Business Process Monitoring of Data Consistency between SAP BI and Source Systems

Best Practice for Solution Management

Version Date: November 2008

1 Applicability, Goals and Requirements
2 Introduction to Data Consistency Monitoring in General
3 Overview for the Example Scenario
4 Introduction to the Example Scenario Logic
5 Process Model
6 Implementation in the SAP BI System
   6.1 Step by Step Description
   6.2 Activities in Source System SAP ERP
      6.2.1 Create a View for Extraction with a Generic DataSource
      6.2.2 Create a Generic DataSource
   6.3 Activities in SAP NetWeaver BI
      6.3.1 Create a VirtualProvider
      6.3.2 MultiProvider
      6.3.3 DataStore-Object for storing the result of the analysis process
      6.3.4 Comparison of the key figures
      6.3.5 Analysis Process using a query
      6.3.6 Analysis Process without Query
      6.3.7 Result of the Analysis Process
7 Scheduling of the Analysis Process
8 Automatized Data Consistency Monitoring
   8.1 Setup steps in SAP Solution Manager
   8.2 Monitoring in SAP Solution Manager (Overview)
   8.3 Monitoring Session in SAP Solution Manager
1 Applicability, Goals and Requirements

Tools to check data consistency between the SAP Business Intelligence system (BI) and the source system are often requested. Since data extraction and updating usually includes business logic, a general tool for data consistency checks is not possible. Such a check has to consider the specific needs of the respective business process. Thus, it has to be implemented on an individual project basis.

Data consistency between the transactional system and the reporting system is especially important when the SAP BI reporting results are used for daily operational decisions. Therefore, SAP now provides a solution scenario where data consistency is monitored in SAP Solution Manager Business Process Monitoring (BPMon).

This Best Practice document describes in detail how you can implement a data consistency check between SAP NetWeaver BI and SAP ERP, that is, how to set up the data flows to provide the consistency information to SAP Solution Manager. The scenario described here is based on the SAP ERP SD application and the SAP NetWeaver BI SD Business Content. You can use this implementation example as a guideline for your own implementations in SAP ERP and SAP NetWeaver BI. This concept can be transferred to other ERP applications as well as to other source systems, for example, SAP CRM. Nevertheless, the built-in business-related logic then has to be adapted and matched to your specific needs.
2 Introduction to Data Consistency Monitoring in General

The term “inconsistencies between transactional and reporting system” has different implications. Inconsistency in general means that the data in the InfoProviders in BI does not reflect the original values in the source system.

Unexpected reporting results in BI are not necessarily indicators of inconsistencies; this depends for example on the query design and technical reporting availability of data.

An automated check that verifies the data consistency between ERP and BI-system is interesting because, during the extraction, loading, and transformation of the data, technical and logical failures can occur. If you do not receive error messages in your data loading process, it does not necessarily mean that your data is logically correct. Such errors can appear through the technically correct, but logically incorrect transformation of data and can lead to inconsistent data sets. Also, undetected technical errors can lead to this mismatch between the transactional system and the reporting system. An inconsistent status between these two databases can lead to an incorrect interpretation of reporting results from BI and thus to wrong business decisions. As more and more BI customers use the BI in their daily operations, it is important to have a possibility to monitor the data consistency between these two SAP systems for the most critical business processes.

SAP Solution Manager, as the central tool for cross-system administration, contains functionality for business process monitoring. In this document, we provide you with an example how a scenario for data validation may be set up in BI and connected to the SAP Solution Manager. This scenario uses only the NetWeaver BI standard technology and requires no special programming effort.

However, the implementation of such a scenario strongly depends on your data loading process, that is, the application where data is loaded from, the capabilities of the extraction program, the delta update procedure, the data model in the BI system, and the used upload logic.

Through the transformation of data during the ETL (extract, transform, load) process, you need to be well informed about the whole process that starts at the application table in your SAP ERP system and ends in your reporting InfoProvider in SAP NetWeaver BI. The data model in BI is usually built of several layers with data transformations in between, so it has to be determined carefully, which DataProvider should be compared to the source system. Every transformation, aggregation, or modification of data in the loading process makes the implementation of this scenario more complicated. So be aware that the more complex your business processes are, the more effort the implementation will need.
From our experience, there are two main types of inconsistencies:

**Inconsistencies due to Missing or Duplicated Data Records in SAP BI**

Some possible reasons:
- Data is processed incorrectly in the S-API user exit during upload
- Data is processed incorrectly during updating in BI (update rules, start-/ end-routines, transformations, mapping)
- Technical issues during the loading process, for example, in RFC.

**Inconsistencies due to Wrong or Unexpected Values in Single Keyfigures in SAP BI**

Some possible reasons:
- The extraction program works as designed, but this does not meet your expectations.
- The extraction program provides wrong values to SAP BI.
  - Extractor issue
  - Application customizing issue
- Data is processed incorrectly in the S-API user exit during upload.
- Data is processed incorrectly during updating in BI (update rules, start-/ end-routines, transformations, mapping).
- Technical issues during the loading process, for example, in RFC.

In this document, we refer to such inter-system inconsistencies. Another type of inconsistency can occur within the BI system itself, that is, data in reporting InfoProviders is different from the information that is available in PSA. If it is essential for you to do a consistency check between different layers in BI, the approach may be similar to the one described here for inter-system checks.

The data flow for consistency checks should be self-defined to avoid using the same loading mechanisms as in standard uploads. If standard load and consistency checks use the same technique, you might fail to detect inconsistencies as then the same errors may happen in both loading processes. Therefore, you have to implement the business logic of the standard loading process into the validation data flow.

An automated inconsistency monitoring based on the concept described here is only able to detect that there is an inconsistency. The analysis of the actual issue and the reason for the inconsistency cannot be automated with a reasonable effort and performance. This investigation is manual work, where specific data records have to be tracked from source to target.

If you are using Real Time Data Acquisition (RDA), you should not compare the content of the DataStore Object used for RDA directly with the source system. RDA data may be loaded with very high frequency (every one or two minutes), and comparisons to the source system may take longer than this timeframe. For this kind of data, you should compare the condensed data in a later staging step, that is, in an InfoProvider where data is collected for mid term reporting. Even if it would be technically possible to include data providers that are loaded several times per day in the data consistency monitoring by choosing a very selective comparison criterion (like the UTC-timestamp), it does not seem useful for performance reasons.
This document describes in detail a possible implementation of a validation scenario for SD data. It should indicate the possibilities that you have and things that you should consider. There is another document available in the SAP Developer Network that describes the concepts of a reconciliation scenario more in general (How To... Reconcile Data Between SAP Source Systems and SAP NetWeaver BI):


The concept for data consistency monitoring implies that new objects have to be created, in SAP BI as well as in the source system. As for all implementations in SAP systems, also here the implementation has to be tested in a production-like test system. All created objects have to be assigned to the right development class and transport path.

3 Overview for the Example Scenario

For our example scenario, we chose a BI Business content InfoCube which is filled via a Business Content DataSource. The content of the InfoCube is compared to the content of a suitable application table in the source system. Direct access to the source system is done by using a VirtualProvider. The VirtualProvider and the Business Content InfoCube are joined via a MultiProvider. This MultiProvider contains a chosen keyfigure from both the basis InfoProviders. A potential difference is calculated, and the result is written into a DataStore Object. The content of this DSO is read by a collector program from SAP Solution Manager. In case a difference has been detected, you will be notified in Business Process Monitoring in SAP Solution Manager.
4 Introduction to the Example Scenario Logic

For our example scenario, we chose the BI Business content InfoCube 0SD_C05 (Quotations / Orders, see description in online help) which contains the sales order document data on header level, extracted via the Business Content DataSource 2LIS_11_VAHDR (Sales Order-Header Data (from 20B), see description in online help).

To find a keyfigure that could be used for comparison we analyzed the data available in the sales orders in the SAP ERP system and decided to use the field “Net value” in the header data of the sales orders. This value is saved in SD Application table for the header data (VBAK). The corresponding DataSource 2LIS_11_VAHDR is extracting the header data. This DataSource provides data for the BI content cube 0SD_C05 which is used in this scenario. To use the DataSource and the InfoCube for reconciliation of the net value, a few changes were necessary.

We decided to use the value for field NETWR (net value of a sales order) for our reconciliation. It was necessary to extend the BI business content with the field AEDAT (date on which the document was changed) to also get the changes of the data records. With just the existing field ERDAT, the DataSource would extract the changed documents, but it would not be possible to track them in our validation scenario and thus data consistency could not be ensured for changed documents.

---

1. http://help.sap.com/saphelp_nw70/helpdata/EN/21/93e63736544d4de10000009b38f8cf/frameset.htm

© 2008 SAP AG
With a formula in the transformation rules for 0CALDAY, it was possible to map the date of last change (IF AEDAT >= ERDAT use AEDAT, otherwise use ERDAT) on this InfoObject. Be aware that this is a change in the general update logic. Such a change may also have to be considered in query definitions.

We also extended the Business Content DataSource 2LIS_11_VAHDR (in transaction LBWE in the source system) and the Business Content InfoCube 0SD_C05 (in transaction RSA1 in the BI system) with the information for NETWR which is used for the sum of the net values of the items in a sales order. The cumulated value of this key figure will be used for cross-checking the values in ERP and BI.
5 Process Model

The process model shown below gives an overview over the whole scenario in detail.

In parallel to the normal dataflow of the BI content (shown on the left-hand side, with InfoCube 0SD_C05), another dataflow with the same extraction logic is set up (shown on the right-hand side) which is used for validation. The core of the second loading process is the VirtualProvider which is in our scenario like a view on the application table VBAK. This InfoProvider is joined with the standard InfoCube through a MultiProvider. The reconciliation between the key figure values of the InfoCube and the VirtualProvider is done by an Analysis Process which compares these key figures in the MultiProvider and writes the result into a DataStore-Object. The data records in this DataStore-Object can be evaluated by a data collector of the SAP Solution Manager. Any difference in the values will lead to an alert in Business Process Monitoring (BPMon).
If you are going to implement such a scenario in your system, the main work will be the adaptation of the logic of the standard loading process into the parallel data stream for reconciliation, represented here by the VirtualProvider. In our scenario, the standard data flow is a direct load into an InfoCube. In most BI implementations, a layered data model is used. This means that one or more layers of DataStore Objects and one or more layers of InfoCubes are used in one dataflow. Then, the first decision to take is: data on which level should be compared to which “original” data? Secondly, the extraction and upload logic has to be analyzed in detail and has to be implemented in the reconciliation data stream. This may be pretty complex, so that extraction with using a function module may be necessary.

Also, we chose a keyfigure for comparison that is pretty easy to read from the source system. Depending on your choice which keyfigure(s) to use, the effort for implementing such a scenario may be much higher than shown in this document.

The complexity and individuality of business processes and the enormous flexibility in setting up a data model in BI leads to the situation that there cannot be a general tool for consistency checks, but it has to be implemented based on the individual needs.
6 Implementation in the SAP BI System

6.1 Step by Step Description

The following work stages are necessary to implement the process.

Source System:
- Create a view for the data extraction from the application table
- Create a generic DataSource based on that view

SAP NetWeaver BI:
- Build a VirtualProvider with direct access
- Customize Transformation Rules (routine for 0CALDAY)
- Create a MultiProvider
- Create a DataStore-Object for direct update
- Create an Analysis Process

These steps will be explained in detail in the following sections.
6.2 Activities in Source System SAP ERP

6.2.1 Create a View for Extraction with a Generic DataSource

The first step for the implementation is the creation of a database view which is required to create a generic data source in the source system.

Start transaction SE11. Select “View” and type in the name of the view and press “Create”.

In the next screen, select “Database view” and press “Copy”.

© 2008 SAP AG
Now you are in the maintenance screen of the view. Enter in the column “Tables” the name of the application table you want to extract from. You can also use more tables and join them if you need to. This may be necessary, for example, when you want to extract a key figure on item level but the currency key is on the header level.

On tab strip “View Flds”, you have to add the fields from the application table you entered in the screen before. Press the button “Table fields” to open the field selection.
Now select the fields you want to extract.
In this case the fields MANDT, VBELN, ERDAT, AEDAT, VBTYP, NETWR and WAERK are extracted from the table VBAK.

The screen should look like this now:
In the tab strip “Selection Conditions”, you have the option to create selections on a value from your selected fields. This selection could also be done later in the BI system or in the query but if you make your selection in the source system, you reduce the extracted and transferred data volume. This may be beneficial for the performance of this access to the source system, which may be a critical factor in the usability of this validation scenario.

For this example, we have chosen only the document category “C” (Sales Orders) for the consistency check.

If your extraction logic is more complex, you can also use a function module for the extraction. For more information about how to create a generic DataSource based on a function module, you can check a document in SAP Developer Network SDN: https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/a0f46157-e1c4-2910-27aa-e3f4a9c8df33.

This is not an official SAP document; please notice the disclaimer on the last page.
6.2.2 Create a Generic DataSource

The next step is the creation of a generic DataSource which is based on the view you created. Start transaction RSO2 to create this DataSource.

Choose "Transaction data", enter the name of the DataSource, and press "create".

On the next screen, you have to press the button “Extraction from View” and enter the name of your view in the "Extraction from DB View" area. Now press the save button ( ) to get to the next screen.
Finally, you have to check the boxes of the fields you want to select on and press the save button (✓) again.

The work in the source system is done. This generic DataSource is ready for extraction now. If you want to test your extractor, you can do this with the “Extractor Checker” in transaction RSA3.

Log in into your BI-System. Replicate and activate this DataSource in the transaction RSA1.

© 2008 SAP AG
6.3 Activities in SAP NetWeaver BI

6.3.1 Create a VirtualProvider

After you have replicated your DataSource in your BI-System, you can create the VirtualProvider which directly accesses this DataSource. In RSA1 > Modeling > InfoProvider select the InfoArea where your VirtualProvider should be created and right-click for the context menu.

Choose as InfoProvider of type “VirtualProvider” based on Data Transfer Process for Direct Access.”
The InfoCube should contain the key figures and characteristics that are extracted from your generic DataSource.

In this example, the key figure net value (DC_NETVAL) of the sales document category “C” (0DOC_CATEG) in a specific timeframe (0CALDAY) is reconciled.

<table>
<thead>
<tr>
<th>InfoCube</th>
<th>Techn. name/value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC_SDC05</td>
<td>DC_SDC05</td>
</tr>
<tr>
<td>Object Information</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>In Process</td>
</tr>
<tr>
<td></td>
<td>Saved</td>
</tr>
<tr>
<td>Revised Version</td>
<td>Active Version</td>
</tr>
<tr>
<td>Object Status</td>
<td>Active, executable</td>
</tr>
<tr>
<td>Settings</td>
<td></td>
</tr>
<tr>
<td>Dimensions</td>
<td></td>
</tr>
<tr>
<td>DC_SDC05P</td>
<td>DC_SDC05P</td>
</tr>
<tr>
<td>DC_SDC05T</td>
<td>DC_SDC05T</td>
</tr>
<tr>
<td>Calendar Day</td>
<td>0CALDAY</td>
</tr>
<tr>
<td>Document currency</td>
<td>0DOC_CURREY</td>
</tr>
<tr>
<td>DC_SDC05U</td>
<td>DC_SDC05U</td>
</tr>
<tr>
<td>Sales Document Category</td>
<td>0DOC_CATEG</td>
</tr>
<tr>
<td>Navigation Attributes</td>
<td></td>
</tr>
<tr>
<td>Key Figures</td>
<td></td>
</tr>
<tr>
<td>NetValue R/3</td>
<td>DC_NETVAL</td>
</tr>
</tbody>
</table>

Create a DataTransferProcess for Direct Access for the InfoCube and set the generic DataSource you created as the DataSource.
Now you have to open the transformation rules to map the fields from your DataSource to the InfoObjects of the VirtualProvider.

For the VirtualProvider, it is necessary to add the same formula for the InfoObject 0CALDAY as in the standard InfoCube (here: 0SD_C05).

<table>
<thead>
<tr>
<th>Pos</th>
<th>Key Field</th>
<th>IconDescriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VBELN</td>
<td>Sales Document</td>
</tr>
<tr>
<td>2</td>
<td>ERDAT</td>
<td>Created on</td>
</tr>
<tr>
<td>3</td>
<td>VDTYP</td>
<td>Document cat.</td>
</tr>
<tr>
<td>4</td>
<td>NETWR</td>
<td>Net value</td>
</tr>
<tr>
<td>5</td>
<td>VAERK</td>
<td>Doc. currency</td>
</tr>
<tr>
<td>6</td>
<td>AEDAT</td>
<td>Changed on</td>
</tr>
</tbody>
</table>

For this mapping, some business logic has to be considered. When SD documents are created, the field for the creation date ERDAT is filled, and the field for the change date AEDAT is empty. If such a document is changed later on, the ERDAT stays the same, but the AEDAT is updated with the current calendar day. To get the most recent value into the VirtualProvider, we use ERDAT for unchanged documents and AEDAT for changed documents.

**Note:** The logic that is implemented here cannot be copied to your specific reconciliation scenario. It is only valid in our showcase. In your implementation project for the reconciliation scenario, you have to make sure that only and exactly this data is extracted for the VirtualProvider that reflects the business logic of your specific extraction and update process. There cannot be any best practice recommendations; this is highly dependent on your BI implementation.
This formula checks if the AEDAT is greater than or equal to the ERDAT. If this is true, it gives the AEDAT back, if this is false it gives the ERDAT back. 
(ERDAT = creation date, AEDAT = change date)

Finally, you have to activate the direct access for the InfoCube.
6.3.2 MultiProvider

To make reporting that compares the data in BI with the data in the source system possible, two InfoProviders have to be read. This means, a MultiProvider has to be used for reporting.

The MultiProvider must contain the VirtualProvider with direct access and the normal InfoCube.

Choose the common characteristics from the two InfoCubes. In this example, it is the calendar day and the sales document category.
Assign the characteristics to both InfoCubes.

Both boxes must be checked.

Now save and activate the MultiProvider.
6.3.3 DataStore-Object for storing the result of the analysis process

The result of the comparison between the BI InfoCube and the VirtualProvider has to be written to the database, so that the Solution Manager collector can read this result and transfer it to the Solution Manager. This may be realized by using a customer-defined table, but to provide maximum flexibility and re-usability, we decided to use standard BI functionality and thus use a DataStore Object for direct update instead. The “direct update” DSO consists only of one table (the table for active data). Data does not have to be activated as in a standard DSO.

If you use different currencies in your system, you have to add the currency key as the key field. Otherwise, the data records cannot be aggregated and you have no unique key. That leads to an error in the writing process of the analysis process.

As Data Fields in your DataStore-Object you need three InfoObjects for every key figure that you want to compare:

1. Key figure SAP ERP
2. Key figure SAP NetWeaver BI
3. Key figure Difference

It is important that you choose the “Direct Update” in the settings. Otherwise, your analysis process will not work.
6.3.4 Comparison of the key figures

To compare the key figures between BI data and source system data, we used two different approaches. First, we implemented a solution that uses a query with a calculated key figure to compare the values and provide the differences. In addition to that, we implemented a second scenario that does not use a query, but realizes the result calculation by using APD functionality.

Documentation for the Analysis Process Designer can be found in online documentation:

Remarks: Important SAP Note: 794257 - APD General Performance note.
There is a training class available that introduces you to the APD and Data Mining in BI (BW380: BI - Analysis Processes & Data Mining, 2 days).

6.3.5 Analysis Process using a query

This scenario is supported from BW 3.5 onwards. It probably would be your first attempt if you are not familiar with the Analysis Process Designer. However, we found out in our tests with about 6000 data records that it had a worse performance than the analysis process without the query. Nevertheless, if you are using a BI system below BI 7.0 / SP 12, you have to use the scenario with query.

First of all, you have to create a query on your MultiProvider that compares the key figure and calculates the difference.
Subtract the key figure of your source system from the key figure of your InfoCube in the BI-System.
Now create two new filters. The first is for the SD document category “C”.

The second filter is for the timeframe you want to select. For performance reasons, it is not recommended to compare the complete amount of data in the source system with the data in the BI. To keep the runtime of the comparison on a reasonable level, you should choose a small amount of days to be compared, for example, the last week. This implies that the data consistency check has to run regularly, which is also a need of the Business Process Monitoring.

Set a filter on the InfoObject “calendar day” and select a value range between the variable “Current Calendar Day”. Now, click on the selection you added and set an offset for both variables. The offset value defines how many days are added (positive offset) or how many are subtracted (negative offset). If you want to check the last 8 days except today, you have to select from “0DAT (8-)” until “0DAT (1-)”.

It is recommendable that you choose a timeframe before the last delta update (for example, offset 8- until 2- ; this would be one week until the day before yesterday). If you do not restrict the time frame, every record which was created after or during the last delta update in the source system (and which is therefore not yet loaded to BI) will automatically lead to the detection of an “inconsistency” between the source system and the InfoProvider. This is standard behavior of the systems and does not represent an inconsistency. Take care that this value is adapted to the frequency of your delta uploads. You have to consider also if the extractor uses delta queue, or a timestamp, for example.
At the moment, the result of the query contains all data. To write only the inconsistent data records into the DataStore-Object, you have to set up a condition that gives only the records back which has a difference that is not equal to zero.
The query result in BEX Web Analyzer:

<table>
<thead>
<tr>
<th>Calendar Day</th>
<th>Net Value R/3</th>
<th>Net Value BI</th>
<th>Net Value Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.08.2008</td>
<td>33,107,92</td>
<td>22,761,70</td>
<td>10,346,23</td>
</tr>
<tr>
<td>Overall Result</td>
<td>33,107,92</td>
<td>22,761,70</td>
<td>10,346,23</td>
</tr>
</tbody>
</table>

This query can also be used for manual data consistency checks. For daily operations, we recommend the automated data consistency check in combination with SAP Solution Manager Business Process Monitoring. To do so, the query result has to be written to a database object, where SAP Solution Manager can read from.

In the next step, an analysis process is created which writes the query result into a DataStore Object.

Start transaction RSANWB to open the Analysis Process Designer. Create a new analysis process and choose as DataSource the Query.

Select in this screen your query.
As the DataTarget, choose the DataStore-Object.

Connect the query with the filter and click on the blue banner to map the fields from the query on the DataStore-Object.

Your analysis process is finished now.
6.3.6 Analysis Process without Query

The following scenario is supported from BI 7.0 SP12 onwards. The APD is used without a query. The necessary filtering and calculations are implemented by using APD functionality.

This setup showed a better performance in our test with about 6000 data records than the scenario with a query. The query could be created in addition to make manual analysis possible, but it is not used in the data flow towards SAP Solution Manager.

Start the Analysis Process Designer (transaction RSANWB).

Create a new analysis process and choose as DataSource the InfoProvider.

Now, a popup opens. In the first register, you have to select the MultiProvider.
In the second tab strip, you have to select the characteristics and key figures which you want to use from the MultiProvider. In this example, we need as characteristics the calendar day, the document category, and the currency. As key figures, choose the key figures you want to compare - in our example, the order net value from ERP with the order net value from BI.

Now create a filter by clicking on the filter symbol and connect the MultiProvider with the filter.
After you have connected the DataProvider with the filter, you have to define the filter conditions. To do so, double-click on the filter icon.

For “Sales Document Category”, insert “C” for the sales orders. For the filter conditions of the Calendar Day, you have to specify two variables. Push the variables button and choose as variable “0DAT”. This is the variable for the current calendar day. The offset value defines how many days are added (positive offset), or how many are subtracted (negative offset). If you want to check the last 8 days except today, you have to select from “0DAT (8-)” until “0DAT (1-)”. It is recommendable that you choose a timeframe between the last two delta updates (for example, offset 8- until 2-) because every record which was created after or during the last delta update in the source system and is not yet loaded to BI will automatically lead to a temporary difference between the source system and the InfoProvider. However, this is standard behavior of the systems and does not represent an inconsistency. Take care that this value is adapted to the frequency of your delta uploads.
The next step is the aggregation of the data. Click on the aggregation symbol and connect it with the filter.

In the aggregation menu choose the calendar day and the document currency as grouping fields and the two net values as aggregation fields.

After the aggregation, you have to set a filter again on the calendar day with the same filter conditions as your first calendar day filter.
Select on the calendar day…

…and choose the same filter conditions as before.

Your analysis process should look like this now.

As the next step, a formula for the calculation of the difference between the two key figures is created. Click on the formula symbol and connect it to your filter.

Click on the button to add a new formula field. Enter the field name, the description, and the type of the InfoObject. In the formula builder area you must define your formula to calculate the difference.

For calculating the values, it is advisable to subtract the source system’s value from the value in BI, so a difference due to missing values in BI appears as negative result.
Your analysis process after this step:

You should set a filter on the formula field for the “difference” to filter out those data records that have a zero value as difference. If you do not use such a filter, non-inconsistent data records are also written into the DataStore-Object. This leads to an unnecessarily high data volume in the DSO. Anyway, it does not affect the further process, as the Solution Manager Collector program only selects those data records where the “difference” is not equal to zero.
Click on the filter symbol to add a new filter. In the register “Field Selection”, add the difference field and jump to the register “Filter Conditions”:

In the filter conditions you have to push the “More” button and insert in the register “Exclude Single Values” into the column “Single value” a “0.00”. This has the effect that every record that has a zero in the difference field is filtered out.
Your analysis process should look like this now:

Finally, you have to write the result into your transactional DataStore-Object. Therefore select as data target the DSO and select your DataStore-Object into which the data should be written.
Now, connect the DataStore-Object with your last filter and click on the blue banner to make the field assignment.

The complete analysis process:

![Diagram](image)

6.3.7 Result of the Analysis Process

The content of the result table after an analysis run then looks like this:

![Table](image)

The field RECORDMODE is automatically used in a DSO for technical reasons; it is not used for this scenario.

Only data records with a difference between the ERP-value and BI-value are written into this result table. This keeps the amount of data to be transferred to SAP Solution Manager as small as possible. These data records are then read by the Solution Manager Collector program.

You do not need to delete the content of the DSO; this is done automatically by the APD. This means that in this DSO, you always have only the result of the last analysis run.

© 2008 SAP AG
7 Scheduling of the Analysis Process

If you want to automate this consistency check, you have the possibility to schedule the analysis process.

Right-click on the process you want to schedule and select “Schedule Job”.

A new window opens where you can set the time when the job should start. If you want a periodic execution of this job, check the “Periodic job” box and set in the “Period Values” window how often the job should run.

The starting time of the job has to be adjusted with the timeframe you selected in your query or in your analysis process and your delta updates. You must be sure that every record from the timeframe you select on is out of the delta queue when the job starts.
8 Automatized Data Consistency Monitoring

Data Consistency Monitoring is part of the Business Process Monitoring (BPMon) in SAP Solution Manager. It allows you to configure an automatized monitoring of certain consistency checks together with an alert and notification if defined thresholds are reached.

To learn more about Business Process and Interface Monitoring in general, visit the SAP Service Marketplace at [http://service.sap.com/bpm](http://service.sap.com/bpm).

BPMon offers a special data collector (=monitoring object), that is able to interpret the result of the described BI Consistency Check. It counts the number of found inconsistencies in the DataStore-Object, that is, the number of records with a difference value other than zero. In the setup of this data collector you can choose the name of the DataStore-Object, additional filter criteria on your key fields, and multiple result columns to count the number of differences.

As a prerequisite, you need to have the add-on ST-A/PI version 01L installed in your BI system. This contains the coding for the data collector, which is called from the SAP Solution Manager and reads the result from the DataStore-Object.

For detailed instructions on how to configure this data collector, please see the "Setup Guide for the BI Consistency Check Result Collector".

As a starting point, we have a look again at the structure of the generated database table for the example DataStore-Object DC_OC05. It contains two key fields "Calendar Day" and "Currency", and one result column "NetValue Difference".

**Data Browser: Table /BIC/ADC_OC0500 Select Entries**

<table>
<thead>
<tr>
<th>Calendar Day</th>
<th>Currency</th>
<th>Net/Value R3</th>
<th>NetValue BI</th>
<th>NetValue Difference</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.10.2007</td>
<td>EUR</td>
<td>57.200,00</td>
<td>31.200,00</td>
<td>26.000,00</td>
<td></td>
</tr>
<tr>
<td>31.10.2007</td>
<td>EUR</td>
<td>8.000,00</td>
<td>0,00</td>
<td>8.000,00</td>
<td></td>
</tr>
<tr>
<td>26.02.2008</td>
<td>EUR</td>
<td>2.900,00</td>
<td>900,00</td>
<td>2.000,00</td>
<td></td>
</tr>
<tr>
<td>29.04.2008</td>
<td>EUR</td>
<td>19,78</td>
<td>9,89</td>
<td>9,89</td>
<td></td>
</tr>
<tr>
<td>30.05.2008</td>
<td>EUR</td>
<td>10.040,00</td>
<td>40,00</td>
<td>10.000,00</td>
<td></td>
</tr>
<tr>
<td>17.09.2008</td>
<td>EUR</td>
<td>61.325,20</td>
<td>67.771,00</td>
<td>13.554,20</td>
<td></td>
</tr>
<tr>
<td>18.09.2008</td>
<td>EUR</td>
<td>65.300,00</td>
<td>0,00</td>
<td>65.300,00</td>
<td></td>
</tr>
<tr>
<td>05.10.2008</td>
<td>EUR</td>
<td>105.981,91</td>
<td>0,00</td>
<td>105.981,91</td>
<td></td>
</tr>
</tbody>
</table>
8.1 Setup steps in SAP Solution Manager

In SAP Solution Manager, the Business Process Monitoring consists of the so-called BPMon Setup Session and the actual monitoring for a Solution. Within the setup session, you can attach the monitoring object for BI Consistency Check to a business process step (or to an interface step like between ERP and BI) and define the necessary settings.

For this Best Practice document, we assume that inside the Solution Manager the System Landscape and the Solution Landscape is already set up. If not, refer to the general documentation or contact SAP Support who can assist you in implementing Business Process Monitoring.

Solution: ST418_D023057_1 - SAP Solution Manager

As prerequisites, the System Landscape Maintenance, where a logical component is available for your BI system, and the RFC connections, must be working. Furthermore, in Solution Landscape Maintenance you have at least one Business Scenario with Business Processes (for example, Order-to-Cash including the sales order management), and within the Processes several Business Process Steps, for example upload of sales order statistics to BI-System.
The goal is now to be alerted when the sales order net value between ERP and BI differs.

In transaction DSWP, choose your solution and go to "Operations Setup → Setup Business Process Monitoring".

Choose the business process, for example, Sales Order Management.

Change "Setup Business Process Monitoring - ST418_D023057_1"
Choose a business process step, or if defined rather an interface step.

**Change "Setup Business Process Monitoring - ST418_D023057_1"**

Attach an application monitor to the interface (here: to the receiving BI-System "B70"). The Monitor Name is the data collector for the BI Consistency Check (technical name DCBICCCRC). In addition, enter a free text as monitoring name.

**Change "Setup Business Process Monitoring - ST418_D023057_1"**

On tab strip "Detail information", double-click on Counter 001 to configure the parameters for this monitoring object. Enter the name of the Compare Result DSO, for example, DC_OC05. In addition, you can optionally define up to three sets of filters, consisting of the field name and the corresponding selection options. This can be used if the result DSO contains more records than you want to monitor, for example, if you want to distinguish certain organizational attributes like a country or sales organization, by setting up multiple monitoring objects with different filter criteria and different alert receivers.
On tab strip "Key figures", select as many key figures as you have difference result columns to monitor (up to five). Additionally, you can choose key figure "Age of last consistency check result" to monitor the last successful execution of the analysis process.

On tab strip "Monitoring Schedule", set up how often the data collection should run, for example, each weekday at 8am. Synchronize the monitoring schedule with the periodicity of the scheduled analysis process in BI. Running the data collector more frequently than the analysis process would not show any newer result.
Now, for each key figure, you have to specify the corresponding column (field name) of the DataStore-Object, in which the differences should be counted. Furthermore, you can define some alert text template with variables as placeholders (see setup guide for more details). Additionally, specify thresholds that must be exceeded to raise a yellow or red alert, based on the number of counted differences.

As last step, the customizing for the new monitoring object must be generated and activated in the monitored remote system using button **Generate+Deactivate+Activate**.

### 8.2 Monitoring in SAP Solution Manager (Overview)

In SAP Solution Manager, there are two graphical User Interfaces for the alert overview in Business Process Monitoring.

**Option 1: Transaction DSWP → Operations → Business Process Monitoring**

The first overview screen shows the scenarios, like Order-to-Cash, with the affected business process (sales order management).
By choosing the business process, the next overview screen displays the graphical view of the business process steps and interfaces.
Clicking on an alert icon displays the alert texts for this interface.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Alert Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>B70: Consistency of Sales Order Net Value between ERP and BI/CoS before last consistency check result</td>
<td>Compare DC_OC05: Last analysis run ended 1 day ago (09.10.2008 00:27:14)</td>
</tr>
<tr>
<td>☑</td>
<td>B70: Consistency of Sales Order Net Value between ERP and BI/CoS before last consistency check result</td>
<td>In DC_OC05 the Order Value shows 8 differences in /I/BIC/DCNET/VALD</td>
</tr>
<tr>
<td>☑</td>
<td>B70: Consistency of Sales Order Net Value between ERP and BI/CoS before last consistency check result</td>
<td>BIC/DCNET/VALD</td>
</tr>
</tbody>
</table>

By clicking on the alert text, you can navigate into the hierarchical monitoring session with further navigation options like detail displays (see section 8.3).

**Option 2: Transaction SOLMAN_WORKCENTER**

"My Home" entry screen with a general overview:

**SAP Solution Manager: Work Centers**

Overview on alerts per solution:

© 2008 SAP AG
Overview by business process:

**SAP Solution Manager: Work Centers**

- Integration
- Order-to-Cash
- Data Exchange
- Sales Order Management
With the button [Goto Monitoring session], you can navigate into the hierarchical monitoring session with further navigation options like detail displays (see section 8.3).

### 8.3 Monitoring Session in SAP Solution Manager

From both transaction DSWP and transaction SOLMAN_WORKCENTER, the forward navigation ultimately leads into the monitoring session. It provides a hierarchical view on all alerts for the entire scenario (all business processes, steps and interfaces).

For each monitoring object assigned to a business process step or interface, you can see the rating for its key figures, together with the alert history. For each alert, you can see the time it was determined, the alert message text, the measured value and some additional information (here, for example, the name of the analysis process and the DSO's table name).

When selecting an alert, you can perform further actions, like creating a service desk message, confirming an alert, or performing a detail analysis.

With the button [Detail Info.], you can reach the assigned detail analysis which depends on the selected key figure.

For the difference columns, the detail analysis shows the content of the DSO's database table, filtered by the monitoring objects additional filter criteria. The selected difference column is highlighted in red.
Best Practice: Business Process Monitoring of Data Consistency between SAP BI and Source Systems

Detail display of DSO DC_OC05 (8 lines)

<table>
<thead>
<tr>
<th>Calendar Day</th>
<th>Currency</th>
<th>NetValue R/3</th>
<th>NetValue BI</th>
<th>NetValue Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.10.2007</td>
<td>EUR</td>
<td>57.200,00</td>
<td>31.200,00</td>
<td>26.000,00</td>
</tr>
<tr>
<td>31.10.2007</td>
<td>EUR</td>
<td>8.000,00</td>
<td>0,00</td>
<td>8.000,00</td>
</tr>
<tr>
<td>26.02.2008</td>
<td>EUR</td>
<td>2.900,00</td>
<td>900,00</td>
<td>2.000,00</td>
</tr>
<tr>
<td>29.04.2008</td>
<td>EUR</td>
<td>19,78</td>
<td>9,89</td>
<td>9,89</td>
</tr>
<tr>
<td>30.05.2008</td>
<td>EUR</td>
<td>10.040,00</td>
<td>40,00</td>
<td>10.000,00</td>
</tr>
<tr>
<td>17.09.2008</td>
<td>EUR</td>
<td>81.325,20</td>
<td>67.771,00</td>
<td>13.554,20</td>
</tr>
<tr>
<td>18.09.2008</td>
<td>EUR</td>
<td>65.306,60</td>
<td>0,00</td>
<td>65.306,60</td>
</tr>
<tr>
<td>05.10.2008</td>
<td>EUR</td>
<td>106.981,91</td>
<td>0,00</td>
<td>106.981,91</td>
</tr>
</tbody>
</table>

EUR = 231.852,60

For the age key figure, you see instead the monitoring transaction of the analysis runs, to check the status of the last successful analysis.

Monitor for Analysis Run

© 2008 SAP AG