



Data Consistency Check for Logistics

Best Practice for Solution Management

Version Date: October 2008

Contents

1.	Introduction.....	3
1.1	Applicability, Goals, and Usage of this Document.....	3
1.1.1	Goal of this Best Practice Document and Assessment Guide.....	3
1.1.2	How to use this Best Practice.....	3
1.1.3	Legend.....	4
2.	General Best Practice Procedure for Inconsistencies.....	5
2.1	Background Information.....	5
2.1.1	Definition of Terms and Types of Inconsistencies.....	5
2.1.2	Typical Root Causes for Inconsistencies.....	6
2.2	Overall Procedure.....	9
2.3	Assessment Guide.....	13
2.3.1	Understanding the Situation and Inconsistency.....	13
2.3.2	Process Understanding.....	15
3.	Tools and Procedures within an SAP Environment.....	18
3.1	Tools to Check Consistency Between ECC/CRM and Within CRM.....	18
3.1.1	Data Exchange for the Most Common CRM Scenario - CRM Sales.....	18
3.1.2	Data Exchange for the CRM Scenario CRM Mobile.....	19
3.1.3	DIMa – The Tool to Detect and Repair Inconsistencies Between SAP ECC and SAP CRM.....	19
3.1.4	Analysis and Correction Reports in Data Exchange of One Order Documents Between SAP CRM and R/3.....	20
3.1.5	Miscellaneous Check, Correction and Repair Reports in CRM.....	22
3.2	Tools to Check Consistency Between ECC/SCM and Within SCM.....	23
3.2.1	Introduction.....	23
3.2.2	Product Allocation (PAL): Internal Inconsistencies Within SCM or Between ECC/SCM.....	23
3.2.3	Time Series of DP and SNP.....	25
3.2.4	Shipping and Vehicle Scheduling.....	25
3.2.5	Integration Models (External Consistency).....	25
3.2.6	Internal Consistency Check for SCM regarding LC <-> DB.....	25
3.2.7	Consistency at Interface: Transaction Data (External Consistency).....	26
3.2.8	Temporary Quantity Assignments (TQAs).....	27

Best Practice: Data Consistency Check

3.2.9	Transaction Data: All Order Types and Stock (External Consistency)	33
3.2.10	Sales order requirements and SD mapping tables (external consistency).....	35
3.3	Tools to Check Consistency within ECC	37
3.3.1	Tools for Processes Involving SD and LE	37
3.3.2	Tools for Processes Involving MM.....	47
3.3.3	Inconsistencies between MM and FI.....	49
3.3.4	Tools for Processes Involving WM	51
3.3.5	Tools for Processes Involving PP.....	52
3.3.6	Tools for Processes Involving PS.....	53
3.3.7	Tools to Check the Logistics Information System	54
3.4	Tools not Depending on a Particular System	55
3.4.1	Tools for ALE Consistency Check	55
3.4.2	Tools for ALE Recovery	55
3.4.3	Generic Consistency Check for Two Linked Tables in One System	56
3.4.4	The Generic External Compare	60
4.	Appendix.....	61
4.1	General Roadmap for Analysis.....	61
4.2	Dos and don'ts for Data Consistency	64
4.3	Advantages and Disadvantages of Different Recovery Methods	64
4.4	Further Information	67
4.4.1	Related information	67
4.4.2	Troubleshooting	67

1. Introduction

1.1 Applicability, Goals, and Usage of this Document

To ensure that this Best Practice is the one you need, consider the following goals and requirements.

1.1.1 Goal of this Best Practice Document and Assessment Guide

This Best Practice provides information how to monitor data consistency and what you should do once an inconsistency is reported. The document provides the general monitoring procedure as well as details for several areas where potentially business critical inconsistencies could occur. The more detailed sections should facilitate the use of available consistency check tools by providing the general description and logical design of such reports including relevant business data needed for some frequent instances of inconsistencies. The goal is to enable you to use them and to understand the guiding principles behind these tools, so that they may be available as templates for specific cases involving non-SAP systems.

The Best Practice document is divided into two major areas: Section 2: General Best Practice Procedure for Inconsistencies provides background information about typical root causes of inconsistencies, as well as the basic questions you need to ask to identify the most likely root causes for the specific case you encounter. Section 3: Tools and Procedures within an SAP Environment describes the most important tools available from SAP and how they can be used to monitor data consistency on a regular basis and correct possible inconsistencies. Section 4: Appendix provides additional information like analysis roadmaps and a comparison of different recovery methods.

In addition to the consistency check methods, the Best Practice document provides basic deciding factors for recovery methods to determine which is preferable in case of data loss and how to evaluate them for a given case. Please check the Best Practice Document “Business Continuity Management for SAP System Landscapes” for a more specific and detailed view on the area of data recovery and business continuity.

1.1.2 How to use this Best Practice

Read through section 2: General Best Practice Procedure for Inconsistencies to get background information about inconsistencies and common root causes for inconsistencies. Afterwards, you should understand the general procedure so that you can deal with specific new cases. The result of working through this section will be a flow chart for the business processes where the difference has been reported, including an understanding of the logical steps, as well as an understanding of the involved technical data.

Afterwards, you can set up your data consistency monitoring and investigations of specific root causes using the information available in section 3: Tools and Procedures within an SAP Environment. If you have enhanced your solution, you should consult section 3: Tools and Procedures within an SAP Environment as well to check whether some reports could be used as a template in your situation. You will find a clear link to the corresponding section in the generic part.

Section 4: Appendix provides a graphical roadmap and deciding factors for typical data recovery methods. You can use this section to get a quick overview as well as a basis to discuss your own goals and procedures.

1.1.3 Legend



This symbol indicates a paragraph from where you can navigate to another section of this document for more detailed information on a topic.



This symbol indicates a paragraph from where you can navigate to another document within the SAP Service Marketplace for more detailed information on a topic.



This symbol indicates an important paragraph where you find special hints or best practices.

2. General Best Practice Procedure for Inconsistencies

2.1 Background Information

This part will build up the basic understanding needed to deal with inconsistencies and is the base line for the more detailed discussions later on in this document.

2.1.1 Definition of Terms and Types of Inconsistencies

It is very important to differentiate between the terms *Difference* and *Inconsistency*. While *Difference* relates to a temporary data mismatch that occurs due to processing time of asynchronous processes, *Inconsistency* means a permanent mismatch that does not disappear once all system activities are completed successfully.

Real inconsistencies can be classified by the different root causes and general behavior: A *Technical Inconsistency* is everything that can be found on a database level and needs appropriate correction in the underlying database, while a *Logical Inconsistency* is a permanent mismatch that is due to a misunderstanding of process or misinterpretation of data. While *technical inconsistencies* can be identified by technical means like check tools, *logical inconsistencies* need to be identified by mapping the intended business process to the underlying data structures. They cannot be identified by technical means, as the underlying data is consistent on a one-to-one level.

Looking at a business object instance A.obj in a system A and the respective business object instance B.obj in system B, the following inconsistencies for the business object instance between both environments may occur:

Difference type	System A	System B	Difference type description
1	A.obj	B.obj	Some value in the business object instance of A and B differs
2	A.obj	Respective instance is missing	Business object instance is present in A but missing in B
3	Respective instance is missing	B.obj	Business object instance is present in B but missing in A

Any good consistency check tool should compare current business object instances between the two systems, A and B, display all differences categorized into the three difference types and – if possible – provide drilldown functionality into the exact different fields for difference type 1.

Three different cases are usually distinguished when investigating inconsistencies:

- Inconsistencies between the real world and a system
- Inconsistencies between two systems
- Inconsistencies within one system

As the data needs to be compared between the two environments independently of the nature of the environment, and the investigation needs to be based on tools for easy handling the general procedure to identify the inconsistencies and the data correction will always take place in a computer system (in case 1, the real world will always be the “leading system”). Thus, a unified overall procedure can be applied and we will not distinguish between the three cases unless it is essential for a specific procedure or understanding how to apply a procedure in the given environment.

2.1.2 Typical Root Causes for Inconsistencies

Most inconsistencies are caused by four different root causes:

- Incorrect programming
- Incorrect error handling
- Incorrect manual data entry
- No clear leading system

These different root causes and the distinguishing factors are described in the next sections.

2.1.2.1 Incorrect Programming

Inconsistencies can be caused by programming errors leading to incorrect data between different process steps or applications.

A special case of incorrect programming leading to inconsistencies is neglecting the transactional correctness of programs. In this case, the logical unit of work (LUW) concept is neglected, so that in the worst case only part of a business object is created or changed.

This type of root cause is very common in interfaces. A typical indicator that warrants a detailed analysis is an inbound interface where several business steps are triggered by one incoming document (see IF1 in Figure 2.1). If such an interface is found during the assessment, an SQL-Trace and a code inspection should be performed to check whether commit work statements are triggered between the different steps. If this is the case, and it cannot be changed, the interface should update appropriate status information on which steps have been finished successfully and should be restartable based on this status information. Without these measures, it could happen that one step is finished successfully but this information has not been updated in corresponding documents so that data is created twice or not at all. In the given example in Figure 2.1, the delivery could be updated and goods issue is posted but the sales order is not updated about this fact if a severe error happens between step "Post Goods Issue" and step "Update Sales Order".

SAP Active Global Support provides two services that can assist you with the detailed investigation and improvement proposals for such situations: SAP Interface Management Service for the technical investigation of interfaces or a Customer Program Optimization if a similar situation is encountered outside an interface (for example, by a custom-made report updating several documents).

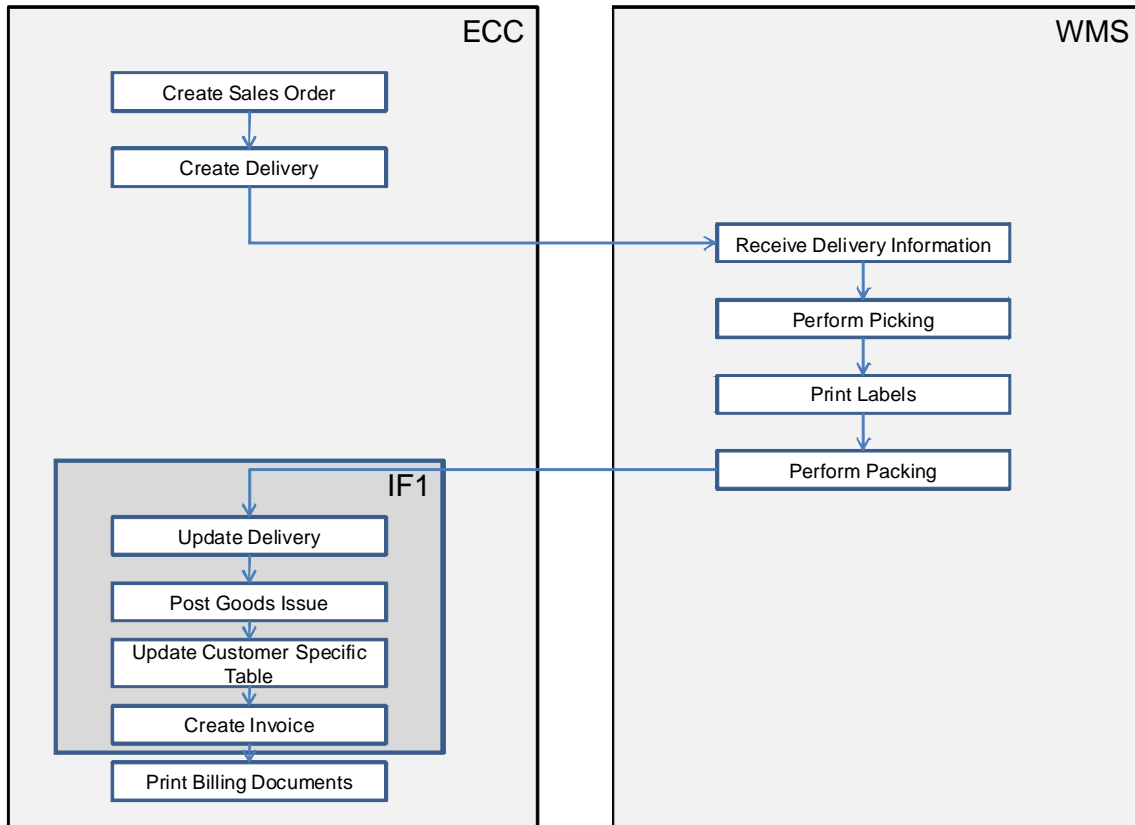


Figure 2.1: Inbound Interface Triggering Several Business Steps

2.1.2.2 Incorrect Error Handling

Sometimes, the program code itself is correct but the operation of the system has led to errors. This situation is most common in a CRM system or for business processes spanning two systems. Typical scenarios for this case are:

- Missing data in one of the two systems caused by BDocs that are in error state and are not reprocessed in a timely fashion.
- Overwriting newer data by incorrect reprocessing of an old IDoc.

Finding a large number of old erroneous interface data, for example IDocs or BDocs, in the system is usually a good indicator for such a case.

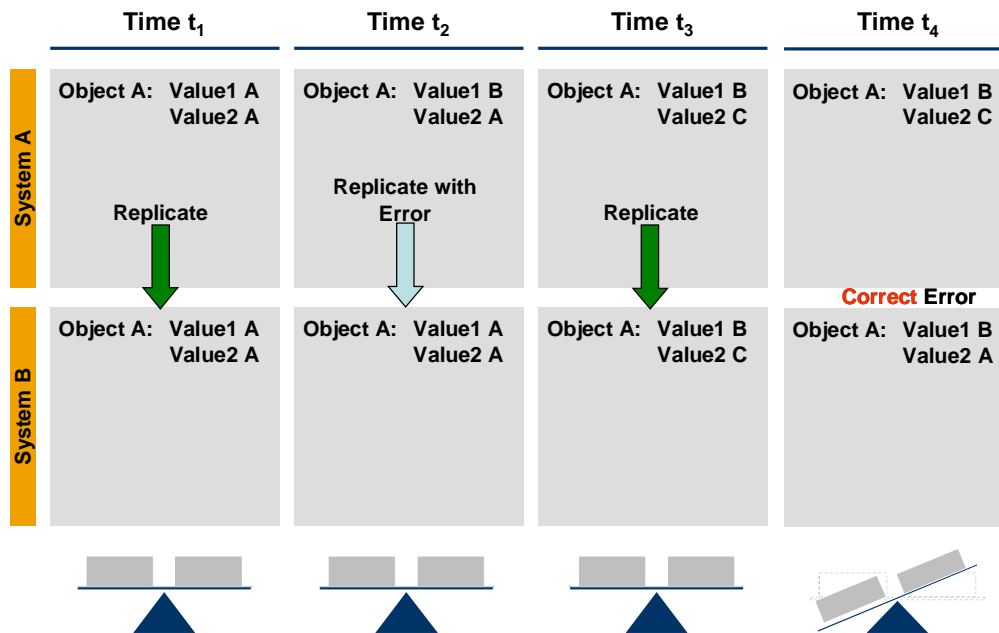


Figure 2.2: Inconsistency Caused by Interface Error Handling

2.1.2.3 Incorrect Manual Data Entry

Very often, the program code is correct but the system has not been used in the intended way, leading either to a manual entry of incorrect data or to not entering important data at all. This is most common when a new system/business process release has been introduced and the end users have been used to the old system for a long time; this results in a discrepancy between the physical world and system. A typical scenario would be the introduction of a new warehouse management system release where the end users post the stock still using old transactions or do not use some of the new transactions. The best way to identify this root cause is to follow the daily business of one of the end users to check whether the system usage corresponds to the designed and intended business process. In parallel, a member of the basis team should continue to investigate the interfaces and error handling especially if two systems are involved. The appropriate counter measure whenever this situation is encountered is to perform another end user training with a special focus on handling exceptions. Typical exception situations to be covered by this end user training should be topics like “How do I proceed if I want to post material to a storage location and the system does not allow this?” and “I receive material in an inbound delivery, which I should not receive according to the delivery advice/has not been maintained in the system – How do I proceed?”

2.1.2.4 No Clear Leading System

In this case, data cannot be changed in one system only, but in two systems independently. This situation creates the risk that outdated data being sent from one leading system overwrites new data already sent from another system. This interchangeability of data in two systems would not pose an issue when only distinct data sets are maintained in both systems (for example, another material type in both systems or sales data in one system and production data in another). If you encounter a situation where the same data can be changed in two systems, you should verify how it is ensured that no updated data (for

example, from MD1) is overwritten by outdated data from the other system during an assessment. This situation is especially common for master data distribution

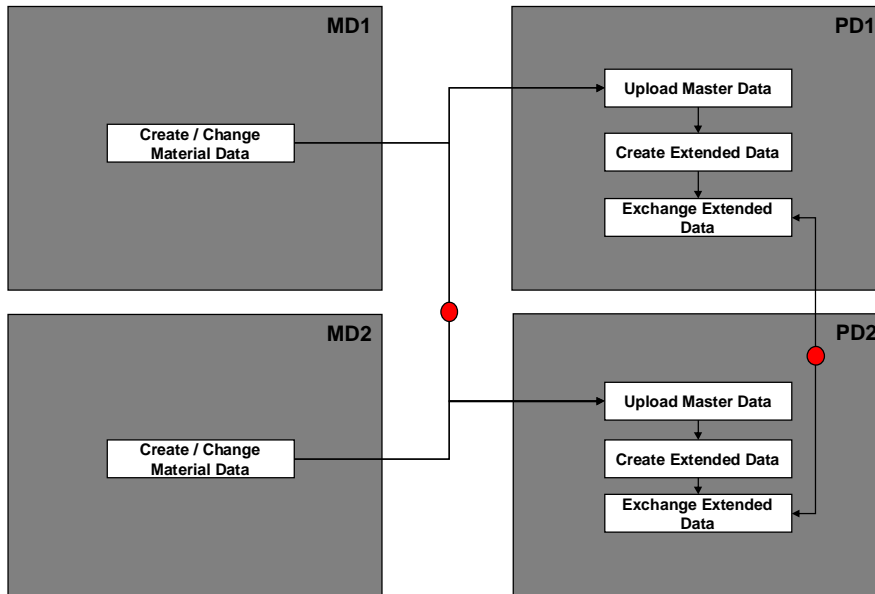


Figure 2.3: Possible Inconsistencies by Unclear System Roles

2.2 Overall Procedure

The goal of any procedure regarding inconsistencies should be two-fold: The procedure should identify the root cause of the inconsistency and identify and correct the inconsistent data including dependent data. In addition, it needs to be decided quickly whether productive use of the system should be continued, disrupted or available workarounds should be employed. Bearing these goals in mind, the procedure outlined below has proven beneficial in the past in providing a starting point and a guideline for a detailed analysis independent of the individual systems and processes involved. Specific tools and detailed data models for common occurrences are described and outlined in section 3. In addition to this procedure, SAP has provided a Best Practice called "Business Continuity Management for SAP System Landscapes" to minimize system impact by setting up and using a business continuity concept.

It is common that an inconsistency is initially reported on a very basic level which does not contain very detailed information regarding the technical data or core business processes involved. A typical example could be a complaint by an end user that orders cannot be delivered as the stock inventory is incorrect. Furthermore, it is usually not known how this inconsistency has been detected. Thus, at this stage it is frequently not determined whether there are any true inconsistencies in the system or whether only temporary differences have been observed. Temporary differences could, for example, be caused by pending IDocs or update tasks and are resolved once all pending system activities are completed while inconsistencies will remain after the activities are finished. Sometimes it is even not known whether the root cause for the described business issue is an inconsistency or difference at all.

Taking into account the missing information at this stage, the first step should be to obtain the missing information to reach a detailed understanding of the involved business processes, monitoring and reporting activities, and technical objects leading to the observed difference. The best procedure to obtain this data would be the assessment of the business processes and the involved data flow. It may not be necessary to understand all core business processes, but only the process involved in the inconsistency detection (which may

not be a core business process) as well as the core business processes which could lead to such a reported inconsistency or will be affected by it. For example, if an inconsistency in stock level information was reported, the process of identifying that data is inconsistent (stock reports, error messages, and so on) as well as the business processes affecting the stock (for example, Sales Order Management (goods issue), Production (goods issue, goods receipt), and so on) have to be understood. While at this stage, not all core business processes of the solution are important, it is essential to know between which business data and technical data the inconsistency has been observed, especially for the inconsistency detection and the business process steps handling the inconsistent data. In addition, a very detailed, technical view needs to be obtained of the data origin and the use of the data by the processes which are in close vicinity to the reported inconsistency, as well as any error detection and handling around these steps.

Example: If a stock inconsistency has been reported, it is essential to know how the inconsistency was detected/measured, how business transactions involving stock (for example, goods movements) are executed, what data is used on a technical level (for example, tables and fields, if several systems are involved: how data is mapped). The goal is to understand the steps leading to the detection of the inconsistencies and to map corresponding steps of data derivation in case different business processes are involved (for example, stock calculation in an external reporting system and in ECC or reporting inconsistencies between accounting in ECC and reporting in BI). An example for such a mapping of a data derivation process can be found in Figure 2.4. The underlying data for an inconsistency being reported between data, that was aggregated using several steps in both systems, could, for example, be an LIS-Table in ECC (S*) for Process 1 corresponding to an InfoCube in BW for Process 2. In such circumstances, it is important to understand which information corresponds between the two systems.

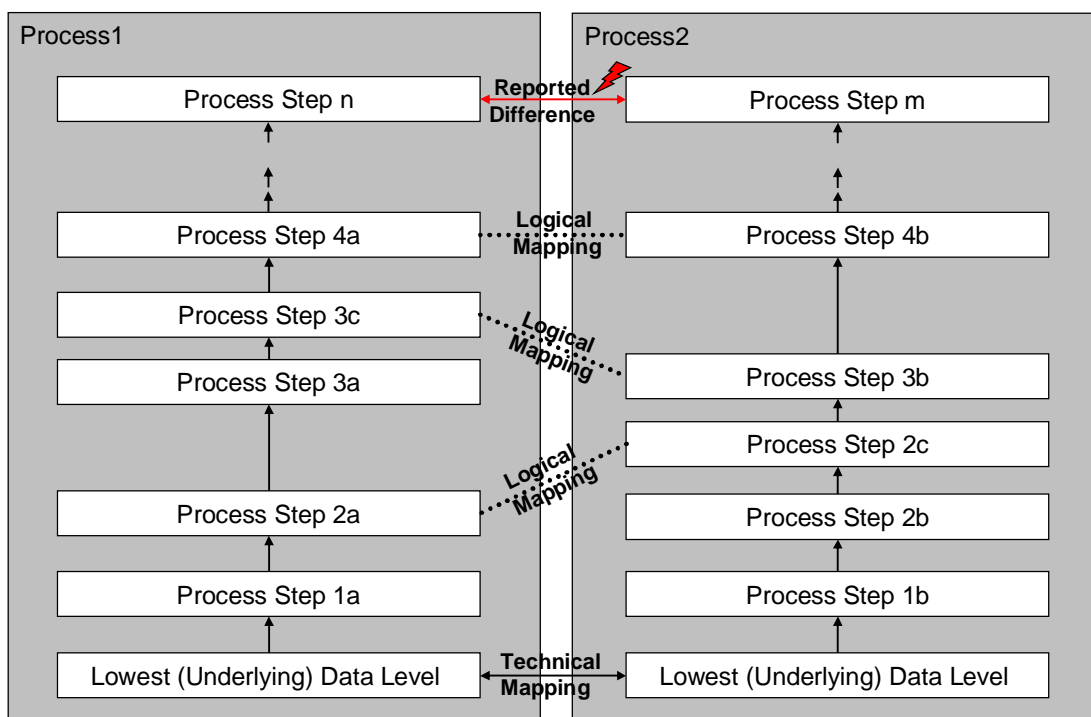


Figure 2.4: Logical Mapping of Steps Between Two Processes



Details regarding the assessment of the processes, including questions to obtain a first working hypotheses and typical pain points can be found in section 2.3: Assessment Guide.



Best Practice would be to document all this data already during the implementation project of the core business processes and to think about possible impacts.

Once the first step of the inconsistency investigation - the understanding of the involved business processes and data derivation processes - has been finished, the root cause of the inconsistency and whether a real inconsistency exists have to be identified. The next step is therefore to check for the common root causes in the steps involved directly with the inconsistent data. The general rule here is not to dig straight into a technical analysis but first to rule out the more operational root causes unless the inconsistency is reproducible.

A common root cause for inconsistencies is application and interface monitoring and error handling (see chapter 2.1.2). This means that investigation of interface monitoring and error handling should be an important part of the assessment, especially if different systems are involved. Once the relevant interfaces and business process steps have been identified by the recording of the involved processes, we need to determine how these are monitored (application monitoring and interface monitoring) and whether erroneous data is contained in the system. Besides additional, detailed questions described in the next section in table 5, the system should be checked for old erroneous data using available standard transactions (for example, within an SAP system ST22 for short dumps, WE02 for IDocs in error state, and so on). The exact transactions available to check for this data depend on the interface technique, the business process steps, and the system itself. If data in error state is found, the handling of such incorrect data should be discussed within your monitoring team. If incorrect handling has been identified (for example, reprocessing of BDocs or Updates without checking that this data is still valid), appropriate error handling procedures need to be derived.

If the reported differences cannot be explained by error handling procedures and are not temporary differences, the logical mapping of corresponding steps including an understanding of the underlying technical data as reached during the assessment should now be used to verify the data consistency on the lowest technical level available (in our example: comparison of S-table versus InfoCube). The investigation needs to be started on the lowest level as higher level, derived data will always show inconsistencies if the lower level data is already corrupted, even if the involved intermediate processing steps are correct. Check reports are needed to compare this data. It is important that temporary differences are filtered out during the use of check reports by using a time window where no system activities are executed affecting the involved data. Filtering out differences means that all update tasks or interface processing for the relevant data should be finished and that no new data is created during the time of the analysis. If such a time window is not available, the comparison has to be repeated and results need to be compared between the different runs. Temporary differences will disappear between different runs of the check reports/tools while real inconsistencies will remain in the result for all runs of the same report. An example would be to compare the information between the S-Table and the InfoCube at a time where both systems are in sync and no additional data is written to the S-table during comparison. If this is not taken into account, the S-table may contain newer data missing in the InfoCube which would be corrected with the next extraction.

Sometimes, the situation arises that inconsistencies exist in very old data. The old data should be corrected before attempting to identify the root cause in these cases by reloading or correction reports whenever possible as old inconsistencies whose root cause has probably been fixed already may obscure the root cause for new inconsistencies or whether inconsistencies exist at all.



Possible reports and methods to correct the outdated data can be found in section 3: Tools and Procedures within an SAP Environment.

The business process steps leading to the low level data need to be traced and debugged when temporary differences have been filtered out and inconsistencies have already been identified in the low level data, as a program error is most likely, for example, the Extractor

Best Practice: Data Consistency Check

filling the InfoCube and the update routines for the S-table need to be investigated in our given example of S-table versus InfoCube. Programming errors leading to a mismatch between both data sets could be purely technical (for example, an incorrect sign in some formula), or logically, if the data calculation rules are incorrect (for example, using the document date instead of a posting date). If temporary differences are ruled out and no inconsistencies are found in the underlying data, the dependent derived data needs to be investigated from bottom level to top level. It is important that only data mapped logically is compared at this stage and no intermediate steps are taken into account. Using the example for a logical mapping displayed in Figure 2.4 once the processes involved in the technical mapping of the underlying data has been ruled out as root cause, steps 2a-2c, 3c-3b, and 4a-4b should be compared. If step 2a would be compared to step 2b, just following the order of steps without taking into account whether a logical connection exists between the two different data sets, the result of the consistency verification between process1 and process2 would be meaningless at this stage as data would be compared that is not comparable by definition.

The process chain should be investigated from bottom to top using appropriate means like check reports or queries until the first real occurrence of inconsistencies has been identified. Once an inconsistency between two logical connected steps has been found, the technical process steps involved between these steps need to be debugged and investigated to identify the technical root cause like, transactional incorrect interfaces, technical programming error, or logical error.

The root cause for the inconsistency needs to be corrected once it has been identified. The appropriate measure to do so depends on the nature of the root cause. If a coding error was identified as the root cause, the coding needs to be corrected. On the other hand, if the issue was caused by the operation of the solution, appropriate measures, for example, would be to train your Operations team (for example, using a System Administration workshop) and to adopt your operation procedures.



Details regarding typical root causes, the distinguishing factors between these, and appropriate solutions can be found in section 2.1.2: Typical Root Causes for Inconsistencies.

How dependent data is affected needs to be investigated after the root cause has been identified and corrected. This could be either consolidated reporting data or follow-up errors like incorrectly created documents. The business processes using the original data identified as inconsistent have to be followed to understand the impact of an inconsistency in one business object on depending business objects. If follow-up documents are created (for example, creation of controlling or financial data from logistics data), these need to be corrected as well using appropriate measures.

To recover or correct lost and incorrect data sets, several possibilities exist:

- Restore into a parallel system and reload of data from the parallel system by RFC, LSMW and so on
- Reload of data from a leading system
- Use correction tools to recover data by relationships to redundant stored data
- Manual correction
- Complete restore of a backup / point-in-time recovery



The last two methods should only be used as a last resort.



Very often, a combination of data recovery methods and tools is required, for example, individual incorrect sales documents could be corrected manually on the data base and dependent data could be corrected afterwards by correction reports. Each of the different methods has certain advantages and disadvantages leading to

different use cases.



You can find a more detailed overview about these methods in Appendix 4.3: Advantages and Disadvantages of Different Recovery Methods.

Important questions that influence the choice of recovery method and therefore need to be discussed are:

- Does dependent data exist for the inconsistent/lost data?
- Could these be used for a reconstruction of data?
- How many Business Objects and how many instances are affected?
- What quantity/complexity of objects is affected?
- Is a backup available for a point-in-time recovery?
- How much time would the different methods require?

2.3 Assessment Guide

An inconsistency reported does not necessarily have to be based on incorrect system behavior or need to be an inconsistency at all as we have seen in the proceeding section. The assessment given in this chapter should provide a general guideline to obtain a first working hypothesis on what the root cause could be and a very detailed, clear understanding of the involved business process steps and technical data which form the basis for any later steps during root cause analysis and possible correction.

2.3.1 Understanding the Situation and Inconsistency

The starting point is always that some inconsistency is reported but the root cause as well as the exact impact of the inconsistency and how it has been observed may not be known. To guide the decision whether a business usage of the affected system in the current situation is possible or not, the possible impact of a continuation versus a stop of system usage needs to be gauged carefully based on the business impact in both cases.

The first step of the assessment should be to obtain a general understanding of the business impact and which processes and data are affected (if this has not been done in advance as described in the Best Practice document “Business Continuity Management for SAP System Landscapes”) as well as whether an inconsistency really exists.

Questions	Answers
Which business objects are affected?	
Describe the differences and your expectations. Background: Sometimes a difference may be reported due to misinterpretation of the data's meaning.	
Which business processes are affected?	
Which business process steps are affected?	
Describe the business impact. Background: The business impact is an important factor to decide whether you should continue working in the system until the differences are resolved.	

Table 2.1: Questions to Understand the Business Impact and Severity

The results of these questions determine the business processes and the business process steps which have to be investigated in more detail. Furthermore, the determined impact is one of the most important basis factors how business operation should proceed.

Once the severity of the inconsistency has been established and it is known which business process are affected, the next step should be to investigate the inconsistency detection

Best Practice: Data Consistency Check

process and the inconsistency in more detail. Questions to help with this task are collected in the following table.

<i>Question</i>	<i>Answer</i>
How did you notice and identify the inconsistent behavior?	
If observed by consistency check tools: Which transactions/reports did you use?	
Is the difference reproducible or did it occur only once?	
If it is reproducible: Are the same inconsistencies reported? How did you see that they are the same?	
Did you observe any rules between inconsistencies? Background: If rules are observed, they provide important information about the nature and possible root cause of the inconsistency. If it is observed at certain times only, it could indicate differences due to performance bottlenecks; if it is only observed for certain subsets of one business objects (for example, only certain storage locations), this could indicate a mapping error, and so on.	
Do the inconsistencies disappear and appear again after some time?	

Table 2.2: Questions for a First Understanding of the Inconsistency

The results of these questions establish whether we have an inconsistency or only a temporary difference between two data sets. Any rules/connections identified between the observed inconsistencies need to be considered in the detailed business process root cause investigations. The next questions will collect more details on the process and provide the technical data background to verify the correctness of the data mapping and inconsistency checking process. At this stage, data is collected to function as starting point for the technical business process and inconsistency investigation as well as for a later correction.

<i>Question</i>	<i>Answer</i>
What are the input parameters and what did you expect?	
If observed by a custom made report/transaction: How does the report/transaction work technically?	
Which tables and table fields are compared?	
If between different systems: How is the data mapped? Background: This information is used to verify that the data is compared correctly. Sometimes mapping of data is done incorrectly, for example, by mapping quantity in bin to quantity in storage	

location or mapping document date to posting date. You should evaluate the meaning of table names and fields as well as a description of the data he wants to use for the SAP systems.	
--	--

Table 2.3: Detailed Questions to Improve the Understanding of the Inconsistency

The description of data mapping needs to be used for new check tools or may be entered directly into the generic report given in section 3.4.3 if two linked tables in one system are involved.

2.3.2 Process Understanding

Sometimes an error in the program/interface logic or a mismatch between process design and system usage leads to inconsistencies, especially when inconsistencies between different systems or between the system and the physical world are observed. Thus, the understanding of the design of the core business processes which could lead to the observed inconsistency forms an essential part of finding the root cause, as you have to map the process design to the system's use in the real physical world. A detailed description of the end-to-end process across all systems and applications has to be recorded during this business process analysis if not documented in your company already. It is important to get an overview of the business in general and why the chosen process has distinct priority.

For each process and process step the following information must be known:

- Has a process owner been defined for this process?
- Are there any other known critical problems/issues?
- Short description of the process step
- Important technical objects used in the process step (transactions, programs, tables)
- Online/background execution of transactions

In addition to these general questions, the following questions which are especially important to inconsistencies have to be answered. They are used to cross check whether an operational issue (error handling, monitoring) or process design issue could cause the detected inconsistency.

<i>Question</i>	<i>Answer</i>
Has a new system or process been introduced recently?	
Did the end user training contain instructions how to handle exceptions?	
If yes: What exceptions in this area have been included?	
Have the users been used to another system/transaction/report for similar tasks in the past?	
What testing activities (user acceptance / integration / volume) have been performed?	

Table 2.4: Questions to Assess the System Usage

The next block of questions should be used if more than one system is involved. They are intended to determine whether the inconsistency could be caused by several leading systems, monitoring and error handling in the area of interface monitoring, or whether an investigation toward transactional correctness should be started.

Question	Answer
Is more than one system involved in the business process steps?	
<p>If yes: What are the roles of the involved systems? Which one is the leading system?</p> <p>Background: Sometimes no clear leading system is defined. In these cases, a high effort is needed to keep data consistent between the systems as changes in one system could be overwritten by changes in another system. Common cases are master data development systems.</p>	
Describe the relevant business process steps affected by the observed inconsistency	
Are interfaces to other systems involved in these steps or in prior steps regarding the used business data?	
<p>If yes, how do you perform reconciliation between the involved systems?</p> <p>Background: Sometimes the temporal order of reconciliation reports does not fit the timing of the business process steps. For example, when running an MRP in an external legacy planning system, the current requirements and stock need to be synchronized with the SAP ERP backend.</p>	
What interface technology do you use between the two systems?	
Is custom-made coding used?	
Do you trigger more than one step by one call of the interface?	
What measures have been taken to ensure that the interfaces are transactionally correct?	
<p>Please describe your monitoring and error handling concept?</p> <p>Background: Especially if more than one step is triggered by an interface using custom code, the transactional correctness could be compromised. Even when the transactional correctness is ensured, incomplete monitoring and error handling could lead to data loss. Using the example of running MRP in an external legacy planning system, unobserved errors in the transfer of requirements and stock will lead to incorrect planning results due to inconsistencies between the systems. Hint: If one of the first two questions is</p>	

Best Practice: Data Consistency Check

answered yes, you should involve an interface or ABAP expert skilled with transactional correctness investigations in the next steps or order an Interface Management Service.	
--	--

Table 2.5: Questions for Processes Involving Several Systems

3. Tools and Procedures within an SAP Environment

Tools to find and correct inconsistencies within an SAP environment will be provided within this section. These include standard tools as well as commonly used tools that are provided in SAP notes only and must be implemented in customer namespace. Only the most important ones will be described in this section.

3.1 Tools to Check Consistency Between ECC/CRM and Within CRM

An SAP CRM System usually consists of more than one logical database. Every SAP CRM System has a CRM database. In most cases, data exchange with one or more R/3 back-end systems is necessary. The consolidated database (CDB) is the basis for data exchange with Mobile Clients. It is very important to keep these databases synchronized and consistent, which is the task of the SAP CRM Middleware. However, inconsistent data may occur if:

- The Delta Load was inactive for some time. For more information about Delta Load refer to Activating Object Classes for Delta Synchronization in the CRM online documentation.
- The Initial Load could not be carried out completely. For more information about the Initial Load, refer to Starting the Initial Data Transfer in the CRM online documentation.
- Data was accepted in the CRM System but rejected in an R/3 back-end.
- Filter conditions were changed for the load from an R/3 back-end. For more information about filter conditions, refer to Defining Filters for Objects in the CRM online documentation.

To analyze and repair inconsistencies between SAP CRM and SAP R/3, SAP created the tool DIMa (Data Integrity Manager). The tool is used to analyze and correct inconsistencies resulting from the exchange of the most important business objects between R/3 and SAP CRM. The features of DIMa and covered business objects are described in the next section. Furthermore, SAP has created various analysis and correction reports for the SAP CRM and R/3 exchange scenario and also for the SAP CRM Mobile scenario. The most important of these reports are described later in this document.

3.1.1 Data Exchange for the Most Common CRM Scenario - CRM Sales

The most commonly used CRM scenario is CRM Sales. In this scenario, sales orders and other objects are exchanged between SAP CRM and R/3. Inconsistencies of business objects may occur due to errors in the exchange process. The DIMa is a good tool to uncover inconsistencies between business objects in R/3 and SAP CRM. The data exchange toolbox (note 718322) provides various tools which are described later to analyze these inconsistencies in more detail.



Be aware that inconsistencies can be temporary. Business objects are exchanged asynchronously in queued RFCs of the middleware. If you detect a difference for a business object, the changes that rectify the difference might already be in the queues but not yet processed by R/3 or CRM. On the next run of DIMa, this temporary difference would not occur anymore as the changes would be processed from the queues. Persisting differences after several runs of DIMa can be considered as real inconsistencies. Usually, if you detect errors (inconsistencies in DIMa) in the exchange, for instance, of a sales document created in SAP CRM and distributed to R/3, you often run into the error that R/3 rejects the created sales document. The document exists in CRM but does not exist in R/3. A correction, for instance, customizing in R/3 has to be performed to enable successful reprocessing of the rejected sales order. (The order can be resent by an action in the CRMD_ORDER transaction in SAP CRM).

Also, errors in middleware like items deleted in queues can result in inconsistencies between CRM and R/3. To deal with such inconsistencies proactively, interface monitoring is advisable.

As CRM often uses filters between R/3 and CRM, special care must be taken, as changing these filter conditions can result in inconsistencies. For instance, if the condition on R/3 side is widened, business objects may be sent as delta messages which do not have a corresponding object on CRM side. If, for instance, filter conditions are widened on the R/3 side, an initial load of the respective business objects must be performed.

3.1.2 Data Exchange for the CRM Scenario CRM Mobile

In the CRM scenario CRM Mobile, laptop users (mobile clients) exchange data (business objects) with a central CRM system via a complex middleware. In this scenario, the CRM system has a special database for mobile client data processing: the CDB (consolidated database). Inconsistencies can occur between the CDB and the central database of the CRM system as well as between the CDB and the mobile clients. The DIMa can be used to analyze inconsistencies between the central database of the CRM system and the CDB.



Be aware that inconsistencies in a CRM environment can be temporary. Business objects are exchanged asynchronously in queued RFCs of the middleware. If you detect a difference for a business object, the changes that rectify the difference might already be in the queues but not yet processed by CRM. On the next run of DIMa, this temporary difference would not occur anymore as the changes would be processed from the queues. Only persisting differences after several runs of DIMa can be considered as real inconsistencies.

Two common cases of inconsistencies exist between the mobile clients and the CRM server.

1. Data is missing on mobile clients but is present in the CDB. In this case, the administrator corrects the inconsistency by performing an extract of the business object in question using transaction SMOEAC for the respective mobile client.
2. A more serious inconsistency is that data changed on the client is not processed properly by the CRM server because the server rejects the change. These inconsistencies are only detected in rare cases as the respective BDocs have the status finished (F02). To detect such inconsistencies, the system administrators should monitor for BDocs with state F02 in SMW01/SMW02 transaction. As well on the mobile clients, rejected objects can be found by navigating to the CRM inbox tileset. If a rejected change is detected, it can be further analyzed why the rejection occurred. Then, either correction at CRM server level or corrections on the mobile client have to be performed in order to reprocess the BDoc successfully.

As the outbound queues are sometimes huge, especially after rollout of mobile clients or mass updates triggered in R/3, outbound queues are sometimes deleted which results in inconsistencies. To correct these inconsistencies, an extract of the business objects for the respective clients has to be performed.

3.1.3 DIMa – The Tool to Detect and Repair Inconsistencies Between SAP ECC and SAP CRM

The Data Integrity Manager (DIMa) helps to detect and repair inconsistencies between databases in the SAP CRM system, the R/3 backend and SAP CRM's mobile part. It supports consistency checks for the most commonly used business objects: business partners, products (materials), sales documents and pricing information. The DIMa uses the SAP CRM middleware and the CRM R/3 plug-in to carry out the extraction and comparison of business objects. DIMa also provides the possibility to correct inconsistencies in addition to detecting inconsistencies.

3.1.3.1 Features of DIMa

The DIMa compares data in different databases and displays inconsistencies. The data comparisons can be carried out between:

- The CRM database and an R/3 back-end database

Best Practice: Data Consistency Check

- The CRM database and the CDB database.

For many objects, it is not only possible to compare but also to synchronize the data via DIMa. There are two compare types available in the DIMa:

- Header Compare:
A header compare checks, whether an object instance exists in both databases.
- Detail Compare:
A detail compare compares all data of an object instance found in both databases.

Some objects may not allow a header compare. The detail compare is then carried out.

3.1.3.2 Consistency Check Scope of DIMa

DIMa supports consistency checks for the following business objects:

DIMa check object	Business object	Starting from CRM Rel., SP
ACT_DIMA	DIMa Object for Activity	4.0*
BP_DIMA	DIMa Object for BP	4.0*
CAM_DIMA	DIMa Object for Campaign	
CDB_CAPGEN	Bus. partner CRM ↔ CDB	5.0
CDB_CONGEN	Contact pers. CRM ↔ CDB	5.0
CONTACTS	R/3 Customer Master: Contacts	3.0
CP_DIMA	DIMa Object for Contact Person	4.0*
CUSTOMER	R/3 Customer Master: Customer	3.0
CONDITION TABLES	Condition tables R/3 ↔ CRM	4.0
EMP_DIMA	Compare Employee	4.0*
MATERIAL	Material: ERP => CRM	4.0; detailed 5.0
OPP_DIMA	DIMa Object for Opportunity	
PARTNER	R/3 BP Master: Partner	3.0 SP14; 3.1 SP04; 4.0; detailed 5.0
PRODUCT_MAT	Material: CRM => CDB	3.0; detailed 4.0
PRODUCT_SRV	Service: CRM => CDB	3.0; detailed 4.0
REBATE_DNL_AGR	Agreements R/3 ↔ CRM	5.0
RELATIONS	R/3 BP Master: Relations	3.0 SP14; 3.1 SP04; 4.0; detailed 5.0
SAD_DIMA	Salesdocument CRM -> CDB	
SALESDOCUMENT	Documents R/3 -> CRM	3.0 SP14; 3.1 SP04; 4.0
SAMPLE_CDB	Sample for CDB ↔ CRM	
SAMPLE_R3	Sample for R/3 ↔ CRM	
SERVICE_MASTER	Service: ERP => CRM	3.0; detailed 4.0

Table 3.1: DIMa Check Objects

For further details about the use of DIMa, refer to the DIMa documentation and the online help.



You should execute DIMa for the most important business objects on a regular basis. The monitoring object "DCCRMDIM" may be used to monitor these regular runs within the SAP Solution Manager.

3.1.4 Analysis and Correction Reports in Data Exchange of One Order Documents Between SAP CRM and R/3

SAP created a data exchange toolbox report to investigate the exchange of one order documents in the CRM middleware. This toolbox is provided by composite SAP note 718322 which includes the most important analysis and correction reports regarding data exchange of one order objects between CRM, R/3 and CRM Mobile.

Best Practice: Data Consistency Check

An often used analysis report is CRM_ODE_CHECK_CFG (SAP note 718797). It checks the configuration of the data exchange between SAP CRM and SAP ERP/R/3, both on the CRM and the R/3 side. Information like the RFC destinations for the R/3 system are displayed as well as the queue prefixes of the queues relevant for the data exchange between CRM and R/3 on the CRM side. The RFC connections in the opposite direction are displayed on the R/3 side, as well as various information about the configuration, like the queues involved in the exchange.

A very important report in root cause analysis of issues with one order objects is CRM_ORDER_READ. This report displays on the database level all the information linked to a CRM one order object. Often, this report is the initial starting point in analyzing the root cause for problems with one order objects. A one order object saved in a CRM system with errors is not replicated to R/3. Sometimes, users want to have one order objects replicated although errors exist. The report CRM_MESSAGES_DELETE deals with this issue (SAP note 719224). The errors are deleted by the report from the application log for the one order objects in question to enable the replication into R/3. The affected orders are replicated by saving the one order objects in the report.

The central SAP note for all the reports in the data exchange toolbox is 718322. A list of all the reports and a short description follows:

SAP Note	Short Description	Report
718751	Read Tables Business Transaction	CRM_ORDER_READ
Root cause analysis: Display data structure of a business transaction in detail		
718797	Check Basis Configuration Data Exchange	CRM_ODE_CHECK_CFG
Configuration of data exchange between R/3 and CRM (RFC destinations, and so on)		
718916	Set Message Log Level	CRM_MESSAGES
Set Log Level in CRM for your terminal session / Set breakpoints for certain messages produced		
718918	Display BDocs for a Business Transaction	CRM_DATAEXCHANGE_10_BDOCS
Display m/sBDocs for an business document (sales order, activity, and so on) in CRM		
719032	Compare a Document in CRM and R/3	CRM_ODE_CHECK_DOC
The report checks whether a transaction is available in R/3 and CRM if both if they are unequal.		
718929	Incorrect Documents that were created in R/3	CRM_DATAEXCHANGE_CHECK_OLTP
Troubleshooting for documents that were exchanged between R/3 and CRM but resulted in an error on CRM side		

Table 3.2: Analysis Reports

SAP Note	Short Description	Report
648621	Checks after initial data transfer if incorrect documents exist in CRM	ZZ_DOWNLOAD_REPAIR_ERRORS
Checks after the initial data transfer, if incorrect documents exist in the CRM server, deletes them in the CRM server and retrieves them from the R/3 system again.		
686063	Check after initial data transfer, if all documents have arrived in CRM	CRM_INIT_DNL_MERGE_ORDERS
Checks after the initial data transfer, whether all documents have arrived in the CRM server and retrieves possibly missing documents from the R/3 system.		
717478	Deleting a CRM OneOrder Document	CRM_ORDER_DELETE
Deletion of one order objects in CRM. Subsequent deletion in CRM Mobile, R/3 can be omitted.		
719109	Creating Missing Replication Records	CRM_CREATE_REPL
When creating one order objects in CRM, they are uploaded to R/3 and return as replication records for the one order objects. Sometimes, these replication records get lost. To create these confirmation records use this report.		

719111	Setting Completed Status for a Sales Document	CRM_SET_STATUS_COMPLETED
Set the status of one order objects exchanged between R/3 and CRM in CRM to 'completed'. Not completed may be caused in CRM for various reasons.		
719113	Delete Links to PRCD_HEAD	CRM_LINK_OBJTYPE_18_DELETE
CRM 3.0/3.1: Wrong pricing documents due to program error for CRM order documents. Correction report deletes wrong pricing documents. (see also 579336/599970)		
719224	Delete Messages from Error Log	CRM_MESSAGES_DELETE
Error messages in application log are deleted for a certain one order document.		
719225	Delete Unnecessary Replication Records	CRM_REPL_CHECK_AND_DELETE
Replication records exist in the CRM server for which no more CRM documents exist. This report is used to delete these orphaned replication objects. See also note 719109.		
719227	Delete Orphaned CDB Entries	CRM_ORDER_DELETE
You used report CRM_ORDER_DELETE to delete documents in the CRM server and selected the 'Do not send BDoc type' flag. As a result, the CDB entries were not deleted as well. (See 717478)		
719229	Contracts from R/3: Adapt Doc. Flow	CRM_CONTRACT_UPDATE_R3
Document flow records and release values for contracts which were loaded from the R/3 system to the CRM server are incorrect. See also note 736044.		
719960	Resend CRM Documents	CRM_START_REQUEST_FROM_CRM
During the data exchange between the CRM server and the R/3 system, documents can only be transferred to the R/3 system by means of a change in the CRM server. A request download from the CRM server to the R/3 system does not exist.		
721107	Compare delivery - and Bill. Status between R/3 and CRM	CRM_CRM_R3_CHECK_STATUS
You can use the CRM_R3_CHECK_STATUS correction report to read the delivery and billing status from the R/3 system and to adjust the status correspondingly in the CRM system.		

Table 3.3: Correction Reports

3.1.5 Miscellaneous Check, Correction and Repair Reports in CRM

3.1.5.1 Inconsistency Check and Correction Reports

An important correction report is CRM_ORGMAN_DOWNLOAD. Every time you change the organization model in CRM online, you have to run this report to update the organizational changes in CRM Mobile (CDB). Please refer to the documentation in the customizing transaction SPRO CRM -> Master Data -> Organizational Management -> Data transfer -> Copy Organizational Structure to CRM Mobile.

When applying SAP notes for index access optimization, often new fields are introduced in the CRM_ORDER_INDEX table for instance. After introducing such fields, the CRM_ORDER_INDEX is inconsistent as the new fields do not hold the necessary data. To repair the inconsistency, run CRM_INDEX_REBUILD to rebuild the whole table or for instance report CRM_INDEX_FAST_TEMPL_TYPE_UPD to populate only the necessary new fields.

For the CRM Mobile scenarios, certain temporary inconsistencies between mobile client and the CRM server can be avoided by setting specific entries in the table SMOHNOCSTA on the CRM server. The temporary inconsistencies can occur if you perform the following:

First, you make changes to a business object on the mobile client and you upload them to the CRM server by ConnTrans. Secondly, you proceed after the ConnTrans to change the business object again. After performing a second ConnTrans, the changes you just made are reset by the server to the state after the first ConnTrans. These changes are updated correctly on the client to the state of your last change after a next or later ConnTrans run.

To avoid these temporary inconsistencies, changes in the SMOHNOCSTA table on the CRM server are necessary, for further details refer to SAP note 845693.

Best Practice: Data Consistency Check

As well for the CRM Mobile scenario, internal inconsistencies in look up tables or extract queues for intelligent and interlinkage objects can be corrected by using the report described in SAP note 622693. The corresponding correction report is Z_30/40_EXTRACT_LUTAB_EXTRACTED_F. Refer to the SAP note 622693 for further details.

3.1.5.2 Root Cause Analysis Reports

A report for analyzing the Mobile Client distribution model is ZTN_RR_LU_ANALYSIS. This report is very useful for analyzing the distribution model of the mobile clients to uncover incorrect configuration in the distribution model. The most important functionality is the possibility to analyze the distribution model of intelligent objects. Refer to SAP note 835149 for further documentation.

3.2 Tools to Check Consistency Between ECC/SCM and Within SCM

3.2.1 Introduction

The term *Internal consistency* refers to the consistency between the liveCache data and the APO database while the term *External consistency* refers to the consistency between the SCM system and one or more dedicated R/3 Systems. Please make sure that the internal consistency of the APO system is guaranteed before you check the external consistency and set it up again.

All check reports provided by official SAP notes in the area of SCM can be executed by the system administrator or scheduled or manually by the end-user in update mode.

3.2.2 Product Allocation (PAL): Internal Inconsistencies Within SCM or Between ECC/SCM

The functionality of SAP SCM PAL is very flexible to use, however, expectations to its functionality are usually very specific and sometimes differ from the standard design. Several user-exits are available to adapt the functionality to specific needs.

SAP did a fine-tuning of the accuracy of product allocation in 2004 and several notes were created. At the least, all those notes from 2004 should be applied when using product allocation. Specific newer notes should be applied to each R/3 and SCM system when using product allocation in addition to these notes.

Generally, the probability that frequent PAL inconsistencies in a system are due to a customer modification or user-exit is very high when the above mentioned notes are applied.

3.2.2.1 Correction Report /SAPAPO/RMQUOT_USAGE_CHECK

Correction report /SAPAPO/RMQUOT_USAGE_CHECK (corresponding transaction is /SAPAPO/ATPQ_CHKUSG) checks the consistency and corrects product allocation (PAL) assignment within SCM. You can find this report in the SAP Menu: *'Advanced Planning and Optimization -> Global ATP -> Environment -> Product Allocations -> Repairs'*. See SAP note 676128 for further information on this report.

3.2.2.1.1 Check Frequency and Scheduling

Run this report at least weekly during low system activities affecting product allocation, or if needed, once a day. Note that this report has to be started online and requires manual interaction to correct inconsistencies. The report is often scheduled in the background, to evaluate the spool lists only. In case of errors, it is started manually.

It is possible to run this program in the background with an automated update of the checked items by the following workaround: Run the Batch Input for the program /SAPAPO/RMQUOT_USAGE_CHECK using SAP Menu: *System -> Services -> Batch Input -> Recorder*. Perform all the steps and then schedule it.

3.2.2.2 Correction Report /SAPAPO/SDRQCR21 with Product Allocation

/SAPAPO/SDRQCR21 can compare and correct the product allocation assignments across R/3 and SCM (tables: QTVB in R/3 and table /SAPAPO/SDQTVB in SCM), if the flags and fields for product allocation are used.

Table QTVB can be interpreted as a backup in case PAL data is lost in SCM. Therefore, this comparison is usually used as a data recovery from ERP, if data in SCM was lost.



Note that this report does a neutral PAL check and an over confirmation might occur that should be taken care of by a BOP run as a second step. For further information, see the F1 help in the report /SAPAPO/SDRQCR21 for the PAL checks.

3.2.2.3 Typical Examples of PAL Inconsistencies

There are some typical occurrences of product allocation inconsistencies:

- Incoming order quantity is greater/smaller than the product allocation assignments in a period
- Negative values with product allocations (which makes no sense at all):
- Negative order incoming quantity
- Negative product allocation assignments

Example:

The Result list of the product allocation check report in SCM found in this case a negative product allocation assignment:

Product Allocation Group	Planning object no.	Check Date	P FY	ConfigChar	Png area
Valid from Valid to	Prod. alloc. qty Incoming Orders Qty	Confirmed Quantity			Shortage Blin
6T_CHIER_SOLDTO_MTH_BACK_CONS	HDkopCsH0Du0000eDM960	LFDAT	M		
6T_CHIER_SOLDTO_MTH_BACK_CONS	HDkopSsH0Du0000eDM960	LFDAT	M		
6T_CHIER_SOLDTO_MTH_BACK_CONS	HDkopysH0Du0000eDM960	LFDAT	M		
6T_CHIER_SOLDTO_MTH_BACK_CONS	HDkopCsH0Du0000eDM960	LFDAT	M		
6T_CHIER_SOLDTO_MTH_BACK_CONS	HDkopSsH0Du0000eDM960	LFDAT	M		
01.09.2006 30.09.2006	25	21	2-		23 CS

Figure 3.1: Result List for Product Allocations

Such negative PAL confirmations can also be found in table /SAPAPO/SDQTVB:

FLUID	TSTAMP	POSGUID_CMP	TSTAMP_VAR	TSTAMP_VAR_TYPE	MENGE	BASIS
A52000eDM960	HDkopSsH0Du0000eDM960	20.060.901.000.000	20.060.901.000.000	M	20,000	CS
800000eDM960	HDkopSsH0Du0000eDM960	20.060.901.000.000	20.060.901.000.000	M	2,000	CS
EC0000eDM960	HDkopSsH0Du0000eDM960	20.060.901.000.000	20.060.901.000.000	M	14,000	CS
7y0000eDM960	HDkopSsH0Du0000eDM960	20.060.901.000.000	20.060.901.000.000	M	7,000	CS
9a2000eDM960	HDkopSsH0Du0000eDM960	20.060.901.000.000	20.060.901.000.000	M	1,000	CS

Figure 3.2: Corresponding Table View

PAL errors with “negative confirmations” do not correspond to any economical concept and are meaningless. Possible root causes for such situations are:

- Missing SAP notes fixing a bug
- User-exits
- Modifications

3.2.2.4 Overview of Notes for PAL Consistency:

Component	Sap Note	Description
SCM-APO-ATP-BF-PAL	676128	Product allocations: Control of Product allocation assignment
SCM-APO-ATP-BF-PAL	*	All *PAL notes from year 2004: note search with ERP AND SCM patchlevel for "product allocation"

Table 3.4: Notes for PAL Consistency

3.2.3 Time Series of DP and SNP

3.2.3.1 Reports to Check Consistency within this Area

/SAPAPO/TS_LCM_CONS_CHECK performs a consistency check for time series of DP and SNP for a selected planning area. See SAP note 425825 for further information on this report.

/SAPAPO/TS_LCM_CONS_CHECK_ALL does the consistency check for all time series but can only display the results, without correcting errors.

3.2.3.1.1 Check Frequency

It is recommended to check the consistency of time series once a day.

3.2.4 Shipping and Vehicle Scheduling

/SAPAPO/VS_CONS_CHECK detects incorrect customizing settings for the shipment creation process. This includes inconsistencies between SCM and R/3 settings (integration customizing), optimization profiles, inconsistencies within the different database tables and SAP APO master data that is relevant for TP/VS, for example vehicle resources, transportation lanes, and locations.

If inconsistencies are found, you can correct such errors by forward navigation. See SAP note 425825 for further information.

3.2.4.1 Check Frequency

Run the report /SAPAPO/VS_CONS_CHECK once a day if needed.

3.2.4.2 Important Notes for TP/VS Consistency:

Component	SAPNote	Description
SCM-TEC	425825	Consistency checks, ...

Table 3.5: Important Notes for TP/VS Consistency

3.2.5 Integration Models (External Consistency)

The following reports run for example daily in the R/3 / ECC system:

- Program RAPOKZFX
This program checks and adjusts the field MARC-APOKZ to keep it in synchronization with integration models.
- Report RCIFIMAX
This report generates and reconciles runtime versions of integration models.

3.2.6 Internal Consistency Check for SCM regarding LC <-> DB

Transaction /SAPAPO/OM17 is the main tool for an overall internal consistency between liveCache and APO-DB.



SAP recommends that you perform consistency checks during a posting-free period of time. If you cannot do this, it is possible that inconsistencies between liveCache and the SAP APO database that existed only briefly will be displayed.

You should re-check the inconsistencies that were displayed (Ctrl+F2: Button “Check again”) in this case.

If the inconsistencies are still displayed after the check, you can assume that the inconsistencies did not just exist temporarily and use the appropriate transaction to correct them. It is recommended to execute the program in the background (F9 or Menu Path *Program -> Execute in Background* as it is possible to use the functionality “Evaluate Last Background Job” afterwards. See SAP note 425825 for further information on this report. Several consistency check reports were integrated in this transaction as of APO release 3.1 like transaction /SAPAPO/REST02 for resources. During this reimplementation, the scope has been enhanced so it is recommended to use the functionality within /SAPAPO/OM17.

3.2.6.1 Check Frequency

The different business objects should be checked once per day if needed.



The monitoring object “DCAPOM17” exists within the SAP Solution Manager’s Business Process Monitoring functionality to facilitate the regular monitoring with OM17

3.2.6.2 Important Notes for LC<->DB Consistency:

Component	SAP Note	Description
SCM-TEC	425825	Consistency checks, ...

Table 3.6: Important Notes for LC <-> DB Consistency

3.2.7 Consistency at Interface: Transaction Data (External Consistency)

There are several tools to monitor the inbound and outbound queues across the systems.

Activate and use CIF post processing (transaction /n/SAPAPO/cpp1 or report /SAPAPO/CIF_POSTPROCESS) or otherwise use the qRFC monitors/SCM Queue Manager/Core Interface Cockpit to monitor issues with blocked queues to identify master data problems, correct master data, and retrigger the queues.

External consistency checks like delta report3 will display inconsistencies resulting from blocked queues and resend the data. However, this will not correct the inconsistencies as the queues are still blocked.



By just deleting blocked queues, delta report3 would correct the hereby provoked inconsistencies but such a procedure is not at all recommended. You should identify the root cause of the blocked queue and retrigger the queue processing instead.



You could utilize proactive online monitoring and alerting within the SAP Solution Manager’s Business Process Monitoring Framework besides the reactive CIF specific monitoring. The appropriate monitoring object for qRFC monitoring is IMQRFMO

3.2.7.1 Example

CIF post processing displays SCM inbound queues that cannot update SCM. The queues relate to VMI Sales Order 4185622/10, 4185622/10 and 4186738/20. The delta report detects this error:

Promotion	R/3 Order Number	Item	Sch.	Material	Plant	ChgPointer	PP Record	Error Code
	0004185622	10	1	000000000000415209	1202			221
	0004186738	20	1	000000000000415209	1202			221
	0004186885	10	1	000000000000415209	1202			221

Figure 3.3: Example for External Inconsistencies

Best Practice: Data Consistency Check

Reactivating the queues, the following error messages are found in the SCM application log in transaction /SAPAPO/c3: "Location 0001065469 does not exist":

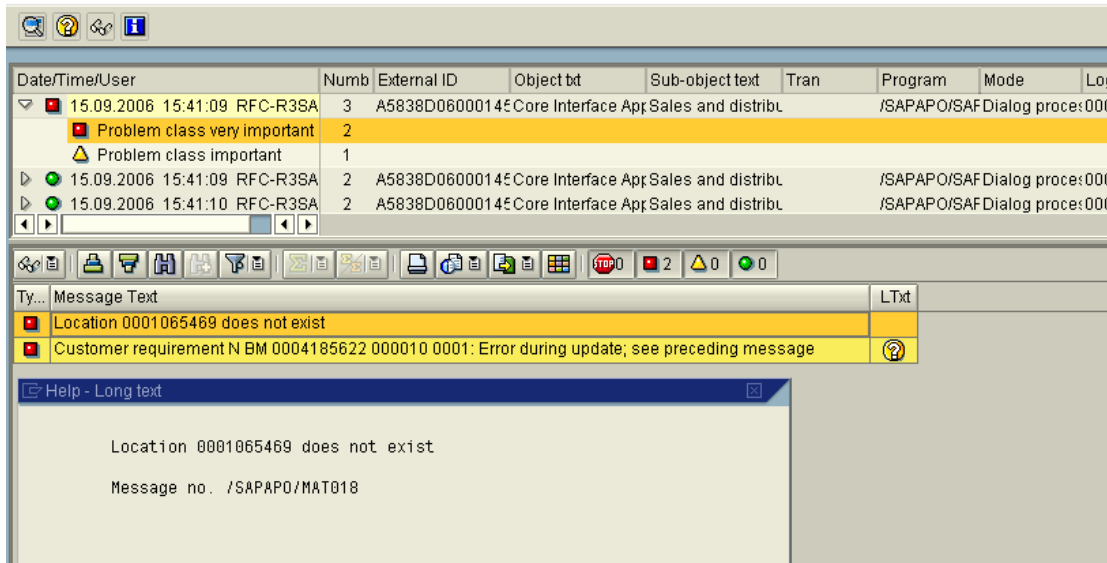
Display logs

Figure 3.4: Application Log

A comparison with the VMI order header leads to the finding that a new R/3 Ship-to-party was created which is used with VMI orders.

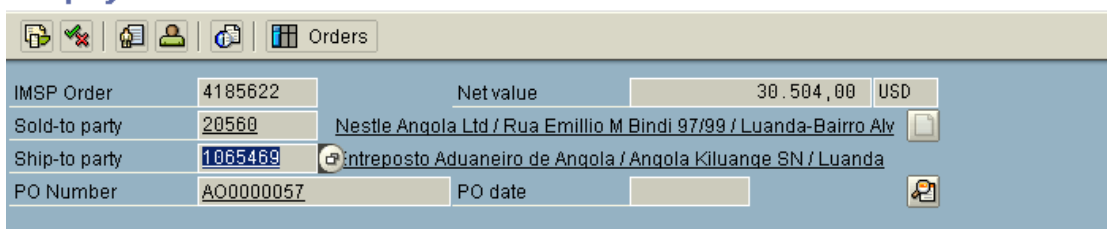
Display IMSP Order 4185622: Overview

Figure 3.5: Details of the Affected Order

The reason was therefore a master data problem and the new Ship-to-party was not yet transferred to SCM.

3.2.8 Temporary Quantity Assignments (TQAs)

3.2.8.1 Root Causes of Old "Expired" TQAs and Business Impact

Typical root causes of old non-persistent TQAs are:

- Manually deleted queues (for example, because of unresolved "sysfail" status)
- Manually deleted transactional RFCs in transaction SM58
- Time out dumps in R/3 (monitoring and RC analysis with transaction ST22)
- Update errors in R/3 (monitoring and RC analysis with transaction SM13)
- Issues with mass delivery processing (transaction VL10)
- Back order processing (transaction /SAPAPO/BOP)

The business impact of incorrect temporary quantity assignments are sales orders which will not be confirmed during manual GATP checks, interactive backorder processing (BOPI), and backorder processing (BOP), even if stock is actually available (ATP under confirmation). If deleting required TQAs manually, ATP over confirmations (= negative ATP) can occur.

3.2.8.2 General Monitoring and Tools

You should ensure that the temporary quantity assignments you want to delete do not originate from processes that have not yet finished. Therefore, it is essential to check whether there are corresponding queue entries using transaction SMQ2. If so, these must be

restarted. Since locking problems are often the reason why temporary quantity assignments are kept, you should check transaction SM58 regularly and process transactional RFCs if necessary. This transaction should be scheduled as a batch job (for example, every 15 minutes) by scheduling report RSARFCEX as a job.

3.2.8.3 Check Frequency

The deletion of old temporary quantity assignments should be performed regularly by scheduling the report /SAPAPO/OM_DELTA_REMOVE_OLDER at least once (or, if required, several times) per day as a general recommendation. The definition of the deletion offset depends strongly on the customer's business scenarios and monitoring procedures. Examples are 1 hour, 1 day or 2 days. In addition, if required, the TQAs should be monitored regularly using transaction /SAPAPO/AC06 and deleted manually on demand.

TQAs caused by backorder processing are deleted using transaction /SAPAPO/BOP_DELETE. But the BOP result lists may be kept longer in the system. See also SAP note 488725 for further information.

3.2.8.4 Overview of notes for TQA consistency:

Component	SAP Note	Description
SCM-APO-ATP	488725	FAQ: Temporary quantity assignments in Global ATP

Table 3.7: Notes for TQA Consistency

3.2.8.5 Tips for Root Cause Analysis of Erroneous TQAs

3.2.8.5.1 Use Central System Log

Larger installations often consist of many application servers, whereby the system log is written for each single application server. To reduce the time needed to monitor each application server using transaction SM21, it is possible to use central logging choosing the menu path: *System log -> Choose central system log / all remote system log*.

3.2.8.5.2 Use of Transaction /SAPAPO/AC06

1. Define a layout (Ctrl + F8) that contains the additional displayed column "Generation Date" and save it.

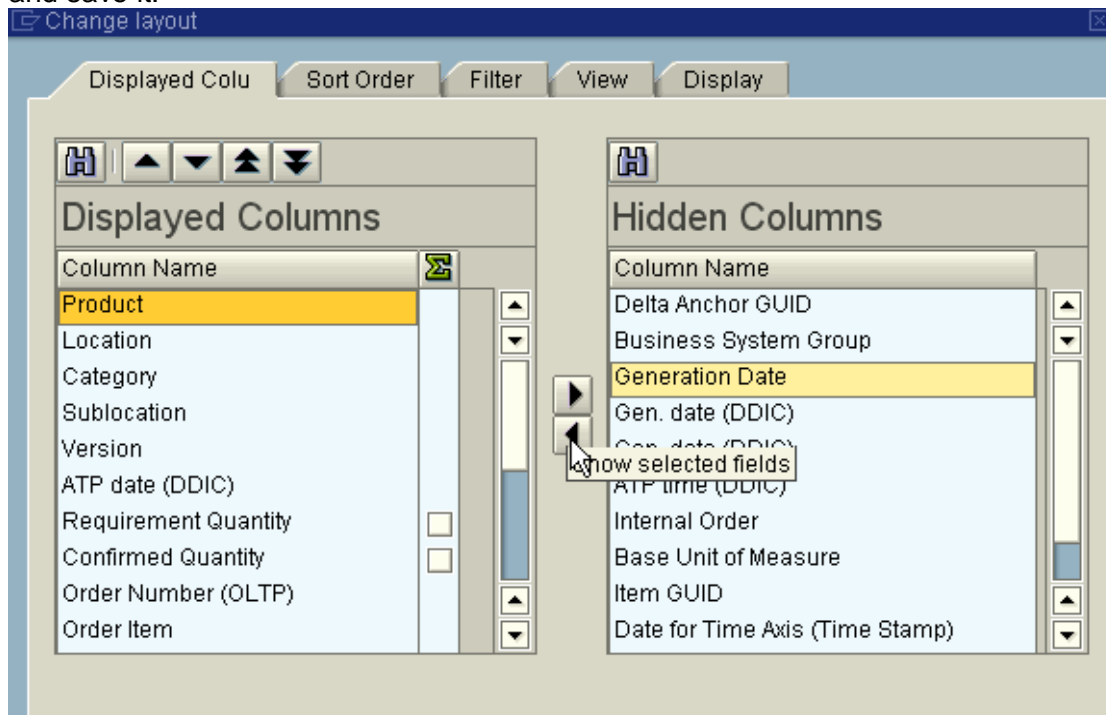


Figure 3.6: Defining the Screen Layout

Best Practice: Data Consistency Check

2. Set a filter in transaction /SAPAPO/AC06 by marking PerIndTQA, right click and select "Set filter..." by Persistency Indicator = N (non-persistent):

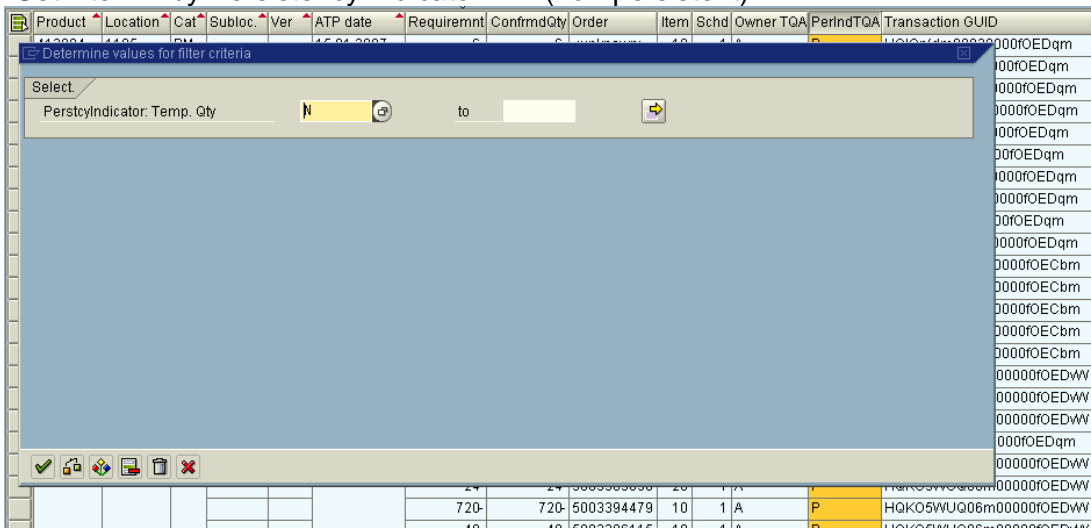


Figure 3.7: Defining a Filter

Comment on the "Generation Date":

Do not use the fields "gen. date / time (DDIC)", as they refer to the local time zone of the end-user while dumps and the system log are written in system time which differs very often from the time zone of the end-user. The generation date of TQA is given in system time = UTC. Generally, you can compare the system time and time zone by selecting menu path: *System -> Status*.

Previous logon	04.09.2006	14:54:37
Logon		18:58:55
System time		23:07:12
Time zone	CET	05.09.2006 01:07:12

Figure 3.8: Comparison of Time Zones

The main issue that makes it very complicated to find the root cause of non-cleared TQAs is that the originating transaction cannot be seen in monitoring transaction /SAPAPO/AC06 but only the transaction GUID. The transaction GUID (TRGUID) is a key for temporary objects that have been created during an availability check. All these objects have the corresponding transaction GUID and can therefore be identified by it. The transaction GUID is not saved in R/3 and linked to the transaction in a special table.

Temporary Quantity Assignments: ATP

Product	Location	Cat	Subloc	Ver	ATP date	Requirement	Confirmed Qty	Order	Item	Sched	Owner TQA	PerIndTQA	Transaction GUID	Name
414450	1195	BM			20.10.2006	60	0	5000067200	30	1	A	N	HJ0q9RB06a20000fOECm	BRLEPLERL
					21.10.2006	10	0	5000067200	30	1	A	N	HJUq9RB06a20000fOECYm	
					23.10.2006	0	10	5000857208	30	1	A	N	HJUq9RB06a20000fOECYm	
					20.10.2006	10	10	<unknown>	70	1	A	P	HJ0IE4p406a20000fOECYm	BRGOUVEIARI
					23.10.2006	10	10	<unknown>	30	1	A	P	HJSEYWK0Dq20000fOEDLW	
					19.10.2006	20	0	5000030070	70	1	A	N	HJSuSm1t0B40000fOEDkm	BRLIMAE
						20	10	<unknown>	30	1	A	N	HJSVT1Li0Gi0000fOEDkm	
						20	20	<unknown>	30	1	A	N	HJSuSm1t0B40000fOEDkm	

Record Ty.	AcAssgtObj	ExprDat	TmpQtyAs	Generation Date	Gen. date	Gen. time	Date for Time Axis	ATP time
A		31.12.2037	23:59:59	04.09.2006 23:05:28	05.09.2006	01:05:28	25.09.2006 15:00:42	17:00:42
A		31.12.2037	23:59:59	04.09.2006 23:05:28	05.09.2006	01:05:28	25.09.2006 15:00:42	17:00:42

Figure 3.9: Investigation of TQAs

The following checks are recommended based on this data:

1. Take the end-user name (field Name) and look for update errors in transaction SM13 by selecting this user and flag "terminated" only.
2. Check the system log in transaction SM21 by central logging for this user

Best Practice: Data Consistency Check

3. Check dumps in transaction ST22 for this user
4. Use the time stamp of the TQA (field Generation Date) and check whether the identified potential issues match this point in time
5. Check the affected document (field Order) and change history in the corresponding application transaction (for example, va03: Menu path: Environment -> Changes): If the “suspicious” changes were executed by a background job, you can find the job by searching in transaction sm37, using the button “Extended job selection” and filtering for jobs that were running around the time of the document changes (for example, in the depicted case from 07:02:15 till 07:02:20)



Figure 3.10: Verification of Change Logs

6. Check whether related BOP runs exist by searching with the Transaction GUID (TRGUID) in table /SAPAPO/BOP or /SAPAPO/BOPHEAD using transaction se16

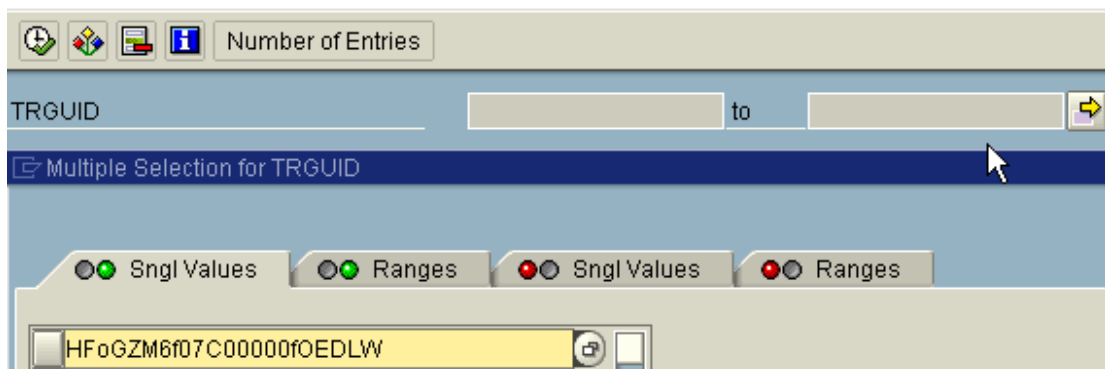
Data Browser: Table /SAPAPO/BOPHEAD: Selection Screen

Figure 3.11: Verification of /SAPAPO/bophead using the Data Browser

In case of regular TQA issues with deliveries, go to transaction VL10 and check if there are delivery runs by user name (field “Name”) that can match also with (field “Generation Date”). If you find no related error messages in the VL10 log, you can open a customer support message for SAP development.

3.2.8.6 Root Cause Analysis of Erroneous TQAs: Examples

To resolve the root causes of inconsistencies, you need to use monitoring to identify typical and reoccurring errors. The main challenge is to reproduce these errors. Reproducing the errors is the only way to correct modification errors.

These modification errors can then be resolved either by your internal development department or by SAP development but might be charged as remote consulting.

3.2.8.6.1 Example

In the case of many TQAs linked to one transaction GUID with the ATP category BR = deliveries and one user, it is likely that the root cause is the collective delivery runs performed by the same user:

This case was monitored with **/SAPAPO/AC06**: Note the large transaction GUID for the same Generation date 08.11.2006 and Generation time 15:24:54 till 15:25:00 by the same user. (The ATP category BR = deliveries is not visible in this screenshot).

Item	Schd	Owner T...	PerIndT...	Transaction GUID	Name	S	Spec.Sto...	Record Ty.	AcAssgtObj	Generation Date	Gen. date	Gen. time
330	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:25:00	08.11.2006	15:25:00
60	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:55	08.11.2006	15:24:55
240	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:58	08.11.2006	15:24:58
230	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:58	08.11.2006	15:24:58
70	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:55	08.11.2006	15:24:55
220	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:58	08.11.2006	15:24:58
30	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:54	08.11.2006	15:24:54
170	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:57	08.11.2006	15:24:57
20	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:54	08.11.2006	15:24:54
50	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:55	08.11.2006	15:24:55
80	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:55	08.11.2006	15:24:55
100	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
110	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
140	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
120	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
160	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:57	08.11.2006	15:24:57
150	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
130	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
210	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:57	08.11.2006	15:24:57
90	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:56	08.11.2006	15:24:56
290	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:59	08.11.2006	15:24:59
40	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:55	08.11.2006	15:24:55
300	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:59	08.11.2006	15:24:59
310	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:59	08.11.2006	15:24:59
320	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:59	08.11.2006	15:24:59
180	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:57	08.11.2006	15:24:57
200	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:57	08.11.2006	15:24:57
260	1	A	N	HL7N0a2H04420000f0EDkm	USJONBRID			A		08.11.2006 14:24:59	08.11.2006	15:24:59

Figure 3.12: Overview of TQAs

Select the VL10 collective processing logs and select by user and date:

Best Practice: Data Consistency Check

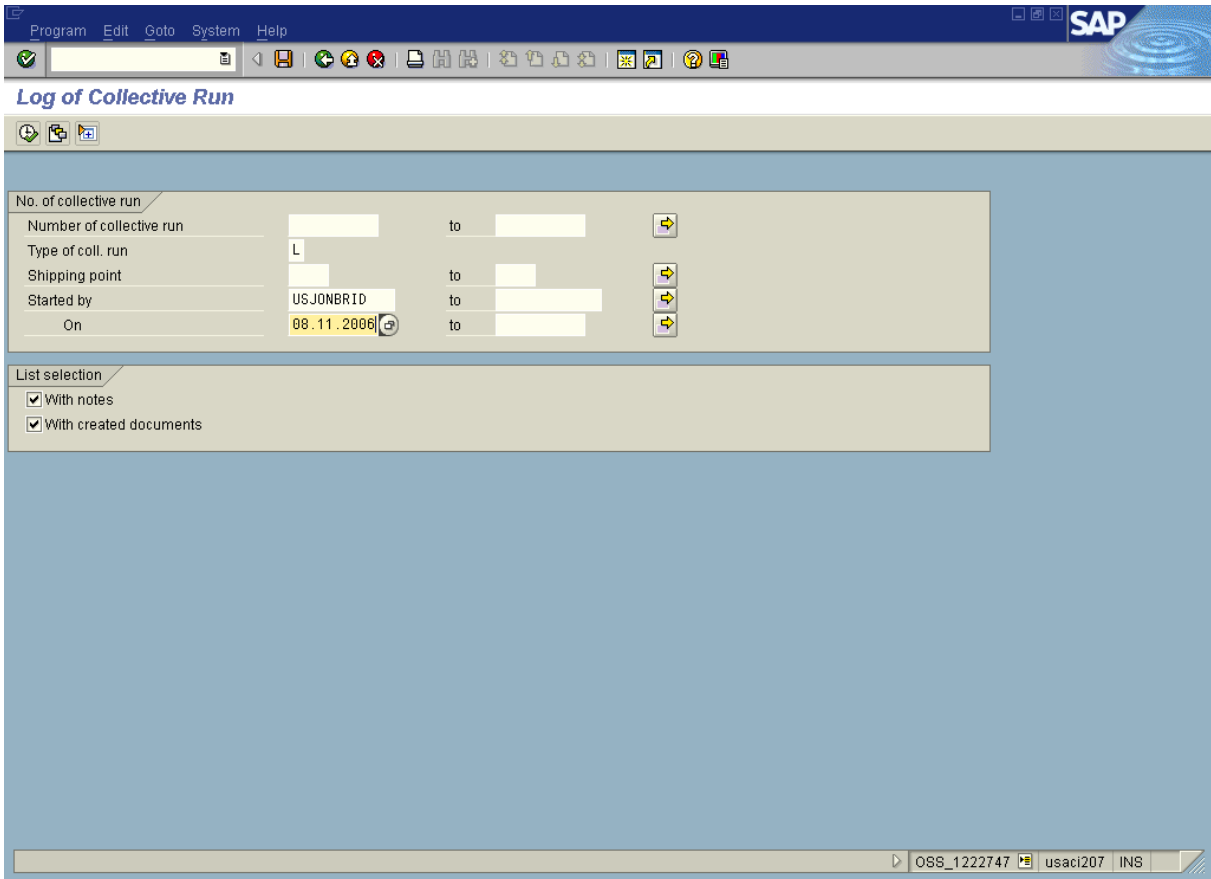


Figure 3.13: Displaying the Log in Transaction VL10

Find the logs of the delivery runs that generated the TQAs:

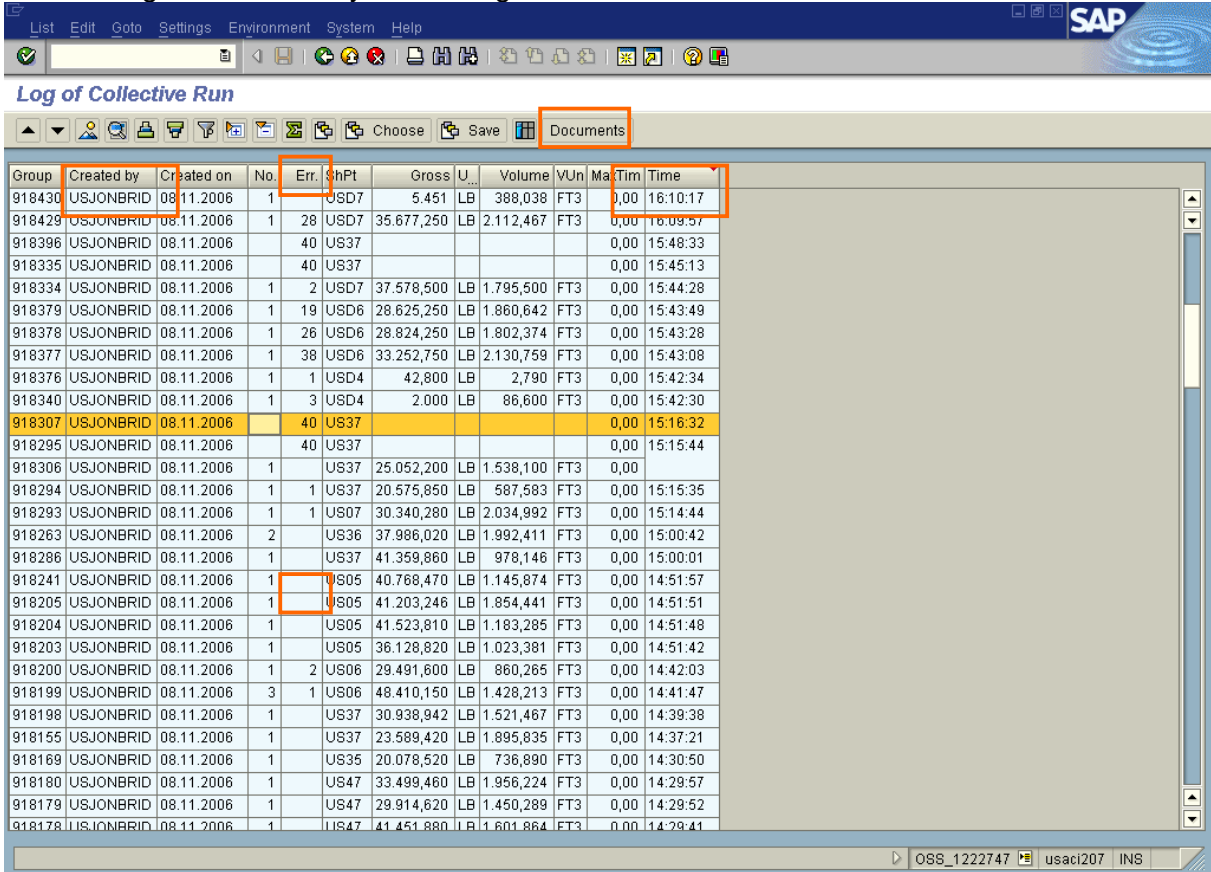


Figure 3.14: Display of the Log-file

Check the errors (you can see the number of errors displayed: for example, 40 errors) and navigate to the errors. If those errors cannot be associated with TQA issues or there are no errors in the log, a trap code could be discussed with SAP development and implemented in order to find the VL10 issues when related expired TQAs remain.

3.2.9 Transaction Data: All Order Types and Stock (External Consistency)

It is recommended to use the /SAPAPO/CIF_DELTAREPORT3 to compare/reconcile transaction data and correct data inconsistencies between SAP APO liveCache and SAP R/3 DB. Data inconsistencies can occur during the transfer using SAP APO Core Interface (CIF), if, for example, you delete entries in the queue manually or if faulty queue entries exist. The CIF comparison/reconciliation takes into account both objects that are not available in one of the systems and objects that differ in the two systems.



The regular monitoring of this report may be facilitated with the monitoring object "DCAPOCCR" in the SAP Solution Manager's Business Process Monitoring Framework.

3.2.9.1 Handling of Temporary Differences

3.2.9.1.1 No Iteration Functionality Applied Yet?

You can determine whether inconsistencies exist between the SAP R/3 DB and SAP APO liveCache by checking the job log, and therefore if the report has to be restarted in online mode. This should only be the case for old releases! See SAP note 425825 for details.

3.2.9.1.2 Iteration Functionality (Online and Background)



It is strongly recommended to apply SAP notes 496779 and 488747, which introduce the iteration functions for the CIF_DELTAREPORT3.

Due to long delta report runtimes, objects may have been changed and may no longer be inconsistent. Temporary data inconsistencies occur during the transfer of data between SAP APO and SAP R/3 and disappear again after the transfer. Note, however, that in the case of lengthy transfers, inconsistencies may still be displayed as errors.

You should use the iteration for the online comparison and for the comparison in the background, as well as when saving and loading results.

The user should interactively compare the displayed incorrect objects from the result list again (iteratively) after the DELTAREPORT3 has performed a comparison and displayed the result when using the online comparison. If you execute the iteration several times in a row, you can further minimize the number of probable errors caused by temporary data inconsistencies.

You should set the indicator Iteration for the comparison in the background where intermediate results are saved and loaded for comparison. Temporary errors or already solved errors are cleansed and do not appear in the result list anymore. The update of the errors is triggered manually as the next step.

You can also use the iteration after the reconciliation to check whether the correction was successful or not. In this way, you can determine whether an error still exists after the reconciliation or not.

3.2.9.2 Recommendation

When using the /SAPAPO/CIF_DELTAREPORT3 for sales orders, setting the flag "Use Table VBBE for Sales Order Comparison" is recommended for a much better performance. You have to run report SDRQCR21 on the SAP R/3 / ERP side in simulation mode first (see

Best Practice: Data Consistency Check

also paragraph 3.3.1.5) to ensure that the requirements situation is correct in the primary system if this variant is selected.

If incorrect requirements exist, run SDRQCR21 again with data transfer before executing /SAPAPO/CIF_DELTAREPORT3 on APO/SCM side. Inconsistencies from ERP are likely transferred to SCM generating a business impact if you skip this step. The BOP runs could be executed on a defective data basis resulting in ATP over confirmations (= negative ATP) or ATP under confirmations.

3.2.9.3 Check Frequency

The different business objects should be checked by CIF comparison/reconciliation of transaction data at least once a week up to once per day if needed.

3.2.9.4 Example for Inconsistencies Detected by the CIF Comparison/Reconciliation

In this example, the result overview of the delta report is displayed. Checked were 750 sales orders and deliveries across a SAP SCM and SAP ERP system:

Results Overview for Delta Report						
Following Objects were Compared:						
Documents						
Object Type:	Total	Error	Not in APO	Not in R/3	Contents	Not in IMR
Sales Scheduling Agmnts	0	0	0	0	0	0
Sales Orders	750	784	149	142	493	0

Detailed Results for Delta Report							
Sales Orders with Errors							
Missing in APO							
R/3 Order Number	Item	Sch.	Material	Plant	ChgPointer	PP Record	Error
0806546360	170		000000000012019039	1306			236
0806546359	260		000000000012023063	1306			236
0806546359	250		000000000012022940	1306			236
0806546359	240		000000000012019461	1306			236

Figure 3.15: Result Screen for CIF Delta Report

The delta report detected 142 Sales orders in SCM that are not in the R/3 system anymore and 149 sales orders missing in SCM (error: Missing in APO). That kind of errors is always critical and might have a business impact.

Typical root causes for both errors are:

- VBBE entries were deleted or generated due to temporary (fake) errors with SDRQCR21
- Queues were deleted

The report detected additional 493 errors of type "Differences in Content". This kind of errors might not be critical depending on the difference in content.

Typical root causes for this kind of error are:

- Queues were deleted
- User-exits regarding the detailed scheduling of an order item
- Bugs

3.2.10 Sales order requirements and SD mapping tables (external consistency)

The report /SAPAPO/SDRQCR21 checks sales order requirements and delivery requirements in SAP R/3 / ERP and SAP APO/SCM.

Only this report checks precisely the SD order tables (mapping tables: /SAPAPO/posmapn, /SAPAPO/ordadm_i, /SAPAPO/schedlin, /SAPAPO/sdqtvb,...) These mapping tables checks are not performed by the CIF delta report!

It is recommended to use this report if the /SAPAPO/cif_deltareport3 report cannot correct the inconsistencies, especially inconsistencies of specific incorrect mapping tables. If you are using the backorder processing (BOP), especially with Rule Based ATP, you should use this report regularly to check the consistency of the SD mapping tables, as BOP relies on these tables being consistent. For more information, refer to SAP Note 444641.

If you run report SRDQCR21 regularly on SAP R/3 / ERP, you can set the flag "read requirements from table VBBE" in report /SAPAPO/SDRQCR21 to improve its runtime significantly (which is the commonly used procedure for checking a large volume of orders). For few orders, it is quite practical to use the (much slower) option "build requirements from document flow".

Recent enhancements of /SAPAPO/SDRQCR21 (by SAP note 987299)

The redesign of report SDRQCR21 (see section 3.3.1.5) resulted in a complete new coding compared to the previous version of SDRQCR21 with respect to regenerating the requirements. In order to avoid redundancy, a new common interface was created and /SAPAPO/SDRQCR21 will use the coding of the new version of SDRQCR21.

In addition, the following improvements were integrated into /SAPAPO/SDRQCR21

- The possibility to check specific order numbers or order items
- Iteration (quite similar to the iteration of the CIF comparison): The report first selects orders in SCM, then in R/3 and compares the selected orders. A new check is triggered in case of inconsistencies until the inconsistencies disappear or the number of maximal iterations is reached. A waiting time [s] can be defined from one iteration step to the next.
- Table /SAPAPO/OBREF (document flow) was only checked for RBA subitems. With SAP note 987299, the report can identify and correct an incorrect document flow including the document flow from Stock Transfer Orders to the delivery document.
- More accurate evaluation and correction of Product Allocation Assignments
- It is now possible to exclude fully delivered orders and/or to check sales orders or stock transfer orders only which improves the run time.
- Redesign of the requirement recompilation

If you use the option to build requirements from the document flow, a very long runtime may occur and the requirements are recreated without document blocks. Therefore, inconsistencies may occur during the operation. With the new version of the report (for which notes 998102 and 1023543 are prerequisite) a new logic for recompiling requirements is introduced. The following new switches are available which allow you to use the report during system operation.

- The first Flag 'Lock documents...' sets a document block if the requirements from table VBBE are rewritten in the R/3 system. This prevents the document from being changed at the same time.
- The second flag writes a planning file entry (net change per material/plant) in the R/3 system if the requirement situation has changed. The next MRP reads this and plans accordingly.

3.2.10.1 Handling of Temporary Differences

Within the new enhancements of report SDRQCR21 a better handling of temporary inconsistencies was introduced which is described in detail in this section. When using the old version, you need to execute appropriate steps manually.

Best Practice: Data Consistency Check



When entering a logical system, first an RFC is triggered to ECC to check if the notes 998102 and 1023543/new version of SDRQCR21 is applied. If yes, specific selection fields accept input, otherwise keep greyed out. If Flag 'Lock documents...' can be entered, the new version of SDRQCR21 is active and used for reconciling and updating the VBBE requirements.

Build Reqmts from Doc. Flow
 Update DB Table VBBE
 Lock Documents During Update Write Planning File Entry

Figure 3.16: Specific functions available with new version in ECC

Example settings for iteration:

Number of Iteration Steps 5
 Wait Time Between Iter. Steps 2

Figure 3.17: Definition of Iteration Steps

Example for the iteration execution: An order was changed in R/3; the CIF-Queue still hangs in APO Inbound. Report /SAPAPO/SDRQCR21 would find an inconsistency and resend the order (unnecessarily) to APO. The report waits x seconds (for example, x=2 as shown in Figure 3.17: Definition of Iteration Steps) with the iteration activated. If the CIF queue was updated meanwhile, this sales order is now consistent and will disappear from the result list with the next iteration step. During the execution of the report, the iteration step and waiting time are displayed.

Iteration 3 / 5 Wait 2 Seconds

Figure 3.18: Display of Execution Time and Iteration

The refresh is triggered to resend all data identified as inconsistent after the last iteration step as a real inconsistency is assumed in this case. The information on how many inconsistencies remained after each iteration step may be displayed on the result screen by using the list display "Sort by document number".

Number of Checked Requirement Items	:	341
- Missing in APO	:	51
- Inconsistent in APO	:	8
- To be Deleted in SAP APO	:	0
- To Be Sent to R/3	:	0
Number of Checked Items in SD Order Tables	:	300
- Missing in APO	:	103
- Inconsistent in APO	:	1
- To be Deleted in SAP APO	:	0
Number of Iteration Steps Performed	:	5
- Step 01	:	
Inconsistent Requirements	:	59
Inconsistent SD Order Tables	:	181
- Step 02	:	
Inconsistent Requirements	:	59
Inconsistent SD Order Tables	:	181
- Step 03	:	
Inconsistent Requirements	:	59
Inconsistent SD Order Tables	:	181
- Step 04	:	
Inconsistent Requirements	:	59
Inconsistent SD Order Tables	:	181
- Step 05	:	
Inconsistent Requirements	:	59
Inconsistent SD Order Tables	:	181
Sales Order and Delivery Requirements (liveCache)		
INSERT: R/3 --> APO		
0000014720	000010	0001
0000014720	000010	0002
0000014720	000020	0001
0000014720	000020	0002

Figure 3.19: Result of /SAPAPO/CIF_DELTAREPORT3**3.2.10.2 Check frequency**

You do not need to run the report /SAPAPO/CIF_DELTAREPORT3 for sales orders if report /SAPAPO/SDRQCR21 is scheduled on a regular basis as this is already carried out by the report /SAPAPO/SDRQCR21.



The report /SAPAPO/CIF_DELTAREPORT3 selects customer orders which are contained in active and inactive integration models. You should run report /SAPAPO/CIF_DELTAREPORT3 for sales orders from time to time, to perform a data cleansing of inactive SCM data.

You can always run the report /SAPAPO/SDRQCR21 on demand, selecting as little data as required, for example selecting affected sales order numbers only.

3.2.10.3 Important SAP Notes

Component	SAP Note	Description
SCM-APO-INT-SLS	987299	Report /SAPAPO/SDRQCR21: Iteration options
SCM-APO-INT-SLS	444641	Correction of incorrect sales order requirements
SD-BF-AC	998102	SDRQCR21: Enhancements to support the check
SD-BF-AC	1023543	Function Group V03T enhancements for sdrqcr21

Table 3.8: Important SAP Notes

3.3 Tools to Check Consistency within ECC**3.3.1 Tools for Processes Involving SD and LE****3.3.1.1 General Remarks for SD/LE Corrections**

Most correction reports in SD and LE correct documents by reading the document information, recalculating and saving the changed information on the database. Basically, this corresponds to performing transaction VA02/VL02/VF02, but the following side effects can occur:

- Using updates based on VDATU, data could be written in a period other than the original document date
- Some data could be recalculated with the validity of the date (for example, credit management data).

Both time shifts may affect results in BW and LIS and need to be considered when correcting SD/LE transaction data. As long as only a few documents are changed, the time shifts are negligible, but for larger amounts of data written into the current period a reconstitution of statistical data should be performed.



All correction reports should be handled very carefully as an observed inconsistency could be normal system behavior after certain actions like archiving.

Example: The document flow indicates a missing delivery. In this case, the question before attempting to correct the situation is whether the delivery itself is missing or whether the information in the document flow is incorrect. The document flow needs to be corrected in the latter case, while in the first case a delivery needs to be recreated. A correction attempt of the document flow in the first case will cause inconsistencies with potentially disastrous side effects.



NONE of these correction reports should be planned on a regular basis in correction mode! They should only be used to correct single faulty documents after root cause analysis and may destroy data if used carelessly.

Example: The run of a report for correction of document flow followed by SDVBUK00 after archiving of billing documents and deliveries will open ALL sales documents again thus causing a production down situation.

3.3.1.2 Incorrect Document Flow

Within standard SD, the document flow is derived from two locations: The document flow table VBFA which is mainly used to calculate the document flow from a preceding document to a follow-up document and entries in the documents on item level (fields VGBEL, VGPOS in tables VBAP, LIPS and VBRP) on which the display of document flow from follow-up document to preceding document is based due to performance reasons.

Depending on whether a document is missing or a document flow entry is missing, several reports exist to correct the data:

- ZZKVBFA6: This report is intended to correct the document flow from order to delivery. This report deletes document flow entries in case the document flow has not been deleted when a delivery was deleted. The report is given in note 74513. Please note that for archived deliveries, the document flow is needed. A similar report is RVKVBFA1; for details see SAP note 4483
- ZVBFADDEL: Report for correcting the document flow from order to delivery. This report will delete the document flow in case a temporary document number (begins with \$) has been saved. The report is given in SAP note 134939.
- ZZVBFA_COMPLETE: Report for correcting the document flow from order to delivery. This report is the opposite of the above mentioned reports. While the former reports delete the document flow, this report will create missing document flow entries if a delivery exists for an order. See SAP note 427059 for details.
- WSCOR001: Report for correcting the document flow from delivery to LVS-TO. This report will delete the document flow from the delivery to the TO if a document flow exists from delivery to TO but the transfer order has been deleted. The report is provided by SAP note 70310.
- WSLIPS02: Report for correcting the document flow from delivery to LVS-TO. This report will delete the document flow from delivery to TO if the delivery has been deleted but the flow to the TO still exists (SAP note 28448).
- ZTRVBFA01: Report for correction of document flow from delivery to transport. If a transport has been created for the delivery but the transport is not shown in the document flow, this can be corrected by SAP note 396253.
- ZZVBFA01: Report used for correction of document flow from delivery to invoice. This report creates VBFA-entries similar to report ZZVBFA_COMPLETE but this time from delivery to invoice. See SAP note 38587 for details.
- ZZVBFA02: Report used for correction of document flow from order to invoice. The report given in SAP note 152051 is very similar to report ZZVBFA01 but this report creates the document flow between order and invoice in case of order related billing.
- ZZVBFA03: Report for correcting sales orders after purchase order deletion. If the purchase order is deleted, the field VBFA-RFMNG should be set to zero. This report which is given in SAP note 521416 can be used to display and/or correct documents where this was not the case.

3.3.1.3 Inconsistent Document Status

When talking about the status of SD/LE-documents, we need to distinguish between two different statuses:

- The object status
- The document status

While the document status contains relevant information about the document's status regarding the SD/LE business process, the object status contains information about the operational status of the document. The document status is available on header and item level and is stored in the SD tables VBUK (key field VBELN) und VBUP (key fields: VBELN

POSNR) associated with the individual document. The object status is stored in the general status tables JEST/JSTO and is only linked to the SD-document via field OBJNR.

The main correction report for the document status is report SDVBUK00. This report should always be used if a document status was determined incorrectly for a sales document. The report can only correct the status if the display of the status in the sales document differs from the status on the database as in both cases the same calculation rules are applied. If the same status is displayed in the document and on the database, the report SDVBUK00 will not correct the document. In this case, it is necessary to find the cause for the status determined if you think they are incorrect.



The best way to find out why a document status should change is to debug the calculation of the document status in the sales document using breakpoints in function modules RV_XVBUP_MAINTAIN for item status information and RV_XVBUK_MAINTAIN for the header status. Here, you can see in detail on which data the individual status information is calculated and how the (incorrect) result is determined.

A common root cause for an incorrect overall processing status of a sales order is the maintenance of a completion rule in the customizing of the order's item category (transaction VOV7). The solution would be to remove the incorrectly maintained completion rule from item categories for which no subsequent sales documents will be created during the normal business process. The completion rule should only be entered in item categories which are used for inquiries, customer quotations or contracts. Sales document affected by this incorrect customizing can only be corrected by changing the customizing of the item category, transferring the item category to existing sales document (as in most customizing changes existing sales documents are not updated) by means of report ZZERLRE which can be found in SAP Note 79193. After correction of the customizing error the document status can be fixed using SDVBUK00.

Report SDVBUK00 should definitely not be scheduled daily and for all documents. Instead, it should only be started if required for the correction of an individual document after careful investigation.

A similar report exists to correct document status information in deliveries and is called RVDELSTA. The working model is identical to SDVBUK00 but considers the specific status information available in deliveries.

So far, the document status has been updated but also inconsistencies with the object status can arise. Several correction reports exist to rectify errors within the object status link in the SD documents.

SD documents with object number in SD-Tables but no corresponding status object in table JSTO: Commonly, these cases will result in error message BS001 when changing the affected documents. These inconsistencies can be identified using standard report SDSTATU1 or the generic consistency check report delivered by ST13. In both reports you can list entries on header and item level where field OBJNR is filled but no corresponding entry exists in table JSTO. The found inconsistencies can be corrected using report SDSTATU2. This report creates entries in the status tables if an object exists in VBAK or VBAP but not in the status tables. After creation of the corresponding entries, the report also sets appropriate status information.

SD documents with assigned object number in status tables but missing link in SD tables: Two reports have been delivered by SAP notes correcting this information on header and item level. Report ZONRVBAK corrects documents having an incorrect status object (VBAK-OBJNR initial). The report ZONRVBAK fills VBAK-OBJNR if entries in the status tables exist but the connection got lost in table VBAK. The corresponding report on item level is called ZONRVBAP. Both reports are given in note 162869 and enhanced in note 413555

3.3.1.4 Inconsistency Between Billing and FI

A common inconsistency due to incorrect enhancements in user exits is FI documents without a corresponding SD billing document. To check and monitor these cases, report ZF_MISS_SD can be used which is available via OSS note 904810. The report

Best Practice: Data Consistency Check

ZF_MISS_SD checks for FI documents (table BKPF) with the reference activity VBRK. Afterwards, table VBRK is checked for corresponding entries with document number VBELN equal to BKPF-AWREF. If no corresponding entry is found, company code, fiscal year, period and document number of the respective FI document as well as the number of the missing billing document (=BKPF-AWREF) are listed.

In a second list, the report displays the summary local currency amounts from the affected FI documents, according to company code, fiscal year, G/L and customer account.

If report ZF_MISS_SD has identified missing documents and you have made sure that the missing billing documents have not been archived, perform a transactional correctness analysis of the coding. In all cases identified in the past, incorrect coding in user exits and enhancements has been identified as the root cause where an implicit or explicit commit was performed destroying the logical unit of work. To understand the impact of the coding mistakes and where to find these, it is essential to understand the interaction between SD and FI in this process.

During the posting of billing documents, document number gaps can occur sporadically although the buffering for the SD document number object RV_BELEG is switched off as a document number for the billing document is assigned early in the posting process in the dialog task (Function module RV_INVOICE_DOCUMENT_ADD). Once the document number has been assigned, this is passed on to FI to set up the tables of the accounting interface. If a severe error occurs (for example, a termination in update task), the update is terminated prematurely and all sub processes - including number assignment - are rolled back. But a roll back can only roll back data up to the last database commit. This means that a rollback always goes back to the last COMMIT command of the application, therefore either to the beginning or the last successful update. This means in turn that gaps in a number range only occur when the system cannot roll back the data completely due to intermediate database commits. Unfortunately, database commits are not only triggered by the ABAP command COMMIT WORK but also by a number of other actions like sRFC-calls, screen changes, and so on.

In a standard SAP R/3 system, exactly one COMMIT command is used during the billing process. This in turn means that it is required to examine the process of the billing for possible enhancements (user exits, modifications, and additional printing programs) by means of a code review or by runtime analysis (ST05) of the billing runs including the following programs:

- billing document programs
- programs for the accounting interface
- accounting programs

Besides this detailed investigation, the following organizational measures can help to pinpoint the root cause of such inconsistencies:



Do not transfer billing documents automatically to the accounting; instead use the two step procedure. In this procedure you set a posting block in the customizing of the billing document type and use transaction VFX3 to post the billing documents into accounting after creation of the billing documents.

By separating the two steps, it is possible to determine whether the root cause is to be found within SD or FI coding. As a side effect, it is ensured that billing documents exist for all created accounting documents as the existence of a billing document is a precondition for the creation of the accounting document in this procedure. In addition, you should document missing document numbers in FI by report RFBNUM00 (SAP note 148757) and document update terminations (transaction SM13).

3.3.1.5 Inconsistent SD Requirements

3.3.1.5.1 Introduction

Sales documents (quotation, sales order, scheduling agreement, ...) and deliveries have requirements as long as they are not completed (completely delivered/cancelled/rejected). Those requirements are written, updated, and deleted accordingly in table VBBE. A detailed explanation of the handling of requirements can be found in SAP note 547277. Too many, too few, or simply incorrect sales documents or delivery requirements may trigger follow-on errors in planning, procurement (production, purchase order) or document processing (availability check).

3.3.1.5.2 Report SDRQCR21

Run report SDRQCR21 to ensure VBBE data is correct. If an SCM system is used, it is also necessary to run /SAPAPO/CIF_DELTAREPORT3 in addition for sales orders setting the flag "Use Table VBBE for Sales Order Comparison". Make sure all sales data is consistent. See SAP note 425825 for details. See also the chapter "Tools to check consistency between ECC/SCM" within this document.

How to solve reoccurring requirement errors: The most frequent errors have to be found first by monitoring and analyzing the spool lists. The most challenging step is to find a reproducible example afterwards. Errors in standard can be corrected by SAP development; errors in custom enhancements can be corrected by the responsible developers.

Results of report SDQRC21 can be monitored with the Business Process Monitoring in SAP Solution Manager with monitoring object DCSDQCR after applying SAP note 1036106.

3.3.1.5.3 Restrictions of Report SDRQCR21

No parallel processing with a data transfer: Only execute the SDRQCR21 report with a data transfer once you have made sure that no one will process sales documents or deliveries (for the materials/plants according to the selection) while the report is running. The reason for the restrictions is that the report does not read or set blocks. It therefore produces correct results only if no other program updates sales or delivery requirements, or checks availability while the report is running. The report SDRQCR21 may generate VBBE inconsistencies (temporary or "fake" errors) when it runs in parallel to requirement relevant transactions (for example, online transactions, backorder processing, mass delivery creation, mass batch determination, cancellation of sales orders, IDocs, EDI). The item requirements are wrongly calculated due to these parallel activities.

Further details can be found in SAP Note 25444 describing report SDRQCR21 in detail.

3.3.1.5.4 New Version of Report SDRQCR21



Large companies with global installations are often running 24 x 7 businesses with high volumes that make it nearly impossible to find time windows with a low system activity regarding sales orders and deliveries. Therefore, it was necessary to improve the report SDRQCR21. For the above mentioned reasons, the report SDRQCR21 was redesigned and an updated version was provided in SAP note 998102 (SD-BF-AC).

The new version of SDRQCR21e has a new enhanced selection screen:

Material	34 YT*								
Werk			to						
Document Date			to						
Documents			to						

Update Database
 Compare Mode
 Trigger MRP
 Enqueue

Figure 3.9: New Selection Screen of SDRQR21

A correction is also possible by document number within a very short runtime and/or document date. The other new flag is the Enqueue flag.

Regarding compare and non-compare mode: With locked mode active, the compare mode is useful because you are locking only documents with inconsistencies in requirements whereas the direct mode (non-compare mode) regenerates VBBE for every item (with or without requirement errors) and therefore locks every document.



It is recommended to use the compare mode if you do not want to lock many documents. This compare mode should be the normal usage of SDRQCR21.

Exceptional case: X is the percentage of items with requirement errors in your system (number of items with requirement errors / total number of items selected). It becomes interesting to switch to the non-compare mode instead of the compare mode when X is very high because you skip the pre-selection phase and the compare routines to directly regenerate the items requirement. This non-compare mode should be used exceptionally (only in case the whole VBBE is corrupted for one or several material plant combinations).

Regarding the runtime of locked and non-locked mode: The non-locked mode is always faster than the locked mode. You can really start to see the difference when X becomes high. However, if X is small, the difference should not be very significant. Non-locked mode is not 100% secure but the probability of errors is a lot smaller than for the old version of SDRQCR21. It can be used without problem in a system with low activity.

Regarding the impact of other jobs on the runtime: The runtime of SDRQCR21 might increase due to the locking process if there are many items which could not be locked. This is due to the retry logic. The probability of an item failing to get a lock increases with the ATP activity of the system. Therefore, a slightly longer runtime might occur if transactions V_V2 or VL10 are running in parallel to SDRQCR21. Because SDRQCR21 locks the documents it can influence the runtime of other jobs like VL10 or V_V2. They wait for or retry the documents that were locked by SDRQCR21. Experience shows that the runtime degradations should not be very significant.

The new version of SDRQCR21 can be used both during the week or weekend. It does not really matter and results do not depend anymore on the system activities.

It makes sense to use the compare mode and use the locked (flag "enqueue") and non-locked mode for tests in productive system. The result list with or without simulation mode is the same for simulation runs (see SAP note 998102 for additional information).

3.3.1.5.5 Case Study: Temporary Requirement Errors

What is a temporary or fake error? A temporary or fake error is an error that is not created by the system but by the execution of report SDRQCR21 in update mode in parallel to high peak system activity with sales orders/deliveries. Identified temporary errors get updated on the database which results in real errors; those errors would get detected and repaired in the next execution of the report, but at the same time SDRQCR21 could again generate new errors.

Background: In this case, the old version of SDRQCR21 was running daily in update mode - with a large number of errors in the spool lists, until the middle of month 10. Then, the execution of SDRQCR21 was switched to daily in simulation mode and update mode only on Sundays. As of month 12, the new version of report SDRQCR21 was scheduled daily in simulation mode at the same time as SDRQCR21 to compare the spool lists of both report versions. In the future, only the new version of SDRQCR21 will be scheduled.

The graph indicates the number of changes of Sales Orders and Deliveries measured during 5 weeks, confirming that Sundays are the days with the lowest business activity.

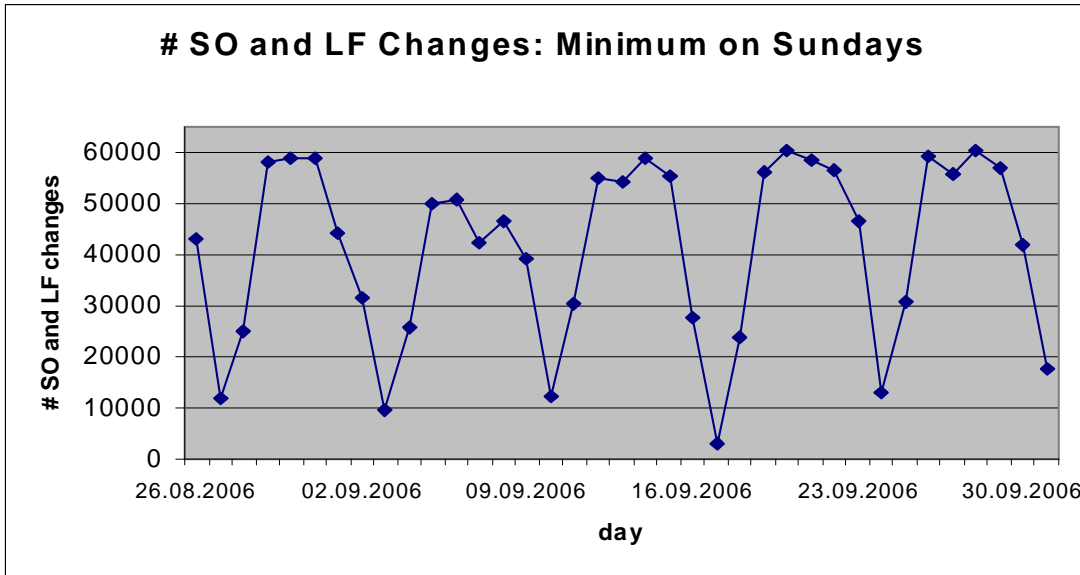


Figure 3.20: Document Changes by SDRQCR21

The next graphs display the number of changes during a Monday and a Sunday. However, it was not possible to identify or setup time windows with zero activity on Sundays due to the business requirements, but the Sundays were chosen as the only and best day to schedule SDRQCR21 in update mode.

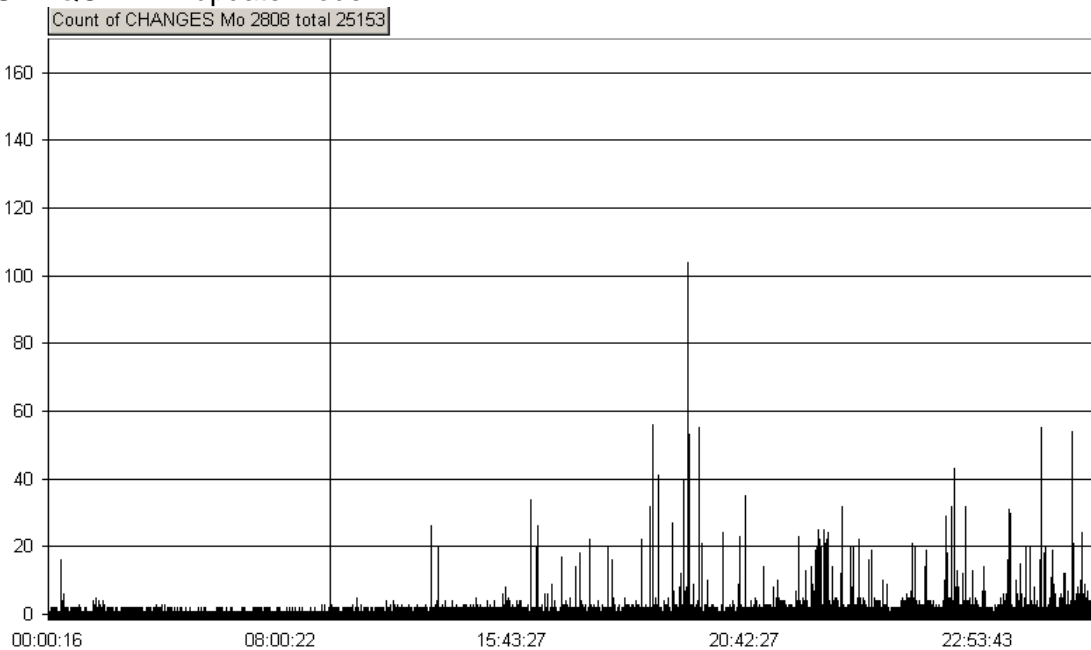


Figure 3.21: Document Changes by SDQR21 on a Monday

Best Practice: Data Consistency Check

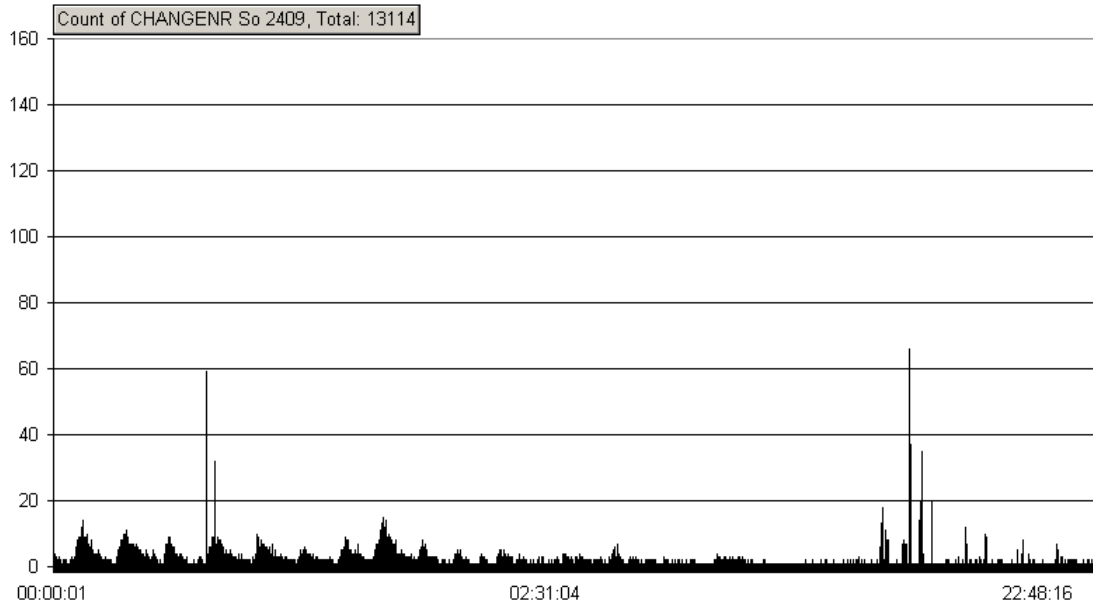


Figure 3.22: Document Changes by SDQR21 on a Sunday

The next comparison chart shows the number of errors found in the spool lists for the old and new version of report SDRQCR21. Evaluating the spool lists, it was found that about 97% of the errors detected by the old version of SDRQCR21 were temporary errors (due to production operation peaks on sales orders and deliveries while the report was running).

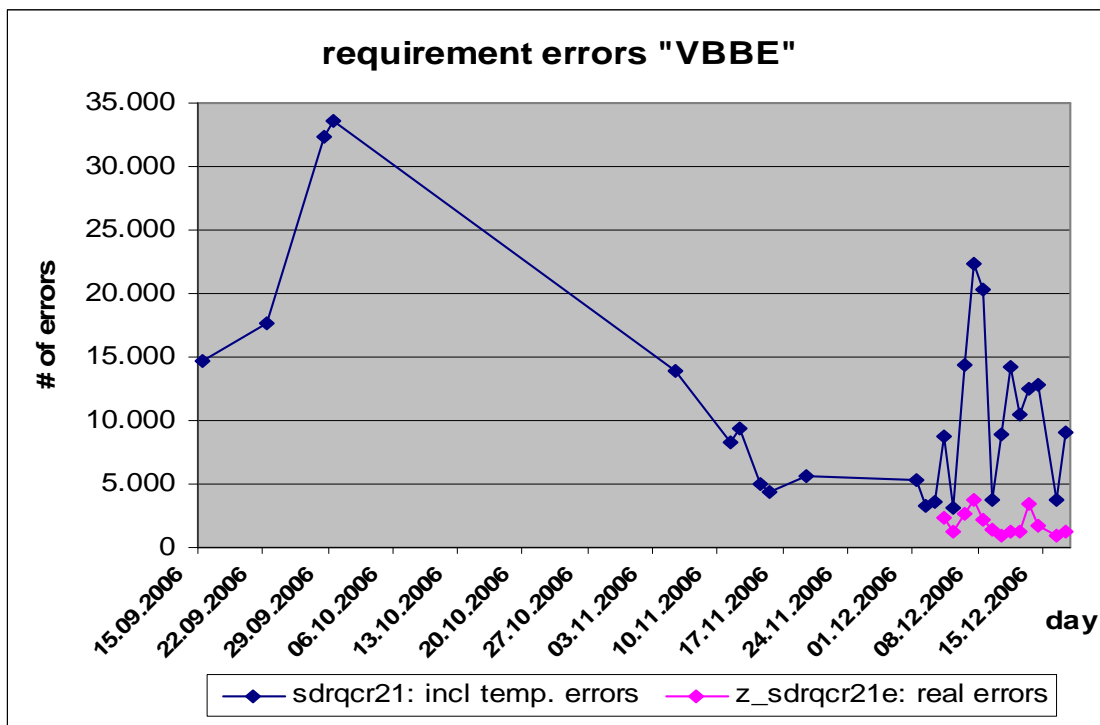


Figure 3.23: Comparison Between Old (sdrqcr21) and New (z_sdrqcr21e) Version of SDRQCR21

Note that from the middle of month 10 on, the old version of report SDRQCR21 was running in simulation mode only, except for Sundays. Running it daily in update mode, the number of errors in the spool lists would have raised further than in month 9 and the beginning of month 10.

Running the report SDRQCR21 in update mode during high peak system activity with sales orders/deliveries, the found temporary errors get updated on the database which results in real errors; those errors would get detected in the next execution of the report.

Best Practice: Data Consistency Check

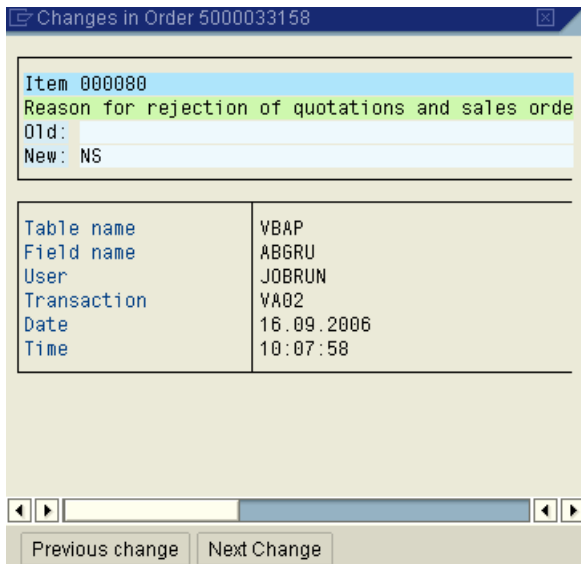


Figure 3.26: Detailed Display of Changes

In this example, a reason for rejection was set as the last change in the document, which may be the root cause of a requirement error. The probability that the requirement error is due to a customer modification or user exit is very high.

3.3.2 Tools for Processes Involving MM

There are several tools to identify and correct inconsistencies in Materials Management. The tools for customer usage are transaction MB5K (report RM07KO01) and the reports RM07MMFI and RM07MCHB. For special cases, the reports ZZWASTOR and ZTRAME are available as well.

The above mentioned reports – except ZZWASTOR – are intended to identify inconsistencies. The correction can be performed only by specially trained SAP consultants using analysis and correction reports from SAP note 32236. These reports shouldn't be used by customers as the results could be misleading without a deep technical understanding of the application.

SAP note 32236 provides a collection of check and correction reports used exclusively by SAP Support to analyze and correct stock inconsistencies. In general, only the stock and valuation tables are corrected, but not the document tables (material & FI document). The target of the correction is a consistent system; it does not matter during this correction whether the stocks are in-line with the physical reality. This is handled by a physical inventory check which should be performed afterwards.

These reports also cover the special stocks used by the Industry Solution "Aerospace & Defense" in the MRO process. Special reports are provided for stock inconsistencies in the Industry Solution "Oil & Gas" in SAP notes 67261, 212707, 378731, and 447714.

The SAP note 32236 contains the analysis report MBFIRST which gives a first overview of the system with important details, for example, valuation level, negative stocks allowed, Material Ledger usage, serial number usage, WM usage, archived material and FI documents, and so on.. This information is very important when starting the correction process.

3.3.2.1 General Inconsistencies within Materials Management

The standard transaction MB5K (report RM07KO01) is able to determine inconsistencies within Materials Management. The report should be planned at least once a week in a time period where as few as possible activities are carried out and can issue the following error messages:

- 011 Calculation number missing
- 045 Incorrect price
- 060 Negative stock not allowed
- 063 Material ledger record does not exist for current period
- 066 Material ledger currency record does not exist for current period
- 500 Actual quantity not equal to total of stocks
- 501 Actual quantity not equal to total of valuation segments

If the errors 060, 500 or 501 occur, make sure the reports from SAP note 32236 are in the production system. Then open a customer message under the component MM-IM-GF-INC and fill in the checklist from SAP note 76926. The further analysis and the corrections will be carried out by the SAP Support.

If the error 060 is issued, the report found negative stocks for materials although this has not been allowed. Furthermore the transaction compares the stock tables (MARD, MARC, MCHB, MKOL, and so on) and the valuation tables (MBEW, EBEW, QBEW, OBEW) (error 500) respectively the valuation segments for split valued materials (error 501).

In the case of errors 063 and 066, please contact the CO-PC-ACT support who will correct the inconsistencies. You can also check the consistency between Material Ledger and Materials Management with the transaction CKMC.

If the MB5K finds out that the calculation number is missing (MBEW-KALN1), you can regenerate it directly from the MB5K.

The message 045 "Incorrect price" means that the current price in the material master (MBEW-VERPR or MBEW-STPRS depending on the value in MBEW-VPRSV) differs from

the calculated price (total value of stock/total valuated stock – MBEW-SALK3/MBEW-LBKUM). These kinds of errors can be corrected with a simple price change in MR21. However, this is only the correction but it doesn't solve the problem. Based on our experience this error occurs mostly due to the following situation: the material has a quite low price, for example, 50 EUR/1000 PC and only small quantities are moved, for example 10 PC. Due to the rounding, for example, at Goods Issues, the price itself doesn't change but the calculated price will slowly differ from the moving average price or standard price. To solve the problem, the material master has to be changed, for example, the price unit should be changed.



The monitoring objects DCMMMB5K and DCMLCKMC exist to facilitate the regular monitoring of transaction MB5K and CKMC using the Business Process Monitoring functionality of the SAP Solution Manager.

3.3.2.2 Inconsistencies of Split Valuated Materials within MM

The report RM07MCHB helps to determine inconsistencies for split valuated materials. It checks the mother segment against the daughter segments, for example, whether the quantity of the mother segment and the sum of the daughter segments' quantity are the same.

If the report shows errors, please make sure the reports from SAP note 32236 are in the production system. Then, open a customer message on the component MM-IM-GF-INC and fill in the checklist from SAP note 76926. The further analysis and the corrections will be carried out by the SAP Support.

3.3.2.2.1 “Dead” Stock-in-Transit

The report ZTRAME can be used to identify “dead” stock-in-transit that can be removed by normal bookings. The full details about the report can be found in the SAP note 392205 where the report itself can be found, as well. The note also describes how to use the movement types 557/558.

3.3.2.3 Inconsistencies Due to Doubled Goods Issues for Deliveries

The report ZZWASTOR is introduced in SAP note 424414. The purpose of this report is to cancel material documents booked via delivery note by mistake twice. It simply cancels the material document that is not needed without updating the delivery note which is not possible by using the standard transaction VL09.

3.3.2.4 Inconsistencies (quantity) Between Material Documents and Stock Tables

The report MBSTOCK compares the material documents and the stock tables. During the run the table MSEG is read and the should-be stock is calculated and compared with the relevant stock table fields. You can find a list of stock tables later in this document. The report presumes that the stock calculated from the material documents is correct and therefore always suggests a correction in the stock tables. It is very important to know whether there are archived material documents as this influences the report. If there are archived documents, the results must be handled with great care! Therefore, the results should only be analyzed by experienced SAP consultants.



As the report reads the table MSEG, the run can be very performance intensive if the selection is not specific enough.

3.3.2.5 Inconsistencies (quantity) Between Valuation Tables and Stock Tables

The report MBQUANT compares the stock tables against the valuation tables. The list of the valuation tables can be found later in this document. The performance is not an issue in most of the cases. This report assumes that the stock tables have the correct information over the valuation tables thus it suggests a correction there.



It is very important to check the consistency between MM and FI before correcting with MBQUANT: If MM and FI are consistent and table MBEW is corrected with report MBQUANT, a MM-FI inconsistency is created.

Archived documents have no influence on this report. For very big selections MBMSSQUA has to be used.

3.3.2.6 Inconsistent Stock-in-Transit

MBTRAME deals with real stock-in-transits from intra-company stock transport orders. It checks & corrects the content of the field MARC-TRAME.

3.3.2.7 Purchasing

The report ZCORREKESBYGR corrects the quantities reduced (MRP) (EKES-DABMG) of confirmations with GR assignment. Further information can be found in SAP note 215072. The report RM06C020 finds and corrects inconsistencies with dependent requirements for subcontracting purchase order proposals. Further information can be found in SAP note 115899.

With the correction report ZKORVETV, the table VETVG (Delivery DueIndex) can be set up for an individual purchase order again. Further information can be found in SAP note 61148. SAP note 100690 provides correction reports ZKO* for inconsistencies of stock transport orders and stock transport scheduling agreements regarding quantities at schedule line level (EKET-GLMNG, EKET-WAMNG, EKET-WEMNG).

SAP note 202875 provides the correction report ZCORRDABMG regarding inconsistencies of EKET-DABMG and EKES-DABMG.

The report ZKOREKET checks and corrects if EKPO-MENGE of a scheduling agreement does not correspond to EKET-MENGE. Further information can be found in SAP note 67278.

3.3.2.8 Invoice Verification

The report ZREPMIR7 finds and corrects invoices without FI follow-up document or entry in the purchase order history. Further information can be found in SAP note 382797.

The report ZREP_MIRO_REMNG finds purchase order items with negative total invoiced quantity (REMNG). The correction can only be done by SAP. Further information can be found in SAP note 491074.

The report Z_FIND_DOUBLE_FI finds logistics invoices with incorrectly created duplicated FI documents. A correction of this issue can only be done by SAP. Further information can be found in SAP note 674190.

3.3.3 Inconsistencies between MM and FI

3.3.3.1 Inconsistencies between MM and FI (Valuation Tables vs. FI Stock Accounts)

Report RM07MMFI can identify inconsistencies between Materials Management and Financial Accounting. The report compares the valuation tables (MBEW, EBEW, QBEW, OBEW) directly with the stock account balances from table GLT0 with the help of the account determination (table T030). The report should be used instead of the transaction MB5L (report RM07MBST) as it is much less performance intensive. The report should be started during a timeframe where no bookings take place to get a stable result. The report should be

Best Practice: Data Consistency Check

run 3-5 times to eliminate errors resulting from bookings during the analysis if the report cannot be executed during such a time window.



Make sure the reports from SAP note 32236 are in the production system if the report shows stable results and has identified inconsistencies. Open a customer message on the component MM-IM-GF-INC afterwards and fill in the checklist from SAP note 76926. The further analysis and the corrections will be carried out by the SAP Support.



The Business Process Monitoring object DCMFIDIF is available to facilitate the regular monitoring of inconsistencies.

The reports provided by SAP note 32236 should only be carried out by experienced SAP consultants as the results of these reports need some interpretation. SAP note 34440 describes the correction procedure applied by the SAP consultants for customers.

3.3.3.2 List of Relevant Stock and Valuation Tables

Below, you can find a list of the relevant valuation and stock tables. The IS-ADEC tables are mentioned as they are covered by SAP note 32236, the special IS-OIL tables are not, as they are handled separately by special reports. The tables ending with H are the history tables. Further information on history tables can be found in SAP notes 171465 and 193554. History tables should be corrected only in exceptional cases. For details see SAP note 34440.

Important valuation tables:

- MBEW Material Valuation
- MBEWH
- EBEW Sales Order Stock Valuation
- EBEWH
- QBEW Project Stock Valuation
- QBEWH
- OBEW Valuated Stock with Subcontractor
- OBEWH

Important stock tables:

- MARC Plant Data for Material
- MARCH
- MARD Storage Location Data for Material
- MARDH
- MSSA Orders on hand Total
- MSSAH
- MSKA Sales Order Stock
- MSKAH
- MKOL Special stock from vendor (vendor consignment, RTP)
- MKOLH
- MSSL Subcontracting stock at vendor Total
- MSLB Subcontracting stock at vendor
- MSLBH
- MSSQ Project Stock Total
- MSSQH
- MSPR Project Stock
- MSPRH
- MSKU Special stock at customer (customer consignment, RTP)
- MSKUH

Important stock tables available only for Discrete Industries

- MCSD Customer Stock
- MCSDH
- MCSS Total Customer Stock
- MCSSH
- MSCD Customer stock with vendor
- MSCDH
- MSCS Customer stock with vendor - Total
- MSFD Sales Order Stock with Vendor
- MSFDH
- MSFS Sales Order Stock with Vendor - Total
- MSID Vendor Stock with Vendor
- MSIDH
- MSIS Vendor Stock with Vendor – Total
- MSRD Project Stock with Vendor
- MSRDH
- MSRS Project Stock with Vendor – Total

3.3.4 Tools for Processes Involving WM

3.3.4.1 Stock Comparison Between WM and IM

Transaction LX23 investigates and corrects stock differences between inventory management and warehouse management. LX23 can be used both for a centralized and for a decentralized scenario. LX23 runs on the warehouse management system in the case of a decentralized scenario and the inventory management stocks are determined by RFC from the central ERP system. The actual differences between the two inventory management levels can be compared via an automatic physical inventory where the stock comparison report first reads all IM stocks including special stocks and reads and summarizes the WM stocks in a second step. The individual stocks are listed and the difference is calculated. An automated correction can be executed if you select the 'Clear differences' check box, but you should bear in mind that the system always assumes that the WM system is the leading system. Correspondingly, in the event of differences, the IM stock is adjusted to the WM stock and NOT vice versa. This correction is performed by a generated batch input session for transaction MI10 which performs an automatic physical inventory which adjusts the IM stocks to the WM stocks. No batch input session is generated in the case of a decentral warehouse scenario, but IDocs of message type MBGMCR are created instead and sent to the central ERP system.

3.3.4.1.1 Restrictions of Transaction LX23

Active handling unit management: Transaction LX23 cannot be used for correction of HU-managed storage locations. However, for storage locations with HU management, you can use transaction LX23 to display stock differences between WM and IM. HU managed stocks cannot be analyzed at all. SAP note 660009 describes how to proceed for HU managed storage locations in case of stock differences for one of the three levels IM, WM and HU.

Materials with serial number requirement: Clearing the differences via a batch input session which has been created by LX23 is not possible for materials with serial number requirement. More information and how to proceed in this case can be found in SAP note 484788.

3.3.4.1.2 How to Handle Temporary Differences with Transaction LX23

You have to keep in mind that temporary stock inconsistencies between warehouse management and inventory management, will always exist in the case of a decentralized scenario when analyzing stock differences due to the asynchronous update between the systems. Before running LX23 in a decentralized scenario you need to ensure that all IDocs

of message type SHP_OBDLV_CONFIRM_DECENTRAL, SHP_IBDLV_CONFIRM_DECENTRAL and MBGMCR are transferred and processed successfully from the decentralized warehouse management system to the central ERP system. Moreover, ensure for both cases - running LX23 for a centralized and for a decentralized scenario - that stock changing postings take place neither on IM level nor on WM level.

The report RLTREORG2 detects inconsistencies either between the quants and the storage bins or between the quants and the storage units, assuming that the quant data is correct. Further information can be found in SAP note 607226.

3.3.4.2 Correction of Picking and WM Status in Deliveries

The report RVDELSTA checks whether the status in a delivery is incorrect according to related system information and must be determined again. This is also done for the picking and WM status. More information can be found in SAP note 506510.

3.3.5 Tools for Processes Involving PP

3.3.5.1 Introduction:

If pure R/3 production planning is used (no other system keeping the data redundant) external consistency problems are not an issue. In general, it is possible to record confirmations with legacy systems for production control. The PP module allows downloading operations of orders into external systems. Beforehand, the work centers have to be transferred to the external system as well. The operations are then confirmed in the legacy system and the confirmations are uploaded into PP subsequently. Upload and download are performed with the transactions CI42 to CI45 within PP. The following recommendations only refer to PP and do not consider if an SCM system or other external systems are integrated to R/3 PP.

3.3.5.2 Erroneous goods movements

The stock situation is not updated whenever goods movements fail. Following requirements, calculations (in R/3 or SCM) will work with incorrect stock levels and might therefore not generate necessary receipt elements or unnecessary receipt elements. This may lead to a production standstill if necessary materials are not available and ATP checks in productions and sales orders will produce wrong results. The actual costs that are calculated for the production order are not correct if the goods movements to the production order are not booked correctly. The costs of production are in such a case not correctly reflected, which might have severe business consequences. Incomplete cost postings result in the end in customers paying not enough.

To reduce such impacts and as a preparation for a reliable period end closing, the following reports should be carried out regularly as standard activities:

3.3.5.2.1 Failed Goods Movements in PP-SFC

Report CORUAFFW/CORUAFWP (transaction COGI) exists for identifying and correcting the reason for erroneous goods movements and subsequent post processing. The stock/requirements situation in APO is updated accordingly! Compared to report CORUAFFW, CORUAFWP is predestined for batch processing having interesting additional features for batch processing as parallel processing, parameters for maximal number of locks, maximal number of document positions, and so on.



The regular monitoring of failed goods movements can be facilitated with the Business Process Monitoring in SAP Solution Manager using monitoring object KPCONF01.

3.3.5.2.2 Failed Cost Postings in PP-SFC

Report CORUCOFC (transaction COFC) is used for identifying and correcting the reason of failed cost postings and subsequent post processing. See SAP note 565370 "FAQ: Reprocessing incorrect actual costs (COFC)" for details. Such costs are only of interest for the R/3 system and are not relevant for SCM nor do they have an impact on APO PP/DS.

3.3.5.2.3 Failed Goods Movements from Repetitive Manufacturing

Reprocessing of failed records from repetitive manufacturing may be processed online with COGI or in the background using either the report RMSEI13 or COGI with batch input. The reports CORUAFWP and CORUPROC are not working for repetitive manufacturing. The tools cannot be clearly classified as consistency checks or housekeeping jobs. For completeness, the reports when using decoupling are mentioned:

- Postprocessing of goods movements in PP-SFC (only relevant when decoupling is activated): report CORUPROC (transaction CO1P)
- Postprocessing of Cost in PP-SFC (only relevant when decoupling is activated): report CORUPROC (transaction CO1P)

Both jobs should be regular business steps when using decoupling and are not related to failed postings.

3.3.5.3 Important Notes for PP Reprocessing:

Component	SAP Note	Description
PP-SFC-EXE-CON	565370	FAQ: Reprocessing incorrect actual costs (COFC)
PP-SFC-EXE-GM	544520	FAQ: Picking
PP-SFC-EXE-GM	540392	FAQ: Automatic goods movements
MM-IM-RS-RS	519580	FAQ: Reservations

Table 3.10: Important Notes for PP Processing

3.3.6 Tools for Processes Involving PS

3.3.6.1 Introduction

Usually, only a pure R/3 project system is used. As a result, there are generally no external consistency problems, as no other system keeps the data redundantly. Therefore, a best practice for consistency check in PS is not available. Nevertheless, some generic system internal checks should be carried out under special circumstances. On the application side, networks and orders, especially in big projects, should get charged, closed and then where applicable get archived. As preparation for a reliable period end closing, the actions described in the following sections should be carried out to ensure a consistent state.

3.3.6.2 Failed Postings from Confirmations

3.3.6.2.1 Failed Automatic Goods Movements

As a general impact, erroneous goods movements that are not post processed may lead to a mismatch of the stock/requirements situation between system and reality which might falsify ATP checks and any requirements planning for material components. In order to resolve these issues, transaction COGI (report CORUAFWP) should be used regularly for post processing of erroneous goods movements. (This also applies to PP and PM).



The regular monitoring of failed goods movements can be facilitated with the Business Process Monitoring in SAP Solution Manager using monitoring object KPCONF01.

3.3.6.2.2 Failed Postings of Costs

Impact of erroneous cost postings: Erroneous cost postings lead to missing costing records, which for instance could be missing for resource related billing. That would cause the end customer to be billed less. You should run transaction COFC (report CORUCOFC) for post processing of erroneous cost calculations on a regular basis to decrease the impact of such situations. This also applies to PP and PM.

3.3.6.2.3 Failed Non-Interactive Confirmations

Non-interactive confirmations can occur from the transfer of postings from the cross application time sheet (CATS) or from posting confirmations via the PDC data interface - communication channel KK4. Consequently, the actual hours, costs, and automatic goods movements are not posted with all consequences for subsequent processing like incorrect stock, missing costs, too little billing, too little work in progress, and so on.

To decrease the impact of such failed confirmations, you should run transaction CN30 (CORUPROC) for post processing of the error pools for non interactive confirmations on a regular basis. This also applies to PP - transaction CO10N and PM - transaction IW46. These three steps are also described in the SAP note 570471.

3.3.6.3 Value Category Customizing

The values for costs/revenues, payments/receipts are not reliable anymore in the Hierarchy Reports (for example, transaction CJE0), the Structure Overview (transaction CNS41 / CN41) and the Project Planning Board (transaction CJ2B) if the customizing of the value categories is incorrect. Strange effects like disappearing costs/revenues, mixing of costs/revenues, or costs appearing as revenues are possible without having this customizing set up completely and correctly. The completeness of the value category customizing should be checked in consistency check transaction CJVC according to SAP note 676123. Please adjust your customizing following the output of the consistency check. After having corrected everything, the project information database must be rebuilt by transaction CJEN (only, if really something has been adjusted in this customizing).

3.3.6.3.1 Special Topics:

Correction reports for wrong commitments in PS reports are provided by SAP note 152571. A general FAQ regarding inconsistencies in the area of the Project System is provided in SAP note 678522.

3.3.6.4 Overview of SAP Notes for PS Consistency:

Component	SAP Note	Description
PS-CON	570471	FAQ 2: Confirmations in the Project System
PS-IS-ACC-HIE	676123	Project Information Database / Value Category Customizing
PS-IS-ACC-HIE	678522	FAQ 2: Data inconsistencies in the Project System

Table 3.11: SAP Notes for PS Consistency

3.3.7 Tools to Check the Logistics Information System

Report RMCSUTIC provided by SAP note 382977 can be used to verify the customizing as well as the definition of the information structure for customer made info structures. As not all of the control table entries delivered within the SAP standard are needed for an error-free update, the system displays errors that are not errors in the case of a check of the update rules for SAP standard infostructures S000 to S500.

3.4 Tools not Depending on a Particular System

As soon as more than one system is involved, the consistency check has to cover the interface related parts. In the SAP environment ALE (IDocs) are frequently used to transmit data between systems, and here the following tools are relevant.

On the sender side, the following tools/checks have to be done: Are the interfacing documents already out of the system, or are they still in outbound processing?

With transaction BD87, you can check the status of IDocs and the tRFC communication status. Make sure that the interface data has completely left your sending system. For all other interface technologies you have to use the available tools that the interface and used middleware offer you. If you are using the SAP Exchange Infrastructure as your interface middleware, you have to check the status of your XI messages in the XI Adapter Framework as well as on the XI Integration Server for complete processing (for example, using the Runtime Workbench – Message Monitoring). If you are for example transferring your interface data via files, you have to make sure that all files are transferred completely.

On the receiver side, you have to check that the relevant interface data has been processed completely and correctly. As long as data is still in status to be processed or in error status on the receiver side, the executed consistency check report will not provide correct data. For SAP systems with incoming IDocs, you use the same transaction BD87 to check the status of your inbound IDocs. For SAP systems with incoming XI messages via ABAP Proxy communication you use transaction SXMB_MONI to check the processing status of inbound messages. For non-SAP systems you have to check for tools and processes, how to make sure that all relevant interface data for the processing area has been completely and successfully processed.

3.4.1 Tools for ALE Consistency Check

If you want to check the complete status of your transferred IDocs in the sending system, you have to setup ALE Audit and ALE tracing.

For more information on this topic, please check the documentation in SAP Online Help under:

http://help.sap.com/saphelp_erp2005vp/helpdata/en/0b/2a655d507d11d18ee90000e8366fc2/frameset.htm

3.4.2 Tools for ALE Recovery

In distributed environments, you can also recover an SAP system following a system failure caused by a database error. Using backup and point-in-time recovery, you can reset the database to the status at the time the system failed and continue working with it.

In this case, transactions may still have taken place after the time the system is reset to, which are subsequently lost and have to be carried out again. External interactions (for example, ALE, EDI, mail, fax, telex), possibly carried out under a different key, are duplicated and require special processing. For example, if a process sent a message to another system and started another process here during the time between when the error occurred and when it was discovered on the recovered system, the data related to this process will no longer exist on the recovered system. The results of the second process carried out on the receiving system still exist in this system. The inconsistent data in the two systems can be put right by canceling the data resulting from the communication in the receiving system.

In point-in-time recoveries, certain documents and messages must be canceled in the communication partners' systems and messages sent earlier from the receiving system must be sent again. You have to determine all the actions that need to be carried out in the recovery process. The ALE Recovery tools help you to do this.

For Tools and more see SAP Online Help and topic "ALE Recovery for Data Consistency" under:

http://help.sap.com/saphelp_erp2005vp/helpdata/en/26/29d829213e11d2a5710060087832f8/frameset.htm

3.4.3 Generic Consistency Check for Two Linked Tables in One System

3.4.3.1 Introduction

Very often, you have to verify data between two different tables which are linked. For example you could be interested in identifying line items without corresponding header information (for example, sales order items VBAP without corresponding sales order header data in VBAK). Or during data load/creation into a new system all materials of a certain material type (for example, HAWA) should have a specific entry in one of the material master views (for example, MARC or MVKE). Usually, you would have to write a program or a query to find this kind of data. Within the service software portal transaction ST13, a new report /SSA/Z>CD1 has been delivered that can answer all these questions.

The report starts with a selection screen where you can enter the table names you want to investigate as well as field names and restrictions. You can also differentiate what kind of check you would like to execute between the linked table (e.g. whether you would like to see entries where a field has an undesired content or whether you would like to identify missing fields).

The report creates an appropriate dynamic SQL-statement for the entered table names, use case, and field restrictions. Once the selection has finished the resulting list is displayed together with the used SQL-statement in the ALV. As the created SELECT-statement may not be supported by an appropriate index, the run time of the report may be very long in certain cases.



The results of regular executions of this report may be monitored from SAP Solution Manager with the monitoring object "DCGEN001".

3.4.3.2 Use Case 1: Completely Missing Parent Table Entries

This scenario of the report provides the possibility to list all entries of a dependent table where no corresponding entries exist in a leading table. A typical usage scenario would be line items without corresponding header information (for example, sales items in table VBAP without header information in table VBAK). This information could be of interest after a data base crash or after a report accidentally has deleted some data. In report /SSA/Z>CD1, you need to enter the leading table as "Table 2" and the dependent table in field "Table 1". In addition, you need to specify the foreign key relation between the two tables in the area "Define join condition between tables". This case is activated by setting the flag "missing entries in table2". A typical example can be found in Figure 3.27: Finding Missing Header Information where the range is restricted to certain document numbers using a select-option.

Best Practice: Data Consistency Check

Define tables	
Table 1	Table 2
<input type="text" value="VBAP"/>	<input type="text" value="VBAK"/>

Define join condition between tables			
fieldname 1	<input type="radio"/> <input type="radio"/>	fieldname 2	value restriction
<input type="text" value="VBELN"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="text" value="VBELN"/>	<input type="text" value="0000010000"/> bis <input type="text" value="0000100000"/>
and	<input type="radio"/> <input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
and	<input type="radio"/> <input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>

Define additional where-clause			
Tbl 1	Tbl 2	field name	value restriction
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="text"/>

Define operation	
missing entries in table2	<input checked="" type="radio"/>
inconsistent field content	<input type="radio"/>

Define output	
Use ALV?	<input checked="" type="checkbox"/>
Enter field names	<input type="text" value="VBELN"/> <input type="text" value="POSNR"/> <input type="text"/>

Figure 3.27: Finding Missing Header Information

After starting, the report creates an appropriate SQL-statement for the entered table names, use case, and field restrictions and displays the resulting list in the ALV (see Figure 3.28: Result Screen of the Generic Report). The upper part of the screen lists the total number of hits (inconsistencies) found while the middle part contains further details which can be used for detailed investigations or correction. The used SQL-statement is displayed in the lower part of the screen. In the example we see that we have an item of sales order 47870 in the database while no header information exists for this document number.

Inconsistency check results

Total number of inconsistencies found: 1

Verkaufsb.	Pos
47870	10

```

Used select statement:
SELECT VBELN POSNR
FROM VBAP AS A
INTO CORRESPONDING FIELDS OF TABLE fill_tab
WHERE NOT EXISTS ( SELECT * FROM VBAK WHERE
b~VBELN = a~VBELN
AND b~VBELN IN so_jn1
)
AND
a~VBELN IN so_jn1

```

Figure 3.28: Result Screen of the Generic Report

As the created SELECT-statement may not be supported by an appropriate index, the run time of the report may be very long in certain cases.

3.4.3.3 Use Case 2: Incorrect Field Content

While the first option has shown table entries where no corresponding information was found at all in a leading table, sometimes it is also of interest whether two tables contain a correct field content. A typical example would be to verify the quality of master data after a cut-over or to check for outdated information after a customizing change. Here, the leading criteria may be contained in another table than the actual field containing the value (for example, do all materials of type HAWA (table MARA) have the intended MRP type in table MARC?). In order to use the report for this use case, both tables need to be specified. In addition, the link between the tables and all selection criteria that define the condition to be checked have to be entered. In the example of comparing MARC and MARA this would be the material number (MATNR) as a link between the two tables in the screen area "Define join condition tables" and the criteria MTART = HAWA to restrict to the desired material types and the MRP-type to be used by this material (see Figure 3.29: Example to Find Table Entries not Having the Desired Field Content). The other criteria are only entered to restrict the hit list in the given example.

Best Practice: Data Consistency Check

Define tables

Table 1 <input style="width: 90%;" type="text" value="MARC"/>	Table 2 <input style="width: 90%;" type="text" value="MARA"/>
--	--

Define join condition between tables

	fieldname 1 <input style="width: 90%;" type="text" value="MATNR"/>	<input checked="" type="radio"/> <input type="radio"/>	fieldname 2 <input style="width: 90%;" type="text" value="MATNR"/>	value restriction <input style="width: 90%;" type="text"/> bis
and	<input style="width: 90%;" type="text"/>	<input checked="" type="radio"/> <input type="radio"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/> bis
and	<input style="width: 90%;" type="text"/>	<input checked="" type="radio"/> <input type="radio"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/> bis

Define additional where-clause

Tbl 1	Tbl 2	field name	value restriction
<input checked="" type="radio"/>	<input type="radio"/>	<input style="width: 90%;" type="text" value="MATNR"/>	<input style="width: 90%;" type="text" value="000000001998"/> bis
<input checked="" type="radio"/>	<input type="radio"/>	<input style="width: 90%;" type="text" value="WERKS"/>	<input style="width: 90%;" type="text" value="1000"/> bis
<input type="radio"/>	<input checked="" type="radio"/>	<input style="width: 90%;" type="text" value="MTART"/>	<input style="width: 90%;" type="text" value="HAWA"/> bis
<input checked="" type="radio"/>	<input type="radio"/>	<input style="width: 90%;" type="text" value="DISMM"/>	<input style="width: 90%;" type="text" value="PD"/> bis
<input checked="" type="radio"/>	<input type="radio"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/> bis
<input checked="" type="radio"/>	<input type="radio"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/> bis

Define operation

missing entries in table2

inconsistent field content

Define output

Use ALV?

Enter field names

<input style="width: 90%;" type="text" value="MATNR"/>
<input style="width: 90%;" type="text" value="WERKS"/>
<input style="width: 90%;" type="text" value="DISMM"/>

Figure 3.29: Example to Find Table Entries not Having the Desired Field Content

While we have checked so far individual field content with a specified field content, you may very often also be interested in a consistency check between field contents of two linked tables independent of the value. An example would be verification between customizing settings and the associated stored values in the movement data. In this case, the only required information is whether the values correspond, but not actually which field content it is supposed to be. This type of check can be performed with the report as well. An example for a selection variant is displayed in Figure 3.30: Compare Two Linked Tables without Knowing the Intended Field Content.

Best Practice: Data Consistency Check

Table 1		Table 2	
VBAP		TVAP	

Define join condition between tables			
fieldname 1		fieldname 2	value restriction
PSTYV	<input checked="" type="radio"/> <input type="radio"/>	PSTYV	<input type="text"/> bis <input type="button" value="→"/>
and ERLRE	<input type="radio"/> <input checked="" type="radio"/>	ERLRE	<input type="text"/> bis <input type="button" value="→"/>
and	<input checked="" type="radio"/> <input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>

Define additional where-clause			
Tbl 1	Tbl 2	field name	value restriction
<input checked="" type="radio"/>	<input type="radio"/>	VBELN	0000001000 bis 0000010000 <input type="button" value="→"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>
<input checked="" type="radio"/>	<input type="radio"/>	<input type="text"/>	<input type="text"/> bis <input type="button" value="→"/>

Define operation	
missing entries in table2	<input type="radio"/>
inconsistent field content	<input checked="" type="radio"/>

Define output	
Use ALV?	<input checked="" type="checkbox"/>

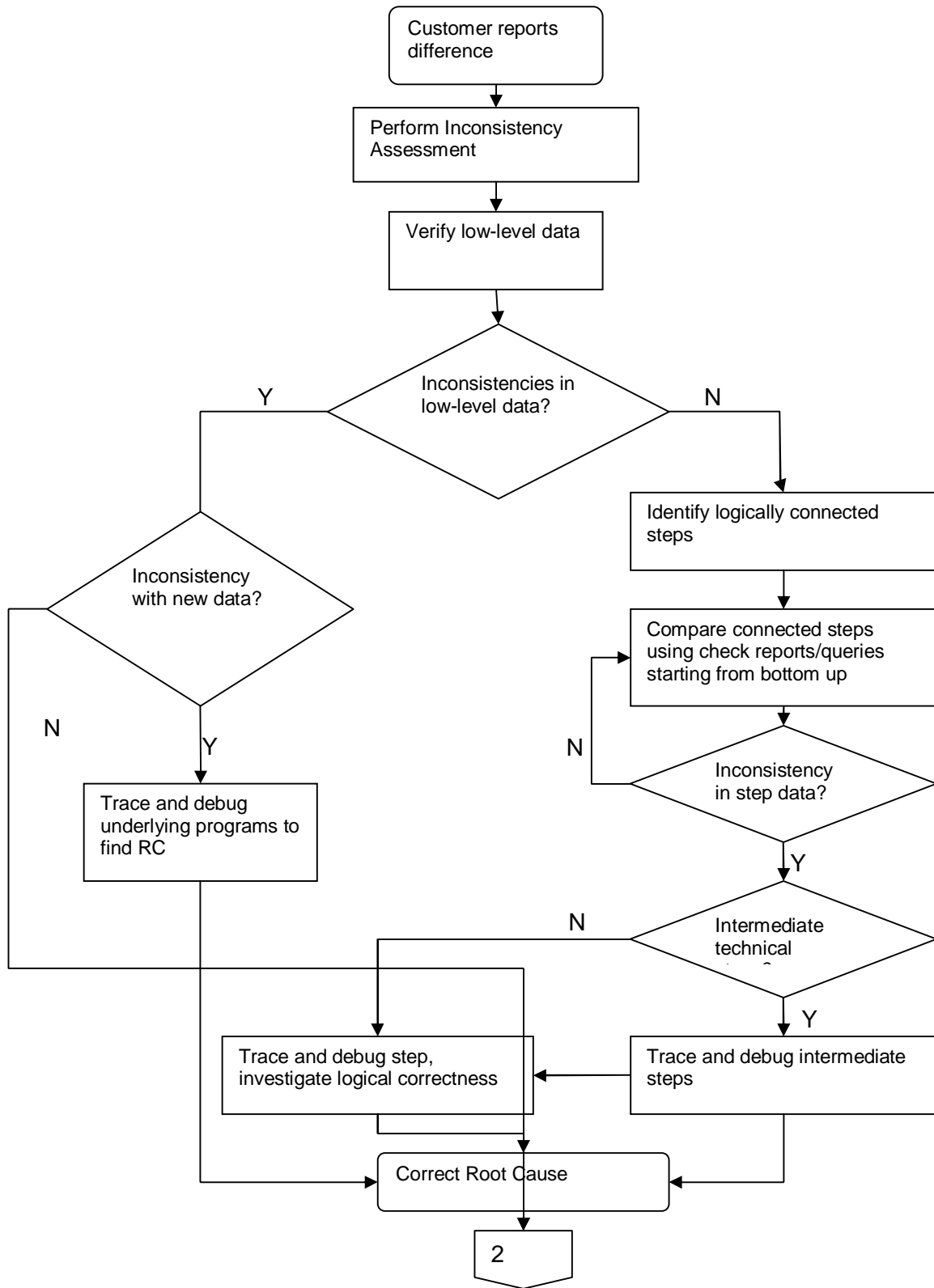
Figure 3.30: Compare Two Linked Tables without Knowing the Intended Field Content

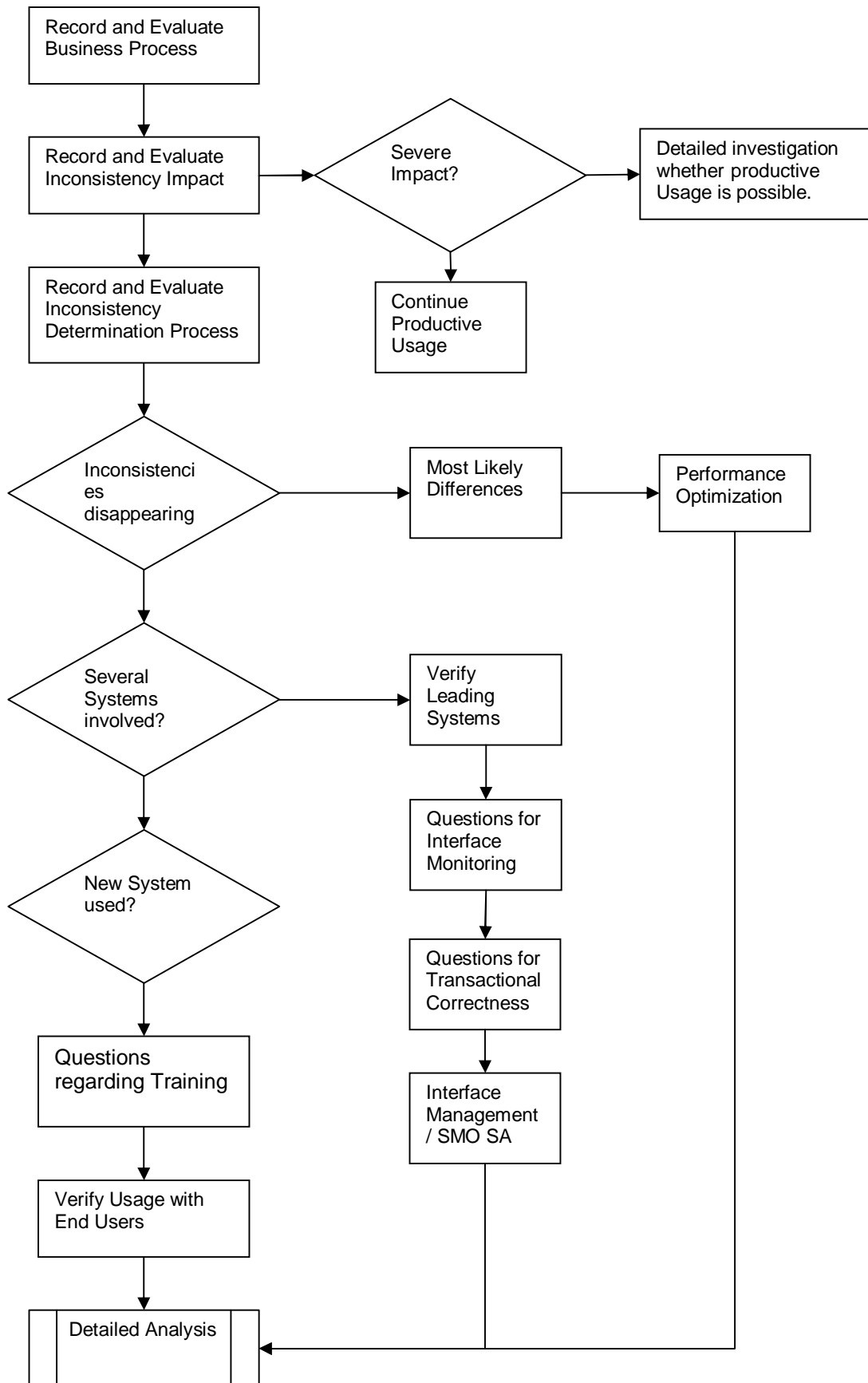
3.4.4 The Generic External Compare

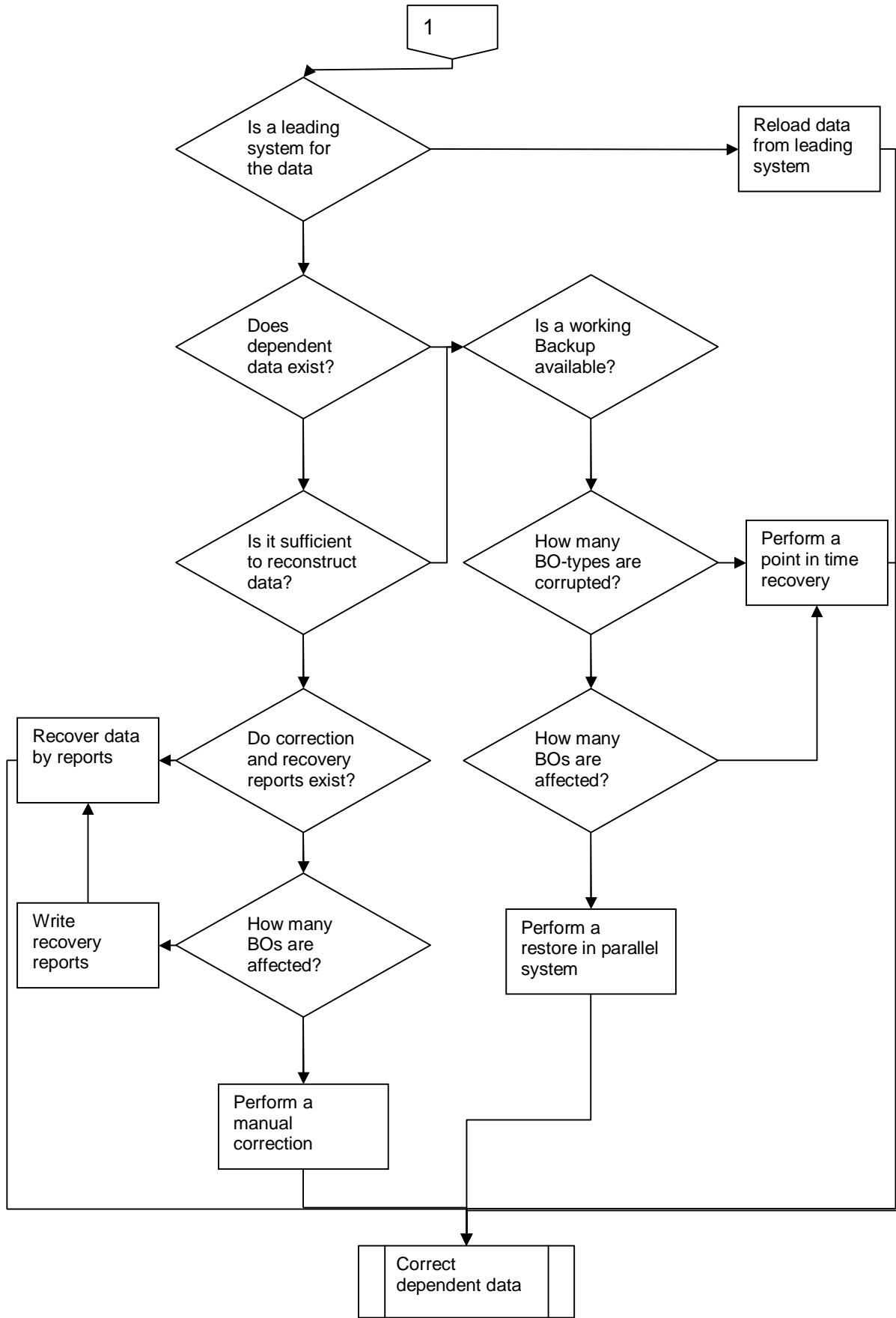
Another generic tool to verify data consistency where no standard report is available is the “Generic External Compare”. This tool set facilitates the extraction of data from two systems and the following comparison. For details see the document “Documentation for the Generic External Compare” available in the service marketplace under <http://service.sap.com/bpm>.

4. Appendix

4.1 General Roadmap for Analysis







4.2 Dos and don'ts for Data Consistency

This section provides the most important data consistency rules for the areas monitoring and error handling, program design, and end user training.

Monitoring for inconsistencies should always include:

- Interface Monitoring
- Business Process Monitoring
- Update Termination Monitoring

Error correction and investigation should:

- Be performed soon after error detection
- Consider dependencies and data actualization before updating/reprocessing data
- Perform corrections only if legally allowed
- Never run correction reports for all documents without prior analysis
- Ensure that no object changes are executed while running correction/check reports
- Not perform data cleansing for temporary inconsistencies
- Document number range gaps for later reference
- Bear in mind that not everything possible is also allowed

Programs and enhancements should:

- Follow the all-or-nothing principle
- Consider implicit and explicit commits
- Lock data to be changed using the SAP enqueue feature
- Provide restartability features
- Provide logging information

To prevent inconsistencies by data changes and to facilitate root cause analysis:

- Customizing in the productive client should not be changeable
- Change the maintenance status for tables that need to be maintained regularly only for exceptional cases
- Change management for customizing should be used
- Customizing for existing data should never be changed. If you do so, consider the impact on existing documents and use appropriate correction reports.
- Changes of field contents using the debugger should be disallowed
- Transports should never be made into a running system

End users should

- Be trained in the handling of exceptions
- Perform business process steps (and their physical pendant) completely and in the correct order, for example, period end closing and period end closing reconciliation
- Be aware of dependencies
- Incorporate a data model of dependencies in the training

Business processes should:

- Have a fall-back strategy in case of system and application down situations
- Be designed rating logical correctness and clarity above possibilities of system usage
- Be designed with a clear leading system having unique dependencies for each business object

4.3 Advantages and Disadvantages of Different Recovery Methods

There are different possibilities to recover data. This appendix lists advantages and disadvantages of individual data recovery methods and provides guidelines for deciding which method should be considered first. Please bear in mind that even though one recovery method may seem best from an issue point of view (complexity, quantity), another one may be the one to be used, as under certain circumstances the requirements for the preferable method may not be met.

No	Recovery Method	Advantages	Disadvantages	Requirements
1	Complete restore of backup (point in time recovery)	<ul style="list-style-type: none"> • Consistent state of complete system • Pure technical procedure to correct complex inconsistencies 	<ul style="list-style-type: none"> • In distributed systems only manageable with high effort • Data after "point-in-time" needs to be reconstructed (for example, from printouts and so on) • During restore and for some follow-up activities no productive usage of system possible 	<ul style="list-style-type: none"> • Time of first occurrence of failure needs to be known • Correct and consistent backup needs to be available • Persistence layer must be able to perform a point-in-time recovery
2	Restore in a sandbox-system and load of data by RFC	<ul style="list-style-type: none"> • Recovery of individual tables and table entries possible • A productive usage of system is in general possible 	<ul style="list-style-type: none"> • Data after "point-in-time" needs to be reconstructed (for example, from printouts and so on) • Programs to load data between systems need to be developed individually 	<ul style="list-style-type: none"> • Same requirements as "complete restore of backup" plus second system with enough resources for processing and storage
3	Reload data from a leading system	<ul style="list-style-type: none"> • A productive usage of system is in general possible • Simple technical method utilizing programs which are usually available already 	<ul style="list-style-type: none"> • Dependent data needs to be reconstructed from loaded data • Usually only for master data available 	<ul style="list-style-type: none"> • Leading system containing correct data
4	Recovery from dependent data and systems	<ul style="list-style-type: none"> • A productive usage of system is in general possible • Many tools and reports available within SAP standard environment (DIMA, CIF-Reports, LX23,...) 	<ul style="list-style-type: none"> • Often, correction reports need to be developed individually 	<ul style="list-style-type: none"> • Relationship between data to be recovered and underlying data
5	Manual correction of data (debug mode, SE16, application transactions)	<ul style="list-style-type: none"> • Fast correction of single corrupted data • A productive 		<ul style="list-style-type: none"> • Manageable quantity and complexity • No existing

		usage of system is possible		correction reports
--	--	-----------------------------	--	--------------------

Table 4.1: Comparison of Recovery Methods

Taking into account the general pros and cons of the recovery methods, the deciding factors for the individual methods are:

- Methods 1 and 2 are the choice if data is contained in another system but is also affected there, for example, the dependent data is already corrupted or does not exist at all.
- Method 1 should only be attempted if the data cannot be reconstructed by other means because too many business objects have been destroyed and no relationships usable for reconstruction exist.
- Method 2 should be preferred over method 1 if the data cannot be restored by relationships or dependent systems but only a few tables or business objects are affected (in contrast to a widespread failure where method 1 should be preferred). It is advisable when the complexity of the data to be restored is manageable.
- Methods 3 and 4 should be preferred if data is contained in another system or can be reconstructed from further existing data as both allow productive usage of the system in most cases.
- Method 5 should only be attempted if only a very small amount of data is affected and no complex relationships exist.
- When the issue is very complex, meaning that several objects with extended relationships are affected, a reconstruction may logically be impossible or too time consuming. In this case, methods 1, 2 or 3 are optimal.
- Method 1 is not applicable if the time between error detection and first occurrence (point-in-time for which data needs to be restored) is very long as too much data needs to be reconstructed from external sources like printouts, memorization, and so on.

If we look at these observations and recommendations, the following picture emerges with respect to quantity and complexity:

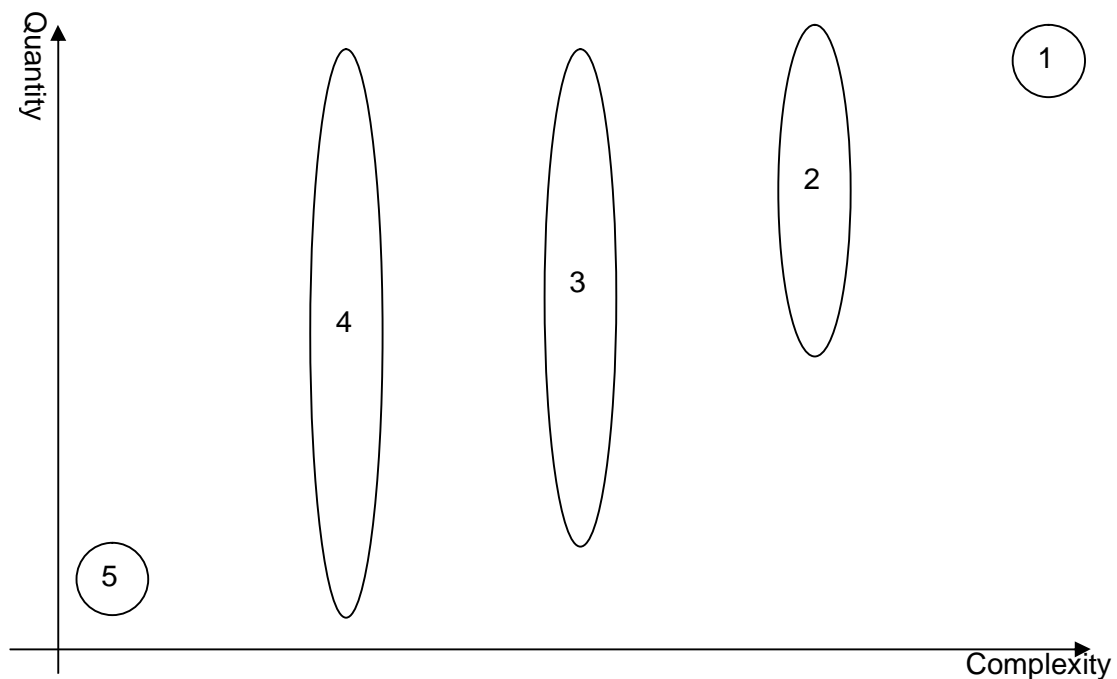


Figure 4.1: Quantity and Complexity of Different Recovery Methods

4.4 Further Information

4.4.1 Related information

Important information regarding monitoring of data consistency with the SAP Solution Manager can be found in the document “Setup Guide – Data Consistency Monitoring” available in the Media Library under <http://service.sap.com/bpm>.

Documentation of a general tool to compare data in different systems with the help of SAP Solution Manager can be found there as well.

If you need more information regarding Continuity Concepts refer to the documents “Emergency Handling for Recovery of SAP System Landscapes” and “Business Continuity Management for SAP System Landscapes”, which you can find in the service marketplace under <https://service.sap.com/solutionmanagerbp>

4.4.2 Troubleshooting

If executing this Best Practice did not produce the desired results

1. Search for related notes in the SAPNet.
2. Open an SAP Customer Message describing your problem

Best Practice: Data Consistency Check

© **Copyright** 2008 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[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

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

HTML, DHTML, XML, XHTML 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.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Disclaimer: SAP AG assumes no responsibility for errors or omissions in these materials. 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.