Data Transfer Process (DTP)

1. The data transfer process makes the transfer processes in the data warehousing layer more transparent. Optimized parallel processing improves the performance of the transfer process (the data transfer process determines the processing mode).

2. To transfer data within BI from one persistent object to another object, in accordance with certain transformations and filters.

3. It replaces the data mart interface and the Info Package.

4. You can use the data transfer process to separate delta processes for different targets and you can use filter options between the persistent objects on various levels.

5. Data transfer processes are used for standard data transfer, for real-time data acquisition, and for accessing data directly.

6. The InfoPackage controls the transfer of data from the source to the entry layer of BI. The data transfer process controls the distribution of data within BI.

7. You use a process chain to define a data transfer process.

8. In process chain maintenance, you can execute a data transfer process in the background. A debug mode is also available.

9. With a data transfer process, you can transfer data either in full extraction mode or in delta mode:
   
   a. In full mode, the entire dataset of the source is transferred to the target; in delta mode, only the data that was posted to the source since the last data transfer is transferred. The data transfer process controls delta handling and therefore allows you to fill several targets with different deltas from one source. With a data transfer process, you do not need to explicitly initialize the delta method as you do when copying data with an InfoPackage.

10. The data transfer process supports you in handling data records with errors (the incorrect data records are sorted and written to an error stack (request-based database table).)

   a. A special error DTP further updates the data records from the error stack into the target. It is easier to restart failed load processes if the data is written to a temporary storage after each processing step. It also allows you to find records that have errors.

**Extraction Mode**

Indicator: Only Get Delta Once

Source requests of a DTP for which this indicator is set are only transferred once, even if the DTP request is deleted in the target.

Use

If this indicator is set for a delta DTP, a snapshot scenario is built. A scenario of this type may be required if you always want an InfoProvider to contain the most up-to-date dataset for a query but the DataSource on which it is based cannot deliver a delta (new, changed or deleted data records) for technical reasons. For this type of DataSource, the current dataset for the required selection can only be transferred using a ‘full update’. In this case, a DataStore object cannot usually be used to determine the missing delta information (overwrite and creation of delta). If this is not logically possible because, for example, data is deleted in the source without delivering reverse records, you can set this indicator and perform a snapshot scenario. Only the most up-to-date request for the DataSource is retained in the InfoProvider. Earlier requests for the DataSource are deleted from the (target) InfoProvider before a new one is requested (this is done by a process in a process chain, for example). They are not transferred again by the DTP delta process. When the system determines the delta when a new DTP request is generated, these earlier (source) requests are considered to have been retrieved.

Setting this indicator ensures that the content of the InfoProvider is an exact representation of the source data.

 Dependencies

Requests that need to be fetched appear with this indicator in the where-used list of the PSA request, even if they have been deleted. Instead of a traffic light you have a delete indicator.

Indicator: Retrieve Until No More New Data

To prevent requests from becoming too large, delta data transfer processes (DTPs) can get data request by request. Using the Get All New Data Request By RequestIndicator, you specified that a DTP request is always to contain just one source request. However, this means that for this DTP, only the first of the new requests is retrieved, even if there is more than once source request at the time of processing. This restricts the way in which these DTPs can be used in process chains, because requests accumulate in the source and the target may not contain the current data.

Use

If you set the Retrieve Until No More New Data indicator and then activate the DTP, once it completes processing, a DTP request checks whether there are any further requests in the source. If the source contains more requests, a new DTP request is automatically generated and processed. Once the DTP is activated, the indicator is no longer visible in DTP maintenance.

This applies when the DTP is started by a process chain and also when you start the DTP directly from DTP maintenance. The indicator is set by default for new DTPs that get data request by request.

If you do not select the indicator, the label for the Get All New Data Request By RequestIndicator changes to Get One Request Only. Once the DTP is activated, only this indicator is displayed in DTP maintenance.

**Type of Error Handling During DTP Load Processes**

Definition

The following settings are possible: Switched off! If an error occurs, the error is reported as a package error in the DTP monitor. The error is not assigned to the data record. The cross-reference tables for determining the data record numbers are not built; this results in faster processing. The incorrect records are not written to the error stack since the request is terminated and has to be updated again in its entirety. No update, no reporting (default) If errors occur, the system terminates the update of the entire data package. The request is not released for reporting. The incorrect record is highlighted so that the error can be assigned to the data.
record. The incorrect records are not written to the error stack since the request is terminated and has to be updated again in its entirety. Update valid records, no reporting (request red) This option allows you to update valid data. This data is only released for reporting after the administrator checks the incorrect records that are not updated and manually releases the request (by a QM action, that is, setting the overall status on the Status tab page in the monitor). The incorrect records are written to a separate error stack in which the records are edited and can be updated manually using an error DTP. Update valid records, reporting possible Valid records can be reported immediately. Automatic follow-up actions, such as adjusting the aggregates, are also carried out. The incorrect records are written to a separate error stack in which the records are edited and can be updated manually using an error DTP.

**RSODSOCHECKONLY**

If you set this indicator, the system terminates the update of the request if no values are available for a data record.

Load the relevant master data before you load the transaction data.

**Dependencies**

**Transformation:**

- If you select rule type *Read Master Data* in the related transformation, the following applies:
  - If you set this indicator, records that do not have values are skipped.
  - If you do not set this indicator, the system writes initial values. The update is not terminated.

**DataStore Objects**

- In DataStore object maintenance, if you set the SIDs Generation upon Activation indicator, the following applies:
  - If you set this indicator, the system terminates activation if master data is missing and produces an error message.
  - If you do not set this indicator, the system generates any missing SID values during activation.

- In DataStore maintenance, if you do not set the SIDs Generation upon Activation indicator, the No Update without Master Data indicator in the DTP has no effect.

**DTP Request: Behavior If There Are Warnings in the Log**

This parameter defines the behavior of a request generated by the current data transfer process if warnings have been logged during processing.

- The two following settings are possible if warnings occur:
  - Technical request status is set to Green
  - Technical request status is set to Red

**DTP Request: Overall Status Behavior**

This parameter determines the behavior of a request generated by the current data transfer process when the technical part of processing is finished and the overall status is to be set. The parameter enables two settings:

- Set overall status automatically
  - With this setting, if the request is finished with the technical status red or green, the overall status is automatically set to the same as the technical status.

- Set overall status manually
  - With this setting, if processing is finished with the technical status red or green, the overall status remains the same. In particular, this means that data for a green request is not released for reporting or further processing. The overall status has to be set manually by the user or by a process in a process chain.

**Dependencies**

- If a third-party tool is acting as an open hub destination, the overall status of the request is set by the third-party tool, using a BAPI. Therefore you cannot set this parameter to Set Overall Status Automatically in this case.

**Processing Mode**

**Definition**

The processing mode describes the order in which processing steps such as extraction, transformation and transfer to target are processed at runtime of a DTP request. The processing mode also determines when parallel processes are to be separated.

The processing mode of a request is based on whether the request is processed asynchronously, synchronously or in real-time mode, and on the type of the source object.

- It also takes the number of parallel processes defined in BI background management into consideration. If only one process is defined there, no additional processes are separated while the DTP request is being processed.
- Serial extraction, immediate parallel processing (asynchronous processing)
- A request is processed asynchronously in a background process when a DTP is started in a process chain or a request for real-time data acquisition is updated.
- The processing mode is based on the source type.
- Serial in dialog process (for debugging) (synchronous processing)
- A request is processed synchronously in a dialog process when it is started in debug mode from DTP maintenance.

You cannot start requests for real-time data acquisition in debug mode.

- No data transfer; delta status in source: fetched
- Processing mode for real-time data packages

A request is processed serially in real-time data acquisition mode if it was started from a daemon. In this case the data packages are processed one after the other by a background process, the daemon.

**Example**

- Domain fixed values:
  - Serial in the background process
  - Serial in the dialog process
  - Serial extraction, immediate parallel processing
  - Parallel extraction and processing
  - Serial in the dialog process (for debugging)
  - Processing mode for real-time data packages
  - Processing mode for direct access
  - Serial extraction and processing of the source packages
  - No data transfer; delta status in source: fetched

**Semantic Grouping Possible**

An extractor has this property if it can return data for a grouping key defined in the DTP package by package to the caller as a semantic unit. The semantic grouping is possible for the following sources: Data Source, Data Store object and Info Cube.
Grouping Key and Grouping Mode

Grouping Key
The grouping key is the subset of the source fields defined in the DTP for the semantic grouping (tab page Extraction pushbutton Semantic Groups). It defines how the data packages that are read from the source (DataSource, DataStore object or InfoCube) are created. The data records for a grouping key are combined into one data package. The grouping key is also the key for the error stack of the DTP.

The grouping key for the source depends on whether error handling is activated for the DTP and whether the transformations called within the DTP and the target require semantically grouped data:

Depending on error handling
If error handling is activated, grouping is required in order to define the key fields for the error stack. This is relevant for DataStore objects with data fields that are overwritten. The target key represents the error stack key for targets in which the order of the updated data is of no importance (such as additive delta in InfoCubes); it is marked as the grouping key in the DTP.

Depending on transformation and target
The example below shows how the transformation and target of a DTP influence the grouping key:

Update from a DataSource that can provide the stock prices accurately to the minute into a DataStore object in which the prices at the end of the day are kept for a given security identity number.

In this example, the transformation between the DataSource and the DataStore object has the task of copying the last stock price of the day to the target and filtering out all other prices. To do this, all values for a given security identity number and date are provided for the exact minute in a package. The grouping key here would be the security identity number and the calendar date.

Grouping Modes
The grouping mode defines whether a semantic grouping is required and whether a grouping key exists in the DTP. As explained, grouping is required if error handling is activated. The following grouping modes are possible:

Case 1: No grouping is required; the grouping key includes all the fields of the source.
Case 2: Grouping is required. There is a grouping key that does not include all the fields of the source.
Case 3: Grouping is required. The grouping key does not contain any fields. This corresponds to an empty set.