



**How-to Guide
SAP NetWeaver '04**

How To... Handle Inventory Management Scenarios in BW

Version 1.3 – January 2005

**Applicable Releases:
SAP NetWeaver '04**

© Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1	OVERVIEW	2
2	BUSINESS SCENARIO	2
2.1	Inventory management with non-cumulative key figures	2
2.2	Inventory management with SnapShots	2
2.3	How to choose the right scenario	2
2.4	Advantages/ disadvantages of the SnapShot scenario	3
3	DETAILED DESCRIPTION	4
3.1	DataSources in R/3	4
3.2	Inventory management with non-cumulative key figures	4
3.2.1	InfoSources/InfoCube in BW	4
3.2.2	Upload sequence and compression	4
3.2.3	Marker update when uploading/compressing, executing a query	15
3.2.3.1	Marker update when uploading/compressing	15
3.2.3.2	Executing a query	16
3.2.4	Validity table	17
3.2.5	Important OSS notes	19
3.3	Inventory management with SnapShots and cumulative key figures	20
3.3.1	Prerequisite	20
3.3.2	Process overview	20
3.3.3	Step by step solution	22
3.3.3.1	Used Business Content InfoObjects	22
3.3.3.2	New Key figures	22
3.3.3.3	Create ODS Object ODSSNAP	24
3.3.3.4	Create InfoCube ICSNAP1- SnapShot monthly	25
3.3.3.5	Transfer rules for InfoSource 2LIS_03_BX, 2LIS_03_BF, 2LIS_03_UM	27
3.3.3.6	Create update rules from 2LIS_03_BX to ODS ODSSNAP → Process 1	27
3.3.3.7	Create update rules from 2LIS_03_BF to ODS ODSSNAP → Process 2	30
3.3.3.8	Create update rules from 2LIS_03_UM to ODS ODSSNAP → Process 2	33
3.3.3.9	Create a generic extractor on the active ODS Table of ODS ODSSNAP	36
3.3.3.10	Create update rules to update the SnapShot InfoCube from the ODS – Process 4	39
3.3.3.11	Create update rules to update the movements in the future – Process 5	41
3.3.3.12	Create an update rule to upload the movement in the past – Process 3	44
3.3.3.13	Create a Query with exceptional aggregation	47
3.4	Example for Implementation Scenario	51

1 Overview

In this document we aim to discuss the most important settings and procedures when implementing inventory management scenarios.

On the one hand, we will be focusing on the special features involved in implementing in the inventory management area with the use of non-cumulative key figures.

In the following, we will pay particularly close attention to sub areas that make up the special features of inventory management scenarios with non-cumulative key figures.

These are:

- ❑ The DataSources to be used and their sequence during data upload
- ❑ Special features in InfoCube compression
- ❑ Special features in query execution
- ❑ The validity table

On the other hand, we will discuss the special implementation scenarios in using SnapShots for the inventory management. This scenario will use cumulative key figures with an exception aggregation on time.

This document does not cover general customizing, for example, the definition of an InfoCube or the creation of a DataSource in an LO cockpit. A basic knowledge of LO extraction is required.

2 Business Scenario

2.1 Inventory management with non-cumulative key figures

The reporting requirement is to report the historical stock balances on a daily level and 90 percent of all materials were moved 1 time in the month.

The query-runtime is excellent. To reduce the data volume in the fact table, it is better to use the non-cumulative key figures.

2.2 Inventory management with SnapShots

The reporting requirement is to report the historical stock balance on a monthly level of all materials. 90 percent of all materials were moved 1 time in the month.

Due to the high amount of movements the query-runtime is bad. To reduce the query runtime of the non-cumulative InfoCube, the SnapShot Scenario should be implemented in parallel. (see 2.3)

2.3 How to choose the right scenario

The SnapShot loads data for every key figure and granularity of the InfoCube. This can end in a very high data volume and batch process time. The movements are not stored in the InfoProviders. An retrace of one special stock is not possible.

The non-cumulative key figures calculate stock during query runtime. The source of the calculation is the loaded movements and a special reference point. In general this means that the report can display stock on a date where no movement is loaded. This feature can end in a bad runtime of the query.

The decision to use the SnapShot Scenario is necessary, if the Query-Runtime of the non-cumulative Scenario is too bad.

If the SnapShot Scenario is chosen to reduce the Query-Runtime it should be taken in account that for recovery reason and for the ability to retrace the data it is only advisable to run an InfoCube with non-cumulative keyfigures with all the movements in parallel for a certain timeframe. For the administrator it is much easier to avoid, find and repair inconsistencies in the data. ¹

2.4 Advantages/ disadvantages of the SnapShot scenario

- Advantages
 - A MultiProvider with multiple PartCubes with constant on time characteristics can be used "Logical Partitioning" (see note 690475)
 - Deletion of facts over time can be easy realized
 - Query runtime is not influenced by the calculation of the stock
 - Condense/rollup of an InfoCube is faster
- Disadvantages
 - Special handling of the past and future movements in transfer/update rules
 - Monthly full upload of the from the ODS into the InfoCube can take a long time
 - One entry per key combination in the ODS can end up in a high amount of data and a long upload of the SnapShot (please check the necessary keys)
 - If the movements are deleted the ability to retrace the data is not available anymore

¹ Inconsistencies could happen if data in the PSA is deleted and the update-/ transferrules do not fulfill all the EndUser requirements or the coding is just wrong. A reinitialization of all the past data could be necessary. In the SnapShot Scenario the past data is nowhere stored.

3 Detailed description

3.1 DataSources in R/3

As of BW 3.0B and 4.0B (PI 2003.1), the next release up, the following DataSources are to be used in the extraction of an inventory scenario:²

- | | |
|---------------------------|--------------------|
| ❑ 2LIS_03_BX ³ | Material stock |
| ❑ 2LIS_03_BF | Material movements |
| ❑ 2LIS_03_UM | Revaluations |

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). At this moment, the opening stock is the operative stock in the source system. "At this moment" is the point in time at which the statistical setup ran for 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, as we will see below). 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 BW system. This DataSource provides the data as material documents (MCMSEG structure).

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. 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 quantities. Everything that is subsequently mentioned here about the upload sequence and compression regarding DataSource 2LIS_03_BF also applies to this DataSource. This means a detailed description is not required for the revaluation DataSource.

3.2 Inventory management with non-cumulative key figures

3.2.1 InfoSources/InfoCube in BW

As of 3.0B, SAP delivers InfoCube 0IC_C03 for reporting on R/3 inventory data in BW. This gets updated from the InfoSources 2LIS_03_BX, 2LIS_03_BF and 2LIS_03_UM.

3.2.2 Upload sequence and compression

These are the required steps:

Please make sure that the OSS notes 315880 and 353042 are applied in the R/3 system before you execute any data load. Otherwise BW enrichment (processing keys, etc.) won't take place.⁴

² The earlier solution with LIS structures S195 / S197 or S196 / S198 is not discussed here (these objects should only be used if you on R/3 release 3.1I - see OSS note 586163).

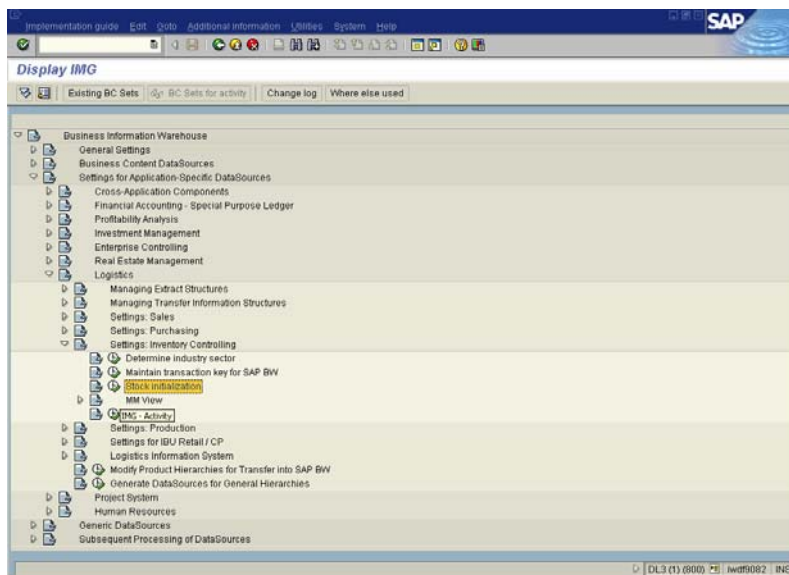
³ For availability see OSS note 588015

⁴ Please check in the SAP Service Marketplace if further notes exist.

1. Initialization of the current stock (opening balance) in the R/3 source system with DataSource 2LIS_03_BX (transaction code in R/3: MCNB). For example, you can make restrictions to materials or plant, providing not all balances are relevant. This provides also the possibility to schedule the report several times. In such cases the data is added to the setup table MC03BX0SETUP.⁵ Note: No material postings must be carried out during this process, since material movements influence the current stock. This run is not very time-intensive because the system does not read the individual documents from the very large MSEG table. Instead, it accesses the substantially smaller tables MARC, MARD and so on. Only if you use the option "Zero stock to be transferred" runtime increases rapidly. So use this option only if really necessary (see OSS notes 655798 and 728360 for this).

Activate the extract structure MC03BF0 for the corresponding DataSources 2LIS_03_BX/2LIS_03_BF in the LO cockpit (transaction code in R/3: LBWE) before initializing the opening balance (see OSS note 631345).

You can also find the initialization of the opening stock in transaction SBIW in the R/3 source system under the menu entry specified below:



After executing the menu entry you will come to the definition of the initialization run where you can restrict by plant, material and storage location. The definition screen is shown below. Please consider that the termination date is in the future (name of the report: RMCBINIT_BW, use transaction code MCNB!).

⁵ Setup tables can be deleted by transaction LBWG.

BW: Initialization of opening stocks in transfer structure

General control info

Name of run: STOCK_INIT ☒ New run

Termination date: 20.01.2003

Time of termination: 20:00:00 5000 No. of data recs per LUW

Data destination info

Transfer structur: 2LIS_03_BX

Data source info

Plant: to

Material: to

Storage location: to

☒ only valuated stocks

☐ All stocks (including non-valuat)

☐ Zero stock to be transferd

2. Setup of statistical data for material documents in the R/3 source system. Here, all historical material movements are to be set up that has lead to the opening stock in step 1 (this makes it later possible in BW to calculate historical stock values). Ideally you still keep the posting block during this run. If this is not possible you have to make sure that by **restrictions for posting date or the material document number you only select those material documents that have created the opening stock (step 1)**. Please consider that material movements that have been posted into the future for a date (field BUDAT) after the initial stock date are not getting selected if the upper restriction bound is exactly this date. In this case you have to choose an appropriate upper bound like 31st December 2099 **and to keep the posting block**. Also if you post new documents to a date (field BUDAT) earlier than the initial stock date you have to keep the posting block as time restrictions won't help to determine the right documents (maybe restrictions by document number help).

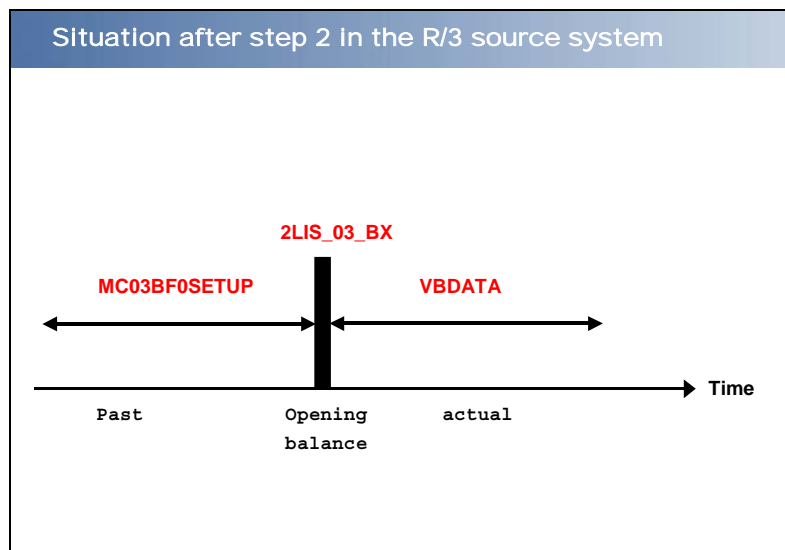
The setup of statistical data can also be executed in parallel by starting it separately for mutually exclusive selection criteria (for example, for a run for the previous month, for a run for the month before that, and so on). You should execute this program in the background.⁶ Similar to step 1 data is added to the setup table MC03BF0SETUP. It is possible by using appropriate selections in BW to load this missing data to BW with Full or Full Repair Requests after the BW environment is running. This enhances the period where historical stocks can be calculated.

⁶ In some cases, the setup of statistical data can take a very long time due to the volume of data involved. This means that, during the whole period, (newly created) material documents are written to the central update table VBDATA (if no posting block) without being received into the V3 update (step 7). Data is only extracted from the update table when the delta process of DataSource 2LIS_03_BF has been successfully initialized. For a very large data circulation there is also the risk that the central update table receives more data than it can process. We are talking about approximately 300,000 records here. To circumvent this danger, it is only advisable to set up statistical data for a small number of data records (for just a week or a day, for example). This keeps the time for the setup to a minimum. Material documents not present in the setup - but still required (for example, all material documents older than one week) - can be set up after step 8. Thus, they can be extracted via DataSource 2LIS_03_BF using one or more full (full repair if ODS) uploads (with selection criteria in the InfoPackage). It is also important to mention here that these requests **must** be compressed. You must carry out compression without marker update (compression and the marker update will be discussed later).

Activating the extract structure results in new material documents being included in the extraction queue (transaction LBWQ) or central update table VBDATA (transaction SM13).⁷ It has to be guaranteed that such records are not included in the setup table either by posting block or appropriate selections otherwise they get extracted double to BW.

The following source system constellation applies after steps 1 and 2:

- Current stock balance with DataSource 2LIS_03_BX in the setup table MC03BX0SETUP.
- Historical material documents with DataSource 2LIS_03_BF in the setup table MC03BF0SETUP belonging to extract structure MC03BF0.
- Postings done after the set up are found in the extraction queue or central update table. They can be later transferred into the BW using the delta queue (transaction RSA7), see step 7.



If there has been a complete posting block during both setups, the current stock appears in the setup table of DataSource 2LIS_03_BX with the previous postings appearing in the setup table MC03BF0SETUP. There are no entries in the extraction queue or central update table (VBDATA) here, since no further postings have been made in the meantime.

You can find the setup of statistical data under transaction SBIW in the R/3 system by following the menu entry shown in the following screenshot (or use transaction code OLI1BW, the report is RMCBNEUA):

⁷ For the available update methods in application 03 (Inventory Management) see OSS notes 505700, 580779 and 486784 (see also composite OSS note 586163 chapter 3.2.2, issue 1).

After executing the menu entry, a dialog box appears in which you must specify whether you want to carry out a setup for material movements (to be extracted using the DataSource 2LIS_03_BF, Report RMCBNEUA, transaction OLI1BW) or for revaluations (DataSource 2LIS_03_UM, Report RMCBNERP, transaction OLIZBW). In the previous scenario, we only looked at material movements. For this reason, the setup of revaluations is to be ignored here as well (setup runs analogously to that of material movements). After selecting the activity, a screen appears in which the setup is started:

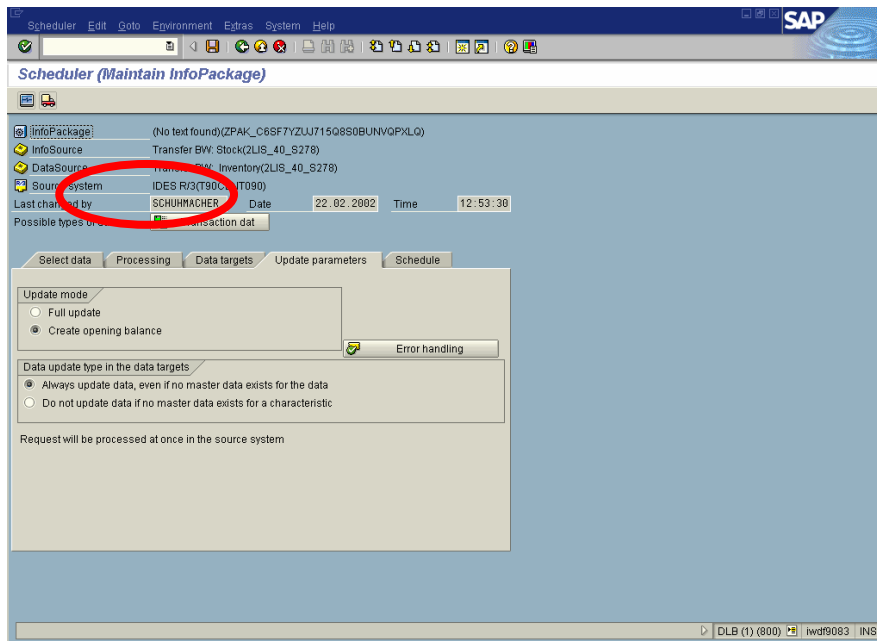
The screenshot shows the SAP 'Statistical Setup of Info Structures from Material Movements' dialog box. The window has a title bar with 'Program Edit Goto System Help' and the SAP logo. The main area is divided into three sections:

- Specifications for the target structures:** Contains a checked checkbox for 'Extraction structures BW'.
- Document data restriction:** Contains an 'Archiving run' section with two radio buttons: 'Selection by posting date' (selected) and 'Selection per material doc.'. Below these are input fields for 'Posting date' and 'Material document', each with a 'to' field.
- Control of the setup run:** Contains a 'Name of run' field with the value 'setup', a checked checkbox for 'New run?', and fields for 'Date of termination' (20.02.2002) and 'Time of termination' (09:35:00).

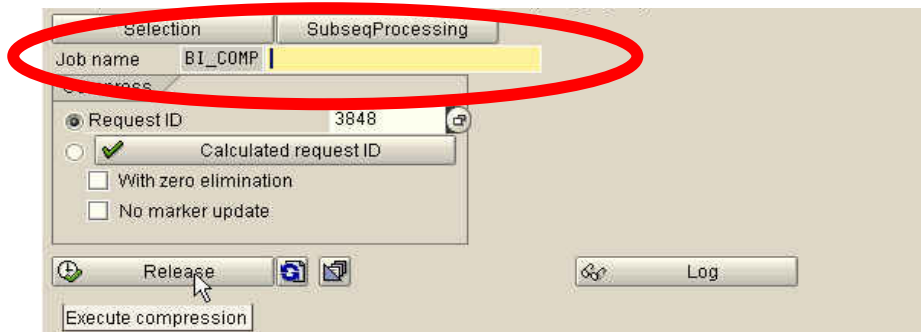
At the bottom, a status bar shows a green checkmark and the message 'Data source 2LIS_03_BF contains data still to be transferred'. On the right side of the status bar, it displays 'DL3 (1) (800)' and 'Iwdf9082 INS'.

Note: Here, you need to ensure either by posting date or by material document number, that all documents to be considered are already contained in the current stock (step 1). If you still have the posting block no restrictions are necessary. Only a required reduction of the data volume can force restrictions for the past.

3. Loading the opening stock balance (InfoSource 2LIS_03_BX) into InfoCube 0IC_C03. In the InfoPackage, choose the upload mode "Create opening balance" (see screenshot).

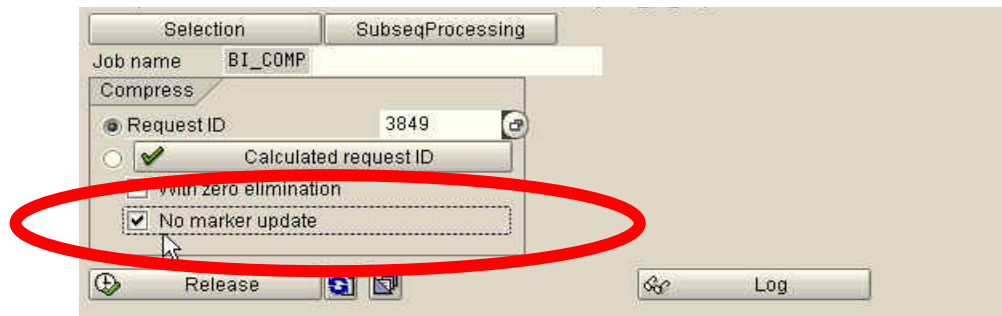


4. Compressing the request containing the opening stock that was just uploaded. Make sure the "No marker update" indicator is not set. Please consider note very carefully 643687 before you carry out the compression of requests in stock InfoCubes!



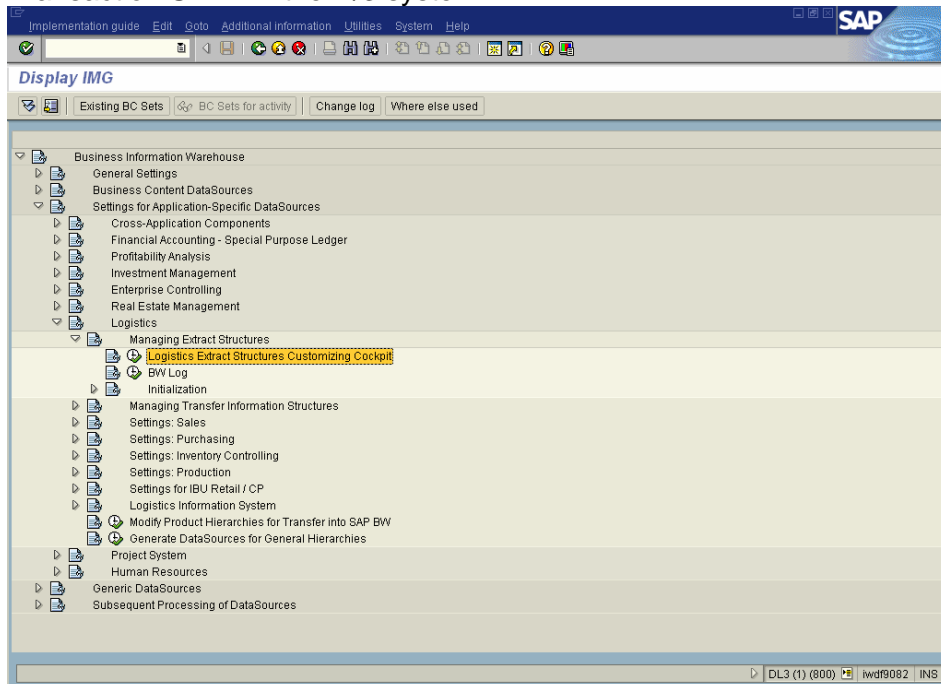
5. Loading the historical movements into the InfoCube 0IC_C03 via the DataSource / InfoSource 2LIS_03_BF. Choose the upload mode "Initializing the delta process" here.

6. After successfully uploading the historical material movements, the associated request has to be compressed. You must make sure the "No marker update" indicator is set. This is necessary because the historical material movements are already contained in the opening stock. We'll look more closely at markers later on.



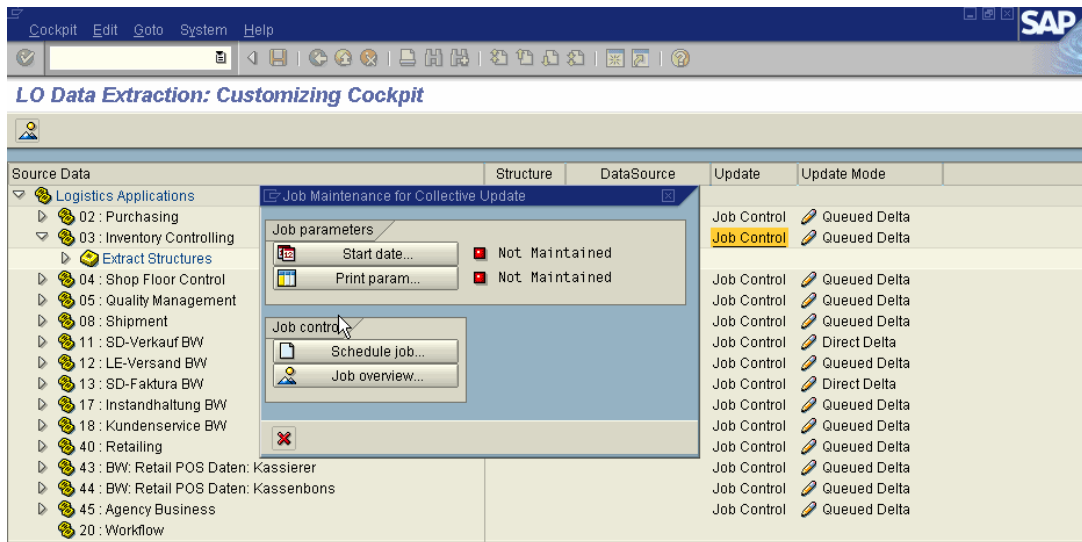
7. Material documents that appear in the system during this time (that is, between the end of step 1 and the successful initialization of DataSource 2LIS_03_BF) now appear in the extraction queue or central update table. You must not start the unserialized V3 update or queued delta control run before this point. Executing this run before successfully initializing the DataSource for the movements results in the records contained in the extraction queue or central update table being lost.⁸ During this run, data is written from the extraction queue or central update table into the delta queue (transaction code in R/3: RSA7). Only at this point is it available for extraction for the BW. Initialization of the run is described using the screenshots below:

Transaction SBIW in the R/3 system:



After executing the menu entry, the LO cockpit appears (transaction is LBWE):

⁸ Many customers also use program RSM13005 (schedule job periodically) to import data from an application from the central update table to the delta queue or RMBWV303 to write data from the extraction queue to the delta queue. In this case, also make sure that the program is not to be started before the initialization has been carried out successfully as this results in the loss of data records in the extraction queue or central update table.



Here, you can manually start the run for the different applications, or schedule it periodically (as is normally the case).

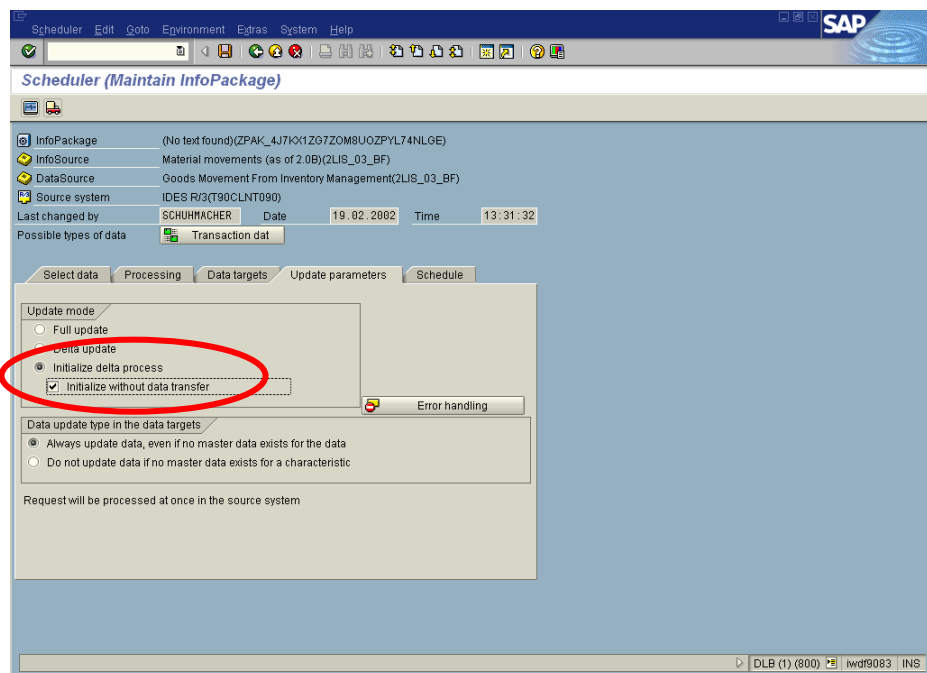
8. Successive (for example, daily) delta uploads can be started with the DataSource 2LIS_03_BF from this point in time on. These requests should also be compressed after successful validation. You need to use a marker update with this compression, meaning that the indicator must not be set.

Notes:

- You do not need to compress the requests directly after the upload. If you do compress afterwards, this has no effect on the compression settings. This means in all cases:
 - a) The request in which the opening stock was loaded must always be compressed with a marker update
 - b) The request in which the historical material documents were contained must always be compressed without a marker update and
 - c) Successive delta uploads must always be compressed with marker updates.

You only need to compress the request when you have loaded historical material movements (step 5). In this case, not compressing the requests for the opening balance and for the historical movements leads to false results in reporting. Compression is optional where no historical data has been transferred into the BW, though it is recommended for performance reasons. Please consider OSS note 655798 for compression of stock InfoCubes.

- You do not need to initialize the stock (steps 1 and 3). If you do not initialize, the system assumes a zero opening stock balance. You do not need to carry out a stock initialization if e.g. the following applies: The productive start of the R/3 system occurs at the same time as that of the BW system, and the existing stock (residing in other systems up until now) is imported in material document form into the R/3 system at the time of going live. This is because the stock is received as material documents. In this case, the stock is modeled at the going-live date in the form of a material inflow to precisely this date.⁹
- You can also model a stock scenario partially or completely using flat files. This allows you to define a flat file DataSource as an "opening balance". This is used to upload the opening stock and works in the same way as the R/3 DataSource 2LIS_03_BX. In addition, material movements are loaded via a "normal" DataSource. The above applies to the upload sequence and compression (steps taken in the R/3 system do not apply here of course).
- Where historical material movements are not required, the following changes are made to the procedure:
 - a. Step 2 is removed
 - b. In step 5, the delta process is initialized without data transfer (see screenshot)



- c. Step 6 is not required. The compression of the request from step 5 takes place with the successive delta requests. The marker has no importance for (only) this request since no data was transferred.
- **Don't update InfoCube OIC_C03 from an ODS layer.** Update the InfoCube and the ODS layer in parallel from the PSA layer. See composite note 586163 (chapter 5.c, Inventory Management and ODS) why it is recommended.

⁹ This means that usage of 2LIS_03_BX is not absolutely necessary (if you have all material documents in the source system). Especially for constellations that are mentioned in OSS notes 655798 and 586163 (5. Stock Analysis in BW, c) Tips and tricks, Enhancing the OIC_C03 InfoCube with additional characteristics) it is advised not to use 2LIS_03_BX.

3.2.3 Marker update when uploading/compressing, executing a query

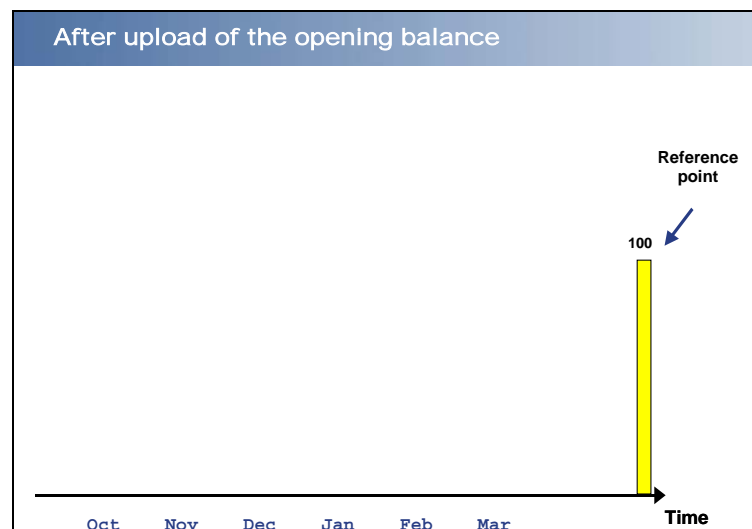
3.2.3.1 Marker update when uploading/compressing

We will use an example in this section to explain the procedure for a stock InfoCube when executing a query. The scenario is as follows:

- Current date: 31.03.2002
- You have set up an opening balance of 100 units on 01.01.2002 and loaded it into the stock InfoCube.
- Historical material movements from the three previous months (October 2001: 10 units; November 2001: 20 units; December 2001: 10 units) are loaded into the BW. The procedure here is the same as in steps 5 and 6 in section 3.2.2.
- Since this point, successive material movements have been transferred into the BW in the delta process. Delta requests transferred at the end of January (20 units) and February (10 units) were already compressed after successful validation, the last delta request from the end of March (10 units) is still in the InfoCube in uncompressed form.

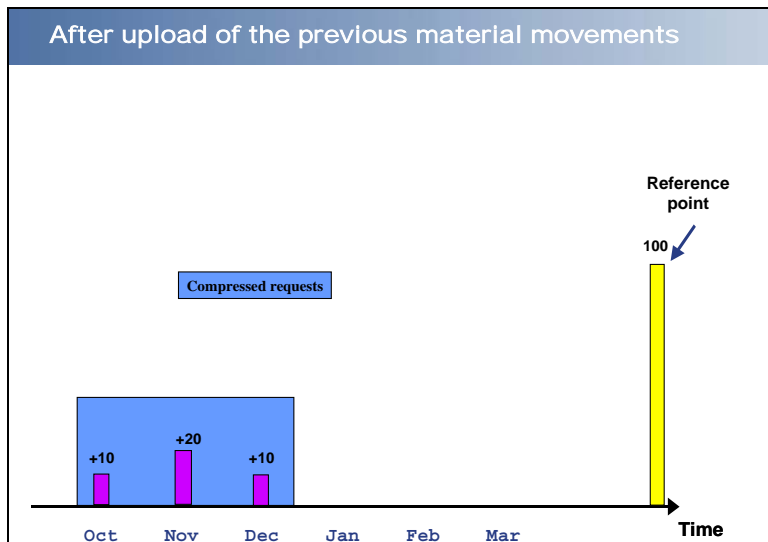
To help explain the role of the marker (= reference point), the different upload steps are considered over time.

After uploading the opening balance, the InfoCube looks like this:



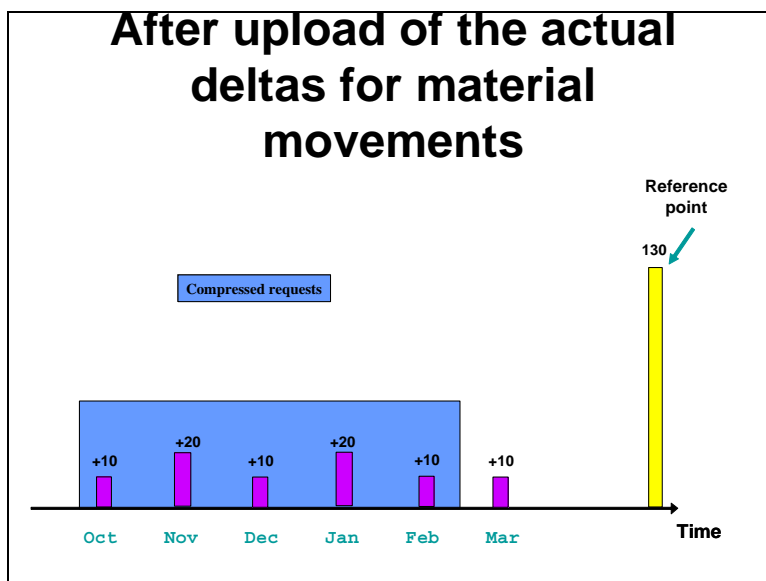
You can see that the opening stock is not assigned to the actual date, but posted to a point in infinity (0CALDAY= 31.12.9999, for example).

After the three previous months have been uploaded and compressed, the InfoCube content looks like this:



Note here that the marker value remains unchanged at 100 units. This can be achieved using the “No marker update” indicator during compression (see section 3.2.2, step 6). The marker is thus not changed.

After successively uploading deltas from January to March, of which only the first two are compressed, the InfoCube content has the following appearance:



Compressing the requests for January and February executes a marker update that can be seen by the marker now having the value 130 units. The values for March have not been included in the marker yet.

The following section will use this scenario to show how a stock balance is calculated when executing a query.

3.2.3.2 Executing a query

Let us assume that the stock balance is to be evaluated in a query at the end of January.

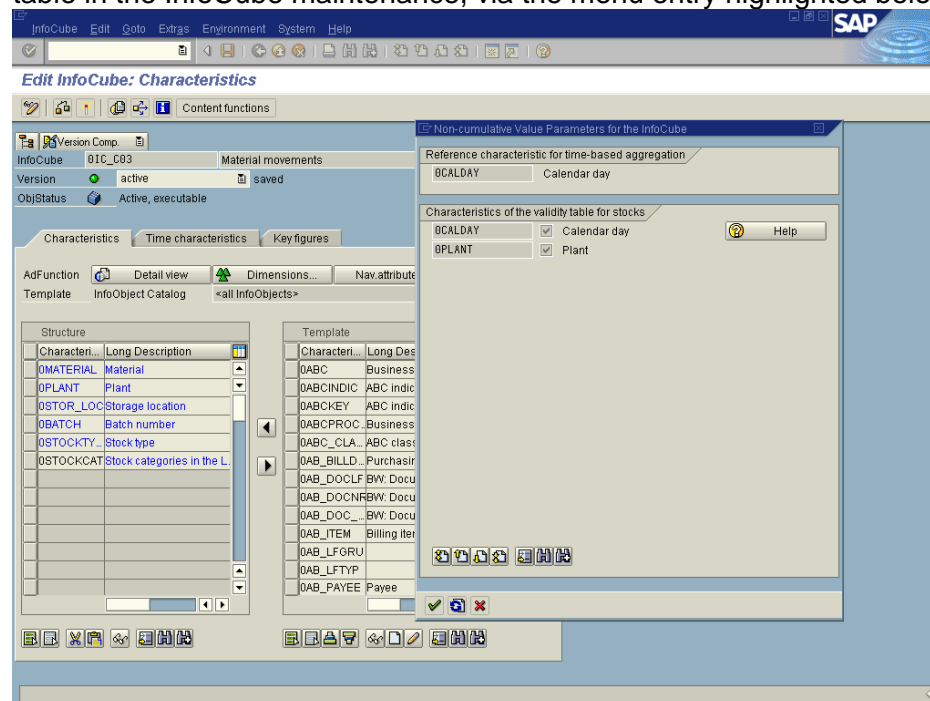
This is done by adding all values that have yet to be compressed to the marker value.

This value is called the “last value” and is 140 units for the described situation (140 = 130 marker units + 10 units from March). All values that are older than the requested value are now subtracted from this value. In our case, this means that 10 units from March and a further 10 units from February are subtracted, which leads to a stock balance of 120 at the end of January. Thus, the system firstly carries out a forward calculation. The system then calculates back from the resulting amount to the required point in time in the query.

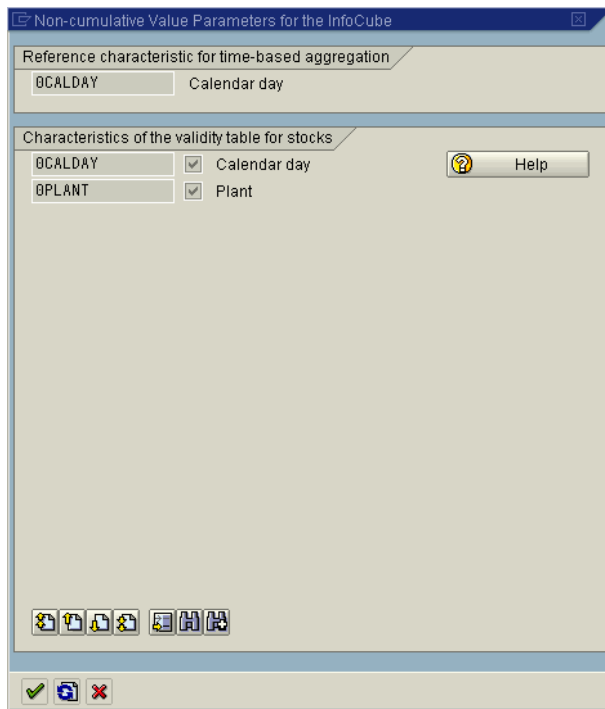
As a result of this procedure, we recommend you compress as many requests as possible so as to minimize the time required for forward calculations. This can even reach zero (when all InfoCube requests are compressed).

3.2.4 Validity table

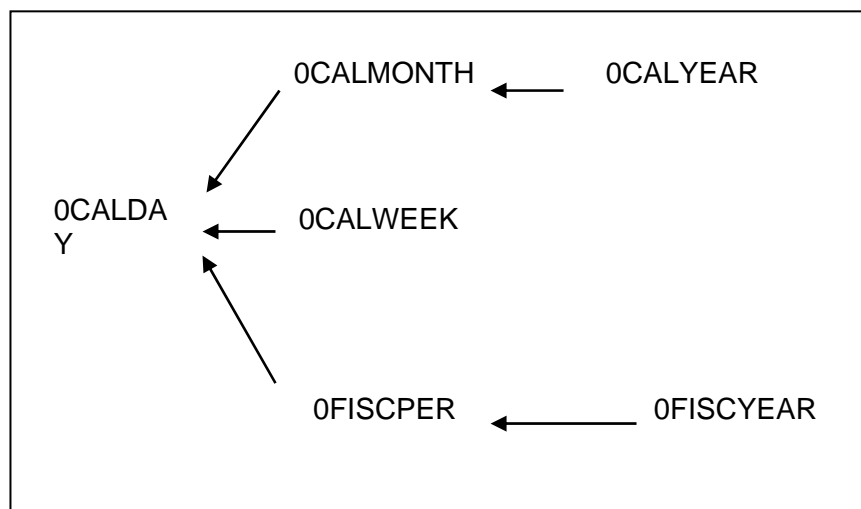
For InfoCubes with non-cumulative key figures, you need to maintain a validity table when creating the InfoCube. This table specifies the time interval for which the non-cumulatives are valid for a specific characteristic combination. You can find the validity table in the InfoCube maintenance, via the menu entry highlighted below.



For an InfoCube containing the characteristics 0CALDAY (and possibly other time characteristics), 0MATERIAL, 0PLANT, 0STOR_LOC, 0BATCH, 0STOCKTYPE and 0STOCKCAT, the dialog box for maintaining the table structure has the following appearance (0CALDAY and 0PLANT selected):



By default, the validity table only has the time dimension checked (that is, the time reference characteristic). The time reference characteristic is the time characteristic from which all other time characteristics can be derived. The hierarchy for deriving the time reference characteristic has the following appearance:



This hierarchy shows how you need to include the 0CALDAY time characteristic in an inventory management InfoCube even when you are only really interested in the two characteristics 0CALWEEK and 0FISCPER. Generally this means that a time reference characteristic is required from which all other time characteristics can be consistently derived. You can always include just one characteristic (for example, 0CALYEAR), meaning you do not need to include the characteristic 0CALDAY.

The validity table is automatically filled during the upload. If you leave the structure in its original state (that is, containing only the time reference characteristic), the lower validity interval limit is defined with the data record that is the first according to the time characteristic. Analogously, the upper limit is defined by the highest value according to

the time characteristic. For example, if data with document data was loaded into the InfoCube with values from 01.01.2002 to 15.02.2002 (assigned to the respective time characteristic), the validity interval is also determined by these two date values. You can extend the intervals by maintaining the table (transaction RSDV). Stock balances are displayed for requests that relate to this period. If you start a query that requests the stock balance for 16.02.2002 or later, the result is displayed in parenthesis (a blank value is displayed in BW 2.0B and 2.1C), since it lies outside the validity area (providing that the validity table was not manually extended using transaction RSDV).

In most cases, it is sufficient to include only the time dimensions in the validity table, that is, to leave the table in the state in which it is displayed. It is only worth including additional characteristics (for example, OPLANT) when certain conditions apply. If these conditions are not met, enhancing the table with additional characteristics leads to unexpected query results. If additional characteristics are included in this table, validity intervals are updated for all resulting characteristic combinations. **You should never use fine granular objects like material in the validity table due to performance reasons (besides the possibility of unexpected results in the query result).**

We will use an example to outline the conditions for when it is worth enhancing the validity table structure. In this example, you want to load material movements from different plants from separate systems into the BW system. For some reason the uploads may be delayed, for example, plant 2 may sometimes only be able to supply its data to the BW a few days after plant 1. Let us assume that the data is to be transferred from both plants into the BW at the end of every month. The assumption here would be that data from the previous month is already available at the beginning of a particular month. However uploads may be delayed, as mentioned before. If you want to avoid reporting on stock balances that are already old (since in our example plant 2 has not provided the changes in stock), you should include the plant in the validity table. In this case, plant 2 of our example does not display any balances for the previous month until these are actually transferred into the BW. The stock balances in plant 1 are displayed as expected. Thus, the table defines an individual validity interval for each plant.

By including additional characteristics in the validity table, the validity intervals are kept to a very detailed level. In this example, we have a plant and a time characteristic, for example the calendar day. An extreme case would be a validity table containing the characteristics calendar day, material and storage location. In this way, validity intervals are kept to a daily level for all possible material and storage location combinations. Where there were no movements for a specific day, storage location and material, a query does not display a balance for precisely this day, storage location and material, providing that material movements for the same characteristic combination (material and storage location) were not loaded on a later day.

3.2.5 Important OSS notes

All necessary OSS notes and additional information is available in the composite OSS note 586163. We strongly recommend reading this note carefully before starting the implementation. Additionally, OSS notes 315880/353042 and 637927/643687 are very important.

3.3 Inventory management with SnapShots and cumulative key figures

3.3.1 Prerequisite

First the “Inventory Management” DataSources/InfoSources and the corresponding Business Content InfoObjects have to be activated. This includes the InfoCube 0IC_C03 and the corresponding update- and transfer rules. The InfoCube 0IC_C03 will be used as backup and will be filled in parallel for data verifications.

The InfoSources (2LIS_03_BX, 2LIS_03_BF and 2LIS_03_UM) are connected to the corresponding LO cockpit extractors (see section 3.1).

Section 3.2.2 paragraphs 1 and 2 describe how the LO cockpit is used. Data can then be loaded into the BW system, as described in section 3.2.2 paragraphs 3 and 5. The only difference is the data is loaded in the non-cumulative InfoCube and the SnapShot ODS in parallel.

If the SnapShot Scenario will be implemented after the initialization of the non-cumulative InfoCube, this means the InfoCube 0IC_C03 is already filled with movements. The SnapShot InfoCube can be filled with past stock data via Business Process Designer (APD¹⁰) or DataMart Interface¹¹. This not part of the Howto paper.

In the following chapters are describing the usage of the SnapShot scenario on the pattern of the Business Content InfoCube 0IC_C03. Please remember that it is not recommended to use the SnapShot Scenario as a stand alone solution. (see 2.3) The reason of the usage of the SnapShot Scenario is to reduce the query-runtime for the comparison of the monthly end stock.

System basis settings like key order or indexes are not included in this paper.

Please implement the note 773823 before starting the implementation.

3.3.2 Process overview

Illustration 1 Upload Process Overview” offers an overview of the various load processes.

Process 1 and process 2 are the init or delta extraction, as made available by the LO cockpit. These are also used in this form for the SnapShot process. Please do **not** load the 2LIS_03_BF and the 2LIS_03_UM as a full upload into the SnapShot ODS ODSSNAP.

The data extract is written to a non-reportable ODS. It is crucial here that the date for initialization (process 1) is fixed to one fixed value. This value must not occur in the future. An example would be the month 12.9999. Due to sorting reasons in the Query a month in the future is used. All keyfigures in the ODS are updated in the addition mode.

The delta upload process 2 distinguishes whether the material movement is valid in the current month, in the future, or in a month in the past. For the current month, the month is fixed to the same month as for initialization: 12.9999, in this example. For movements

¹⁰ How to use the APD is described in the SAP documentation. (<http://help.sap.com>)

¹¹ The DataMart Interface can extract the data with Recordmode 1. This means the data has the status of the last compression.

in the future, the month for which the movement is valid is updated. For movements in the past, the month is fixed to the same month as for initialization. Additionally these movements are updated in parallel to the SnapShot InfoCube. This is in the process overview of process 3.

In process 3, which runs parallel to process 2, all movements for the past periods (months lower than the actual month, in this case) are selected. Furthermore, movements for all periods in between are multiplied. For example, if a movement is loaded to the current month 03.2004 via delta upload but this movement is valid for the month 01.2004, this movement also has to be multiplied for the month 02.2004. Three records are updated, one into the ODS for the current periods and 2 straight to the SnapShot InfoCube.

At the end of each period, all records from the ODS (filtered by 12.9999) are loaded to the SnapShot InfoCube from the ODS via full upload. (Process 4) This is the inventory of the current period.

In process 5, after each period change (the start of the next month in this example), all movements are read from the ODS with the new current period and converted to the fixed date 12.9999. Data first has to be activated in the ODS. Extraction then takes place on the basis of a generic extractor from the active ODS table.

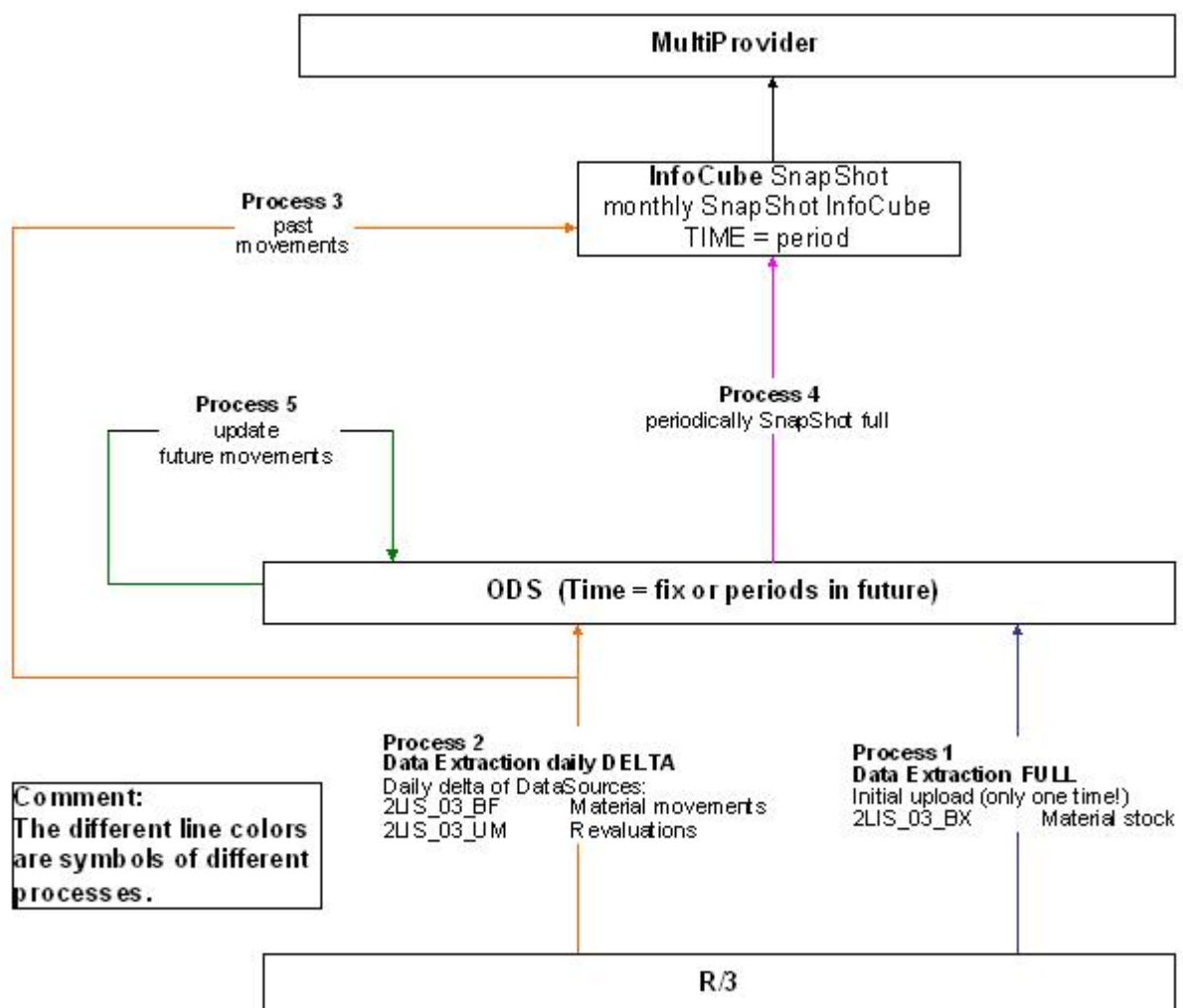


Illustration 1 Upload Process Overview

3.3.3 Step by step solution Used Business Content InfoObjects

The following Business Content Characteristics are needed from the InfoCube 0IC_C03:

Characteristics:

0CALMONTH	Calendar month
0CALYEAR	Calendar year
0DOC_DATE	Booking date (ODS)
0INDSPECSTK	Indicator: Valuation of Special Stock
0MATERIAL	Material
0PLANT	Plant
0STOR_LOC	Storage location
0BATCH	Batch number
0STOCKTYPE	Stock type
0STOCKCAT	Stock Categories

Units:

0LOC_CURRCY	Local currency
0BASE_UOM	Base Unit of Measure

3.3.3.2 New Key figures

For the SnapShot ODS and InfoCube three new key figures are created.

These key figures are the counterpart of the non-cumulative key figures used in the InfoCube 0IC_C03.

Key figure 1:

ZTOTALST Quantity Total Stock (Quantity, Unit/Currency: 0BASE_UOM)

The counterpart in the InfoCube 0IC_C03 is:

0TOTALSTCK	Quantity Total Stock
(Inflow: 0RECTOTSTCK	Receipt Quantity Total Stock)
(Outflow: 0ISSTOTSTCK	Issue Quantity Total Stock)

Key figure 2:

ZVALSTCKQ Quantity of valuated stock (Quantity, Unit/Currency: 0BASE_UOM)

The counterpart in the InfoCube 0IC_C03 is:

0VALSTCKQTY	Quantity of valuated stock
(Inflow: 0RECVALSTCK	Quantity Received into Valuated Stock)
(Outflow: 0ISSVALSTCK	Quantity Issued from Valuated Stock)

Key figure 3:

ZVALSTCKV Value of valuated stock (Amount, Unit/Currency: 0LOC_CURRCY)

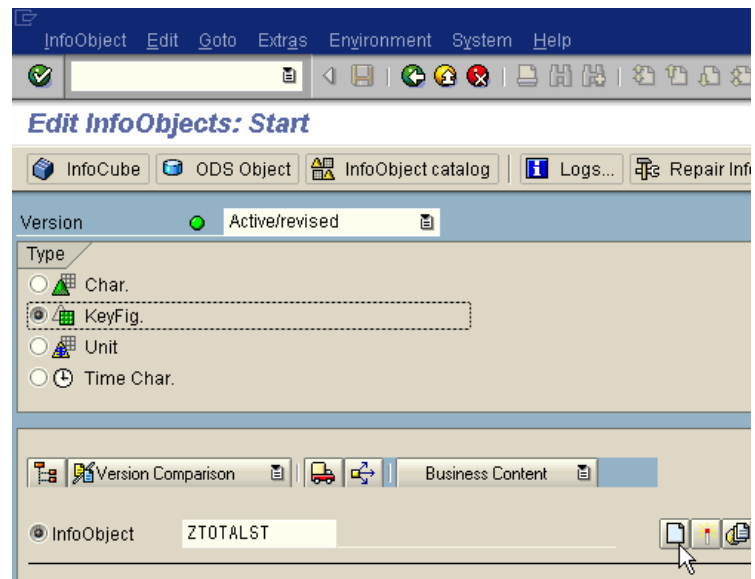
The counterpart in the InfoCube 0IC_C03 is:

0VALSTCKVAL	Value of valuated stock
(Inflow: 0RECVS_VAL	Value Received into Valuated Stock)
(Outflow: 0ISSVS_VAL	Value issued from valuated stock)

The following steps are describing the creation of one of these key figures.

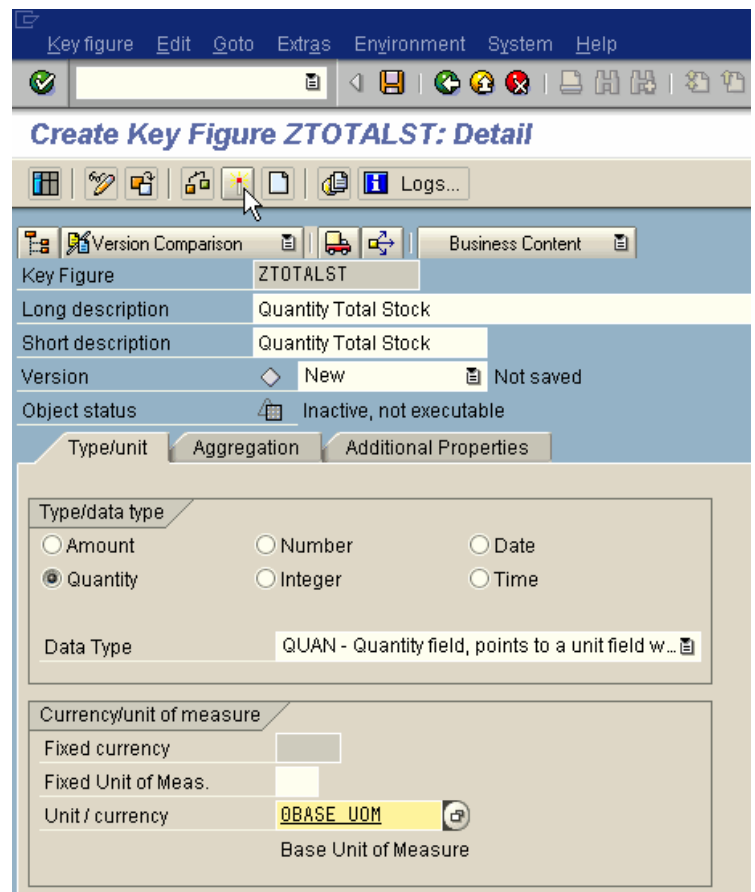
1. Transaction RSD1

Create key figure ZTOTALST



2. Activate the key figure

Attention: the key figure ZVALSTCKV is a Amount. The Unit/currency is 0LOC_CURRCY.



3.3.3.3 Create ODS Object ODSSNAP

3. Right mouse click on InfoArea

Create ODS object...

ODS object:

ODS name: ODSSNAP

ODS description: ODS SnapShot
month for OIC_C03

4. Fields of the ODS

Key fields:

0CALMONTH
0DOC_DATE
0INDSPECSTK
0MATERIAL
0PLANT
0STOR_LOC
0BATCH
0STOCKTYPE
0STOCKCAT

Key figures:

ZTOTALST
ZVALSTCKV
ZVALSTCKQ

Units:

0LOC_CURRCY
0BASE_UOM

Settings:

No reporting flag is set!
No navigational Attributes.
Set quality status to OK
automatically
Activate ODS object data
automatically
No update data targets from ODS
object automatically

Save and activate

The screenshot shows the SAP Administrator Workbench interface. The top menu bar includes Administrator Workbench, Edit, Goto, Tools, Environment, Settings, System, and Help. The main window is titled 'Administrator Workbench: Modeling'. On the left, the 'Modeling' tree shows 'InfoProvider' expanded, with 'Inventory Man...' selected. A right-click context menu is open, showing options like 'Create...', 'Rename...', 'Delete', 'Create InfoCube...', 'Create ODS Object...', 'Create MultiProvider...', 'Create InfoSet...', and 'Insert Characteristic as Data Target...'. The 'Create ODS Object...' option is highlighted.

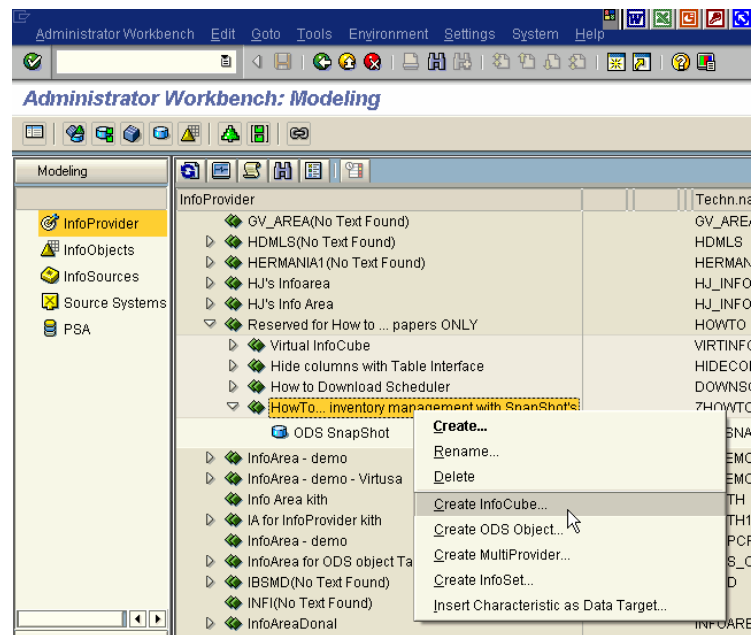
Below the screenshot, a table displays the configuration for the ODS object 'ODSSNAP'.

ODS Object	Techn. name / value	O...	Data...	L...	Key
ODS SnapShot month for OIC	ODSSNAP				
▼ Status Info					
Version	△ Revised				
Save	□ Not saved				
Status	☑ Active, executable				
▼ Settings					
▼ Key fields					
Calendar Year/Month	0CALMONTH	NUMC	06		
Document Date	0DOC_DATE	DATS	08		
Indicator: Valuation of S	0INDSPECSTK	CHAR	01		
Material	0MATERIAL	CHAR	19		
Plant	0PLANT	CHAR	04		
Storage location	0STOR_LOC	CHAR	04		
Batch number	0BATCH	CHAR	10		
Stock type	0STOCKTYPE	CHAR	01		
Stock categories in the	0STOCKCAT	CHAR	01		
▼ Data Fields					
Quantity Total Stock	ZTOTALST	QUAN	09	Qua	
Value of valuated stock	ZVALSTCKV	CURR	09	Ami	
Quantity of valuated sto	ZVALSTCKQ	QUAN	09	Qua	
Base Unit of Measure	0BASE_UOM	UNIT	03		
Local currency	0LOC_CURRCY	CUKY	05		
▼ NavAttribute					

3.3.3.4 Create InfoCube ICSNAP1- SnapShot monthly

5. Right mouse click on InfoArea

Create InfoCube...



6. InfoCube: ICSNAP1
Name: SnapShot - monthly for
OIC_C03

InfoCube - SnapShot

Characteristics:

0INDSPECSTK
0MATERIAL
0PLANT
0STOR_LOC
0BATCH
0STOCKTYPE
0STOCKCAT

Time Characteristics:

0CALMONTH
0CALYEAR

Key figures:

ZTOTALST
ZVALSTCKV
ZVALSTCKQ

Units:

0LOC_CURRCY
0BASE_UOM

7. Create for every Characteristic a Dimension and assign the characteristic as LINE-ITEM dimension.

Dimension	Long description	Line Item	Card
ICSNAP11	0INDSPECSTK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP12	0MATERIAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP13	0PLANT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP14	0STOR_LOC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP15	0BATCH	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP16	0STOCKTYPE	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICSNAP17	0STOCKCAT	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Create Delete ID -> Text

Fixed dimension		
Data Package	ICSNAP1P	Data Package
Time	ICSNAP1T	Time
Unit	ICSNAP1U	Unit

8. Create and assign the dimensions

Every dimension is a LINE-ITEM

Activate the InfoCube

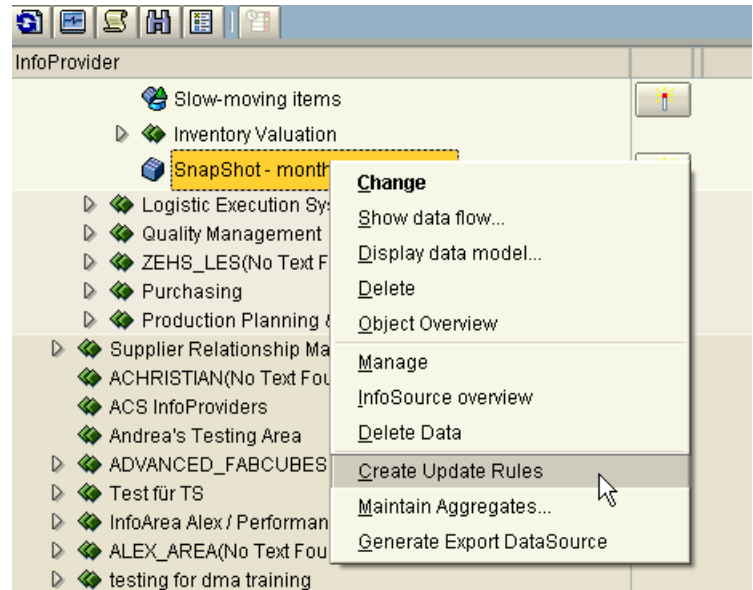
3.3.3.5 Transfer rules for InfoSource 2LIS_03_BX, 2LIS_03_BF, 2LIS_03_UM

9. The transfer rules are the same as in the business content provided.

Save and activate the transfer or communication structure

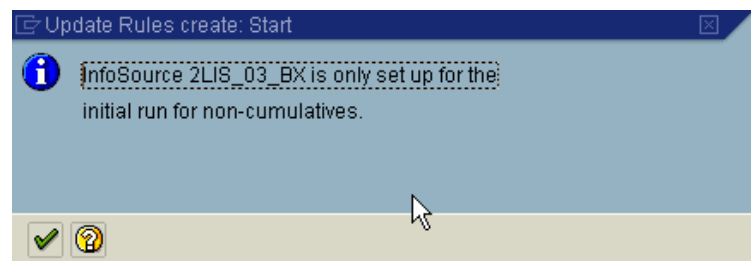
3.3.3.6 Create update rules from 2LIS_03_BX to ODS ODSSNAP → Process 1

10. Create update rule

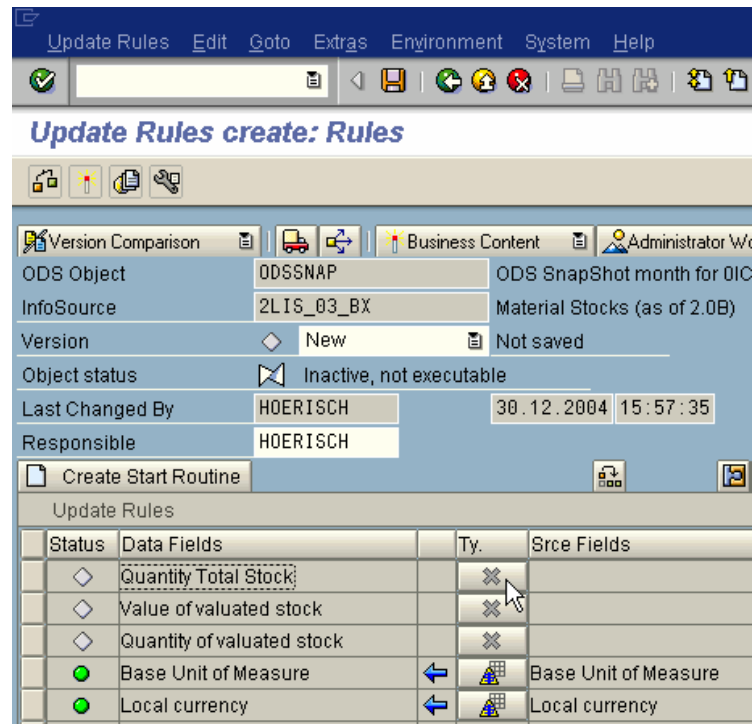


11. Please ignore the message and warnings:

Press return (2 times)



12. Map the data fields and source field:



13. Create Routine:

Data field:
Quantity Total Stock

Source field:
0CPQUABU
BW: Amount in base unit

```
* all stocks are considered
IF COMM_STRUCTURE-cpquabu <> 0.
* result value of the routine
  RESULT = COMM_STRUCTURE-cpquabu.
* if the returncode is not equal zero, the result
will not be updated
  RETURNCODE = 0.
ELSE.
  RETURNCODE = 4.
ENDIF.
* if abort is not equal zero, the update process
will be canceled
```

ABORT = 0.

14. Create Routine:

Data field:
Value of valuated stock

Source field:
0CPPVLC
BW: Purchase Value in Local
Currency

```
* only valuated stock is considered
* see OSS notes 630254 and 655798
IF COMM_STRUCTURE-stockcat IS INITIAL OR
( COMM_STRUCTURE-stockcat CA 'EQ' AND
  COMM_STRUCTURE-indspecstk CA 'AM' ).
* result value of the routine
  RESULT = COMM_STRUCTURE-cppvlc.
* if the returncode is not equal zero, the result
will not be updated
  RETURNCODE = 0.
ELSE.
  RETURNCODE = 4.
ENDIF.
* if abort is not equal zero, the update process
will be canceled
```

ABORT = 0.

15. Create Routine:

Data field:
Quantity of valuated stock

Source field:

```
* only valuated stock is considered
* see OSS notes 630254 and 655798
IF COMM_STRUCTURE-stockcat IS INITIAL OR
( COMM_STRUCTURE-stockcat CA 'EQ' AND
  COMM_STRUCTURE-indspecstk CA 'AM' ).
* result value of the routine
  RESULT = COMM_STRUCTURE-cpquabu.
* if the returncode is not equal zero, the result
```

0CPQUABU
BW: Amount in base unit of measure

will not be updated

RETURNCODE = 0.

ELSE.

RETURNCODE = 4.

ENDIF.

* if abort is not equal zero, the update process
will be canceled

ABORT = 0.

16. Create an update rule for the
calendar month:

Press Detail

Status	Data Fields	Ty	Src Fields
●	0CPPVLC BW: Purchase Value ...	←	0CPPVLC BW: Purchase Va
●	0LOC_CURRCY Local currency	←	0LOC_CURRCY Local cur...
●	0CPQUABU BW: Amount in bas...	←	0CPQUABU BW: Amount in
●	0BASE_UOM Base Unit of Measu	←	0BASE_UOM Base Unit of

17. Select key fields

Select blank

for calendar year/month

Key Field	Method	Source	Auto...
0BATCH Batch number	←	0BATCH Batch number	
0INDSPECSTK Indicato...	←	0INDSPECSTK Indicat...	
0MATERIAL Material	←	0MATERIAL Material	
0PLANT Plant	←	0PLANT Plant	
0STOCKCAT Stock cate...	←	0STOCKCAT Stock cat...	
0STOCKTYPE Stock type	←	0STOCKTYPE Stock ty...	
0STOR_LOC Storage lo...	←	0STOR_LOC Storage l...	
0CALMONTH Calendar ...	blank		

18. Create a routine

Name:

Update rule for CALMONTH

Routine:

19. Add the routine:
set for all records the CALMONTH to
12.9999

*\$\$ begin of routine - insert your code only below
this line * *

* fill the internal table "MONITOR", to make
monitor entries

* result value of the routine

RESULT = '999912'.

* if the returncode is not equal zero, the result will not be updated

RETURNCODE = 0.

* if abort is not equal zero, the update process will be canceled

ABORT = 0.

20. **Activate** the update rules

3.3.3.7 Create update rules from 2LIS_03_BF to ODS ODSSNAP → Process 2

21. Create update rules for ODS:
ODSSNAP

Ignore the message with **RETURN**.

22. Create the update rule for the
key field

Calendar Year/Month

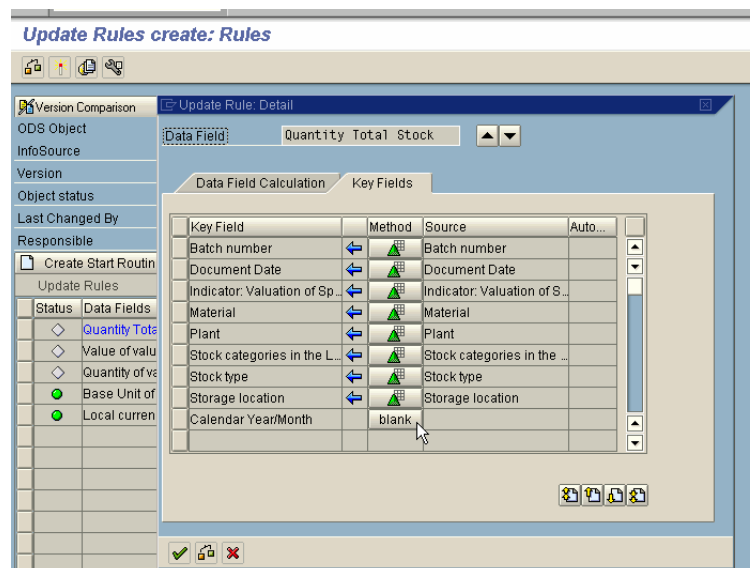
Name:

Update rule for CALMONTH

Transfer this routine to all key
figures

Comment: The Document date is updated. This means the Posting date of each delta upload is updated in the ODS. This can end into a high amount of rows in the ODS. If the number of rows is too high and can not be handled, the Document date should be set to BLANK in the update rule.

23. Routine CALMONTH.



PROGRAM UPDATE_ROUTINE.

*\$\$ begin of global - insert your declaration only
below this line *-*

* TABLES: ...

DATA: 1_month type /BIC/AODSSNAP00-CALMONTH.

*\$\$ end of global - insert your declaration only
before this line *-*

FORM compute_key_field

TABLES MONITOR STRUCTURE RSMONITOR "user
defined monitoring

USING COMM_STRUCTURE LIKE /BIC/CS2LIS_03_BF
RECORD_NO LIKE SY-TABIX
RECORD_ALL LIKE SY-TABIX
SOURCE_SYSTEM LIKE RSUPDSIMULH-LOGSYS

CHANGING RESULT LIKE /BIC/AODSSNAP00-CALMONTH

RETURNCODE LIKE SY-SUBRC

ABORT LIKE SY-SUBRC. "set ABORT <> 0 to

cancel update

*


```

*$$$ begin of routine - insert your code only below
this line          *_*
* fill the internal table "MONITOR", to make
monitor entries
* get month from movement

move COMM_STRUCTURE-PSTNG_DATE(6) to result.
move sy-datum(6) to l_month.

if l_month >= result. " movement in the past
RESULT = '999912'.
else. " movement in future
* move COMM_STRUCTURE-PSTNG_DATE(6) to result.
endif.

* if the returncode is not equal zero, the result
will not be updated
RETURNCODE = 0.
* if abort is not equal zero, the update process
will be canceled
ABORT = 0.

*$$$ end of routine - insert your code only before
this line          *_*
*

```

24. Create update rule for key figure

ZTOTALST

Quantity Total Stock

Update type is ADDITION

```

*****
* Please check Processkeys needs to consider!
*****
IF COMM_STRUCTURE-cpquabu <> 0.
* INFLOW
  IF ( COMM_STRUCTURE-processkey EQ '100' "Other
Issues
    OR COMM_STRUCTURE-processkey EQ '101'
"Returns / Vendor
    OR COMM_STRUCTURE-processkey EQ '104'
"Material Transfer
    OR COMM_STRUCTURE-processkey EQ '105' "Stock
Adjustment InvD
    OR COMM_STRUCTURE-processkey EQ '106' "Stock
Adjustment Other
    OR COMM_STRUCTURE-processkey EQ '110' ) "Issues
from Stock Transfers
    AND COMM_STRUCTURE-cppv1c <> 0.

    RESULT = COMM_STRUCTURE-CPQUABU * -1.
* OUTFLOW
  elseif ( COMM_STRUCTURE-processkey EQ '000'
"Other Receipts
    OR COMM_STRUCTURE-processkey EQ '001' "Goods
Receipt / Vendor
    OR COMM_STRUCTURE-processkey EQ '004'
"Material Transfer / Receipt
    OR COMM_STRUCTURE-processkey EQ '005' "Stock
Adjustment InvD
    OR COMM_STRUCTURE-processkey EQ '006' "Stock
Adjustment Other
    OR COMM_STRUCTURE-processkey EQ '010' )
"Receipt from Stock Transfer
    AND COMM_STRUCTURE-cppv1c <> 0.

    RESULT = COMM_STRUCTURE-CPQUABU.

ENDIF.
ENDIF.

```

25. Create update rule for key figure

ZVALSTCKV

Value of valuated stock

Update type is ADDITION

```
*****
* Please check Processkeys needs to consider!
*****
* only valuated stock is considered
* see OSS notes 630254 and 655798
  IF COMM_STRUCTURE-stockcat IS INITIAL OR
    ( COMM_STRUCTURE-stockcat CA 'EQ' AND
      COMM_STRUCTURE-indspecstk CA 'AM' ).
* INFLOW
  IF ( COMM_STRUCTURE-processkey EQ '100' "Other
Issues
    OR COMM_STRUCTURE-processkey EQ '101'
"Returns / Vendor
    OR COMM_STRUCTURE-processkey EQ '104'
"Material Transfer
    OR COMM_STRUCTURE-processkey EQ '105' "Stock
Adjustment InvD
    OR COMM_STRUCTURE-processkey EQ '106' "Stock
Adjustment Other
    OR COMM_STRUCTURE-processkey EQ '110' ) "Issues
from Stock Transfers
    AND COMM_STRUCTURE-cppvlc <> 0.

    RESULT = COMM_STRUCTURE-CPPVLC * -1.
* OUTFLOW
  elseif ( COMM_STRUCTURE-processkey EQ '000'
"Other Receipts
    OR COMM_STRUCTURE-processkey EQ '001' "Goods
Receipt / Vendor
    OR COMM_STRUCTURE-processkey EQ '004'
"Material Transfer / Receipt
    OR COMM_STRUCTURE-processkey EQ '005' "Stock
Adjustment InvD
    OR COMM_STRUCTURE-processkey EQ '006' "Stock
Adjustment Other
    OR COMM_STRUCTURE-processkey EQ '010' )
"Receipt from Stock Transfer
    AND COMM_STRUCTURE-cppvlc <> 0.

    RESULT = COMM_STRUCTURE-CPPVLC.

ENDIF.
ENDIF.
```

26. Create update rule for key figure

ZVALSTCKQ

Quantity of valuated stock

Update type is ADDITION

```
* only valuated stock is considered
* see OSS notes 630254 and 655798
  IF COMM_STRUCTURE-stockcat IS INITIAL OR
    ( COMM_STRUCTURE-stockcat CA 'EQ' AND
      COMM_STRUCTURE-indspecstk CA 'AM' ).
*****
* Please check Processkeys needs to consider!
*****
  IF COMM_STRUCTURE-cpquabu <> 0.
* INFLOW
  IF ( COMM_STRUCTURE-processkey EQ '100' "Other
Issues
    OR COMM_STRUCTURE-processkey EQ '101'
"Returns / Vendor
    OR COMM_STRUCTURE-processkey EQ '104'
"Material Transfer
    OR COMM_STRUCTURE-processkey EQ '105' "Stock
Adjustment InvD
    OR COMM_STRUCTURE-processkey EQ '106' "Stock
Adjustment Other
    OR COMM_STRUCTURE-processkey EQ '110' ) "Issues
```

```

from Stock Transfers
AND COMM_STRUCTURE-cppv1c <> 0.

RESULT = COMM_STRUCTURE-CPQUABU * -1.
* OUTFLOW
elseif ( COMM_STRUCTURE-processkey EQ '000'
"Other Receipts
OR COMM_STRUCTURE-processkey EQ '001' "Goods
Receipt / Vendor
OR COMM_STRUCTURE-processkey EQ '004'
"Material Transfer / Receipt
OR COMM_STRUCTURE-processkey EQ '005' "Stock
Adjustment InvD
OR COMM_STRUCTURE-processkey EQ '006' "Stock
Adjustment Other
OR COMM_STRUCTURE-processkey EQ '010' )
"Receipt from Stock Transfer
AND COMM_STRUCTURE-cppv1c <> 0.

RESULT = COMM_STRUCTURE-CPQUABU.

ENDIF.
ENDIF.
ENDIF.

```

3.3.3.8 Create update rules from 2LIS_03_UM to ODS ODSSNAP → Process 2

27. Create update rules

2LIS_03_UM to ODS ODSSNAP

Ignore the message with RETURN.

28. Set the stock key figures
ZTOTALST, ZVALSTCKQ and the
corresponding unit "base unit of
measure": to NO UPDATE

29. Create the update rule for the
key field

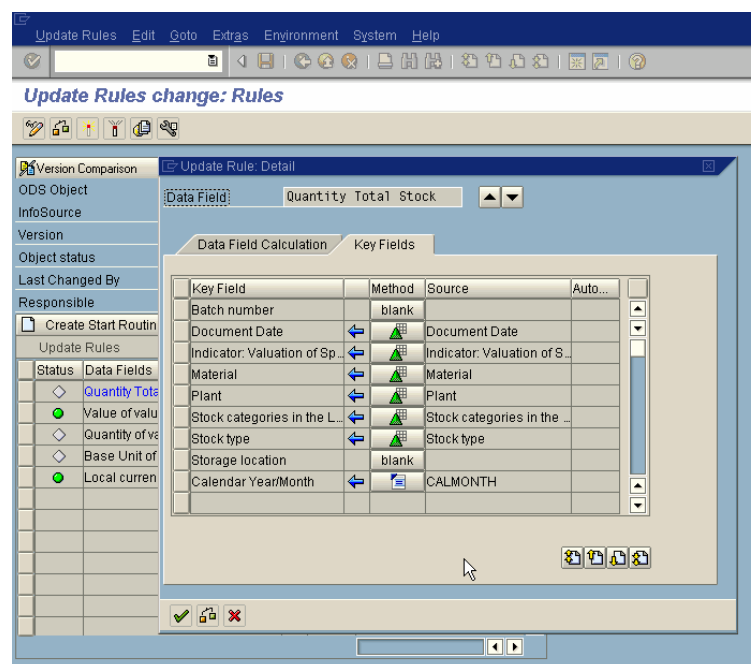
Calendar Year/Month

Name:

Update rule for CALMONTH

Transfer this routine to all key
figures

Comment: The Document date is updated. This means the Posting date of each delta upload is updated in the ODS. This can end into a high amount of rows in the ODS. If the number of rows is to high and can not be handled, the Document date should set to BLANK in the update rule.



30. Create a routine for 0CALMONTH. Set all future records to the corresponding month. The past and actual records are set to '999912'.

```

PROGRAM UPDATE_ROUTINE.
*$$$ begin of global - insert your declaration only
below this line  *_*
* TABLES: ...
DATA: 1_month type /BIC/AODSSNAP00-CALMONTH.
*$$$ end of global - insert your declaration only
before this line  *_*

FORM compute_key_field
TABLES  MONITOR STRUCTURE RSMONITOR "user
defined monitoring
USING   COMM_STRUCTURE LIKE /BIC/CS2LIS_03_UM
        RECORD_NO LIKE SY-TABIX
        RECORD_ALL LIKE SY-TABIX
        SOURCE_SYSTEM LIKE RSUPDSIMULH-LOGSYS
CHANGING RESULT LIKE /BIC/AODSSNAP00-CALMONTH
        RETURNCODE LIKE SY-SUBRC
        ABORT LIKE SY-SUBRC. "set ABORT <> 0 to
cancel update
*
*$$$ begin of routine - insert your code only below
this line  *_*
* fill the internal table "MONITOR", to make
monitor entries
* result value of the routine
* get month from movement

move COMM_STRUCTURE-PSTNG_DATE(6) to result.
move sy-datum(6) to 1_month.

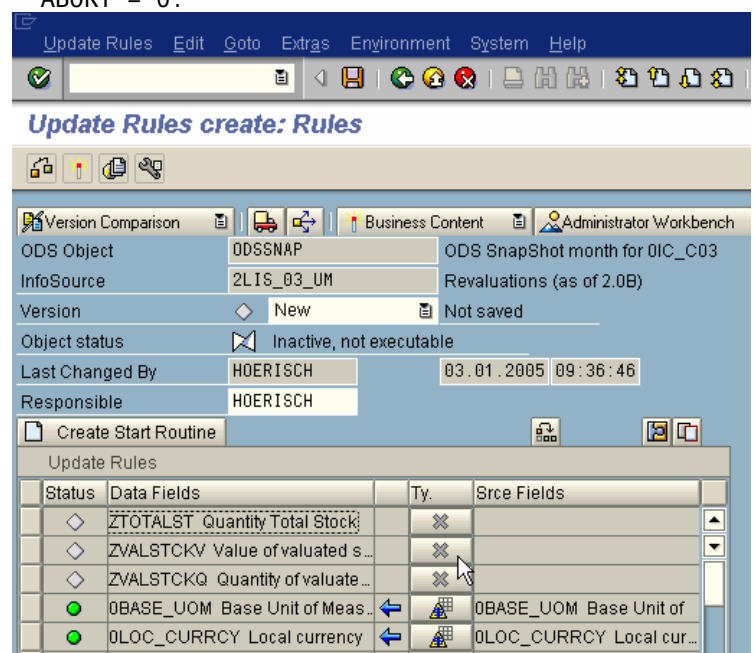
if 1_month >= result. " movement in the past
RESULT = '999912'.
else. " movement in future
* move COMM_STRUCTURE-PSTNG_DATE(6) to result.
endif.

* if the returncode is not equal zero, the result
will not be updated
RETURNCODE = 0.
* if abort is not equal zero, the update process
will be canceled

ABORT = 0.

```

31. Change the update of the key figure ZVALSRKV to routine



32. Add the routine for key figure
ZVALSRKV

Set the update mode of the key
figure to Addition

```
* fill the internal table "MONITOR", to make
monitor entries
* to add the value
IF COMM_STRUCTURE-PROCESSKEY = '050' "Sonstiger
Zugang / Umbewertung
OR COMM_STRUCTURE-PROCESSKEY = '051'
"Preisänderung + / Umbewertung
OR COMM_STRUCTURE-PROCESSKEY = '052'.
"Rechnungsprüf. + / Umbewertung
RESULT = COMM_STRUCTURE-CPPVLC. " add value
RETURNCODE = 0.

elseif COMM_STRUCTURE-PROCESSKEY = '150' "Sonstiger
Abgang/Umbewertung
OR COMM_STRUCTURE-PROCESSKEY = '151'
"Preisänderung - / Umbewertung
OR COMM_STRUCTURE-PROCESSKEY = '152'.
"Rechnungsprüf. - / Umbewertung
* result value of the routine
RESULT = COMM_STRUCTURE-CPPVLC * -1. " subtract
the value

* if the returncode is not equal zero, the result
will not be updated
RETURNCODE = 0.
ELSE.
* if the returncode is not equal zero, the result
will not be updated
RETURNCODE = 4.
ENDIF.
```

3.3.3.9 Create a generic extractor on the active ODS Table of ODS ODSSNAP

33. Go to transaction RSO2

→ Create the DataSource
“ZSNAPEXTRACT”

DataSource Goto System Help

Maintain Generic DataSources

DataSource

☒ Transaction data ZSNAPEXTRACT

☐ Master Data Attributes

☐ Texts

Create Change Display

34. Create a DataSource
The active ODS table from the
SnapShot ODS “ODS SnapShot” is
/BIC/AODSSNAP00

It is generated like “/BIC/A” + <ODS
Name> + “00”

Click Save

DataSource System Help

Create DataSource für Transactn data: ZSNAPEXTRACT

Generic Delta

DataSource ZSNAPEXTRACT Extraction from View

Applic. Component BW Extraction from Query

Obj. status New Extraction by FM

Texts

Short Description SnapShot Extract

Medium description SnapShot Extract from ODS

Long description SnapShot Extract from ODS

Extraction from DB View

View/Table /BIC/AODSSNAP00

ExtractStruct.

Extraction from SAP Query

InfoSet

Extraction by Function Module

Function Module

Extract.Struct.

35. The following fields are open for selection:

0CALMONTH to select the actual values and not the values from the future

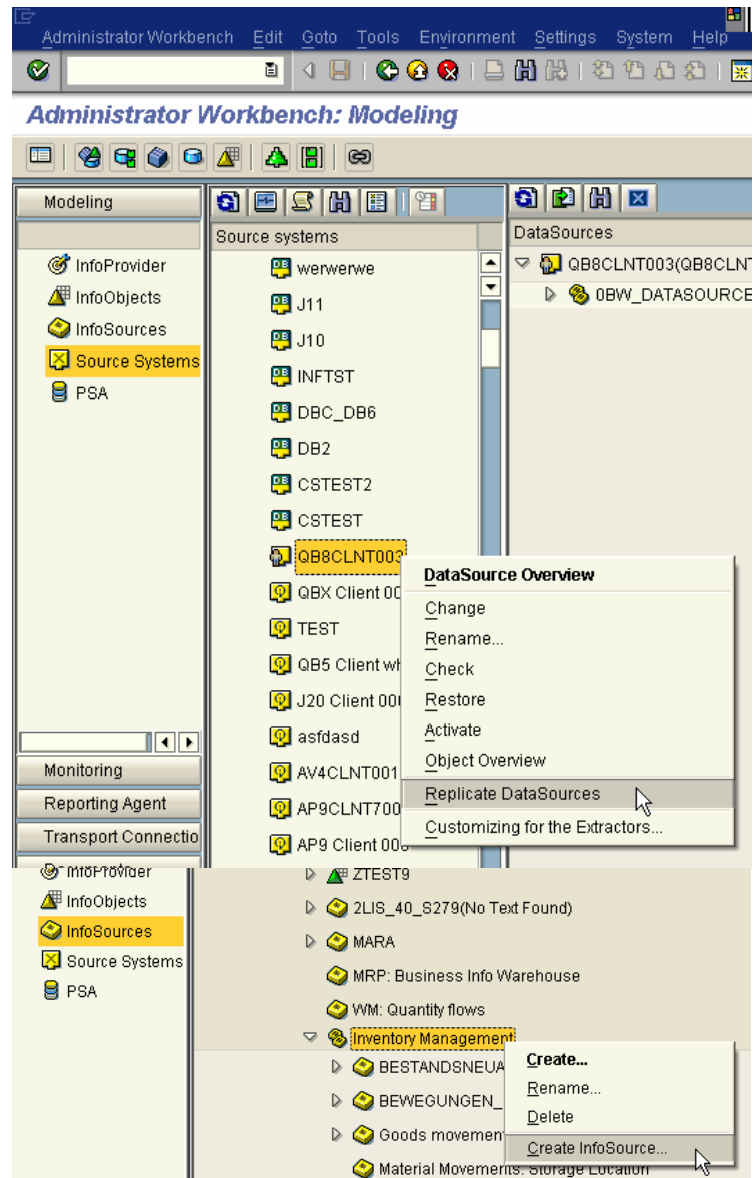
CALMONTH
INDSPECSTK
MATERIAL
PLANT
STOR_LOC
BATCH
STOCKTYPE
STOCKCAT

These fields are selected just in case, If a parallelism with different InfoPackages is needed..

Save the customer DataSource

Field Name	Short text	Selection	Hide field	Inve
CALMONTH	Calendar Year/Month	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
DOC_DATE	Document Date	<input type="checkbox"/>	<input type="checkbox"/>	
INDSPECSTK	Indicator: Valuation of Special Stock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MATERIAL	Material	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
PLANT	Plant	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STOR_LOC	Storage location	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
BATCH	Batch number	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STOCKTYPE	Stock type	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STOCKCAT	Stock categories in the Logistics Inform...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
RECORDMODE	BW Delta Process: Update Mode	<input type="checkbox"/>	<input type="checkbox"/>	
/BIC/ZTOT...	Quantity Total Stock	<input type="checkbox"/>	<input type="checkbox"/>	
/BIC/ZVAL...	Value of valuated stock	<input type="checkbox"/>	<input type="checkbox"/>	
/BIC/ZVAL...	Quantity of valuated stock	<input type="checkbox"/>	<input type="checkbox"/>	
BASE_UOM	Base Unit of Measure	<input type="checkbox"/>	<input type="checkbox"/>	
LOC_CURRCY	Local currency	<input type="checkbox"/>	<input type="checkbox"/>	

36. Replicate and save the DataSources from the Myself Source System

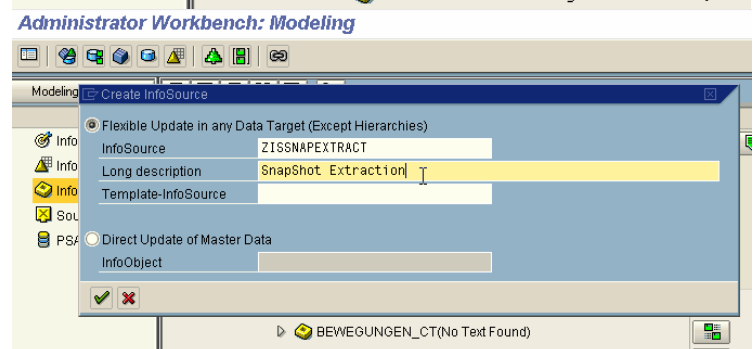


37. Create the InfoSource based on the DataSource "ZSNAPEXTRACT".

Right mouse-click on the application component in the InfoSource view.

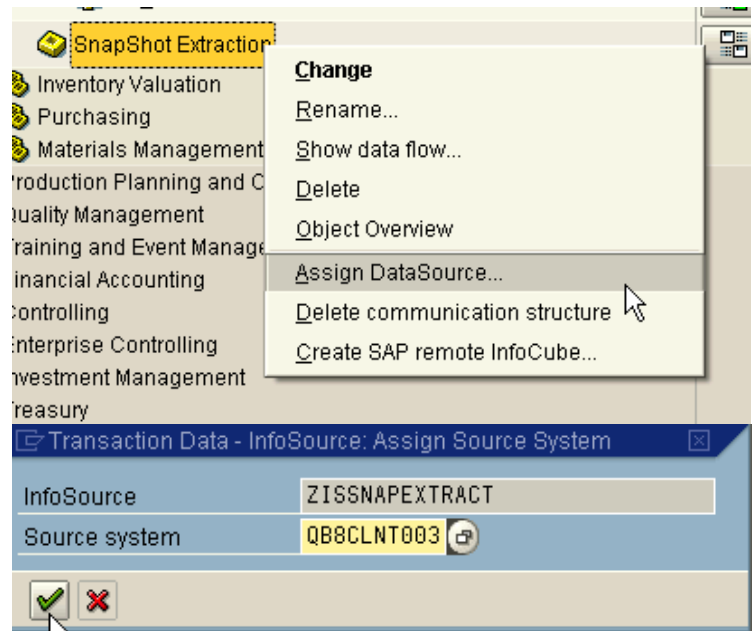
→ Create InfoSource

38. Type in the name of the InfoSource and the short text and press Return



39. Assign a DataSource to the InfoSource

→ Right mouse-click on the InfoSource and click “Assign DataSource”



40. The SourceSystem is the MySelf source system. In this case it is “QB8CLNT003”

Press “Return”

41. Select the DataSource “ZSNAPEXTRACT”

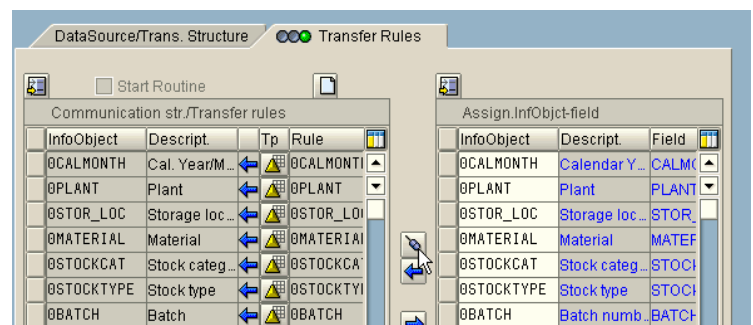
42. Assign the DataSource to the InfoSource

Click “Yes”

43. Click “Propose transfer rules”

The light should be green!

Activate the transfer rules



3.3.3.10 Create update rules to update the SnapShot InfoCube from the ODS – Process 4

44. Create the update rule from ODS ‘ODSSNAP’ to InfoCube “ICSNAP1”

Right click on InfoCube “ICSNAP1”
→ Create update rules

45. The InfoSource is “ZSNAPEXTRACT”

Change the update rule of the Time Characteristic Calendar Year / Month “0CALMONTH” and move the current month and year to this characteristic.

RESULT = sy-datum(6) .

Change the update rule of the Time Characteristic Calendar Year "OCALYEAR" and move the current year to this characteristic.

RESULT = sy-datum(4) .

These dates are corresponding to the stock of the actual period.

Activate the update rules

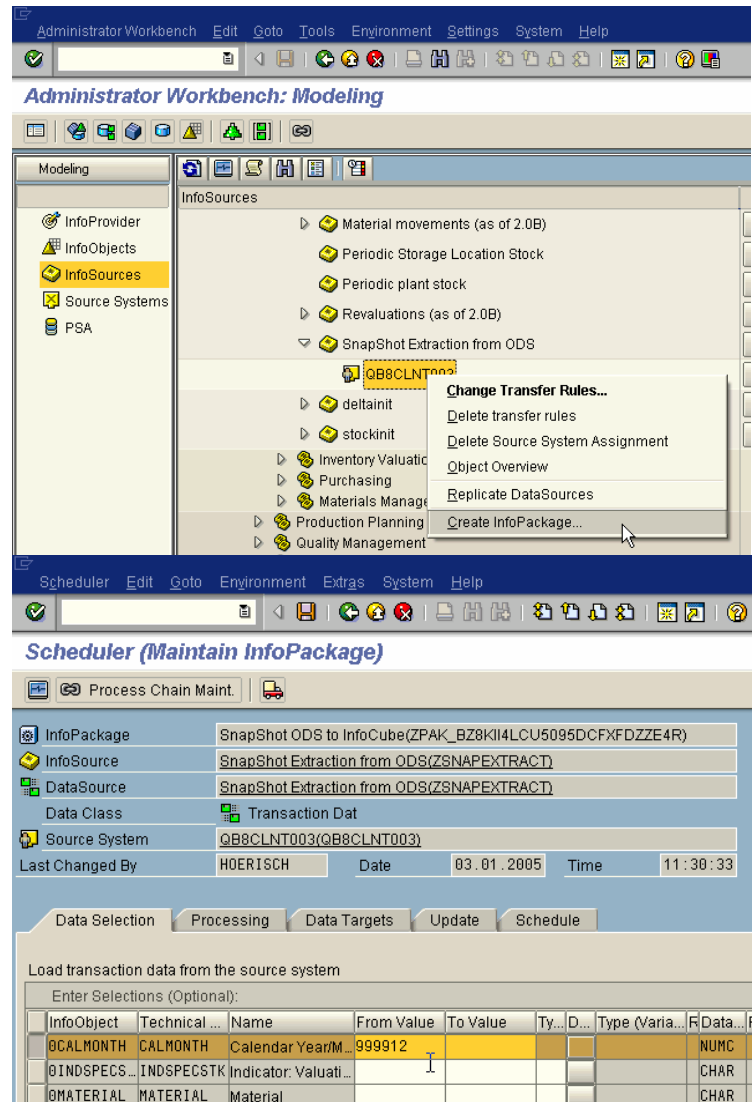
46. Create an InfoPackage to upload the SnapShot data from the ODS into the InfoCube

Goto the InfoSource
"ZSNAPEXTRACT"

Right mouse-click on the source system of the InfoSource and select
"Create InfoPackage"

Name of the InfoPackage:
SnapShot upload ODS to InfoCube

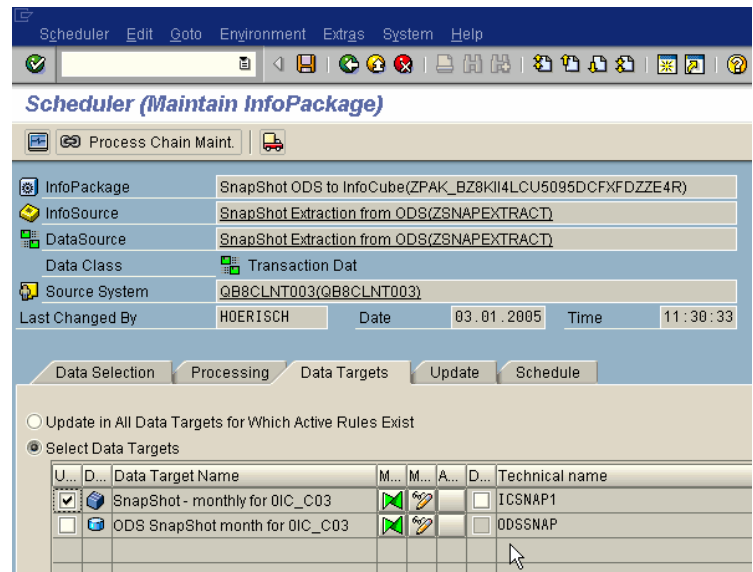
47. Fix the data selection to "999912", because this is the actual stock and the stock in the future is not needed.



48. Select the InfoCube “ICSnap1” as the data target.

and

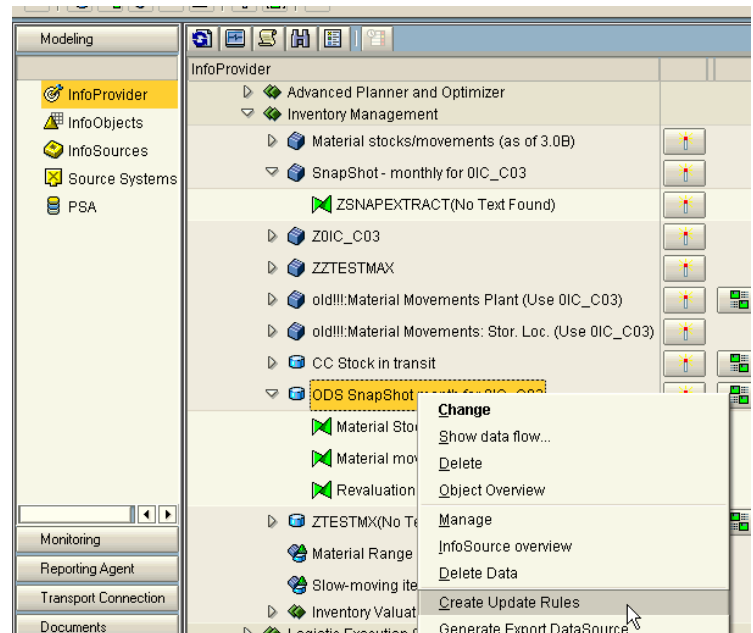
Save the InfoPackage



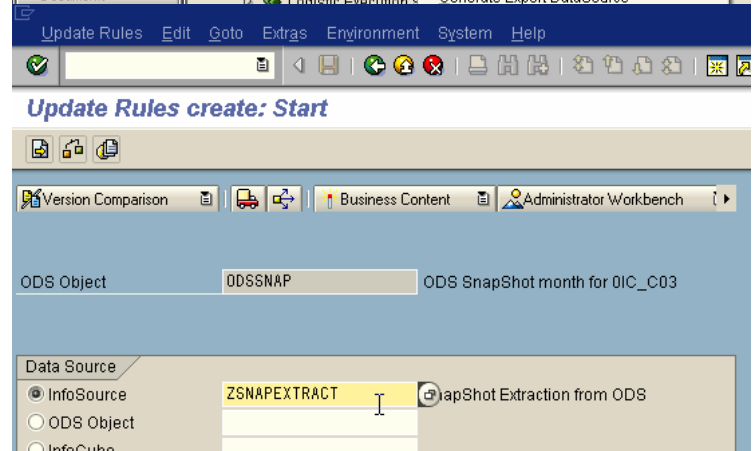
3.3.3.11 Create update rules to update the movements in the future – Process 5

49. Go to InfoProvider view and click right on the ODS Object “ODSSNAP”

Create update rule



50. Select the InfoSource “ZSNAPEXTRACT” and go to the next screen



51. Create a start routine to
 multiply the old row with minus
 1, set the record mode to R
 "Reverse-Image" and create a new
 row with the month "12.9999", if the
52. Start routine

```

PROGRAM UPDATE_ROUTINE.
*$$$ begin of global - insert your declaration only
below this line  *-*
* TABLES: ...
TYPES:
  BEGIN OF DATA_PACKAGE_STRUCTURE2.
    INCLUDE STRUCTURE /BIC/CSZSNAPEXTRACT.
  TYPES:
    RECNO LIKE sy-tabix,
  END OF DATA_PACKAGE_STRUCTURE2.

DATA:
  DATA_PACKAGE2 TYPE STANDARD TABLE OF
DATA_PACKAGE_STRUCTURE2
  WITH HEADER LINE
  WITH NON-UNIQUE DEFAULT KEY INITIAL SIZE 0.
*$$$ end of global - insert your declaration only
before this line  *-*

* The follow definition is new in the BW3.x
TYPES:
  BEGIN OF DATA_PACKAGE_STRUCTURE.
    INCLUDE STRUCTURE /BIC/CSZISSNAPEXTRACT.
  TYPES:
    RECNO LIKE sy-tabix,
  END OF DATA_PACKAGE_STRUCTURE.
DATA:
  DATA_PACKAGE TYPE STANDARD TABLE OF
DATA_PACKAGE_STRUCTURE
  WITH HEADER LINE
  WITH NON-UNIQUE DEFAULT KEY INITIAL SIZE 0.

FORM startup
  TABLES  MONITOR STRUCTURE RSMONITOR "user
defined monitoring
           MONITOR_RECNO STRUCTURE RSMONITORS "
monitoring with record n
           DATA_PACKAGE STRUCTURE DATA_PACKAGE
           USING          RECORD_ALL LIKE SY-TABIX
           SOURCE_SYSTEM LIKE RSUPDSIMULH-LOGSYS
           CHANGING ABORT LIKE SY-SUBRC. "set ABORT <> 0 to
cancel update
*
*$$$ begin of routine - insert your code only below
this line  *-*
* fill the internal tables "MONITOR" and/or
"MONITOR_RECNO",
* to make monitor entries
  loop at DATA_PACKAGE.
    DATA_PACKAGE-/BIC/ZTOTALST = DATA_PACKAGE-
/BIC/ZTOTALST * -1.
    DATA_PACKAGE-/BIC/ZVALSTCKQ = DATA_PACKAGE-
/BIC/ZVALSTCKQ * -1.
    DATA_PACKAGE-/BIC/ZVALSTCKV = DATA_PACKAGE-
/BIC/ZVALSTCKV * -1.
    move 'R' to DATA_PACKAGE-RECORDMODE.
    move DATA_PACKAGE to DATA_PACKAGE2.
    append DATA_PACKAGE2.
    DATA_PACKAGE-/BIC/ZTOTALST = DATA_PACKAGE-

```

```

/BIC/ZTOTALST * -1.
DATA_PACKAGE-/BIC/ZVALSTCKQ = DATA_PACKAGE-
/BIC/ZVALSTCKQ * -1.
DATA_PACKAGE-/BIC/ZVALSTCKV = DATA_PACKAGE-
/BIC/ZVALSTCKV * -1.
move '999912' to DATA_PACKAGE-CALMONTH.
move ' ' to DATA_PACKAGE-RECORDMODE.
modify DATA_PACKAGE.
endloop.
append lines of DATA_PACKAGE2 to DATA_PACKAGE.
* if abort is not equal zero, the update process
will be canceled
ABORT = 0.
*$$$ end of routine - insert your code only before
this line *-*
*
ENDFORM.

```

53. Set the key figure update to
Addition

and

Activate the update rules

54. Create an InfoPackage to update
the future movement to the fix month
"12.9999" (remove the flag "Use
conversion routine)

Go to the InfoSource
"ZSNAPEXTRACT"

Right mouse-click on the source
system of the InfoSource and select
"Create InfoPackage"

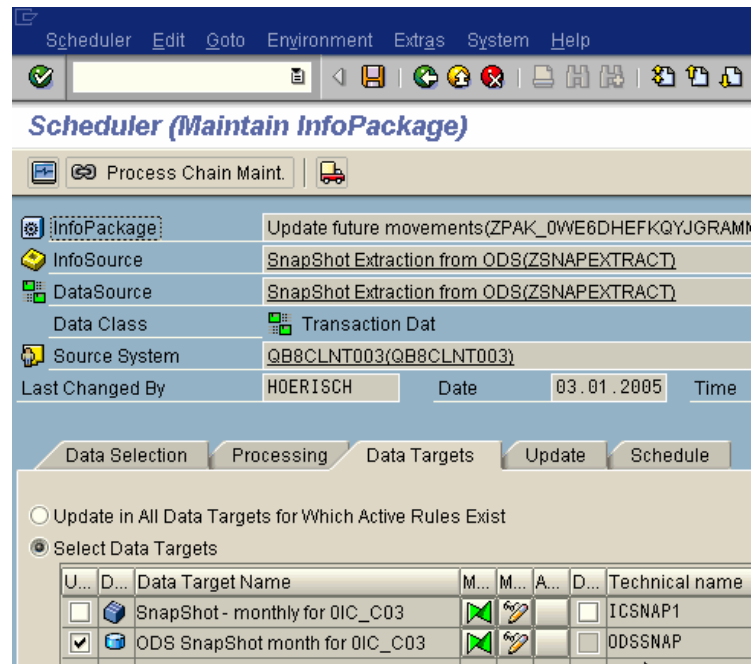
The name of the InfoPackage is
"Update future movements to the
current SnapShot"

55. Restrict the month where the future movements are stored
e.g. "02.2004"

All rows in the ODS

The data target is the ODS
"ODSSNAP"

Save the InfoPackage



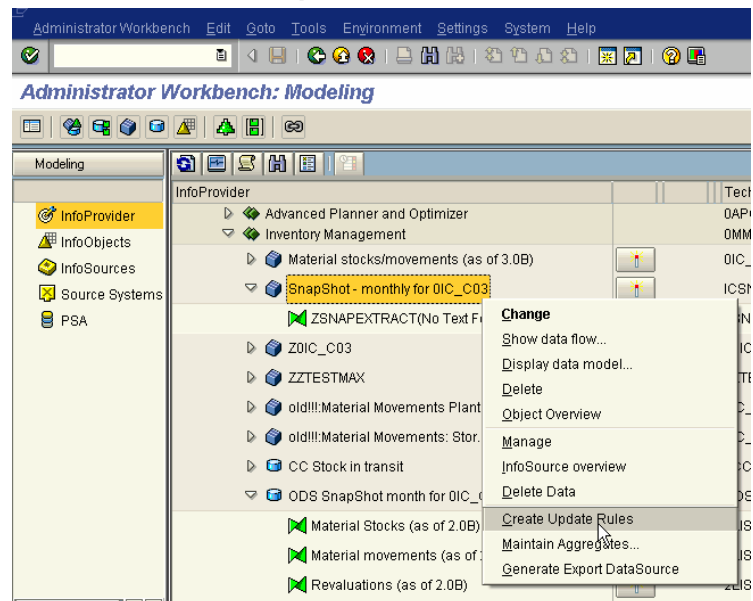
3.3.3.12 Create an update rule to upload the movement in the past – Process 3

56. Go to InfoProvider view and click right on the InfoCube object
"ICSNAP1"

Create update rule for the

InfoSources:

2LIS_03_BF
and
2LIS_03_UM



57. Create the **start routine** and double the data to the corresponding months

e.g. If the SnapShot for month 01.2004 is in the InfoCube and a movement is uploaded for month 11.2003 the movements needs to be doubled for month 12.2003 and 01.2004.

Create a start routine for each of the InfoSources

2LIS_03_BF

```
PROGRAM UPDATE_ROUTINE.
*$$$ begin of global - insert your declaration
only below this line *-*
* TABLES: ...
TYPES:
  BEGIN OF DATA_PACKAGE_STRUCTURE2.
    INCLUDE STRUCTURE /BIC/CS2LIS_03_BF.
  TYPES:
    RECNO LIKE sy-tabix,
  END OF DATA_PACKAGE_STRUCTURE2.

DATA:
  DATA_PACKAGE2 TYPE STANDARD TABLE OF
  DATA_PACKAGE_STRUCTURE2
  WITH HEADER LINE
  WITH NON-UNIQUE DEFAULT KEY INITIAL SIZE
  0.
DATA: post_month type /BI0/OICALMONTH.
```

or
2LIS_03_UM

Note: Please change the TYPES declaration to the corresponding InfoSource.

For InfoSource 2LIS_03_UM the include structure needs to be changed to be changed to

/BIC/CS2LIS_03_UM

The actual period will NOT be updated in the InfoCube. The SnapShot of the actual period will be loaded into the InfoCube at the last day of the period.

```
DATA: actual_month type /BI0/OICALMONTH.

*$$$ end of global - insert your declaration
only before this line   *-*

* The follow definition is new in the BW3.x
TYPES:
  BEGIN OF DATA_PACKAGE_STRUCTURE.
    INCLUDE STRUCTURE /BIC/CS2LIS_03_BF.
  TYPES:
    RECNO    LIKE sy-tabix,
  END OF DATA_PACKAGE_STRUCTURE.
DATA:
  DATA_PACKAGE TYPE STANDARD TABLE OF
  DATA_PACKAGE_STRUCTURE
    WITH HEADER LINE
    WITH NON-UNIQUE DEFAULT KEY INITIAL SIZE
  0.

FORM startup
  TABLES   MONITOR STRUCTURE RSMONITOR "user
defined monitoring
            MONITOR_RECNO STRUCTURE RSMONITORS "
monitoring with record n
            DATA_PACKAGE STRUCTURE DATA_PACKAGE
            USING          RECORD_ALL LIKE SY-TABIX
            SOURCE_SYSTEM LIKE RSUPDSIMULH-
LOGSYS
  CHANGING ABORT LIKE SY-SUBRC. "set ABORT <> 0
to cancel update
*
*$$$ begin of routine - insert your code only
below this line   *-*
* fill the internal tables "MONITOR" and/or
"MONITOR_RECNO",
* to make monitor entries
  data: diff(2) type i,
        diff_month(2) type i,
        diff_year(4) type i.
  data: dummy_year(4) type n.
* get actual period
  if actual_month is initial.
    move sy-datum(6) to actual_month.
  endif.

* get movements in the PAST!
  loop at DATA_PACKAGE.
* get month of the movement
    move DATA_PACKAGE-PSTNG_DATE(6) to
post_month.
* check if movement in the past
  if post_month < actual_month.
* If the movement is in the past, get the
difference of the periods
    diff_year = actual_month(4) -
post_month(4).
    diff_month = actual_month+4(2) -
post_month+4(2).
    diff = ( diff_year * 12 ) + diff_month.
* Check If the difference of the periods >
0.
    if diff > 0.

        post_month = actual_month.

* double the row for all past periods
do diff times.
```

```

        move DATA_PACKAGE to data_package2.

        post_month = post_month - 1.

* If the month lower than 1 set the month to 12
and the year - 1
        if post_month+4(2) LT 1.
            post_month+4(2) = '12'.
            move post_month(4) to dummy_year.
            dummy_year = dummy_year - 1.
            post_month(4) = dummy_year.
        endif.

        move post_month to data_package2-
PSTNG_DATE(6).
        move '01' to data_package2-
PSTNG_DATE+6(2).
*      add the movement in the past / row
per row
        append data_package2.
        enddo.
    endif.
endif.
endloop.
* delete all rows of the DataPackage
refresh DATA_PACKAGE.
clear DATA_PACKAGE.
* add the movements in the past
append lines of DATA_PACKAGE2 to
DATA_PACKAGE.
* delete all rows of the DataPackage2
refresh DATA_PACKAGE2.
clear DATA_PACKAGE2.

* if abort is not equal zero, the update
process will be canceled
ABORT = 0.

*$$$ end of routine - insert your code only
before this line      *_*
*
ENDFORM.

```

58. Copy the update rules for key figures from the update rules to the SnapShot ODS

see

3.3.3.7 Create update rules from 2LIS_03_BF to ODS ODSSNAP → Process 2

3.3.3.8 Create update rules from 2LIS_03_UM to ODS ODSSNAP → Process 2

The time characteristics 0CALMONTH and 0CALYEAR are derived from 0PSTNG_DATE Posting date.

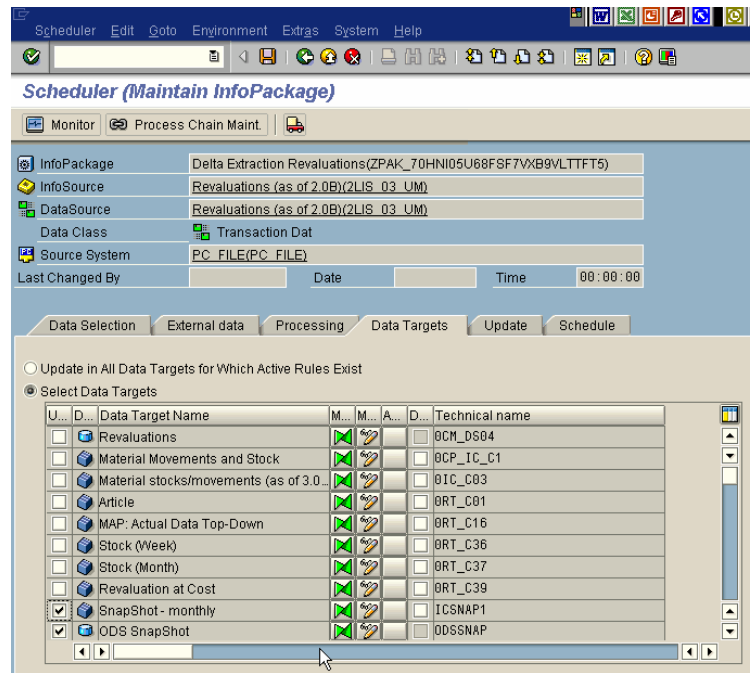
The unit of the routine is 0BASE_UOM Base Unit of Measure or 0LOC_CURRCY Local currency.

The Source Characteristics for the Time Ref. 0CALMONTH and 0CALYEAR is 0PSTNG_DATE Posting date in the document.

59. Change data targets in InfoPackage for both DataSources

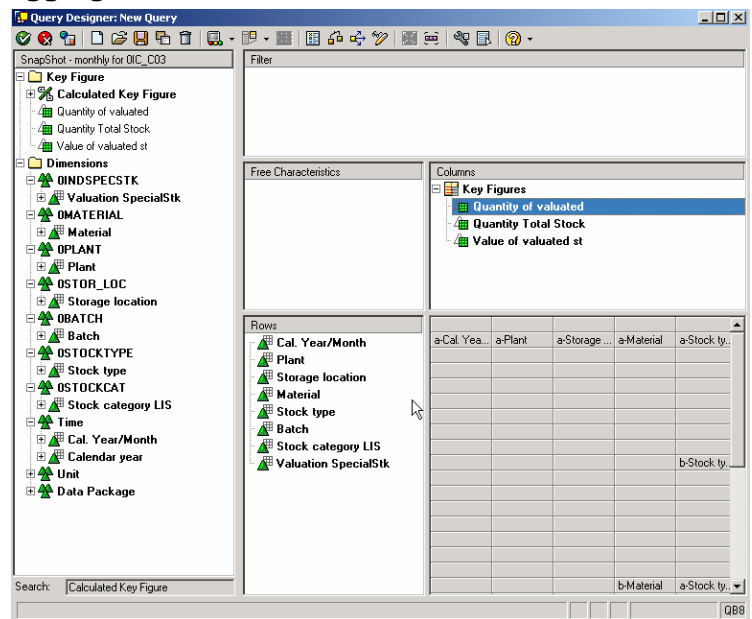
2LIS_03_BF
and
2LIS_03_UM

Data targets are:
the SnapShot ODS “ODSSNAP” and
the SnapShot InfoCube “ICSNAP1”.



3.3.3.13 Create a Query with exceptional aggregation

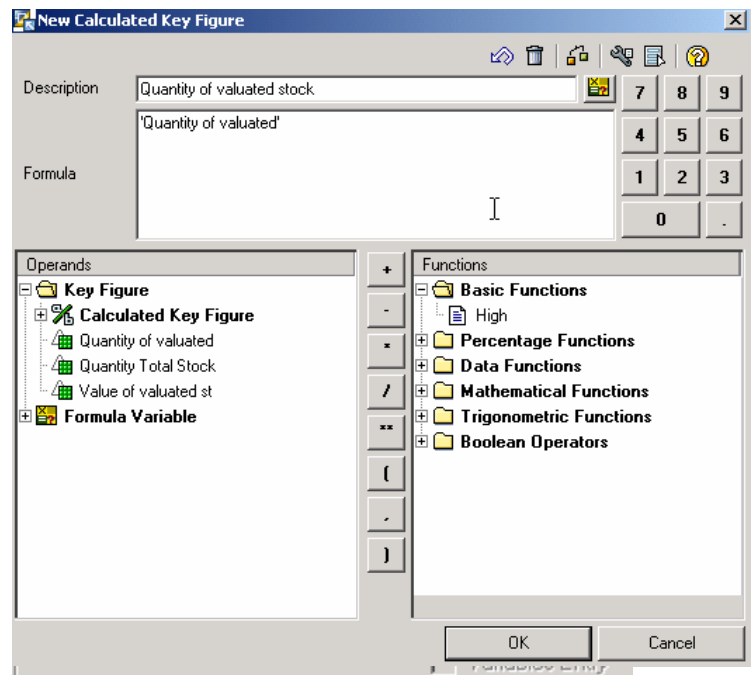
60. Create Query on InfoCube “ICSNAP1”.



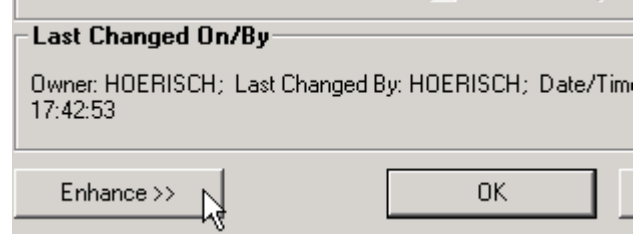
61. Create calculated keyfigures with exceptional aggregation last on 0CALMONTH

Right mouse click on Key Figure

62. Press OK

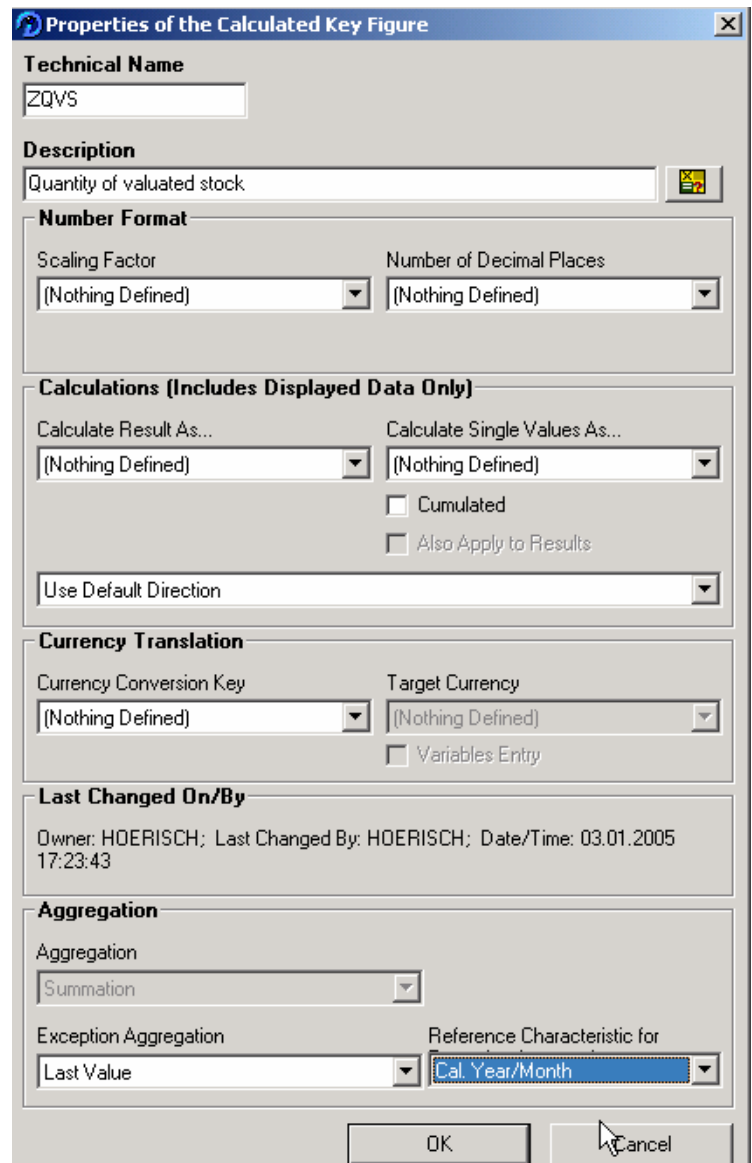


63. Press Enhance



64. Exception Aggregation → Last Value
Reference Characteristic → OCALMONTH

technical name of the key figure → SNAPSTOCK



Properties of the Calculated Key Figure

Technical Name
ZQVS

Description
Quantity of valuated stock

Number Format
Scaling Factor: (Nothing Defined) Number of Decimal Places: (Nothing Defined)

Calculations (Includes Displayed Data Only)
Calculate Result As...: (Nothing Defined) Calculate Single Values As...: (Nothing Defined)
☐ Cumulated
☐ Also Apply to Results
Use Default Direction

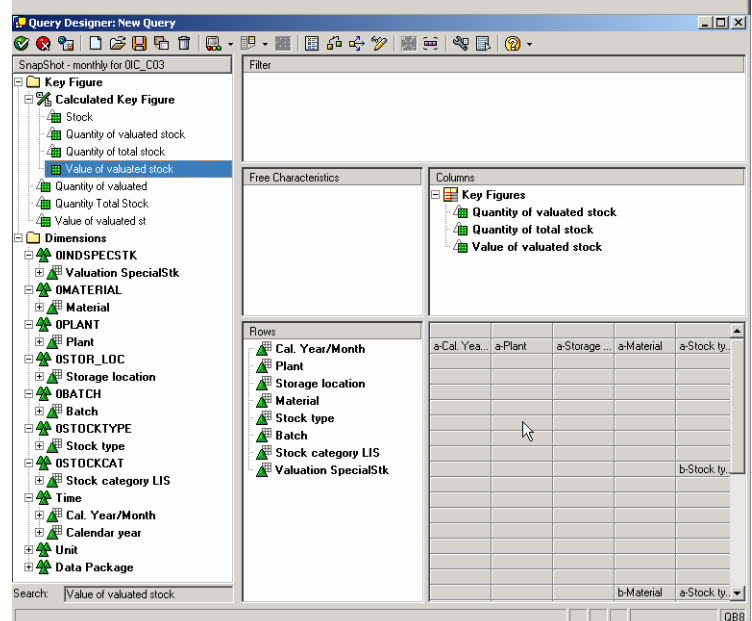
Currency Translation
Currency Conversion Key: (Nothing Defined) Target Currency: (Nothing Defined)
☐ Variables Entry

Last Changed On/By
Owner: HOERISCH; Last Changed By: HOERISCH; Date/Time: 03.01.2005 17:23:43

Aggregation
Aggregation: Summation
Exception Aggregation: Last Value Reference Characteristic for: Cal. Year/Month

OK Cancel

65. Define and save the Query



Query Designer: New Query

Snapshot - monthly for OIC_C03

Key Figure
Calculated Key Figure
Quantity of valuated stock
Quantity of total stock
Value of valuated stock

Dimensions
OINDSPECSTK
Valuation SpecialStk
OMATERIAL
Material
OPLANT
Plant
OSTOR_LOD
Storage location
OBATCH
Batch
OSTOCKTYPE
Stock type
OSTOCKCAT
Stock category LIS
Time
Cal. Year/Month
Calendar year
Unit
Data Package

Filter

Free Characteristics

Columns
Key Figures
Quantity of valuated stock
Quantity of total stock
Value of valuated stock

Rows
Cal. Year/Month
Plant
Storage location
Material
Stock type
Batch
Stock category LIS
Valuation SpecialStk

Search: Value of valuated stock

QB8

66. Result

Microsoft Excel

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Business Explorer ?

Frage hier eingeben

100%

E17 30000

Mappe3

	A	B	C	D	E
1	SnapShot Query 2				
2					
3	Batch				
4	Valuation SpecialStk				
5	Stock type				
6	Stock category LIS				
7	Cal. Year/Month	01.2005			
8	Key Figures				
9	Material				
10	Plant				
11	Storage location				
12					
13	Cal. year / month	11.2004, 12.2004, 01.2005, 02.2005			
14	Plant	Werk 0001			
15					
16	Material	Plant	Quantity of valued stock	Quantity of total stock	Value of valued stock
17	A4-P01-01	0001	1.000 EA	1.000 EA	30.000,00 EUR
18	AB-P-100-110	0001	100 PC	100 PC	10.000,00 EUR
19	AB-P-100-120	0001	100 PC	100 PC	500,00 EUR
20	AB-P-100-200	0001	100 PC	100 PC	5.000,00 EUR
21	AB01-P-100-110	0001	1.000 PC	1.000 PC	100.000,00 EUR
22	AB01-P-100-120	0001	1.000 PC	1.000 PC	5.000,00 EUR
23	AB01-P-100-200	0001	1.000 PC	1.000 PC	50.000,00 EUR
24	AL-005-BLP	0001	2.000 PC	2.000 PC	40.000,00 EUR
25	AL-005-VLP	0001	4.500 PC	4.500 PC	0,00 EUR
26	AP5-VMI-1	0001	11.003 PC	11.003 PC	165.045,00 EUR
27	AP9-VMI-1	0001	562 PC	562 PC	8.430,00 EUR
28			3.000 EA	3.000 EA	

Tabellen1 Tabellen2 Tabellen3

3.4 Example for Implementation Scenario

The following picture (Illustration 2 Overview Stock Reporting) shows the necessary layers and process in BW for the reporting of the stocks and the corresponding values. This picture shows in addition the coexistence of the SnapShot and the non-cumulative handling of the stocks.

During the business blueprint phase in a project the necessity of the different layers should be checked. The flexibility, reliability, redundancy and the business needs should be taken in account.

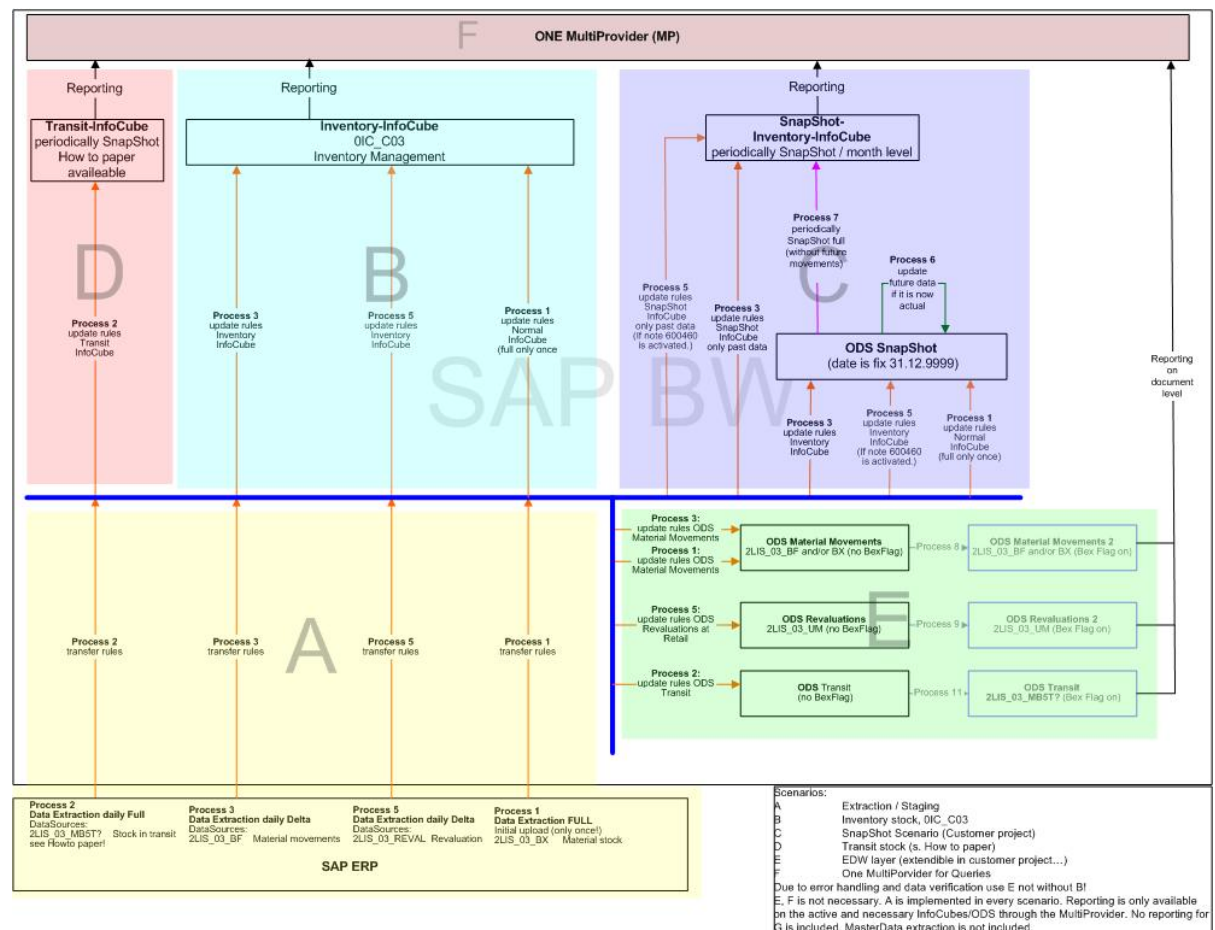


Illustration 2 Overview Stock Reporting

The component A is the staging part. E is the data warehouse layer. D, B and C stand for the reporting layer. Through this MultiProvider F all reportable Objects are accessible.

The MasterData extraction is not included. Especial in the Retail area due to the size of the master data objects the modeling of navigational Attributes to the right corresponding InfoObject is important. In some cases it makes sense to split the needed attributes from the loading and the reporting perspective.

The main decision should be dependent to the time that is needed to get the data from extraction to reporting. The runtime of change run of the aggregates is usually very important.

The part A includes the DataSources and transfer rules and should be maintained corresponding which data is needed in the different layers.

www.sdn.sap.com/irj/sdn/howtoguides