How To...
Data Recovery in SAP BW Systems
Version 1.00 - December 2005

Applicable Releases:
SAP NetWeaver ‘04 and BW 3.x
1 Scenario

A major incident has occurred within your SAP BW system, or one of its source systems, that has resulted in data inconsistency. You consider recovering one of the systems incompletely, or recovering the data in the inconsistent objects individually. The steps described in this paper describe recovery procedures in the event of accidental deletion of data from ODS Objects or InfoCubes. In order to perform the steps for ODS Object or InfoCube data recovery; the possibility to reload the data back from a copied system (or database) is required. The status of this copied system (or database) must be recovered to the point in time just before the disruptive incident occurred. Furthermore, approaches are discussed herein which handle an event in which data inconsistencies arise due to events occurring in an SAP source system.

2 Introduction

The techniques described in this paper are not officially supported, and should be used only in exceptional situations if advised by SAP Active Global Support (or a knowledgeable consultant). This discussion will detail various recovery strategies for SAP BW data. Please note that some of the approaches described in this paper involve activities performed directly on database tables, which are not provided by the application’s utilities. Generally, activities performed outside of the SAP application (directly in the database) are not included within the normal scope of SAP support, and these techniques are no exception.

Please refer to the document Backup and Restore for mySAP Business Suite, for a general overview of the topic, especially the section Incomplete Recovery. The document can be found with the following link: https://service.sap.com/~sapidb/011000358700004447112001E alternatively, available at http://service.sap.com/solutionmanagerbp → topic area → Backup and Restore for mySAP Business Suite.

Please read SAP Note 739863 and related notes for more background information. Data reconciliation scenarios are described in the area https://service.sap.com/~sapidb/011000358700003015092005, alternatively available at http://service.sap.com/bi → BI InfoIndex → Data Quality

The approach taken in this document helps prevent an incomplete recovery (for the reasons described in the document mentioned above) by utilizing data sourced from a database backup or a copy (e.g. stand-by database). If no stand-by database is available, the procedures described here require that a restore / recovery be performed of the SAP BW database onto another server, from which the necessary data can be obtained for use in the SAP BW production system.

There are two different types of inconsistency scenarios for which repair options are described in this document:

1. Data was changed in an inconsistent manner in SAP BW only and can thus only be restored using data from an SAP BW copy. An accidental deletion of a request is an example.
2. Inconsistencies exist not only within SAP BW, but also between SAP BW and one of its SAP source systems. For example, data has been loaded with an
erroneously coded transformation routine, or some records were missed in the extraction process (possibly due to flawed exit coding).

In general, you should always utilize SAP BW application functionality to recover data whenever possible, such as:

- Reload data from the PSA
- Reload data from an ODS Object or an InfoCube (in a layered architecture, EDW approach)
- Repeat a delta load.

Thus, you should apply the options described in this document only if SAP BW standard functionality cannot be utilized.

In the event that an incomplete recovery of either the SAP BW system or a source system is necessary, please read section 3.5 of this paper.
The Step By Step Solution

All SQL-statements provided here describe the recommended general procedures, and must be adjusted according to your database type, and its requirements (for example add storage clauses for partitions). The addition "@BWP" (production system) and "@BWC" (stand-by / copy system) indicates the database from which the table holding the "good" data is read. These steps should be performed by a database administrator (or equivalent). The general approach is to re-acquire the data to the production SAP BW system from a separate stand-by or copy database of that system. It is further assumed that database utility is employed where the databases can be remotely connected and data transferred at the database level. You can choose from two different options:

Option 1. Replace the entire table in BWP with the corresponding table in BWC. This approach can be utilized if BWC contains the exact set of data that you would like to transfer back into BWP.

```
Delete from <table>@BWP where <selection field> = <value>;
Insert into <table>@BWP select * from <table>@BWC where <selection field> = <value>;
```

Option 2. Insert only the data that exists in BWC but not in BWP. This approach impacts only the absolutely required table entries, thus, there is no risk of changing the existing records in BWP.

```
Insert into <table>@BWP select * from <table>@BWC where <selection field> = <value> and <key> not in (select <key> from <table>@BWP where <selection field> = <value>);
```

We will refer to these options as restore option 1 or 2 respectively. In the examples ahead only the table name, key, selection field and value are specified. The table names of generated SAP BW tables are prefixed with /BI0/ or /BIC/ depending on whether it is a business content object or a customer-defined object. The examples provided contain customer-defined object names, and if it is required to restore data to an SAP-delivered business content object, adjust the table name accordingly.

3.1 InfoCube requests have been deleted

In this example, through administration error, good data was accidentally deleted from an InfoCube, using the "delete by request" option. Additional requests may have been loaded after the initial data loss incident occurred (which are not contained in the other, restored system BWC). If so, additional considerations are necessary if restore option 1 is to be employed, because the data in the requests loaded in BWP subsequent to the point-in-time of the recovered system BWC would be lost such an operation. The requests loaded subsequent to the point-in-time of the recovered system can be compressed, and you can still utilize restore option 1, and reload these requests later from PSA (if available); alternatively it would be possible to utilize functionality such as export DataSource or Open Hub to copy the data from the subsequently loaded requests to a holding place to reload later. If such a reload strategy is employed, it is important to consider update rules and transfer rules – there should be no inconsistency with those built into the normal data load pathway.
The InfoCube used as an example is named \textit{K0SD\_C01}.

1. The first step is to verify which requests are missing. If you can log on to BWC with SAP GUI, call the Administrator Workbench (transaction RSA1). From the context menu of the InfoCube choose \textit{manage}. In the Requests tab you can see the SID (Request ID) as well as corresponding RNR (Request Number) of the requests. Compare these entries with the ones in BWP to identify missing requests.

2. Optionally you could utilize SQL to identify missing request, you can get a list of request IDs (SID) using these (example) statements as a template. Verify that these are the requests that should be restored to BWP.

   \begin{verbatim}
   SQL> Select RNR from RSICCONT@BWC where ICUBE = 'K0SD\_C01' and RNR not in
       (select RNR from RSICCONT@BWP where ICUBE = 'K0SD\_C01');
   SQL> Select SID from /BI0/SREQUID@BWC where
       RNR in (<list of RNR retrieved with previous statement>).
   \end{verbatim}

3. Restore the necessary entries to the table RSICCONT, which contains the updated InfoProviders per request.

   Run restore option 1 or 2 with
   \begin{verbatim}
   <table> = RSICCONT
   <selection field> = ICUBE
   <value> = K0SD\_C01 (your InfoCube name)
   <key> = RNR
   \end{verbatim}

4. Restore the necessary entries to the table RSREQICODS, which contain the updated InfoProviders and PSA per request.

   Run restore option 1 or 2 with
   \begin{verbatim}
   <table> = RSREQICODS
   <selection field> = TABNAME
   <value> = K0SD\_C01 (your InfoCube name)
   <key> = RNR
   \end{verbatim}

5. Restore the necessary entries to the table RSMONICDP, which contains information about the data packets in requests.

   Run restore option 1 or 2 with
   \begin{verbatim}
   <table> = RSMONICDP
   <selection field> = ICUBE
   <value> = K0SD\_C01 (your InfoCube name)
   <key> = RNR
   \end{verbatim}
6. Restore the corresponding dimension table entries for the data package (request) dimension.

Run restore option 1 or 2 with:
<table> = /BIC/DK0SD_C01P
<key> = SID_0REQUID
Note: Here no selection is required since the table is InfoCube specific. Don’t forget to replace K0SD_C01 with your InfoCube name.

7. Caution: perform this step only if no other requests have been loaded, which are not contained in the copy system BWC: Restore the record for RSMDATASTATE containing the status of the data / requests in InfoProviders.

Important: Run restore option 1, but DO NOT run restore option 2 because an update of one record is required, not an insert! If more than one record is affected by the SQL statement, the statement must be wrong, and therefore rollback immediately - this table is very important for system consistency.
<table> = RSMDATASTATE
<selection field> = INFOCUBE
<value> = K0SD_C01 (your InfoCube name)

8. When restoring the data to the InfoCube fact table, performance of the restore operation can be improved by dropping the indexes first. In BWP with SAP GUI, call the Administrator Workbench (transaction RSA1). From the context menu of the InfoCube choose manage. In the Performance tab choose Delete Indexes (Immediately).

Select partition_name from dba_tab_partitions@BWC where table_name = '/BIC/FK0SD_C01' and partition_name not in (select partition_name from dba_tab_partitions@BWP where table_name = '/BIC/FK0SD_C01');

Alter table "/BIC/FK0SD_C01"
Split partition "/BIC/FK0SD_C010000000607" at (0000000606) into (partition "/BIC/FK0SD_C010000000606",
partition "/BIC/FK0SD_C010000000607");
reloaded for 606. The storage parameters in the statement may vary, compare existing partitions to determine the correct storage parameters.

11. Restore data for the relevant request.

```
insert into "/BIC/FK0SD_C01"
select * from "/BIC/FK0SD_C01"@BWC where
KEY_K0SD_C01P >= 603 and
KEY_D0SD_C01P < 606;
```

12. Please perform the steps from section 3.4
3.2 Restoring data to an InfoCube which has been deleted (but not deleted by request)

In the event that data has been accidentally deleted from an InfoCube entirely or selectively (as opposed to deletion by request), it is necessary to restore the entire InfoCube. The steps described in this section can be used for any scenario in which InfoCube data has been changed accidentally, for example selective deletion or deletion of the entire InfoCube's data. If the object itself was deleted (the meta data) you have to first re-transport the InfoCube into BWP from the development system and then perform the steps described in this section. The InfoCube in BWC must contain all necessary data required for recovery.

You may have loaded additional requests after the deletion event has occurred (which are not contained in the copy system BWC). If so, you have to reload these requests later from PSA if available or unload them from the InfoCube to some other structure first (copied InfoCube, Open Hub), in order to reload them later. Please keep in mind that update and transfer rules may have a significant impact on the data (if/when) reloading data from other structures.

1. Restore the necessary entries to the table RSICCONT, which contains the updated InfoProviders per request.

Run restore option 1
<table> = RSICCONT
<selection field> = ICUBE
<value> = K0SD_C01 (your InfoCube name)

2. Restore the necessary entries to the table RSREQICODS, which contains the updated InfoProviders and PSA per request.

Run restore option 1
<table> = RSREQICODS
<selection field> = TABNAME
<value> = K0SD_C01 (your InfoCube name)

3. Restore the necessary entries to the table RSMONICDP, which contains information about the data packets in requests.

Run restore option 1
<table> = RSMONICDP
<selection field> = ICUBE
<value> = K0SD_C01 (your InfoCube name)

4. Restore the necessary records for RSMDATASTATE, which contain the status of the data / requests in the InfoProviders.

Important: Run restore option 1, but DO NOT run restore option 2 because an update of one record is required, not an insert! If more than one record is affected by the SQL statement, the statement must be wrong, and therefore rollback immediately - this table is very important for system consistency.

<table> = RSMDATASTATE
<selection field> = INFOCUBE
<value> = K0SD_C01 (your InfoCube name)
5. If the BI system is running on an Oracle database, the InfoCube F fact table must be recreated, and include the full partitioning statements in the table structural definition when recreating the F fact table. (/BIC/FK0SD_C01) First drop this table in BWP, then export the table structure in BWC with using Oracle tools or statements, and import the meta data definition into BWP (or create the partitioned table manually with DDL statements).

   Drop table "/BIC/FK0SD_C01"@BWP

6. Restore the F-Fact table

   Run restore option 1 (restore entire table)
   <table> = /BIC/FK0SD_C01

7. Restore the E-Fact table

   Run estore option 1 (restore entire table)
   <table> = /BIC/EK0SD_C01

8. Identify all dimension tables. The dimension tables are called /BIC/DK0SD_C01x where x designates dimension number. If a dimension is a line item dimension, there is no dimension table (instead there is a view on the SI table of the characteristic with the same name).

9. Restore the dimension tables

   Run restore option 1 for all dimension tables (restore entire table)
   <table> = /BIC/DK0SD_C01x (replace x with dimension number for all dimensions which are NOT line item dimensions)
10. Please perform the steps from section 3.4

3.3 Scenario: ODS object data has been deleted, restore required.

The restore technique for an ODS Object differs from that of an InfoCube. ODS data cannot be restored at the request level, but instead the entire set of ODS data must be restored. Data from individual requests cannot be restored, because one request could affect any record in the active data table (due to the ODS after image handling). The steps described in this section can be utilized for any scenario in which ODS object data has been erroneously deleted. Examples of such an event include selective deletion, request deletion, or deletion of all of the ODS object’s data. If the ODS object itself was deleted (the meta data), first the ODS object must be transported into BWP from the development system, and then perform the steps described below. If transport is required, ensure the meta data definition that is transported matches the meta data definition that had been deleted.

If additional requests have been loaded subsequent to the event which resulted in data loss (which are not contained in the copy system BWC), these requests must be reloaded later from PSA, if available; another option is to load them from the ODS object to some other structure first (copied ODS Object, Open Hub), in order to reload those requests later. Please keep in mind that update and transfer rules may have a significant impact on the data (if/when) reloading data from other structures.

1. Restore the necessary table entries to the table RSICCONT, which contains the updated InfoProviders per request. In this example, the ODS object name K0SD_ODS is used. Run restore option 1 or 2 with
   <table> = RSICCONT
   <selection field> = ICUBE
   <value> = K0SD_ODS (your ODS Object name)

2. Restore the necessary table entries to the table RSREQICODS, which contains the updated InfoProviders and PSA per request. Run restore option 1 or 2 with
   <table> = RSREQICODS
   <selection field> = TABNAME
   <value> = K0SD_O (your ODS Object name)

3. Restore the necessary table entries to the table RSMONICDP, which contains information about the data packets in requests. Run restore option 1 or 2 with
   <table> = RSMONICDP
   <selection field> = ICUBE
   <value> = K0SD_ODS (your ODS Object name)
4. Restore the necessary table entries to the table RSODSACTREQ, which contains load and activation request information.

Run restore option 1 or 2 with
<table> = RSODSACTREQ
<selection field> = ODSOBJECT
<value> = K0SD_ODS (your ODS Object name)

5. Restore the necessary record to RSBODSLOGSTATE, which contains the status of the data / requests in InfoProviders.

Important: Run restore option 1, but DO NOT run restore option 2 because an update of one record is required, not an insert! If more than one record is affected by the SQL statement, the statement must be wrong, and therefore rollback immediately - this table is very important for system consistency.

<table> = RSBODSLOGSTATE
<selection field> = ODSOBJECT
<value> = K0SD_ODS (your ODS Object name)

6. Restore the necessary entries to table RSBODSLOG, which contains the requests of the ODS change log.

Run restore option 1 or 2
<table> = RSBODSLOG
<selection field> = ODSOBJECT
<value> = K0SD_ODS (your ODS Object name)

7. Restore the necessary record to table RSMDATASTATE, which contains the status of the data / requests in InfoProviders.

Important: Run restore option 1, but DO NOT run restore option 2 because an update of one record is required, not an insert! If more than one record is affected by the SQL statement, the statement must be wrong, and therefore rollback immediately - this table is very important for system consistency.

<table> = RSMDATASTATE
<selection field> = INFOCUBE
<value> = K0SD_ODS (your ODS Object name)

8. If your system runs on an Oracle database, the change log table must be re-created, including its partitions. First identify the change log table and drop it in BWP. (You can identify in through the context menu option for ODS object manage. In the Contents tab you can push the button change log to view the table name). Then export the table definition in BWC with Oracle tools (such as exp), and then import it into BWP (or create the table and its partitions manually).
9. Restore the change log table.

Run restore option 1 (restore entire table)
<table> = /BIC/B0014845000 (your change log table)

10. If your system runs on an Oracle database, it is necessary to restore the application-specific partitioning information of the change log table.

Run restore option 1
<table> = RSTSODSPART
<selection field> = ODSNAME_TECH
<value> = /BIC/B0014845000 (your change log table)

11. Restore the active data table.

Run restore option 1 (restore entire table)
<table> = /BIC/AK0SD_ODS00

12. Restore the activation queue.

Run restore option 1 (restore entire table)
<table> = /BIC/AK0SD_ODS40

13. Please perform the steps from section 3.4.

3.4 Post-Recovery Steps

As the procedures outlined previously include SQL statements which have been executed directly on the database, some aspects of the application may no longer be consistent with the database:

- Subsequent (downstream) data targets in the data flow may no longer be consistent with the restored object, or the delta mechanism may be rendered defunct. For example there may be an InfoCube K0SD_AGG that contains aggregated data that is loaded from another InfoProvider K0SD_ODS as DataSource.
- Aggregates are no longer by consistent with the InfoCube.
- The OLAP cache is no longer consistent with the InfoProvider.
- Pre-calculated reports or variables are no longer consistent.

If indexes were dropped in the process, recreate them before for performance reasons.

Helpful SAP Notes (for Oracle databases):
1. If you have dropped indexes recreate them.

2. If you find the downstream data target K0SD_AGG in an inconsistent state, with the DataSource InfoProvider K0SD_ODS, delete the delta initialization, delete the data in K0SD_AGG that has been loaded from K0SD_ODS, and reinitialize the delta. Note that this may be challenging if K0SD_AGG is loaded from different DataSources.

3. If the data in K0SD_AGG is consistent with the data in K0SD_ODS but the delta mechanism is defunct, you can perform a delta initialization without data transfer to restore the delta mechanism.

4. Deactivate, reactivate and fill all aggregates on K0SD_C01.
5. Empty the entire OLAP Cache, or delete all OLAP Cache entries manually in transaction RSRCACHE that contain data from the K0SD_C01 or K0SD_ODS. Note that MultiProviders may include one of these InfoProviders - all OLAP cache entries of queries belonging to that MultiProvider must be deleted as well. These steps must be performed for all application servers, and application server independent and all cache locations (memory, file, cluster, and blob).

6. Re-calculate any pre-calculated items (reports, variable values, etc).

7. Consider the implications for external systems, if you use the Open Hub service based on an InfoProvider, which has been restored.

8. Check the scheduling of processes involving the object that has been recovered. Deletion operations may have an impact on these processes.

3.5 Inconsistencies between SAP BW and source systems

There are several causes which can lead to inconsistencies between an SAP BW and a source system:

Case 1: Erroneous customer exits transfer or update rules or other causes (source-system based) of incorrect data being loaded.
Case 2: Incomplete recovery of the source system without recovering the SAP BW system to the same point in time.
Case 3: Incomplete recovery of the SAP BW system without recovering the source system to the same point in time.

Preliminary remarks:
The approaches described in the following passages focus on an SAP source system, particularly a mySAP ERP system. Nevertheless, the rationale behind the solutions described applies to any type of source system.
In case of doubt, DataSources for which a delta mechanism is used need to be reinitialized (Delta-Initialization) to guarantee consistency. In some situations this will not be necessary as described below. For some mySAP ERP DataSources where the delta mechanism is based on timestamps, there is a possibility to reset
the timestamp instead of reinitializing - in other words, data with a higher (more recent) timestamp than the new (reset) timestamp would be extracted again. Note that this procedure is only valid if you have one and only one SAP BW system as target system for the DataSource.

If you have to reinitialize the delta mechanism, you may be able to delete and reinitialize data selectively (for example only this year’s data) because a large amount of the data (for example older data) may not be affected whatsoever by the inconsistency.

Conversely, if older data is needed, but is no longer available – neither in the source system nor in an archive (or the extractor can’t load from an archive) – there is no alternative but to delete and reload selectively to prevent data loss.

Important SAP Notes:
- 602260: Procedure for reconstructing data for BW
- 691721: Restoring lost data from a delta request
- 739863: Repairing data in BW
- 731682: Backup in BW and OLTP: Experiences, risks & recommendations

**Case 1:** In the event that incorrect routines transfer or update rules have corrupted the data, you can delete and reload the data from PSA (if available) after fixing the transformation errors. However, if the relevant data in the PSA is corrupted itself, or if the data is not available in the PSA, then the nature of the issue is equivalent to that which is described in Case 3; follow the steps outlined there, to repair the data accordingly.

**Case 2** can be addressed by identifying and deleting the requests that have been loaded from the source system since the point in time to which the system is recovered. We can distinguish four different sub-cases (SC) for each DataSource. These sub-cases can be distinguished by comparing the recovery point of the source system with the date of the requests in the BW system. The following illustration depicts the nature of these different sub-cases.
SC 1: Source system is restored to a point after the last data upload to BW

Since the last delta data was extracted there will be no data inconsistency between the newly recovered source system and BW. Thus there is no additional activity required in BW.

SC 2: Source is set back to a point when the last data extraction was running

If the entire extraction dataset has been loaded to PSA completely, you can load the data from the PSA to the data targets. Otherwise, set the load status to red and restart the load. If the load was a delta load a delta repeat will automatically be triggered, loading the data from the last delta plus all new data that has been collected in the source system since it’s point-in-time recovery.

SC 3 and 4: Source is set back to a point before the start of the last data extraction / load.

In this case, any data extractions and loads that had performed since the date/time of the point-in-time recovery in the source system must be deleted in the BW system.

The recommended actions are as follows:

If the requests loaded more recently than the recovery point-in-time into InfoCubes were not compressed (Data is available in F-fact table):

1. Identify the request-ID \((ID^w)\), which had been loaded most recently to BW (in other words, the last request loaded from current state of the recovered source system). The corresponding request-ID can be found in the source system table ROOSPRMSC; for each DataSource, the table field DELTARNR contains this request-ID.

2. In BW, delete all request-IDs loaded after \(ID^w\) in the relevant InfoProviders (ODS Object, InfoCube). You may want to consider also
deleting master data that is not (no longer) referenced in the InfoProviders after deletion of the latest requests.

3. In BW, delete all request-IDs that had been loaded after \( ID^w \) from the PSA. (Prior to deletion this data may be helpful to recreate some lost data in the source system.)

However, if the requests loaded more recently than the recovery point-in-time BW were compressed in InfoCubes (which means that the data is available in the E-fact table and corresponding request-IDs were deleted in the InfoCubes during compression), there are three alternatives:

1. A reverse post data load of the relevant Request-IDs if possible.
2. Delta re-initialization.
3. Restore the complete BW system to a point in time before BW InfoCube compression took place. This point should be close after the recovery point of the source system, which may result in being in SC1 status. It is important to note that this alternative may impact consistency, with respect to data loaded from other source systems, or other DataSources, and thus may not be applicable, depending on the particulars of your system landscape.

**Case 3** can be more difficult than other cases, because the source system typically preserves only the most recent delta dataset, but no older requests. We can distinguish four different sub-cases (SC) for each DataSource. The distinction is done by comparing the recovery point of the SAP BW system, with the number of the requests that had been extracted from the source system, and were loaded into BW more recently than the recovery point-in-time.
SC 1: SAP BW is set back to a point more recent that the last data load. Since the last delta was loaded to SAP BW, there is nothing to be done with respect to extraction, data is consistent in this sub-case.

SC 2: SAP BW is set back to a point in time during which an extraction / load was running.

The last data upload from a source system is not completed.

Actions:
1. Set the status of the last upload to red.
2. Restart the upload job.

There are no concerns for data consistency in this sub-case (2).

SC 3 and 4: SAP BW is set back to a point-in-time prior to the start of the last data extractions / loads.

Keep in mind that SAP BW has lost the request-ID(s) due to the incomplete recovery, lost data cannot be recovered by repeating the upload jobs (only the last request-ID is preserved in the delta queue). In this case, you must re-initialize loads for the affected delta DataSources (master data also). Alternatively, for some DataSources the extraction timestamp can be reset for some as follows:

Please read the following SAP Notes in which prerequisites and procedures are described in detail.
- 422173: Information: BW/CO-PA upload data inconsistency correction
- 836288: BW-BCT-CO-OM: BWOM2_TIMEST maintenance
- 860500: BW-BCT-FI: Wiederholung von Delta-Extraktionen

We describe a step by step solution for resetting a timestamp with the example of DataSource 0CO_OM_CCA_9.

1. In this example, it is determined the best strategy to reset the timestamp of DataSource 0CO_OM_CCA_9. Execute the program BWOM2_TIMEST in the source system (transaction SE38).
2. Enter the name of your DataSource and execute.

3. You see the list of extractions that were executed for this DataSource.

4. Assume that the last delta request that was correctly loaded was the one on April 20th, 2005 at 1:28:50 pm (In this example, afterwards, e.g. the update rules routines contained corrupting transformations; or the system is recovered to a point in time after this request was loaded successfully and before the next delta request). Change the Last TS column such that this request is flagged as the last successful one. The next delta extraction will get the data that has been loaded subsequent to this request.