☒ Handle Duplicate Record Keys

SIDs Note the remarks about SIDs when updating master data

This option is not set by default.

It handles the data as shown below:

- Time Independent Objects

The data of the last record resides as it keeps on overwriting the data.

- Time Dependent Objects

In this scenario, it creates various entries of the same object with different validity periods.

Related Content

http://help.sap.com/saphelp_nw70/helpdata/en/42/fa50e40f501a77e10000000a422035/frameset.htm

http://help.sap.com/saphelp_nw2004s/helpdata/en/42/f98e07cc483255e10000000a1553f7/frameset.htm

http://help.sap.com/saphelp_nw2004s/helpdata/en/44/03a25c0bbe025ce10000000a1553f7/frameset.htm

For more information, visit the [Business Intelligence homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.