## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>Emphasized words or expressions.</td>
</tr>
<tr>
<td><strong>EXAMPLE</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2014-12-12</td>
<td>First version created</td>
</tr>
</tbody>
</table>
Table of Contents

1 SAP Standards for E2E Solution Operations ................................................................. 5
   1.1 Control Center Approach ......................................................................................... 6

2 Overview of the SAP Standard for Data Consistency Management .............................. 9
   2.1 Data Consistency Aspects ....................................................................................... 9
   2.2 General Architecture and Process Flow ................................................................. 10
   2.3 Transactional Consistency ..................................................................................... 11
   2.4 Temporary and Permanent Data Inconsistencies ................................................... 11
   2.5 Causes of Data Inconsistencies ............................................................................. 11

3 Lifecycle of Data Consistency Management ............................................................... 13
   3.1 Plan ....................................................................................................................... 13
      3.1.1 System Architecture ..................................................................................... 13
      3.1.2 Logging and Tracing ..................................................................................... 14
      3.1.3 Reprocessing ................................................................................................. 14
      3.1.4 Correct Programming ................................................................................... 14
   3.2 Build .................................................................................................................... 14
      3.2.1 Prevention ...................................................................................................... 15
      3.2.2 Monitoring ..................................................................................................... 15
      3.2.3 Detection ....................................................................................................... 16
      3.2.4 End-User Training ....................................................................................... 18
   3.3 Run ....................................................................................................................... 18
      3.3.1 Prevention ...................................................................................................... 19
      3.3.2 Monitoring ..................................................................................................... 19
      3.3.3 Analysis ......................................................................................................... 20
      3.3.4 Error Handling .............................................................................................. 23
      3.3.5 Correction ..................................................................................................... 23
      3.3.6 Business Continuity ..................................................................................... 25
   3.4 Optimize ............................................................................................................... 26

4 Driving Continuous Improvement ............................................................................. 27

5 Training .................................................................................................................... 29

6 More Information .................................................................................................... 30
1 SAP Standards for E2E Solution Operations

IT organizations face new challenges every day as they attempt to remain effective and future safe while also keeping costs for day-to-day operations as low as possible. They are also being challenged more than ever to demonstrate their value to businesses. Therefore, it is important to optimize the day-to-day tasks that have less obvious business value and to use KPI and benchmark-based reporting to make IT processes more visible, demonstrating the real value that IT can provide.

In order to minimize the costs of IT, it is necessary to standardize and automate IT processes end-to-end (E2E) without reducing the SLAs required by the business, such as stability, availability, performance, process and data transparency, data consistency, IT process compliance, and so on.

Based on the experience gained by SAP Active Global Support (AGS) while serving more than 36,000 customers, SAP has defined process standards and best practices to help customers set up and run E2E solution operations for their SAP-centric solutions.

The Build phase of SAP best practices supports a Build SAP Like a Factory approach, consisting of the following processes:

- Custom code management
- Change, test, and release management
- Incident, problem, and request management
- Solution documentation
- Remote supportability

During the Run phase of a solution, adapting your IT infrastructure to a Run SAP Like a Factory operation impacts both application operations and business process operations. Therefore, operations processes, such as technical monitoring, end-to-end root-cause analysis, technical administration, and data volume management need to be optimized to achieve state-of-the-art application operations. In business process operations, the same applies to business process and interface monitoring (including performance optimization), data consistency management, and job scheduling management.

Quality management processes and tasks need to be established throughout the lifecycle to guarantee continuous improvement of the end-to-end operations processes while simultaneously ensuring the flexibility needed to react to changing requirements.
This figure shows an organizational model for solution operations that aligns SAP best practice topics and E2E standards with SAP’s control center approach. The Operations Control Center executes and controls the Run SAP Like a Factory processes, while the Innovation Control Center ensures optimal custom code management and a smooth transition to production with integration validation procedures. SAP connects to these control centers from the Mission Control Center to ensure that professional support is available to the customer. The following Application Lifecycle Management (ALM) functions are not provided directly in one of the control centers because they must be handled across different areas:

- Change, test, and release management
- Incident, problem, and request management
- Solution documentation
- Remote supportability

The quality management methodologies are an essential part of SAP’s Advanced Customer Center of Expertise (CoE) concept and ensure that the KPI-driven processes are continuously improved across all processes and teams. In addition, the quality manager roles ensure consistent and value-centric reporting to the business and management. This unified reporting platform is known as the Single Source of Truth.

1.1 Control Center Approach

The Operations Control Center (OCC) is the physical manifestation of the Run SAP Like a Factory philosophy. The OCC allows for automated, proactive operations, which simultaneously reduces operational costs while increasing the quality of IT services, leading to improved business satisfaction. The OCC also drives continuous improvement of business processes and IT support. To achieve these goals, it relies on a close interaction with both the Innovation Control Center (ICC) and the SAP Mission Control Center (MCC).
The OCC is a central IT support entity at the customer site, which monitors the productive SAP environment as well as important non-SAP applications. During operation, the OCC requires a workforce of 2 full-time equivalent (FTE) per shift to ensure that incidents are detected and resolved as quickly as possible. The OCC is equipped with large screens that display the status of business processes, IT landscape components, as well as exceptions and alerts. If problems occur, you use a video link to get live support from SAP and partners. The customer usually sets up the room with assistance from SAP Active Global Support (AGS). The customer is responsible for managing the OCC and the team of technical and functional IT operators who act on the alerts. The OCC is most effective when closely integrated with other IT processes, such as IT Service Management (ITSM) and Change Management. Central monitors and dashboards based on application and business process operations display the current status of business and IT-related processes. This data can also be used to drive continuous improvement. An effective system monitoring and alerting infrastructure is fundamental to the success of an OCC.
The OCC is most effective when closely integrated with other IT processes, such as IT Service Management (ITSM) and Change Management. Central monitors and dashboards based on application and business process operations display the current status of business and IT-related processes. This data can also be used to drive continuous improvement.

An effective system monitoring and alerting infrastructure is fundamental to the success of an OCC. For Job Scheduling Management, the OCC supervises all background monitoring processes, SAP controls and legacy background operations. It reacts to job monitoring alerts according to predefined error-resolution activities, and triggers follow-up activities for error handling if the relevant task are not completed within a certain timeframe.

**Figure 3: OCC Concept**
2 Overview of the SAP Standard for Data Consistency Management

2.1 Data Consistency Aspects

Data consistency and correctness used to be ensured by creating an application architecture that only contained one system, one on-disk subsystem, and one database. Commit cycles ensured the completeness of transactions at all times. In today’s distributed system landscapes, data consistency is no longer guaranteed. Leading systems for different business processes require data to be synchronized across applications. Individual databases (both on disk and in memory) are usually consistent, but this consistency is difficult to maintain across multiple systems and subsystems. There is no longer a central synchronization point to ensure data consistency end-to-end (E2E).

The SAP Standard for Data Consistency Management considers the following basic concepts:

- **Transactional correctness**
  - Data consistency within single transactions in one business system

- **End-to-end transactional consistency (application consistency)**
  - Data consistency within many business transactions targeting one or more applications and systems.

- **Proactive data consistency checks**
  - Ongoing monitoring of data to detect consistency issues before they affect your daily operations

- **End-to-end process completion checks**
  - Monitoring of business processes to determine whether a process is finished and correct

- **System integration checks**
  - Monitoring of queues to ensure processing completion (backlog) and identify errors

- **Protection against inconsistencies created by end users**
  - Ensuring that the order in which users enter data on different screens does not affect the database

Data consistency management affects all areas of Application Lifecycle Management (ALM) from designing your solution to managing operations after going live.

SAP provides best practices, consistency reports, and procedures that enable you to synchronize transactional data end-to-end across different business applications. Depending on customer-specific parameters and requirements, for example, data volume, system resources, and 24-hour operations, the SAP Standards for E2E Solution Operations require additional implementation focus to ensure transactional consistency and correctness. IT must perform regular checks and integrate recovery mechanisms to rebuild following system failures caused by hardware errors, user errors, or data manipulation.
2.2 General Architecture and Process Flow

The data consistency management process can be divided into the following four phases:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prevention</td>
<td>You should consider how to prevent inconsistencies during the Plan phase and ensure transactional correctness during the Build phase. End users also need relevant training before go-live so that they can manage exceptions according to predefined procedures.</td>
</tr>
<tr>
<td>Detection</td>
<td>Sufficient interface monitoring and corresponding error-handling procedures are a vital prerequisite for data consistency. By implementing and performing inconsistency monitoring, you can detect inconsistencies at an early stage, before they cause serious problems.</td>
</tr>
<tr>
<td>Analysis</td>
<td>To prevent further inconsistencies caused by the same problem, you need to identify the root cause. You should also have a business continuity concept so that when an inconsistency is detected, you can quickly decide whether you can continue to work with the system and to what extent, as well as identify how to return to normal business operations as quickly as possible.</td>
</tr>
<tr>
<td>Correction</td>
<td>After identifying the root cause, you should eliminate it before the data inconsistency becomes a critical problem. You should define standardized correction procedures for each business-critical object.</td>
</tr>
</tbody>
</table>
2.3 Transactional Consistency

Transactional consistency plays a vital role in maintaining data integrity. Therefore, it is essential that you have a solid understanding of the basic principles of transactional consistency, such as the concept of logical units of work (LUWs).

A database LUW (DB LUW), or simply database transaction, is a non-separable sequence of database operations. At the beginning and end of the DB LUW, the database is in a consistent state, that is, the DB LUW is either fully performed by the database system or is not performed at all. A DB LUW opens 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.

You end the DB LUW by issuing a COMMIT statement; only then is data written to the database. Before issuing a COMMIT statement, you can undo changes using a ROLLBACK statement. However, after a commit, you can no longer reverse your changes. Database LUWs enable you to logically group related actions in a business process.

An SAP logical unit of work (SAP LUW) is the SAP method for mapping several changes onto one DB LUW, or as a sequence of dialog steps forming one logical unit, for example, a business process.

When programming your interfaces or using a user exit, you either make all changes to a database in a single LUW or you do not post them at all. In addition, you need to provide the application with the appropriate status information to check whether the data has been successfully committed to the database. This ensures that if there is an error, you can identify the correct restart point without posting data twice.

2.4 Temporary and Permanent Data Inconsistencies

It is important to differentiate between temporary and permanent inconsistencies. Temporary inconsistencies are differences between data that always occur in connected running systems. This could be caused by different processing times of individual update tasks, IDocs, BDocs, and other interfaces, or by different scheduling frequencies between systems. Therefore, temporary inconsistencies usually correct themselves when all system processes are complete.

A permanent inconsistency does not disappear when all system processes are complete and is, therefore, much more important for businesses to correct. Before attempting to correct an inconsistency, it is essential that you identify whether it is temporary or permanent. For more information, see Analysis Process.

2.5 Causes of Data Inconsistencies

Data inconsistencies in an end-to-end solution landscape are usually caused by one of the following scenarios:

- End users or key users are not aware of manual activities that they have to perform to guarantee data consistency. For example, they change master data in one component without manually sending this update to other systems, or enter incorrect data into the system.
- An interface is not working properly. Typically, this is recorded in an error log on one side of the interface and triggers a workflow for resolving the issue.
- Jobs are not running correctly or were not started due to time restraints. For example, if a job for loading data into your data warehousing solution is not started on time, business reports that use this data will be outdated and inaccurate, which could lead to poor business decisions.
• During system recovery, older system data is not restored in one of the systems.
Solution operations must be 100% transparent in these areas. Therefore, you need to produce best practices for dealing with data inconsistencies that cover the following topics:
• Application design principles
• Checks to prevent inconsistent data entry
• Interface, workflow queue, and batch job monitoring
• CIO-level reporting on any exceptions or inconsistencies
Failure to establish proper integration monitoring and reporting processes will almost certainly result in data inconsistencies.
3 Lifecycle of Data Consistency Management

This section outlines the necessary steps to implement the SAP Standard for Data Consistency Management based on the following key stages:

- Prevention
- Monitoring
- Analysis
- Correction

For each of these stages, you must plan your processes, build the relevant tools and processes into your existing SAP implementation, run the processes according to your plan, and then monitor your implementation to identify areas for improvement and subsequently optimize your processes.

3.1 Plan

In the planning phase, you define the key aspects of your implementation plan. It is important to consider the following aspects:

- Defining your architecture
- Logging, tracing, and repeating processes
- Preventing programming errors

3.1.1 System Architecture

If your solution landscape contains different systems that use the same master or transactional data, you should define one leading system for each data object. This means that every change to data is validated in the leading system and distributed to all other systems that use the changed data. Direct data distribution between non-leading systems can result in data inconsistencies and is, therefore, not recommended.

You should define the leading systems for each data object in the early stages of your implementation project, for example, when creating your project blueprint. You also need to define replication procedures for the different data objects.

→ Recommendation

SAP recommends blocking changes to master or transactional data in any system other than the leading system.

From a technical perspective, you should also make sure that the distribution of changes is properly serialized. The chronological sequence of changes needs to be consistent to ensure that information is correct across all
connected systems, even when different fields within the same data set are changed in different systems and then replicated.

In addition, you should store data that logically belongs together in the same leading system. It is essential that you define whether complete sets of data are exchanged between the systems or whether only delta information is replicated after data is changed.

If business-process requirements mean that you cannot define a clear leading system, maintain distinct data sets in each system. For example, store different material types in different systems or store sales data in one system and production data in another. You can also implement a cross-system lock, which prevents you from changing data in a connected system if the affected data set is being changed in another system in your solution landscape.

### 3.1.2 Logging and Tracing

Logging and tracing processes is important for end-to-end process consistency. For each database update performed in a distributed environment, you must log whether the step was started or finished in the client system or the server system. In addition, you should log information about the created business object and any data used.

### 3.1.3 Reprocessing

It is particularly important to evaluate logged data about synchronous interface steps performed in multiple systems. This means you can skip successfully performed steps and focus on reprocessing any missing or erroneous steps if necessary.

### 3.1.4 Correct Programming

Programming errors can lead to data inconsistencies between two data sets. These inconsistencies may be purely technical, for example, when an incorrect symbol is inserted into a formula, or they may occur if the data calculation rules are incorrect, for example, if you use a document date instead of a posting date.

Incorrect programming leading to inconsistencies occurs when the logical unit of work (LUW) concept has been neglected. Therefore, it is important to implement your solution according to the Application Lifecycle Management (ALM) process, which involves rigorous testing of all coding.

### 3.2 Build

This section outlines the actual implementation of a data consistency management concept. It relates to implementing prevention measures, monitoring processes, and detection mechanisms, which are necessary to ensure correct and up-to-date data.
3.2.1 Prevention

To ensure that your data architecture and applications guarantee transactional correctness and data consistency, you need to consider inconsistency prevention measures not only during the planning phase, but also while designing your business processes, Customizing, and data change processes. It is also important to provide sufficient end-user training before the go-live date to make sure your end users can handle exceptions according to defined procedures.

Effective change management plays a vital role in preventing data inconsistencies. To adhere to the guidelines outlined in this document, you must ensure that proper change management procedures are set up, including clearly defined procedures for transporting coding between the different parts of your system landscape and sufficient testing of any developments and Customizing entries. Your custom development team and application management team should be heavily involved in this process. End users and key users participate by testing any changes before they are applied to the production system.

You use general tools for developing programs and reports and other tools for testing and transporting these developments between components in your system landscape. A detailed methodology and process for preventing data inconsistencies is described in the following SAP Standards for Solution Operations:

<table>
<thead>
<tr>
<th>SAP Standard for Solution Operations</th>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Code Management</td>
<td>Basic concepts of custom code operations and optimization</td>
</tr>
<tr>
<td>Change Control Management</td>
<td>Deployment and analysis of changes within a landscape</td>
</tr>
<tr>
<td></td>
<td>Efficient and low-risk implementation of changes</td>
</tr>
<tr>
<td>Test Management</td>
<td>Methodology and approach for functional, scenario, integration, and technical system tests for SAP-centric systems</td>
</tr>
</tbody>
</table>

3.2.2 Monitoring

Another way to prevent data inconsistencies is to set up effective monitoring. You first have to identify which objects to monitor and define appropriate procedures for monitoring them. Based on the results, you can produce detailed error-handling procedures and key performance indicators (KPIs) to help you identify the appropriate measures to take.

Monitoring activities affect all parties within your organization. Therefore, you should consult different business units, for example, the business process champion or key users, to help you identify the most suitable objects for monitoring and set priorities based on the business criticality of the objects. The application management team and business process operations team are responsible for the monitoring activities and executing the defined error-handling procedures.

Which tools are required for interface and business process monitoring depends on the type of components deployed in your SAP solution landscape. A detailed methodology for interface and business process monitoring is described in detail in the SAP Standard for Business Process Monitoring.
One of the main ways of preventing data inconsistencies is to implement a monitoring process that covers all relevant components and process steps, can generate alerts quickly, and contains detailed work instructions and clear responsibilities in case of an incident or error. For more information, see Detection.

**Interface Monitoring**

As interfaces are a major cause of data inconsistencies, you have to constantly monitor all interfaces in your SAP solution landscape to ensure a quick response to any unforeseen incidents. As 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 increases the longer you wait to act. Therefore, you need to correct any inconsistencies you identify and determine the root cause as quickly as possible. It is important to set up appropriate interface monitoring for all key interfaces within your SAP solution landscape, regardless of whether they connect to SAP components or non-SAP components.

**Consistency Check Reports**

Data inconsistencies are not only caused by erroneous interface communication. They can also be caused by errors, such as background jobs that have not been properly executed or problems in a transaction or program. In addition to monitoring the possible root causes, you need to consider data inconsistency detection as part of your business process monitoring procedures. SAP offers various data consistency reports for different applications and business objects. When you modify these business objects, you need to adapt or expand the reports so that they cover your new data structures. You have to execute these reports regularly to be able to react as quickly as possible.

**Application-Specific Monitoring for Business Objects**

Application-specific monitoring for business objects enables you to detect consistency issues at the application level, which appear when a step in an end-to-end process has failed or has not been executed. For example, failure to post a goods movement notice will lead to an inconsistency between the actual stock information and the digital inventory data. This business error can be monitored using Business Process Monitoring for SAP Solution Manager.

**End to End Process Monitoring**

You have to monitor different exceptions in the process flow. How you do this depends on the logging mechanisms you have built into your landscape. Monitoring should show which process steps have been executed successfully, which have failed, and which steps are still missing. Continuous and proactive monitoring is essential so that guided procedures for reprocessing and error correction can be started quickly, for example, by using the Exception Management Cockpit and Business Process Monitoring.

### 3.2.3 Detection

If an error occurs despite your inconsistency prevention measures, you must have a defined procedure for detecting and handling the inconsistency. SAP recommends a process similar to the one shown in the following figure.
You business departments, the application management team, and the business-process operations team should all contribute to the definition of monitoring objects and procedures.

After blueprinting and during the technical development of your solution, it is crucial that you implement an inconsistency monitoring concept. Therefore, you must specify which inconsistency reports need to be scheduled, run, and monitored regularly. This applies mostly to the business processes that pose the greatest danger of interrupting your core business processes if they are not kept in a consistent state. Define appropriate variants for each consistency report that you want to monitor. This enables you to distinguish between different recipients and target groups. In addition, define and document error handling procedures for inconsistency and thresholds for alert triggering. If possible, take into account other system activities and system loads when scheduling your inconsistency reports so that temporary inconsistencies are not included in the results. The defined monitoring activities and corresponding responsible persons should be included in the service-level management concept.

To make sure that your monitoring procedures follow the end-to-end approach, test whether they cover all components and contain detailed instructions of how to detect and correct data inconsistencies and how to identify and solve the root cause. These procedures should be documented clearly in an operations handbook that is available for every employee involved. In addition, all employees must be sufficiently trained in these processes and you must define clear escalation procedures.

You must design your inconsistency monitoring process in conjunction with business process champions because detailed knowledge about critical inconsistencies and the involved business processes is required. However, the business process operations team can implement and operate the monitoring process. Several application-specific consistency reports and tools are available as standard in SAP Solution Manager and you can add others by implementing various SAP Notes.
SAP Solution Manager provides a dedicated Business Process Operations work center, which includes areas for business process and interface monitoring, as well as data consistency management. The latter provides a separate inbox for all alerts relevant to data consistency and provides access to tools for investigating and correcting data inconsistencies.

If you need help identifying data inconsistencies, you can order a data consistency check service from SAP.

### 3.2.4 End-User Training

Data inconsistencies can also occur because of incorrect data entry by end users. This may be due to one or more of the following factors:

- Misunderstandings caused by a confusing user interface
- Missing or incorrect master data
- Missing definitions of workarounds or instructions for non-standard situations
- Incomplete training, for example, if the end user does not know how to handle exceptions
- Normal human error

These root causes often lead to inconsistencies between the real world and the data stored on the system, but they can also lead to inconsistencies between systems.

To avoid these situations, ensure that end users are fully trained in the following areas:

- Performing the business process steps completely and in the correct order. For example, ensure that they know how and when to perform period-end closing and period-end closing reconciliation
- Handling any exceptions that might occur
- Identifying dependencies between the different data. For example, you could include the underlying data model in the end-user training sessions and documentation.

You can also avoid data inconsistencies by restricting the authorizations of each user. For example, do not allow end users to use debugging to change field input in a production system.

The business process operations team is responsible for performing the root cause analysis and correcting the inconsistency. Therefore, make sure they are given sufficient training in your guidelines for root cause analysis and correction of detected inconsistencies. You should make sure this is done as part of your inconsistency monitoring process. The relevant documentation must be available to all parties involved in addition to the general and specific error-handling procedures.

### 3.3 Run

This section contains general guidelines about the methodology that you should apply when handling data inconsistencies. It is important that you adhere to the recommended process and consider all topics addressed in this chapter.
3.3.1 Prevention

When you have planned and implemented the data consistency management and change management processes, you need to ensure that they are executed properly. This means following the following basic principles.

Do Not Transport to a Running System

To prevent data inconsistencies in your production environment, avoid uncontrolled imports of coding transports into running production systems. Either only allow transports into your system during downtimes, that is, when no business processes are being executed, or carefully control the process while understanding the possible consequences of each change and taking the appropriate measures to avoid them.

Do Not Change Existing Customizing Objects

To help ensure data consistency across every component in your SAP system landscape, complete all Customizing and related testing before the go-live date.

As a general rule, you should never change the Customizing for existing objects in business processes. Therefore, SAP strongly recommends that you make the Customizing in the production client non-changeable and that you only change the maintenance status of tables that need to be regularly maintained in exceptional cases.

In cases where Customizing changes may be required, consider the impact of your changes on further processing and usage of existing documents, and use appropriate correction reports if necessary. Do not forget to test these reports before running them in your production environment and work closely with the business department to prevent undesired results.

Ideally, Customizing changes should be included in the change management process. Before making any changes, make sure you have a thorough understanding of the Customizing and related business processes.

3.3.2 Monitoring

The goal of data consistency monitoring is to detect problem situations as early as possible to solve them before they become a critical problem for the business. Consistency monitoring enables the customer’s support organization to respond to and solve problems proactively.

The team responsible for interface monitoring monitors the availability of the interfaces that are relevant to your business process. This enables you to quickly detect any unavailability. In addition, 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 and detailed monitoring and error handling procedures. For this, 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 each group that is involved.
You should perform data inconsistency monitoring regularly by running the corresponding inconsistency reports for the most critical business objects and the business process operation team must monitor the results.

Check if the inconsistency reports can be configured so that they do not show temporary inconsistencies when run. If not, 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 checks if it is a real technical inconsistency or a temporary one. After you have eliminated the root cause, the inconsistent data is deleted.

Inconsistencies may cause further issues in data relating to business process handling. By implementing and performing an inconsistency monitoring procedure, you can detect inconsistencies at an early stage, so that further harm can be prevented.

For more information on setting up and executing your interface, business process, and data consistency monitoring can be found in the SAP Standard for Business Process Monitoring.

### 3.3.3 Analysis

#### 3.3.3.1 Roles for Analysis

As part of the data consistency management process, you have to analyze the inconsistent data, including dependent data, determine business impact, identify the root cause of the inconsistency, and decide whether you need to stop using the production system. This process involves all of the parties working to implement the solution.

The following table indicates typical roles and their key responsibilities.
Role | Responsibilities
--- | ---
Business process operations team | • Analyze whether an inconsistency is temporary or permanent
 | • Determine whether the inconsistencies exist in old data, new data, or both types of data
End users and key users | Identify the affected business process
SAP technical operations group or IT infrastructure group | Identify any problems in your technical infrastructure and its dependent systems
Application operations teams | Play a role in every step of the process

Inconsistency analysis also involves using monitoring tools to determine whether pending system activities are causing the inconsistencies and to check reports to determine if and where inconsistencies persist. You have to use debugging and development tools for deeper analysis. Which tools to use depends on how your SAP landscape is set up and which specific components you are using, including any non-SAP components.

### 3.3.3.2 Analysis Process

The general process described in this chapter provides a starting point and guide for detailed inconsistency analysis. The chapter does not assume a specific setup and can be applied no matter what individual systems and processes you use.

When an inconsistency has been discovered and analyzed, key information is often missing, for example, technical or core business-process data. You have to investigate the problem further to decide how best to correct the problem.

**Determine the Nature of the Inconsistency**

At first, it is difficult to know whether a detected inconsistency is permanent or temporary. For more information on, see *Temporary and Permanent Data Inconsistencies*.

To determine whether the reported inconsistency is temporary or permanent, you need to analyze any data flows that involve the affected data, for example, check whether middleware components are working properly, related interfaces are up and running, or whether a transfer backlog exists in one of the interfaces. If this is not the cause of the problem, correct the detected error and check whether the inconsistency remains.

**Obtain Detailed Information About System Landscape and Business Processes**

If the inconsistency is permanent, you have to obtain an understanding of the business processes, monitoring and reporting activities, and technical objects, which led to the inconsistency. At this stage, not all core business processes of the solution are important, but it is essential to know in which business and technical data the inconsistency has been observed.

In addition, you need a detailed technical overview of the data origin, how similar data is used by processes, and any relevant error detection and handling procedures used. The purpose of this task is to understand the steps leading to the detection of the inconsistencies, and to map the corresponding data derivation steps, in case different business processes are involved.

You also have to obtain detailed information about the different landscape components in which the inconsistency occurred or the affected business process is running. It is also important to obtain detailed knowledge of the
technical relationship between the affected system components, for example, which system is the leading system for the affected data object.

If you following this process, you should have a clear understanding of the system context from both a technical and business-process-oriented point of view.

Identify the Origin of the Inconsistent Data

When you have analyzed the related business processes and the corresponding system architecture, it is important to identify where the inconsistency originally appeared.

Use check reports to verify the connected data sets. Make sure that temporary differences have been filtered out. To check this, you run check reports when the relevant system data is not being used. Filtering out temporary inconsistencies ensures that all update tasks or interface processing for the relevant data is finished and that no new data is created during the analysis. If the data is constantly being used, you have to repeat the analysis and compare the results. Temporary differences will not be present in one of the runs, while permanent inconsistencies will be present in both.

Non-Reproducible Inconsistencies

Sometimes, inconsistencies exist in old data. You test this by attempting to reproduce the inconsistency. The same inconsistencies will not reoccur when you repeat the business process steps that produced that error. If this is the case, correct the old data before attempting to identify the root cause of the inconsistency. If possible, do this by reloading the data or executing correction reports.

If you cannot reproduce the inconsistency using this method, it only affects old data and you do not need to perform a root-cause analysis. You need to check that all inconsistencies have been corrected. This includes checking inconsistencies that may have been caused by an earlier inconsistency in a previous business process step.

Identify the Root Cause

If you can reproduce the inconsistency, the inconsistency does not only affect old data, and you have to conduct a root-cause analysis. There are many different root causes that can lead to inconsistent data and they depend highly on the specific setup of your SAP solution landscape. Therefore, this section only outlines the most common root causes.

Begin looking for root causes by assessing operational causes and progress to a more technical analysis. First, check whether one of the following root causes is responsible:

- Transports into running system
- Changes in existing custom objects
- Incorrect error handling
- System misuse
- No clear leading system
- Erroneous user input

If one of these is not the cause, thoroughly check the process chain starting at the bottom to find the first real occurrence of the inconsistency. When you have found an inconsistency between two logically connected steps, debug the technical process involved in these steps and investigate the process to identify the technical root cause. Possible technical root causes include incorrect interfaces, technical programming errors, or logical errors.
For more information see the following best practices for the appropriate field:

<table>
<thead>
<tr>
<th>Documentation</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Consistency Check for Logistics</td>
<td><a href="https://websmp105.sap-ag.de/~sapidb/011000358700001955832008E">https://websmp105.sap-ag.de/~sapidb/011000358700001955832008E</a></td>
</tr>
</tbody>
</table>

### 3.3.4 Error Handling

Error-handling procedures deal with all operational and certain technical errors. They do not relate to programming or logical errors. If you do not have adequate error-handling procedures, although the system programming is correct, you may still experience data inconsistencies. This situation is most common in business processes spanning two systems.

Typically, missing data in one of the two systems is caused by IDocs that are in an error status and are not reprocessed quickly enough. Inconsistencies could also appear when you overwrite newer data with old data, for example, if you accidentally reprocess an older IDoc. This is usually the case if there is a large amount of older erroneous interface data, such as IDocs or BDocs, in the system.

Trained operational staff and effective administrative standards are essential to avoid incorrect error handling. For more details see the SAP Standard for System Administration.

### 3.3.5 Correction

When you have identified the root cause, you have to correct it. How you do this depends on the individual root cause. If a coding error has been identified as the root cause, you need to fix the coding. However, if the issue was caused by an operational error, for example, incorrect error handling, you need to make appropriate changes. This includes updating administrative procedures or developing new operational procedures.
The following figure shows a typical workflow for correcting inconsistencies:

After you have identified and corrected the root cause, you should check whether any dependent data has been affected. Inconsistencies might affect consolidated reporting data or cause follow-up errors, for example, incorrectly created documents.

To check what impact the inconsistency has had on dependent data, you need to follow the relevant business processes. If follow-up documents are created, for example, controlling or financial data, these need to be corrected appropriately.

Before correcting inconsistent data, check whether new or more up-to-date data exists in the system. Do not overwrite this data unless absolutely necessary.

If you lose any data sets or they are irreversibly damaged, you can use the following methods to recover them:

- Restore the data in a parallel system and reload it
- Reload data from a leading system
- Implement correction tools and recover the data using relationships with other current data or redundant stored data
- Manually correct the data
- Restore the data from a backup or perform a point-in-time recovery

**Note**

Restoring data from a backup or performing a point-in-time recovery is a last resort, as this may cause other inconsistencies.

A combination of data recovery methods and tools is often necessary. For example, you can correct individual incorrect sales documents manually in the database and then correct any dependent data afterward using correction reports. Which methods to use depends on the following factors:
- Availability of dependent data used for reconstruction
- Quantity and complexity of affected business objects and instances
- Availability of backup information to perform a point-in-time recovery
- Timeframe for the recovery

For more information on which method to choose, see the best practices for the Data Consistency Check for Logistics at https://websmp105.sap-ag.de/~sapidb/011000358700001955832008E.

### 3.3.6 Business Continuity

When data inconsistencies have been detected, you need to decide whether it is possible to continue working with the system and identify the severity of the impact on your business. If the business has to be disrupted, it is crucial to fix the problem and return to normal operations as quickly as possible.

![Business Continuity Decision Process](image)

**Figure 8: Business Continuity Decision Process**

Understanding the business processes that caused the inconsistency and the affected processes is an essential part of defining the best way to return to normal operation. During this business-process analysis, you have to record a detailed description of the end-to-end process across all systems and applications. When deciding whether to continue operations, you should consider this information along the severity analysis.

Continuing productive business operations using inconsistent data can have a severe impact on your business. Therefore, SAP recommends contacting SAP Support to help you decide whether it is safe to continue operations.

If you decide to shut down certain business processes in the production system, you have to implement a fallback plan. A fallback plan is a set of procedures and processes that business users have to apply if the productive system is unavailable for a certain length of time. This maximum allowed downtime is defined in the SLA. Having a clearly defined fallback plan is essential for any implemented SAP solution.

For more information on business continuity, see following best practices:

<table>
<thead>
<tr>
<th>Best Practices</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency Handling for Recovery of SAP System Landscapes</td>
<td><a href="https://websmp103.sap-ag.de/~sapidb/01100035870000577832008E">https://websmp103.sap-ag.de/~sapidb/01100035870000577832008E</a></td>
</tr>
</tbody>
</table>
Several roles are involved in the business continuity process when the data inconsistency is detected. Key users help identify the business impact. Application support employees with an understanding of the technical background provide the business process champion important information regarding the severity of the inconsistency. In addition, the application management department initiates communication with SAP Support. The business process champion decides on business continuity by using the input from the key users, application management department, SAP Support, and existing understanding of the business process flow. The implementation of a business continuity strategy is described in the Implementation Methodology in the Application Management work package of the Run SAP Roadmap. For more information, see the SAP Service Marketplace at http://service.sap.com/runsap.

### 3.4 Optimize

After inconsistencies are corrected, you have to review documentation and procedures to check and optimize the monitoring, analysis and correction stages. Performing follow-up activities for incidents and documenting the measures taken help to prevent inconsistencies in the future.

SAP offers many application-specific consistency reports and tools, some of which are provided as standard, while others are implemented as SAP Notes, which can be found on SAP Service Marketplace. The Business Process Operations work center in SAP Solution Manager provides a central point of access for all features related to data consistency. You can use these tools to analyze root causes and find relevant correction and recovery reports.

Implementing a guided procedure enables you to create a service session for detailed interaction, navigation, and status tracking of data inconsistencies you identify. You can also provide a status overview for each consistency analysis session.
4 Driving Continuous Improvement

To assess the quality of your implementation, you have to define clear KPIs and measurable objectives. Information on these KPIs should be collected and evaluated in regular reports. Historical data that is created in this way can be used to identify trends and derive necessary optimization measures. KPIs should be determined by considering the main aims of the optimization process. The following table gives examples of possible KPIs, how they benefit your organization, how to measure them, and the ideal result of the test.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Benefit</th>
<th>Measure</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased end-user satisfaction with application support</td>
<td>Reacting to possibly critical situations before the end users and business processes are impacted means that the end user has to open fewer tickets and trusts the solution Application Support team more.</td>
<td>Compare the number of tickets opened by end users per month before and after the business process and interface monitoring concept has been implemented.</td>
<td>End users should only open tickets could not be avoided by successfully implementing your monitoring program.</td>
</tr>
<tr>
<td>Decreased business process downtime</td>
<td>Reacting to possibly critical situations before the have become a serious issue, problems can be solved before any business downtime.</td>
<td>Measure the current monthly business process downtime and identify which business process downtime could have been prevented by the monitoring concept.</td>
<td>Identify which business process downtime could have been prevented by the monitoring concept</td>
</tr>
<tr>
<td>Improved stability of business process</td>
<td>By monitoring stability, solving alerts, reporting on the number of alerts and trends, finding the root cause of critical trends, and applying counter measures, business process stability improves and becomes more stable at a given level.</td>
<td>Measure the number of alerts that have occurred before implementation. After a defined period, determine the number of alerts that are handled by both first and second application support.</td>
<td>The number of alerts should decline and stabilize at this level.</td>
</tr>
</tbody>
</table>

You could also consider the following KPIs:

- Number of documented business processes
- Number of monitored processes
- Number of monitoring objects (interfaces, data consistency, business objects)
- Number of documented error handling procedures
- Time to fix inconsistency
- Availability of business processes
• Number of inconsistencies
• Number of alerts
5 Training

E2E300 - Business Process Integration and Automation Management
This course provides training for members of a customer support organization. It covers a wide range of topics including data consistency and how to implement the SAP Standard for Data Integrity and Transactional Consistency.

Course content
- Introduction to Business Process Operations
- Job Scheduling Management
- Business Process and Interface Monitoring
- Data Consistency Management
  - Using Data Consistency Monitoring
  - Using the Data Consistency Toolbox
  - Using Cross-Database Comparison in SAP Solution Manager
  - Using tools for transactional correctness
- Business Process Improvement
- Business Process Performance Optimization

For more information see the SAP training portal at https://training.sap.com
# 6 More Information

<table>
<thead>
<tr>
<th>Documentation</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Consistency Check for Logistics</td>
<td><a href="https://websmp105.sap-ag.de/~sapidb/011000358700001955832008E">https://websmp105.sap-ag.de/~sapidb/011000358700001955832008E</a></td>
</tr>
</tbody>
</table>