Version: 1.0

April 2007

SAP Standards for Data Integrity and Transactional Consistency

Whitepaper

Activ Global Support
SAP AG

SAP® Standards for Data Integrity and Transactional Consistency

Table of Content

1 Management Summary

Transactional correctness and data consistency are crucial factors for the success of each SAP solution: Your daily business operation – from the end user to management decisions – relies on correct and up-to-date data being available at the right time.

As a consequence, data inconsistencies can lead to severe costs – e.g. due to lost business deals or a non-availability of your solution.

This white paper explains which measures have to be taken and which preconditions need to be assured in order to avoid and handle inconsistent data in your SAP solution landscape. This paper does not only outline how you should set up your organization and processes in order to reliably detect and delete inconsistencies, it also focuses on the measures you have to take before your actual start of production – transactional correctness and data consistency needs to be addressed during the complete lifecycle of an SAP implementation project.

This document outlines the basic concepts SAP recommends to ensure transactional correctness and data consistency. The standard process with its four core steps – Prevention, Monitoring, Business Continuity and Correction - is explained in detail. In addition you will find detailed information about the methodology SAP recommends ensuring that your interfaces and programs are transactional correct to achieve the highest possible standards of data consistency. Furthermore, the tools that need to be used are explained in a dedicated chapter. Finally it is also described which roles and functionalities need to be represented in your organization to ensure good prevention, fast detection and appropriate correction of inconsistent data.

2 SAP Standards for E2E Solution Operations

Mission-critical operations is a challenge. While the flexibility of SAP-centric solutions rises, customers have to manage complexity, risks, costs, as well as skills and resources efficiently. Customers have to run and incrementally improve the IT solution to ensure stable operation of the solution landscape. This includes the management of availability, performance, process and data transparency, data consistency, IT process compliance, and other tasks.

Typically, multiple teams in the customer organization are involved in the fulfillment of these requirements. They belong to the key organizational areas Business Unit and IT. While the names of the organizations may differ from company to company, their function is roughly the same. They run their activities in accordance with the corporate strategy, corporate policies (for example, corporate governance, compliance and security), and the goals of their organizations.

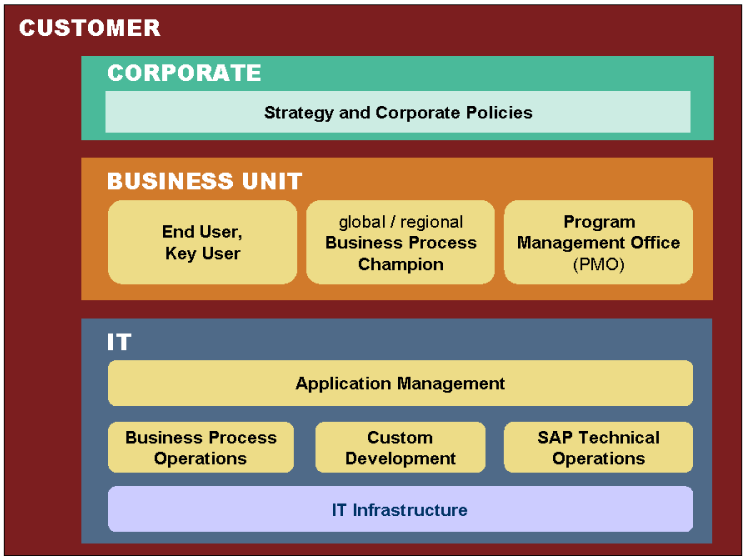


Figure 2.1: Organizational Model

The different teams specialize in the execution of certain tasks: On the business side, **end users** use the implemented functionality to run their daily business. **Key users** provide first-level support for their colleagues. **Business process champions** define how business processes are to be executed. A **program management office** communicates these requirements to the IT organization, decides on the financing of development and operations, and ensures that the requirements are implemented.

On the technical side, the **application management** team is in direct contact with the business units. It is responsible for implementing the business requirements and providing support for end users. **Business process operations** covers the monitoring and support of the

business applications, their integration, and the automation of jobs. **Custom development** takes care of adjusting the solution to customer-specific requirements and developments. **SAP technical operations** is responsible for the general administration of systems and detailed system diagnostics. And the **IT infrastructure** organization provides the underlying IT infrastructure (network, databases, …). Further specialization is possible within these organizations as well. For example, there may be individual experts for different applications within SAP technical operations.

Efficient collaboration between these teams is required to optimize the operation of SAP-centric solutions. This becomes even more important if customers engage service providers to execute some of the tasks or even complete processes. Customers have to closely integrate the providers of outtasking and outsourcing services into the operation of their solutions.

Key prerequisite for efficient collaboration of the involved groups is the clear definition of processes, responsibilities, service level agreements (SLAs), and key performance indicators (KPIs) to measure the fulfillment of the service levels. Based on the experiences gained by SAP Active Global Support while serving more than 36,000 customers, SAP has defined process standards and best practices, which help customers to set up and run End-to-End (E2E) Solution Operations for their SAP-centric solutions. This covers not only applications from SAP but also applications from ISVs, OEMs, and custom code applications integrated into the customer solution.

There are 16 standards for solution operations defined by SAP:

- **Incident Management** describes the process of incident resolution
- **Exception Handling** explains how to define a model and procedures to manage exceptions and error situations during daily business operations
- **Data Integrity** avoids data inconsistencies in end-to-end solution landscapes
- **Change Request Management** enables efficient and punctual implementation of changes with minimal risks
- **Upgrade** guides customers and technology partners through upgrade projects
- **eSOA Readiness** covers both technical and organizational readiness for enterprise service-oriented architectures (eSOA)
- **Root Cause Analysis** defines how to perform root cause analysis end-to-end across different support levels and different technologies
- **Change Control Management** covers the deployment and the analysis of changes
- **Minimum Documentation** defines the required documentation and reporting regarding the customer solution
- **Remote Supportability** contains five basic requirements that have to be met to optimize the supportability of customer solutions
- **Business Process and Interface Monitoring** describes the monitoring and supervision of the mission critical business processes
- **Data Volume Management** defines how to manage data growth

- **Job Scheduling Management** explains how to manage the planning, scheduling, and monitoring of background jobs
- **Transactional Consistency** safeguards data synchronization across applications in distributed system landscapes
- **System Administration** describes how to administer SAP technology in order to run a customer solution efficiently
- **System Monitoring** covers monitoring and reporting of the technical status of IT solutions

Out of this list, this white paper describes the data integrity standard as well as the transactional consistency standard.

3 Transactional Correctness and Data Consistency - Standards at a Glance

3.1 Transactional Consistency

In the early IT days, transactional consistency and correctness was given by the fact that there was an application architecture with one system, one disk sub system, and one database. Commit cycles ensured the completeness of transactions at any time. In today's **distributed system landscapes,** transactional consistency cannot be ensured anymore. Different business process "leading" systems require data synchronization across applications; several data bases, on disk and in memory, have consistent states within itself, but not across the units; on the sub system level, data is stored across several storage systems. To summarize today's situation: within distributed system landscapes there is **no synchronization point** across systems within the business landscape (end to end) any more that ensures data consistency and correctness.

SAP provides **standards for check routines and consistency reports/procedures** that allow synchronizing the transactional data end to end across the different business applications. Depending on customer specific parameters and requirements (like data volume, system resources, 24x7 operations, etc) SAP's given standards need additional implementation focus to ensure end-to-end transactional consistency and correctness. IT operation has to ensure the on-going daily check procedures as well as recovery mechanisms to rebuild after a failure situation potentially caused by hardware errors, user error, data manipulation, etc.

3.2 Data Inconsistencies

Often, the underlying root cause for data inconsistencies in an end-to-end solution landscape is typically one of the following scenarios. On scenario centers around that End Users or Key Users are not aware of manual activities. They have to perform in order to guarantee data consistency. E.g. they change master data in one component and send (as a manual procedure) this update to other systems. In another typical scenario, due to whatever reason an interface is not working properly. Typically this is recorded in an error log on one side of the interface and may or may not trigger a workflow to somebody for resolving the issue. As a final scenario mentioned here, jobs are not running correctly or due to insufficient time where not started at all. If these jobs are for example for loading data into a data warehousing solution (BI) the business reports are not accurate and may lead to wrong business decisions.

It is vital for solution operations to create 100% transparency with regard to all of these issues. The best practices to address issues with data inconsistencies covers several grounds that reach from application design principles such as guiding the user and checks to simply avoid inconsistent data entry, up to proper interface, workflow queues and batch jobs monitoring and CIO level reporting on any exceptions and inconsistencies. If the proper "integration monitoring" and IT reporting is not setup properly, inconsistencies are guaranteed.

All levels of monitoring are the core tasks for **business process operations.** Incomplete processing of batch jobs must be reported to the **business process champions.** They have to inform the business community and define contingency plans.

4 What are the Basic Concepts of the Standards for Data Integrity and Transactional Consistency?

4.1 Architecture and Process Flow

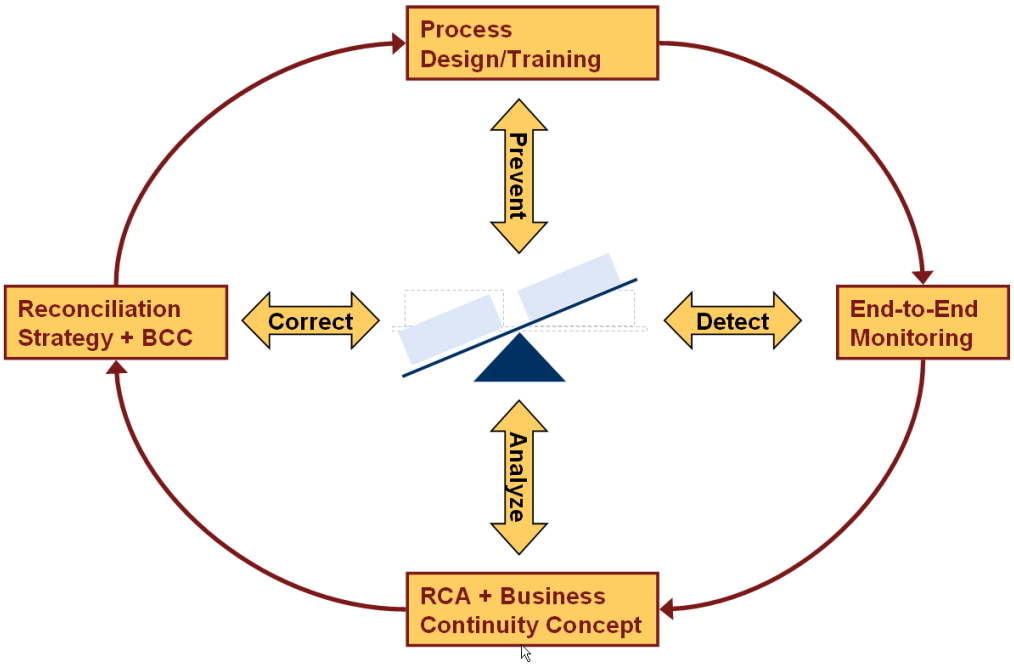


Figure 4.1: Data Consistency – Handling Overview

Handling of inconsistencies can be divided into 4 phases:

Prevention of inconsistencies should be taken into account during the process design and training phase already. Transactional correctness needs to be ensured already during the program design phase, there should be sufficient end user training before Go-Live to enable your end-user to handle even unforeseen exceptions according to defined procedures. Preventing data inconsistencies should also be kept in mind in the business process design phase as well as during the design of customizing and system data change processes. Sufficient interface monitoring and corresponding error handling procedures are a prerequisite for data consistency.

By implementing and performing an inconsistency monitoring procedure inconsistencies can be detected at an early stage, so that further harm is prevented.

There should be a business continuity concept so that when an inconsistency is detected it can be decided quickly if and to what extent you can continue to work with the system and how to revert back as fast as possible to a normal business operation.

Correcting data inconsistencies should include a root cause analysis, so that the root cause is eliminated before the data inconsistency is corrected by an appropriate method, as well as defined correction procedures for each important, business critical object.

4.2 Prevention

4.2.1 Promptly Cross-System-Monitoring

One of the key activities to prevent data inconsistencies is a monitoring that is set up efficiently and promptly and contains detailed work instructions.

Since one of the main origins of data inconsistencies are interfaces, you have to ensure that all interfaces in your SAP solution landscape are constantly monitored to ensure a timely reaction to any unforeseen incidents. Since a data inconsistency is not an isolated event in your business process and can have consequences on the data quality of subsequent business process steps, the impact of inconsistencies is the larger the longer no appropriate actions are taken. So the observed inconsistencies need to be corrected and their root cause needs to be eliminated timely.

Detailed information about how to set up and execute your interface monitoring can be found in the respective SAP standard for solution operation, so here we will only summarize the main points you should focus on, but note that appropriate interface monitoring needs to be set up for all interfaces within your SAP solution landscape – no matter whether they connect SAP components or non-SAP components.

The team responsible for interface monitoring should monitor the availability of the interfaces that are relevant for your business process so that unavailability is detected timely. In addition, the performance and throughput needs to be monitored so that business processes that rely on a fast communication between systems can run at their required level.

While setting up your interface monitoring, you need to define monitoring objects as well as detailed monitoring and error handling procedures. Here you should keep in mind the priorities certain interfaces have for the different business processes. The responsibilities for the different tasks need to be clearly defined and communicated within your organization and the required detailed documentation needs to be available for all involved groups.

Furthermore you have to be aware that data inconsistencies have not necessarily to be caused by erroneous interface communication, but can also have other causes. Therefore, data inconsistencies should also be a topic within your business process monitoring procedures. For details, please refer to the appropriate SAP standard for solution operations, but the main principles –clearly defined and promptly monitoring and error handling procedures as well as clear responsibilities – are applicable for this area as well.

4.2.2 No Transports into a Running System

Another precaution that needs to be taken in order to prevent data inconsistencies in your productive environment is that you avoid **uncontrolled** imports of coding transports into your productive system while it is running.

In order to avoid data inconsistencies by imports into your production environment, you should either transport changes into your system only during downtimes when there is no business activity executed or carefully control the process while understanding possible consequences and taking appropriate measures to avoid them.

4.2.3 No Changes to Existing Customizing Objects

A complete and correct setup of your customizing is an important prerequisite to keep your data consistent over all components of your SAP solution landscape. You have to ensure that this prerequisite is met before the start of your production and thoroughly tested before the GoLive date.

As a general guideline, customizing for existing objects participating in the business processes should never be changed. Therefore we highly recommend to make the customizing of the productive client “non-changeable” and that you change the maintenance status for tables that need to be maintained regularly only for exceptional cases.

In cases where customizing changes may be required, consider the impact of your changes on existing documents and use appropriate correction reports. Do not forget to test these reports before running them in your productive environment and work closely together with the Business Department in order to prevent undesired results!

In general customizing changes should be included in the change management process (for details on change management please see the respective SAP standards for solution operations). Keep in mind that deep knowledge of the application’s customizing as well as of the intended business process using these customizing entries is needed when you intend to change your customizing entries. In general logical correctness and clarity should be rated above possible system usages.

4.3 Monitoring

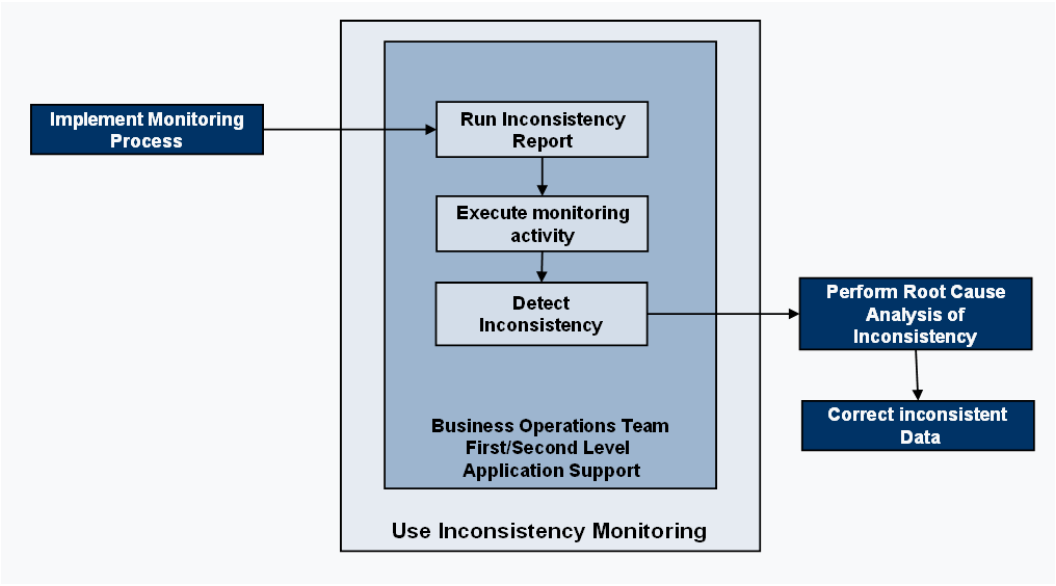


Figure 4.2: Inconsistency Monitoring Process

Inconsistency monitoring should be performed regularly by running the corresponding inconsistency reports for the most critical business objects and monitoring the results by the business operation team and the first and second level application support. Check if the inconsistency reports can be run in a way that they do not show temporary inconsistencies. If this is not possible you can schedule the reports twice and filter out temporary inconsistencies by comparing the results.

If an inconsistency is detected, a root cause analysis should be performed. The root cause analysis should still check if there is a real technical inconsistency and not just a temporary one. Inconsistent data should be corrected after the root cause has been eliminated.

Inconsistencies may cause further issues with business processes handling dependent data. By implementing and performing an inconsistency monitoring procedure inconsistencies can be detected at an early stage, so that further harm can be prevented.

Chapter 5.1.2 describes how to implement the monitoring process. Guidelines of how to perform a root cause analysis and how to correct inconsistent data are provided in chapter 4.5.

4.4 Analysis

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. The general procedure described in this chapter should provide a starting point and a guideline for a detailed analysis independent on the individual

systems and processes involved. The recommended procedure for inconsistency correction is described in the respective chapter of this document.

4.4.1 "Temporary" or "Permanent" Inconsistency?

When an inconsistency is reported, it is usually common that it is reported on a very basic level which does not contain very detailed information regarding the technical data or core business processes involved. At this stage, it is frequently not determined whether there are any "real" 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.

To judge whether the reported incident is a difference or an actual inconsistency you should analyze the data flow "around" the affected data and check whether middleware components work properly, related interfaces are up and running etc. If this is not the case, please correct the detected fault and check whether the assumed inconsistency still persists.

4.4.2 Detailed Information about System Landscape and Business Processes

If it turns out that the observed incident is due to a permanent inconsistency, the next step should be to reach a detailed understanding of the involved business processes, monitoring and reporting activities and technical objects leading to the observed difference. While at this stage not all core business processes of the solution are important, it is essential to know in 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 on data origin and 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. 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.

Related to this is the requirement to obtain detailed information about the different landscape components in which the inconsistency occurred or that the affected business process is running in. In this context it is also crucial to obtain detailed knowledge about the technical relationship between the affected system components – e.g. which system is the leading system for the affected data object etc.

At the end of this process step, the business process context and possible consequences from a technical and business point of view should be clearly visible.

4.4.3 Origin of the Inconsistent Data

The logical mapping of corresponding steps and the understanding of underlying technical data reached during the assessment should be used to verify the data consistency on the lowest technical level available. Data derived on a higher level will always show inconsisten-

cies if the lower level data is already corrupted. This is even true if the involved intermediate processing steps are correct. Expressed differently, one can say that the inconsistent data that is visible to the user in the current business process step may be the result of data that has been used in a preceding step. Now it has to be clarified if the inconsistency has its origin in the current business process step or whether also preceding business process steps were executed with inconsistent data.

At this point in time, check reports are needed to verify the connected data sets. It is important that temporary differences have been 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 inconsistencies 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 analysis should be repeated and results be compared between different runs of the same consistency analysis. Temporary differences will disappear between runs of the check reports/tools while permanent inconsistencies will stay in the results between all runs of the same report.

4.4.4 Timeliness of Inconsistencies

Sometimes the situation arises that inconsistencies exist in “old” data, which means that if the business process steps that produced that data are executed again, the same inconsistencies will not reoccur. If this is the case the old data should be corrected before attempting to identify the root cause of the inconsistency. This should be done by reloading the data or executing correction reports if possible.

If then tests show that the inconsistency does not reoccur, no further root cause analysis needs to be done. It only has to be reassured that all inconsistencies (also the ones that may have been caused by the “original” inconsistency in preceding business process steps) are completely corrected. Please turn the respective chapter of this document for further advise.

4.4.5 Root Cause Identification

If the analysis has revealed so far that the inconsistencies are permanent and do not only affect “old” data, a root cause analysis has to be conducted. The root causes that can lead to inconsistent data are numerous and highly depend on the specific setup of your SAP solution landscape – therefore in the following paragraphs only the most common root causes for inconsistencies are briefly outlined.

The general rule for an inconsistency root cause analysis is not to dig straight into a technical analysis but first to rule out the more operational root causes unless the inconsistency is reproducible. As a first step it should be checked whether one of the measures described in chapter 4.2 “Prevention” (“promptly cross-system-monitoring”, “no transports into a running system”, “no change to existing customizing objects”) can be named as a root cause.

If these causes can be excluded, the following areas should be checked:

No clear definition of a leading system

If within your solution landscape different systems contain and use the same master or transactional data, you should clearly define one leading system for every data object. This means that every change of data is validated in the leading system and from there is distributed to all other systems that use these data objects. Direct data distribution between non-leading systems without the involvement of the leading system may lead to data inconsistencies and therefore has to be avoided in any case!

Which systems are leading for which data object should have been defined in a very early stage of your implementation project – ideally while blueprinting your solution. When defining the leading systems for the different data objects, you also need to define replication procedures for the different data objects.

Please note that another important question that needs to be handled is whether master or transactional data should be able to be changed at all in systems other than the system defined as the leading system.

From a technical point of view, you also have to ensure that distribution of data changes are properly serialized. In case different fields of the same data set are changed in different systems and are replicated it is important that the chronological sequence of the changes is kept in order to have correct information in all connected systems.

In addition, data belonging together from a logical point of view should be kept in the same leading system and you need to define in detail if complete sets of data should be exchanged between the systems or only delta information should be replicated after a change of data.

If it turns out that the business process requirements that are addressed to your SAP project are as such that it is not possible to define a clear leading system, you should ensure that 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) or you should implement a cross system lock. In this case data cannot be changed in a connected system if the affected data set is changed (and locked) in one system of your solution landscape.

Faulty end user input

Data inconsistencies may also occur due to entry of incorrect data by end users. This may be due to a non intuitive user interface misleading the end user, missing or incorrect master data, missing definition of workaround and work instructions for non standard situations; incomplete training for example if the end user does not know how to handle exceptions, or normal human failure.

Most often, this root cause leads to inconsistencies between the real world and the information stored in the system, but it can also lead to inconsistencies between systems.

Basically you should ensure the following:

- End users are sufficiently trained to be able to perform the business process steps (and their physical pendant) completely and in the correct order (for example, period end closing and period end closing reconciliation)

- End users need to be trained in handling any exceptions that might occur
- End users are aware of dependencies between the different data. Therefore information about the underlying data model should be incorporated in the end user training sessions and documentation

Another aspect of faulty user input are authorization – e.g. users should not be allowed to change field input via debugging in a productive system.

Incorrect programming

If incorrect programming is assumed as being the root cause of the inconsistency, the affected business process needs to be traced and debugged. Programming errors leading to a mismatch between both data sets could be purely technical (e.g an incorrect sign in some formula) or logically if the data calculation rules are incorrect (e.g. using the document date instead of a posting date) somewhere.

A special case of incorrect programming leading to inconsistencies is mistakes where the logical unit of work (LUW) concept has been neglected.

To explain how transactional correctness can be achieved by adhering to the “logical unit of work” (LUW)-principle, the concept of database LUWs and SAP LUWs needs to be quickly explained.

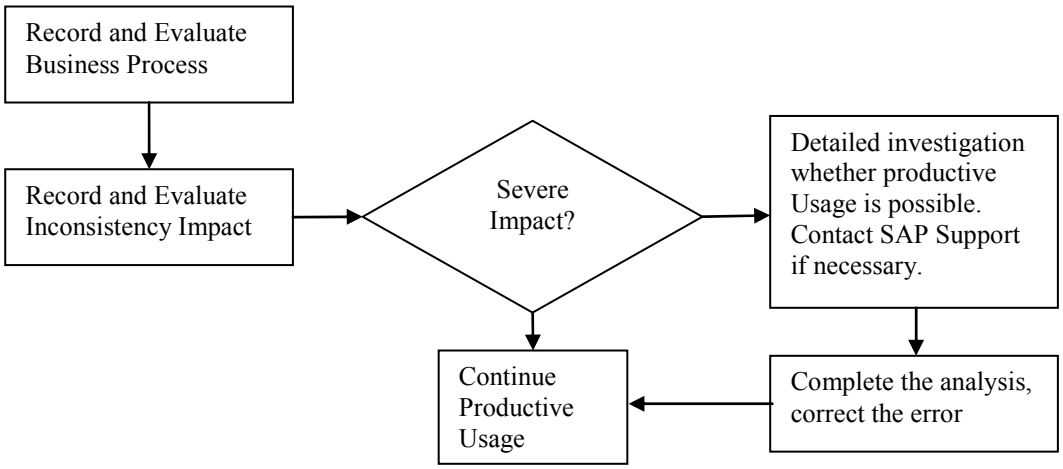
A database LUW (DB LUW) is a non-separable sequence of database operations. At the beginning and end of the LUW, the database is in a consistent state and the LUW is either fully carried out by the database system, or is not carried out at all. It is opened at the start of a transaction (when the connection with the database is made), and whenever a previous DB LUW is closed following a database commit.

The database LUW is closed with a database commit. It is only in the commit that the data is written to the database (after which it can no longer be reversed). Before the database commit, you can undo the changes using a database rollback. Database LUWs allow you to encapsulate logically related actions from a business process.

An SAP logical unit of work (LUW) can be described as the SAP method for mapping several changes into one single DB LUW or as a sequence of dialog steps building one logical unit in the sense of a business process.

Basically, you can say that when programming your interfaces or utilizing a userexit the “all or nothing”-principle should be kept, which means that either all changes are posted to the database in a single LUW or they are not posted at all. In addition it needs to be ensured that appropriate status information is provided to the application whether the data has been successfully committed to the database or not so that in case of errors the right restart point can be detected without posting data twice or not at all.

4.5 Business Continuity



When data inconsistencies have been detected, one of the most important decisions to be made is whether it is possible to continue working with the system or not; and if possible, to what extent (business processes that are not affected by the inconsistent data)? Furthermore, it is crucial to ensure a fast return to normal operations if the business has to be disrupted. Business continuity handles exactly these questions.

Understanding the design of the business processes, that could lead to the observed inconsistency, as well as processes, that can be affected by the inconsistency, forms an essential part of defining the best approach to continue operation. During this business process analysis a detailed description of the end-to-end process across all systems and applications should be recorded. This information along with the analysis how severe is the business impact should be the starting point for the decision regarding business continuity.

Very often contacting SAP Support is essential to avoid even more serious business impact by the productive work in the system with inconsistent data.

4.6 Correction

Once the root cause has been identified it needs to be corrected. The appropriate measure to do so depends on the nature of the root cause. If a coding error has been 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 deliver a System Administration workshop or to develop new operation procedures.

After the root cause has been identified and corrected, it needs to be investigated how dependent data is affected. This could be either consolidated reporting data or follow-up errors like incorrectly created documents.

Several possibilities exist to recover or correct lost and incorrect data sets:

- Complete Restore of a Backup / Point-in-time recovery
- Restore into a parallel system and reload of data from the parallel system by RFC, LSMW etc
- Reload of data from a leading system
- Correction tools and recovery by relationships to other data or redundant stored data
- Manual correction

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 be corrected afterwards by correction reports. Each of the different methods has certain advantages and disadvantages leading to different use cases.

Important questions that influence the choice of recovery method 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?
- Which quantity/complexity of objects are affected?
- Is a backup available for a point-in-time recovery?
- How much time would the different methods require?

5 How to Implement the Data Integrity and Transactional Consistency Standard?

5.1 Prevention

The principles how to prevent data inconsistencies that are outlined in chapter 4 can be divided into two areas – interface and business process monitoring on one hand and change management on the other.

A detailed methodology for interface and business process monitoring is described in detail in the respective SAP standard for solution operations – therefore this topic will only be slightly touched here.

When you want to set up a promptly monitoring in order to prevent data inconsistencies you first have to define monitoring objects and appropriate procedures to monitor these objects. Based on this, detailed error handling procedures have to be worked out that also include key performance indicators (KPIs) based on which appropriate actions are taken. Furthermore, detailed priorities need to be assigned to each of these elements.

The monitoring activities affect all parties within your project organization. The business unit (for example, the business process champion and key users) need to help to define monitoring objects from a business point of view and need to set priorities depending on the business criticality of possible errors related to these monitoring objects. The application management team and business process operations team are the groups that are responsible for executing the monitoring. They are also the people that execute the defined error handling procedures.

Which tools are required for interface and business process monitoring highly depends on the type of components deployed in your SAP solution landscape.

Change management is another important topic when it comes to prevent data inconsistencies. To adhere to the guidelines outlined in this document, you need to ensure that proper change management procedures are set up that include sufficient tests of your developments and your customizing entries. In addition, procedures for coding transports between the different parts of your system landscape have to be clearly defined.

Here your custom development team as developers is highly involved – as well as your application management team. End and key users participate in this process as testers of any changes which are to be applied to the system.

The tools that are involved here are general tools to development programs and reports as well as tools for testing and transporting these developments between components of your system landscape.

5.2 Monitoring

Figure 5.1: Implement Monitoring Process

Monitoring objects and monitoring procedures should be defined jointly between your business department, your application management and business process operation teams.

After the blueprinting and during the technical development of your solution, it is crucial that you implement an inconsistency monitoring concept. Therefore, it needs to be defined which inconsistency reports should be scheduled, run and monitored regularly. This should be done for the most important business objects which mostly endanger the smooth and reliable flow of your core business processes if they are not in a consistent state. If necessary, such inconsistency reports still need to be created.

You should agree for each inconsistency report on inconsistency areas as job variants. Error handling procedures for inconsistency cases as well as thresholds for alert triggering should be defined and documented. There should be a scheduling plan for the inconsistency reports, which also considers other system activities because of the system load and to prevent that temporary differences are shown, if possible. Monitoring activities as well as the responsible persons should be defined and the defined monitoring should be included into the service level management concept.

The monitoring procedures also need to be thoroughly tested and need to fulfill the End-to-End-approach, which means that they cover all involved components and also contain detailed instructions how to detect data inconsistencies, how to correct them and how to find and solve their root cause. These procedures need to be clearly documented in an operation handbook that needs to be available for every involved employee. All employees need to be trained sufficiently in these processes and escalation procedures need to be clearly defined.

The implementation of the inconsistency monitoring process should be done by a business process expert, because detailed knowledge about critical inconsistencies as well as the involved business processes is required, whereas the monitoring can be done by the business operations team. To monitor inconsistencies, several application specific consistency reports/ tools exists which are available per SAP notes as well as SAP standard in the system.

If support is needed for identifying data inconsistencies, there is the possibility to order a data consistency check service.

5.3 Analysis

General guidelines about the methodology to be applied for analyzing data inconsistencies can be found in chapter 4 of this document. It is important that you adhere to the recommended process and consider all topics addressed in that chapter.

The people that help to put this process into practice are in general all parties involved in your SAP implementation, since depending on the business process step different groups are involved.

The analysis whether an inconsistency is temporary or permanent is a task that involves the business process operations group, whereas end users and key users are involved when it comes to understanding the business process “around” the detected inconsistency. To learn more about the underlying technical infrastructure and its interdependencies also your SAP technical operations group or even your IT infrastructure group might be involved. Tasks like determining whether the inconsistencies occur only in “old”, or also in “new” data lie within the area of responsibility of your business process operations team. The final root cause analysis can also involve your development group. Also first and second level application support teams play a key role in the described process.

The process of an inconsistency analysis involves your monitoring tools for determining whether “pending” system activities causes the inconsistencies and check reports whether and where inconsistencies persist. Debugging and development tools are required for a deeper analysis. All these tools highly depend on the specific components and the setup of your SAP solution landscape and also from the fact, which non-SAP-components you deploy.

5.4 Business Continuity

Top-to-bottom and bottom-to-top analysis can be performed to identify affected business processes. Example: business process A is affected -> database table XXX contains inconsistent data -> table XXX is accessed in business process B -> means business process B

may also be affected by the inconsistency. In the majority of cases this method requires direct SAP involvement.

Another possibility is to involve integration tests or other tests included in your test management strategy to verify which business processes are affected by the data inconsistency.

In all cases, SAP Support should be contacted before making the final decision whether to work in the productive system or not.

If the decision is made that certain business processes have to be shut down in the productive system, a fall-back scenario should be activated. A Fall-back scenario is a set of procedures and processes that have to be used by the business users in case the productive system is not available for the time exceeding maximum allowed downtime defined in the SLA. Having a clearly defined fall-back scenario described with the necessary level of detail is a must for any implemented SAP solution.

Automated test scenarios may help to quickly identify the business processes affected by the occurred inconsistency. CATT/e-CATT or other tools may be used at this point.

Several roles are involved in the business continuity process when the data inconsistency is detected. Key users deliver an important input for the identification of the business impact. Application support employees with the understanding of the technical background also provide important information regarding the severity of the inconsistency to the business process champion. In addition, application management department is initiating the communication with SAP Support. The business process champion is making the decision for the business continuity using the input from the key users, application management department, SAP Support and based on his understanding of the business process flow.

5.5 Correction

Figure 5.2 shows a roadmap for the correction of inconsistent data.

The described guidelines for the root cause analysis and correction of detected inconsistencies should be included in the implementation of the inconsistency monitoring process as general error handling procedures and made available for the involved people.

The first and second level support, which is responsible for performing the root cause analysis as well as the correction of the inconsistency, should be trained with the content of these general guidelines.

The same is true for error handling procedures for specific inconsistency cases.

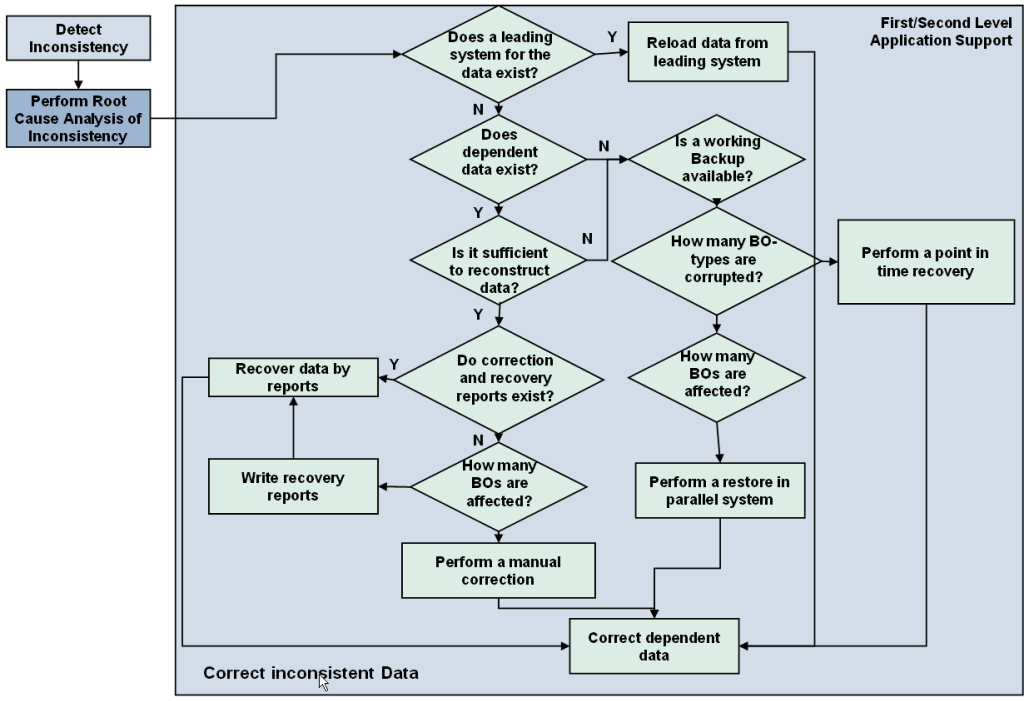


Figure 5.2: Correct Inconsistent Data

A lot of application specific consistency reports/ tools are partly available per SAP notes, partly available per standard in the system for the correction of inconsistencies.

Before proceeding with the correction of inconsistencies, the following factors should be considered:

1) Before starting with the correction of data inconsistencies, make sure that real inconsistencies and not temporary differences are observed. Refer to the chapter 4.4.1 of this document.
2) During the correction process, not only originally detected inconsistencies should be fixed, but also the inconsistencies or erroneous data resulting from the original inconsistencies. This could be either consolidated reporting data or follow-up errors like incorrectly created documents. To understand the impact of the inconsistency on depending data, the business processes using the original data have to be followed. If follow-up documents are created (for example controlling or financial data for logistics data), these need to be corrected as well using appropriate actions.
3) Before proceeding with the repair of inconsistent data, an investigation is required to determine if new or more recent (up-to-date) data exist in the system. If possible, these data should not be overwritten with the correction process.

6 How to Measure the Success of the Implementation?

If transactional correctness of your implemented SAP solution landscape is not ensured and data inconsistencies occur this can lead to numerous consequences – e.g.:

- Financial loss due to lost business: For example, if a customer order is not correctly transferred from one system to another, ordered goods will not be sold and therefore no revenue will be generated. Furthermore the customer satisfaction will decrease, which can lead to business loss in the future, which is indirectly linked to the data inconsistencies that occurred.
- Financial loss due to wrong management decisions: Management decisions are always based on information. If this information is generated out of your SAP solution landscape, data inconsistencies in your SAP system can lead to wrong Management decision. For example, wrong figures in BW due to inconsistencies between BW and the related system can lead to assumption that are not correct and have wrong business decisions as a consequence. Another example are inconsistencies in your warehouse stock when having connected several warehouse systems to your SAP solution. When the purchasing department assumes that there is not enough stock of a certain product and orders this product, but in reality the available storage bins are fully packed this can lead to additional warehousing costs which would not have occurred without data inconsistencies
- Costs due to correcting data inconsistencies: If data inconsistencies have occurred the IT department needs to take measures to correct these inconsistencies. This is related to additional effort which would not have been required if inconsistencies had not occurred – as detecting all inconsistencies, deleting the inconsistencies and finding and deleting their root cause. After the provision of a solution for the inconsistencies' root cause testing by the business department (end users and key users) is required, which again causes additional costs.

SAP® Standards for Data Integrity and Transactional Consistency