SAP First Guidance

SAP NetWever BW 7.30
SAP HANA™ Appliance

SAP First Guidance – SAP Netweaver BW 7.30 on HANA Inventory InfoCubes

Applicable Releases:

SAP BW 7.30*powered by SAP HANA

This First Guidance Document is the beginning of a series of documents that should help to better understand the various concepts of SAP BW powered by SAP HANA.
The documents are still “work in progress”, so these guides are not intended to be exhaustive so far, and might never be. The purpose of these documents is to deliver additional information besides SAP Help and Blogs to get a better understanding of the concepts of SAP BW on HANA.

For more information or feedback please contact: Stefan.Hoffmann@sap.com.

Version 1.6

November 2013

Document History

Document Version	Description
1.00	First draft of this guide
1.10	Removed 'Internal Only'
1.20	Removed reference to note 584158
1.30	Added guideline for migration and conversion (notes 1766577, 1780575)
1.31	Changed description of picture 22
1.5	Reworked section extraction from ECC
1.6	Added in section extraction

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Introduction 2

2. Scenario 2
   - 2.1 Data model Inventory InfoCube 3
   - 2.2 Transformation 3
   - 2.3 Source table and extractor 4
   - 2.4 Database structures and their initial views 5

3. Data Staging 6
   - 3.1 Initialization 6
   - 3.2 Historical Transactions 8
   - 3.3 Delta transactions 11

4. Contents of the fact table 13

5. Differences between HANA based version and non-Hana based version of Inventory data staging 13
   - 5.1 Logic of 'No marker update' pushed down 13
   - 5.2 No 'E fact table' any longer 14

6. Query 14

7. Influence of InfoCube Compression 18

8. Extraction from SAP ERP 21

9. Considerations regarding conversion of a non In Memory Optimized Inventory InfoCube into a In Memory optimized Inventory InfoCube 38
   - 9.1 Single Node Systems 38
   - 9.2 Multi Node System 39

10. Conclusions / Recommendations 39

11. Appendix 40
   - 11.1 Query logic 40
   - 11.2 Query examples 42
   - 11.3 Notes 44

1. Introduction

As the handling of Inventory InfoCubes changed within SAP BW 7.30 based on SAP HANA, this document wants to briefly describe the differences.

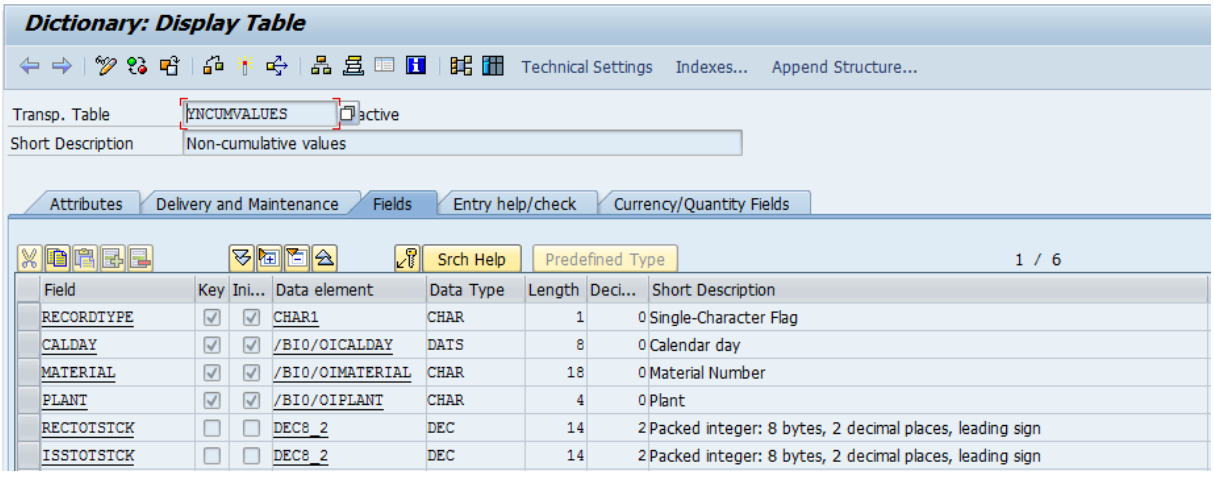
In a first section the document describes the data model of the InfoCube. As there is already a How to Guide based on previous release in SDN (http://scn.sap.com/docs/DOC-15987), the reader is referred to the 'old' How To Guide, as the majority of the content is still valid.

Furthermore we recommend to read through note 1548125 (http://service.sap.com/sap/support/notes/1548125) as this gives explains very well the handling of non-cumulative InfoCubes in SAP BW.

2. Scenario

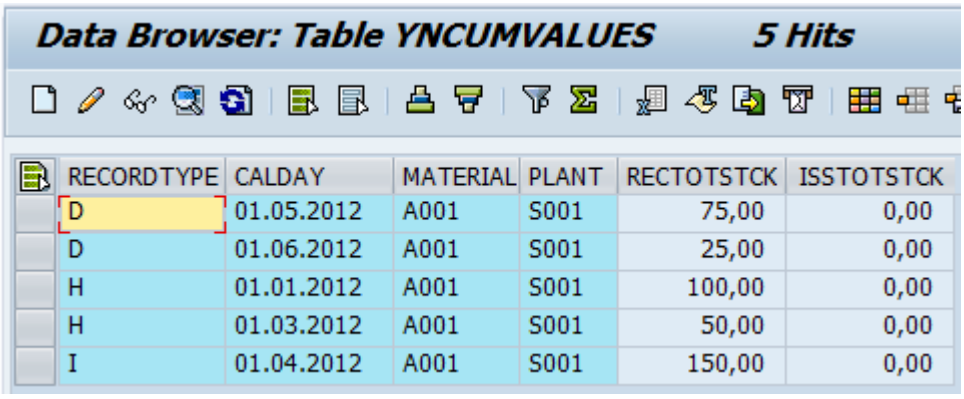
The basis for the explanation in this First Guidance document is an In Memory optimized InfoCube (technical name ZDGRNCUM1), which gets populated by extracting data from a flat table.
In non-cumulative InfoCubes SAP BW on HANA distinguishes between Initialization, Historic and Delta records. To explain you the concepts what happens behind the scenes, we are using a transparent table from which we extract data and run the various DTPs to get transactional data into the In Memory optimized InfoCube.

In the flat table (see Picture 1) we use a field called 'Recordtype' to differentiate the record types. Recordtype = 'I' is used to identify records valid for initialization, Recordtype ='H' stands for records valid for historic movements, Recordtype = 'D' stands for movements of type delta. This table is the basis for the whole explanation how inventory values are handled within SAP BW on HANA.



Field	Key	Ini...	Data element	Data Type	Length	Deci...	Short Description
RECORDTYPE	☑	☑	CHAR1	CHAR	1	0	Single-Character Flag
CALDAY	☑	☑	/BI0/OICALDAY	DATS	8	0	Calendar day
MATERIAL	☑	☑	/BI0/OIMATERIAL	CHAR	18	0	Material Number
PLANT	☑	☑	/BI0/OIPLANT	CHAR	4	0	Plant
RECTOTSTCK	☐	☐	DEC8_2	DEC	14	2	Packed integer: 8 bytes, 2 decimal places, leading sign
ISSTOTSTCK	☐	☐	DEC8_2	DEC	14	2	Packed integer: 8 bytes, 2 decimal places, leading sign

Picture 1: Definition of source of Inventory records



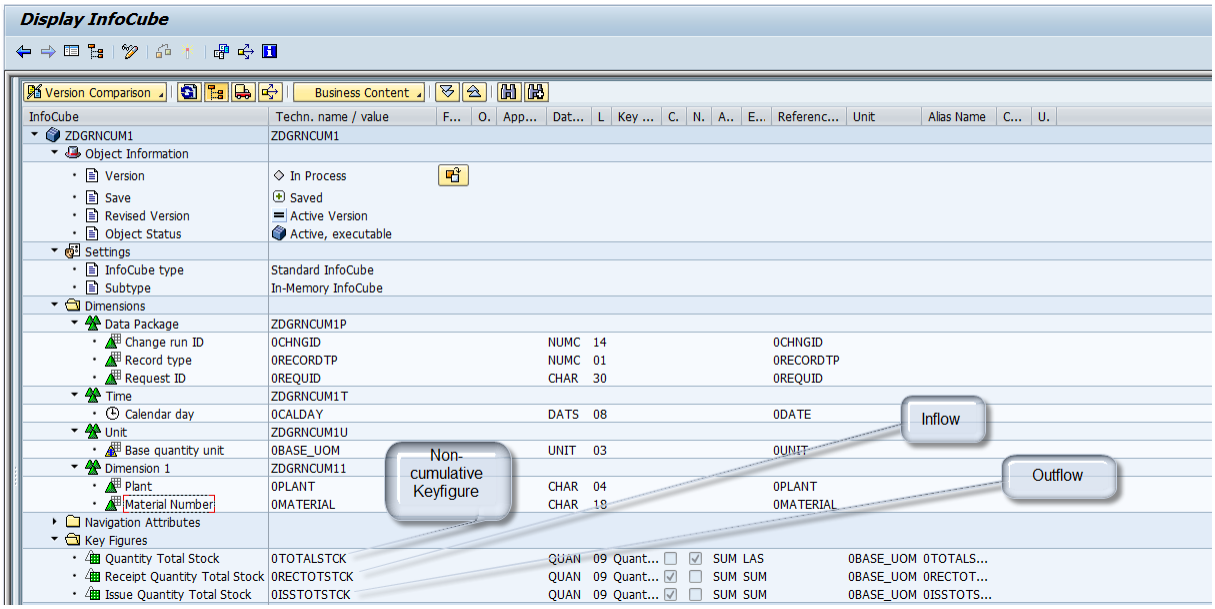
RECORDTYPE	CALDAY	MATERIAL	PLANT	RECTOTSTCK	ISSTOTSTCK
D	01.05.2012	A001	S001	75,00	0,00
D	01.06.2012	A001	S001	25,00	0,00
H	01.01.2012	A001	S001	100,00	0,00
H	01.03.2012	A001	S001	50,00	0,00
I	01.04.2012	A001	S001	150,00	0,00

Picture 2: Table contents of source table

2.1 Data model Inventory InfoCube

Picture 3 shows you the data model of the In Memory Optimized InfoCube ZDGRNCUM1. As you can see there is a non-cumulative keyfigure Quantity Total Stock (0TOTALSTCK) which uses exception aggregation 'LAST'.

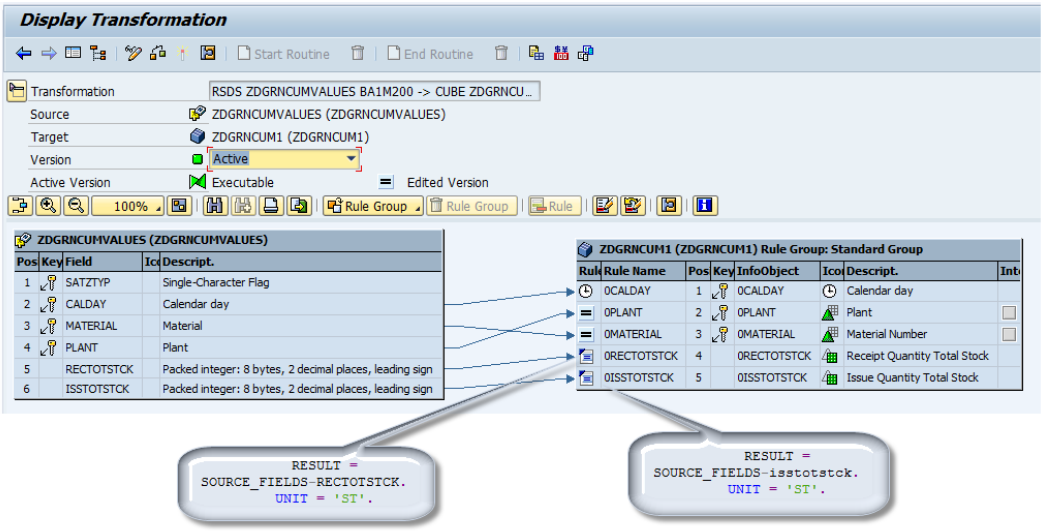
The value of this keyfigure gets calculated by the sum of Inflows and Outflows. In our example Inflows are stored in the relevant field of Receipt Quantity Total Stock (0RECTOTSTCK), while Outflows are stored in Issue Quantity Total Stock (0ISSTOTSTCK). To make the example easy to understand, we have mapped the source fields to the target fields within the transformation without applying complex logic (see Picture 4). In real life, there are in most cases more complex algorithms. In retail for example the inflow and outflow keyfigures are only populated, if certain process keys and movement types are fulfilled.



Picture 3: Data model of Inventory InfoCube

2.2 Transformation

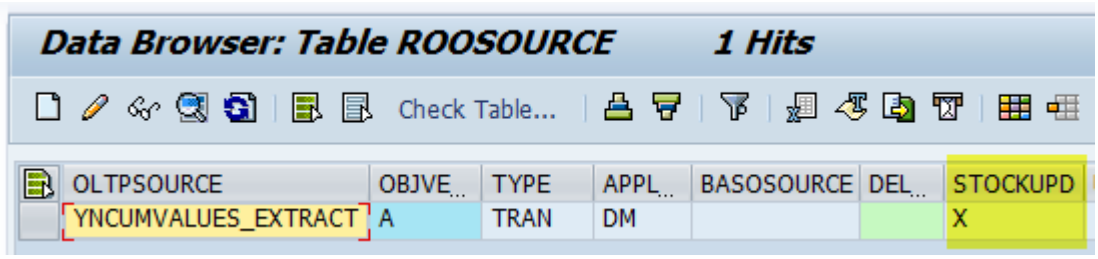
To get the structure of the In Memory InfoCube populated we mapped the source fields to the InfoCube fields using the transformation seen in Picture 4.



Picture 4: Transformation

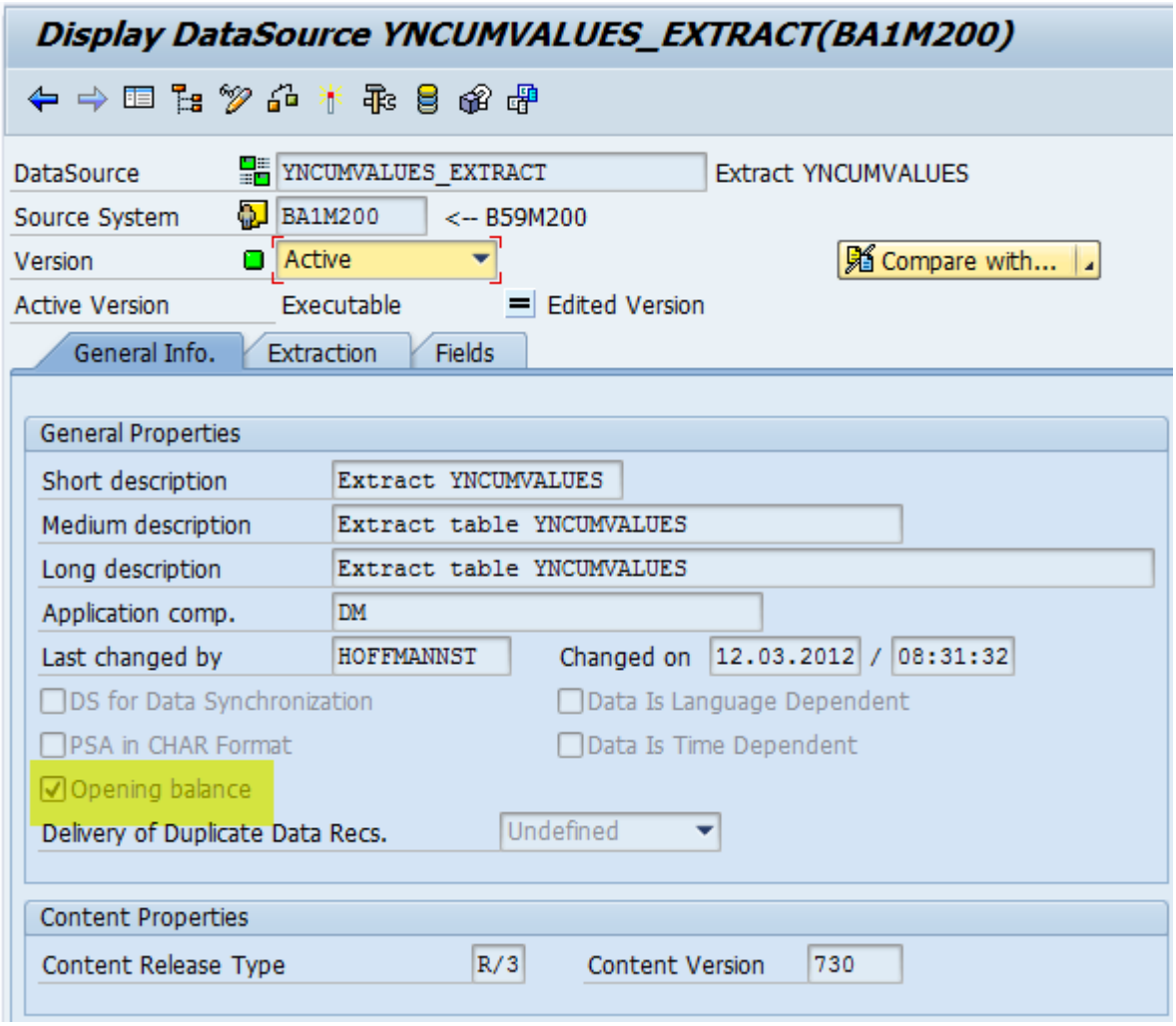
2.3 Source table and extractor

To enable extraction from our simplified source table, we created a generic extractor and set the flag for stock update in table according ROOSOURCE (seePicture 5).



Picture 5: Stockupd = ‚X‘ set in table ROOSOURCE

After replication to BW we get the following datasource (see Picture 6).

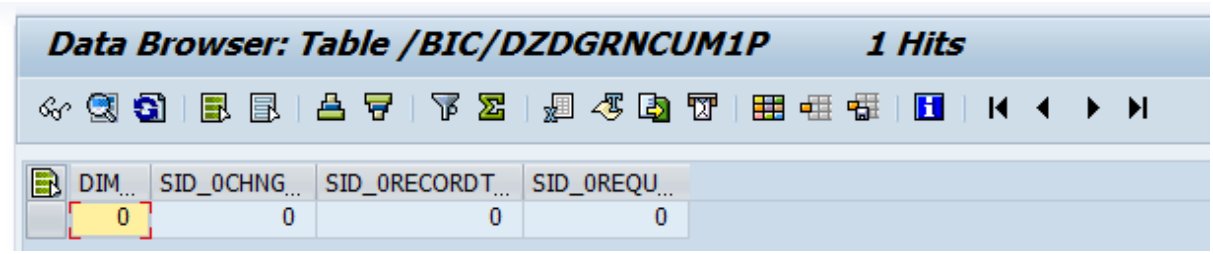


Picture 6: Datasource

2.4 Database structures and their initial views

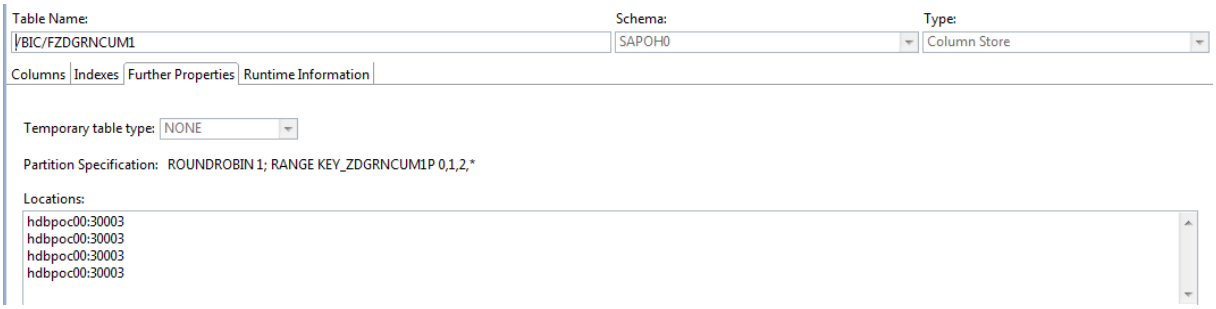
Before we now start to populate the In Memory optimized InfoCube with data let us show you how the database structures look like after we have activated the InfoCube.

Within the package dimension (see Picture 7) we find as supposed the initial entry.



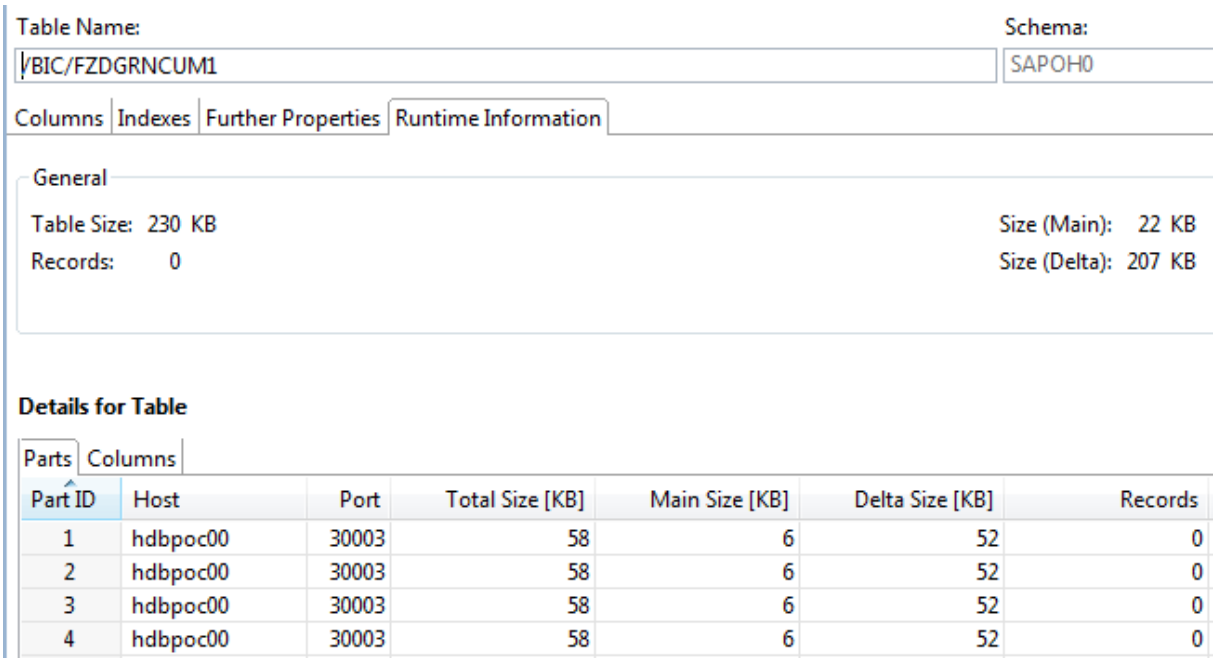
Picture 7: Package dimension (initial view)

As the InfoCube is partitioned, let us have a look to the partition specification by opening a HANA studio view (see Picture 8).



Picture 8: Partition Spec (HANA Studio)

As you can see by choosing Runtime Information, our InfoCube has 4 partitions.



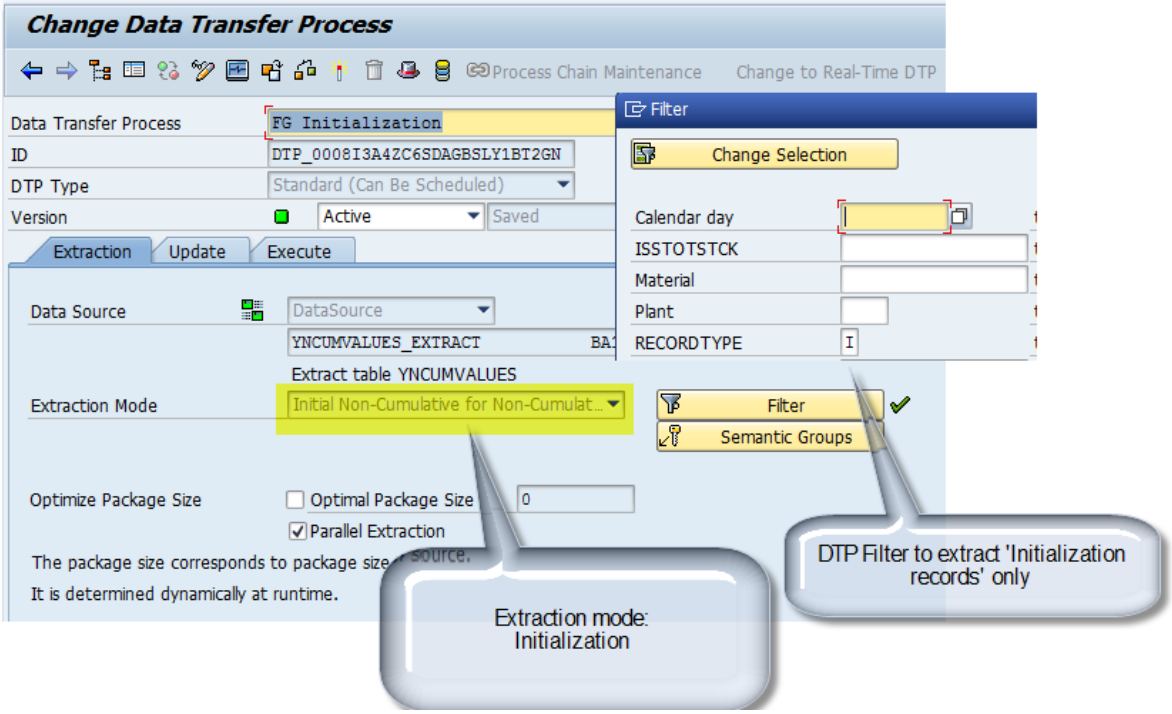
Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	58	6	52	0

Picture 9: Runtime Information before data staging

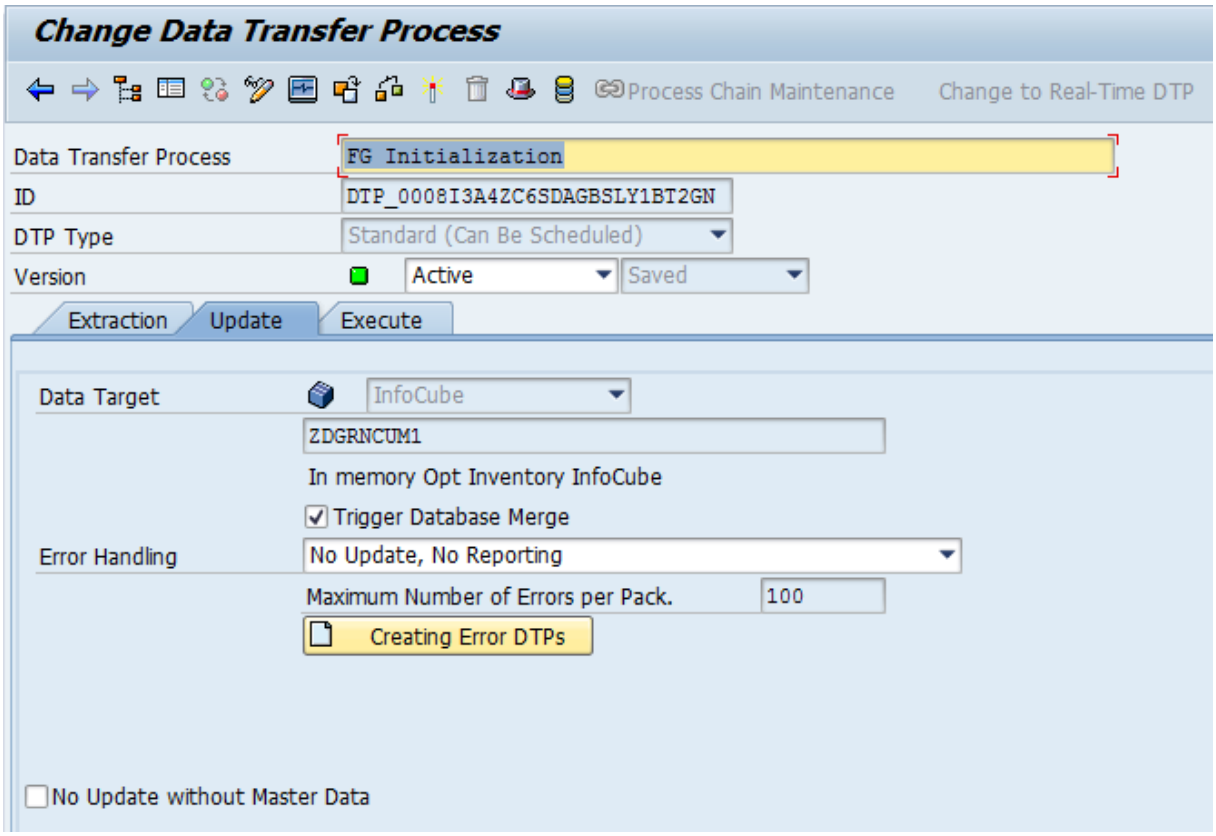
3. Data Staging

3.1 Initialization

To initialize the In Memory Optimized InfoCube, we use a DTP (see Picture 10 and Picture 11), which is using extraction mode Initialization. In our underlying source table we can filter which records are to be extracted just for the Initialization. That's why we also use a filter in our DTP. Be aware, this is just for simplification reasons (just for better explanation). In case you extract from ERP, please consider the relevant chapter 'Extraction from SAP ERP'.



Picture 10: DTP Initialization (Extraction Tab)



Picture 11: DTP Initialization (Update tab)

If we now schedule the DTP and look what happens behind the scenes we get the following pictures.

As you can see by having a look to the package dimension of our InfoCube (see Picture 12), by execution of the DTP we created an entry DIMID = '20', which correlates to field SID_0RECORDTP = '1'. The comparison between Picture 12 and Picture 7 shows also that additional entries had been created (DIMID = '1', '2') in the package dimension.

Data Browser: Table /BIC/DZDGRNCUM1P 4 Hits

DIMI...	SID_0CHNGID	SID_0RECORDTP	SID_0REQUID
0	0	0	0
1	0	1	0
2	0	2	0
20	0	1	3.620.664

Picture 12: Package dimension after execution of Initialization DTP

You might ask, why that?

Well, that's a basic concept of the non-cumulative handling within SAP BW on HANA. The non-cumulative InfoCubes differentiate between Initialization records (stored with SID_0RECORDTP = '1'), Historic Movements (stored with SID_0RECORDTP = '2') and Deltas (stored with SID_0RECORDTP = '0') by choosing the corresponding DIMIDs to those identifiers.

Let's have a look what happened on HDB level by choosing runtime information view of our fact table (see Picture 13).

Table Name: /BIC/FZDGRNCUM1 Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 234 KB Size (Main): 22 KB

Records: 1 Size (Delta): 211 KB

Details for Table

Parts | Columns

Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	61	6	56	1

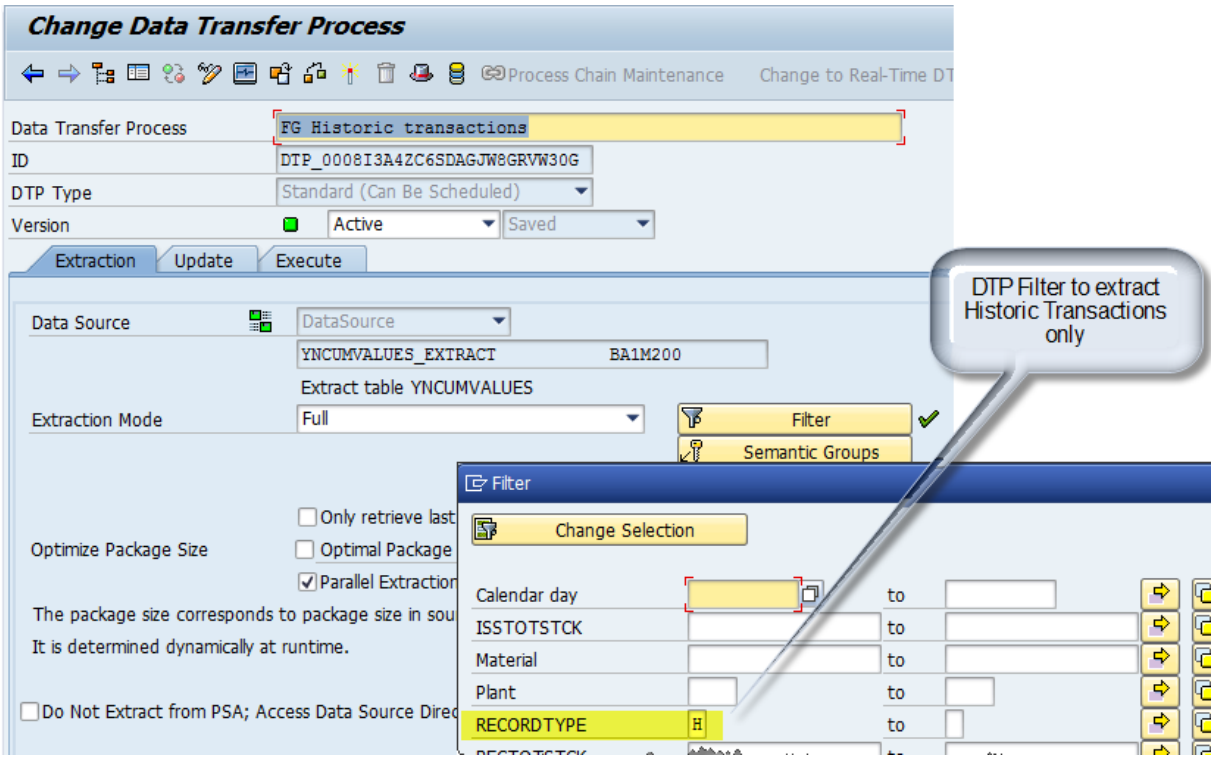
Picture 13: Runtime Information after execution of Initialization DTP

As you can see our initialization record of table YNCUMVALUES (Picture 2) was added to the '*' Partition (Part ID 4).

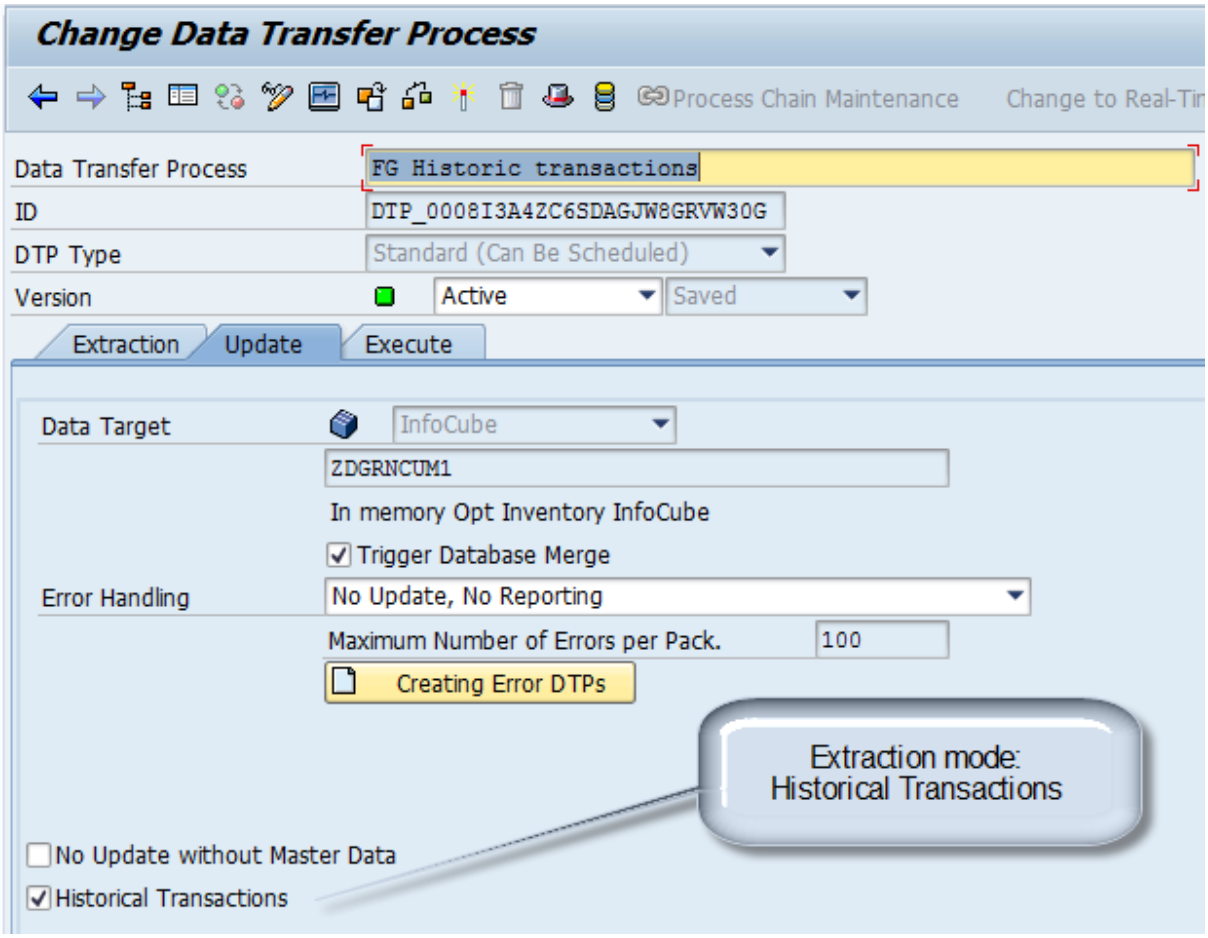
3.2 Historical Transactions

To transfer historical movement records to the In Memory Optimized InfoCube, we use a second DTP (see Picture 14). As you can see in Picture 14 we filter again only those records which are historical records (see Picture 2). By setting the flag for Historical Transactions in the DTP (see Picture 15), we ensure that SAP BW understands that the records we transfer are only Historical

Transactions. That's the major difference to the handling in SAP BW releases on traditional databases, as you there have to set flags whether you want to update a marker or not.



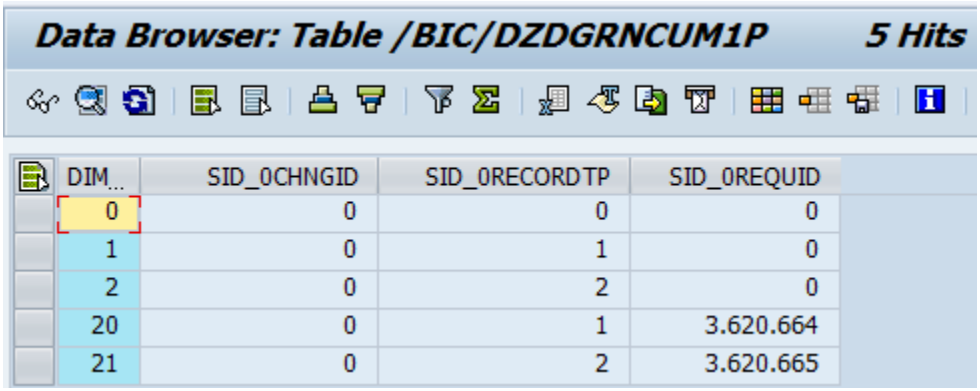
Picture 14: DTP Historical Transactions (Extraction Tab)



Picture 15: DTP Historical Transactions (Update tab)

Again, when executing the DTP we transfer the 2 historical transactions from the source table to the InfoCube (see Error! Reference source not found.)

As you can see by having a look to the package dimension of our InfoCube (see Picture 16), by execution of the DTP we created an entry DIMID = '21', which correlates to field SID_0RECORDTP = '2'.



DIM...	SID_0CHNGID	SID_0RECORDTP	SID_0REQUID
0	0	0	0
1	0	1	0
2	0	2	0
20	0	1	3.620.664
21	0	2	3.620.665

Picture 16: Package dimension after execution of Historical Transactions DTP

Let's have a look what happened on HDB level by choosing runtime information view of our fact table (see Picture 17).

Table Name: /BIC/FZDGRNCUM1

Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 234 KB

Records: 3

Size (Main): 22 KB

Size (Delta): 211 KB

Details for Table

Parts | Columns

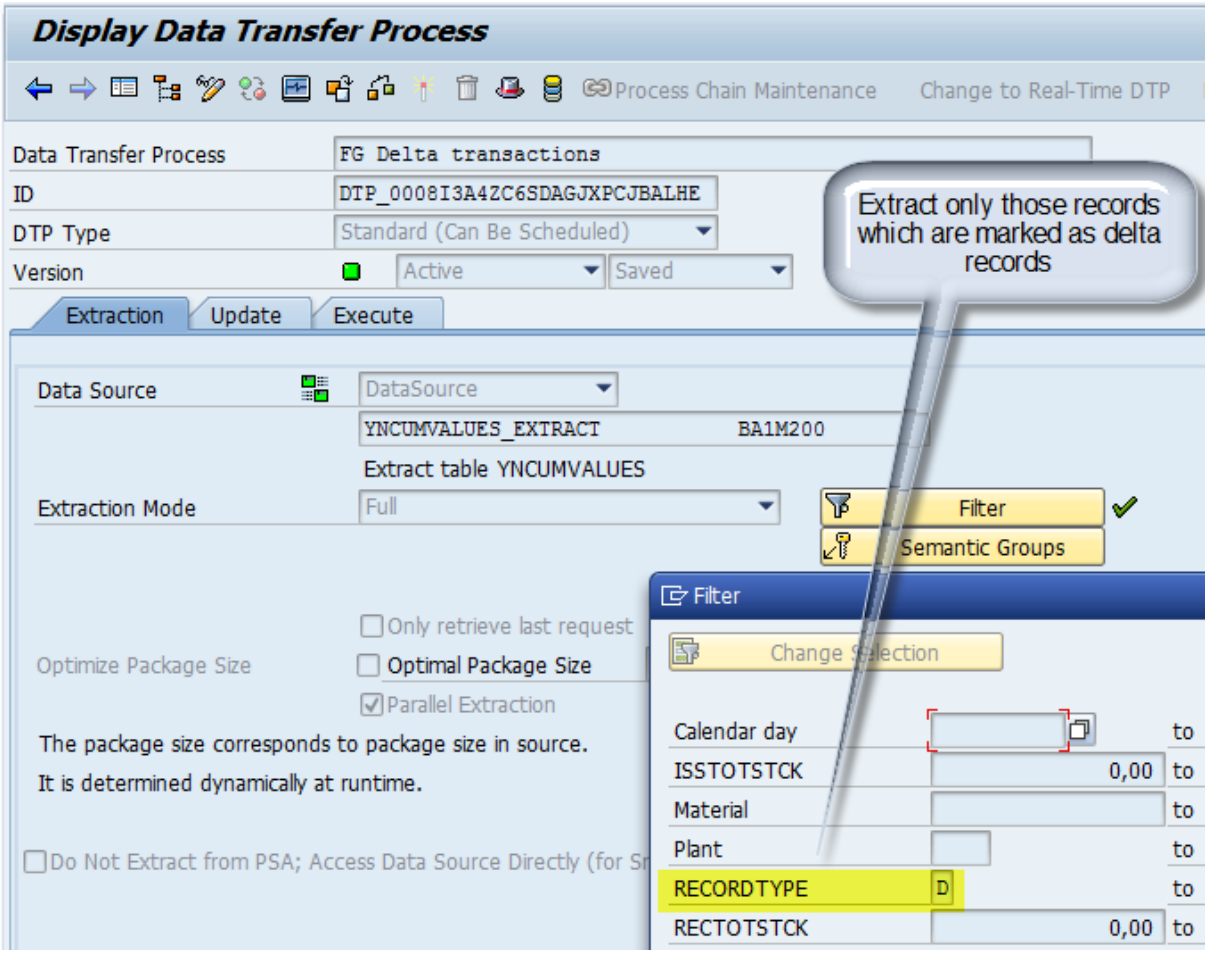
Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	62	6	56	3

Picture 17: Runtime information after execution of Historical Transactions DTP

As you can see our historical records of table YNCUMVALUES (Picture 2) had been added again to the '*' Partition (Part ID 4).

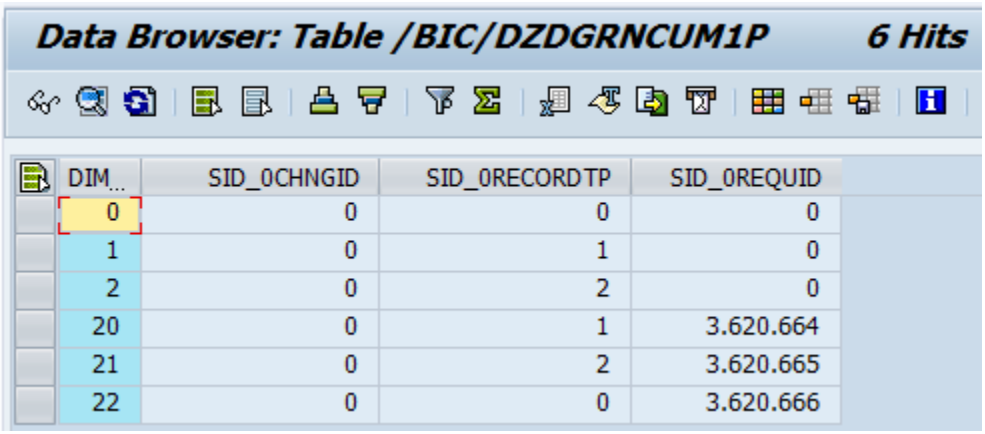
3.3 Delta transactions

Again, we have created a DTP which can be used to extract delta records (see Picture 18) which extracts only those values we have marked as delta records.



Picture 18: DTP Delta transactions (Extraction tab)

When executing the DTP we transfer the 2 delta transactions from the source table to the InfoCube. As you can see by having a look to the package dimension of our InfoCube (see Picture 19), by execution of the DTP we created an entry DIMID = '22', which correlates to field SID_0RECORDTP = '0'.



Data Browser: Table /BIC/DZDGRNCUM1P 6 Hits

DIM...	SID_0CHNGID	SID_0RECORDTP	SID_0REQUID
0	0	0	0
1	0	1	0
2	0	2	0
20	0	1	3.620.664
21	0	2	3.620.665
22	0	0	3.620.666

Picture 19: Package dimension after execution of Delta DTP

Let's have a look what happened on HDB level by choosing runtime information view of our fact table (see Picture 20).

Table Name: /BIC/FZDGRNCUM1 Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 236 KB Size (Main): 22 KB

Records: 5 Size (Delta): 213 KB

Details for Table

Parts | Columns

Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	64	6	58	5

Picture 20: Runtime information after execution of Delta DTP

As you can see our delta records of table YNCUMVALUES (Picture 2) had been added again to the '*' Partition (Part ID 4).

4. Contents of the fact table

Well, after now transferring all our data records from the source through usage of the various DTPs, how does now our fact table look like?

Data Browser: Table /BIC/FZDGRNCUM1 5 Hits

Check Table...

KEY_ZDGRNCUM1P	SID_0CALDAY	SID_0BASE_UOM	SID_0PLANT	SID_0MATERIAL	RECTOTSTCK	ISSTOTSTCK
20	99.991.231	1.000.000.267	23.029	10.222.270	150,000	0,000
21	20.120.101	1.000.000.267	23.029	10.222.270	100,000	0,000
21	20.120.301	1.000.000.267	23.029	10.222.270	50,000	0,000
22	20.120.501	1.000.000.267	23.029	10.222.270	75,000	0,000
22	20.120.601	1.000.000.267	23.029	10.222.270	25,000	0,000

Picture 21: Contents of the fact table (Fact table view) prior to compression

5. Differences between HANA based version and non-Hana based version of Inventory data staging

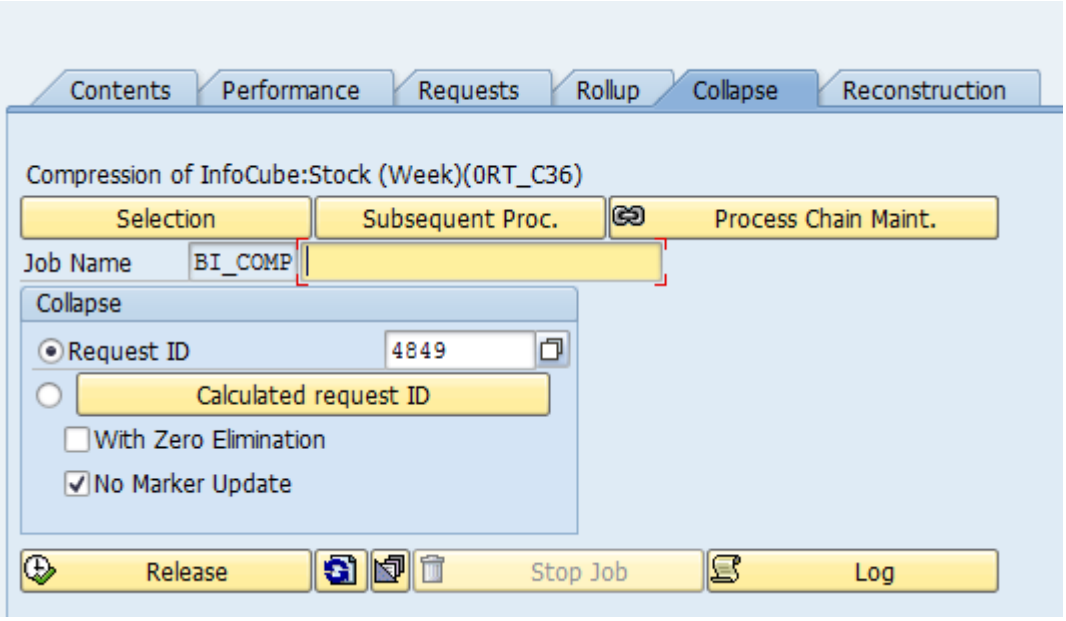
5.1 Logic of 'No marker update' pushed down

Instead of setting the flag for Marker update during the compression of an InfoCube you now have to decide within the DTP whether the records are treated to be historical transactions or deltas. That means the logic of the marker update had been pushed down to the data staging DTPs.

The storage of the records has been changed.

Historical transactions are stored in an own partition on database level which can be identified by SID_0RECORDTP = 2. In the non-HANA based handling all historical transactions had been assigned to SID_0RECORDTP = 0. These historical transactions had to be compressed with the flag set for 'No marker update' (see Picture 22).

Compression of the InfoCube does not change the value of the Initialization any longer. The concept of the previous known reference point in infinity changed. Therefore the keyfigures of the Initialization request remain unchanged during compression. This was completely different before, as all the delta movements led to an update of the reference point.



Picture 22: Flag for No Marker Update in non HANA based BW versions

5.2 No 'E fact table' any longer

In contrast to the non-HANA based Inventory InfoCube, the HANA based Inventory InfoCube has only one F fact table. The fact table itself is partitioned having in total 4 partitions. The purpose of the single partitions is summarized in Table 1: Partition purpose.

Partition	Purpose
1	Compressed deltas
2	Initialization records
3	Historic transactions
4	Uncompressed requests

Table 1: Partition purpose

For details see Picture 41 and Picture 42.

Recommendation

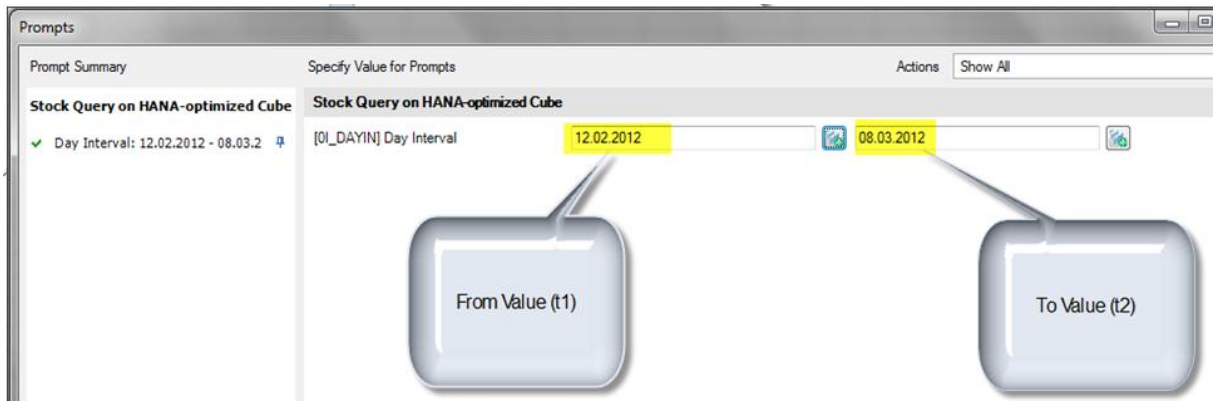
Collapse the Inventory InfoCube regularly to populate the relevant partitions.

6. Query

To make explanation of the query behavior of a query containing non-cumulative keyfigures easier, we decided to use a simple query. As you can see in Picture 25 the query has the option to select a range of days and displays the Product, Store and Day in the rows. As keyfigure to be displayed we have selected Total Stock Quantity.

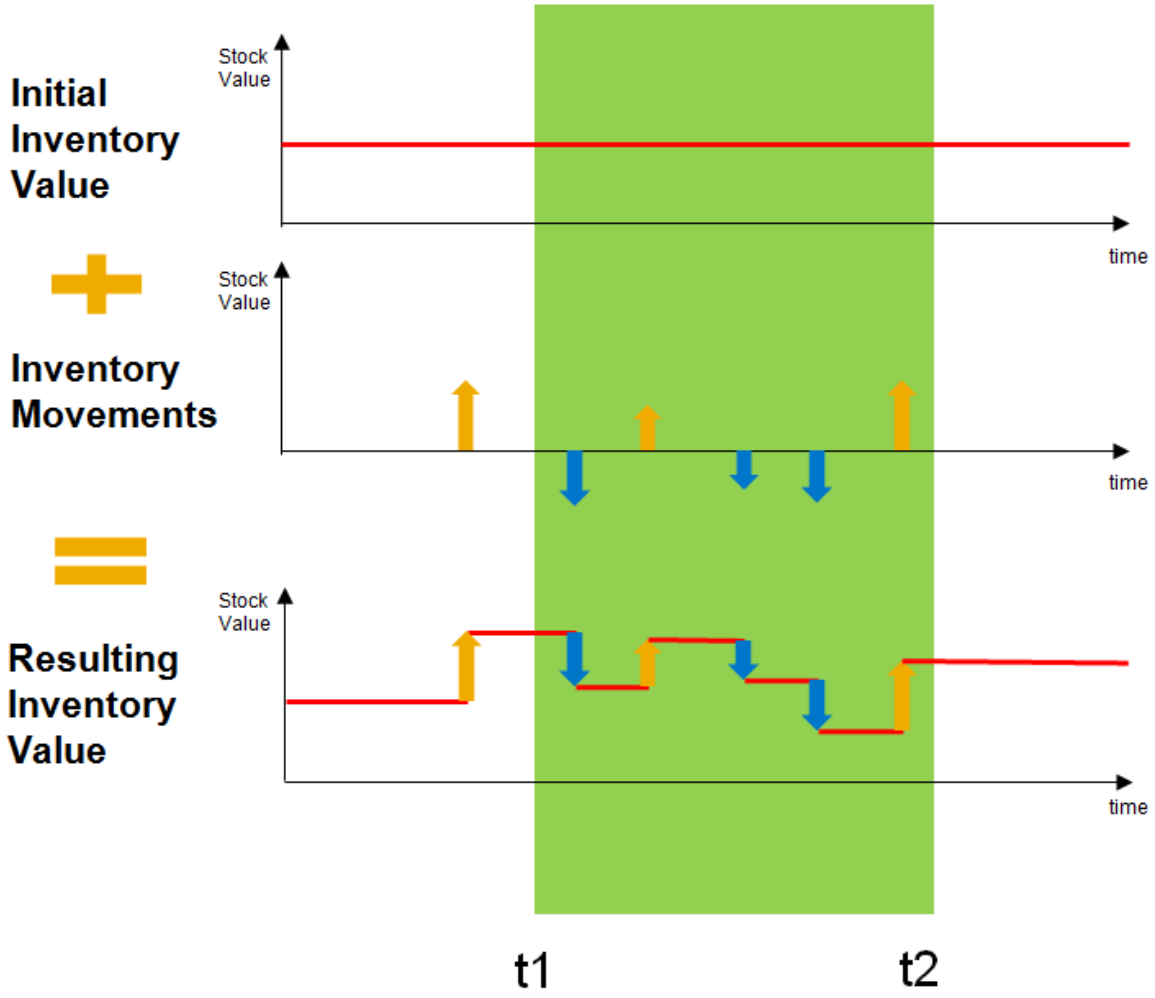
Technically the system always calculates the stock by running 2 queries.

Let us introduce a few terms so we can explain the handling of a non-cumulative query better.



Picture 23: Selection range t1, t2

Within the query below we have the possibility to select a range of days. The upper limit of the selection should be t2 (in the case above equal to 08.03.2012), the lower limit t1 (in our example 12.02.2012) (see Picture 23).



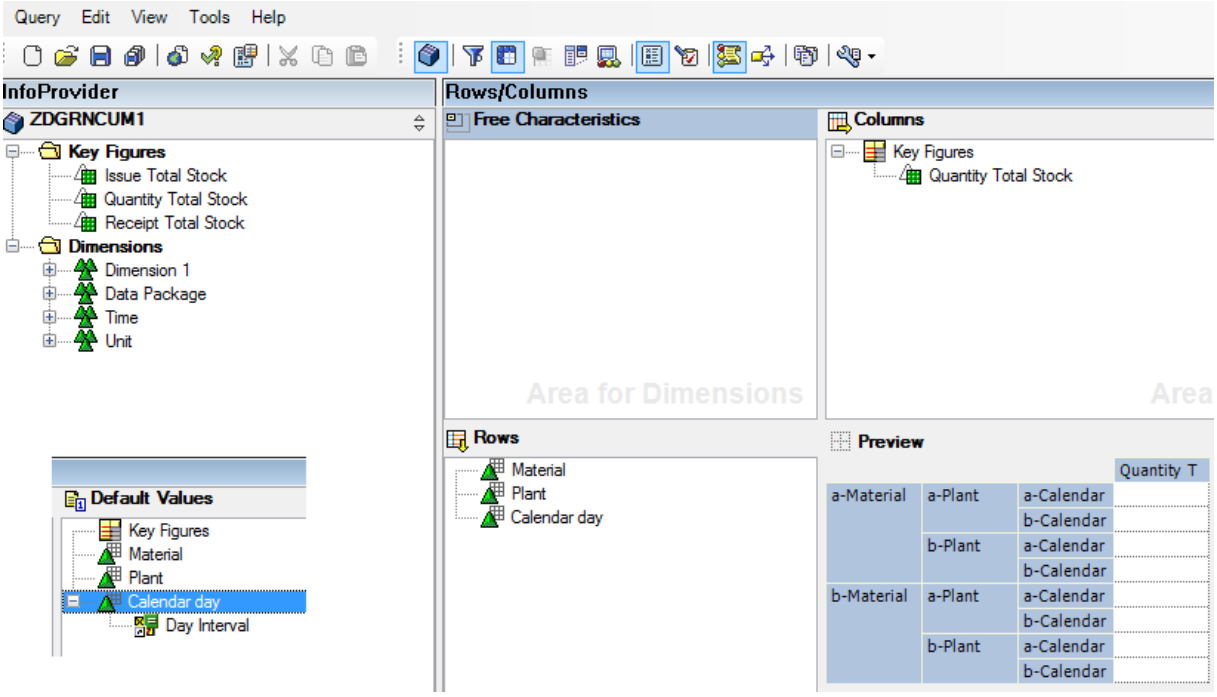
Picture 24: Non-cumulative handling in SAP BW on HANA

Non-cumulative handling (simplified expressed):

In a first query the system has to determine the offset (Initial Inventory Value) for the whole calculation. This offset is always the stock value of the upper time limit of the query time selection (here t2).

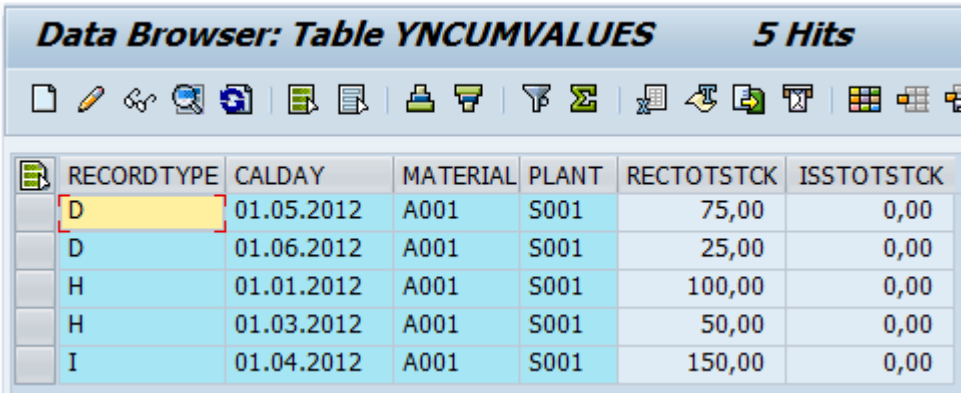
This offset is determined by selecting three different terms. Term A is the initialization record (0recordtp = '1'). The query also has to select all historic transactions larger than t2 (0recordtp = '2'), as these have influence on the height of the offset. Let's call this selection B (=all historic transactions larger than t2). Finally the system also needs to know all deltas smaller or equal than t2 (0recordtp = '0'). This should be our selection C. By now aggregating this information and returning the values of A, B and C to the OLAP engine, the OLAP engine calculates the offset as A – B + C.

In a second query the system than collects all deltas between t2 and t1. By considering whether the delta was an in- or outflow the OLAP can now calculate any stock value in time between t1 and t2 (see Appendix Picture 70).



Picture 25: Query definition

To understand the output of our query, let us allow a small repetition what was stored in our source table.



Data Browser: Table YNCUMVALUES 5 Hits

RECORDTYPE	CALDAY	MATERIAL	PLANT	RECTOTSTCK	ISSTOTSTCK
D	01.05.2012	A001	S001	75,00	0,00
D	01.06.2012	A001	S001	25,00	0,00
H	01.01.2012	A001	S001	100,00	0,00
H	01.03.2012	A001	S001	50,00	0,00
I	01.04.2012	A001	S001	150,00	0,00

Picture 26: Table contents of source table

If we now run our query on the Inventory InfoCube and select Day equal to the 01.04.2012 we get the following output (t2 = 01.04.2012, t1 = 01.04.2012), which is exactly our Initialization record. In the following screen shots you see the output of Advanced Analysis Office. To make the display easier, we used the selection of specific members for 0CALDAY in the filter of the Advanced Analysis output (especially in Picture 30).

	A	B	C	D
1				Quantity Total Stock
2	**Material**	**Plant**	**Calendar day**	PCE
3	A001	S001	01.04.2012	150

Picture 27: Query t2=01.04.2012

In a second query we filter now day = 29.02.2012. As there was a inflow on March 1st we now get Total Stock Qty = Init – Inflow of March 1st.

	A	B	C	D
1				Quantity Total Stock
2	**Material**	**Plant**	**Calendar day**	PCE
3	A001	S001	29.02.2012	100

Picture 28: Query t2=29.02.2012

In the third query we filter day to 02.05.2012. Remember, there was a inflow on Mai 1st. Therefore Total Stock Qty is equal to Init + Inflow of May 1st.

	A	B	C	D
1				Quantity Total Stock
2	**Material**	**Plant**	**Calendar day**	PCE
3	A001	S001	02.05.2012	225

Picture 29: Query t2=02.05.2012

In the last query we now filter day between 31.12.2011 and 01.06.2012. Remember, there had been inflows on Jan 1st, March 1st, Mai 1st and June 1st.

	A	B	C	D
1				Quantity Total Stock
2	**Material**	**Plant**	**Calendar day**	PCE
3	A001	S001	31.12.2011	0
4			01.01.2012	100
5			02.01.2012	100
6			29.02.2012	100
7			01.03.2012	150
8			02.03.2012	150
9			31.03.2012	150
10			01.04.2012	150
11			02.04.2012	150
12			30.04.2012	150
13			01.05.2012	225
14			02.05.2012	225
15			01.06.2012	250

Picture 30: Query t2=01.06.2012

7. Influence of InfoCube Compression

In the last section we now want to demonstrate what happens if you compress (collapse) the InfoCube (changes highlighted)

Status before compression:

Contents | Requests | Collapse | Reconstruction

InfoCube requests for InfoCube:In memory Opt Inventory InfoCube(ZDGRNC

Requ...	R...	C...	Re...	Lo...	DTP/InfoPackage	Request Date	Update Date	Selection Conditions	Transf...	Added...
36206...					FG Delta transactions (...	12.03.2012	12.03.2012	RECORDTYPE=D;	2	2
36206...					FG Historic transactions ...	12.03.2012	12.03.2012	RECORDTYPE=H;	2	2
36206...					FG Initialization (DTP_0...	12.03.2012	12.03.2012	RECORDTYPE=I;	1	1

Picture 31: Monitor Status before compression

Again, let us have a look at the contents of the fact table (see Picture 32), package dimension table (see Picture 33) on HDB level (Picture 34) before compression.

Data Browser: Table /BIC/FZDGRNCUM1 5 Hits

Check Table...

KEY_ZDGRNCUM1P	SID_0CALDAY	SID_0BASE_UOM	SID_0PLANT	SID_0MATERIAL	RECTOTSTCK	ISSTOTSTCK
20	99.991.231	1.000.000.267	23.029	10.222.270	150,000	0,000
21	20.120.101	1.000.000.267	23.029	10.222.270	100,000	0,000
21	20.120.301	1.000.000.267	23.029	10.222.270	50,000	0,000
22	20.120.501	1.000.000.267	23.029	10.222.270	75,000	0,000
22	20.120.601	1.000.000.267	23.029	10.222.270	25,000	0,000

Picture 32: Fact table view before compression

Data Browser: Table /BIC/DZDGRNCUM1P 6 Hits

DIM...	SID_0CHNGID	SID_0RECORDTP	SID_0REQUID
0	0	0	0
1	0	1	0
2	0	2	0
20	0	1	3.620.664
21	0	2	3.620.665
22	0	0	3.620.666

Picture 33: Package dimension view before compression

Table Name: /BIC/FZDGRNCUM1 Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 236 KB Size (Main): 22 KB

Records: 5 Size (Delta): 213 KB

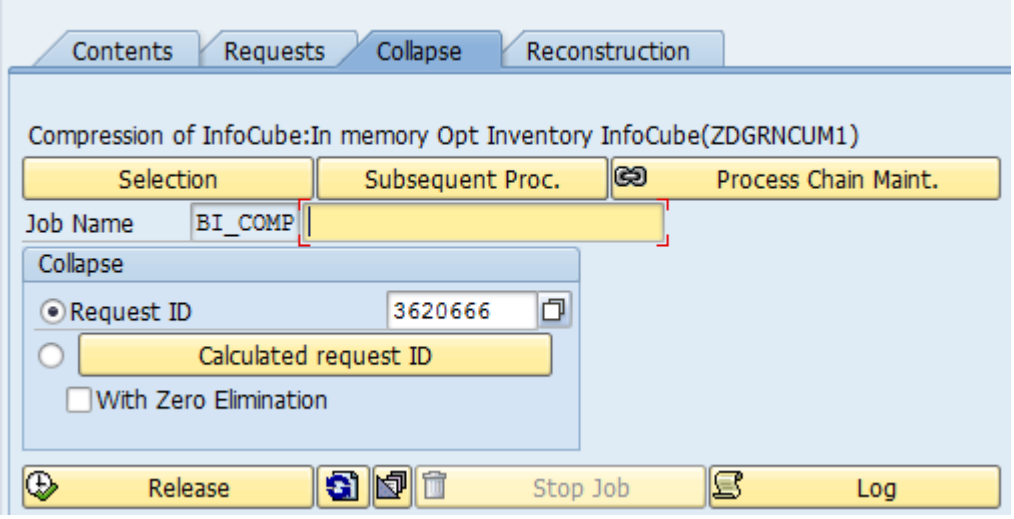
Details for Table

Parts | Columns

Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	64	6	58	5

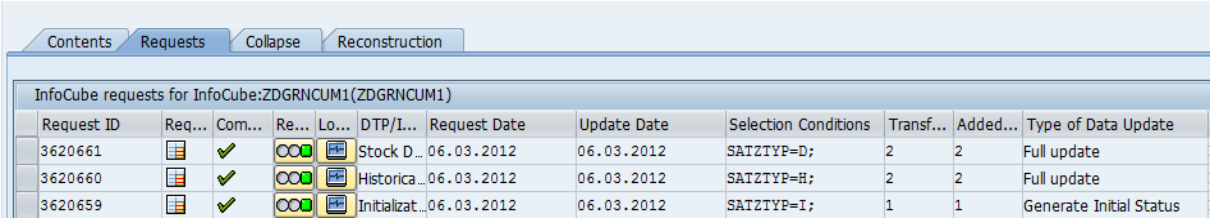
Picture 34: HDB View before compression

We now start the InfoCube compression by collapsing the InfoCube (see Picture 35).



Picture 35: Collapsing InfoCube

The success of the compression can also be seen by looking at the monitor (see Picture 36)



Picture 36: Monitor after compression

By comparison between Picture 37 and Picture 38 you can see, that the records of the initialization request got moved into the dimension relevant for Initialization records (0recordtp = '1'). The values of the keyfigures remain the same, so these are not updated. This is also completely different to the behavior of BW releases on traditional database technologies. Also the historic transactions are still visible (DIMID = '2'), that's also different to the concept in a non HDB version.

Data Browser: Table /BIC/FZDGRNCUM1 5 Hits

Check Table...

KEY_ZDGRNCUM1P	SID_0CALDAY	SID_0BASE_UOM	SID_0PLANT	SID_0MATERIAL	RECTOTSTCK	ISSTOTSTCK
20	99.991.231	1.000.000.267	23.029	10.222.270	150,000	0,000
21	20.120.101	1.000.000.267	23.029	10.222.270	100,000	0,000
21	20.120.301	1.000.000.267	23.029	10.222.270	50,000	0,000
22	20.120.501	1.000.000.267	23.029	10.222.270	75,000	0,000
22	20.120.601	1.000.000.267	23.029	10.222.270	25,000	0,000

Picture 37: Fact table view before compression

Data Browser: Table /BIC/FZDGRNCUM1 5 Hits

Check Table...

KEY_ZDGRNCUM...	SID_0CALDAY	SID_0BASE_UOM	SID_0PLA...	SID_0MATERIAL	RECTOTSTCK	ISSTOTSTCK
0	20.120.501	1.000.000.267	23.029	10.222.270	75,000	0,000
0	20.120.601	1.000.000.267	23.029	10.222.270	25,000	0,000
1	99.991.231	1.000.000.267	23.029	10.222.270	150,000	0,000
2	20.120.101	1.000.000.267	23.029	10.222.270	100,000	0,000
2	20.120.301	1.000.000.267	23.029	10.222.270	50,000	0,000

Picture 38: Fact table view after compression

Data Browser: Table /BIC/DZDGRNCUM1P 6 Hits

DIM...	SID_0CHNGID	SID_0RECORDTP	SID_0REQUID
0	0	0	0
1	0	1	0
2	0	2	0
20	0	1	3.620.664
21	0	2	3.620.665
22	0	0	3.620.666

Picture 39: Package dimension before compression

Data Browser: Table /BIC/DZDGRNCUM1P 3 Hits

DIM...	SID_0CHNG...	SID_0RECORDT...	SID_0REQU...
0	0	0	0
1	0	1	0
2	0	2	0

Picture 40: Package dimension after compression

As you can see, the DIMIDs 20, 21 and 22 got removed from the package dimension table.

Table Name: /BIC/FZDGRNCUM1 Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 236 KB Size (Main): 22 KB

Records: 5 Size (Delta): 213 KB

Details for Table

Parts | Columns

Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	58	6	52	0
2	hdbpoc00	30003	58	6	52	0
3	hdbpoc00	30003	58	6	52	0
4	hdbpoc00	30003	64	6	58	5

Picture 41: HDB View before compression

Table Name: /BIC/FZDGRNCUM1 Schema: SAPOH0

Columns | Indexes | Further Properties | Runtime Information

General

Table Size: 246 KB Size (Main): 24 KB

Records: 5 Size (Delta): 221 KB

Details for Table

Parts | Columns

Part ID	Host	Port	Total Size [KB]	Main Size [KB]	Delta Size [KB]	Records
1	hdbpoc00	30003	63	6	57	2
2	hdbpoc00	30003	63	6	56	1
3	hdbpoc00	30003	63	6	57	2
4	hdbpoc00	30003	58	6	52	0

Picture 42: HDB view after compression

On HDB level you can see that the records got distributed to the various partitions. That's the main reason why you should compress your InfoCube regularly.

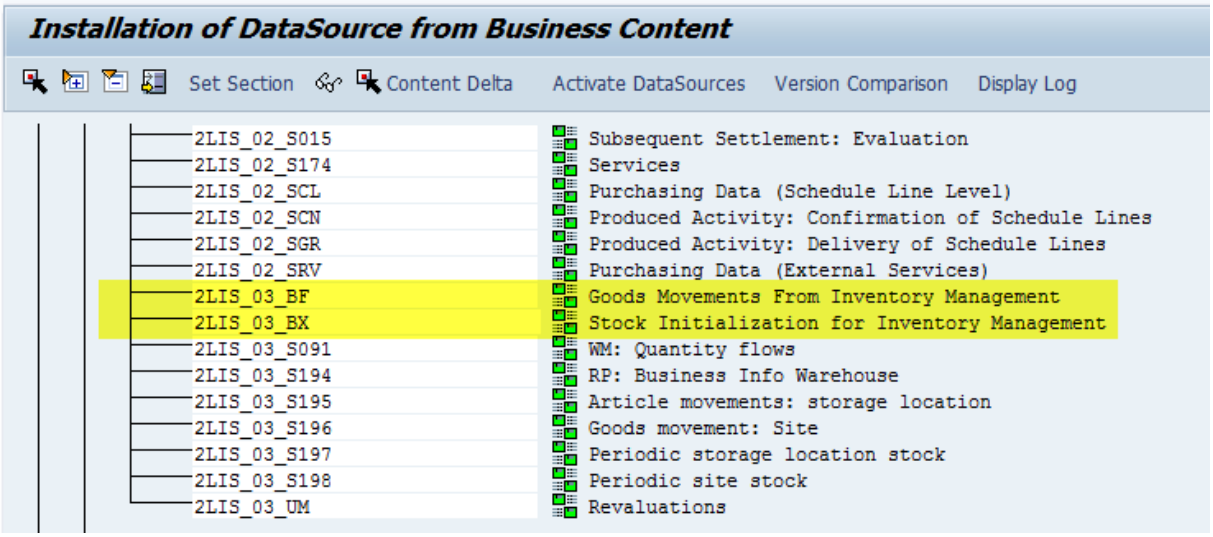
8. Extraction from SAP ERP

The following datasources are to be used in the extraction of an inventory scenario based on SAP ERP (see Picture 43: SAP ERP based inventory sources):

2LIS_03_BX Material stock

2LIS_03_BF Material movements

2LIS_03_UM Revaluations



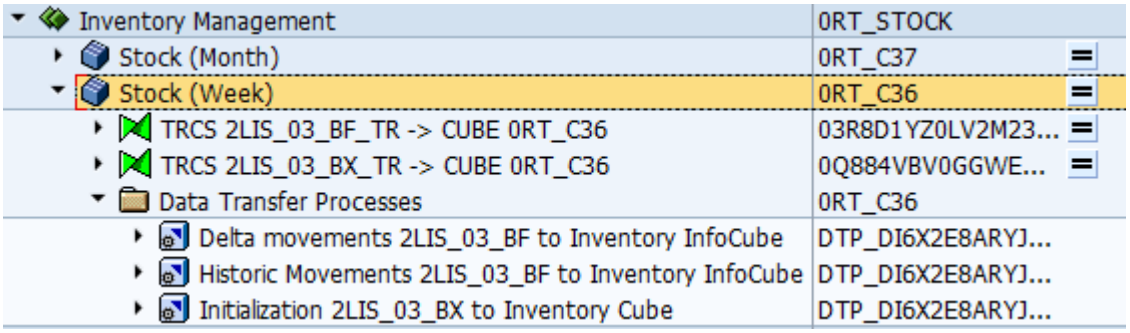
Picture 43: SAP ERP based inventory sources

The first datasource (2LIS_03_BX) is used to extract an opening stock balance on a detailed level (material, plant, storage location and so on). The opening stock is the situation of the operative stock in the SAP ERP system to the time you scheduled the statistical setup ran for the datasource 2LIS_03_BX. (This is because no documents are to be posted during this run and so the stock does not change during this run). It is not possible to choose a key date freely.

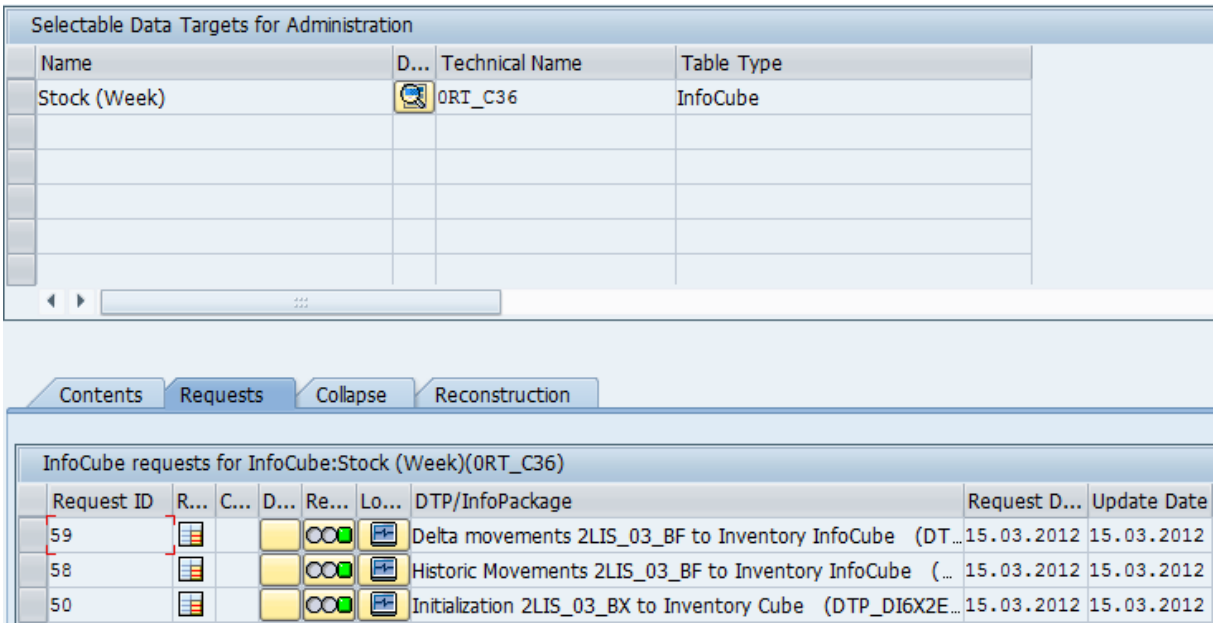
The second datasource (2LIS_03_BF) is used to extract the material movements into the SAP BW system. This datasource is used to extract historic as well as delta transactions (see Picture 45).

The third of the above datasources (2LIS_03_UM) contains data from valuated revaluations in Financial Accounting (document BSEG). This data is required to update valuated stock changes for the calculated stock balance in the BW. This information is not required in many situations as it is often only the quantities that are of importance. Therefore the handling of that DataSource is not treated in this document. This datasource only describes financial accounting processes, not logistical ones. In other words, only the stock value is changed here, no changes are made to the stock quantities. Everything that is subsequently mentioned here about the upload sequence and compression regarding datasource 2LIS_03_BF also applies to this datasource.

The complete picture for a SAP Retail based extractions looks like in Picture 44.



Picture 44: Inventory management for SAP Retail based InfoCube 0RT_C36



Picture 45: Inventory InfoCube fed from SAP sources 2LIS_03_BX and 2LIS_03_BF

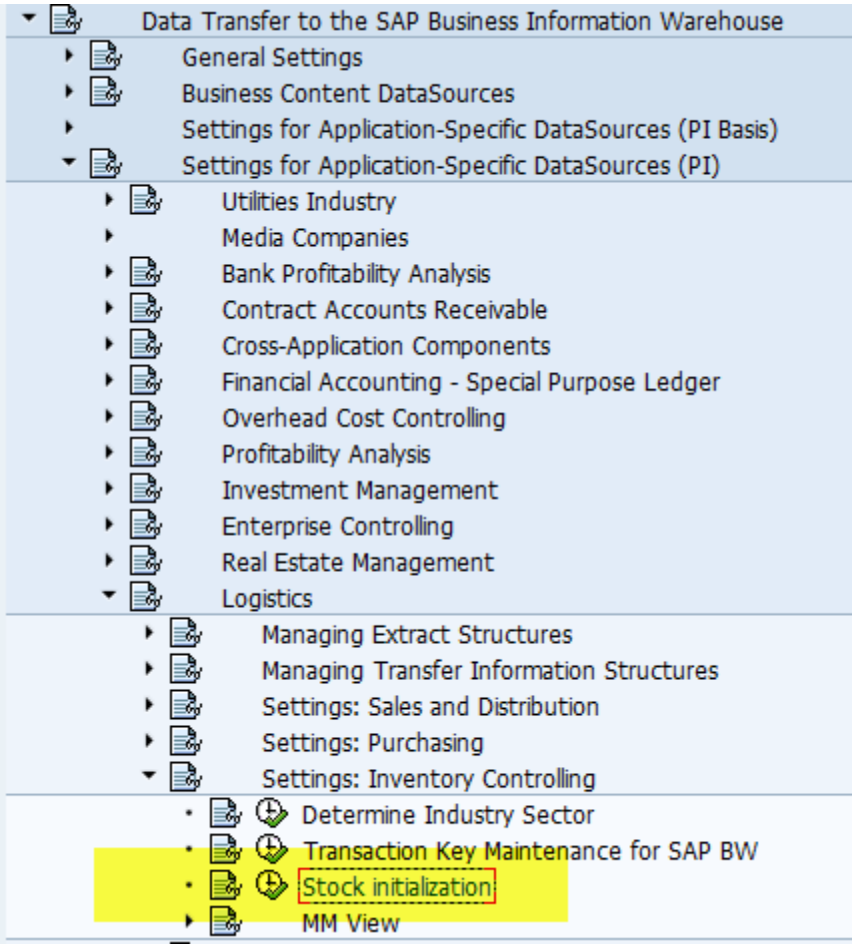
8.1 Extracting inventory data (standard procedure)

For previous releases of BW this is already very well documented in
http://scn.sap.com/docs/DOC-35750

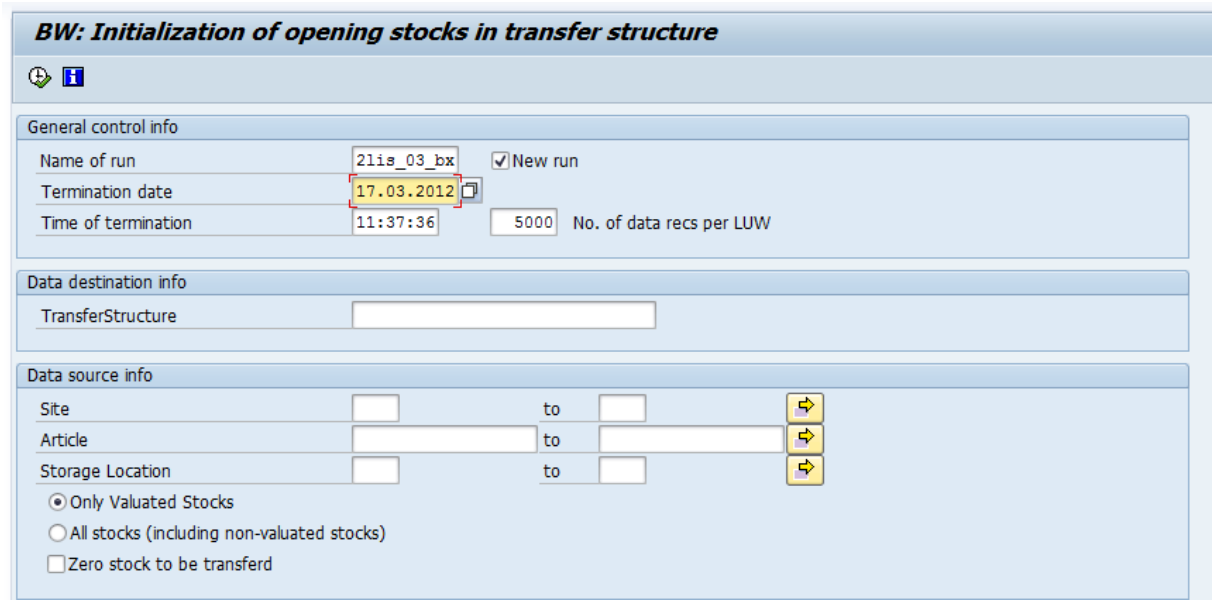
A data flow could look like this:

a) Lock users in ECC
While you initialize the setup tables for inventory transaction changes to the documents that are written to the setup table are not allowed. Therefore you have to lock the application in ERP system against changes to material documents.

b) Fill setup table for 2LIS_03_BX.
This can be done with transaction MCNB. In case you want to speed up the whole process you can schedule several variants of the report with distinct selections.
Before you schedule the initialization please ensure that no data from a previous initialization exists in the setup table. Therefore please delete the content of the table first.

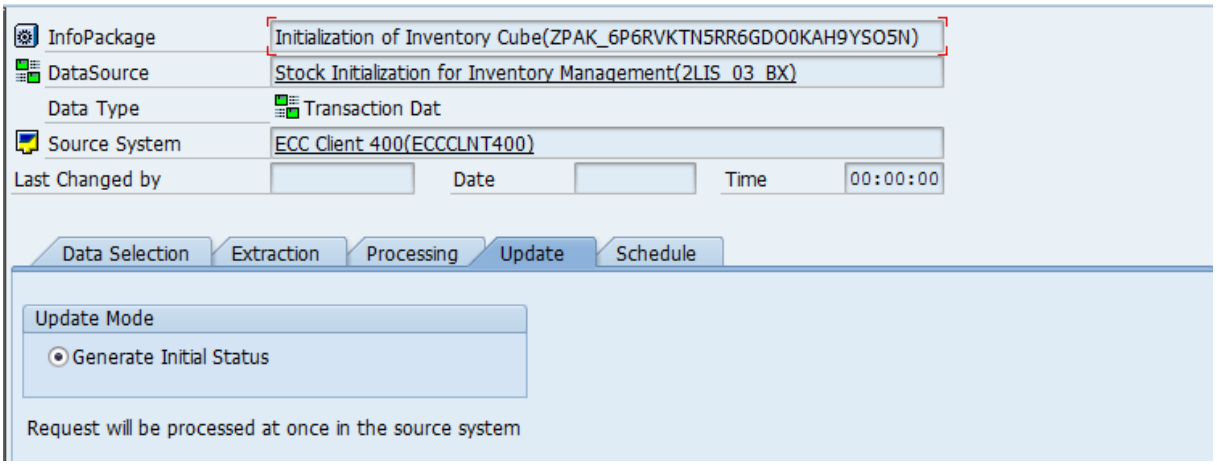


Picture 46: Transaction SBIW: Stock Initialization

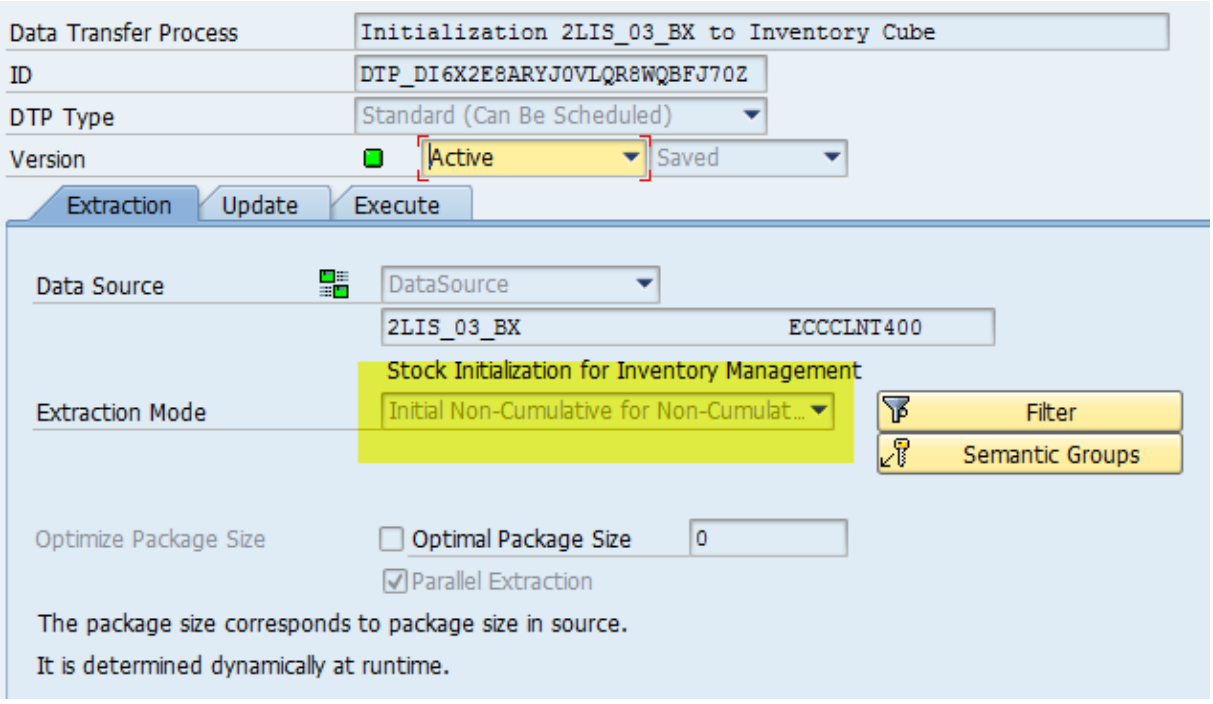


Picture 47: Fill setup table for 2LIS_03_BX

c) Extract Initialization records with DataSource 2LIS_03_BX to PSA and start subsequent DTP for Initialization.



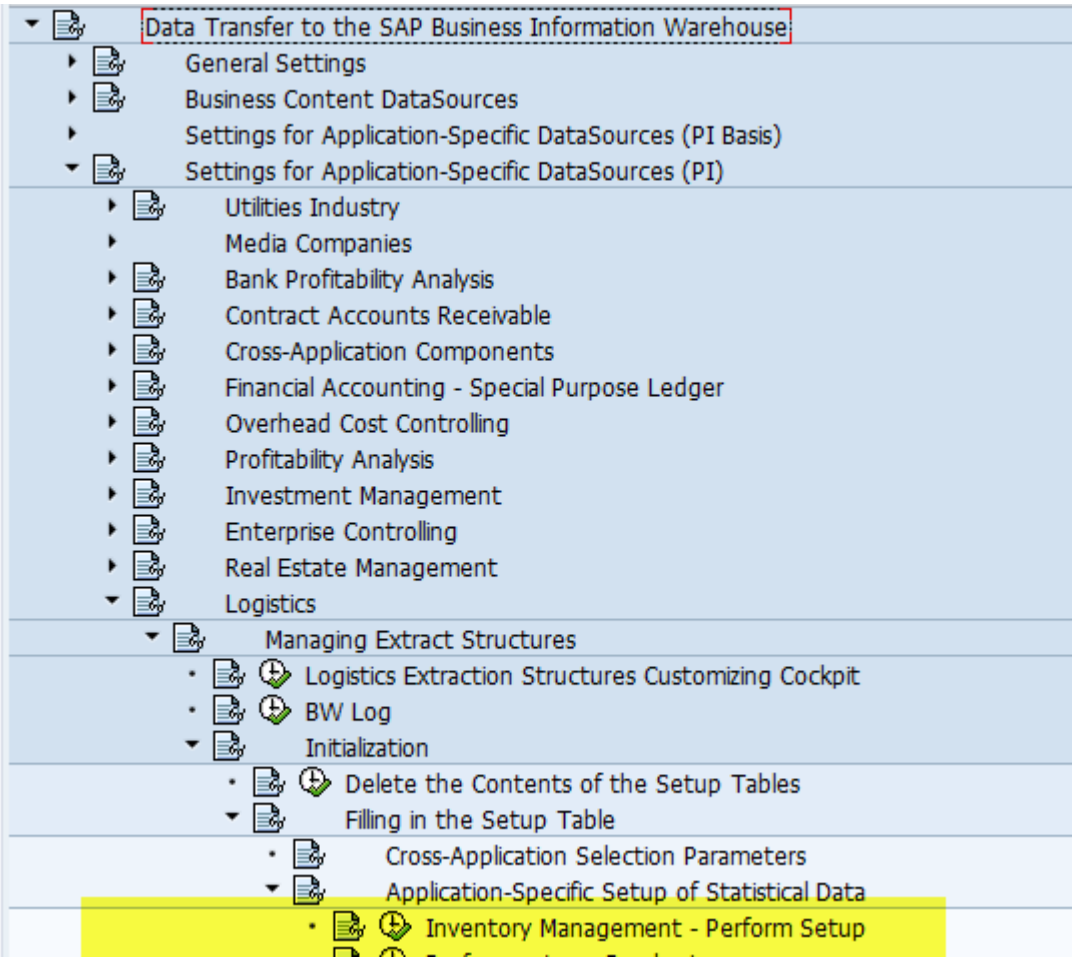
Picture 48: Initialization InfoPackage for 2LIS_03_BX



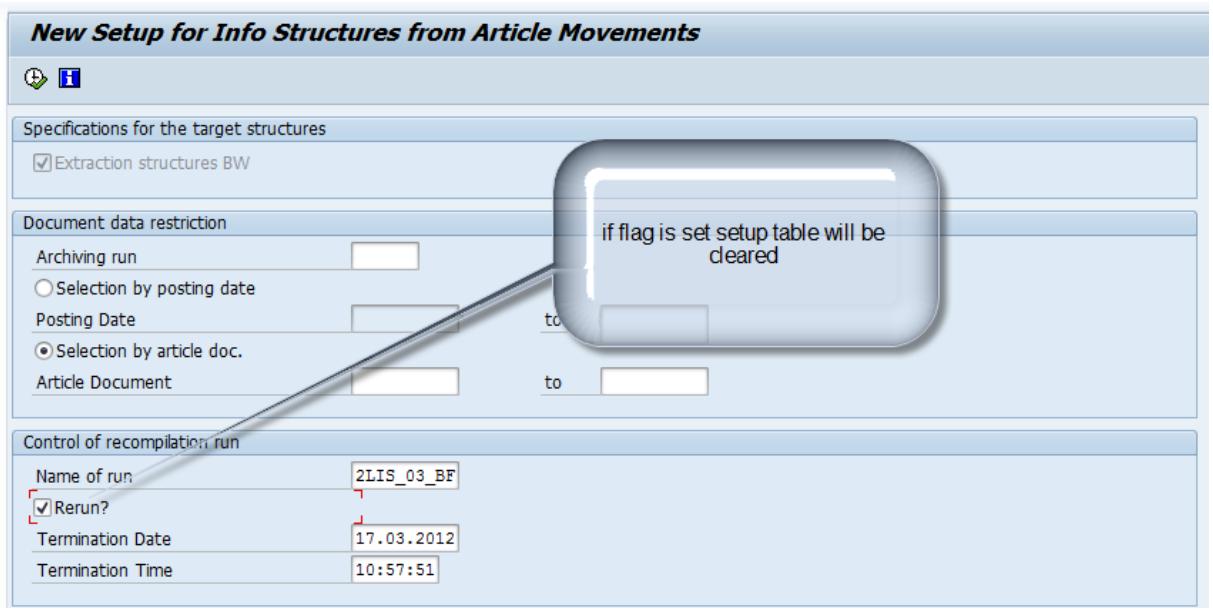
Picture 49: DTP for Initialization (2LIS_03_BX)

d) Fill setup table for material movements in ERP.
Before starting the filling of the setup table please ensure that the setup table is empty. Therefore delete the content of the setup table first.
Fill the setup table by calling transaction OLI1BW. You can schedule various variants of the program with distinct selections to speed up the whole process of filling the setup table.

To minimize the downtime for SAP ERP you can split the history of material movement documents into those for which still changes apply and those for which no changes are allowed anymore (due to closed periods, etc).
I will describe that scenario in section 8.2 of this document. In case you have low volumes in ECC and the lock phase in ECC is not an issue, you can continue with the steps described in this section.

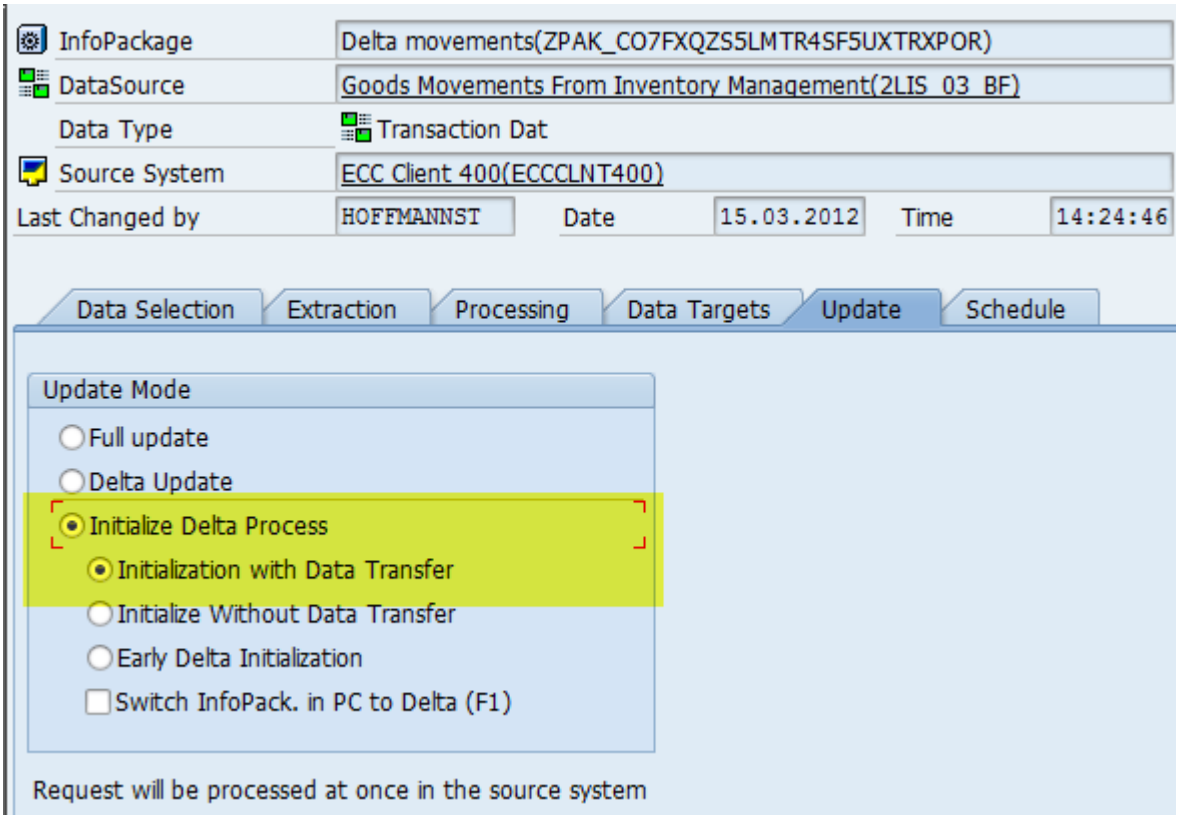


Picture 50: Customizing: Fill setup tables for historic inventory transactions



Picture 51: Fill setup table for 2LIS_03_BF

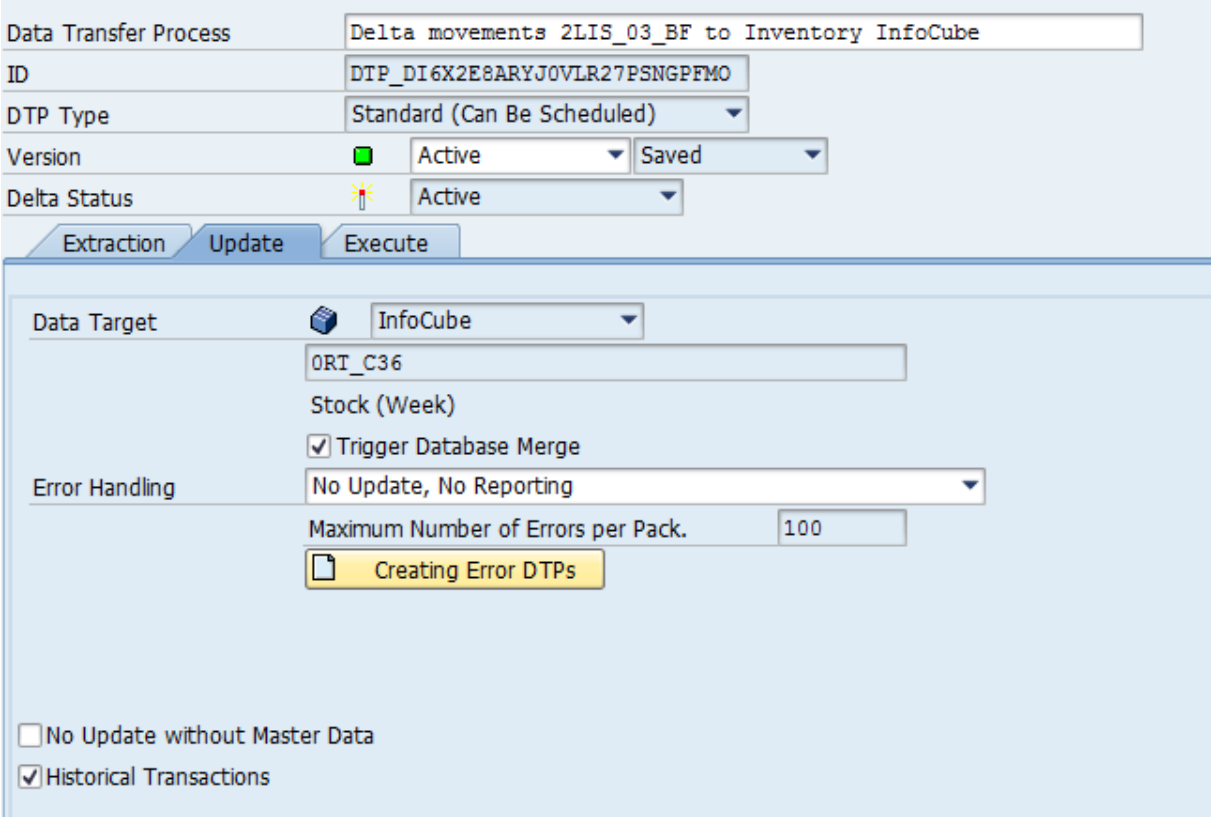
e) Delta initialization for material movements.
Once the filling of the setup table is finished you can initialize the delta upload for DataSource 2LIS_03_BF by starting the InfoPackage in BW.



Picture 52: InfoPackage for delta documents

f) Unlock users in ERP again
After successful creation of the delta queue in ECC for 2LIS_03_BF (can be checked with transaction RSA7 in ECC), you can allow again changes to material movement documents in ECC.

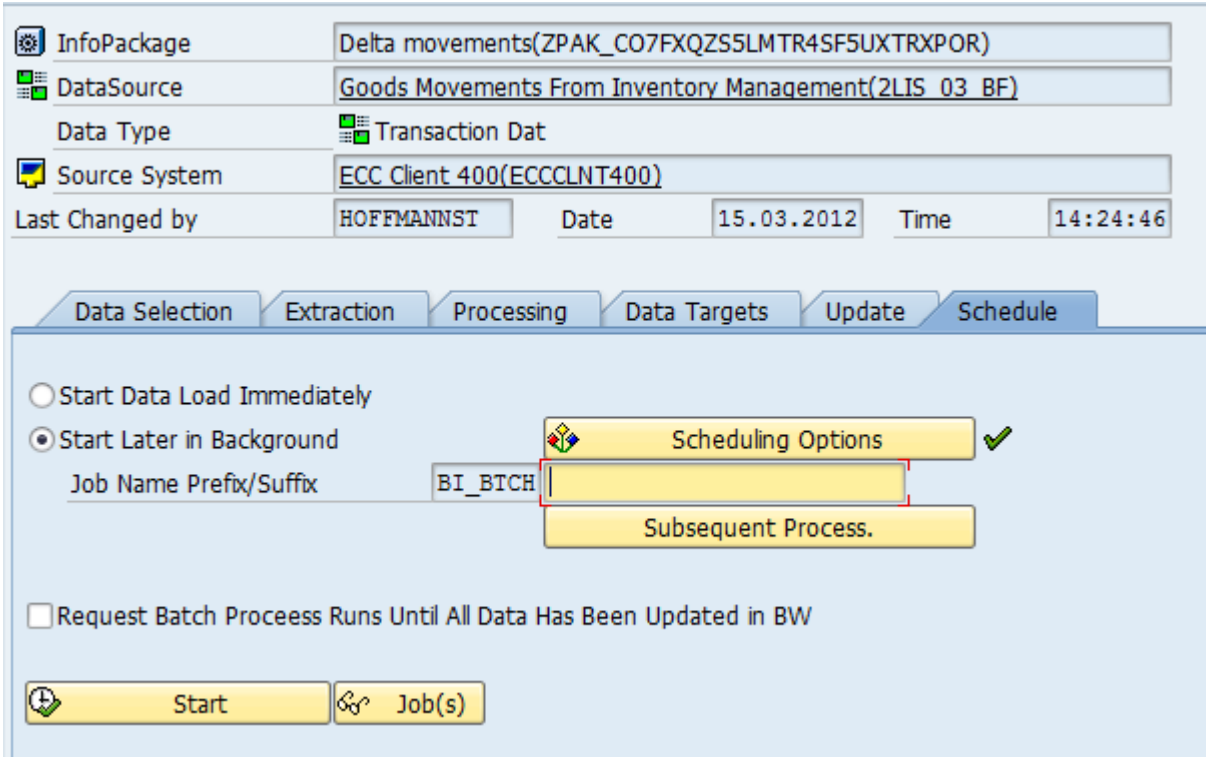
g) Start subsequent DTP for delta transactions (Delta DTP).



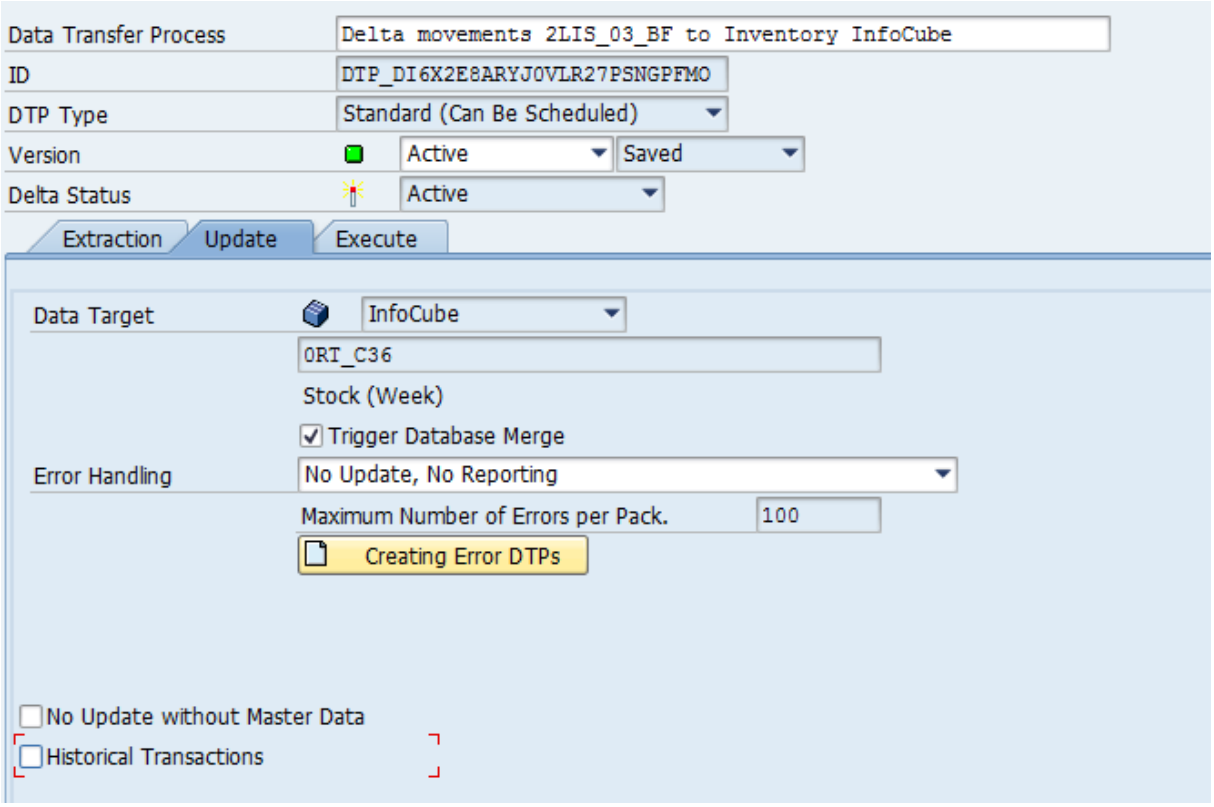
Picture 53: Delta DTP for Delta transactions (material movement documents of 2LIS_03_BF)

We recommend putting a LSA architecture in place from which you could in case of reconstruction at least get the initialization and historic records again.

h) Setup extracting regularly deltas from ECC followed by Delta DTPs



Picture 54: InfoPackage to extract deltas from ECC



Picture 55: Delta DTP for Delta transactions (material movement documents of 2LIS_03_BF)

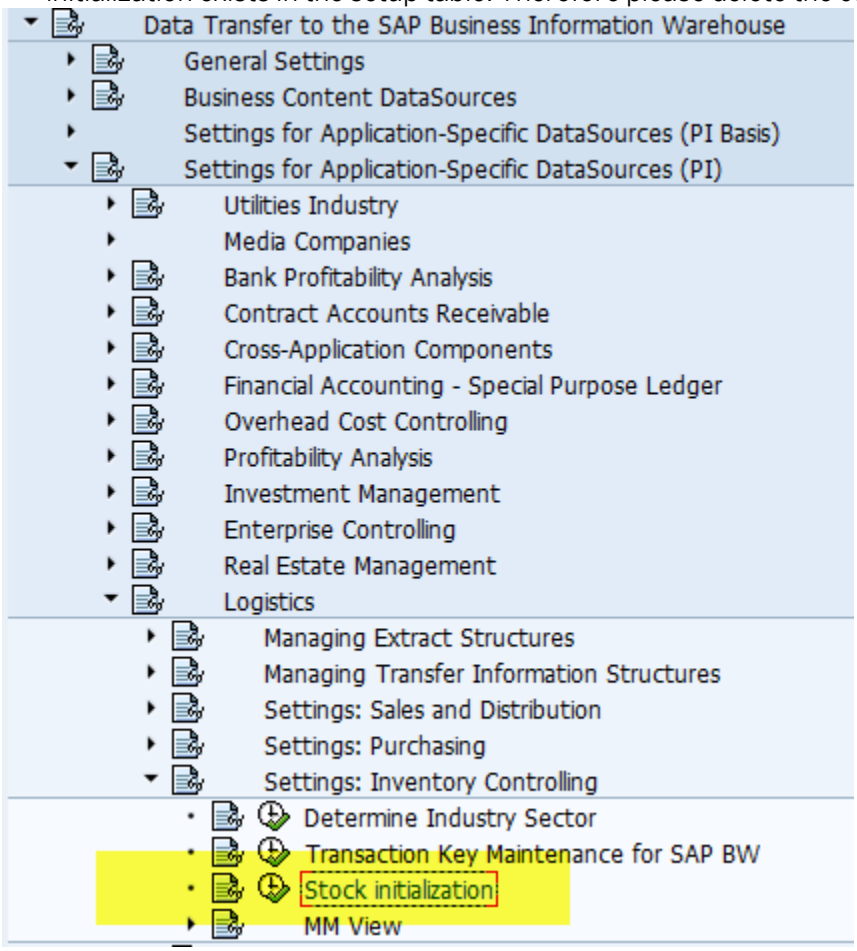
8.2 Extracting inventory data (procedure to minimize locking phase in ECC)

In case you want to minimize the time you have to lock the users in ECC, you can also proceed as follows

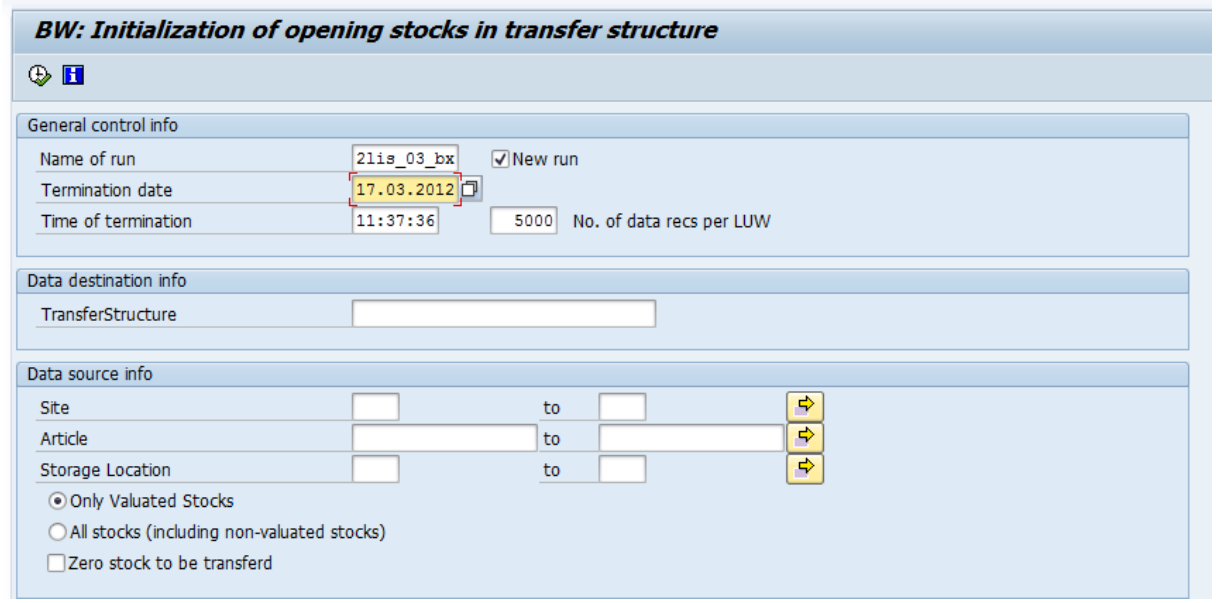
A data flow could look like this:

a) Lock users in ECC
While you initialize the setup tables for inventory transaction changes to the documents that are written to the setup table are not allowed. Therefore you have to lock the application in ERP system against changes to material documents.

b) Fill setup table for 2LIS_03_BX.
This can be done with transaction MCNB. In case you want to speed up the whole process you can schedule several variants of the report with distinct selections.
Before you schedule the initialization please ensure that no data from a previous initialization exists in the setup table. Therefore please delete the content of the table first.

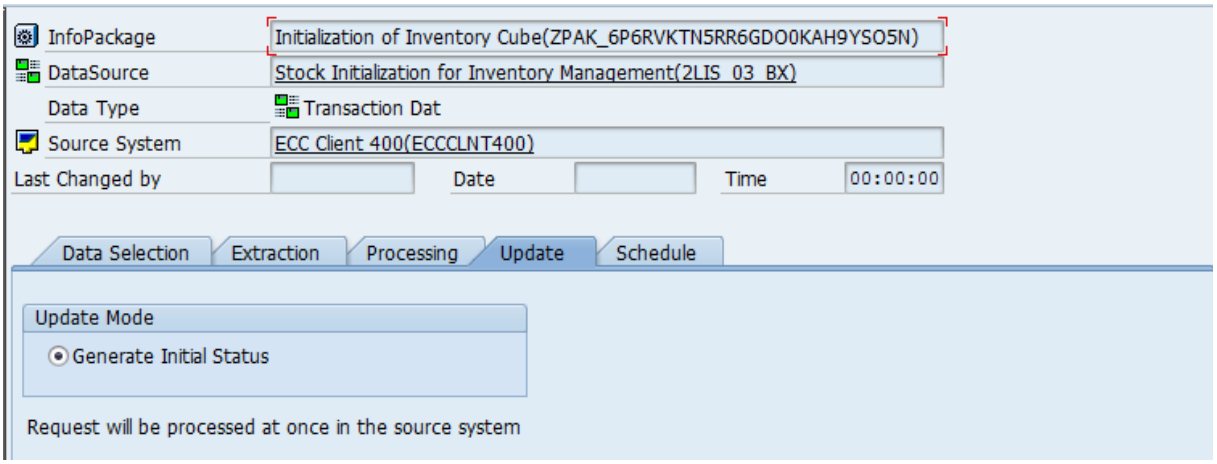


Picture 56: Transaction SBIW: Stock Initialization

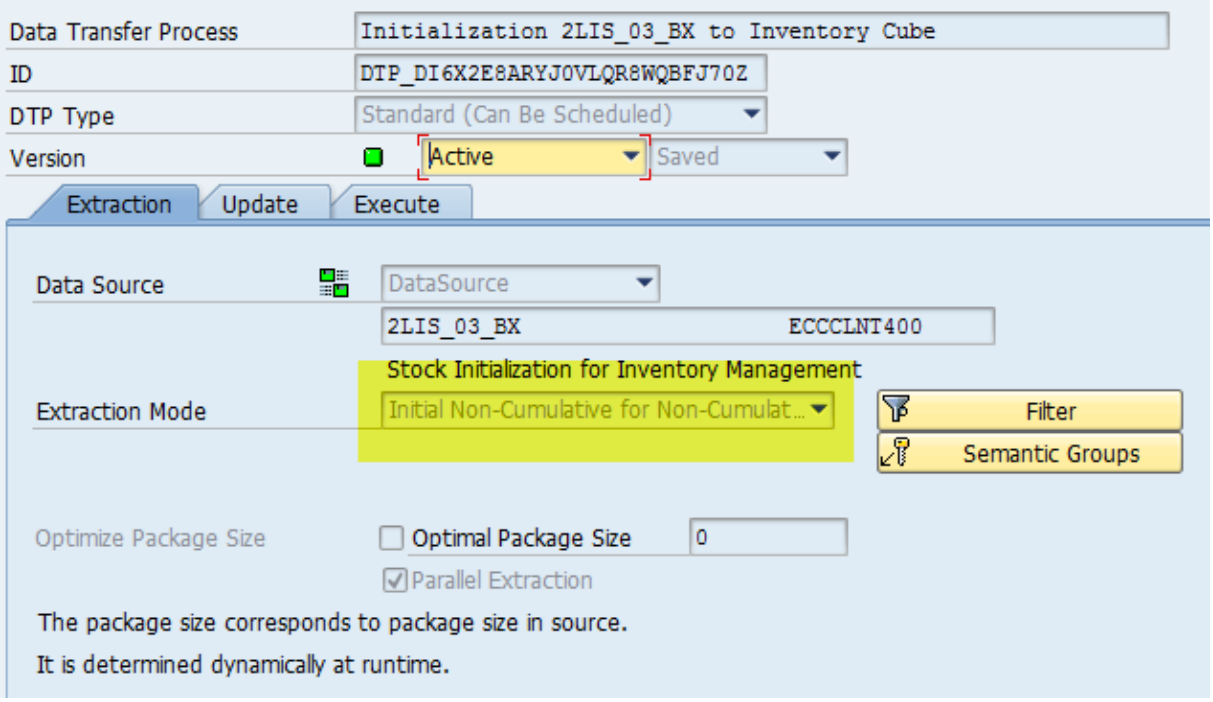


Picture 57: Fill setup table for 2LIS_03_BX

c) Extract Initialization records with DataSource 2LIS_03_BX to PSA and start subsequent DTP for Initialization.



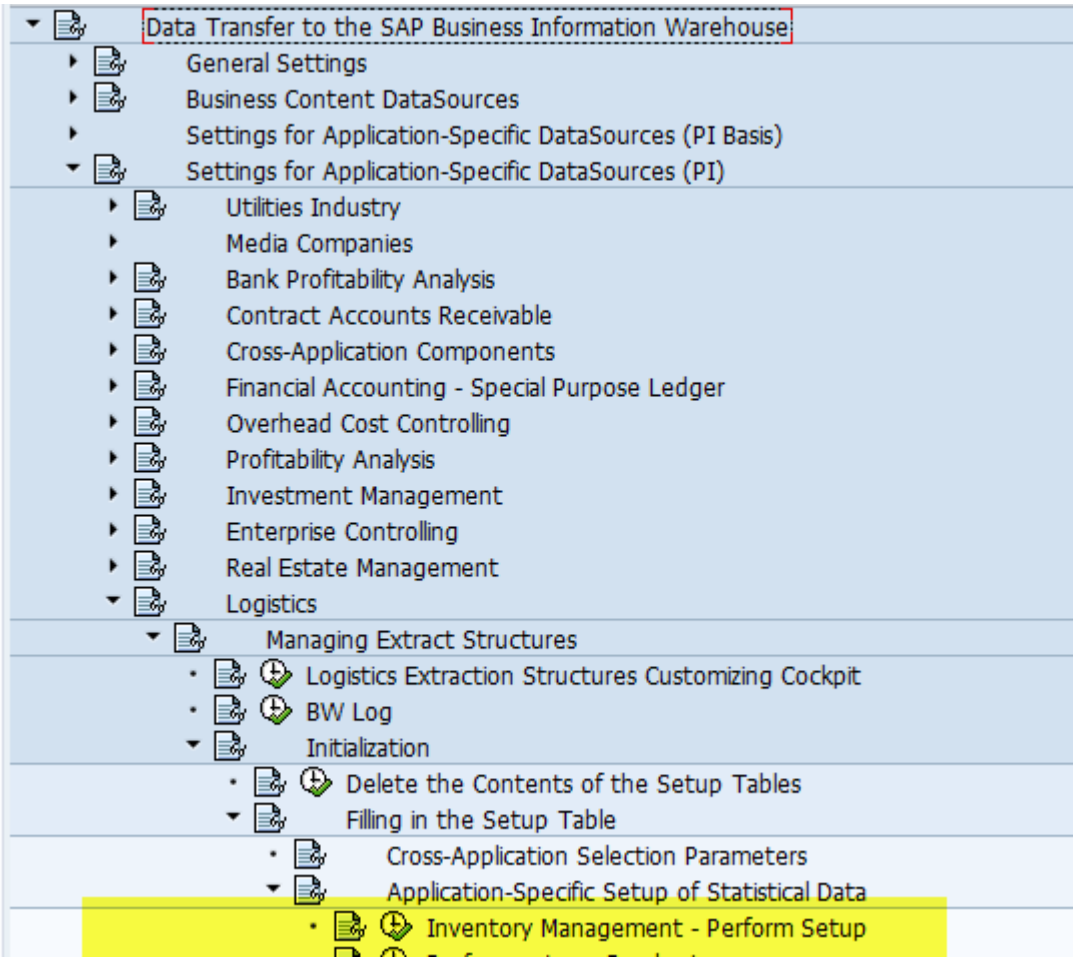
Picture 58: Initialization InfoPackage for 2LIS_03_BX



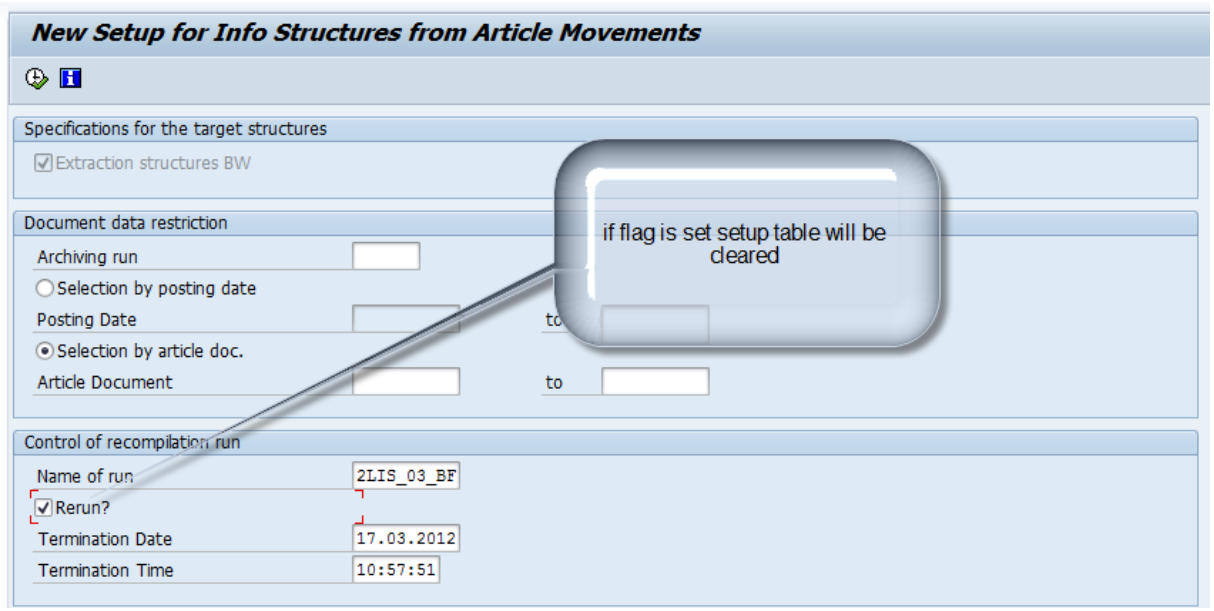
Picture 59: DTP for Initialization (2LIS_03_BX)

d) Fill setup table for material movements transactions in ERP to which still changes can happen (Selection A).
This is performed by calling transaction OLI1BW. You can schedule various variants of the program with distinct selections to speed up the whole process of filling the setup table. Before filling the setup table please ensure that there is no data in from a previous run. Therefore delete the content of the table first.

To minimize the downtime for SAP ERP you split the history of inventory documents into those for which still changes apply (Selection A) and those for which no changes are allowed anymore (due to closed periods, etc) (Selection B).
Fill the setup table by restricting the selection in OLI1BW to that period / documents where changes to material documents still are allowed to occur (Selection A).

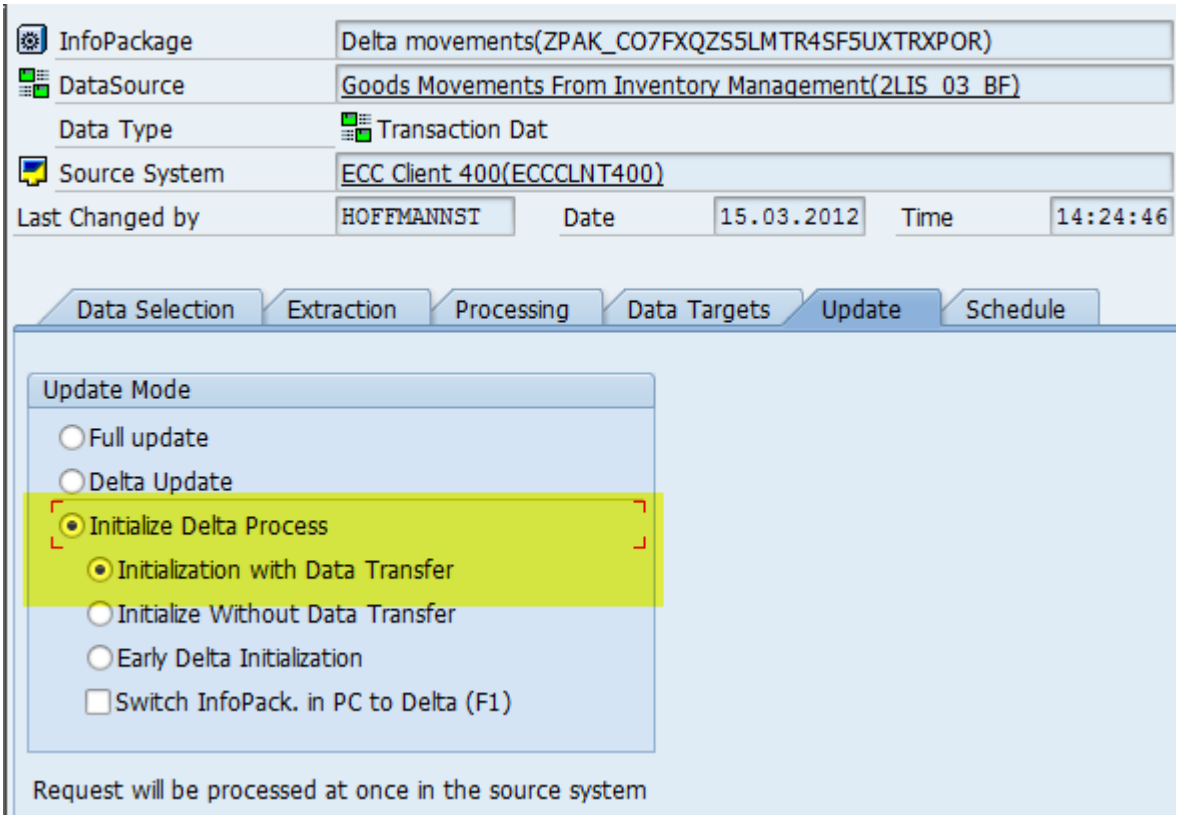


Picture 60: Customizing: Fill setup tables for historic inventory transactions



Picture 61: Fill setup table for 2LIS_03_BF

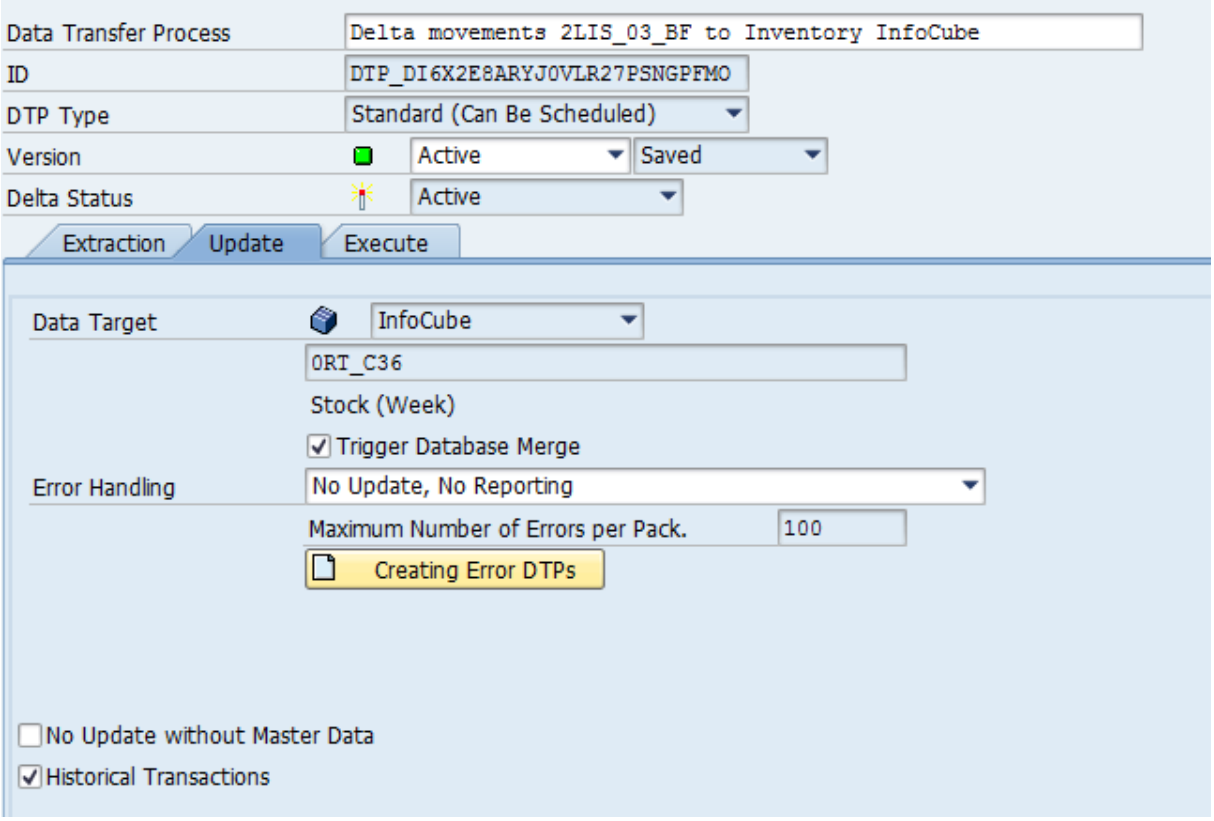
e) Delta initialization for material movements.
Once the filling of the setup table is finished you can initialize the delta upload for DataSource 2LIS_03_BF by starting the InfoPackage in BW.



Picture 62: InfoPackage for delta documents

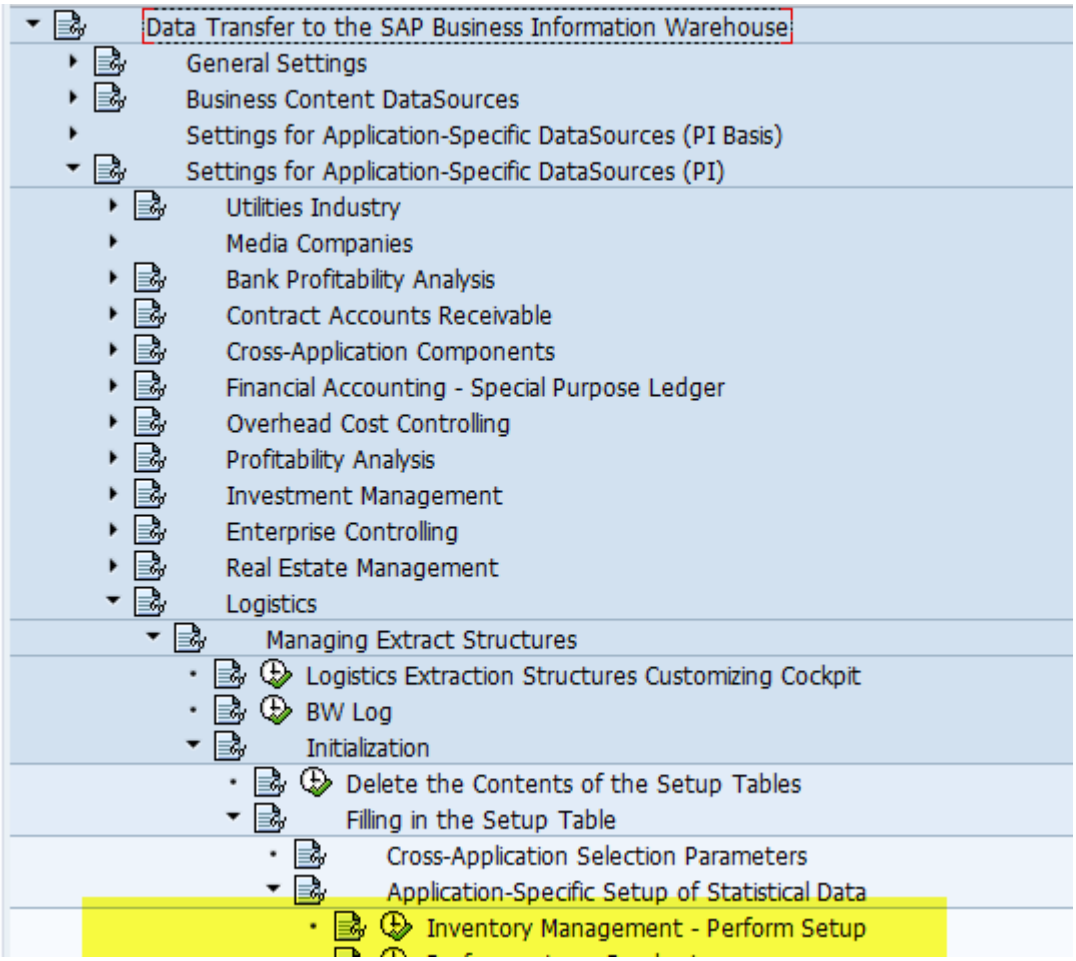
f) Unlock users in ERP again (in case you locked them before)
After successful creation of the delta queue in ECC for 2LIS_03_BF (can be checked with transaction RSA7 in ECC), you can allow again changes to material movement documents in ECC.

g) Start subsequent DTP for delta transactions (Delta DTP).

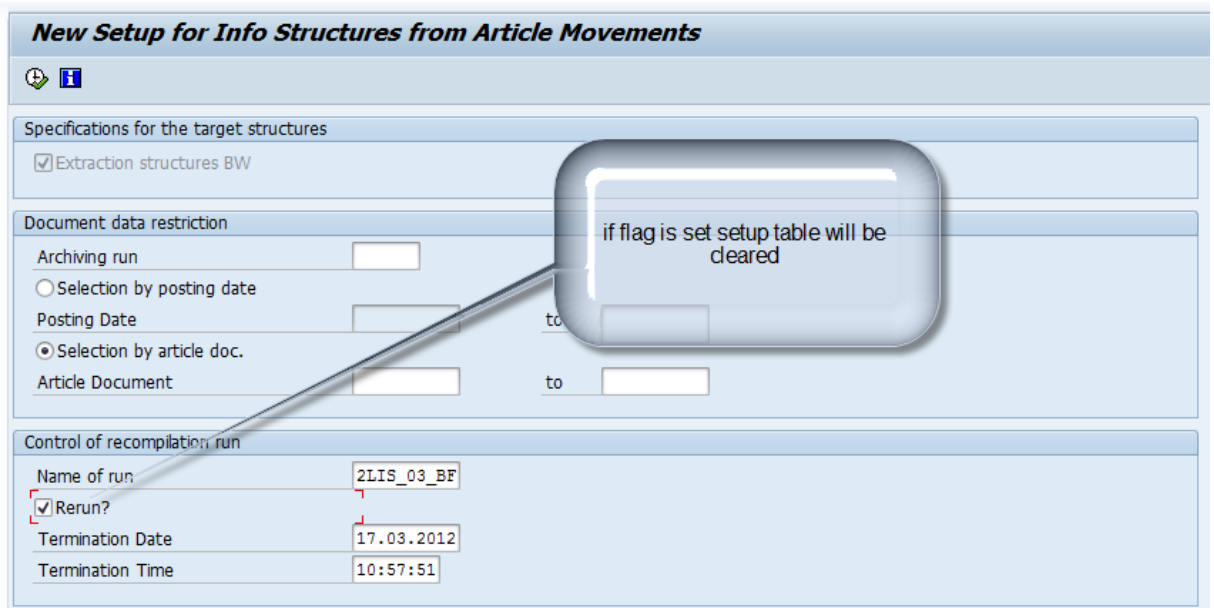


Picture 63: Delta DTP for Historic Transactions

h) Fill setup table for material movements transactions in ERP to which no changes can happen any longer (Selection B).
Before filling the setup table again delete the content of the previous run in the setup table first.

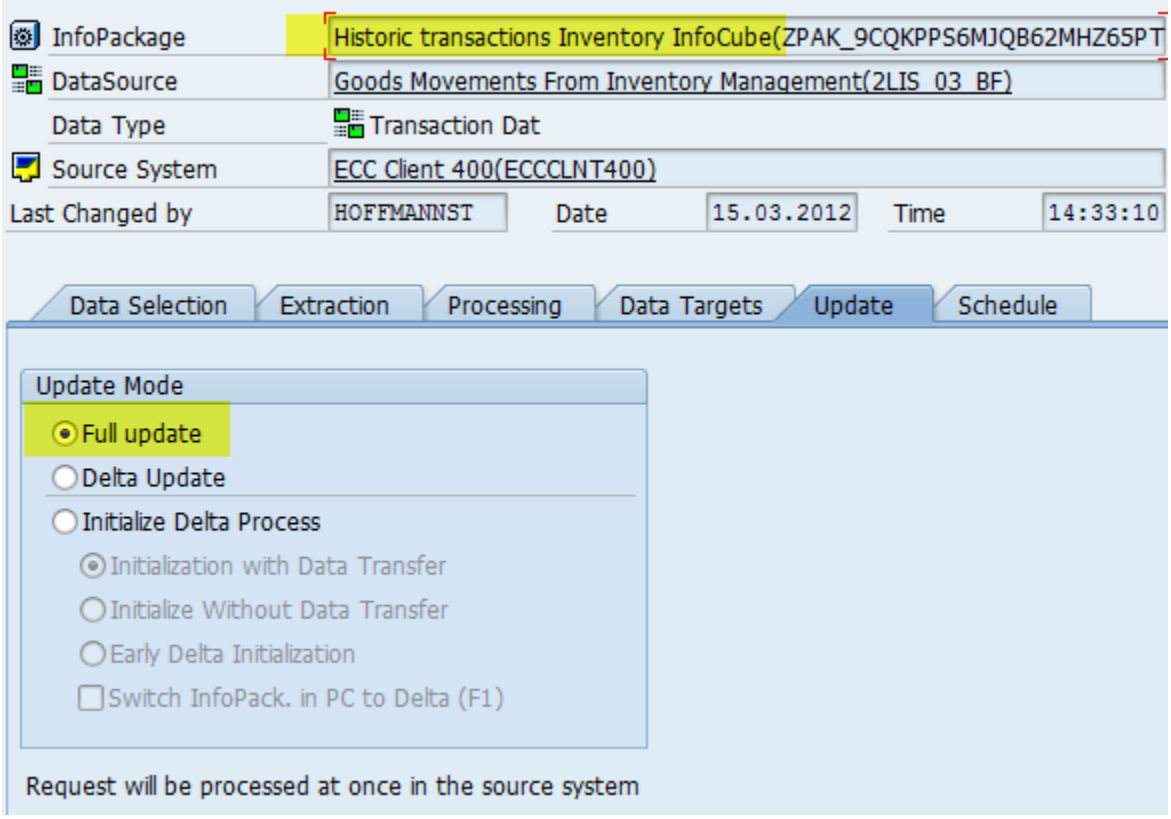


Picture 64: Customizing: Fill setup tables for historic inventory transactions

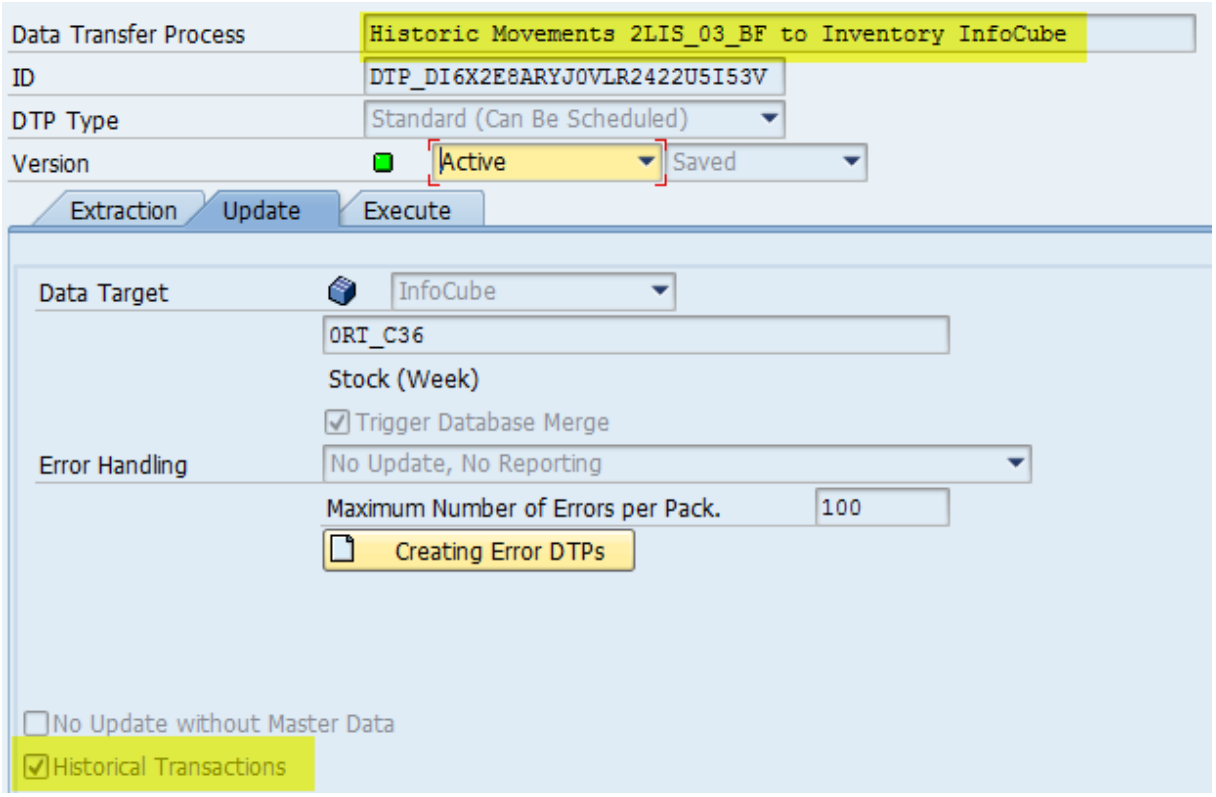


Picture 65: Fill setup table for 2LIS_03_BF

i) Extract historic movements with DataSource 2LIS_03_BF to PSA and start subsequent DTP for historic transactions (Delta DTP).

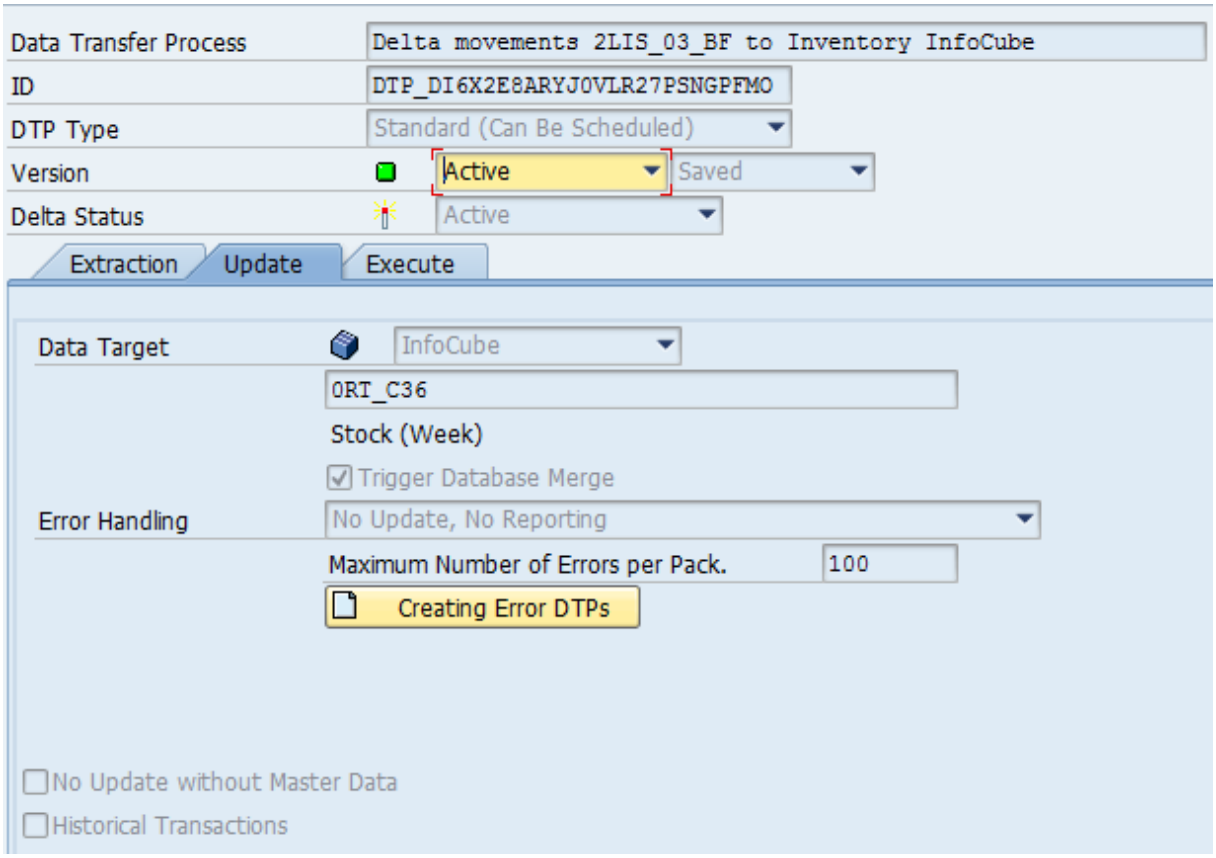


Picture 66: InfoPackage for historic transactions



Picture 67: Full DTP for historic transactions (2LIS_03_BF)

j) Delete full request with historic transactions from PSA
k) Start subsequent InfoPackage to get deltas and DTP for delta transactions (Delta DTP).



Picture 68: Delta DTP for Delta transactions (material movement documents of 2LIS_03_BF)

We recommend putting a LSA architecture in place from which you could in case of reconstruction at least get the initialization and historic records again.

9. Considerations regarding conversion of a non In Memory Optimized Inventory InfoCube into a In Memory optimized Inventory InfoCube

Before running the database migration you should check in the source BW (BW running on classic DB) whether all requests in your Inventory Cube which contain historical movements are compressed (shouldn't update the reference point). If there are such requests still uncompressed in the InfoCube (in the F fact table), please compress those (using no marker update option) and all requests with a REQUID smaller than the one carrying historic movements.

If you already migrated your database to HANA please proceed as follows to convert your standard non-cumulative InfoCube into an In Memory optimized InfoCube using transaction RSMIGRHANADB. As the recommendations depend on the installation type of your HANA database, we have to differentiate between a non-distributed HANA system (single node) and a distributed HANA system (multi-node).

9.1 Single Node Systems

1. Check whether there are uncompressed historic movements in the InfoCube (see note http://service.sap.com/sap/support/notes/1766577). If there are such requests:

   a. Compress with marker update all requests with a REQUID smaller than those with historic movements.
   b. Compress using no marker update all requests with historic movements.
2. In general it is recommended to compress the standard non-cumulative InfoCube completely before running the conversion to an In Memory optimized InfoCube. Therefore, compress all other uncompressed requests with marker update.
3. In case the complete compression of the standard Inventory InfoCube is not possible / desired, please consider the instructions of note http://service.sap.com/sap/support/notes/1780575. This alternative approach is only possible to avoid compression of "regular" requests. All requests with historic movements must be compressed in any case.
4. Convert the InfoCube to an In Memory optimized InfoCube.

9.2 Multi Node System

1. Check whether there are uncompressed historic movements in the InfoCube (see note 1766577). If there are such requests, delete these requests from InfoCube and reload after InfoCube is converted.
2. Set RSADMIN parameter FORCE_NCUM_CUBE_CONVERSION (see note http://service.sap.com/sap/support/notes/1780575).
3. Convert the InfoCube to an In Memory optimized InfoCube.
4. Reload the historic movements.

10. Conclusions / Recommendations

You should compress the In Memory Optimized InfoCube regularly, as this keeps the size of the '*' partition small.

Use semantic partitioning feature to split the InfoCube into semantic partitions. This allows you to control the overall size of the single database partitions. Especially partition 0 and 1 should be of interest. As a rule of thumb you should split your InfoCube into semantic partitions as soon as you reach more than 500 million records per InfoCube.

Non-cumulative InfoCubes should not be partitioned by time.

11. Appendix

11.1 Query logic

Formula – non-HANA BW

Ref-Point at infinity Diff to Ref-Point

$$value(t0) = \left(\sum_{rectp=1} tuple + \sum_{requid>0} tuple \right) - \sum_{t>t0} tuple$$

Algorithmic processing to compute inventory values of interval [t1, t2]:

- Query 1:
  Read the corresponding reference point (recordtp=1) and
  Read all deltas (recordtp=0) which are not yet compressed (requid>0)
  (no time-cha in group-by)
- Aggregate result to get inventory value at infinity
- Query 2:
  Read all deltas between t1 and oo
  (with time-cha in group-by)
- Loop backwards in time over deltas
  subtract values from corresponding ref-point;
  for each requested time return current value

Picture 69: non-HANA BW non-cumulative handling

Formula – BW-HANA

Ref-Point at t2 — Diff to Ref-Point

$$value(t0) = \left(\sum_{rectp=1} tuple + \sum_{\substack{rectp=0 \\ t \le t2}} tuple - \sum_{\substack{rectp=2 \\ t > t2}} tuple \right) - \sum_{t2 \ge t > t0} tuple$$

Algorithmic processing to compute inventory values of interval [t1, t2]:

- Query 1:
  Read initialization records (rectp=1) and
  Read all deltas (rectp=0) smaller than or equal to t2 and
  Read all hist. movements (rectp=2) larger than t2
  (no time-cha in group-by)
- Aggregate result to get inventory value at t2
- Query 2:
  Read all deltas between t1 and t2
  (with time-cha in group-by)
- Loop backwards in time over deltas
  subtract values from corresponding ref-point;

Picture 70: HANA BW non-cumulative handling

11.2 Query examples

```
SELECT
"F"."SID_0PLANT" AS "S____026"
,"F"."SID_0MATERIAL" AS "S____237"
,"F"."SID_0BASE_UOM" AS "S____420"
,"DP"."SID_0RECORDTP" AS "S____020"
, COUNT( * )  AS "Z____024"
, SUM (  "F"."RECTOTSTCK"  )  AS "Z____7828"
, SUM (  "F"."ISSTOTSTCK"  )  AS "Z____7827"
FROM
"/BIC/FZDGRNCUM1" "F"
JOIN
"/BIC/DZDGRNCUM1P" "DP"
ON
 "F" . "KEY_ZDGRNCUM1P"
= "DP" . "DIMID"
WHERE
 ( ( ( (
"DP"."SID_0REQUID"
<= 3620661
 ) ) ) ) AND  ( ( ( (
"DP"."SID_0RECORDTP"
= 1
 ) ) ) OR  ( ( (
"DP"."SID_0RECORDTP"
= 2
AND
"F"."SID_0CALDAY"
> 20120530
OR
"DP"."SID_0RECORDTP"
= 0
AND
"F"."SID_0CALDAY"
<= 20120530
 ) ) ) )
GROUP BY
"F"."SID_0PLANT"
,"F"."SID_0MATERIAL"
,"F"."SID_0BASE_UOM"
,"DP"."SID_0RECORDTP"
WITH HINT( OLAP_PARALLEL_AGGREGATION )
/*$ dmuid=0008I3A4ZC6SDAGK0H2R78V2G */
```

Just get the Initialization records (sid_0recordtp = 1)

History larger than t2 (sid_0recordtp = 2)

Delta records smaller or equal to t2 (sid_recordtp = 0)

no group by time

Picture 71: 'Reference point' query

```
SELECT
"F"."SID_0PLANT" AS "S____026"
,"F"."SID_0CALDAY" AS "S____035"
,"F"."SID_0MATERIAL" AS "S____237"
,"F"."SID_0BASE_UOM" AS "S____420"
, COUNT( * )  AS "Z____024"
, SUM (  "F"."RECTOTSTCK"  )  AS "Z____7828"
, SUM (  "F"."ISSTOTSTCK"  )  AS "Z____7827"
FROM
"/BIC/FZDGRNCUM1" "F"
JOIN
"/BIC/DZDGRNCUM1P" "DP"
ON
 "F" . "KEY_ZDGRNCUM1P"
= "DP" . "DIMID"
WHERE
 ( ( ( (
"F"."SID_0CALDAY"
BETWEEN 20120201
AND 20120530
 ) ) AND  ( (
"DP"."SID_0RECORDTP"
IN (
   0 ,  2
)
 ) ) AND  ( (
"DP"."SID_0REQUID"
<= 3620661
 ) ) ) )
GROUP BY
"F"."SID_0PLANT"
,"F"."SID_0CALDAY"
,"F"."SID_0MATERIAL"
,"F"."SID_0BASE_UOM"
ORDER BY
 "S____026"
, "S____237"
, "S____420"
WITH HINT( OLAP_PARALLEL_AGGREGATION )
/*$ dmuid=0008I3A4ZC6SDAGK0H2R78V2G */
```

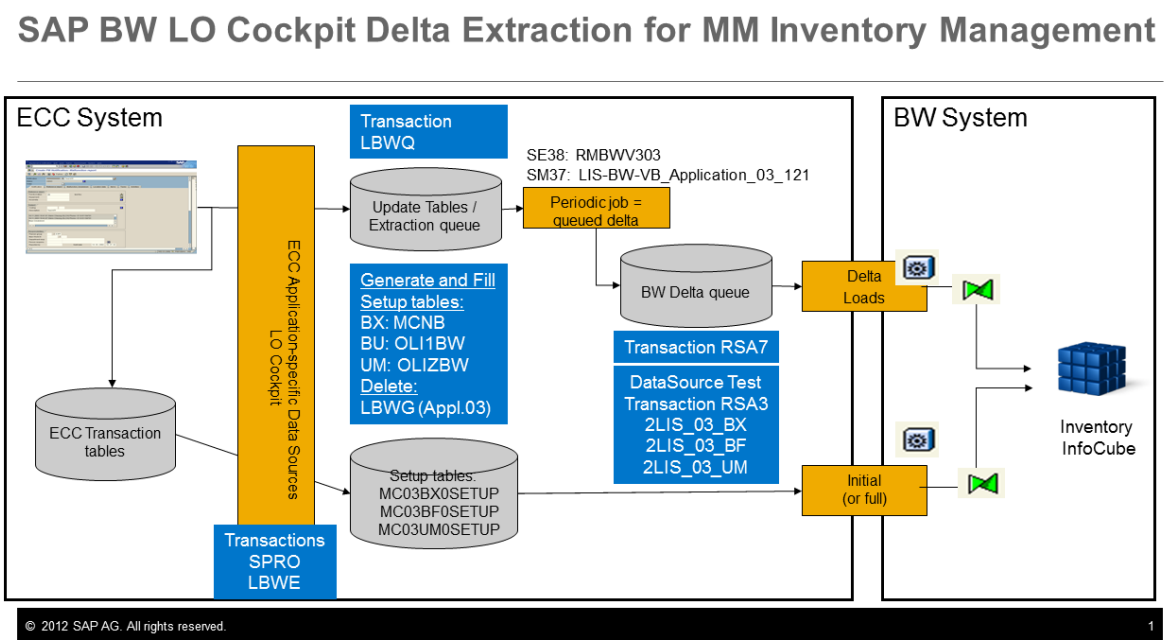
CALDAY between
t1=20120201
and
t2 = 20120530

RECORDTP in (0, 2) means
0: Deltas
2: Historic Transactions

Group by
0CALDAY

Picture 72: 'Movements'

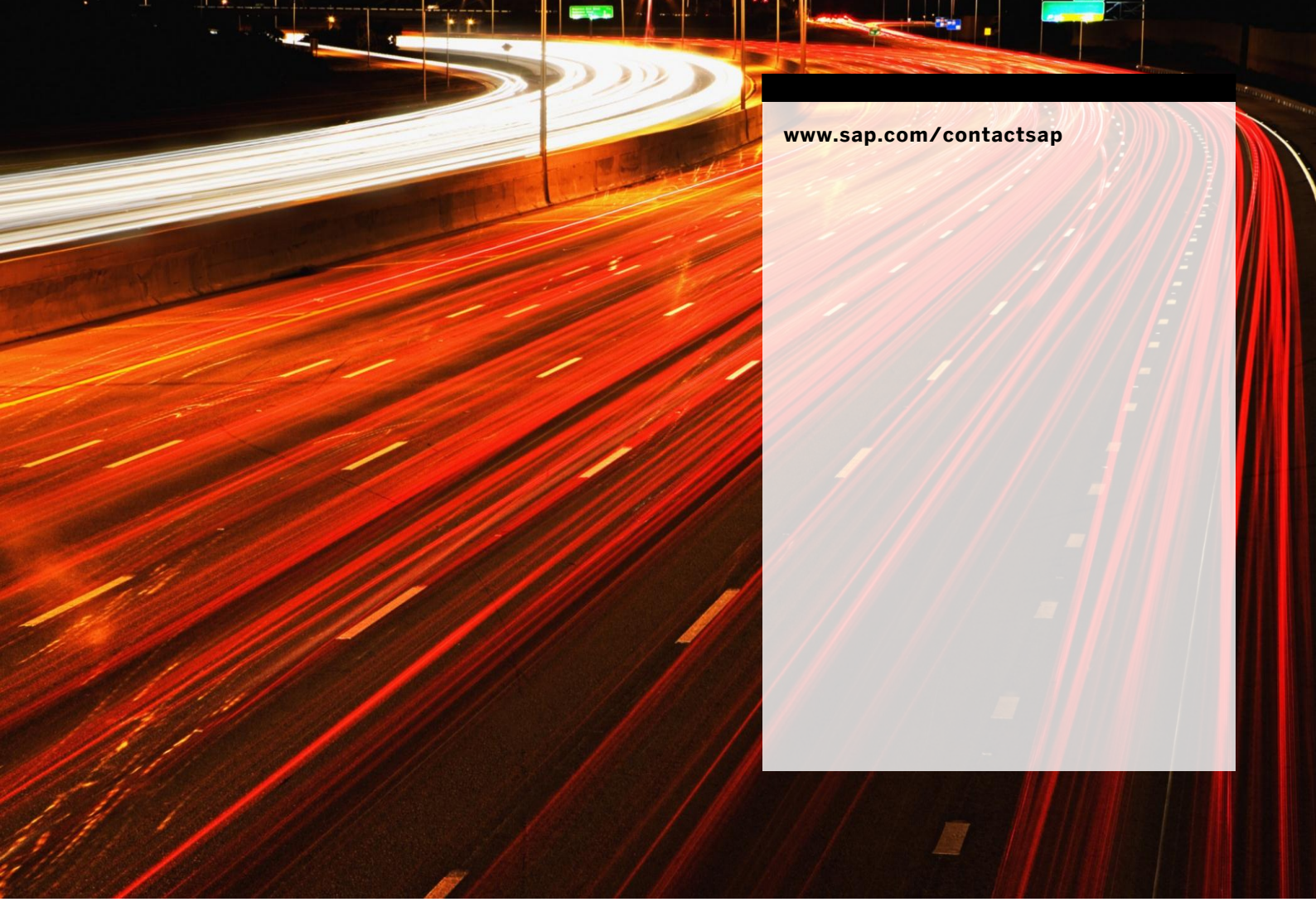
11.3 Overview BW LO Cockpit MM Inventory Management



Picture 73: Overview BW LO Cockpit

11.4 Notes

http://service.sap.com/sap/support/notes/1548125
http://service.sap.com/sap/support/notes/1679974
http://service.sap.com/sap/support/notes/1691932
http://service.sap.com/sap/support/notes/1683889
http://service.sap.com/sap/support/notes/1679099



www.sap.com/contactsap